DIRECTING VIRTUAL HUMANS USING PLAY-SCRIPTS AND SPATIO-TEMPORAL REASONING

by

Christine Talbot

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2018

Approved by:

____

Dr. G. Michael Youngblood

____

Dr. Richard Souvenir

____

Dr. Jing Xiao

____

Dr. Kalpathi Subramanian

____

Dr. Andrew Hartley

ABSTRACT

CHRISTINE TALBOT. Directing Virtual Humans Using Play-Scripts and Spatio-temporal Reasoning. (Under the direction of DR. G. MICHAEL YOUNGBLOOD)

Historically, most virtual human character research focuses on realism/emotions, interaction with humans, and discourse. The majority of the spatial positioning of characters has focused on one-on-one conversations with humans or placing virtual characters side-by-side when talking. These rely on conversation space as the main driver (if any) for character placement.

Movies and games rely on motion capture (mocap) files and hard-coded functions to perform spatial movements. These require extensive technical knowledge just to have a character move from one place to another. Other methods involve the use of Behavior Markup Language (BML), a form of XML, which describes character behaviors. BML Realizers take this BML and perform the requested behavior(s) on the character(s). Also, there are waypoint and other spatial navigation schemes, but they primarily focus on traversals and not correct positioning. Each of these require a fair amount of low-level detail and knowledge to write, plus BML realizers are still in their early stages of development.

Theatre, movies, and television all utilize a form of play-scripts, which provide detailed information on what the actor must do spatially, and when for a particular scene (that is spatio-temporal direction). These involve annotations, in addition to the speech, which identify scene setups, character movements, and entrances /exits. Humans have the ability to take these play-scripts and easily perform a believable scene.

This research focuses on utilizing play-scripts to provide spatio-temporal direction to virtual characters within a scene. Because of the simplicity of creating a play-script, and our algorithms to interpret the scripts, we are able to provide a quick

method of blocking scenes with virtual characters.

We focus on not only an all-virtual cast of characters, but also human-controlled characters intermixing with the virtual characters for the scene. The key here is that human-controlled characters introduce a dynamic spatial component that affects how the virtual characters should perform the scene to ensure continuity, cohesion, and inclusion with the human-controlled character.

The algorithms to accomplish the blocking of a scene from a standard play-script are the core research contribution. These techniques include some part of speech tagging, named entity recognition, a rules engine, and strategically designed force-directed graphs. With these methods, we are able to similarly map any play-script's spatial positioning of characters to a human-performed version of the same play-script. Also, human-based evaluations indicate these methods provide a qualitatively good performance.

Potential applications include: a rehearsal tool for actors; a director tool to help create a play-script; a controller for virtual human characters in games or virtual environments; or a planning tool for positioning people in an industrial environment.

ACKNOWLEDGEMENTS

Over the last several years, I have received support and encouragement from a great number of individuals. First, I would like to thank Dr. G. Michael Youngblood, my advisor and mentor, who has been an amazing teacher and has inspired me throughout my studies. He has offered his advice, time, and support from across the country to ensure I would be successful in completing my research. Without his guidance and persistent help, this dissertation would not have been possible.

I would also like to thank my dissertation committee of Dr. Richard Souvenir, Dr. Jing Xiao, Dr. Kalpathi Subramanian, Dr. Jamie Payton, and Dr. Andrew Hartley for their support from my proposal through to my dissertation's completion. They have provided key insights and feedback that have been instrumental in completing my final stages of research.

During my idea brainstorming, experiments, and paper writing, the members of the Future Computing Lab (now called the Vision and Image Analysis Lab) at the University of North Carolina at Charlotte have been great sounding boards and supporters of my work. Each of them provided the needed encouragement, support, and feedback that kept me moving forward. An extra thanks goes to Dr. Rich Nguyen for his extra support near the end when I did not think I would get enough study participants.

In addition, a special thank you to both Dr. Richard Souvenir and Dr. Dale-Marie Wilson for inspiring me to start my research journey. Without them, I would never have started my research. Also an extra special thank you to Julie Youngblood who has been very supportive in both handmapping numerous scenes and participating wherever needed with my research.

I would also like to thank my family for believing in me and supporting me while working on my dissertation. They have been understanding in the time I have had to dedicate to completing papers and research, including when I have brought it along for the ride. They have listened to my ramblings and stress about completing my work, even when they had no idea what I was talking about.

Finally, I would like to thank my employers, TIAA-CREF and Salesforce, for funding my studies, and enabling me to embark on this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Automatically positioning characters in a scene is a difficult task. Because of this, most virtual character research focuses on realism/emotions, interaction with humans, or discourse. This leaves the majority of the positioning of characters to be one-on-one conversations with humans or placing virtual characters side-by-side while talking. These rely on conversation space as the main driver (if any) for character placement. Once placed, these characters do not tend to move.

The current state-of-the-art includes capabilities such as the Institute for Creative Technology's (ICT) Virtual Human Toolkit (VHT), which provides discourse and emotion engines for interacting with characters [35]. It includes SmartBody, which can provide steering, facial animations, and grasping capabilities, but does not provide true positional logic. Examples of these tools can be seen in ICT's applications, such as Ada and Grace (at the Museum of Science in Boston), who converse with each other and guests within a museum setting (Figure 1.1) [103].

In addition, the ICT team utilizes high fidelity characters, generated by Paul Debevec and their Light Stage technology, as seen in Figure 1.2 [21]. Even with these possibilities to make realistic looking and acting characters, we are unable to easily position these characters realistically in an environment. The ability to interact or go for a walk with a virtual version of a friend or family member seems to be within reach, if only we could position those avatars within the virtual environment and portray their spatial personalities without a large engineering effort.

The question then becomes, how can we both realistically, and easily position characters in a virtual environment?

Today, there are limited capabilities for automatically positioning characters in a

Figure 1.1: Interview with Ada and Grace, Museum of Science in Boston's Virtual Human Museum Guides [103]

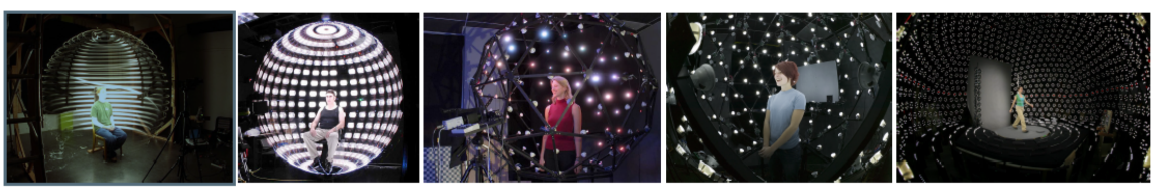


Figure 1.2: Left to Right: Light Stage 1's spiraling spotlight records a reflectance field in 60 seconds; Light Stage 2 records actor Alfred Molina for Spider-Man 2; Light Stage 3 illuminates an actor with a reproduction of the colorful light of the Grace Cathedral High-Dynamic-Range Image (HDRI) map; Light Stage 5 uses high-speed photography to record an actor's reflectance with time-multiplexed illumination; Light Stage 6, at 8m in diameter, allows performance relighting for the whole human body [21].

scene in a virtual environment. Most efforts just position characters side-by-side and ignore any spatial interactions. What work has been done, relies on positioning virtual characters within a scene to support the current actions being performed. This work is focused on positioning and orienting virtual characters to make the characters seem more realistic.

Industry has also utilized animated and virtual characters based on real actors' movements recorded via motion capture (mocap) files. This group comes closest to taking into consideration the implications of spatial reasoning for controlling the virtual characters. This method of recording motions as they are being performed by actors provides intricate details for replaying the motions. However, it comes with several drawbacks, such as expensive tools, good actors, and the creation of realistic environments to perform in. It is not very dynamic and every situation must be recorded for the exact situation being simulated.

The gaming industry relies on modularized low-level code to move characters about in an environment. This requires extensive technical skill to translate high-level actions, as well as extensive time to write all of that code. Most movement is hard-coded on what can be done and when it will occur.

A newer option includes a Functional Markup Language (FML), Behavior Markup Language (BML), and BML Realizers like SmartBody (Figure 1.3). These also require some lower-level coding, but begin to abstract and parameterize the motion of the characters, creating a more dynamic and repeatable motion for the characters.

The problem is that this method still requires a game-writer to write very specific and detailed steps. With BML, one must specify where the character looks, when they look there, how they move, when they move, and when they should pick-up or put-down objects. This can be very time-consuming, even though not everyone is doing this by hand. For instance, its primary uses are to generate characters that emote or move robots around to complete tasks.

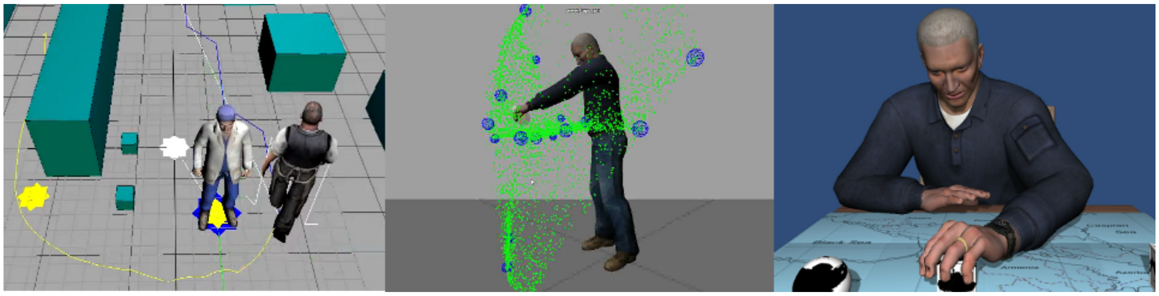


Figure 1.3: SmartBody is a BML Realizer that provides locomotion, steering, object manipulation, lip syncing, gazing, and nonverbal behavior in real-time [26].

So we can move the characters, but how do we impart human spatial tendencies, along with high-level directions to our characters? We know that as a human, our approach for giving directions is much more vague than any of the approaches we have mentioned above. For instance, we do not specify common-sense things, such as a road curves left while you are following it; therefore you should curve left too. We also do not remind people to take the elevator, and press the button numbered three, in order to reach the third floor.

Why can we not control characters with a similar-level of detailed directions, yet obtain natural and realistic looking positioning that can be obtained with mocap files? While asking this question, we observe that almost all theatre and movie productions provide this type of instruction to their actors via a play-script. Actors are given high-level information on where to go, what to do, and what to say. From there, they are able to provide natural and realistic movement on stage or camera. These play-scripts are written in natural language, and most people have had at least some exposure to them through Shakespeare in school. Because they are written in natural language, it is expected that they are easier and faster to create than BML for specifying positional information.

As we form our hypotheses, we want to note that most of these hypotheses are looking for nothing to be statistically different, or for the only statistically significant differences to be skewed in one particular direction. This is atypical to most null-hypothesis construction, since we are looking for nothing to be significant. This will show that there is no discernable difference between human performances and performances completed with our techniques.

Hypothesis 1.1. *A computational algorithm using annotations in a play-script can provide similar positioning of virtual characters as a real actor directed by a human.*

This hypothesis can be evaluated by comparing the positioning of characters from a real performance versus the use of natural language processing, such as named entities and part of speech tagging, on the same play-script. Similar positioning can be defined as within the same general area of the stage, or the same general direction of their gaze. In the work presented in this thesis, we found that we could position the characters 78% of the time matching a real human performance of the same scene, using just these natural language processing techniques.

Play-scripts provide most of the direction and motivation to the actors regarding the director's intended interpretation of the play. The annotations in the script describe what an actor should be doing, as well as when and how. Because play-scripts typically consist of short, to the point, directions from the playwrite (often as sentence fragments), actors are required to apply their interpretation for any gaps. They do this, just as humans infer details from vague directions. Therefore, it is speculated that additional logic, or rules, are required to better interpret a play-script. Some of these might include typical conversational space, theatre rules, or other common-sense conventions.

Hypothesis 1.2. *An algorithm-based director can improve character positioning of virtual characters within a scene if rules are applied to the movements defined in the annotations.*

This hypothesis can be evaluated by comparing the positioning of characters from a real performance to both the natural language processing techniques from Hypothesis 1.1, and a rules engine applied to the same play-script. These rules should improve the matching of general position and gaze for each of the characters by applying similar rules as most actors apply to the annotations they read in a play-script. With these rules, we increased our position matching to 89.8%, and our gaze matching to 53% for the same scenes.

We realize that actors may also take it upon themselves to improvise with a script. We conjecture that perhaps there is something in what the character is saying that cause the actors to perform this extra, unannotated movement. In addition, it is speculated that what the actor(s) are saying may also impact what movement is performed within a scene, and can be inferred by what is said by the characters.

This conjecture can be evaluated by utilizing a real performance's movements and speech lines to learn, then apply that learning to new speech lines. Learning a pattern that determines the future movements in the play-script would provide key insight into what actors do on a regular basis with a play-script. As we explored this area, we found that using a Shakespearean play-script made it difficult to learn any implied movement. However, we believe it may be possible to infer movement utilizing other play-scripts or machine learning techniques.

In a play, actors arrange themselves on the stage according to both basic rules of the theatre, as well as with respect to the positioning of other actors on the stage. This is where we look towards force-directed graphs. These have been used for many years to display large and complex graphs.

Typically, they focus on information stored within the graph about the relationships between the nodes in order to place them on the screen. They have been shown to create both aesthetically pleasing and symmetric graphs. Some have even been shown to preserve edge crossings, minimizing the number of edges that cross each other based on the initial state of the graph [45]. They have been used for displaying social networks because of this easy-to-view layout that can group and organize nodes of the graphs. Some have even used them to show the relationships between actors on-stage [6]. While viewing examples of these drawn graphs, we noticed that several areas form a semi-circular arrangement, closely mimicing a conversational circle. This lead us to hypothesize:

Hypothesis 1.3. *Force-directed graphs can position characters onstage with typical conversational arrangements, avoiding character occlusion.*

This hypothesis can be evaluated by comparing a human-performed performance versus a performance that applies force-directed graph display techniques. The characters should arrange in a circular arrangement, facing the audience, and be within a typical conversational distance from the other characters. They should also avoid occluding the other characters within the scene. Therefore, we will measure the amount of clustering of the characters on the stage, and any occlusion that occurs. We found that the force-directed graphs provided consistent semi-circular arrangements, and evenly spaced characters on the stage.

However, this task becomes more challenging when we do not control all the characters in the virtual world, such as a human-controlled character. This becomes more critical when arranging a mix of human- and AI-controlled characters. Humans do not always follow predictable patterns, and virtual characters must be able to react appropriately (spatially) within the environment.

A simple example of this is within theatre productions as a virtual environment. In real life, actors arrange themselves on the stage according to both basic rules of the theatre, as well as with respect to the positioning of the other actors on-stage. Humans may not always hit their mark like they should, and may move when they are not supposed to, or may not even move at all during the play. This presents issues with the blocking within the play, as the other AI characters on-stage are assuming that the human followed the script. If the agent-controlled characters do not adjust, they could create unrealistic positioning of the characters based on the standard rules of thumb for theatre, but also could obstruct visibility to themselves or the human-controlled character for the audience. In video games, there is also a desire to adjust the positions of the agent-controlled characters based on where the human-controlled character is, in order to provide better visibility (or less visibility) of those characters.

Hypothesis 1.4. *Force-directed graphs can better incorporate human-controlled characters with a set of virtual characters, adjusting the virtual character movements around the human's motion, than a performance done only with the play-script and applied rules.*

This hypothesis can be evaluated by comparing the closeness of characters utilizing force-directed graphs versus just following the play-script and applying rules, when a human-controlled character is included in the scene. We found that utilizing the force-directed graphs reduced the overall occlusion of characters onstage, while increasing the clustering of the characters, even when the human-controlled character did not follow the script.

Once combined, these techniques should be able to appear realistic to a user. Even though the techniques may not provide an exact match for how an actor would perform the script, it should be imperceptible to the typical user, and perceived as a good performance.

Hypothesis 1.5. *An algorithm-based director, using a combination of play-scripts, rules, and force-directed graphs, can equal or surpass the human-perceived threshold of a quality performance for a variety of spatio-temporal play types.*

This hypothesis can be evaluated by performing user studies that compare similar scenes with each of the techniques for their spatial positioning, and evaluating the viewer's perception of goodness. The viewer should perceive the force-directed graph-driven performance as a reasonable performance, as good as an actual human-performed version of the same scene.

Also because these techniques are generic in nature, requiring only basic initializations of characters, pawns, marks, and environment layout, they should be applicable to any play-script.

After defining the appropriate spatio-temporal dimensions of a play, these techniques should provide similar evaluations both quantitatively and qualitatively as our initial evaluations. This hypothesis can be evaluated by both quantitative analysis of positioning for each of the play types and user studies for a qualitative analysis. We identified five spatio-temporal dimensions defined in Chapter 7: GENERALIZATION to define the space of all play-scripts, and found seven play-scripts that provided 100% dimensional coverage, and 71% pairwise coverage of this space. Upon evaluating these techniques for each of these play-scripts, we found that we could match the blocking of the human performance 58% of the time, and provide a similarly qualitatively "good" performance from the viewer's perspective.

In the remaining sections of this document, we will discuss the background, related work, methodology, systems, and experimentation to prove or disprove these hypotheses. Additionally, we will wrap up with our conclusions and what additional, or future, work has been inspired by this research.

CHAPTER 2: BACKGROUND

When pursuing solutions to positioning characters within virtual environments, there are a few key concepts that are helpful to understand. These include: giving human directions, formatting and content of play-scripts, theatre rules, Shakespeare plays, and specifically Shakespeare's *Hamlet* on Broadway in 1964. Here, we will review some important concepts in these areas, which will provide an appropriate background for our approach and decisions used while solving this problem.

2.1 Human Directions

When we give directions to people, we often have a layer of implied meaning built into it. For example:

> **A:** Excuse me...
>
> **B:** Can I help you?
>
> **A:** Where is the conference room?
>
> **B:** Go down the hall and take the elevator to the fourth floor.

Implied in these directions are things like how far is it to the end of the hall; the elevator is within sight when you get there, so you do not mention you have to turn right and go a few feet to the elevator; you do not instruct them how to work the elevator, you assume they know to press the button and wait for it to arrive. As you can see, directions are usually vague, yet they are still sufficient for people to figure out how to get from point A to point B.

2.2 Play-Scripts

In play-scripts, there is a similar level of abstraction and assumptions within the director annotations as we use in every-day language. Play-scripts provide a natural way of directing actors and characters, including any relevant spatial directions. They are written in natural language, but are typically short and to the point statements to instruct an actor on their actions and movements. These scripts follow a relatively standard format, which includes three different types of stage directions.

Scene Directions Overall scene directions will be indented to the right of the page, surrounded by parentheses. It will provide the basics of where and when the scene is set, what is happening as the scene begins, and so forth [1]. An example can be seen in Figure 2.1.

SCENE 2

(A distant bell tolls. Four Courtiers enter from the right and carry the table to stage center; they turn the table on its side, with the top facing the audience, to represent a barricade in front of a grave. The stool is placed at the right end of the table, the armchair is removed. Courtiers exit.)

(The GRAVEDIGGER1 enters through the center doors carrying a lantern. He is followed by GRAVEDIGGER2, who carries a T-spade and a pick and whistles, ironically, "Tomorrow is Saint Valentine's Day.")

Figure 2.1: Scene Directions Formatting Example

Staging Directions Basic staging directions, which describe actions during the scene such as entrances, exits, movements, and so forth are also going to be surrounded by parentheses and on their own line(s) [1]. See the example in Figure 2.2.

GRAVEDIGGER1
(Digs and sings)
In youth when I did love, did love, Methought it was very sweet,

(HAMLET and HORATIO enter center, cross to the side steps, and watch him, amused)

GRAVEDIGGER1
(Digs and sings)
To contract, oh the time for-a my behove, O me thought there-a was nothing a-meet.

Figure 2.2: Stage Directions Formatting Example

Character Stage Directions Character stage directions relate to a particular character and provide details on what they should do as they speak their line. These will follow similar formatting to the basic staging directions by being indented and surrounded by parentheses [1]. See the example in Figure 2.3.

HORATIO
(Laughing)
Aye, my lord.

Figure 2.3: Character Stage Directions Formatting Example

The dialogue for the characters will be in regular text, prefixed by the character name in all caps. To summarize, Figure 2.4 shows how these fit together. Specific formatting standards include:

1. Every time you mention a character in the stage directions their name should be in ALL CAPS. This makes it easier for the actors, director, and team to scan the page and find what the actors are doing.

2. Stage directions are always enclosed in parenthesis.

3. Stage directions show only what is taking place on stage (what the audience can hear or see), they do not tell the interior life or previous life of people or

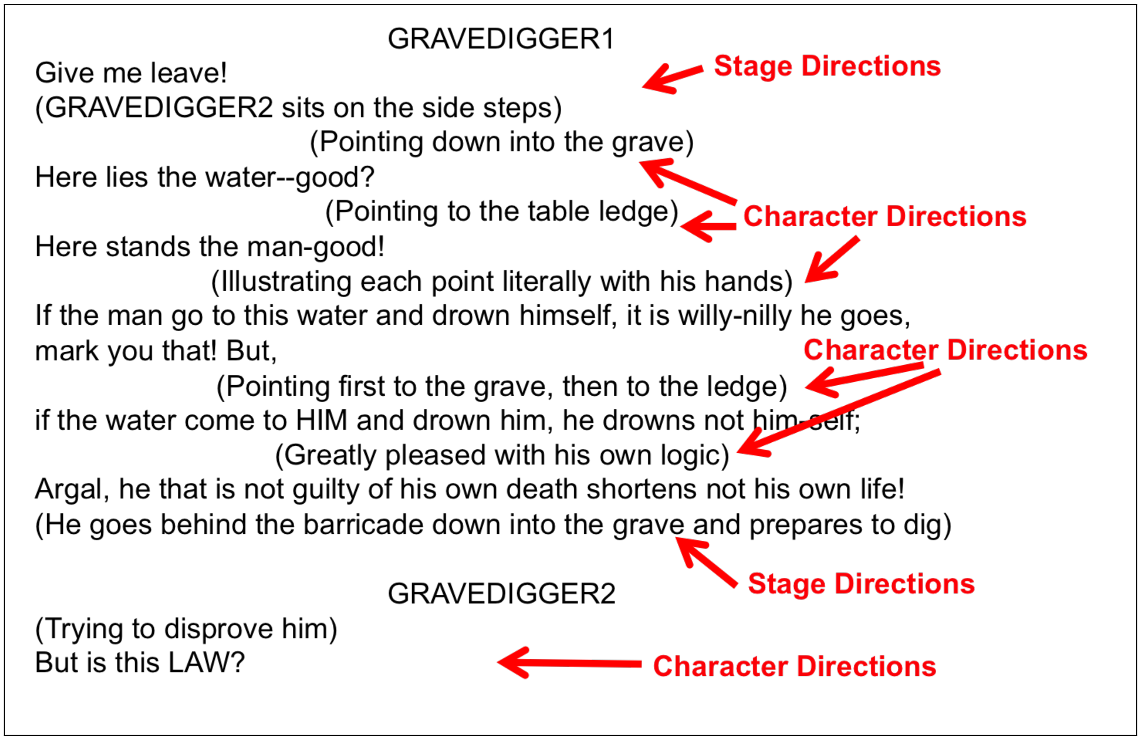


Figure 2.4: Play-Script Excerpt from Sir John Gielgud's *Hamlet* on Broadway 1964

objects. [1]

2.3 Theatre

The scripts tend to utilize stage directions, such as stage left, center stage, and upstage, along with specific marks and props to guide the actors to appropriate locations. Background theatrical knowledge is also applied to cover some of the hidden rules behind performing these scripts, such as avoid putting your back to the audience, try to keep towards center stage as much as possible, primary characters should be closer to the audience than secondary characters, and general personal space and conversational rules.

In the theatre, there are special rules and conventions when staging a play. Many of these guidelines revolve around engaging the audience and visibility onstage. To help with this, the stage is often split into nine areas, upon which basic theatre rules are based. They consist of upstage, stage right, stage left, downstage, and combinations

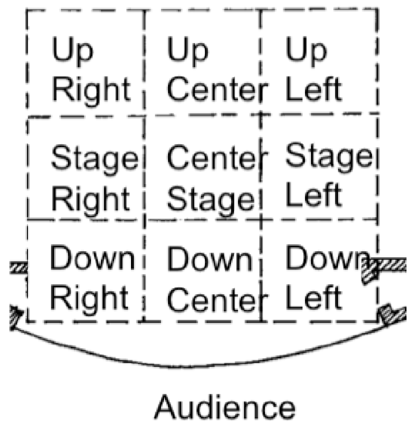


Figure 2.5: Stage Layout

of each as shown in Figure 2.5.

Being downstage (near the audience) is a stronger position than being upstage and should be held by the most important characters in the scene. Also, because we tend to read left to right, downstage right is the most powerful position onstage as audiences tend to look left first, then scan right when watching a play. The more important a line is, the more likely an actor is to fully face the audience, although the most common position is a one-quarter (or 45° angle from the audience) body position as it ensures the audience can see all the characters on the stage properly. Actors should never turn their back to the audience. [5]

Moving onstage can cause many issues including upstaging and covering. Both of these issues should be avoided, which in turn provides additional rules to characters on the stage. Upstaging is where one actor takes a position further upstage, or above a second actor, which causes the second actor to face upstage/away from the audience. Therefore this must be avoided to ensure actors do not present their backs to the audience, especially if both characters are just as important to the scene [55].

Covering occurs when one actor blocks the audience's view to a second character onstage. If this does happen, the covered actor should adjust to provide visibility of him/herself to the audience by counter-crossing (performing a movement in the

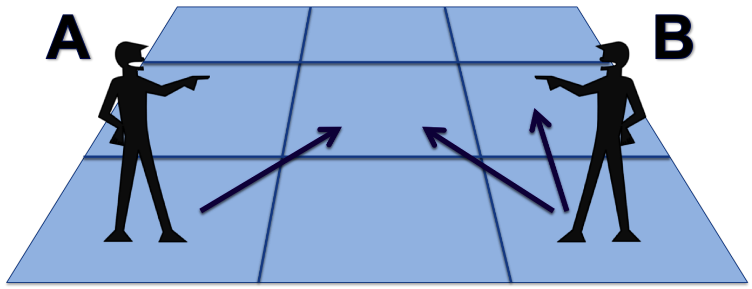


Figure 2.6: Example of a Counter-Cross: Actor B Could Move to Center Stage or to Right Stage to Counter Being Upstaged by Actor A

opposite direction of the other actor—see Figure 2.6). When making these changes, actors should cross downstage from other actors unless their movement should not be noticed by the audience. Finally, when crossing the stage, it will take two separate crosses (movement from one area of the stage to another) to cross upstage–one to the left or right, turn in, then the second to cross upstage [5].

Additional theatre terminology and definitions can be seen in Appendix B: DEFINITIONS.

2.4 Shakespeare

Shakespearean plays happen to be a genre with very few director annotations in them, unless you can find a director's annotated version. This leads to very different interpretations of his plays, and may contribute to their popularity even after over 400 years [60]. Modern plays tend to have more annotations included in the published versions than the original Shakespearean plays.

William Shakespeare has written at least three of the top ten most-produced plays in North America, despite the fact that most lists explicitly exclude Shakespeare's plays from their top ten lists as it would be unfair [39].

Initially, we focus on one particular famous production of one of Shakespeare's plays from 1964. Sir John Gielgud directed *Hamlet* on Broadway with Richard Burton playing Hamlet. This production ran for 138 performances, setting the record as the longest-running *Hamlet* ever to play New York [86]. It was filmed during three successive stage performances in June/July 1964 by Electronovision, Inc. [15]. In addition, Richard Sterne (another actor in this particular production) published a book with very detailed director's annotations and notes for the entire play [86]. We leverage this particular performance as our baseline in our initial studies because it is well-known, and has been established as a qualitatively "good" performance to compare our techniques against.

CHAPTER 3: RELATED WORK

The work in this document focuses on creating an Artificially Intelligent (AI) Director to spatially position characters in virtual environments, utilizing psychology's spatial preposition research findings, natural language processing, play-scripts, robotics influences, theatre rules, and force-directed graphs. The sections of this chapter will explore the related work in these areas.

3.1 Motion Capture Files

Canned/explicit cut scenes are very common in games, films, and virtual environments. This is often accomplished via motion capture (mocap) files, which are typically outputs of sensors on humans performing the required actions. This comes closest to taking into consideration the implications of spatial reasoning for controlling virtual characters. Their methods of recording motions as they are being performed by actors provide intricate details for replaying the motions. However, it comes with several drawbacks, such as expensive tools, good actors, and creation of realistic environments to perform in. It is not very dynamic and every situation must be recorded for the exact situation being simulated.

L.A. Noire, a violent crime thriller game, and the Avatar movie, have both used this animation technology that captures every nuance of an actor's facial performance in extreme detail, as well as the body movements and position in space [32].

The gaming industry relies on modularized low-level code to move characters about in an environment. This requires extensive technical skill to translate high-level actions, as well as extensive time to write all of that code. Most movement is hard-coded on what can be done and when it will occur.

3.2 Dialogue Trees

Dialogue trees are used in many games and applications, providing options for dialogue, based on what has been said before [22]. Handling all the possible branches possible throughout a game can be challenging to author, so in Rich and Sidner's work, they look to provide a method for partially generating dialogue trees by leveraging a hierarchical task network to capture the high-level goals of large dialogues [77]. Ultimately these trees focus just on the speech that should occur, however they could be expanded to include spatial information, which could be used with our techniques described here.

3.3 Markup Languages

Current methods such as Behavior Markup Language (BML) [63], Functional Markup Language (FML) [101], and BML Realizers like SmartBody [26] and Elckerlyc [113] are making it possible to abstract the control of virtual characters. However, these methods still require a level of expertise and time that can be unreasonable. Writers must be fluent in these technical languages, plan out specific points and marks within the environment, and convert the more fluid, natural descriptions into more concrete commands with fewer human assumptions. However, it begins to abstract and parameterize the motion of the characters, as well as creates a set of more dynamic and repeatable motions for the characters.

3.3.1 Behavior Markup Language

Markup languages, such as Behavior Markup Language (BML) [63], are making it possible to abstract the control of virtual characters. BML abstracts the physical realization of behaviors and movements, along with their constraints, and is not concerned with the intent behind the movements. [63] BML is structured like a typical XML message, as seen in Figure 3.1. One can control what is done, when it is done, and what runs concurrently with other commands. However, it is often at such

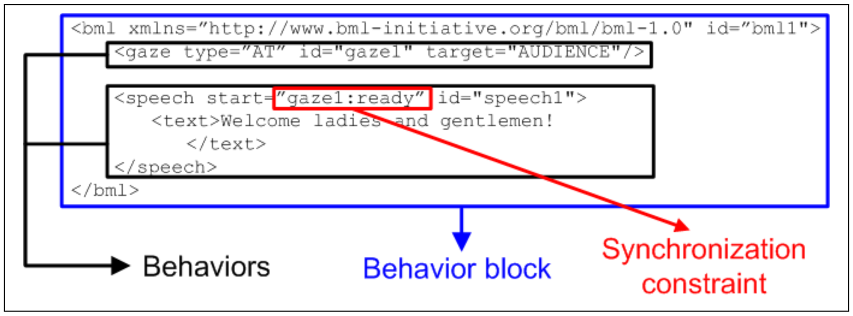


Figure 3.1: Example of a BML Request

a low-level that this can be extremely time-consuming to build, especially for things like non-verbal behaviors (eye saccade, gesturing while speaking, head nods, and so forth).

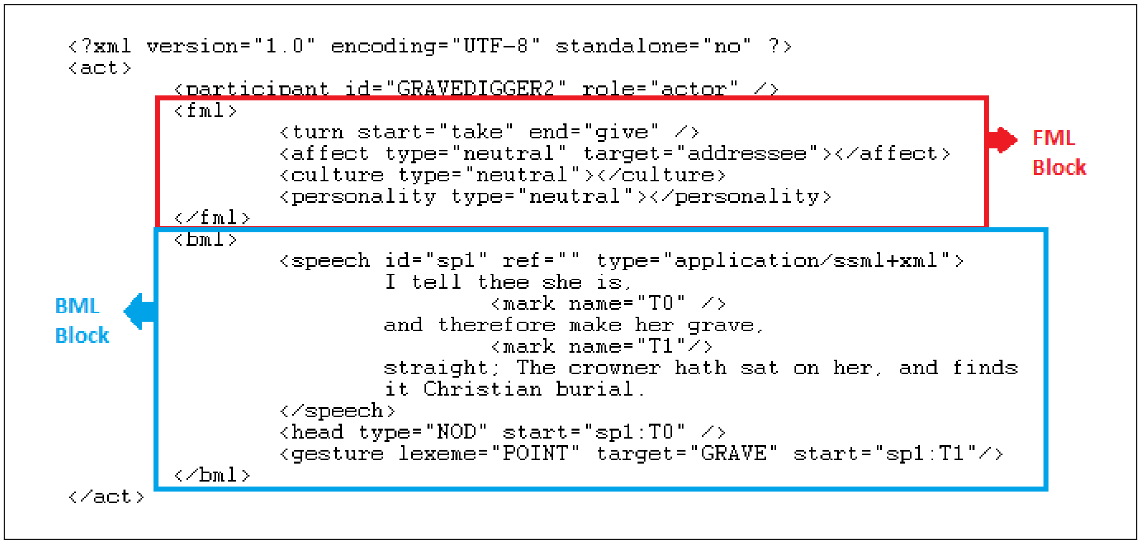
Per the BML standards:

> BML describes the physical realization of behaviors (such as speech and gesture) and the synchronization constraints between these behaviors. BML is not concerned with the communicative intent underlying the requested behaviors.[63]

BML consists of a block of XML that contains a listing of behaviors for a particular character. Within each behavior block are constraints and attributes regarding when and how a behavior should be performed with respect to the other behaviors. There are BML Realizers, such as SmartBody [26] or Elckerlyc [113], which execute behaviors specified by BML on the character in the environment. However, BML realizers are still their early stages of development.

3.3.2 Functional Markup Language

To help with the intent and translation of some nonverbal behaviors while speaking, a Functional Markup Language (FML) has been proposed. The FML should "describe the effect that an intended action or plan should have on the environment, most obviously the agent itself [101]". The Non-Verbal Behavior Generator (NVBG)



```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<act>
        <participant id="GRAVEDIGGER2" role="actor" />
        <fml>
                <turn start="take" end="give" />
                <affect type="neutral" target="addressee"></affect>
                <culture type="neutral"></culture>
                <personality type="neutral"></personality>
        </fml>
        <bml>
                <speech id="sp1" ref="" type="application/ssml+xml">
                        I tell thee she is,
                                <mark name="T0" />
                        and therefore make her grave,
                                <mark name="T1"/>
                        straight; The crowner hath sat on her, and finds
                        it Christian burial.
                </speech>
                <head type="NOD" start="sp1:T0" />
                <gesture lexeme="POINT" target="GRAVE" start="sp1:T1"/>
        </bml>
</act>
```


Figure 3.2: Example of an FML and BML Request

module in the Virtual Human Toolkit (VHToolkit) [43] utilizes these FML commands along with rules to generate BML with nonverbal behaviors inserted into the speech text [51]. Although a standard has not been set for FML yet, this NVBG module's FML is combined with BML and follows the syntax seen in Figure 3.2.

This becomes useful for allowing the author to make assumptions about some of the lower-level actions their characters must make, but still requires a level of expertise in FML and a generation of those personality and culture rules. The Non-Player Character Editor (NPCEditor) [52] component of the VHToolkit provides a utility to translate the questions asked by a user to the answers, which can then be passed on to the NVBG module, which adds the nonverbal behaviors to the response for the character to perform. This may be useful in our work, if it gets expanded to encompass translations to spatial movements as well.

3.3.3 Perception Markup Language

A Perception Markup Language (PML) has been suggested, primarily to assist with robotics feedback loops. PML provides a method for a character (or robot) to react and interact with the environment. One engine, Thalamus, which is based on the existing Situation, Agent, Intention, Behavior, Animation (SAIBA) framework, splits out the behavior scheduling and the behavior execution. This enables it to interrupt the robot's behavior plan based on the perceptions that are sent to it via PML. This allows the support of on the fly changes to the robot's behaviors based on its perceptions, such as an obstacle. [76]

3.4 Robotics

Roboticists have pursued an understanding of spatial language primarily to understand verbal instructions for controlling robots within natural environments. This can be seen in many works, such as Brooks's thesis where he attempted to train a robot to be an actor using verbal directions. The robot could not speak, but shrugged if he did not understand the directions [9]. This is a different approach to teaching a character to enact a scene of a play; however, Brooks' approach required a more detailed and lower-level of communication to his robot than is typically found in a play-script. David Lu and Bill Smart's work with robots in theatre has focused around mimicking actor's movements with robots to help incorporate social interactions into robots without explicitly programming them [59]. They used actors to record specific scenarios and replicated them on robots, making their movements more believable. These were generalized to similar situations and to robots that could not physically replicate the original motions. The focus in their work is on believability; however, this work is based more on a motion capture-like style of replaying actions done by a human and does not address our concerns with dynamically positioning multiple characters without pre-recording.

Langley, Schermerhorn, and Scheutz also provide an approach to human-robot interaction that allows for communicating complex tasks, which can be translated into procedures for the robot [104]. Matuszek and Herbst take natural language and robotic perceptions and translate it into a robot control language for following route directions [65]. Dzifcak, Scheutz, and Baral utilize natural language to determine actions and goals for the robot [25]. All of these incorporate telling a robot what to do or where to go.

3.5 Virtual Agents

The focus of much research has involved virtual characters; however, very little of this work has investigated spatial movement of those characters. The emphasis appears to be more on the speech and emotional interaction with humans or other characters. For instance, Dias proposed changes to the FAtiMA (FearNot Affective Mind Architecture) architecture to include the skill of understanding emotions of others in determining next steps [24]. The FAtiMA architecture was built to create autonomous believable characters that allowed the establishment of empathetic relationships with other characters in the FearNot! system [23].

Then there are things like the Virtual Storyteller, which enables characters to tell a story with the appropriate gestures, prosody, and so forth [99]. Here, along with others, they focus on plot and story creation, mostly in the area of interactive storytelling. For instance, Kriegel proposes a design to help solve the authoring problem for interactive storytelling utilizing the FAtiMA architecture [47]. Thespian expands on these to reduce the programming effort for the speech actions of a story by pre-authoring sections of the speech and utilizing goals to control choices by the characters [82].

Other research utilizing virtual agents focuses primarily on the conversational and nonverbal domains, such as Thespian [83], Virtual Storyteller [99], and Stability and Support Operations (SASO) [43]. The emphasis appears to be more on the speech and emotional interaction with humans or other characters. However, with the growing focus on realistic virtual environments, the spatial domain is becoming a more critical component in creating that realism.

However, these do not emphasize the spatial aspects of the interactions between multiple characters. They center around the emotional and one-on-one interactions of characters with humans in the real world.

3.6 Natural Language Processing

In the natural language processing community, many researchers are working towards better understanding of the written and spoken word. There is quite a bit of work in niche areas for natural language understanding, such as a focus on spatial language expressions. These examine different prepositions, which indicate the temporal, spatial, or logical relationship of objects to the rest of the sentence (e.g., *in, on, near, between*). For instance, Regier built a system that assigns labels such as "through" or "not through" to movies showing a figure moving relative to a ground object for learning how we qualify the particular term "through" [36]. Kelleher and Costello [41] and Regier and Carlson [75] built learned models for the meanings of static spatial prepositions such as "in front of" and "above" while Tellex focused on "across" [98].

Some groups are pursuing the complexities of spatial cognition within language on object representations and geometry, as well as the number and structure of the objects utilizing the prepositions that situate them in space [50]. Kelleher also proposed a framework for understanding prepositions primarily around the closeness of objects and the visual representation of those objects [41]. Her research explores how humans describe where objects are within space, which is the key in extracting spatial information from natural language. This information has been used by other methods, such as WordsEye, which takes natural language to draw a scene utilizing the spatial locations described in text [19].

From the perspective of cognitive psychology of language, Coventry describes spatial language and how humans describe different situations using prepositions, such as a pear being in a bowl or not. He elaborates with many different prepositions such as *in*, *on*, *near*, *far*, *at*, and *between* [18]. However, these prepositions are very dependent on the frame of reference used for the spatial description. Describing spatial locations using an intrinsic, absolute, or relative frame of reference can dramatically change the interpretation of the same sentence [53]. Stating "a ball is in front of the chair" can mean different things depending on which way the object is facing, where the observer is, or what global spatial reference that is being used—all with respect to which reference the person describing the spatial relationship is using.

Once we are able to determine the frame of reference being used for the spatial descriptions, we can utilize methods of mapping objects based on cardinal directions as described in Frank's work [28]. Other methods include the use of spatial templates to identify acceptable locations with respect to a given object for a particular preposition [57], and vector sum models [75] to formalize spatial relationships.

3.7 Psychology and Spatial Cognition

Conversational space, spatial prepositions, and group dynamics have been studied for years in psychology. A lot of their work around personal space and conversational space will be extremely useful in applying our spatial logic. For instance, Jan and Traum describe six different forces that affect when/why a person may shift position when in a group of people:

- one is listening to a speaker who is too far and or not loud enough to hear
- there is too much noise from other nearby sound sources
- the background noise is louder than the speaker
- one is too close to others to feel comfortable
- one has an occluded view or is occluding the view of others [37]

Additional research shows that friendship and attraction can affect the spatial distances between people (decreases as attraction increases), while negative attitudes may not have much effect on the spatial distances [88], as seen in Figure 3.3. People also prefer to be across from one another than next to each other in most situations,

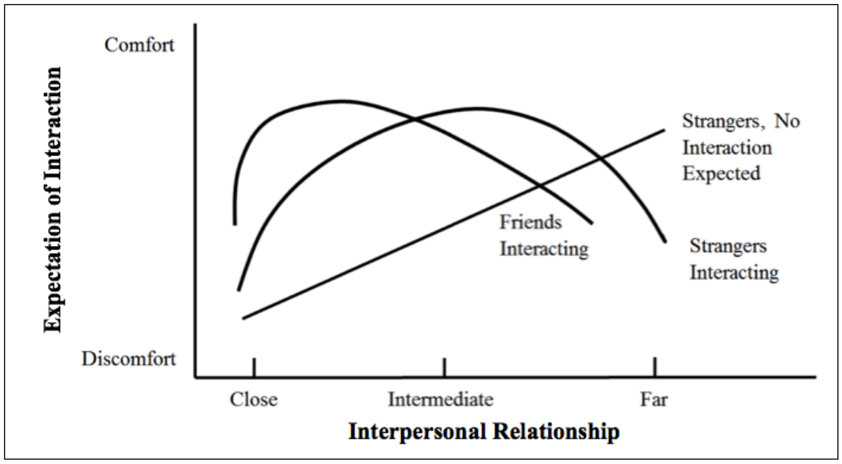


Figure 3.3: A Theoretical Model of Personal Space as a Function of Interpersonal Relationship and Expectation of Interaction[88]

but there is importance to the environment for determining what distance is comfortable, such as how far apart the couches are in the room [85]. According to studies reviewed by Sundstrom, comfortable face-to-face distances for speaking while sitting is approximately five feet, and comfortable face-to-face conversation while standing is approximately three feet [88]. There is also a discussion around the effects of spatial invasion on character behaviors and movements within Sundstrom's review.

3.8 Force-Directed Graphs

Force-directed graphs utilize repellent and attractive forces between connected nodes in a graph to spatially arrange graphs. Also known as string embedders, they utilize the information contained within the structure of the graph for placement of the nodes. The goals of force-directed graphs are to be aesthetically pleasing, meaning that all edge lengths should be the same length, and it should maximize symmetry over the entire graph layout.

Looking at some of the different implementations of force-directed graphs out there, we must start with Tutte's algorithm from 1963, which was one of the first force-directed graph drawing methods [105]. In his algorithm, he guarantees a crossings-free drawing and that all faces of the drawing are convex for a 3-connected planar graph. The forces in this model are proportional to the distance between vertices, with no repulsive forces, and places each free vertex at the barycenter (center of mass) of its neighbors. This is useful in our work since we are concerned with not obstructing the audience's view of all the characters on-stage. However, there are some results of this algorithm that produce a graph with infinite area [45], or would not place our characters within our stage's confines. Also ensuring 3-connectedness and a convex drawing may be challenging in a dynamic environment with a human-controlled character.

Fruchterman and Reingold's algorithm from 1991 introduces an equalization of vertex distributions. It calculates the forces between adjacent vertices as well as between all pairs of vertices, plus introduces the concept of temperature to reduce the amount of movement of vertices as the layout improves. This algorithm was targeted for small graphs, such as those with 40 or fewer vertices. Its cooling of movement via temperature is a specialized use of simulated annealing, which helps to limit oscillations of the layout. However the forces are based on the size of the grid that is to be drawn on, and therefore tries to maximize the real estate used. [29]

Then there is the algorithm by Kamada and Kawai, which tries to minimize the distance of vertices from their corresponding underlying graph distances [40]. This method requires more computation and storage space since it requires a shortest distance calculation on every vertex before running its minimization function [45]. This additional computation could take up to $O(|V|^3)$ time, and $O(|V|^2)$ storage, depending on the algorithms utilized for the shortest path computations. Even though we could calculate the underlying shortest distances for the graph ahead of time, we would need to adjust this each time a character is introduced into the scene or creates a new association to a targeted position onstage.

Also, we have some key relationships that encourage a character to hit their mark(s) and remain there until their next movement in the play. Kamada and Kawai's method would equally distribute the characters from each other as well as their marks, which is undesirable in the theatre.

There are also more complex force-direct graph drawing algorithms in existence that can accommodate tens and hundreds of thousands of vertices. These attempt to break down the graph into simpler structures, like Hadany and Harel [33] or Gajer, Goodrich, and Kobourov [31]. They often involve three-dimensional drawing of the graphs and zooming in order to provide visibility to the nodes of the graph. However, we are focused on very small numbers of vertices and a planar drawing area, so these do not provide much use for our current work.

Force-directed graphs have been used for many different purposes, like social networks, with Bannister et al's work. Their work attempts to centralize vertices that are more theoretically central in the graph [6]. This is interesting because of its close relationship to our work—visualizing relationships between nodes.

Network visualizations use force-directed graphs to help identify information about different clusters, and arranges graphs into symbolic shapes to help recognize the relative size of the clusters. These allow viewers to be able to estimate overall sizes of the graphs, as well as recall the layout of the graph at a high level. It does best with clusters of about eight vertices, and may not do well scaling to sparse clusters [80]. This could prove useful in arranging clusters of characters into particular shapes, such as a semi-circle.

3.9 Judging Criteria

There is a lack of existing tools to qualitatively evaluate the spatio-temporal reasoning within a performance. However, one-act play competitions are often critiqued by judges and include spatial aspects of the performance in their evaluations. Therefore, we reviewed their evaluation criteria for one-act performance competitions.

One group we looked at was the Georgia High School judging sheets for one-act plays. The criteria defined in the judges evaluation sheets included: movement, composition, listening, response, and ensemble criteria. Movement is an obvious tie-in to analyzing the spatial aspects of a performance, so it was included in our evaluation tool. The judges typically verify if the movement within the performance is motivated and free of distractions. With composition, the plays are evaluated on how the performers convey the theme and mood of the play, and whether the movements of the performers aid in providing proper dramatic emphasis. There is also a concern of the variety and balance in the use of the stage space included in the judges' checklist. Finally, reviewers are asked if the performers appear to work together and be involved in group events. [4]

We also reviewed the Texas University Interscholastic League's (UIL) one-act play official standards. The UIL's judging packet is much more comprehensive and included more detailed guidance on each of the criteria for evaluation a one-act play. Some important evaluations were described around characterization, movement, timing, business (exits and entrances), and composition. We added several questions regarding the believability of the characters' movements, whether the movement appears random, the overall pace of the performance, and whether the characters frequently blocked each other. [68]

Another source for evaluating performances is available via Pavis' survey to use when evaluating a performance. Her questions are more open-ended, and meant to guide the spectator in describing the aesthetic experience and overall production after seeing it. Some key spatio-temporal questions are included in Pavis's questionnaire, such as: space organization, relationships between actors, and pacing. [71]

Lastly, we referred to The Theatre Handbook, written in conjunction with several theatre groups: Independent Theatre Council (ITC), The Society of London Theatre, and Theatrical Management Association (TMA). This handbook provided useful recommendations around grouping questions for evaluating a performance's quality, such as the frequency of attending performances, and the use of self-rating with a newspaper's five-star scale. [69]

3.10 Planners

There are many planners out there, such as the one by Vidal that attempts to enable a planner and plan execution system to run concurrently, to support real-time requirements [110]. This work may assist with our translation of natural language to commands asynchronously from the actual execution by the realizer engine, enabling more real-time execution. Also, we may look at better movement target predictions utilizing the stage directions and details from Frank's work [28] on logic for geographic locations with respect to known object locations.

3.11 Crowd Modeling

Crowd modeling at first thought appears to be an appropriate approach to positioning characters. Upon further investigation, it can be seen that crowd modeling focuses more on modeling people's behaviors as opposed to the close-knit intricacies of the relationships between the characters onstage. It does not focus on spatially pleasing arrangements as a whole, but rather looks at each individual's contribution independent of the others. In theatre, the goal is to have the actors work together as a whole, not as independent entities, and thereby is not suitable for a theatre-type environment.

CHAPTER 4: METHODOLOGY

Now that we have reviewed what related work exists to support our work in positioning characters in virtual environments, we will discuss the techniques and approaches used in our work. We begin with a method to reduce the authorial burden for controlling characters in a scene by leveraging play-scripts. With these play-scripts, we extract the movements for the characters and translate them to Behavior Markup Language (BML). Next, we apply some basic rules, such as theatre and grouping rules, which actors subconsciously apply when performing a play-script. Lastly, we define some force-based algorithms to better dynamically arrange the characters on the stage, when applying the constraints provided by the play-script and rules from the previous components.

4.1 Natural Language Processing to BML

Because our goal is to decrease the authorial burden for producing scripted acts that involve spatial movements and actions, we will need to utilize some natural language processing to translate components of the play-script. As a first pass, we will look at parsing the spatial directions in the annotations (surrounded by parentheses) to determine the action within those statements and translate them into one of our spatial motions such as walking, pointing, gazing, picking up an object, and put down an object [58].

We utilized a simplistic natural language processor to identify the actor, what they are doing (of our identified spatial movements), and to whom/what they are doing that action to. Due to the nature of most play-scripts, we decided to focus on the basic noun-verb-noun structure of spatial commands within the script. Sentences are parsed to determine the verbs and nouns. The verbs and their synonyms are each reviewed against a list of synonyms for our key spatial movements (walk, turn, point, and pick-up/put-down). Meanwhile, the nouns and their synonyms are each reviewed against our known objects—Hamlet, Horatio, Gravedigger1, Gravedigger2, Shovel, Lantern, two Skulls, Stairs, Stool, Grave, and our nine basic stage positions (upstage left, upstage center, upstage right, center stage left, center stage, center stage right, downstage left, downstage center, downstage right). Taking the verbs and nouns we identify, we make the assumption that these sentences will take on the basic form of (actor, action, target). Using these triplets, we generate and send the Behavior Markup Language (BML) to our simulator to perform the action.

Generalizations were made in this approach due to our understanding of typical play-script contexts, including our simplistic sentence structures. Typically, director's annotations are short and to the point. Often, they are just barely sentences, if not sentence fragments. Therefore our expectation was that the sentence fragments would contain very little information outside of the actor, action, and target. Pronouns or unspecified actors were assumed to be the last speaker via the last Character line in the play-script.

Additional generalizations were made about the timing of these spatial events. All sentences, or sentence fragments, within a single set of parentheses were defined to be independent of each other and required to be acted upon at the same time. These were also to be performed with whatever the next speech action was, unless we were changing the speaking character. The basis for this generalization comes from a basic understanding of how scripts are acted and formatted. Directions are provided before, or in the middle, of whatever is being said by the characters. Items at the end of a speech usually complete before the next person speaks, otherwise it would appear at the beginning of the next character's speech.

These annotations were observed to take on a structure like Figure 4.1, and were parsed using Algorithm 1. The algorithm leverages the wordNet application for identifying the parts of speech and synonymns for the text included in the annotations [66]. These synonymns were then matched against our named entities to find a match.

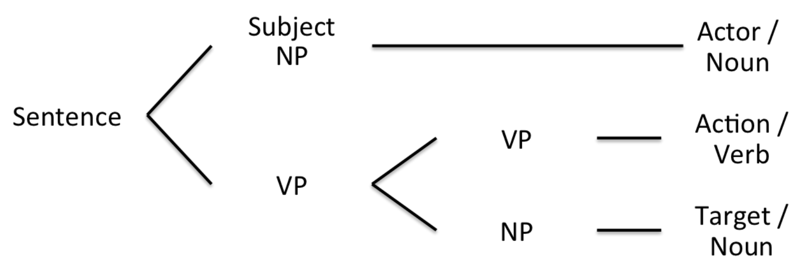


Figure 4.1: Sentence Parsing Structure

The natural language module was based on a simple part of speech tagging and named entity recognition process that focused primarily on the scene and stage directions within the play. It takes a command, such as:

GRAVEDIGGER1: (Pointing down into the grave)

and translates it into

actor=GRAVEDIGGER1 (current speaker)
action=POINT
target=GRAVE

This information was translated directly into a BML command for GRAVEDIGGER1, such as:

<gesture lexeme="POINT" target="GRAVE" />

This parsing process for a sentence can be seen in Algorithm 2, which uses the below nouns and verbs for this particular play-script:

Example Nouns: GraveDigger1, GraveDigger2, Hamlet, Horatio, Steps, Grave, Audience, Center Stage, Stage Left
Example Verbs: Move to, Follow, Look at, Pick up, Put down, Speak, Point to

Algorithm 1 Pseudo-Code for Natural Language Parse Line Algorithm

```
function parseLine(thisline)
    if isCharacterLine(thisline) then
        curCharacter = thisline
    else if isSpeechLine(thisline) then
        say(curCharacter, thisline)
    else
        mvmtLines = thisline.split(punctuation)
        for sentence in mvmtLines do
            for word in tokens do
                if isCharacter(word) then
                    saved[index] = word
                else if isPawn(word) then
                    saved[index] = word
                else if knownActionWord(word) then
                    saved[index] = word.translated
                else
                    wordLookup = wordNetLookup(word)
                    if wordLookup != null then
                        saved[index] = wordLookup.synonyms
                    else
                        do nothing
                    end if
                end if
            end for
            parseSentence(saved)
        end for
    end if
end function
```

Algorithm 2 Pseudo-Code for Natural Language Parse Sentence Algorithm

```
function parseSentence(sentence)
    find first noun in saved sentence
    find first verb in saved sentence
    if first noun position >first verb position then
        assume curCharacter is doing the acting
    else
        actor = first noun
        check for second noun or position
    end if
    translate stage direction position target as needed
    call verb-mapped function for the actor with the to what object or postn
end function
```

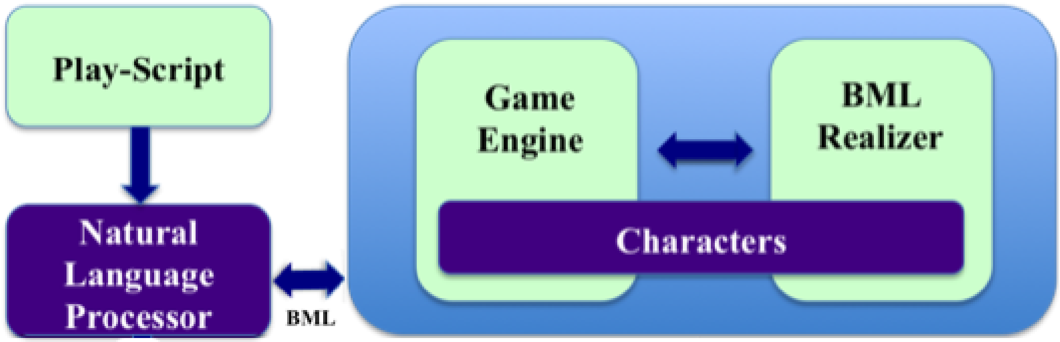


Figure 4.2: NLP Engine Architecture

The output of these translations were BML statements that were then passed to a BML Realizer, as can be seen in Figure 4.2. These techniques can also be applied more broadly since they only rely on the components that are inherent to play-scripts, movie scripts, and television scripts. The only scene-specific setups are ones based on identifying the characters and starting positions of key props within the scene–all of which are part of the manual setup of any scene for any play.

4.2 Rules

Next, we look to expand upon the natural language processing to incorporate rules to better our translation of motion from the play-script. We have pulled from many different areas to encompass the types of rules that are typically utilized when performing plays. We have categorized these rules into four basic areas:

1. Grouping Spatial Rules
2. Conversational Spatial Rules
3. Theatre Rules
4. General Rules

In the next few sections we discuss what is involved in each of these rule groups to provide a background for our work.

4.2.1 Grouping Spatial Rules

The grouping spatial rules refer to how the characters position themselves in groups. Jan describes six different forces that affect when/why a person may shift position when in a group of people; however, the main reason that could affect the positioning of characters in a play is that one person is too close to others to be comfortable [37]. Hall describes four different zones that personal space is divided into: intimate, personal, social, and public zones [34]. The actual distances involved in each zone differs for each culture and its interpretation may vary based on an individual's personality. If the speaker is outside a person's comfort area, the person will move toward the speaker. Similarly, if someone invades the personal space of a person, the person will move away [37]. Also, when there are several people in a conversation, they will tend to form a circular formation. This provides a sense of inclusion for all participants, and provides a better view of all members while conversing [42].

4.2.2 Conversational Spatial Rules

When conversing, people have certain tendencies with respect to where they stand / where they look. Research from psychology shows that people prefer to be across from one another than side-by-side in most situations, but there is importance to the surrounding area for determining the distance that is comfortable [85]. Also, friendship and attraction can affect the spatial distances between people by decreasing them, while negative attitudes may not have much affect on the spatial distances [88].

According to studies reviewed by Sundstrom, comfortable face-to-face distance for speaking while sitting is approximately five feet and comfortable face-to-face conversation standing is approximately three feet [88]. He also discusses the effects of spatial invasion for character behaviors and movements, and provides an overview of multiple research efforts looking at conversational space for both sitting and standing positions [88].

4.2.3 Theatre Rules

In the theatre, there are special rules and conventions when staging a play. Many of these guidelines revolve around engaging the audience and visibility onstage. Some of these special rules that actors apply within a theatre environment are also useful in virtual environments too. For instance, being downstage (near the audience) is a stronger position than being upstage and should be held by the most important characters in the scene. Actors should never turn their back to the audience when performing in a proscenium-style stage. [5]

Moving onstage can cause many issues including upstaging and covering. Both of these issues should be avoided, which in turn provides additional rules to characters on the stage. Upstaging is where one actor takes a position further upstage, or above a second actor, which causes the second actor to face upstage/away from the audience. Therefore this must be avoided to ensure actors do not present their backs to the audience, especially if both characters are just as important to the scene [55].

4.2.4 General Rules

The last group of rules encompasses all those things that we often think of as common sense. For instance, when we are walking we are usually looking at where we are headed. Similarly, when we pick up or point to an object, we tend to look at it; and when we are listening to someone, we look at the speaker. When someone points to something or something/someone moves, we are usually drawn towards looking at that person or object. If someone wants to pick up an object, they need to be close to it. Finally, characters should always perform natural movements and not have their gaze or orientation jump from one position to another.

4.2.5 Architecture

When we put all these rules together, we are able to formulate an intricate engine to control the movements of the characters to present a realistic interpretation of the play, similar to an actor. We built these rules on top of our existing natural language processor (NLP) engine, which utilizes a part of speech tagging and named entity recognition module to extract the high-level movements of the characters.

These NLP-extracted movements were fed into our rules engine (as seen in Figure 4.3) to adjust the motion based on these rules:

r_1: Characters should face the audience as much as possible, and avoid turning their back to the audience

r_2: Characters should face the person speaking

r_3: Characters with higher importance or larger roles should be placed slightly closer to the audience relative to lesser role characters

r_4: Characters should try to stay closer to center line as much as possible to improve visibility for the maximum portion of the audience

r_5: Characters should avoid unnatural movements by adhering to basic frame coherence rules, such as not having their gaze or orientation jump from left to right immediately

r_6: Characters should maintain appropriate personal space based on inter-character relationships within the play

r_7: Characters should be next to an item they wish to pick up

As the natural language processor identifies the action that needs to be performed, it sends it into our rules engine as an (actor, action, target) command. From there, our rules engine applies these seven rules to the action, translating it to one or more BML commands that are sent to the BML Realizer and Game Engine. A high-level overview of the process flow can be seen in Figure 4.4.

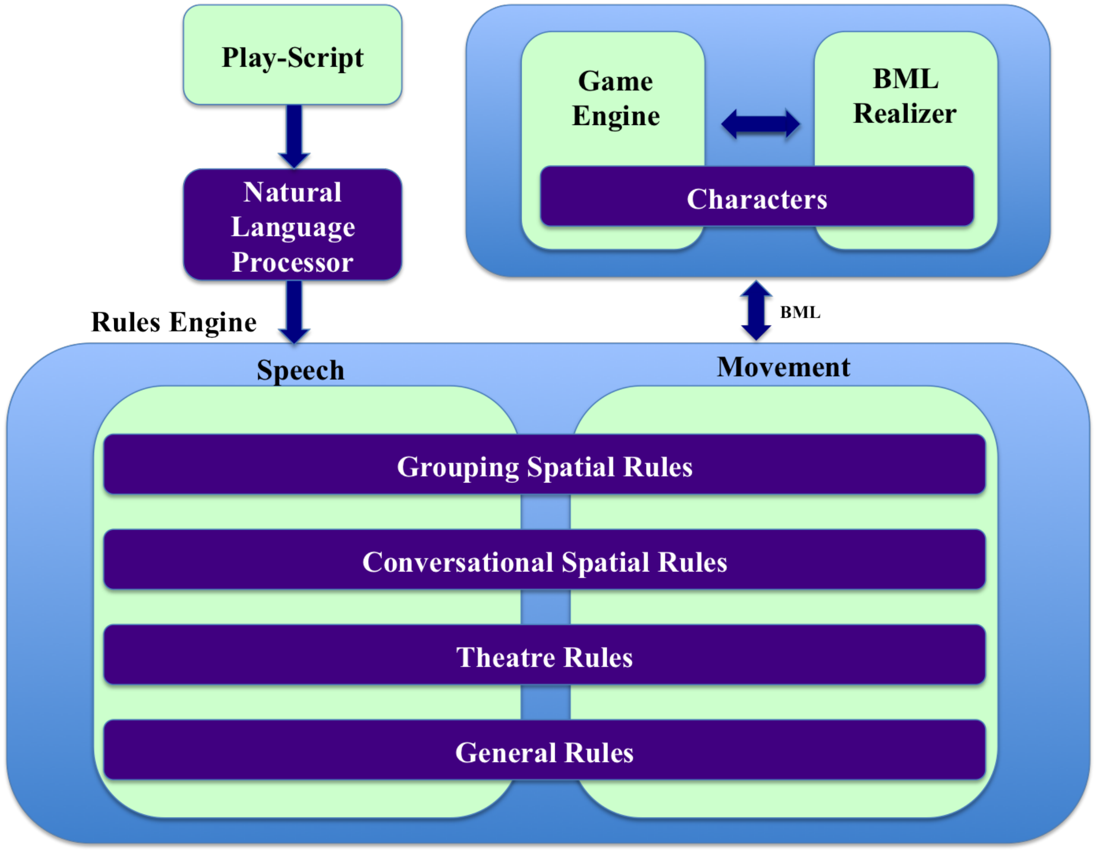


Figure 4.3: Rules Engine Architecture

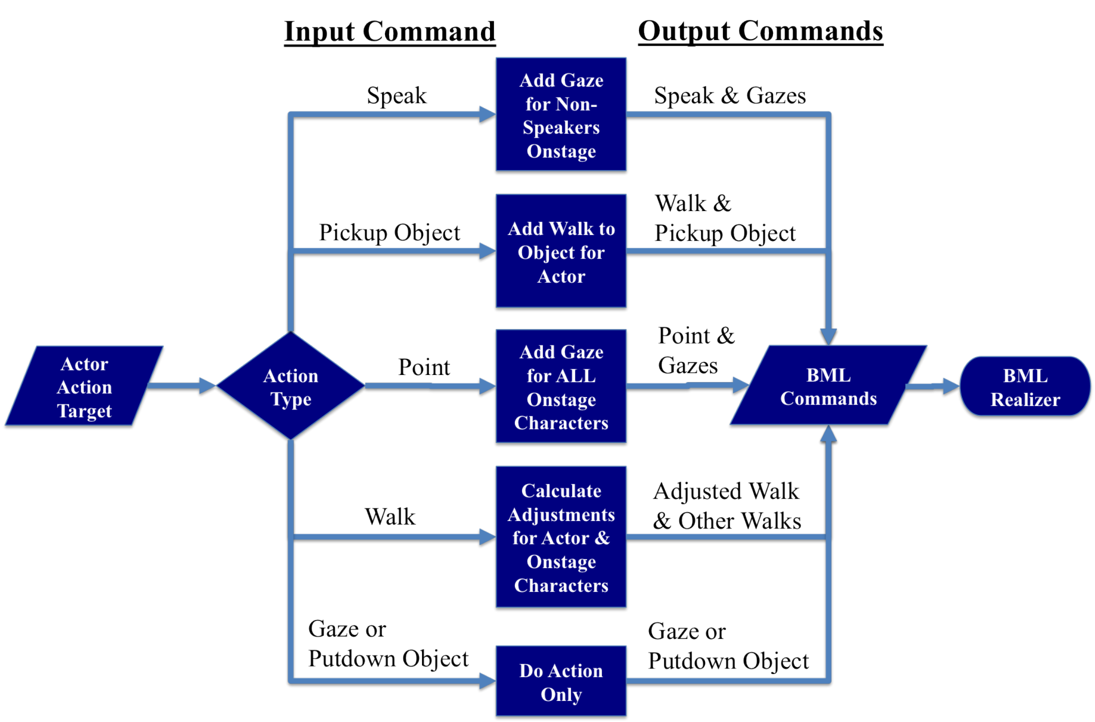


Figure 4.4: Logic in the Rules Engine

For speech commands, the rules engine adds additional commands for each onstage character to look at the speaker. This angle is adjusted based on the current position of the characters to ensure no one is looking more towards backstage than the audience. The speaker's gaze is also adjusted to look at the last speaker, assuming that character is still onstage.

With walk or locomotion commands, the rules engine takes into consideration the position of all the characters onstage to determine the best destination with respect to the requested target. Each character's overall importance to the scene was prioritized such that every character's importance relevant to every other character was clear, such as below:

$$Hamlet > Gravedigger1 > Gravedigger2 > Horatio$$

As can be seen above, Hamlet was the most important character in the scene, followed by Gravedigger1. This prioritization was used to determine who should be closer to the audience at any point of time. If the action's actor defined by the natural language processor (actor character) had a higher priority than one or more characters onstage, then the lower priority character(s) were moved to adjust for the relocation of the actor character, ensuring the distance to the audience was shorter for the higher priority character(s).

Also, when characters were directed to approach another character, the target locations were adjusted to accommodate any grouping or conversational space. If they were approaching a single character, they were directed to stop at approximately three feet from the other character. If they were approaching two or more characters, they were instructed to maintain an arc-like configuration facing the audience and maintain three feet from the closest character.

These character spacing adjustments were performed only once per annotation that incurred a walk command. This prevented characters from constantly adjusting and creating unnatural movements onstage, as well as aligned the timings of the movements with the intended actions within the play.

When a command is sent for a character to pickup an object, the rules engine will check to see where the character is on stage with respect to the target object. If they are not near the object, they will walk to the object before trying to pick it up. If this movement conflicts with any of the aforementioned stage locations based on character importance, the other character(s) will receive a walk command to move them to an appropriate location.

Finally, as a character pointed to a target, the characters that are onstage are directed to look at what the character is pointing to. With gazing and releasing objects, the BML Realizer handled ensuring appropriate frame coherence for the characters and did not require any additional logic before performing the action(s). Therefore, these commands were submitted directly to the BML Realizer and Game Engine for controlling the characters.

4.3 Force-Directed Graphs

To build on this work, we want to introduce a better positioning component for the characters that will work with the natural language processing and rules engine for the AI-controlled characters, as well as any adjustments required due to a human-controlled character being on-stage. Our assumption is that a human character that will not always follow the play-script perfectly, or with the same patterns as the other characters, and may move at incorrect, additional, or fewer times than they should. There is a need to be able to adjust for those scenarios, which we have done by adding a new component that receives any AI-controlled character re-positioning, as well as human-initiated repositioning, and adjust all the characters on-stage appropriately (Figure 4.5). This will help us to accommodate the unpredictable actions of the human on-stage with respect to the overall production of the play.

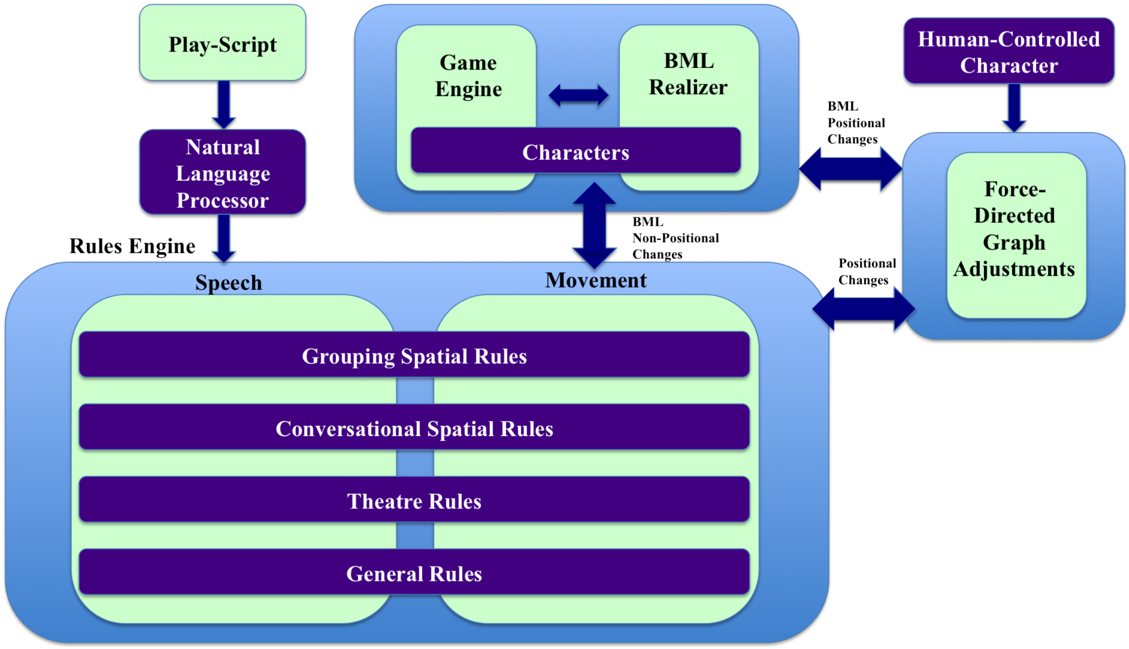


Figure 4.5: Rules Engine Architecture with Force-Directed Graph Adjustments for Human-Controlled Character(s)

The force-directed graph algorithms can provide a method for making minor adjustments to positions based on these unplanned movements by the human actors. After reviewing all of the different approaches to spatially displaying different types of graphs, we based our character positioning adjustments on the algorithm by Fruchterman and Reingold [29]. In doing this, we wanted to incorporate different properties of some of the other algorithms to center on appropriate theatre configurations. Therefore, we outlined some of the key requirements and approaches needed to support the blocking of characters on-stage, which are outlined below.

4.3.1 Graph Features

In this section, we will review the different features of graphs that are important to support their use in positioning characters on-stage. These include: even vertex distribution, small number of vertices, crossings-free drawings, fixed vertices, oscillation-free arrangements, strength of relationships over time, centering and encircling groups, and varying attractive and repellent forces. Each has its own impact to our solution.

4.3.1.1 Even Vertex Distribution

In theatre, it is important to maintain a sense of balance in the positioning of characters. We wanted the characters to be spaced relatively evenly on the stage within their targeted area. This is something that the Fruchterman and Reingold algorithm gives us for free, so no adjustments to their algorithms are needed to accomplish this aspect.

The distance between each character on-stage must be measured and compared. This should utilize the same relationships between every character to allow us to measure the true effect of the algorithm on symmetry. Some discrepancies are expected due to the differing relationships between the characters. These discrepancies will be accommodated during the comparison by ensuring they follow similar ratios and sequences as the character relationships. Measurements to the audience and props/targets should be measured also, but not be included in the calculations for symmetry.

4.3.1.2 Small Number of Vertices

To have a dozen or more characters on-stage at one time is not very common. Even if we incorporate additional vertices to represent key positions or objects on-stage, the audience, cameras/view angles, and so forth (as we will discuss in subsequent sections), we will find it difficult to end up with a large number of vertices (> 40). Again, this is helpful with our chosen algorithm since it is geared towards small graphs with less than 40 vertices.

Several different scenes will be reviewed to determine our typical number of vertices. We can introduce a scene with ten or more characters on-stage to determine the maximum typical vertices for a scene, as well as a scene with one or two characters on-stage. The review of these numbers can determine the order of magnitude of space

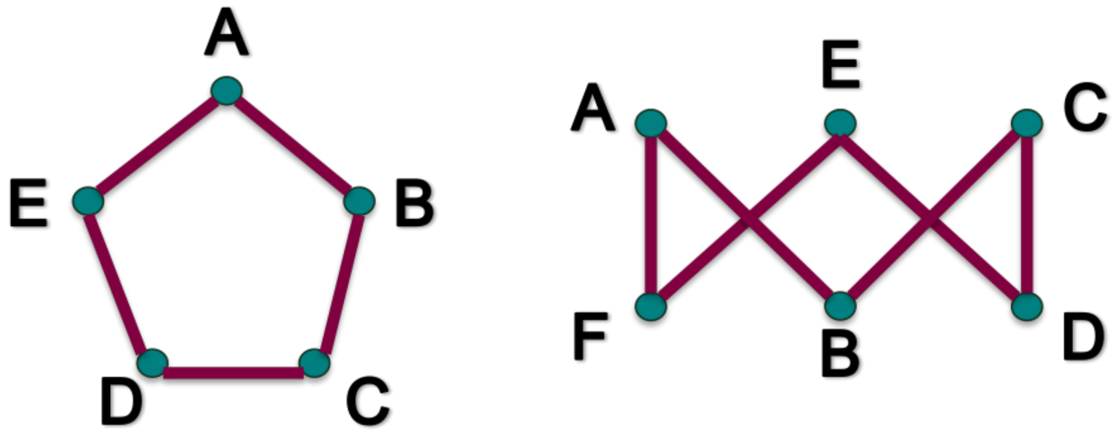


Figure 4.6: Left: A Crossings-Free Graph; Right: A Non-Crossings-Free Graph

and time complexity required for our algorithm, as well as to ensure we fall within the desired vertex targets of Fruchterman and Reingold's algorithm.

4.3.1.3 Crossings-Free Drawing

When aligning the nodes of the graph, we want to avoid characters being in front of each other or occluding each other's views. Tutte's algorithm [105] would be able to accommodate this if we can guarantee we have a three-connected graph with convex faces to start with. However, we may not be able to guarantee this with our graphs, so our approach may not always result in a crossings-free drawing. Figure 4.6 shows on the left, a crossings-free diagram, and on the right a non-crossings-free diagram.

To measure our occlusion results, each scenario will be tested to determine any intersections between the characters and the audience edges after our adjustment. The connectedness and convexity of the graph at each of these adjustment steps should be measured to determine whether we have met Tutte's prerequisites for accommodating this requirement of being crossings-free. Each line from the character to the audience that crosses another character's line to the audience should be discounted for within our calculations.

4.3.1.4 Fixed Vertices

We want to be able to show relationships with fixed points on stage to help ground the arrangement of characters based on the script's annotated destination for a character. To accomplish this, we need to introduce these fixed points as additional vertices in our graph, which will never change position (with only a few exceptions). These could be things like the location of a prop on the stage, where a chair is situated, or even where the cameras/view angles to be optimized for are within the audience. As a character is told in the play-script to move towards a particular object, it would be given an edge to connect it to the object.

How far a character is from its connected, fixed vertices needs to be measured to determine whether having these types of connections will help keep characters in-place.

4.3.1.5 Oscillation-Free Arrangements

We also need to ensure we would not run into a constant oscillation of positions for a single arrangement of characters, so the introduction of a cooling effect that slows down movements over multiple iterations, which Fruchterman and Reingold used is useful. We utilize an inverse linear function to decay the temperature to zero over several iterations for a single re-arrangement request.

Measurements of how far each character moves within each adjustment (per iteration) must be compared to determine how much, if any, oscillation occurs within the adjustment algorithm. Also, locations of characters from one adjustment to the next should be compared to ensure minimal oscillations between character movements occur. This is a main focus during tests that will trigger a re-adjustment where no character is actually moved.

4.3.1.6 Strength of Relationships over Time

To adjust for the connections with fixed vertices, we need to be able to decrease the strength of the attraction of those relationships over time. In addition, we want a relationship (two characters entering at the same time) to decay over time as they should move together initially, but may start to deviate the longer they are on-stage to a more neutral set of movements.

The effectiveness of changing the strength of the edge relationships over time can be analyzed through entire scenes. The change in each character's position when triggering the re-adjustment with no real character movement is measured to observe the effect of relationships.

4.3.1.7 Centering and Encircling Groups

We want to ensure the spacing between multiple characters presents a more uniform circle/semi-circle by introducing an extra "dummy" vertex that is always connected to every character on-stage. This should act as a pulling force to center characters around this point as much as possible.

Another key attribute of centering is to be able to establish character positions relative to the center of the stage for most instances. This helps to prevent visibility issues from the audience's perspective and centralizes the action on the stage.

Different numbers of characters will be experimented with to determine the resultant shape of the group. The more similar the curve produced by the character vertices are to a circle or semi-circle, the better we have done. The distances from a circle with the specified radius can be used to measure our accuracy.

4.3.1.8 Varying Attracting and Repellent Forces

We want different connections between vertices to use different types of forces. For instance, we want a connection to the audience to be weaker than a connection to another character on-stage. We also want the strength of the connection to the audience to vary based on the character's importance in the play or scene. Fruchterman and Reingold's algorithm bases the forces on the size of the drawing area, trying to maximize coverage. Since we do not want characters to be spread out on the stage, we will need to adjust the standard forces to trend towards grouping characters, but not overlapping them.

Playing with the different forces and their resultant measures for their effect on several different character configurations is key. Comparing the stronger attraction forces and weaker attraction forces can ensure they result in slightly different arrangements of the characters on-stage, with respect to the amount of attraction in place. The ideal attraction forces will be the ones that result in an average spacing of characters within three to five feet of each other.

4.3.2 Graph Structure

In composing the force-directed graphs, we should define how each aspect of the character positioning relates to the graph structure, which is shown in Figure 4.7. First, we have the characters themselves, which will be represented as a node within the graph. These will each have a position attribute that corresponds to their position on the stage. Next, we have the targets or marks on the stage that the characters are supposed to hit based on the play-script. These could be a particular object on the stage, a relative location to the audience or another character, etc. These targets are represented by a node in the graph, and also have a position attribute associated with them. Obviously, we will also have a node for the human-controlled character. This character/node will not be adjustable by the AI Director, but is key in guiding the positioning of the other characters onstage.

The other nodes in the graph are a little more complex in nature. The audience nodes are created for each character that is onstage. This node will maintain the same x-coordinate as its corresponding character, and will help to pull the character towards the front of the stage. There is also a node to represent the center of all

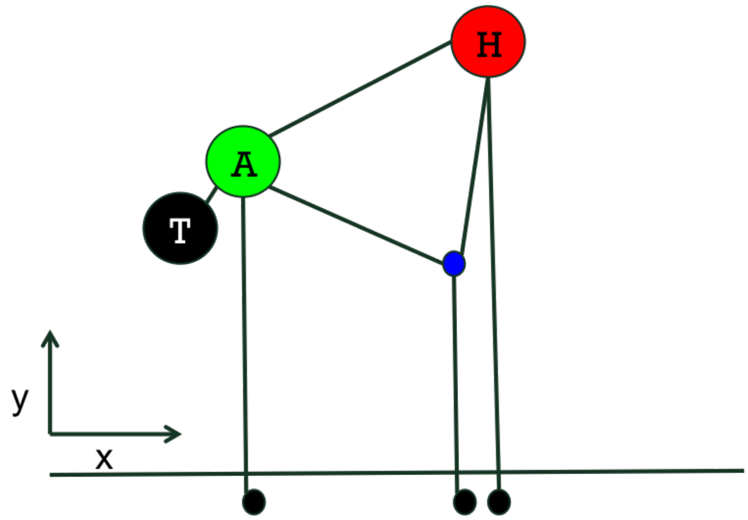


Figure 4.7: Force-Directed Graph Structure

characters onstage, residing in the center of all the characters. The center node will only be part of the graph if there are two or more characters onstage, and will assist with forming a semi-circular arrangement of the characters facing the audience (in conjunction with its own audience node).

Edges of the graph will connect all of these nodes in different ways, each with different attractive and repellent forces based on the relationship represented. First, the obvious, is the character-to-character edge. This edge will represent an attractive and repellent force to help the characters maintain a reasonable conversational distance from each other. If two characters enter onstage at the same time, their attractive forces on this edge will be stronger to help enforce the characters' relationship spatially.

Edges from each of the characters to the human-controlled character will also be created to help pull the scripted virtual characters towards the human, thereby creating an inclusive arrangement for the human. Every character will have an edge to their personal audience node, which will pull them towards the front half of the stage. In addition, each character (including the human-controlled character) will have an edge connected to the center point. These edges will force characters into a semi-circle instead of a circle due to the additional edge for the center point to the center point's audience node.

Finally, each character will have an edge to their target or mark on the stage. This connection will help to ensure characters remain close to their intended/scripted position in order to maintain the integrity of the play-script. It will also lose attraction force strength over time, just as characters lose the need to remain on a specific mark over time.

4.3.3 Algorithms

To accomplish the positioning of characters using force-directed graphs, we introduced the following functions based primarily on Fruchterman and Reingold's algorithm from 1991, which calculates an equalization of forces within the graph, and introduces a time cooling to minimize oscillations of the layouts [29]. Adjustments were made to remove the feature that tries to maximize the real estate used for drawing the graph. Additional algorithms were defined to handle when characters are added to a scene, when a character moves to a new position, when the human-controlled character moves, and when a character leaves the scene [94]. The full list of defined or adjusted algorithms include:

- Add Character(s) (Algorithm 3)
- Character Move to Position (Algorithm 4)
- Human Moves (Algorithm 4)
- Character(s) Leave (Algorithm 5)
- Time Step (Algorithms 4 & 6)

When adding a character on the stage (as seen in Algorithm 3), if they are the only one on the stage, we will introduce vertices for their targeted position, the audience

Algorithm 3 Pseudo-code for Adding Characters Method

```
G ← (V, E);                       ▷ The graph contains all vertices and edges for onstage characters
audience.y ← position             ▷ Default y position for audience front row
function AddCharacters(charlist, targetlist)
    for char ϵ charlist do                        ▷ Repeat for each character being added
        V ← v_char ← (target.x, target.y)
        V ← v_char_aud ← (v_char.x, audience.y)
        V ← v_char_target                         ▷ only add when not already in G
        E ← e_char_aud                            ▷ strength based on character importance
        E ← e_char_target@timestamp               ▷ save temperature information for connection
        if OnStageCount = 1 then                  ▷ Adding the second character onstage
            V ← v_centerpt
        end if
        E ← e_char_centerpt
        for v in V_char do                        ▷ Only review the character vertices
            E ← e_char_v                          ▷ give stronger strength to edges if v ϵ charlist
        end for          ▷ also give stronger strength based on character relationship
        CHARACTERMOVE(all)                        ▷ update strength of degrading edges
    end for
end function
```

(with the same x coordinate as the character's target position and the default y coordinate for the audience), and a vertex for the character itself. The audience vertex will be semi-fixed, in that it will change x position only as the character's x position changes. This connection is intended to help alleviate occlusions of characters on the stage. Initially, the strength of the connection between the character and its targeted position will be strong to ensure they end up in their targeted location as accurately as possible. The strength of the connection to the audience will remain constant, but will be based on the importance of the character within the scene—the more important the character is, the greater the attraction force will be.

If we are adding more than one character on the stage, there will also be a single "extra" vertex for the characters' center (regardless of how many characters are on-stage). This vertex will be connected with equal strength to all characters that are on-stage. It will serve as a gravitational central point, causing a circular effect for multiple characters, just like the typical conversational positioning we see for groups

Algorithm 4 Pseudo-code for Moving Character Method

function CHARACTERMOVE(char)
  $v_{char\_aud} \leftarrow (char.x, char.y)$
  for $e_{char\_char} \epsilon E$ **do** ▷ Only review edges between two chars
    if $e_{char\_char}.strength > charRelationship$ **then** ▷ Cool if entered together
      $e_{char\_char}.strength \leftarrow cool();$
    end if
  end for
  for $e_{char\_target} \epsilon E$ **do** ▷ Only review edges between chars and targets
    if $e_{char\_target}.char = char$ **then**
      remove e_{char_target} ▷ remove edge if char has new target
    else
      $e_{char\_target}.strength \leftarrow cool();$
      if $e_{char\_target}.strength = 0$ **then**
        remove e_{char_target} ▷ remove edge if has been there too long
      end if
    end if
  end for
  $ADJUSTALL()$
end function

of people. It is only removed if either only one character or no characters remain on-stage.

Also, if multiple characters are entering the stage at the same time, their connection will be strong to encourage a synchronization of movements for those two or more characters. Over time, this connection's strength will degrade to the default attraction, losing some of the synchronization of movement. This should mimic the typical importance to characters entering the stage concurrently.

Each time a character moves (human-controlled or otherwise), we will degrade the strength of the variable attraction forces for all characters on-stage (as seen in Algorithm 4). This ensures that the importance of a character's connection to a particular location (whether a co-entering character or target position) will fade as time passes. If a character moves to a new location, but still has a connection with a previous target that has not fully degraded in strength to zero, we will remove that connection to avoid conflicts. Any character that moves offstage will lose all

Algorithm 5 Pseudo-code for Remove Character Method

function RemoveCharacter(char)
  for $e_{char\_*} \epsilon E$ **do** ▷ Find edges tied to char being removed
    remove e_{char_*}
  end for
  remove v_{char_aud}
  remove v_{char}
  if $v_{char\_target}.edges$ is *empty* **then** ▷ If nothing else tied to this, remove it
    remove v_{char_target}
  end if
  if $OnStageCount = 1$ **then** ▷ If leaving only one char onstage
    remove $v_{centerpt}$
  end if
end function

its connections to anyone and anything still on-stage. Each movement will trigger a re-adjustment of the remaining characters ONLY once the new targeted location of the moving character has been determined. This will ensure we do not constantly adjust mid-step for character movements, causing too much attention and movement for the audience.

The only character that will be treated as a fixed point is the human-controlled character and whichever character is performing the movement (if not the human). We will encourage the AI-driven characters to follow the human-controlled character's initiative, whether it is correct or not. Therefore, adjustments made by the human will result in an adjustment of the other characters for each move, with some restraint so there is no constant movement. When a character moves offstage, we will remove all linkages to that character from the graph to reduce calculations of forces (as seen in Algorithm 5), and remove the center point if there is no one, or only one person, left onstage.

Finally, we will utilize multiple iterations of the forces calculations with a temperature control to prevent oscillating within each re-arrangement trigger. The temperature will degrade over each iteration of the algorithm's loop, but be reset for each re-arrangement request (as seen in Algorithms 6 and 4). The strength of the con-

Algorithm 6 Pseudo-code for Force-Directed Graph Adjustments within Virtual Stage Environments

```
function AdjustAll
    G ← (V, E);   ▷ the vertices are assigned initial positions based on annotations
    function f_a(x) ← return AttractiveForce
    function f_r(x) ← return RepellentForce
    for i = 1 → iterations do                              ▷ calculate repulsive forces
        for v in V do                  ▷ each vertex has two vectors: .pos and .disp
            v.disp ← (0, 0);
            for u in V do
                if (u ≠ v) then                  ▷ δ is the difference vector between
                    δ ← v.pos − u.pos;             ▷ the positions of the two vertices
                    v.disp ← v.disp + (δ/|δ|) ∗ f_r(|δ|);
                end if
            end for
        end for                                           ▷ calculate attractive forces
        for e in E do            ▷ each edge is an ordered pair of vertices .v and .u
            δ ← e.v.pos − e.u.pos;
            e.v.disp ← e.v.disp − (δ/|δ|) ∗ f_a(|δ|);
            e.u.disp ← e.u.disp + (δ/|δ|) ∗ f_a(|δ|);
        end for                             ▷ limit max displacement to temperature t
        for v in V do                           ▷ Only update non-human characters
            v.pos ← v.pos + (v.disp/|v.disp|)∗min(v.disp, t);
            v.pos.x ←min(W, max(0, v.pos.x));
            v.pos.y ←min(L, max(0, v.pos.y));  ▷ Prevent displacement off the stage
            v_aud.pos.x ← v.pos.x
        end for                          ▷ reduce the temperature over iterations as
        t ← cool(t);                          ▷ layout approaches a better configuration
    end for
end function
```

nections will remain constant through the multiple iterations of the algorithm when finding the local minima and positioning. This strength will move characters within three to five feet of each other to mimic typical conversational spacing of characters.

During any arrangement adjustment, only non-human characters are moved on the stage, to avoid any perception issues with the human controlling a character on-stage. This also means that the arrangements of the characters may not align perfectly with a real production of the play, but the goal should be to align based on the human-controlled character's position and movements to maintain the integrity of the script,

Table 4.1: Attractive and Repellent Forces
$|\delta|$ = Separation Distance, α = Desired Separation Distance

	Force Type	**AI Char**	**Center Pt**
AI Character	Attract	$\|\delta\|^2 - \alpha^2$	
AI Character	Repel	$-\|\delta\|^2 + \alpha^2$	
Human Character	Attract	$\|\delta\|^2/2 - \alpha^2$	
Human Character	Repel	0	
Audience	Attract	$\|\delta\|^2 - L^2/4$	$\|\delta\|^2 - L^2/16$
Audience	Repel	$-\|\delta\|^2 + L^2/4$	0
Center Point	Attract	$\|\delta\|^2 - \alpha^2$	
Center Point	Repel	$-\|\delta\|^2 + \alpha^2$	
Target/Pawn	Attract	$\beta\|\delta\|^2 - \alpha^2$	
Target/Pawn	Repel	0	

but include the human-controlled character.

4.3.4 Forces

The main algorithm utilizes the forces and the graph for repositioning characters on-stage whenever the human-controlled character moved. Each character relationship (edge) has its own unique forces that push or pull the virtual characters (vertices) around the stage. Some vertices are setup to be unmovable, such as the human-controlled character and the targets/pawns. For instance, the relationship between the virtual character and its mark/target would pull the virtual character closer to the mark, depending on how long it had been since the character moved to that location. The targets are identified by the play-script, with the assumption that all characters (including the human-controlled one) hit their marks correctly and on-time.

The goal for the non-moveable vertices is to act as attractors, but not repellers for the moveable characters. This can be seen in the table of forces in Table 4.1. The attraction and repelling of the vertices is setup to be a quadratic function of the distance of the two vertices. This ensures a stronger pull or push between the vertices as they get further or closer together, respectively. The special vertices of the audience to the character helps to attract the characters to the front of the stage as much as possible, while the center point is intended to act as a barycenter (or mass center point) for the characters onstage. By providing the center point a stronger attraction to the audience, it forces the group of characters to form a semi-circle facing the audience. Each of these parameters can be adjusted to fit the specific scene being performed, but has been set for this work to apply typical conversational space of approximately three feet.

4.3.5 Architecture

To incorporate the force-directed graphs into our current architecture, we allow our natural language processing module and rules engine module to determine an initial target for a character's position onstage. We then feed this information, along with all other onstage character positions, targets, and relationships into a force-directed graph. Each character is provided a link to their intended target (the position provided by the natural language processing and rules engine), a link to all other characters onstage, a link to the audience, and a link to a central point for the onstage characters. Each of these linkages have different strengths of attraction and repellant forces, dependent upon the type of relationship between the entities.

As any character moves (including the human-controlled character), each of the forces are re-evaluated to determine the need to adjust a character's position, as shown in the architecture in Figure 4.8. The rules around facing direction are re-applied once the movements are completed since the force-directed graph approach does not handle facing directions.

With all evaluations covered in the rest of this work, it needs to be remembered that these techniques build upon each other and are cumulative in nature. Each component relies on the prior component for its input. Therefore, the Rules engine

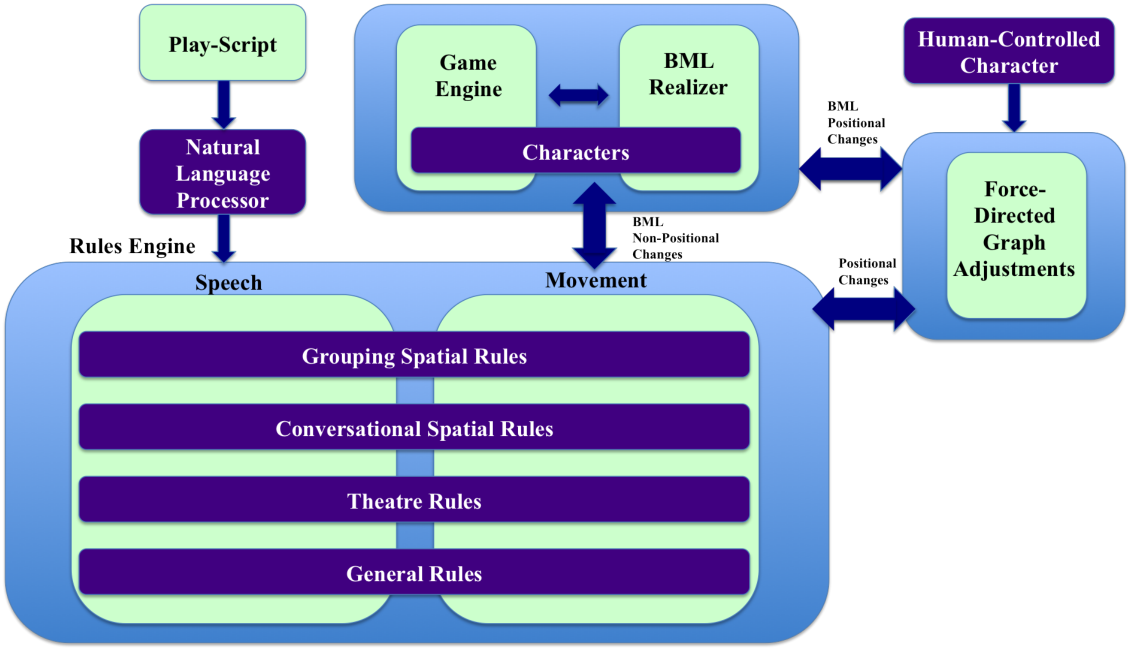


Figure 4.8: Rules Engine Architecture with Force-Directed Graph Adjustments for Human-Controlled Character(s)

output inherently includes the NLP output since it is required in order to apply any rules. Similarly, the FDG engine output inherently includes both the Rules engine and NLP outputs. All future mentions of FDG will imply the use of NLP plus Rules plus FDG in its evaluation.

CHAPTER 5: APPLICATION

To visualize our methods, and enable qualitative evaluations, we built several applications to simulate the performances. These applications applied the methods described in Chapter 4: METHODOLOGY. We started with a 2D visualization to assist with initial evaluations, then built 3D visualizations to support the user studies. The final application built was used to allow the user to interact with the performance, and added additional features.

5.1 2D Simulation

A simple jsGameSoup (https://github.com/chr15m/jsGameSoup) and NodeJS (https://nodejs.org/en/) application was built to visualize the results of the Behavior Markup Language (BML) and Functional Markup Language (FML) generated by the natural language processor (NLP). Each character in the play is represented by a circle. Their current gazing direction is indicated by the line inside the circle. When they point to an object or location, a line is drawn from the outer edge of the circle towards the object or location being pointed at. Objects are represented by smaller gray and black circles with letters inside them. For the *Hamlet* "Graveyard" scene, only a Lantern (L), Spade (S), and two Skulls (X) were required as props. When a character picks up an object, it will become black, with white lettering; upon placing the object back on the ground, it will become gray, with black lettering, again. This can be seen in Figure 5.1. Additional information on the source code for this simulation can be found in Appendix C: TOOLS.

This simulation was utilized for capturing and visualizing the quantitative analysis for the *Hamlet* scene prior to building a 3D simulation for qualitative analysis with

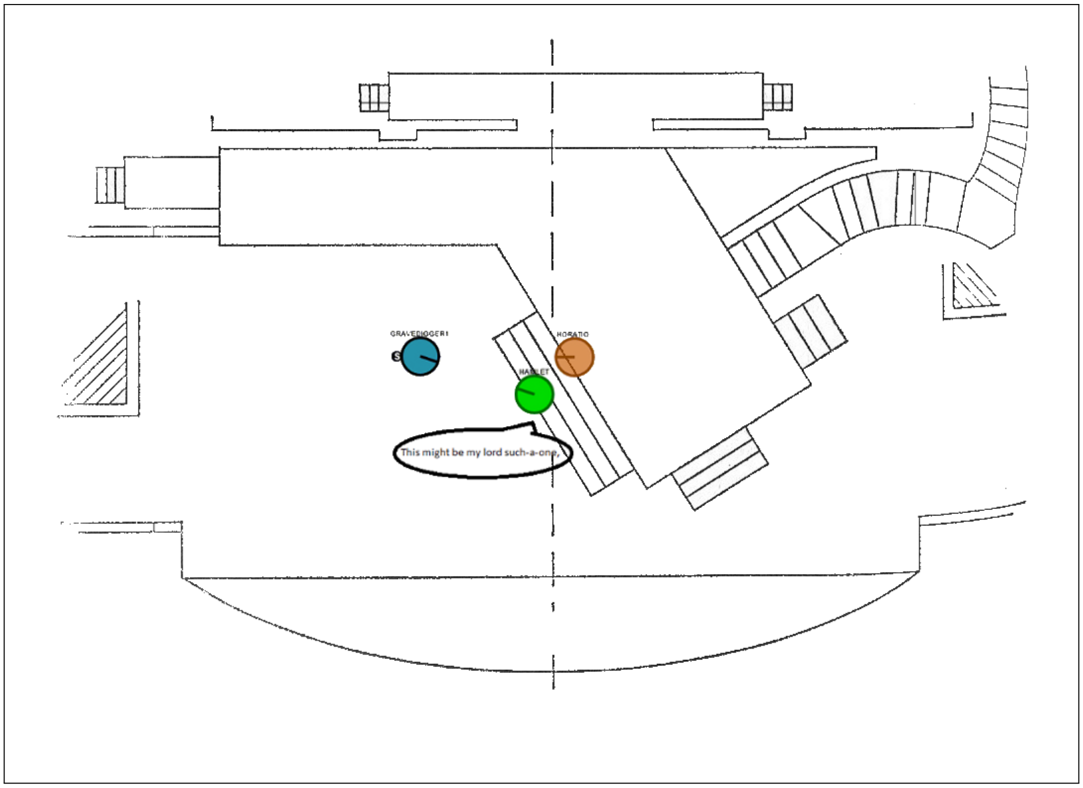


Figure 5.1: 2D simulation using hand-written BML commands

human participants.

5.2 3D Unity Simulation

The first Unity3D (https://unity3d.com/) simulation we built was intended to utilize the SmartBody BML Realizer and a robot-like character, as seen in Figure 5.2. Due to some work still being underway with SmartBody to fully realize their BML Realizer, we were unable to proceed with utilizing this solution. Therefore, we built our own pseudo-BML Realizer that did not do much path planning.

This custom-built 3D simulation leveraged a box character with a face panel that had sides to simplify our environment, as seen in Figure 5.3. This eliminated any bias regarding human versus virtual or block characters, as well as provided better visibility to where a character was facing due to the face sidepanels. It left only a small window when the character was looking away from the camera where the viewer had some ambiguity with which direction the character was really facing. This also helped

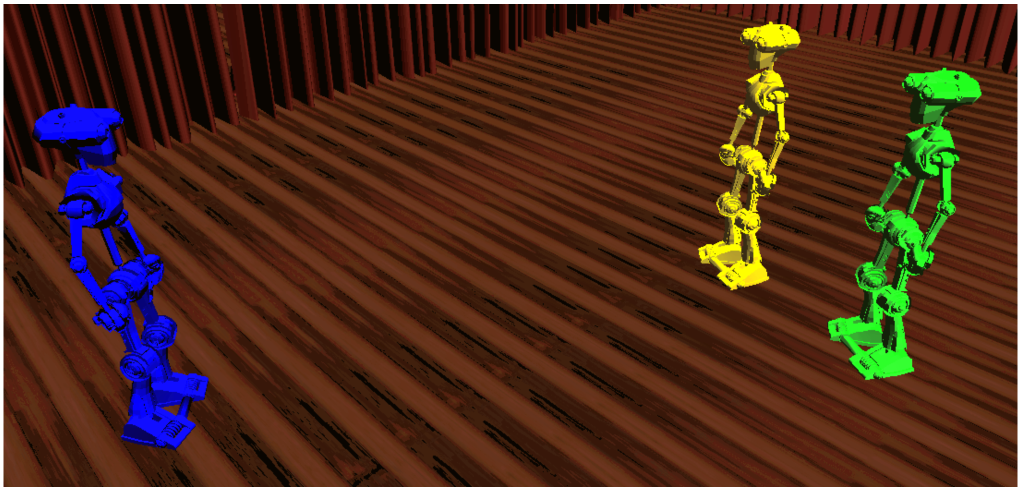


Figure 5.2: 3D Enactment of *Hamlet* in Unity Using the SmartBody BML Realizer

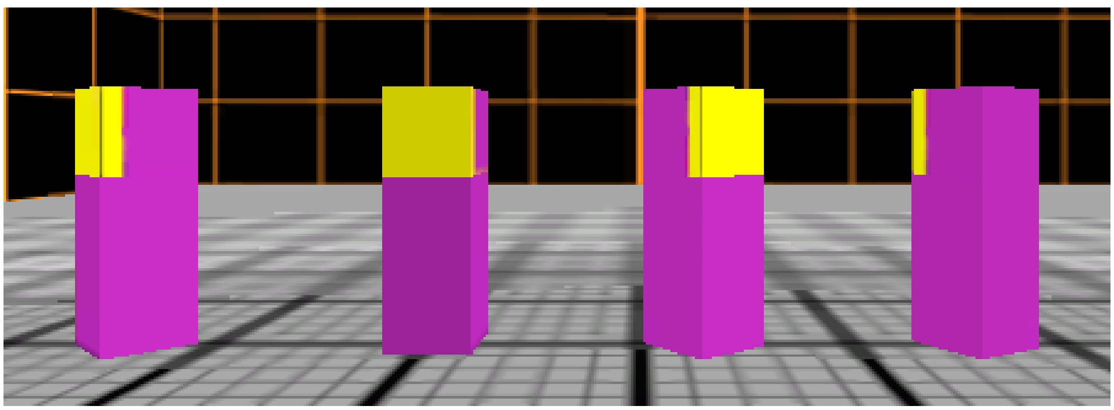


Figure 5.3: Block Character Representation

viewers to focus on the spatial aspects of the performance instead of any animations or character representations.

This environment represented pawns as cubes, and all marks (including physical pawns that were used more as a mark) were represented by an invisible sphere. A legend for these items was visible to the right of the stage window, which listed the colors for each character and each pawn the characters could interact with. At the bottom of the screen was a speech window where a mini image of the character speaking would appear next to the words that were being spoken. The system utilized the built-in "say" application within the Mac OSX. This layout can be seen in Figure 5.4.

The characters were able to point, move, gaze, pick up objects, put down objects,

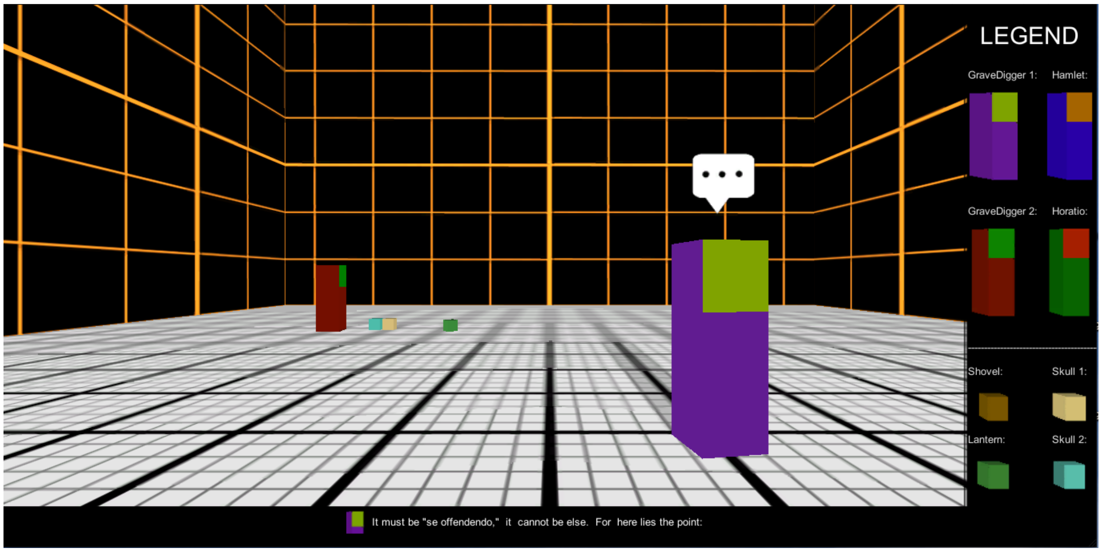


Figure 5.4: Block World Representation Utilized in 3D Videos

carry objects, and speak. When a character moved to a location, it performed no real path planning—it only moved in a direct route unless it ran into another character, which would cause it to shift around the character before moving on. Characters were able to speak while doing any other movement, but could not move until they had rotated enough so they could see the target they were headed. Otherwise, characters could turn and move at the same time as well. When a character spoke, a speech bubble would appear over their head as a visual indicator of who was speaking. To pick up or put down an object, the character would shrink to half its size, then grow back to normal size, and could only have one object held at a time.

The system accepted a formatted BML file from our NLP processor, along with an initialization file. The initialization file indicated all the characters (and their importance), the pawns, and the marks within the scene. Additional information on the source code for multiple versions of this simulation can be found in Appendix C: TOOLS.

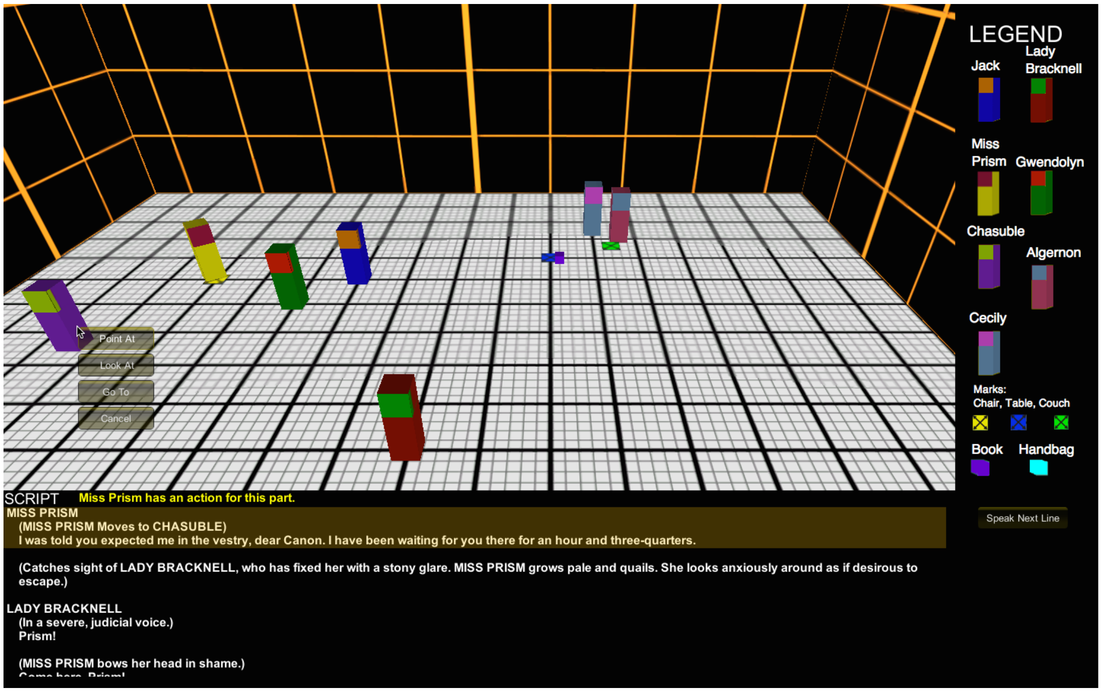


Figure 5.5: Interactive Block World Representation

5.3 3D Unity Interactive Simulation

The Interactive simulation had a similar setup as the Unity3D simulation—it had the same legend area, characters, and pawns, as seen in Figure 5.5. However, the simulation needed to allow the user to control one character within the scene, thereby affecting the positioning of the other characters on the stage. So additional features were added to better enable this interaction with the scene.

The marks that represent physical items, but cannot be picked up, were represented by colored squares with an "X" in the middle. The camera angle was adjusted so it was easier for the user to see the entire floor of the stage in order to move to specific locations easily, as well as see these colored X's. The bottom of the screen was enlarged to show the entire play-script, with a highlight box showing the block that was currently being performed. This play-script would scroll so that the highlighted area was always at the top of the script area. There was a "Next" button below the Legend, which would allow the user to progress through the script at their own pace.

The user could control one character within the scene. If that character had an action (to speak, to move, or to do both) for the highlighted block, there would be a yellow warning at the top of the script to remind the user. If the character was supposed to speak, the "Next" button would say "Speak Next Line." If the user wished to move the character, they only had to click their mouse on the screen. A menu would appear that gave the user their options with what they clicked on. If they chose a moveable pawn, then they could go to, look at, point to, or pick up. If they were holding that pawn, they could only put it down. If they chose another character, or anywhere else on the screen, they could go to, look at, or point to. If the spot was offstage, then they could only point to or look at that location.

The system guided the user through three scenes—the first as a practice, where no characters, except the human-controlled character, moved. The second was randomly chosen to be either the NLP+Rules version or the NLP+Rules+FDG version of the engine. Once completed, the user was then presented the other/unseen version of the engine.

The system accepted a formatted BML file from our NLP processor, along with an initialization file. The initialization file indicated all the characters (and their importance and if human-controlled), the pawns, and the marks within the scene. For additional simplicity in engineering, this system accepted the original play-script and a formatting file for presenting the play-script to the user in the UI. There were three sets of each of these files to support each of the three scenes this simulation would run through (practice, FDG, and rules). Additional information on the source code for this simulation can be found in Appendix C: TOOLS.

5.4 Generalization and Assumptions

Both of the 3D simulations described have been setup to be generalizable to any play. They do assume a single audience point and a proscenium-style stage. To utilize these simulations for other plays would only require inputting the NLP-generated

BML for the formatted play-script, and creating an initialization file for all the defined characters, pawns, marks, character priorities, legend image, voice names, and colors to use. Additional details on these files can be found in the Appendix C: TOOLS.

CHAPTER 6: EXPERIMENTATION AND DISCUSSION

In the last chapter, we discussed the algorithms and techniques that we developed, and would leverage in our evaluations. Here, we will review the experimentations performed to validate these approaches, including discussions on the overall authorial burden for a designer to position characters in a scene. We review both quantitative and qualitative analysis on each component of our work, showing that: our methods can match the positioning of characters from a real performance at 89%, forces provide less occlusion and better clustering of characters, and our techniques are indistinguishably "good" versus a real human performance from a viewer's perspective.

6.1 Authorial Burden

Shakespeare is still one of the top ten plays produced today. In fact, they are so popular that they are not included in the top ten play lists because at least five of them are always written by Shakespeare. Also, Shakespeare is free to use, and free of any copyrights. We were able to find a detailed annotation of *Hamlet* [86], which also happened to be the longest running production of *Hamlet* ever to play in New York, at 138 performances. Along with this detailed annotated script (which is unusual for Shakespeare plays), we found an Electrovision video [15] of the actual production on Broadway in 1964. These assets provided key inputs for quantitatively evaluating positioning characters in virtual environments.

We utilized the Electrovision video [15] and annotated play-script [86] to hand-map the movements and positions of the characters in the "Graveyard" scene on stage (*Hamlet* ACT V, SCENE 1). We used this mapping as a comparison to a basic natural language translation of the same annotated scene, and refer to it as our "baseline" going forward.

We manually mapped out about 14 minutes of Act V, Scene I from *Hamlet*, as produced by Sir John Gieguld in 1964 (Figure 6.1). This happens to be the graveyard scene where Hamlet reminisces about a skull that may have been Yorick, an old friend, and can be seen at https://www.youtube.com/watch?v=rFgd_4YrraU. The play consists of 280 lines and actions when mapped following the play-script standards for formatting, with the additional annotations provided by Sterne. The position of each of the characters were hand-mapped against the stage layout, utilizing the recording of the 1964 play as a guideline. Key aspects captured included walking, pointing, gazing/turning, and picking up/carrying objects. These movements were the focus of the spatial aspects of the play, which could be rendered in 2D, and were converted into Behavior Markup Language (BML) (and Functional Markup Language (FML) where appropriate for speech), as seen in Figure 6.2. Physical grid locations and marks were required to be created and manually mapped in both the initializations and within the BML itself to mimic the spatial dimensions that were manually mapped out.

For a 14 minute heavily annotated scene with less than 100 lines, it took over four hours to create the appropriate BML commands. This merely covered speech and spatial movements such as walking, pointing, picking up items, and looking at characters or items. It turned the script into about 400 speech and spatial commands in BML. This is a 142.86% increase in commands that were needed to be written to accommodate just the four spatial aspects of moving, pointing, gazing, and picking up objects. Even with this amount of time and effort, it probably only covered about 78% of the movement assumptions that an actor would utilize in performing the script, and no non-verbal behaviors. Imagine trying to accomplish this level of detail for something as dynamic as a game! Then add on the complexity of trying to change this or tweak this segment of action as the author or player edits the plot. Not to mention the issue of needing this specialized expertise in order to create a plot-line

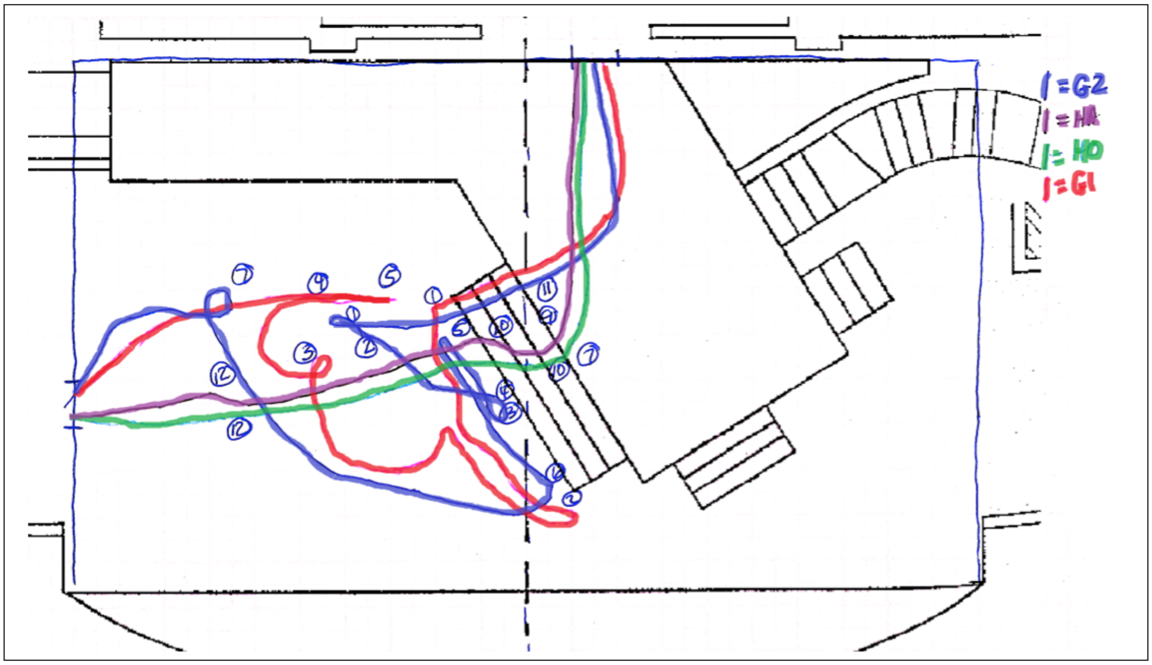
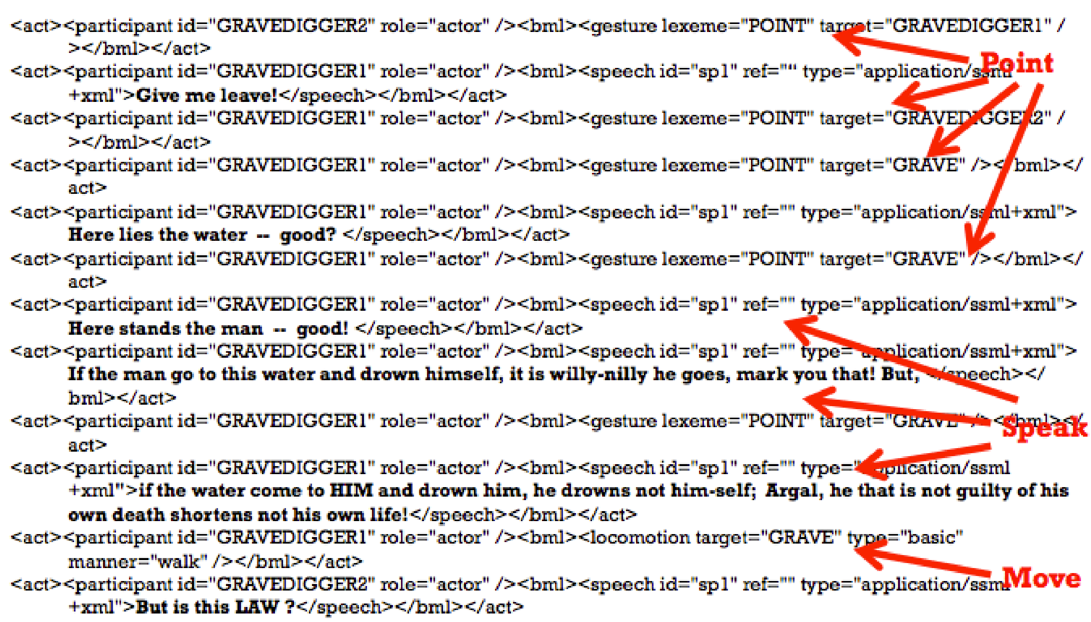


Figure 6.1: Hand-mapped Blocking of Richard Burton's *Hamlet* Play from 1964



```
<act><participant id="GRAVEDIGGER2" role="actor" /><bml><gesture lexeme="POINT" target="GRAVEDIGGER1" /
    ></bml></act>
<act><participant id="GRAVEDIGGER1" role="actor" /><bml><speech id="sp1" ref="" type="application/ssml
    +xml">Give me leave!</speech></bml></act>
<act><participant id="GRAVEDIGGER1" role="actor" /><bml><gesture lexeme="POINT" target="GRAVEDIGGER2" /
    ></bml></act>
<act><participant id="GRAVEDIGGER1" role="actor" /><bml><gesture lexeme="POINT" target="GRAVE" /></bml></
    act>
<act><participant id="GRAVEDIGGER1" role="actor" /><bml><speech id="sp1" ref="" type="application/ssml+xml">
    Here lies the water -- good? </speech></bml></act>
<act><participant id="GRAVEDIGGER1" role="actor" /><bml><gesture lexeme="POINT" target="GRAVE" /></bml></
    act>
<act><participant id="GRAVEDIGGER1" role="actor" /><bml><speech id="sp1" ref="" type="application/ssml+xml">
    Here stands the man -- good! </speech></bml></act>
<act><participant id="GRAVEDIGGER1" role="actor" /><bml><speech id="sp1" ref="" type="application/ssml+xml">
    If the man go to this water and drown himself, it is willy-nilly he goes, mark you that! But, </speech></
    bml></act>
<act><participant id="GRAVEDIGGER1" role="actor" /><bml><gesture lexeme="POINT" target="GRAVE" /></bml></
    act>
<act><participant id="GRAVEDIGGER1" role="actor" /><bml><speech id="sp1" ref="" type="application/ssml
    +xml">if the water come to HIM and drown him, he drowns not him-self; Argal, he that is not guilty of his
    own death shortens not his own life!</speech></bml></act>
<act><participant id="GRAVEDIGGER1" role="actor" /><bml><locomotion target="GRAVE" type="basic"
    manner="walk" /></bml></act>
<act><participant id="GRAVEDIGGER2" role="actor" /><bml><speech id="sp1" ref="" type="application/ssml
    +xml">But is this LAW ?</speech></bml></act>
```


Figure 6.2: Work Required to Write BML

with spatial implications.

If we look to generalize this reduction of authorial burden, we can see that one line of natural language in a play-script can turn into approximately four lines of BML. If each line takes the same amount of time to write (both BML and natural language), then we can clearly see that BML will take longer to write. Then, when you take into consideration some of the work that has been done with writing in a secondary language (which BML can be considered to be), it has been shown that writers working in secondary language have more errors and take longer to write similar text [44]. Therefore, we can see that writing the character blocking in natural language will be faster and more accurate than forcing an author to compose the blocking in BML.

These play-script annotations, along with our system, saved us over four hours of encoding explicit directions in BML code, and can now be reused with other scenes as well. This BML script became our standard for determining how well our method could provide similar spatial controls, while reducing the technical effort and time required to author the script (as seen with our previous example of creating BML).

6.2 Quantitative Evaluation

We compared our system's blocking for a scene to an actual performance of the same scene, leveraging the same play-script. To accomplish this, we took logs of each of the characters' positions and gaze directions throughout the scene. We mapped these exact positions to one of our ten grid locations: offstage, up right, up center, up left, stage right, center stage, stage left, down right, down center, and down left. We also mapped the exact gaze directions to one of our four directions: upstage, stageleft, downstage, and stageright. Next, we normalized the timestamps for these logged locations to ensure the duration of both scenes being compared were the same. With that normalization of time in place, we compared whether the grid locations and directions matched or not.

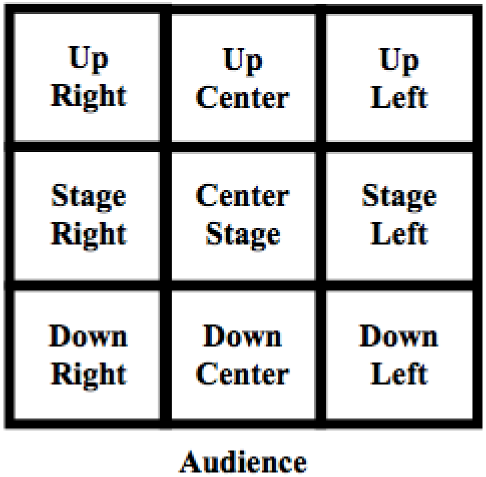
The play-script annotations, along with some basic natural language processing, provided the ability to position characters correctly approximately 78% of the time. The inclusion of the rules engine increased our performance for character positioning to 89% matching [95] with our existing hand-mapped baseline of *Hamlet* from Broadway. The main outlier from this work appeared to be due to one actor's interpretation, or whim, during the performance.

We also evaluated the incorporation of a human-controlled character within a scene to determine the amount of inclusion of the characters. We measured the arrangement of the characters when using the force-directed graphs to ensure proper arrangements, including conversational space, semi-circle arrangements, and even spacing. We also found an decreased amount of occlusion and stable clustering of characters when incorporating our force-directed graphs, when compared to our baseline. In addition, these numbers remained consistent regardless of the accuracy of the mock-human-controlled character, whereas our baseline showed increasing occlusion and less clustering as the accuracy decreased [94, 92, 96].

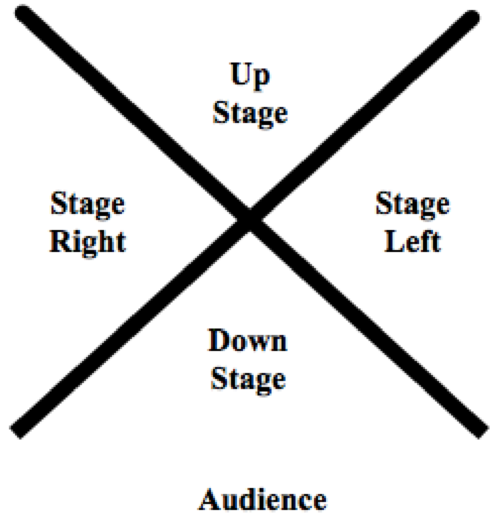
6.2.1 Natural Language Processing to BML

We took the character traces from both our hand-coded BML and our new method and compared them. We normalized the time durations for the two scenes to ensure a similar baseline for comparing positions. We then mapped the positions to the ten grid locations, and the gazes to the four grid areas, as seen in Figure 6.3, using the trace logs from both scenes. We wanted our new method to result in character positioning as close to our baseline as possible; however we did not want to penalize for being "close enough."

Doing this comparison, we see in Table 6.1 that we accomplished a relatively similar (78% matching) character position trace over time with this method of named entities and part of speech tagging. We only matched the gazes 34% of the time, partly due to the difficulty with capturing all the gazes when handmapping a scene.



(a) Stage Positions



(b) Stage Gaze Directions

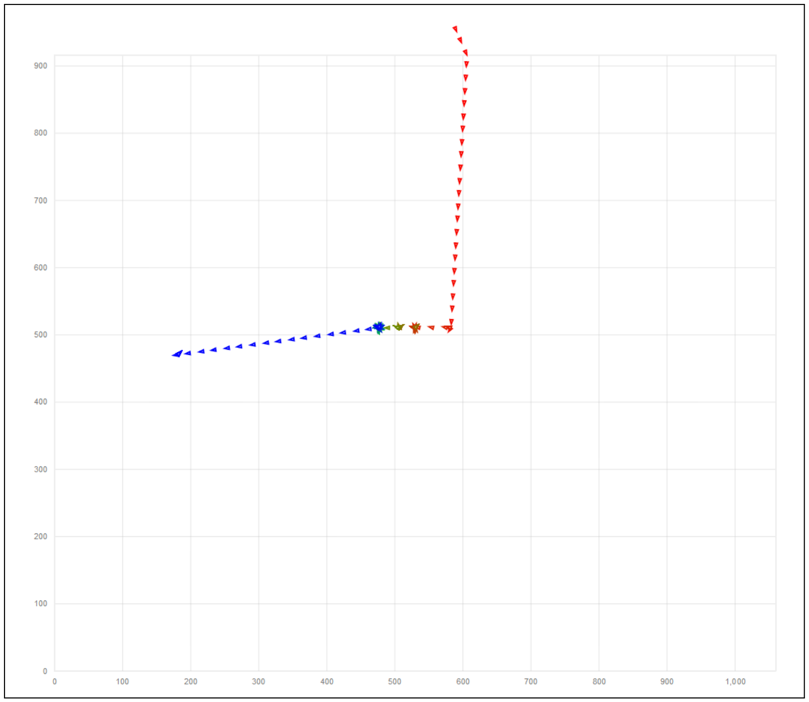
Figure 6.3: Stage Area Breakdown for Position and Gaze Comparisons

Table 6.1: *Hamlet* Character Traces Match for Baseline vs. NLP

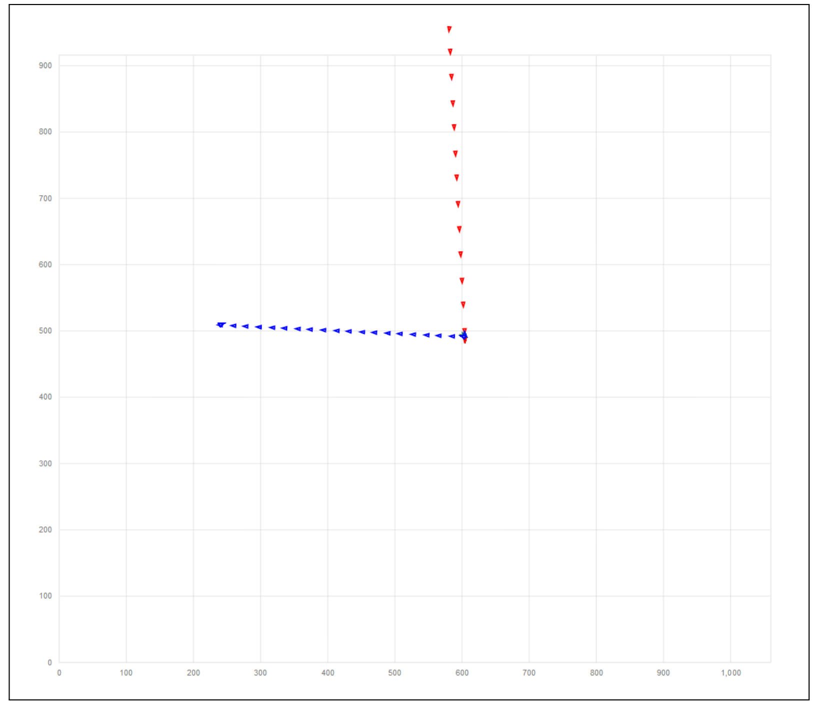
Character Name	Gaze Match	Position Match
GraveDigger1	57.95%	80.11%
GraveDigger2	19.43%	34.96%
Horatio	2.37%	98.23%
Hamlet	56.17%	99.18%
Overall	33.98%	78.12%

We mapped out each character's position over time visually in Figures 6.4, 6.5, 6.6, and 6.7. These figures show small arrows that represent where the character was facing at that moment, along with where they were on the stage. The colors indicate time, and go from red to blue as time progressed within the scene.

This representation was difficult to visually compare the two versions of the scene, and did not incorporate the "close enough" feature that the grids provided. Therefore, we re-mapped the same information into gridded line charts—one that represents the stage position with ten grid areas, and one that represents the gaze direction with four directions. These new traces can be seen in Figures 6.8, 6.9, 6.10, 6.11, 6.12, 6.13, 6.14, and 6.15. Even though these charts require more space, we decided to represent

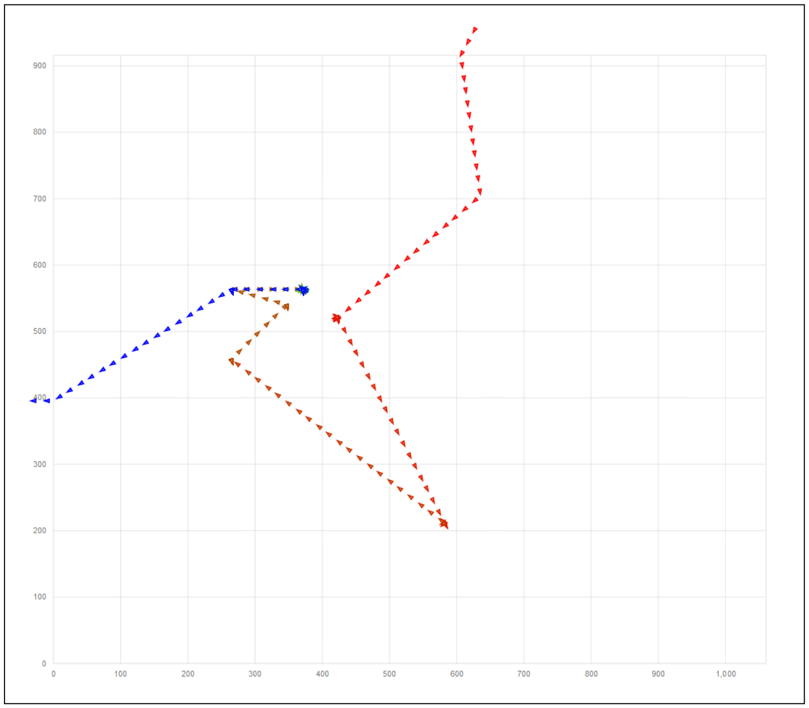


(a) Baseline Trace

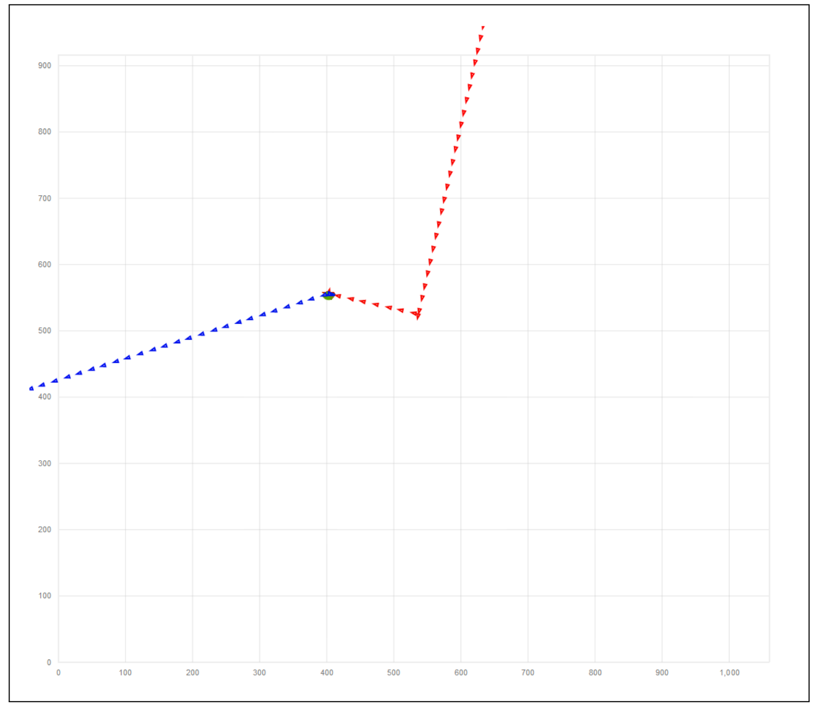


(b) NLP Trace

Figure 6.4: Baseline and NLP Character Traces for Hamlet

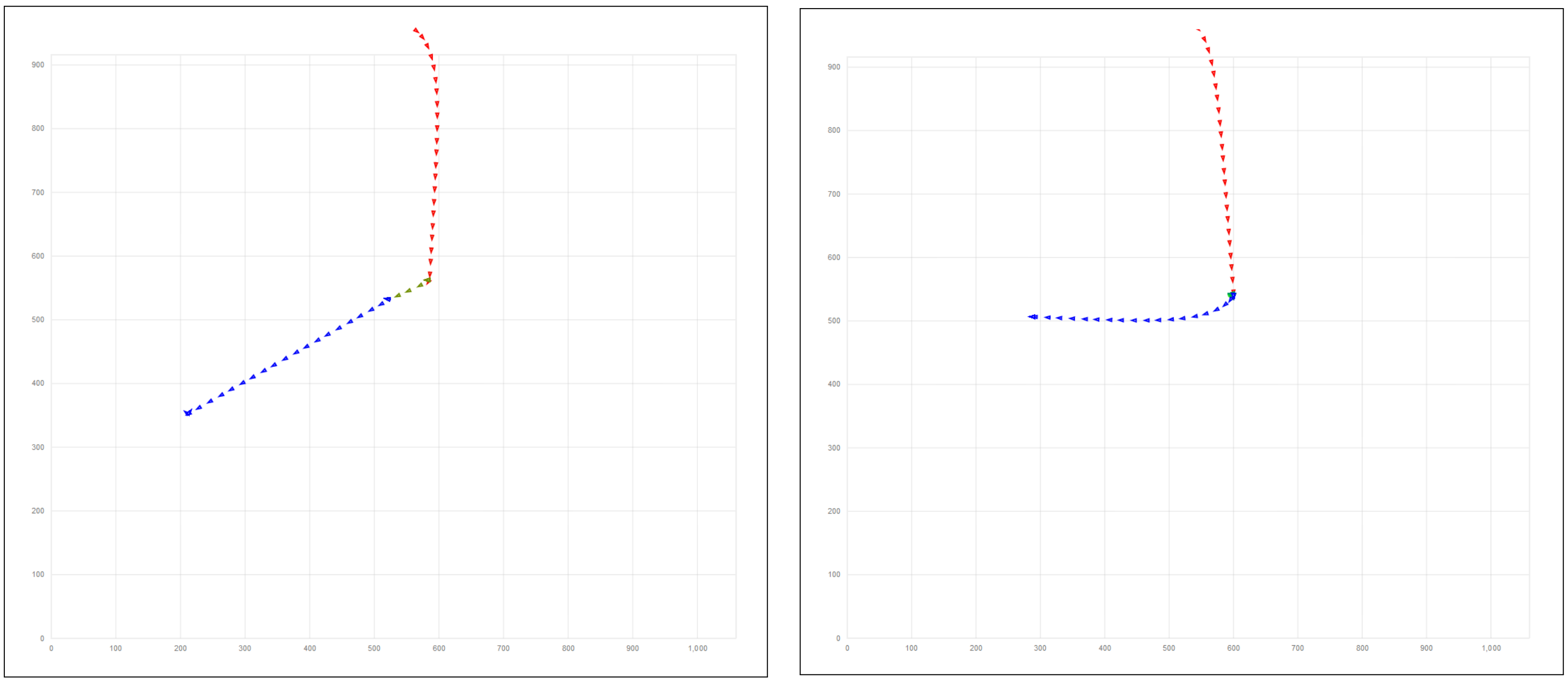


(a) Baseline Trace



(b) NLP Trace

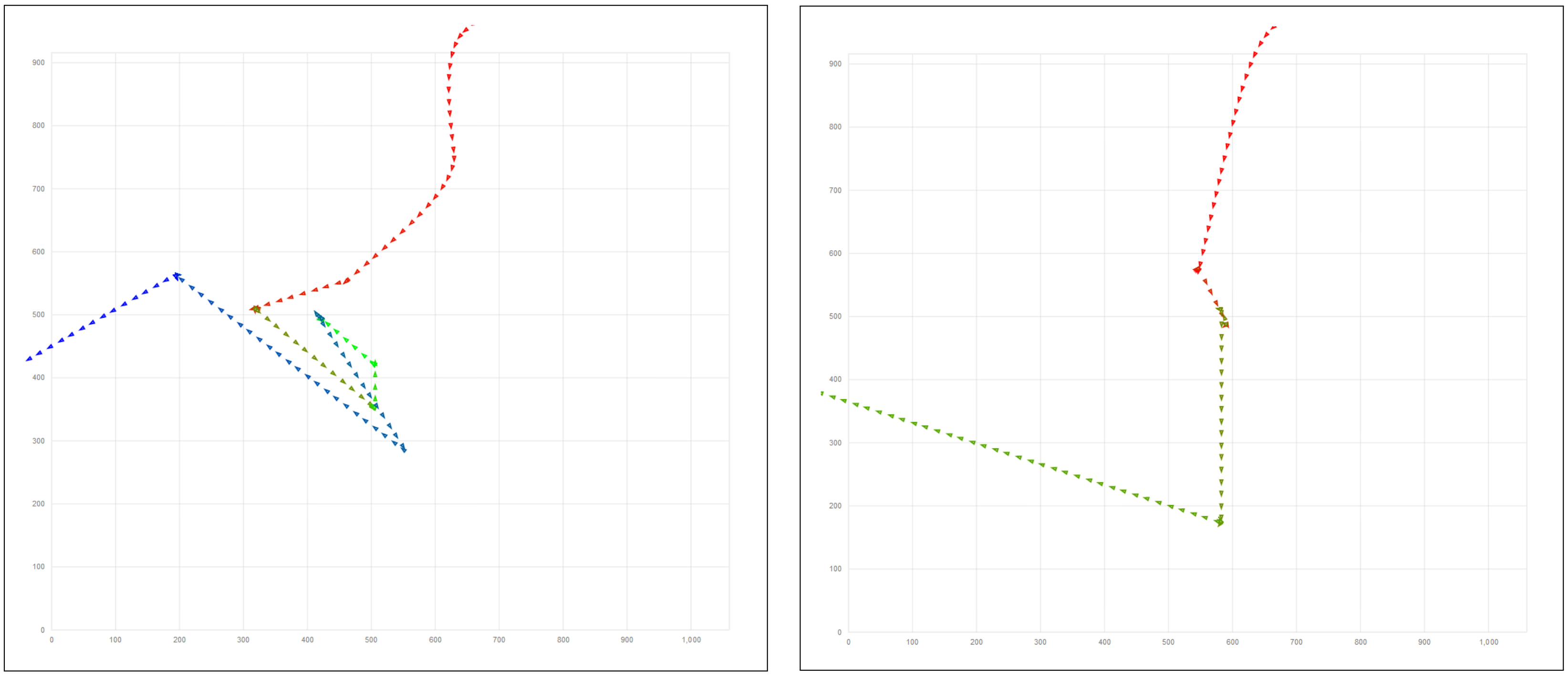
Figure 6.5: Baseline and NLP Character Traces for Gravedigger1



(a) Baseline Trace

(b) NLP Trace

Figure 6.6: Baseline and NLP Character Traces for Horatio



(a) Baseline Trace

(b) NLP Trace

Figure 6.7: Baseline and NLP Character Traces for Gravedigger2

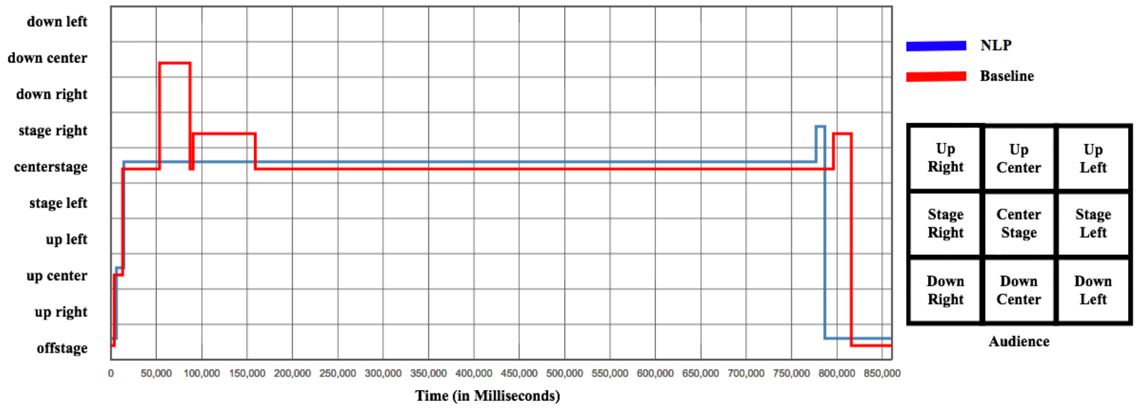


Figure 6.8: Character Position Traces for GraveDigger1 in *Hamlet*

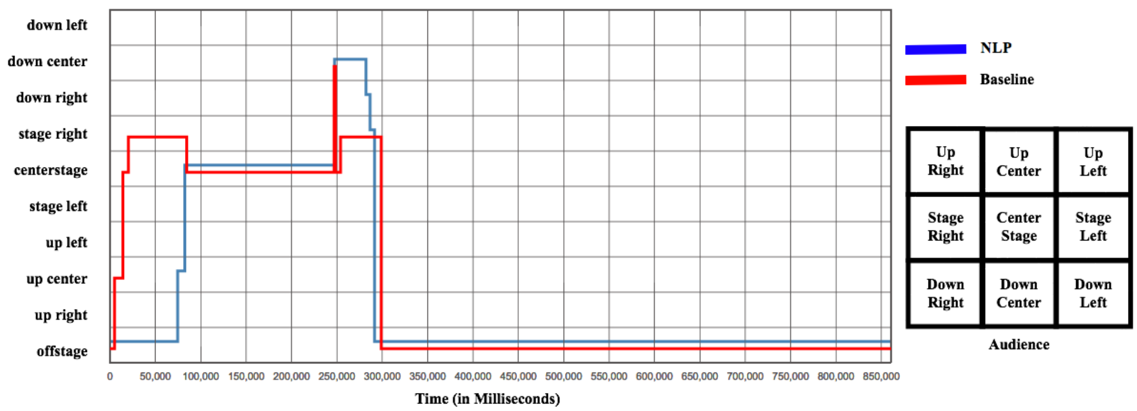


Figure 6.9: Character Position Traces for GraveDigger2 in *Hamlet*

these traces, going forward, in separate position and gaze charts because they are clearer to interpret. Reviewing these charts closely, and comparing the actual numbers behind these traces, we can see that we are typically very close to the baseline for our characters. This shows we were able to accomplish a reasonable blocking for this play, thereby saving us more than four hours of work and technical expertise for these 14 minutes of script.

Issues include the fact that sequence does not always indicate the actor versus the recipient, such as "He is followed by GRAVEDIGGER2," where GRAVEDIGGER2 is actually the actor. Also, prepositions are an issue as assumptions need to be made across movement statements and speech acts. Also, because we are working in 2D and the annotations for the play are 3D in nature, and include how to say/speak certain

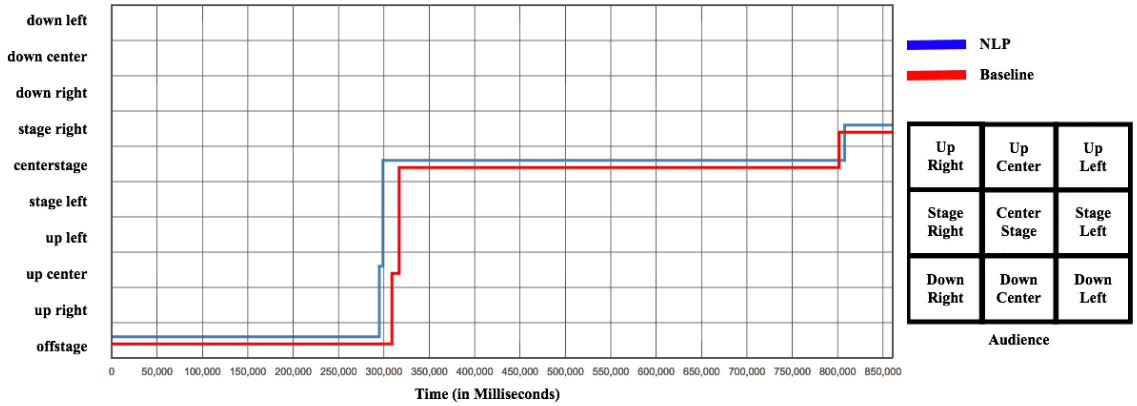


Figure 6.10: Character Position Traces for Hamlet in *Hamlet*

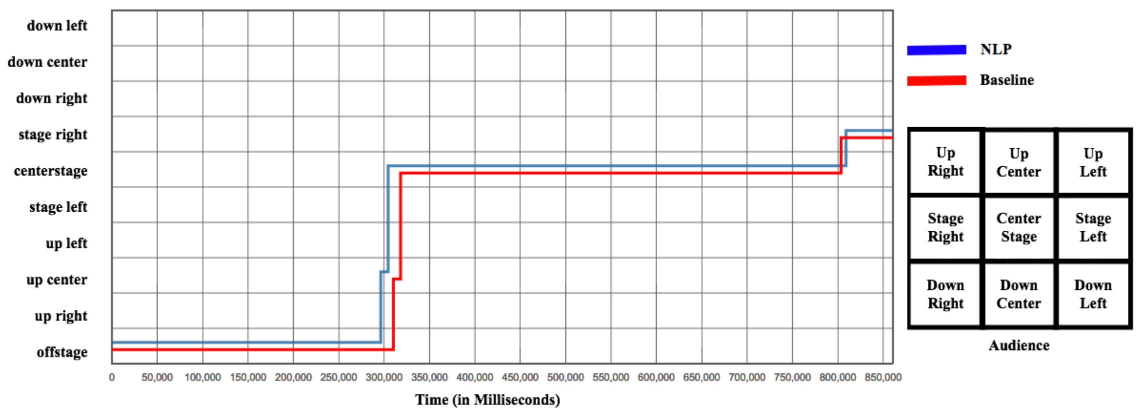


Figure 6.11: Character Position Traces for Horatio in *Hamlet*

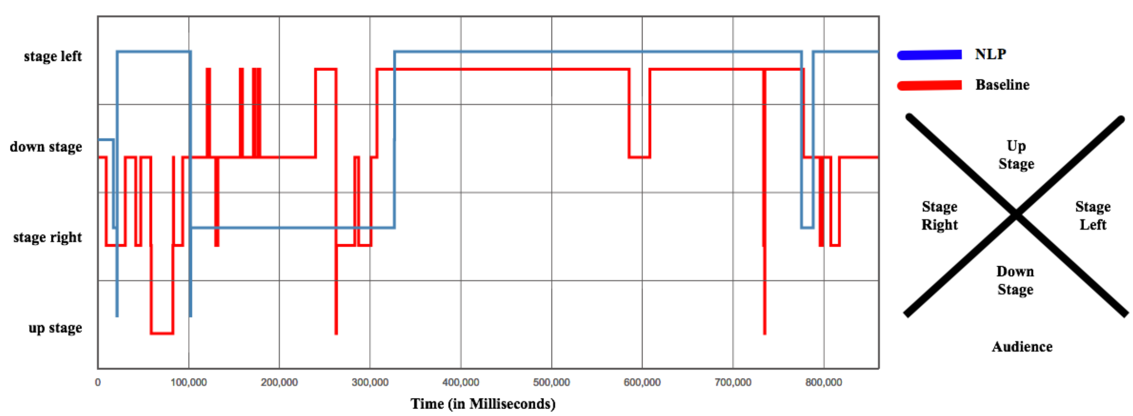


Figure 6.12: Character Gaze Traces for Gravedigger1 in *Hamlet*

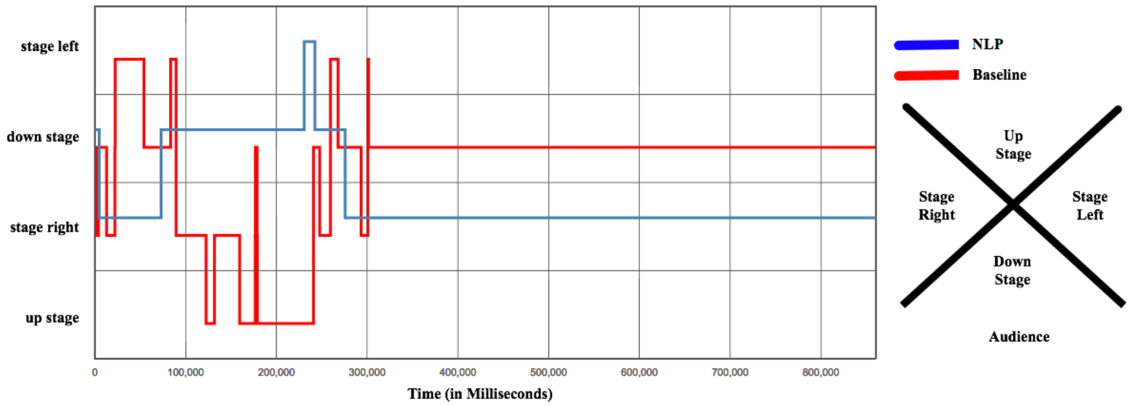


Figure 6.13: Character Gaze Traces for Gravedigger2 in *Hamlet*

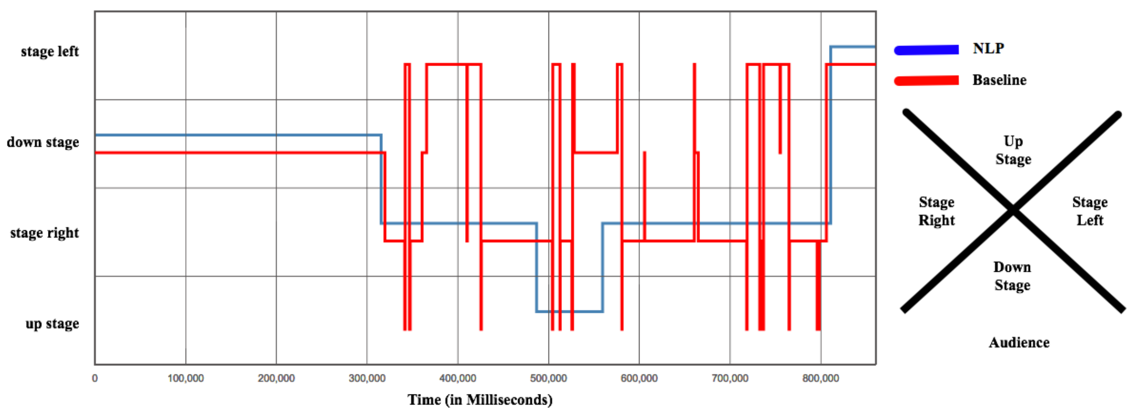


Figure 6.14: Character Gaze Traces for Hamlet in *Hamlet*

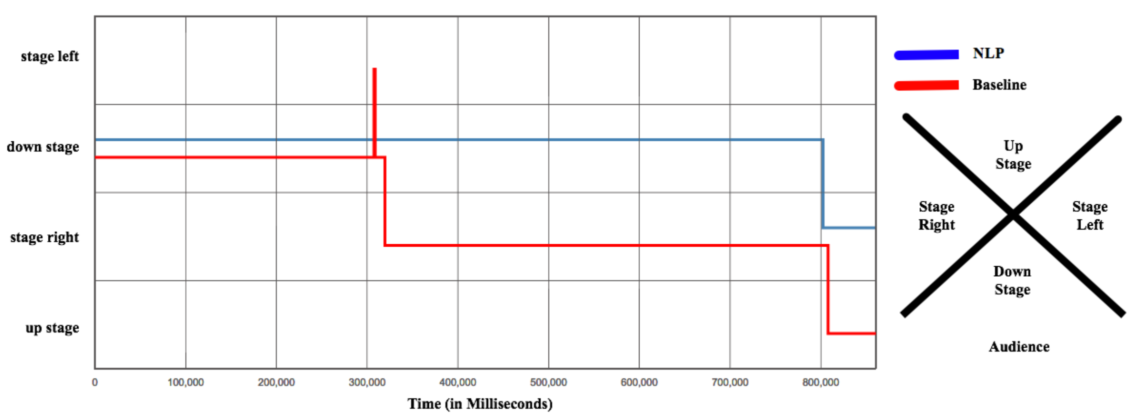


Figure 6.15: Character Gaze Traces for Horatio in *Hamlet*

items, many statements end up being irrelevant for our 2D model and end up being discarded. For instance, "(laughing)," or "(The sound of the bell fades out)" have no actions in a 2D world without sound; however still require processing to determine the sentence is irrelevant for this work.

Compound statements (e.g., "He is followed by GRAVEDIGGER2, who carries a T-spade and a pick and whistles"), although uncommon in this particular play-script, do cause issues with performing all directed actions. We split this text using the punctuation, such as the comma in this statement, but the "and" which indicates multiple items that the Gravedigger2 carries gets dropped when using our actor action target structure. Additional work could be done to accomodate such compound statements, but for this work we focused on the more simplistic statements for annotations.

Some additional issues also arose from the fact that this script was written in British English and the dictionary utilized (WordNet) was American English (due to availability). Words with multiple meanings or word types caused some confusion, such as steps (verb or noun), hands (verb or noun). Finally, due to the time required for the word lookups, this algorithm lends itself to being a pre-processor for the script and may require parallization to accomodate running in real-time.

6.2.2 Rules

We took the character traces from both our baseline (hand-coded BML based on the Electronovision video [15]) and our natural language processor with a rules engine and compared them. We wanted our new method to result in character positioning as close to our baseline as possible; however, we did not want to penalize for being "close enough."

As can be seen in Figures 6.16, 6.17, 6.18, 6.19, overall we were able to position characters on the stage well, despite the natural language processing issues that come with any machine translation. During analysis, we split the stage into the nine squares to represent the nine general locations on the stage–combinations of: upstage,

Table 6.2: *Hamlet* Character Traces Match for Baseline vs. Rules

Character Name	Gaze Match	Position Match
GraveDigger1	46.07%	83.44%
GraveDigger2	11.76%	84.61%
Horatio	80.01%	95.42%
Hamlet	72.77%	95.70%
Overall	52.65%	89.79%

downstage, center-stage, stage-right, and stage-left.

We found that our method was able to position the characters within 0.12 squares (Euclidean distance) of our baseline BML method and placed them correctly 89.8% of the time on the stage, as seen in Table 6.2. The other 11.1% of the time, the characters in the video added their own unannotated movements to what was directed by the director. For instance, near the beginning of the scene, Gravedigger1 walks towards the audience, then turns around and heads back towards the grave. This movement was not annotated in the play-script and therefore was not performed by our rules-based characters. This highlights one aspect of the actor's initiative to improvise despite the directions provided by the script.

For gaze, we divided the directions into the four basic gaze directions: towards the audience, stage-right, stage-left, and upstage/backstage.Here we found our results did not match as well (as seen in Figures 6.20, 6.21, 6.22, 6.23), with the gaze being correct only 52.7% of the time and, on average, within 0.53 quadrants of our baseline gaze direction.

One key reason for some of the discrepancies in the character traces is due to the input utilized for the baseline versus our method. The baseline BML was written to include movements and motion that were not included in the play-script that our method utilized, but the actors performed. It included some movements based on what was seen in the video, but may not have fully encompassed all the gazes that occurred within the play due to user-translation error. Also, our rules were based on

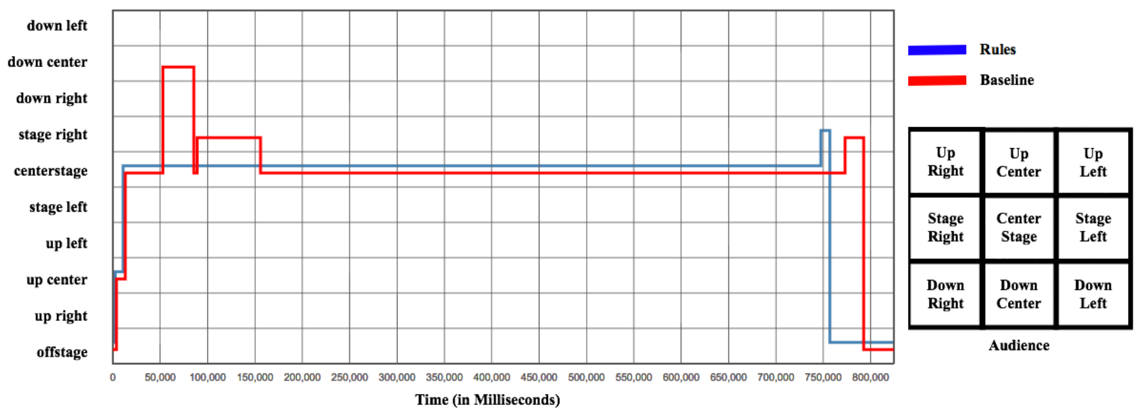


Figure 6.16: Character Position Traces for GraveDigger1 in *Hamlet*

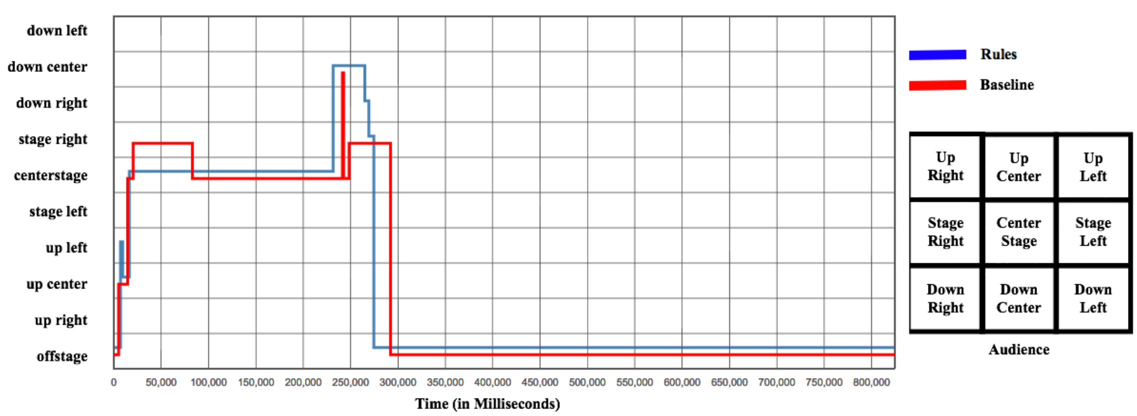


Figure 6.17: Character Position Traces for GraveDigger2 in *Hamlet*

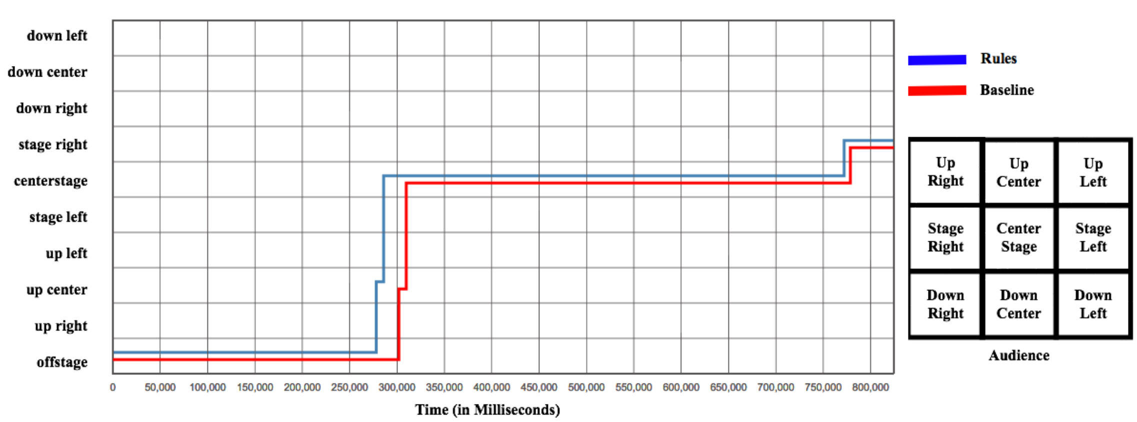


Figure 6.18: Character Position Traces for Hamlet in *Hamlet*

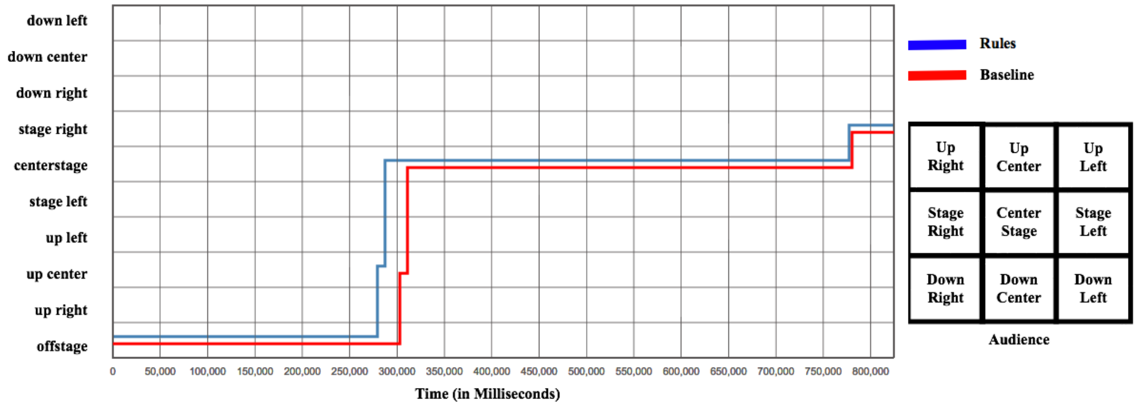


Figure 6.19: Character Position Traces for Horatio in *Hamlet*

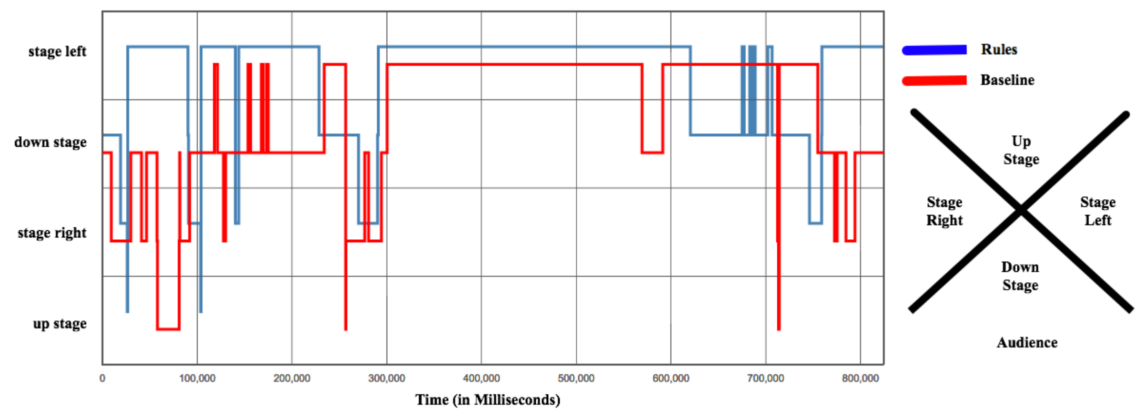


Figure 6.20: Character Gaze Traces for Gravedigger1 in *Hamlet*

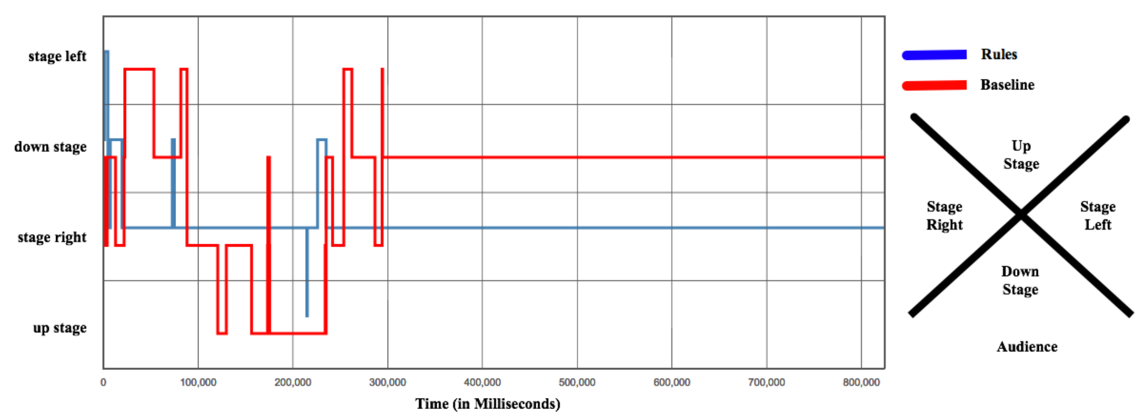


Figure 6.21: Character Gaze Traces for Gravedigger2 in *Hamlet*

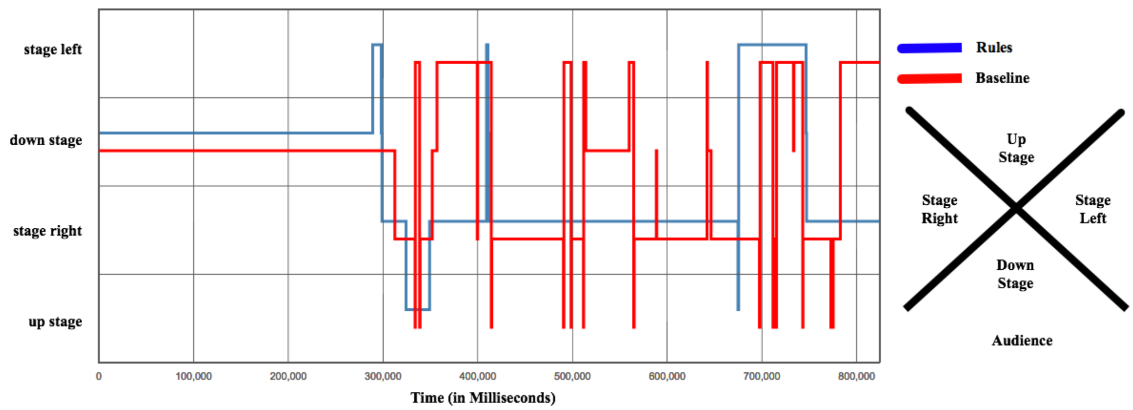


Figure 6.22: Character Gaze Traces for Hamlet in *Hamlet*

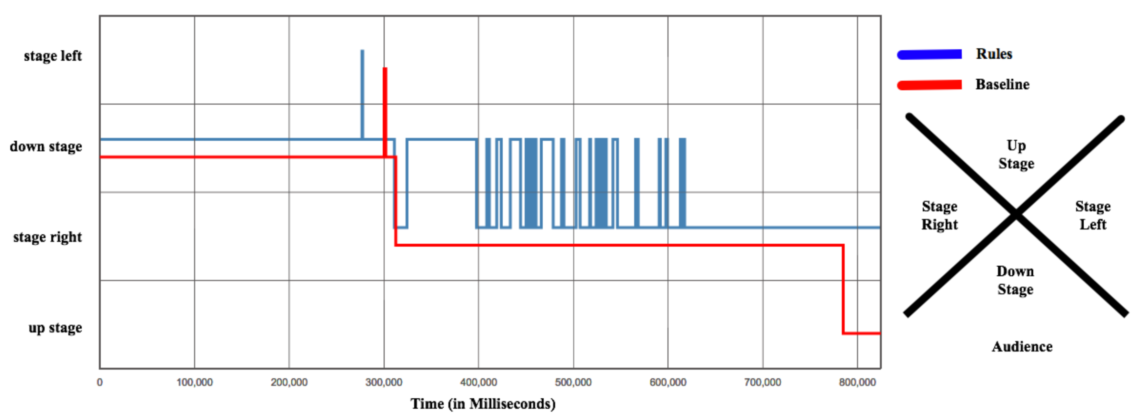


Figure 6.23: Character Gaze Traces for Horatio in *Hamlet*

always performing adjustments with every command that was brought into the rules engine, whereas a real actor may not follow these rules 100% of the time. However, our rules did better than our prior version, which just utilized a natural language processor by approximately 11% for position and approximately 20% for gaze, even though it still incurred similar issues around duality of word meanings and pronouns found in our first experiment. Our biggest impacting change by incoporating rules was around the gaze, since most gazes are not annotated.

6.2.3 Force-Directed Graphs

When incorporating our force-directed graphs (FDG), we hoped to more dynamically arrange the character positions than our rules could do. Our rules would focus on just adjusting for character priority and being too close for conversational space, and was limited in its ability to really look at the overall positioning of all characters on the stage. With our force-directed graph drawing algorithms, we can better balance the overall positioning on the stage and enforce better semi-circular/conversational arrangements, leveraging the adjustments to position done by the rules engine. To validate our algorithm's effectiveness in doing this, we will need to measure its performance against our requirements, which were discussed in Chapter 4. Here we discuss each item and how to measure our algorithm's accuracy. Each measurement must be performed over several configurations, as well as with the human-controlled character moving correctly, incorrectly, and not at all.

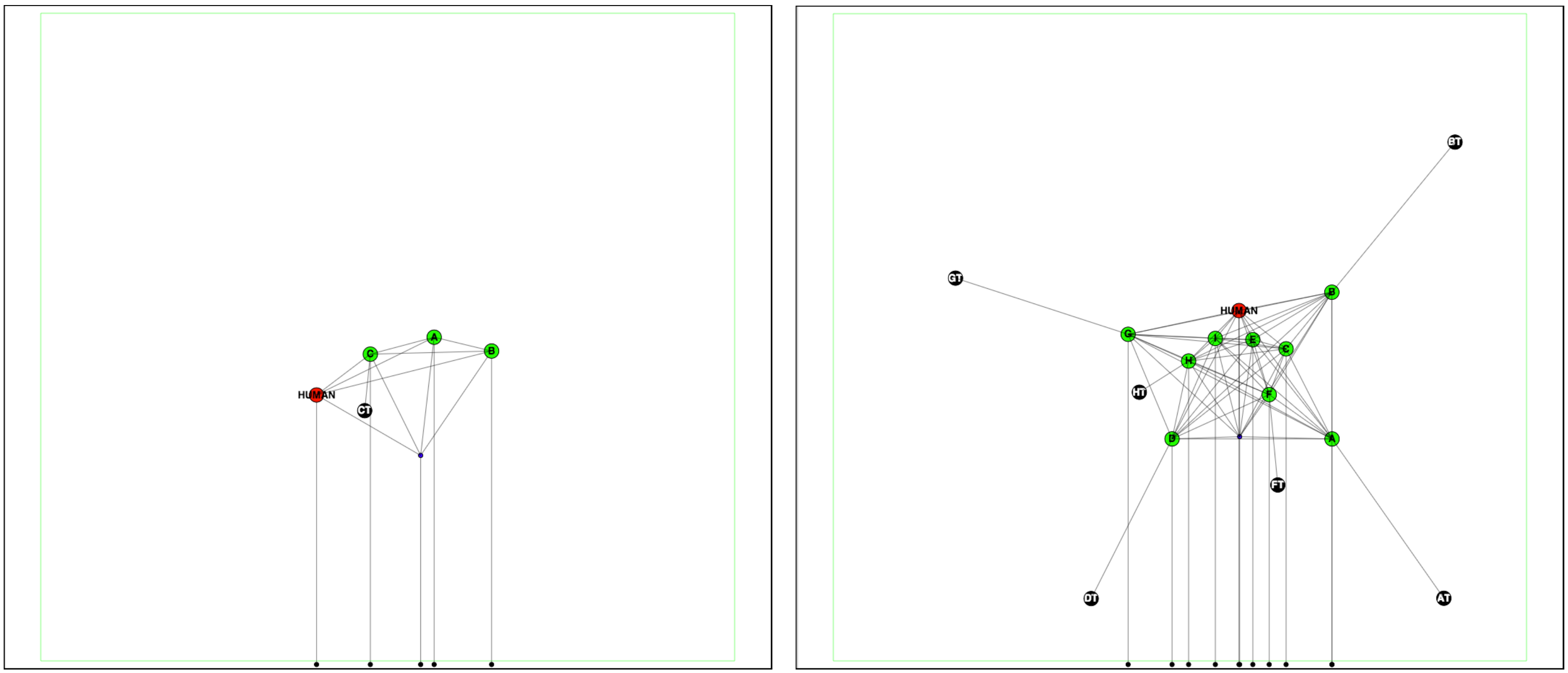
Three main threads of testing need to be performed: validation of positioning from randomized states of the play; validation of positioning sequentially across an entire play; and user studies of perception of the play performance. Each provides a unique validation of the algorithms, which are key to their success in a real virtual environment scenario. The first measurement (randomized states) measures the overall positioning of the characters on-stage for a single point in time, such as symmetry, centering, and force variations.

However, the second measurement (sequential states) verifies that the continuity of the play is preserved, despite human intervention, such as oscillation-free adjustments and decreasing strength of relationships over time. This second measurement is much more difficult to perform as it requires a comparison against a baseline for both correct and incorrect positioning of a single character on-stage with respect to the rest of the characters. The third measurement (user studies) provides a key understanding of human perception of these character positioning within the play, and their realism.

6.2.3.1 Features Evaluation

To test our approach, we implemented the algorithms described in the Methodology section in a JSGameSoup javascript application. Here, the stage was represented as a box within the screen, and characters as circles with connecting lines representing their relationships. One character (the human-controlled character) could be moved by dragging it across the screen, to represent the human-controlled character. Numerous scenarios were tested by randomly placing characters, pawns, and the human character on the screen and applying the force-directed graph drawing algorithms to arrange them on the screen, as seen in Figure 6.24.

Each AI character (in green) is connected to every other AI character on the stage, the human-controlled character, and the audience. They are also sometimes connected to a target position, or pawn, to indicate their correct "mark" on the stage for this moment in the play. These target point connections have no repellent forces, but very strong attractive forces (β) applied to them. Also, if there is at least two AI characters on the stage, they are also connected to a center point, which only has attraction forces applied to it. The human character is also tied to this center point (if it exists), the audience, and every character on the stage. The relationship between the human-controlled character and the AI characters has a weaker attraction force than the between-AI character forces. The center point is given a stronger tendency to be in the front quarter of the stage than any of the AI or human-controlled characters



(a) Force-Directed Layout of Three Characters

(b) Force-Directed Layout of Nine Characters

Figure 6.24: Character Positioning Using Forces
Red=Human; Green=AI Character; Black=Target; Blue=Center Point

to help force the semi-circular arrangement on the stage, as well as an opening in the grouping, which faces the audience. Each force is described further in Table 4.1 in Chapter 4.

These forces allowed for a relatively consistent positioning distance of about 3.14 (SD=1.54) feet between the different characters, which provided our "Even Vertex Distribution" as described above. Even with 12 characters plus the one human-controlled character on the screen, we had at most 40 vertices in our graph, which kept us within reasonable limits for the Fruchterman and Reingold algorithm approach. We observed that the fixed points (also known as the AI characters' target destination) pulled the characters toward them, which helped to minimize movement of the character from their mark. This distance averaged at 494.56 drawing units apart, which represents about 3.30 (SD=1.52) feet of spacing, and was relatively consistent across the different sized character groupings tested (1-12) as seen in Table 6.3.

Table 6.3: Average Distances Between Characters and Pawns

Number of Characters	**Character Connected To:**	**Average Distance (in Drawing Units)**	**Average Distance (in Feet)**
1	audience	370.63	2.47
1	human	526.30	3.51
1	target	496.80	3.31
2	audience	515.92	3.44
2	center	270.80	1.81
2	char	400.74	2.67
2	human	488.35	3.26
2	target	456.22	3.04
3	audience	449.66	3.00
3	center	355.34	2.37
3	char	414.75	2.77
3	human	662.09	4.41
3	target	448.98	2.99
4	audience	528.01	3.52
4	center	296.00	1.97

Continued on next page

Table 6.3 – Continued from previous page

Number of Characters	**Character Connected To:**	**Average Distance (in Drawing Units)**	**Average Distance (in Feet)**
4	char	424.55	2.83
4	human	468.30	3.12
4	target	555.88	3.71
5	audience	419.66	2.80
5	center	378.33	2.52
5	char	529.04	3.53
5	human	466.43	3.11
5	target	611.72	4.08
6	audience	425.97	2.84
6	center	368.86	2.46
6	char	482.32	3.22
6	human	484.72	3.23
6	target	464.94	3.10
7	audience	436.75	2.91
7	center	348.79	2.33
7	char	469.19	3.13
7	human	502.33	3.35
7	target	428.91	2.86
8	audience	415.04	2.77
8	center	401.58	2.68
8	char	564.30	3.76
8	human	462.93	3.09
8	target	487.51	3.25
9	audience	498.57	3.32
9	center	439.40	2.93
9	char	511.48	3.41
9	human	502.24	3.35
9	target	489.51	3.26
10	audience	482.81	3.22
10	center	376.99	2.51
10	char	483.21	3.22
10	human	480.38	3.20
10	target	521.40	3.48
11	audience	496.43	3.31
11	center	394.77	2.63
11	char	478.84	3.19
11	human	495.07	3.30
11	target	480.68	3.20
12	audience	476.52	3.18

Continued on next page

Table 6.3 – Continued from previous page

Number of Characters	Character Connected To:	Average Distance (in Drawing Units)	Average Distance (in Feet)
12	center	403.22	2.69
12	char	477.78	3.19
12	human	498.51	3.32
12	target	493.62	3.29

The forces were varied between the audience, the center point, the target pawns, the AI characters, and the human character, with the target pawn connections being the strongest attraction (with no repelling forces), and the AI character interrelationships being the strongest repellent forces. This, in conjunction with the center point, provided balance with the positioning and provided semi-circle positioning for the smaller number of characters on the stage, as seen in Figure 6.25b. However, for the larger character groups, they often formed a more circular arrangement, with the audience side not quite being enclosed, as seen in Figure 6.26b.

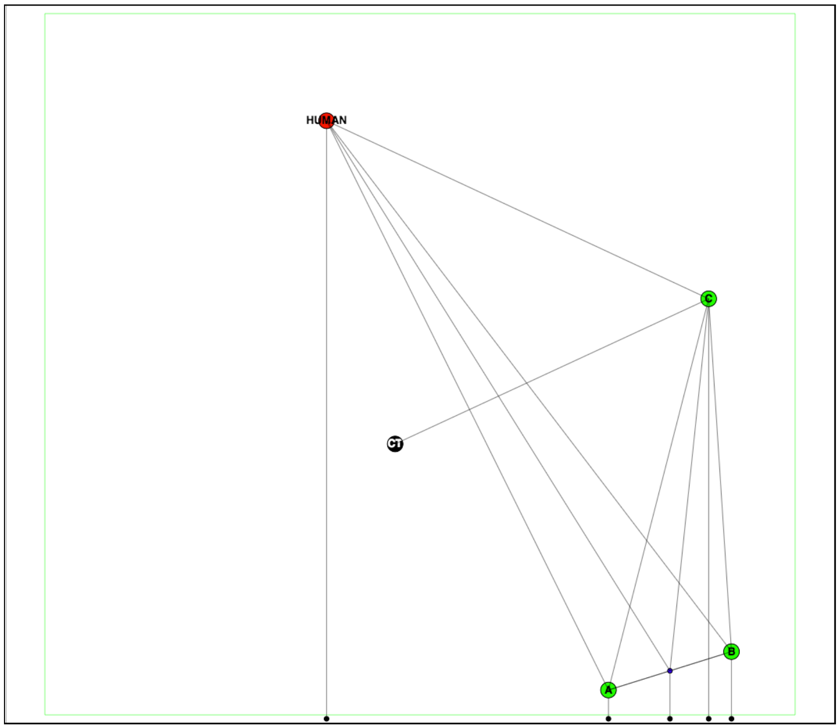
Additional work will be pursued to test the varying relationship strengths over time, as well as the impact of oscillations between arrangements. Some preliminary testing indicated issues with oscillations of character positioning where characters would swap places, but still maintain the overall layout on the stage. Also, more force manipulations may achieve better results than found during this particular experiment.

6.2.3.2 Scene Evaluation

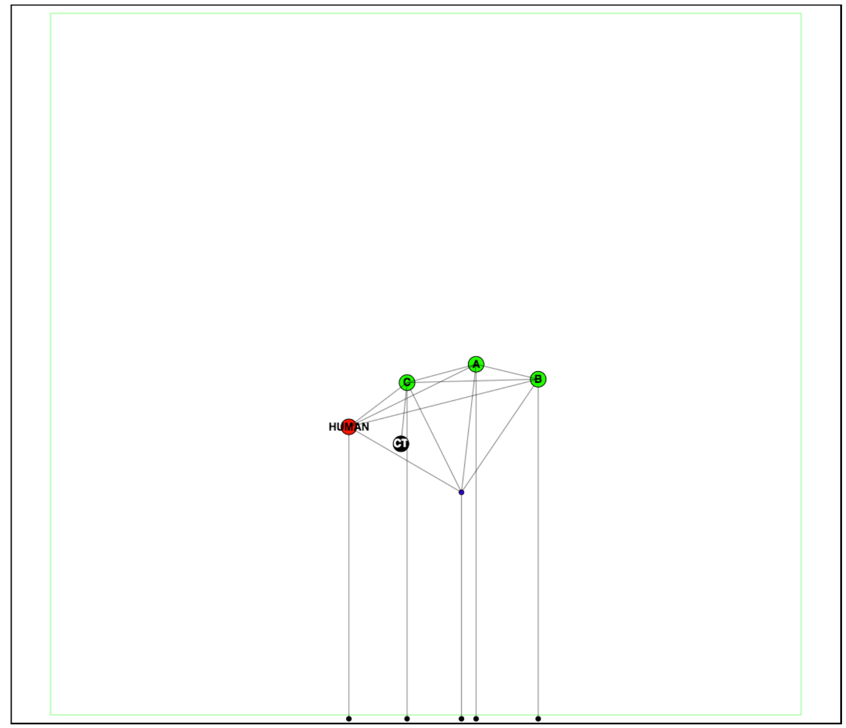
To evaluate the effectiveness of the force-directed graphs for positioning characters throughout a scene, we take two approaches:

1. Direct comparison with the 1964 *Hamlet* production

2. Incorporation of a human-controlled character

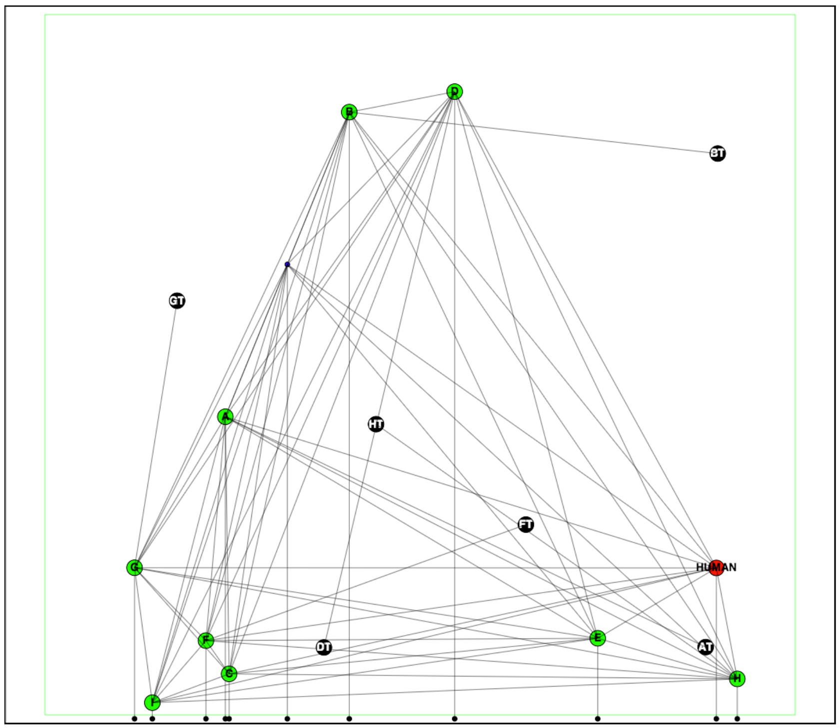


(a) Random Layout
of Three Characters

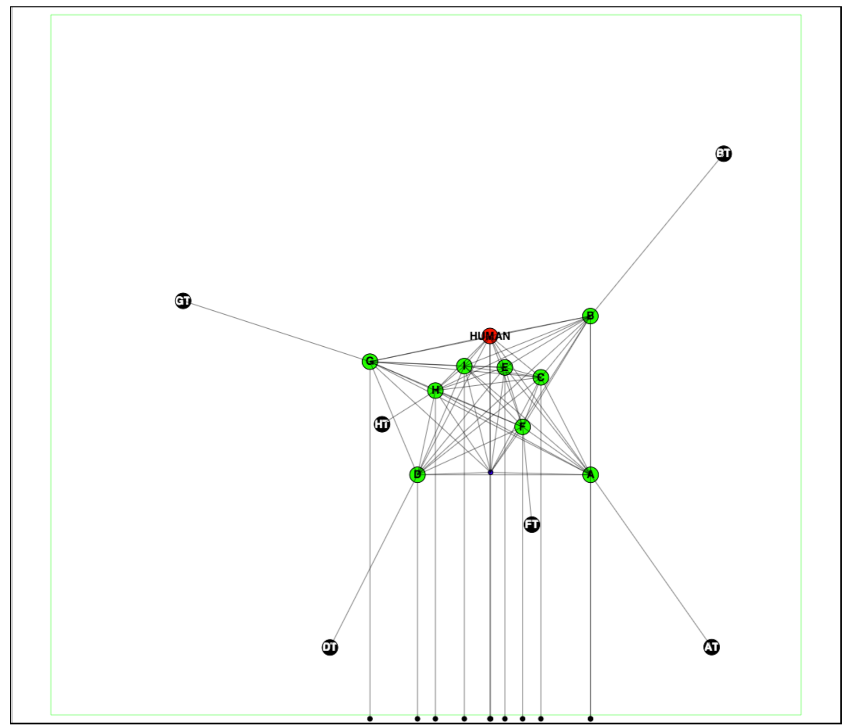


(b) Adjustment of Figure 6.25a
Using Force-Directed Graphs

Figure 6.25: First Example: Character Positioning Using Force-Directed Graphs
Red=Human; Green=AI Character; Black=Pawn; Blue=Center Pt



(a) Random Layout
of Nine Characters



(b) Adjustment of Figure 6.26a
Using Force-Directed Graphs

Figure 6.26: Second Example: Character Positioning Using Force-Directed Graphs
Red=Human; Green=AI Character; Black=Pawn; Blue=Center Pt

The first comparison involves comparing the positioning of characters (all assumed to be AI characters) using our force-directed graphs with our baseline positioning of characters from the same *Hamlet* scene in the Broadway production by Sir John Gielgud in 1964. These are compared for the criteria of occlusion and clustering of characters. This provides a baseline for comparison for the next experiments, which highlight the visual balance that audiences appreciate in imagery.

To further our baseline, we also incorporate one of the characters from the scene as a human-controlled character and vary their accuracy in following the play-script as written. This provides us with a secondary comparison to evaluate the effectiveness of including the human-controlled character with our force-directed graph approach versus the hard-coded play-script approach that is most commonly used today.

Next, we incorporate a human-controlled character and vary their desire to follow the play-script through different runs. We then compare these runs with the same criteria of occlusion and clustering. The intent is that a similarity in the amount of occlusion and clustering should be maintained, regardless of the human-controlled character's movements. This will show that we are able to adjust our positioning to include a human-controlled character, yet still maintain the integrity of the play-script as much as possible.

The human character's movements are simulated by allowing them to move at the right times, but not to the right locations. This is based on how accurately we allow the human to follow the play-script. The more accurate the human is, the more likely they will follow the play-script perfectly. However, when they choose not to follow the play-script, we choose a random location on the stage for the human to move to during that moment, which does not coincide with the play-script.

To evaluate our methods, we have chosen to utilize two criteria: occlusion and clustering. With occlusion, we are looking to avoid the overlap of characters onstage from an audience's perspective. We do not wish to obscure the audience's view of the scene by misplacing a character onstage and block another character. To calculate this, we will assume an orthographic projection for the audience's view of the characters onstage, due to the small variance in viewing angle in a typical theatre. This allows us to use the character's x-position onstage with a buffer to indicate their coverage area for occluding another character. Any overlap distance for each character will be summed up and compared to the length of the stage (or potential occlusion area). This can be seen in Equation 6.1, where we sum the overlaps of each character and divide by the length of the stage, to determine the amount of occlusion across all characters. Each character was given a defined radius to represent the area they would cover while onstage, and was leveraged to determine each character's minimum and maximum coordinates.

$$\sum_{i=0}^{count} \sum_{j:i+1}^{count} \frac{char[i].maxX - char[j].minX}{stageLength} \begin{cases} > 0; & char[i].maxX - char[j].minX \\ \leq 0; & 0 \end{cases} \tag{6.1}$$

The second criteria, clustering, is used to ensure we are not clumping everyone too close together, leaving a large portion of the stage unused. To calculate this, we will simply take the range in both the x and y dimension on the stage to determine the percentage of the stage being utilized in both width and depth. This can be seen in Equations 6.2 and 6.3, where we take the min and max values of both x and y across all characters and divide by the length of the stage in that dimension, to determine the amount of space covered by the characters.

$$\frac{Max_{\forall i}(char[i].x) - Min_{\forall i}(char[i].x)}{stageLengthX} \tag{6.2}$$

$$\frac{Max_{\forall i}(char[i].y) - Min_{\forall i}(char[i].y)}{stageLengthY} \tag{6.3}$$

We will look to minimize the occlusion equation and maximize the two clustering equations to determine quality of the spatial positioning for the scene. We avoid leveraging a more volumetric measure for clustering due to its loss of specificity, since a 10x4 volume would be seen the same as a 20x2 volume, but reflects very different clustering in each dimension.

To evaluate our approach, we ran the experiments as described above. We started with a baseline reading, which utilized the hand-mapped blocking from the 1964 *Hamlet* production on Broadway. We averaged the positioning for each moment across the entire scene for both occlusion and clustering. As can be seen in Table 6.4, we have some minor occlusions of the characters on the stage with that handmapped production, at over three percent. There is also a fair amount of clustering in both dimensions of the stage as well (∼20% along the length of the stage and ∼11% along the depth of the stage).

Table 6.4: Experiment Results of Occlusion and Clustering Averaged Over Scene

Case #	**Case Description (Including Accuracy of Human)**	**Average Occlusion Frequency**	**Average Clustering X**	**Average Clustering Y**
0	Baseline All AI	3.6%	19.5%	14.6%
1	Baseline Human 90%	3.6%	19.1%	15.4%
2	Baseline Human 50%	2.9%	20%	14.7%
3	Baseline Human 10%	4.4%	30.9%	28.7%
4	Forces All AI	2.4%	16.8%	14.6%
5	Forces Human 90%	2.4%	16.8%	14.6%
6	Forces Human 50%	1.6%	20.4%	13.8%
7	Forces Human 10%	2.4%	20.8%	14.0%

When we take a look at our method of controlling all the characters to follow a play-script, we see that we are able to reduce the frequency of characters being occluded on the stage when all the characters are controlled by the AI. We still have the clustering of the characters, and they now occupy less space than we saw with the baseline measurements. We notice that the characters appear to cluster together more with our force-directed graphs than with our hard-coded AI character blocking. This reveals that the human-controlled character is being included in the AI characters' positioning when we use the force-directed graphs, but is separated from the other characters when the AI characters just perform the scene as-is (no adjustments for the human character).

Considering the scene we utilized has at most three characters onstage at any time, we expect to see normal clustering at approximately 28% if we utilized only conversational space for positioning the characters side-by-side. The *Hamlet* production from 1964 produces slightly tighter clustering due to the nature of the scene (characters are focused on the grave). As we introduce the human-controlled character, we see less clustering, which reveals that the human-controlled character is not being included in the AI characters' positioning. However, when we look at the force-directed graph approach, the characters are able to cluster better and include the human-controlled character, which is revealed by the smaller clustering numbers.

We also see that having all the characters behaving correctly provides very similar clustering results to when we have an errant human-controlled character (Forces Human 10%, where the human only follows the script 10% of the time) when we utilize the force-directed graphs. However, with the hard-coded AI character blocking (Baseline Human 10%, where the human only follows the script 10% of the time), we see a jump in the amount of clustering of the characters. This shows that the force-directed graphs not only help to include the human, but is also able to maintain the integrity of the script.

6.3 Implied Movement

When looking at the accuracy of our system when compared to the baseline, we noticed we were unable to match the real performance 11% of the time. This led us to conjecture that perhaps there is something in what the character is saying, which caused the actor to perform this "extra," unannotated movement. We took the entire play-script and captured all the actor movements performed within the real performance. Using several standard machine learning techniques, we attempted to learn if certain phrases would result in a specific movement, or even just a movement in general [93]. Unfortunately, we were unable to learn any of this information from what the characters were saying, partly because of the nature of Shakespeare's writing (iambic pentameter), and partly because of our approach's inability to effectively capture the relationships between the words. We were encouraged by this work because it indicated that these unannotated movements may not be in what is being said, but instead is just an actor's whim during the performance. For more information, refer to Appendix A: LACK OF SPATIAL INDICATORS IN *HAMLET*.

6.4 Qualitative Evaluation

We also performed some qualitative analysis on our system to determine whether we are able to provide a realistic performance, which is similar from a viewer's perspective. We created a spatio-temporal-focused survey to evaluate our performances. This survey was based upon other theatre evaluation tools, including several one-act play competition judging criteria. We performed three different studies, one for the handmapped baseline video versus an NLP generated video using an interval scale as a between subjects study. The other two studies compared all five different components: Random, Baseline, NLP, NLP+Rules, and NLP+Rules+FDG. One of these was a within subjects study (with a shorter questionnaire) and the other was a between subjects study.

6.4.1 Evaluation Tools

Because of the current lack of a single existing tool to qualitatively evaluate only the spatio-temporal reasoning within a performance, we needed to create one from scratch. We wanted to find techniques that were used for evaluating performances in general, in the hopes of finding some embedded spatio-temporal criteria within them. We did find that one-act play competitions are critiqued by judges, and include spatial aspects of the performance in their evaluations. Therefore, we reviewed these evaluation criteria that are used in one-act performance competitions.

One group we looked at was the Georgia High School judging sheets for one-act plays. The criteria defined in the judges evaluation sheets included: movement, composition, listening, response, and ensemble criteria. Movement is an obvious tie-in to analyzing the spatial aspects of a performance, so was included in our evaluation tool. The judges typically verify if the movement within the performance is motivated and free of distractions. With composition, the plays are evaluated on how the performers convey the theme and mood of the play, and whether the movements of the performers aid in providing proper dramatic emphasis. There is also a notation on the variety and balance in the use of the stage space, which is included in the judges' checklist. Finally, reviewers are asked if the performers appear to work together and be involved in group events. These criteria, along with others within the Georgia High School judging sheets, were key questions included in our survey (as seen in Table 6.5, questions LSQ-9 to LSQ-13 and LSQ-20). [4]

We also evaluated the Texas University Interscholastic League's one-act play official standards. The UIL's judging packet is much more comprehensive and included more detailed guidance on each of the criteria for evaluation a one-act play. Some important evaluations were described around characterization, movement, timing, business (exits and entrances), and composition. We added several questions regarding the believability of the characters' movements, whether the movement appears random, the overall pace of the performance, and whether the characters frequently blocked each other. These can be seen reflected in our survey in Table 6.5 as questions LSQ-1 to LSQ-8, LSQ-14 to LSQ-19, and LSQ-21 to LSQ-25. [68]

The questions within the survey were intentionally asked two or more times with

Table 6.5: Spatio-Temporal Likert Scaled Questions (LSQ) in the User Survey

LSQ-1	Characters showed evidence of engaged listening
LSQ-2	Characters appeared to perform suitable movements on cue
LSQ-3	The pace of the performance was too fast
LSQ-4	The pace of the performance was too slow
LSQ-5	The use of the space on stage was appropriate
LSQ-6	The blocking (positioning and timing of the characters) was appropriate
LSQ-7	There was adequate variety in the staging positions of the characters
LSQ-8	The characters' movement onstage during the performance was believable in the context of the performance
LSQ-9	The performance is free from distracting behavior that does not contribute to the scene
LSQ-10	The arrangement of the performers appropriately conveys the mood of the scene
LSQ-11	The character movements provide appropriate dramatic emphasis
LSQ-12	There is adequate variety and balance in the use of the performance space
LSQ-13	All visible behaviors appear to be motivated and coordinated within the scene
LSQ-14	The characters were grouped to give proper emphasis to the right characters at the right time
LSQ-15	The characters frequently covered or blocked each other from your point of view
LSQ-16	The movements of the characters were consistent with the play
LSQ-17	There was a great deal of random movement
LSQ-18	The characters' reactions to other characters were believable
LSQ-19	Characters showed a lack of engagement when listening
LSQ-20	The arrangement of the performers contradicts the mood of the scene
LSQ-21	The more prominent characters in the scene were hidden or masked from your view
LSQ-22	The characters were too close together
LSQ-23	The characters were too far apart
LSQ-24	The stage space was not utilized to its full potential
LSQ-25	All characters were visible from your point of view throughout the scene

Table 6.6: Correlation of Spatio-Temporal Questions

Question #s	Spatial Component Covered
$1 \approx \neg\ 19$	Engaged Listening
$2 \approx 6$	Pace of Performance
$16 \approx \neg\ 17$	Appropriate Movement and Timing
$3 \approx \neg\ 4$	Consistent Movement
$5 \approx \neg\ 24 \approx 12$	Space Usage
$7 \approx \neg\ 22 \approx \neg\ 23$	Variety and Closeness
$9 \approx \neg\ 13$	Motivated Movement
$10 \approx \neg\ 20$	Scene Mood
$14 \approx \neg\ 21 \approx 11$	Character Emphasis
$15 \approx \neg\ 25$	Visible Characters
$18 \approx 8$	Believable

different phrasing to alleviate any bias presented in the wording of the question. The expected correlation between the questions can be seen in Table 6.6.

Additional, open-ended questions were included in the survey to reveal any quality issues that were not covered in the above questions. These questions were primarily pulled from Pavis's questionnaire [71]. Her questions are more open-ended, and meant to guide the spectator in describing the aesthetic experience and overall production after seeing it. Some key spatio-temporal questions are included in Pavis's questionnaire, such as: space organization, relationships between actors, and pacing. We found it most useful for providing open-ended questions within the survey, which can be seen in Figure 6.27 as questions OEQ-2 to OEQ-4 below.

Lastly, we referred to The Theatre Handbook written in conjunction with several theatre groups: Independent Theatre Council (ITC), The Society of London Theatre, and Theatrical Management Association (TMA). This handbook provided useful recommendations around grouping questions for evaluating a performance's quality, such as the frequency of attending performances, and the use of self-rating with a newspaper's five-star scale [69]. These can be seen in Figure 6.27 as questions OEQ-1 and OEQ-5.

Our demographic questions included the ones seen in Figure 6.28.

OEQ-1	If you were reviewing this production for tomorrow's papers, how many stars would you give it? (1 star = lowest rating, 5 star = highest rating)
OEQ-2	Did anything in the production not make sense? What was it and why?
OEQ-3	Were there any special problems that need examining? What were they and why?
OEQ-4	Were there any particular strong, weak, or boring moments in the scene? What were they and why?
OEQ-5	Any other comments?

Figure 6.27: Spatio-Temporal Open-Ended Questions (OEQ) in the User Survey

6.4.2 NLP versus BML Between Subjects Evaluation

We presented 3D videos of both the hand-mapped production from Broadway in 1964 [15, 86] and a simple natural language processing interpretation of the same play-script to users for a between groups comparison. Both videos included a block world where characters and pawns within the scene are represented by blocks, as seen in Figure 6.29. This eliminated any bias regarding human versus virtual or block characters, as well as any differences in camera positioning throughout the recordings. This also helped viewers to focus on the spatial aspects of the performance instead of any animations or character representations. The characters are able to point, move, gaze, pick up objects, put down objects, carry objects, and speak.

Each group viewed only one of the videos and answered the questions about the spatio-temporal reasoning included within the video (between groups experiment). The questions were presented in randomized order to the users after viewing the video.

The survey was posted on Mechanical Turk (MTurk) with criteria to enforce participants were from the United States, to avoid cultural differences of opinion regarding space and performances. Additional controls were put in place to ensure participants

- In what state or U.S. territory do you live?
  - US States and Territories
  - Other - Outside the U.S.
- Which category below includes your age?
  - 17 or younger
  - 18-20
  - 21-29
  - 30-39
  - 40-49
  - 50-59
  - 60 or older
- What is your gender?
  - Male
  - Female
  - Non-binary
  - Prefer not to answer
- What is your employment status?
  - Employed, Full-time
  - Employed, Part-time
  - Student
  - Retired
  - Unemployed
  - Other
- What culture do you relate most to?
  - American
  - Arabic
  - Chinese
  - English
  - French
  - German
  - Italian
  - Japanese
  - Korean
  - Portuguese
  - Russian
  - Spanish
  - Other
- Over the last 12 months, roughly how many times have you been to see a theatre performance (including opera, musical, play, dance)?
  - 0
  - 1-3
  - 4-10
  - 11+
- In the past 7 days, roughly how many hours have you spent playing video games (e.g., gaming consoles, mobile phones, computers, etc.) involving virtual characters?
  - None
  - 1 to 3 hours
  - 4 to 6 hours
  - 7 to 9 hours
  - 10 hours or more
- How familiar are you with theatre, performances, and theatre terminology?
  - Very Familiar
  - Familiar
  - Somewhat Familiar
  - Not Familiar
- Are you familiar with the *Hamlet* play and / or the "Graveyard" Scene prior to today's showing?
  - Read / seen it multiple times
  - Read / saw it once
  - Heard of it
  - Never read, seen, or heard of it
  - Other

Figure 6.28: Demographic Questions in the User Survey

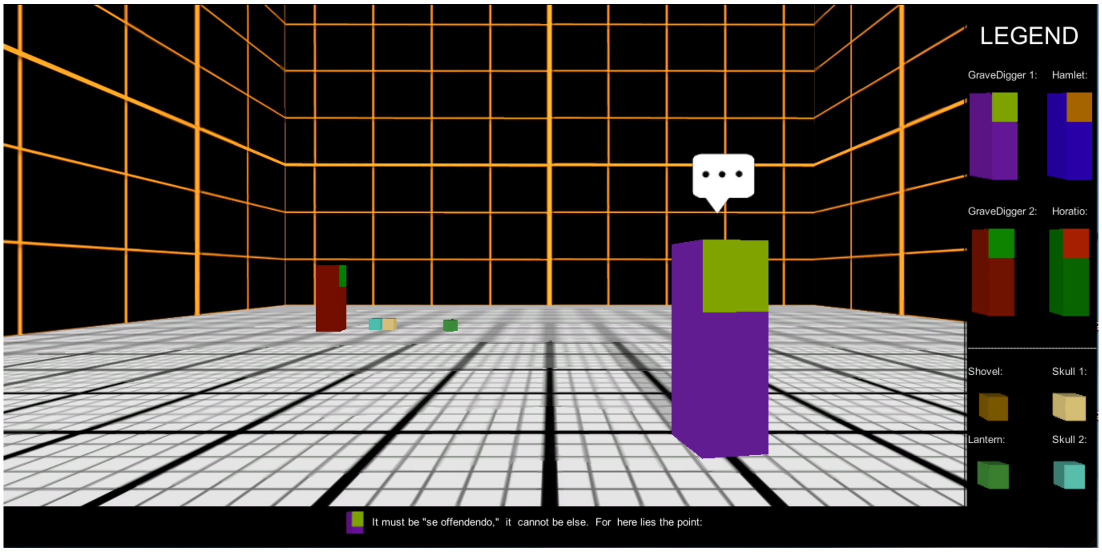


Figure 6.29: Block World Representation Utilized in 3D Videos

viewed the entire video by including a timer on the video viewing page, and including an intermission screen of a particular color mid-way in the video. If the participant did not remain on the video page long enough to watch the video, or if they did not know what color the intermission screen was, they were disqualified from participating. Over 748 participants attempted the survey, with only 214 completing it due to the checks put in place within the survey. Figure 6.30 shows the breakdown of the participants by the different demographics. As you can see, it represents a reasonable sampling of the population.

The study included 214 participants who were asked to evaluate the spatial aspects of a recorded video, as described in the Approach section. 978 people attempted the study, but were unable to complete the study due to the controls in place to ensure proper participation, such as time spent watching the video, and identifying the intermission screen color correctly. Twenty-five questions were asked of each participant regarding different components of the video, ranging from 1 (Strongly Disagree) to 5 (Strongly Agree), all on an interval scale. The sample of 214 responses was split relatively evenly between the two groups: one viewing the hand-mapped version from the 1964 production of *Hamlet* on Broadway (Baseline https://www.youtube.com/

(a) Gender

(b) Age

(c) Residence

(d) Employment

(e) Culture

(f) Games Participation

(g) Theatre Performances

(h) Scene Familiarity

Figure 6.30: Demographic Breakdown of Participants for the *Hamlet* Baseline-NLP Between Subjects Study

embed/fY77-8VjSFY); the other viewing a natural language processing interpretation from the play-script (NLP https://www.youtube.com/embed/Vjgf12niRRY). There were 108 participants viewing the Baseline video, and 106 participants viewing the NLP video. Within these groups, we had 141 females and 75 males participate.

6.4.2.1 Analysis

This experiment provided an estimated power to detect a medium effect (d=0.5) of >0.95. Since the power of this experiment is relatively high, the chances of committing a Type II error is extremely low. We performed a two-tailed, two independent samples t-test on the data gathered. The null hypothesis states that there is no difference in the means between the two groups.

These two groups were compared and revealed that there is a significant difference between the Baseline group (M=2.70, SD=0.97) and the NLP group (M=2.32, SD=1.07), t(212)=2.74, p=0.007 for the question "There was a great deal of random movement." The 95% confidence interval for the difference between the means was 0.11 to 0.66, so the minimum expected difference would be about a tenth of a point on a five point scale. This reflects that there is almost no difference between the Baseline and NLP groups regarding whether there was a great deal of random movement within the scene they viewed.

With the question "The more prominent characters in the scene were hidden or masked from your view," we found a significant difference between the Baseline group (M=2.30, SD=0.80) and the NLP group (M=2.58, SD=1.16), t(186)=2.11, p=0.036. The 95% confidence interval for the difference between the means was -0.56 to -0.02, so the minimum expected difference would be about two one-hundredths of a point on a five point scale. This reflects that there is almost no difference between the Baseline and NLP groups regarding whether the prominent characters in the scene were hidden or masked from view in the scene the participants viewed.

We found a significant difference between the Baseline group (M=3.36, SD=0.71) and NLP group (M=3.58, SD=0.68), t(212)=2.36, p=0.019 for the combined question of "Consistent Movement." The 95% confidence interval for the difference between the means was -0.41 to -0.04, so the minimum expected difference would be about two one-hundredths of a point on a five point scale. This reflects that there is almost no difference between the Baseline and NLP groups regarding whether the movement in the scene was consistent.

Looking at the combined question of "Character Emphasis," we found a significant difference between the Baseline group (M=3.44, sD=0.58) and the NLP group (M=3.22, SD=0.75), t(198)=2.47, p=0.014. The 95% confidence interval for the difference between the means was 0.05 to 0.41, so the minimum expected difference would be about five one-hundredths of a point on a five point scale. This reflects that there is almost no difference between the Baseline and NLP groups regarding whether the characters were properly emphasized within the scene.

To recap, with the amount of power included in this experiment, we were able to find four significant differences in means between our two groups (Baseline and NLP). However, these were such minimal differences that, for practicality purposes, are not relevant differences. Hence, we do not observe any differences between our approach of utilizing NLP to perform a script versus a famous production from 1964.

6.4.2.2 Correlation

We explored the relationships between the twenty-five questions asked of the participants to determine if there was any correlation between the questions, which were in-line with our expected correlations seen in Table 6.6 earlier in this chapter. These comparisons were evaluated using Pearson Correlation coefficients with a two-tailed test on the data gathered. We have an estimated power to detect a medium effect (r=0.3) of >0.99.

There was a significant relationship between several questions asked both as a medium effect and a large effect. Here, we will focus on the large effects for the relationship between the questions. For "The characters' movement onstage during the performance was believable in the context of the performance" (CharMvmtBelievable) ($M=3.29$, $SD=1.06$), and "Characters appeared to perform suitable movements on cue" ($M=3.42$, $SD=0.98$), $r(212)=+0.53$, $p<0.001$. We also found that the question, "The blocking (positioning and timing of the characters) was appropriate" ($M=3.35$, $SD=1.00$), is also positively correlated with the CharMvmtBelievable question, $r(212)=+0.59$, $p<0.001$. "The arrangement of the performers appropriately conveys the mood of the scene" ($M=3.26$, $SD=0.98$) is also shown as directly correlated to the CharMvmtBelievable question, $r(212)=+0.51$, $p<0.001$.

Additionally, both the question "The movements of the characters were consistent with the play" ($M=3.45$, $SD=0.89$), $r(212)=+0.50$, $p<0.001$, and "The characters' reactions to other characters were believable" ($M=3.08$, $SD=1.11$), $r(212)=+0.505$, $p<0.001$, were positively correlated with the CharMvmtBelievable question. This indicates that the believability of the movements is directly related to the blocking of the play, the arrangement conveying the mood, the consistency of movements, the reactions to characters, and the performance of suitable movements on cue.

"The use of space on stage was appropriate" ($M=3.17$, $SD=1.06$) is correlated with the question "There is adequate variety and balance in the use of the performance space" ($M=3.03$, $SD= 1.06$), $r(212)=+0.542$, $p<0.001$. This shows that variety and balance are directly related to how the space on stage is used.

6.4.2.3 Summary

Overall, when comparing our hand-mapped baseline video to our NLP-generated video, participants found that the Baseline video had more random movement than the NLP video. Also, they found that the NLP video obscured characters more than the Baseline video did. With all other questions not being statistically significantly different, and even the two questions that were statistically significant were opposing on which video was better. Therefore we conclude these two videos are qualitatively similar from a viewer's perspective. There was not sufficient power for this experiment to properly evaluate responses based on demographics, such as gender. We also learned through a conference paper review that likert scales should be evaluated as ordinal, not scale, measures.

6.4.3 Multiple Component-Based Between Subjects Evaluation

We leveraged the same questionnaire to perform a between subjects qualitative analysis between each component of our system, plus a random and hand-mapped version. The goal of this study was to determine whether we are able to provide a realistic performance, which is similar to a human-performed scene from a viewer's perspective. Each group viewed only one of the videos and answered the questions about the spatio-temporal reasoning included within the video (between groups experiment). The questions were presented in randomized order to the users after viewing the video.

The survey was posted on Mechanical Turk (MTurk) with criteria to enforce participants were from the United States, to avoid cultural differences of opinion regarding space and performances. Additional controls were put in place to ensure participants viewed the entire video by including a timer on the video viewing page, and including an intermission screen of a particular color mid-way in the video. If the participant did not remain on the video page long enough to watch the video, or if they did not know what color the intermission screen was, they were disqualified from participating.

The study included 538 participants who were asked to evaluate the spatial aspects of a recorded video, with 25 questions asked of each participant. 1020 people attempted the study, but were unable to complete the study due to the controls in place to ensure proper participation, such as time spent watching the video, and identifying the intermission screen color correctly. The sample of responses was split evenly between five groups: one viewing the hand-mapped version of a real production of *Hamlet* (Baseline); one viewing a random movement version of the same scene

Table 6.7: Multiple Component-Based Between Subjects Participants and Videos

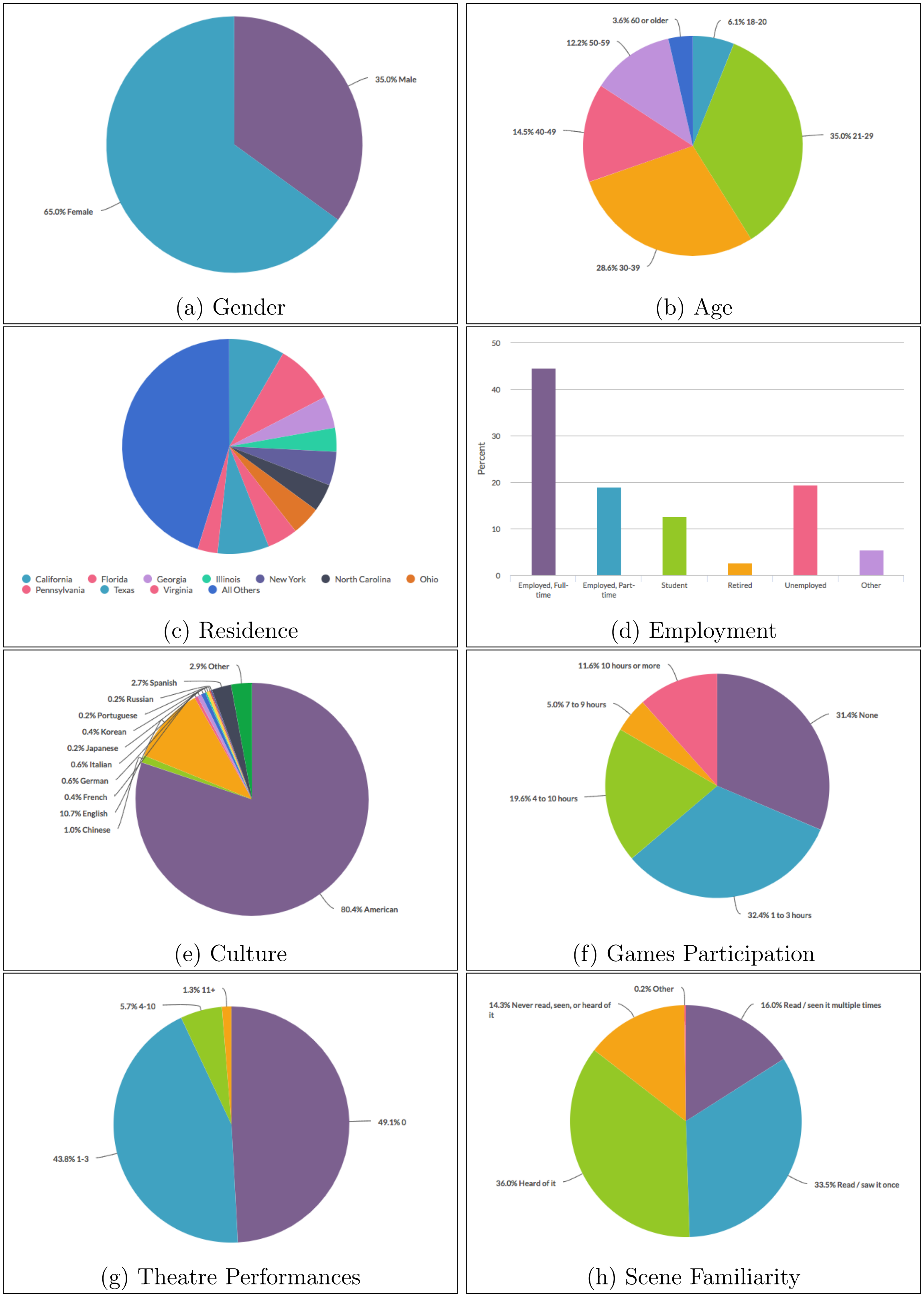
Video	Participants	Video URL
Baseline	99	https://www.youtube.com/embed/fY77-8VjSFY
Random	116	https://www.youtube.com/embed/Xs0TgXA8HtM
NLP	114	https://www.youtube.com/embed/Vjgf12niRRY
Rules	97	https://www.youtube.com/embed/QFO9D_CNcLk
FDG	112	https://www.youtube.com/embed/HHWc-HkDsu4

(Random); one viewing our technique with only the natural language processor component (NLP); one viewing our technique with only the natural language processor and rules engine (Rules); and one viewing our technique with all the components—natural language processor, rules engine, and force-directed graphs (FDG). Table 6.7 shows how many people viewed each video.

Each participant viewed only one video, and answered the 25 Likert questions from the earlier Figure 6.5, with "Strongly Agree," "Agree," "Disagree," "Strongly Disagree," and "I Don't Know" responses. Figure 6.31 shows the breakdown of the participants by the different demographics. As you can see, it represents a reasonable sampling of the population.

This experiment provided an estimated power to detect a medium effect (w=0.3) of >0.99. Since the power of this experiment is relatively high, the chance of committing a Type II error is extremely low. We performed a Kruskal-Wallis H test that showed there was a statistically significant difference in responses between the different videos for 15 of the 25 questions. See Table 6.8 for details on which questions were statistically significantly different, noting that the red-colored questions are phrased negatively, so a higher mean rank would mean more disagreement with the statement, which would mean a better video.

When comparing all the videos except NLP and Random, the Kruskal-Wallis H test showed that there were only two questions with a statistically significant difference in responses: "The characters were too close together" $\chi^2 = 8.655$ $p = 0.013$, "The



(a) Gender
(b) Age
(c) Residence
(d) Employment
(e) Culture
(f) Games Participation
(g) Theatre Performances
(h) Scene Familiarity

Figure 6.31: Demographic Breakdown of Participants for the *Hamlet* Between Subjects Study—All Modes

Table 6.8: Between Subjects Significant Differences and Mean Ranks

Question	χ^2	p	Random	Baseline	NLP	Rules	FDG
Characters appeared to perform suitable movements on cue	14.251	0.007	286.12	274.63	225.63	279.48	283.77
There was adequate variety in the staging positions of the characters	29.951	<0.001	325.88	260.16	222.11	278.30	259.97
The characters' movement onstage during the performance was believable in the context of the performance	23.472	<0.001	308.98	268.91	217.47	278.02	274.71
The character movements provide appropriate dramatic emphasis	11.018	0.026	307.30	268.16	245.79	265.53	259.10
There is adequate variety and balance in the use of the performance space	19.266	0.001	316.26	271.62	241.37	374.38	243.61
All visible behaviors appear to be motivated and coordinated within the scene	13.967	0.007	279.69	272.01	226.85	292.26	280.43
The characters were grouped to give proper emphasis to the right characters at the right time	13.448	0.009	281.88	294.01	235.28	288.28	253.58
The characters frequently covered or blocked each other from your point of view	18.535	0.001	278.46	237.25	313.00	266.62	246.94
The movements of the characters were consistent with the play	11.466	0.022	289.04	278.60	229.57	272.06	279.64
There was a great deal of random movement	13.402	0.009	304.60	263.83	273.10	269.77	234.26
The more prominent characters in the scene were hidden or masked from your view	10.757	0.029	304.13	254.79	276.33	259.44	248.38
The characters were too close together	31.100	<0.001	215.76	293.15	260.52	262.62	319.35
The stage space was not utilized to its full potential	11.817	0.019	237.01	282.47	277.73	253.57	297.10
All characters were visible from your point of view throughout the scene	10.308	0.036	247.43	302.17	251.66	280.91	271.75
The characters were too far apart	22.236	<0.001	312.72	246.41	272.73	279.26	233.40

Table 6.9: Tallied Results of Random, Baseline, NLP, Rules, and FDG Statistically Significant Differences

Compared Videos	Tallied Results	Conclusion
Baseline : NLP	6:0	Baseline Better
Baseline : Rules	0:0	Same
Baseline : FDG	0:0	Same
Baseline : Random	5:5	Same
NLP : Rules	0:7	Rules Better
NLP : FDG	1:7	FDG Better
NLP : Random	2:10	Random Better
Rules : FDG	3:1	Rules Better
Rules : Random	1:4	Random Better
FDG : Random	3:3	Same

characters were too far apart" $\chi^2 = 6.251$ $p = 0.044$. Otherwise, the Baseline, FDG, and Rules videos were not significantly different from each other.

After utilizing several post-hoc Mann-Whitney U tests, we found which combinations of videos had significant differences, and which agreed with the statement more. Table 6.10 shows the specifics of these tests, again with the red-colored questions indicating a negatively worded statement. The video listed in the cell (to the left of the >) is which video had users agreeing more with the statement and was a significant difference, as shown.

Some of these questions are worded negatively (indicated in red), so agreeing more is actually stating that the video was worse than the other one. Taking this into consideration, we find that the questions "The characters frequently covered or blocked each other from your point of view," "There was a great deal of random movement," "The more prominent characters in the scene were hidden or masked from your view," "The characters were too close together," "The stage space was not utilized to its full potential," and "The characters were too far apart" were all negatively worded, and generally show the Random video as being worse, except the use of stage space or spacing. Tallying these up, we see some clear differences, some minor differences, and some that don't appear to be any different, as seen in Table 6.9.

Table 6.10: Pairwise Between Subjects Significant Differences

Question	U	p	Agreed More > Agreed Less
Characters appeared to perform suitable movements on cue	4617	0.013	Baseline > NLP
	4422	0.007	Rules > NLP
	4969	0.002	FDG > NLP
	5158.5	0.002	Random > NLP
There was adequate variety in the staging positions of the characters	4425	0.002	Random > Baseline
	4362	0.005	Rules > NLP
	5415.5	0.037	FDG > NLP
	4085.5	<0.001	Random > NLP
	4612	0.016	Random > Rules
	4813	<0.001	Random > FDG
The characters' movement onstage during the performance was believable in the context of the performance	4914.5	0.049	Random > Baseline
	4608.5	0.016	Baseline > NLP
	4255	0.002	Rules > NLP
	5006	0.003	FDG > NLP
	4367	<0.001	Random > NLP
Character movements provide appropriate dramatic emphasis	5113	0.002	Random > NLP
	4717.5	0.034	Random > Rules
	5326.5	0.014	Random > FDG
There is adequate variety and balance in the use of the performance space	4830.5	0.034	Random > Baseline
	4085.5	<0.001	Random > NLP
	4655	0.021	Random > Rules
	4732.5	<0.001	Random > FDG
All visible behaviors appear to be motivated and coordinated within the scene	4738.5	0.031	Baseline > NLP
	4215.5	0.001	Rules > NLP
	5080	0.005	FDG > NLP
	5272	0.005	Random > NLP
The characters were grouped to give proper emphasis to the right characters at the right time	4442.5	0.004	Baseline > NLP
	4460.5	0.009	Rules > NLP
	5467.5	0.015	Random > NLP
The characters frequently covered or blocked each other from your point of view	4833	0.030	Random > Baseline
	4117	<0.001	NLP > Baseline
	4580	0.022	NLP > Rules
	4857.5	0.001	NLP > FDG
	5654	0.043	Random > NLP

Continued on next page

Table 6.10 – Continued from previous page

Question	U	p	Agreed More > Agreed Less
The movements of the characters were consistent with the play	4644.5	0.020	Baseline > NLP
	4673.5	0.043	Rules > NLP
	5178.5	0.009	FDG > NLP
	5119	0.002	Random > NLP
There was a great deal of random movement	4862	0.040	Random > Baseline
	4773.5	<0.001	Random > FDG
The more prominent characters in the scene were hidden or masked from your view	4702.5	0.014	Random > Baseline
	4684.5	0.023	Random > Rules
	5136.5	0.003	Random > FDG
The characters were too close together	4125	<0.001	Baseline > Random
	5103.5	0.006	FDG > NLP
	5643.5	0.038	NLP > Random
	4561.5	0.008	Rules > Random
	3912	<0.001	FDG > Random
The stage space was not utilized to its full potential	4771.5	0.025	Baseline > Random
	5623.5	0.040	NLP > Random
	5051	0.002	FDG > Random
All characters were visible from your point of view throughout the scene	4556.5	0.005	Baseline > Random
The characters were too far apart	4328	0.001	Random > Baseline
	5496.5	0.039	NLP > FDG
	5676	0.042	Random > NLP
	4529.5	<0.001	Random > FDG

Looking at these tallies, we see that all videos were better than the NLP video by quite a few statistically significant questions. However, all the other videos are the same or only have a couple of statistically significant differences. Also, with seeing that removing both the NLP and Random videos all other videos are the statistically the same, we can summarize that (Baseline = Rules = FDG) > Random > NLP. So both our Rules and FDG techniques provide qualitatively similar performances as the handmapped Baseline video.

The problem with these conclusions is that there is such a lot of variance in the ratings, it leads us to believe that the findings are not as reliable as we'd like. When reviewing further, we realized that doing a between subjects study left participants without a frame of reference for what is "good" versus "bad," and therefore one person's "good" may be much lower than another person's "good." Since we are comparing ranks of un-standardized ratings, we cannot fully understand the results found. Therefore, we looked to repeat this study as a within subjects study, to provide that reference point to compare the various videos against. This would alleviate the differences in each participant's definition of "good," and focus instead on the relative difference between the rankings, which can be more consistent.

6.4.4 Multiple Component-Based Within Subjects Evaluation

Because of the inconsistencies with the between subjects study, we performed a within subjects study of the same videos as before. To alleviate the time-requirements for performing a within subjects study, we leveraged a simplified version of our original questionnaire. The questionnaire consisted of six questions where the participant was asked to rank the five videos they watched from Best (top) to Worst (bottom). These questions, seen in Figure 6.32, were a simplified / reduced number of questions, but still encompassed the content of the 25 questions from the between subjects study. Referring to the earlier Table 6.6, we aligned the reduced questions as specified in Table 6.11.

The goal of this study was to determine which videos were more realistic or similar to a human-performed scene from a viewer's perspective. Each group viewed all five of the videos and answered the questions in Figure 6.32 (within groups experiment). Both the videos and the questions were presented in randomized order to the users after viewing the video.

The survey was posted on Mechanical Turk (MTurk) with criteria to enforce participants were from the United States, to avoid cultural differences of opinion regarding

RQ-1 Rank the Character Positioning within the video performances from Best (top) to Worst (bottom). Ex: Were the characters too close together? Too far apart? Did the arrangement of the characters make sense?

RQ-2 Rank the Character Movements within the video performances from Best (top) to Worst (bottom). Ex: Did the movements appear to be in-sync with the script? Did the characters move at unusual times? Did they move too much? Too little?

RQ-3 Rank the use of the Stage's space within the video performances from Best (top) to Worst (bottom). Ex: Did the characters cover the whole stage? Only one small part of the stage? Did the use of the space make sense with respect to the scene?

RQ-4 Rank the overall character visibility within the video performances from Best (top) to Worst (bottom). Ex: Were characters frequently blocking your view to another character? Were all characters visible throughout the entire scene?

RQ-5 Rank the pace of the scene within the video performances from Best (top) to Worst (bottom). Ex: Did it move too slow? Did it move too fast? Did the scene progress in-line with expectations for the script?

RQ-6 Rank the Overall video performances from Best (top) to Worst (bottom). Ex: Considering the entire scene, which one(s) were more pleasing or believable to you?

Figure 6.32: Spatio-Temporal Simplified Rank Questions (RQ) in the Within-Participants Survey

Table 6.11: Reduction of Spatio-Temporal Questions

Spatial Component Covered from Table 6.6	New Question
Engaged Listening	Character Position
Pace of Performance	Pace
Appropriate Movement and Timing	Character Movement
Consistent Movement	Character Movement
Space Usage	Stage Space
Variety and Closeness	Stage Space
Motivated Movement	Character Movement
Scene Mood	Overall Video
Character Emphasis	Character Position
Visible Characters	Character Visibility
Believable	Overall Video

space and performances. Additional controls were put in place to ensure participants viewed the entire video by including a timer on the video viewing page, and including an intermission screen of a particular color mid-way in the video. If the participant did not remain on the video page long enough to watch the videos, or if they did not know what color the intermission screens were, they were disqualified from participating.

The study included 49 participants who were presented (in random order) all five videos: one viewing the hand-mapped version of a real production of *Hamlet* (Baseline); one viewing a random movement version of the same scene (Random); one viewing our technique with only the natural language processor component (NLP); one viewing our technique with only the natural language processor and rules engine (Rules); and one viewing our technique with all the components—natural language processor, rules engine, and force-directed graphs (FDG). 256 people attempted the study, but were unable to complete the study due to the controls in place to ensure proper participation, such as time spent watching all of the videos, and identifying the intermission screen colors correctly for each video, which can be seen in Table 6.12. They then ranked the videos for the six different dimensions, shown in Figure 6.32. Figure 6.33 shows the breakdown of the participants by the different demographics.

Table 6.12: Multiple Component-Based Within Subjects Videos

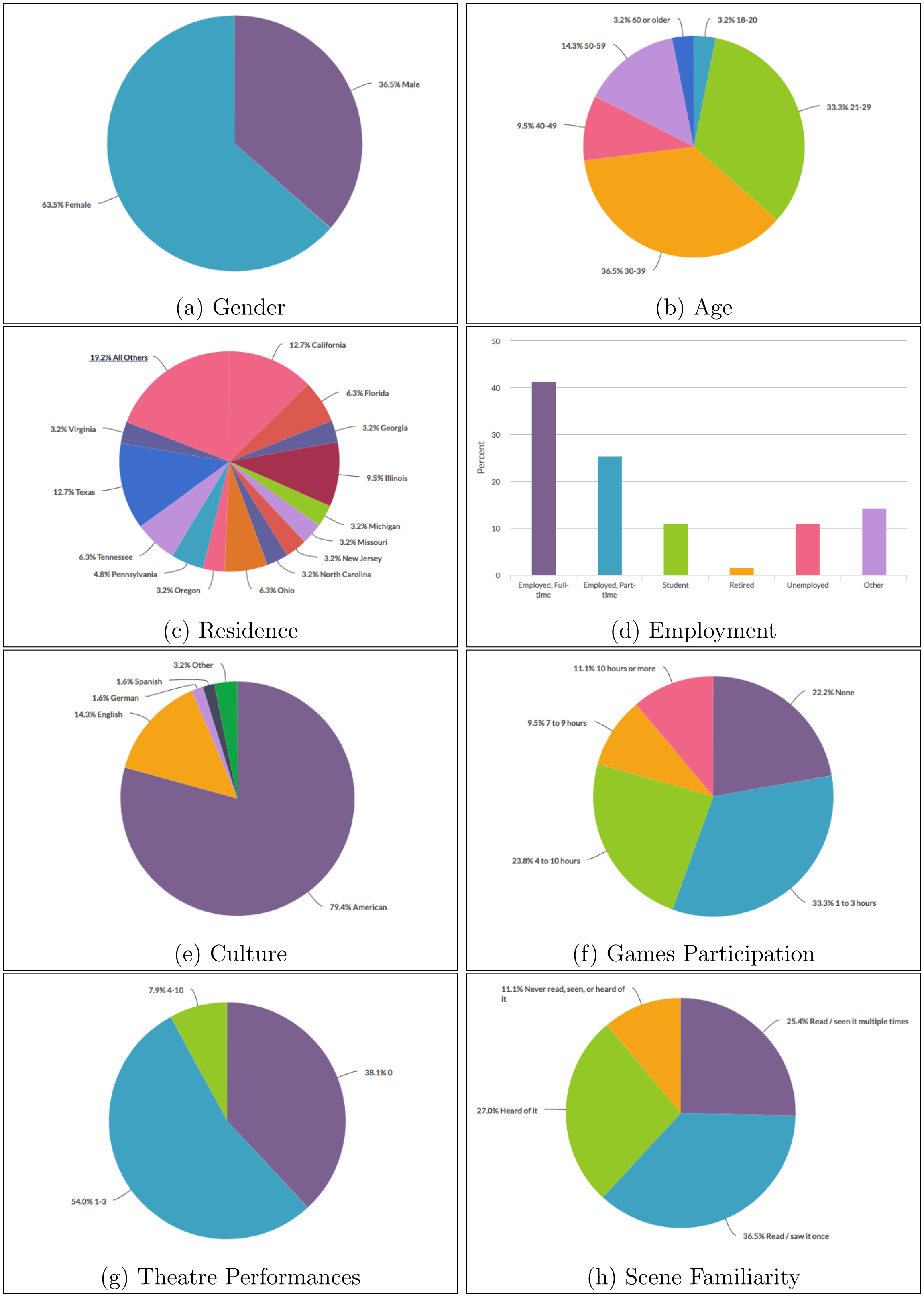
Video	Video URL
Baseline	https://www.youtube.com/embed/fY77-8VjSFY
Random	https://www.youtube.com/embed/Xs0TgXA8HtM
NLP	https://www.youtube.com/embed/Vjgf12niRRY
Rules	https://www.youtube.com/embed/QFO9D_CNcLk
FDG	https://www.youtube.com/embed/HHWc-HkDsu4

As you can see, it represents a reasonable sampling of the population.

This experiment provided an estimated power to detect a medium effect (f=0.25) of >0.99. Since the power of this experiment is relatively high, the chance of committing a Type II error is extremely low. We performed a Friedman test that showed there was a statistically significant difference in rankings for only two questions, "Character Visibility" $\chi^2 = 0.638$ $p = 0.049$ and "Overall" $\chi^2 = 15.171$ $p = 0.004$. However, running the Friedman test without the Random video included, neither of these questions showed any significant differences.

After running post-hoc Wilcoxon tests, we found that there were no statistically significant differences between the Baseline, FDG, Rules, and NLP videos. Now that we have a frame of reference for all the rankings, we have more consistent results. This confirms our expectations that these techniques provide an indistinguishably "good" performance from a viewer's perspective.

We then looked at how the Random video ranked versus the other videos and found for each question the statistically significant differences in rankings that can be seen in Table 6.13. This shows an overall trend of all the videos being better than the Random video, which again was expected.



(a) Gender

(b) Age

(c) Residence

(d) Employment

(e) Culture

(f) Games Participation

(g) Theatre Performances

(h) Scene Familiarity

Figure 6.33: Demographic Breakdown of Participants for the *Hamlet* Within Subjects Study—All Modes

Table 6.13: Within Subjects Significant Differences—Pairwise Comparisons

Question	Z	p	Pairwise Ranking (Ranked Higher > Ranked Lower)
Character Visibility	-2.125	0.034	Rules > Random
Overall Video	-2.165	0.030	FDG > Random
	-3.120	0.002	Baseline > Random
	-3.036	0.002	Rules > Random

6.5 Summary

In this chapter, we reviewed the evaluations performed to validate our techniques of natural language processing, rules, and force-directed graphs. We showed that our engine reduces the authorial burden for the scene designer, and qualitatively provide an equally "good" performance as a real human performance. We showed that we were able to match a real performance at 89% for position and 53% for gaze using our NLP+Rules engine. We explored the capabilities of force-directed graphs to better arrange our characters on the stage than our rules alone. Overall, we saw that our techniques met or surpassed a human's perception of a quality performance, and was able to mimic a famous Broadway production from 1964.

CHAPTER 7: GENERALIZATION

Now that we have reviewed each component of our engine with one play-script, we need to evaluate how well these techniques can perform for other play-scripts. Here, we identify the different spatio-temporal performance dimensions that vary in a performance. We leverage these to identify a covering set of play-scripts to determine our technique's generalizability. We once again evaluate our engine against the human performance of the same play-script for both quantitative and qualitative analysis. We find that in most cases, viewers see our engine as qualitatively as good as the human performance version of the same scene.

7.1 Spatio-temporal Performance Dimensions

When looking to prove the generalization of the techniques described in Chapter 4: METHODOLOGY, we have to define the spatio-temporal dimensions that define the unique types of performances. When reviewing the spatial and temporal aspects of a play, we identified the following dimensions and categories:

- **Speed** Slow, Medium, Fast
- **Number of Characters** One, Two to Four, Five or More
- **Space** Defined Box, Undefined, Defined Partial
- **Audience** Proscenium, Thrust, Round
- **Dynamics** Formal, Informal

The speed of the performance indicates how quickly the characters progress through a scene. For example, *Krapp's Last Tape* is a slow-moving performance, whereas

Noises Off is a fairly fast-paced performance. As we look at generalizing these techniques, we want to ensure they work for different paces. This is leveraged as a parameter within our engine that could reduce or expand the amount of time between actions, and for movement, within the play-script.

All performances include a number of actors that perform the scene—anywhere from one to up to twelve or more. Here, we divided these into three categories to segregate the complexities that arise due to how many actors are on a stage at a time. With just one character on the stage, we see more monologues and less interaction with others. When we have a large number of characters on the stage, there is more complexity in the interactions and options available to the actors. However, most scenes tend to have just a few characters, which we've differentiated with the two to four group.

Space is key to our technique, yet even though it can have many variations of a fully defined box area to a fully undefined area, it can often be reduced down to a single defined area that is currently being enacted upon. So despite it being an important dimension, it does not affect the distinct combinations for generalizing our technique. Defined partial space is when the stage is split into sections that will be used for certain scenes, while the rest of the stage goes unused

The audience can vary due to theatre configurations, or even within games. Usually there is one location that the audience resides—proscenium theatre setup or single player games. There are also scenarios where there may be two, three (thrust theatre setup), or four (in the round theatre setup) locations where the audience resides. Today's theatre is most commonly performed in a proscenium configuration, which limited our ability to explore the generalization to other arrangements due to insufficient video samples available to compare to a human-performed baseline.

Last, but not least, we have different aspects of the intimacy of the scene. Some are more formal and lend themselves to a more distant positioning of the characters, while others might be more informal, with closer associations during a performance. Dynamics drives how closely the characters may be to each other when conversing. This is represented by a parameter in our engine that controls the forces for spacing between characters, which can adjust how familiarly the characters arrange themselves in the scene. Also considered were aspects of the levels of the scene, but it was determined this was not required since it is always part of the defined space and it can always be reduced to a single continuous space in virtual environments.

7.2 Generalization Coverage

With the above defined dimensions, we have a $3 \cdot 3 \cdot 3 \cdot 3 \cdot 2 = 162$ dimensional space to cover to prove generalization. Because all the options for space can be reduced down to a single, currently used, defined space, we can reduce that dimension from three to one, and thereby reduce our dimensionality to $3 \cdot 3 \cdot 1 \cdot 3 \cdot 2 = 54$. Also, due to our limited availability of samples with other audience configurations, we are forced to limit our generalization coverage to just the single audience scenario. This leaves us with a $3 \cdot 3 \cdot 1 \cdot 1 \cdot 2 = 18$ dimensional space for proving generalization across a single audience category for plays.

With so many permutations to cover, we look for ways to reduce testing iterations yet still reach a reasonable coverage of our domain. It has been found that whenever an application has roughly five or more configurable attributes, a covering array is likely to make testing more efficient. Because the number of t-way tests is proportional to $v^t \log n$, for n parameters with v values each, unless configurable attributes have more than eight or ten possible values each, the number of tests generated will probably be reasonable [70]. Pairwise testing has come to be accepted as the common approach to combinatorial testing because it is computationally tractable and reasonably effective [49].

We leveraged the Advanced Combinatorial Testing System (ACTS) tool to help identify potential dimension combinations required to obtain 100% pairwise coverage.

\# Degree of interaction coverage: 2
\# Number of parameters: 4
\# Maximum number of values per parameter: 3
\# Number of configurations: 9

Num Chars	Speed	Dynamics	Audience
One	Slow	Informal	Proscenium
One	Moderate	Formal	Proscenium
One	Fast	Informal	Proscenium
Two to Four	Slow	Formal	Proscenium
Two to Four	Moderate	Informal	*
Two to Four	Fast	Formal	*
Five More	Slow	Informal	Proscenium
Five More	Moderate	Formal	*
Five More	Fast	*	*

Figure 7.1: ACTS Tool Output of Pairwise Coverage with Audience Constraint

Since some combinations cannot be tested because they don't exist for the systems under test, such as our audience dimension, we specified constraints, which told the tool not to include specified combinations in the generated test configurations. The covering array tool then generated a set of test configurations that does not include the invalid combinations, but does cover all those that are essential for pairwise testing [8]. The ACTS tool was used to determine coverage of our scenarios for generalization. It provided nine configurations to obtain 100% pairwise coverage and 100% dimensional coverage. Figure 7.1 show the outputs from the tool.

We were able to find seven of those combinations as video-recorded scenes (and hence a usable baseline to compare to), as seen in Table 7.1. As you can see, this only covers seven of the nine combinations required for 100% pairwise coverage, yet does provide 100% coverage for each dimension independently.

Unfortunately, we were unable to find a usable recorded scene that would fit the scenarios defined in Table 7.2. These missing scenes cover more of the edges of our spatial domain with respect to the number of characters on the scene. We wanted

Table 7.1: Covered Pairwise Combinations

Play Title	Play Author	Num Chars	Speed	Dynamics
The Importance of Being Earnest	Oscar Wilde	5+	Moderate	Formal
Death of a Salesman	Arthur Miller	5+	Moderate	Informal
Krapp's Last Tape	Samuel Beckett	1	Slow	Informal
Noises Off	Michael Frayn	2-4	Fast	Informal
Tartuffe	Moliere	2-4	Moderate	Informal
Hamlet	Shakespeare	2-4	Slow	Formal
The Cherry Orchard	Anton Chekhov	2-4	Slow	Informal

Table 7.2: Missing Pairwise Combinations for 100% Coverage

Num Chars	Speed	Dynamics
Five or More	slow	formal
Five or More	fast	formal
One	moderate	formal
One	fast	formal

to better understand what kind of coverage this would provide us, so we looked to the National Institute of Standards and Technology (NIST) Combinatorial Coverage Measurement (CCM) tool for answers [48]. The CCM tool provides insight into what coverage you have, given a specific set of dimension combinations. It showed that we obtained 71.42% pairwise coverage, and 100% dimensional coverage with our chosen play-scripts.

7.3 Generalization Experimentation

With a total of 162 possible combinations of play-scripts required to fully cover our spatio-temporal dimensions, we reduced our sampling to only seven scenes, with one having already been completed (*Hamlet*). We are comparing a hand-mapped baseline version of each recorded performance to only one version of our techniques—the complete natural language processor, rules, and force-directed graph (NLP+Rules+FDG) engine. We evaluated each of these remaining six play-scripts both quantitatively and qualitatively to determine the effectiveness of our NLP+Rules+FDG techniques for positioning the characters, and prove generalization.

7.3.1 Quantitative Analysis

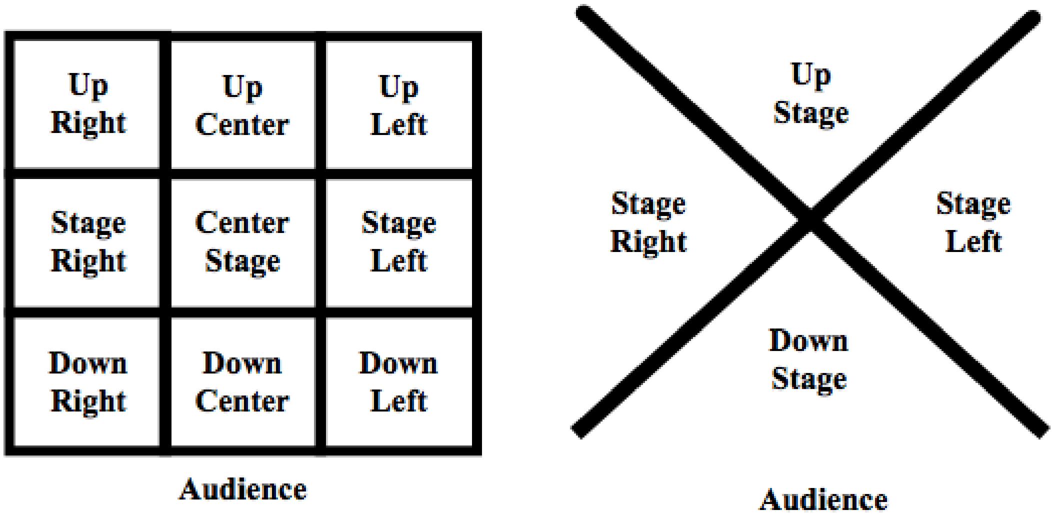
For each play-script, we took the character traces from both our baseline (hand-coded Behavior Markup Language, BML, based on the actual play performance) and our NLP+Rules+FDG engine, and compared them. We wanted our new technique to have similar positioning as the actual performance, but not penalize for being "close enough." Therefore, we divided the stage into nine onstage areas, plus a tenth offstage location, as seen in Figure 7.2a. We also did the same comparison for the gaze direction to determine which of the four directions the character was gazing throughout the scene, as seen in Figure 7.2b.

Adjustments were made to the original codebase to incorporate an initialization file that would identify all the pawns, marks, characters, character prioritization, and colorings. Some additional synonyms were added to align with the language used within the play-script to align the natural language processor (NLP) component's BML output properly.

It is important to notice that most of the performances included movements that were not annotated within the play-script utilized by our techniques. Also, due to the difficulty in handmapping gazes in the scenes, some gaze information in the baseline videos may be missing. Lastly, the forces add a layer of arrangement that may shift characters slightly outside their stage box, but still be close, when adjusting the character positions, which may impact our positional matching.

7.3.1.1 *Tartuffe* Quantitative Analysis

With the *Tartuffe* play-script, we had two characters (Elmire and Orgon), and two pawns in the scene. This particular scene required us to incorporate a feature for moving around an object. The prioritization of the characters for this scene were defined to be:



(a) Stage Positions

(b) Stage Gaze Directions

Figure 7.2: Stage Area Breakdown for Position and Gaze Comparisons

Table 7.3: *Tartuffe* Character Traces Match for Baseline vs. FDG

Character Name	Gaze Match	Position Match
ELMIRE	52.38%	50.79%
ORGON	24.34%	12.70%
Overall	38.36%	31.75%

$$Elmire > Organ$$

We took the logged position and gaze traces for each of the characters, for both the baseline scene and our FDG scene. We normalized the time for both scenes, and mapped the positions and gazes to our grid locations, as before. These grid locations and directions were then compared for a match or not, incorporating the "close enough" criteria by leveraging the generalized locations and directions.

We found that, with the forces, we were able to position the characters correctly on average 31.75% of the time, and their gazes 38.36%, as can be seen in Table 7.3. Detailed character trace information can be seen in Figures 7.3, 7.4, 7.5, and 7.6. Some of these discrepancies appear to be related to movements that were not annotated in the play-script, such as Elmire pacing to the far corners of the stage and back a few times.

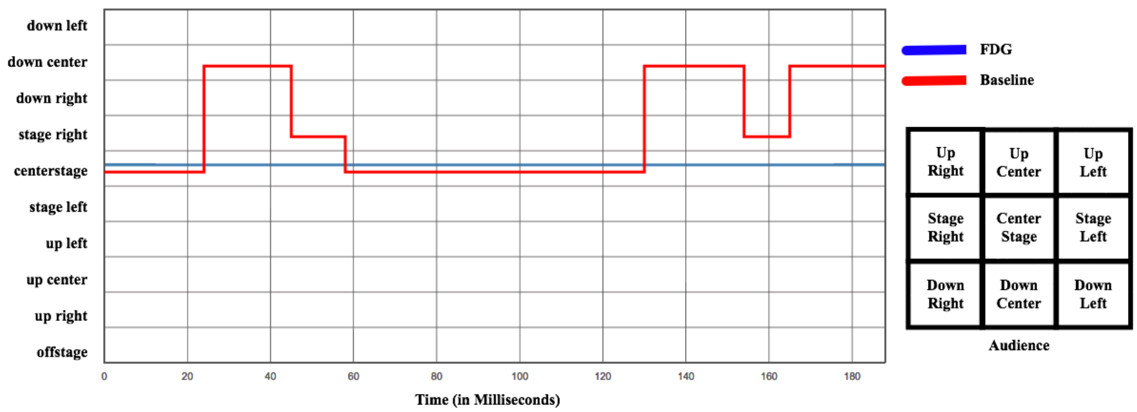


Figure 7.3: Character Position Traces for Elmire in *Tartuffe*

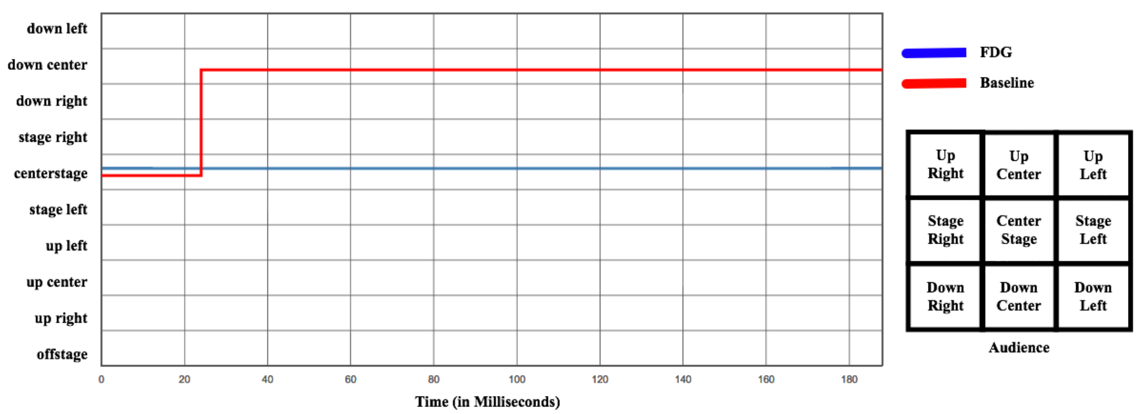


Figure 7.4: Character Position Traces for Orgon in *Tartuffe*

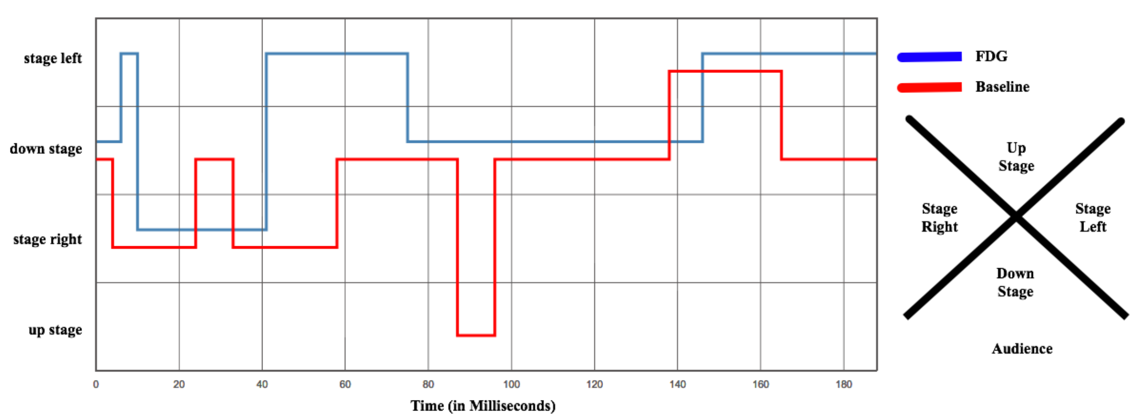


Figure 7.5: Character Gaze Traces for Elmire in *Tartuffe*

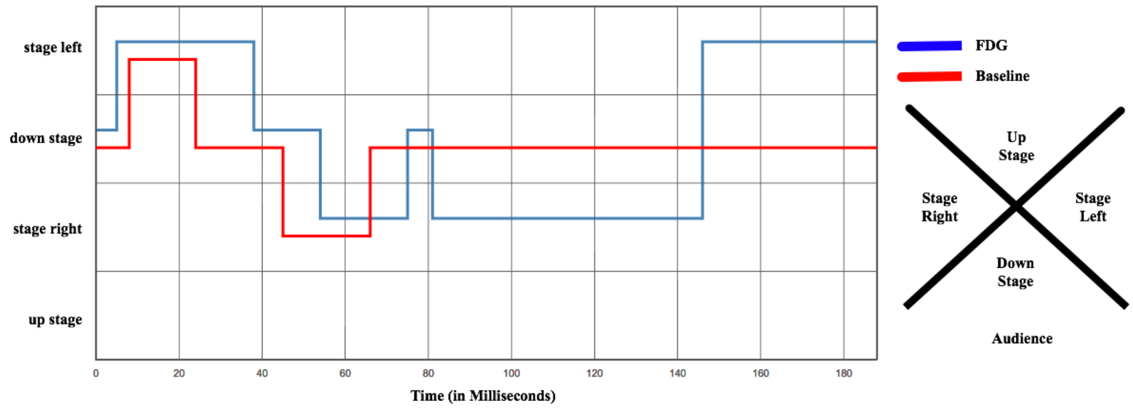


Figure 7.6: Character Gaze Traces for Orgon in *Tartuffe*

7.3.1.2 *Death of a Salesman* Quantitative Analysis

With the *Death of a Salesman* play-script, we had five characters (Linda, Biff, Charley, Happy, and Bernard), and no pawns in the scene. The prioritization of the characters for this scene were defined to be:

$$Linda > Biff > Charley > Happy > Bernard$$

We took the logged position and gaze traces for each of the characters, for both the baseline scene and our FDG scene. We normalized the time for both scenes, and mapped the positions and gazes to our grid locations, as before. These grid locations and directions were then compared for a match or not, incorporating the "close enough" criteria by leveraging the generalized locations and directions.

We found that, with the forces, we were able to position the characters correctly on average 54.76% of the time, and their gazes 41.57%, as can be seen in Table 7.4. Detailed character trace information can be seen in Figures 7.7, 7.8, 7.9, 7.10, 7.11, 7.12, 7.13, 7.14, 7.15, and 7.16. Some of these discrepancies appear to be related to general conversational spacing and just being slightly outside the stage position box from the baseline. This can be seen best with Bernard and Linda, where they are down center in the Baseline video, but centerstage in the FDG video. This is due to our arrangement into a semi-circle by the forces, which shifted their location slightly.

Table 7.4: *Death of a Salesman* Character Traces Match for Baseline vs. FDG

Character Name	Gaze Match	Position Match
BERNARD	35.54%	15.06%
BIFF	54.22%	97.29%
CHARLEY	33.43%	90.36%
HAPPY	62.05%	71.08%
LINDA	22.59%	0.00%
Overall	41.57%	54.76%

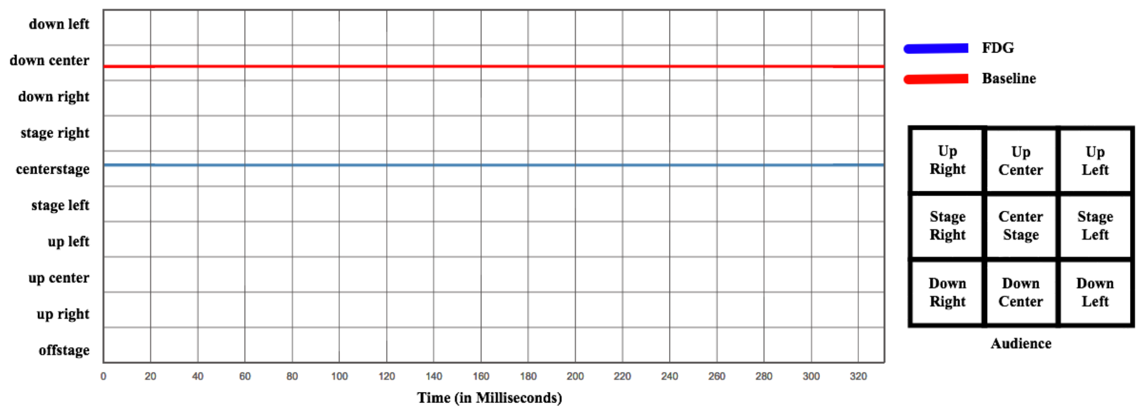


Figure 7.7: Character Position Traces for Linda in *Death of a Salesman*

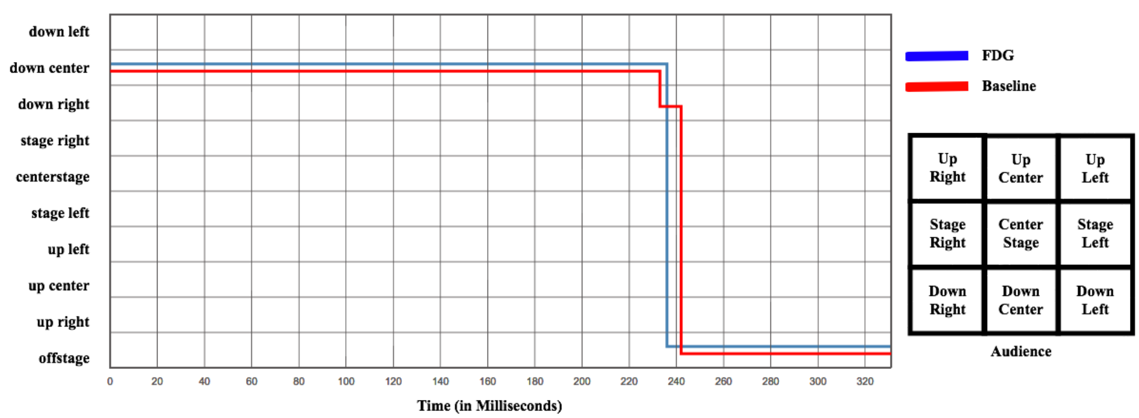


Figure 7.8: Character Position Traces for Biff in *Death of a Salesman*

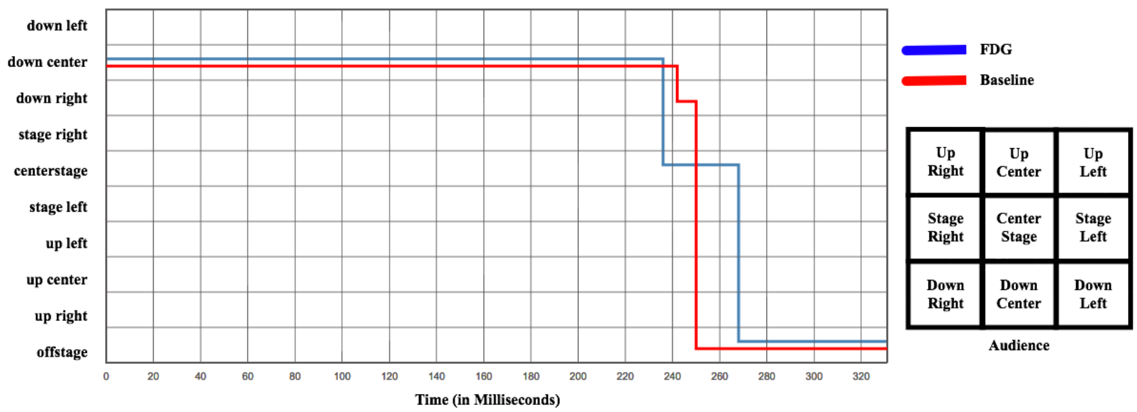


Figure 7.9: Character Position Traces for Charley in *Death of a Salesman*

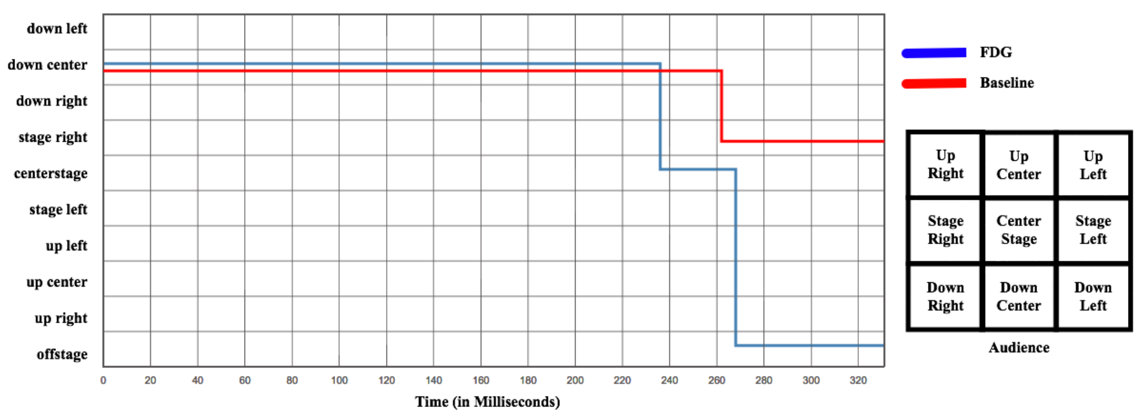


Figure 7.10: Character Position Traces for Happy in *Death of a Salesman*

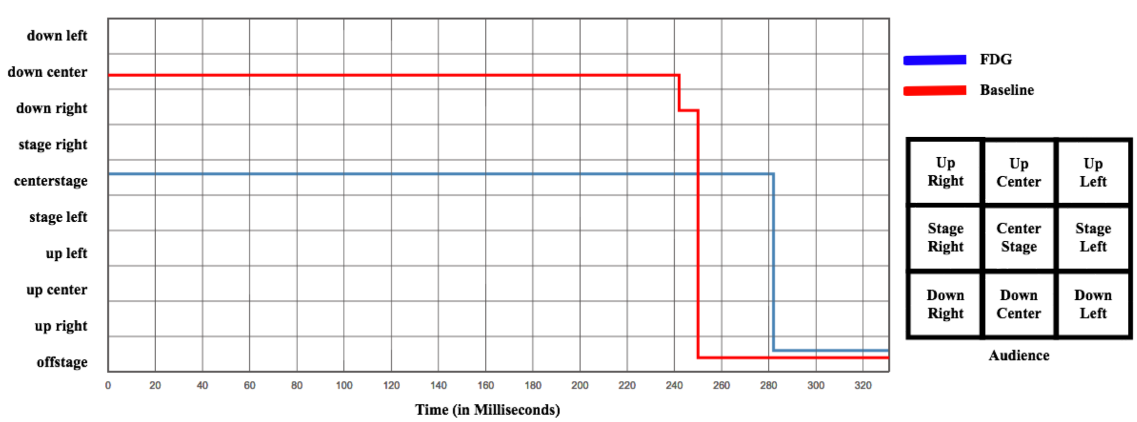


Figure 7.11: Character Position Traces for Bernard in *Death of a Salesman*

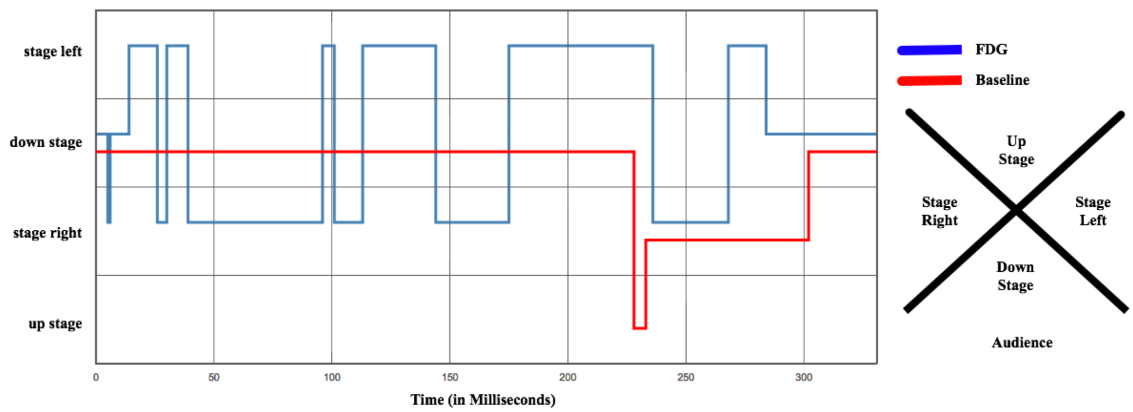


Figure 7.12: Character Gaze Traces for Linda in *Death of a Salesman*

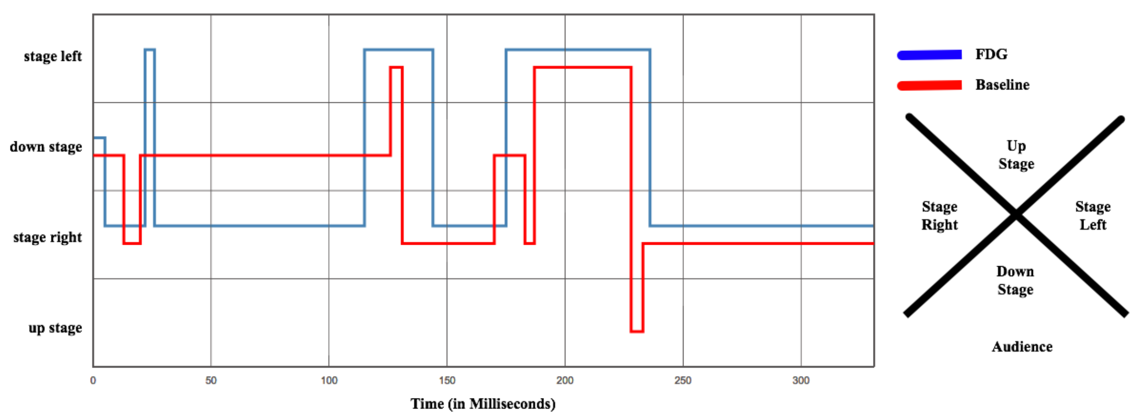


Figure 7.13: Character Gaze Traces for Biff in *Death of a Salesman*

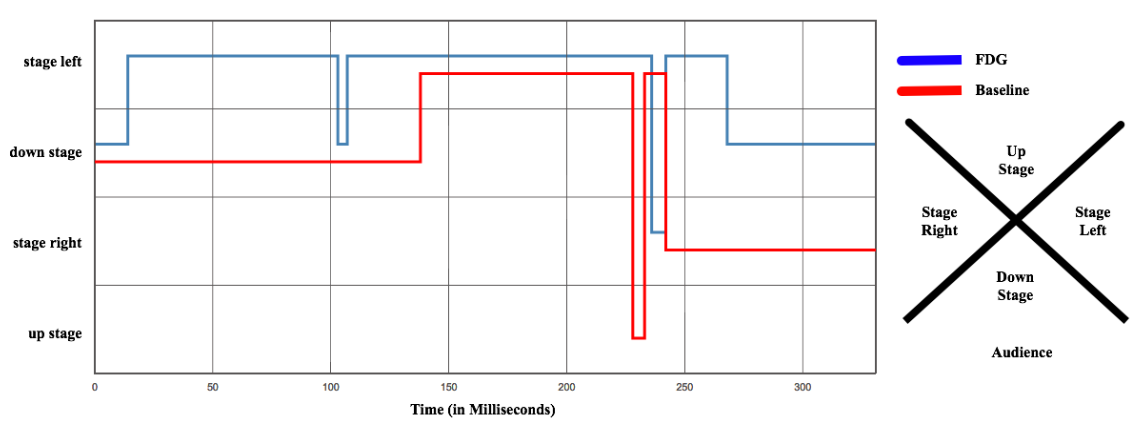


Figure 7.14: Character Gaze Traces for Charley in *Death of a Salesman*

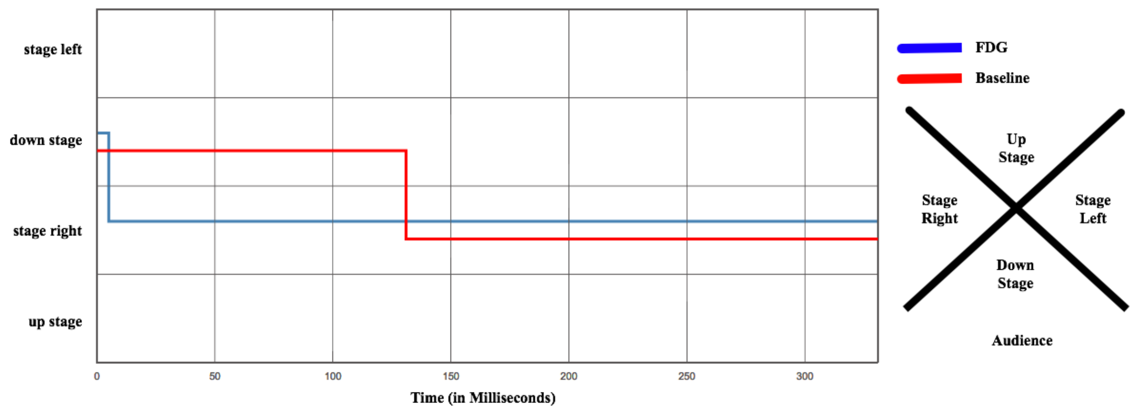


Figure 7.15: Character Gaze Traces for Happy in *Death of a Salesman*

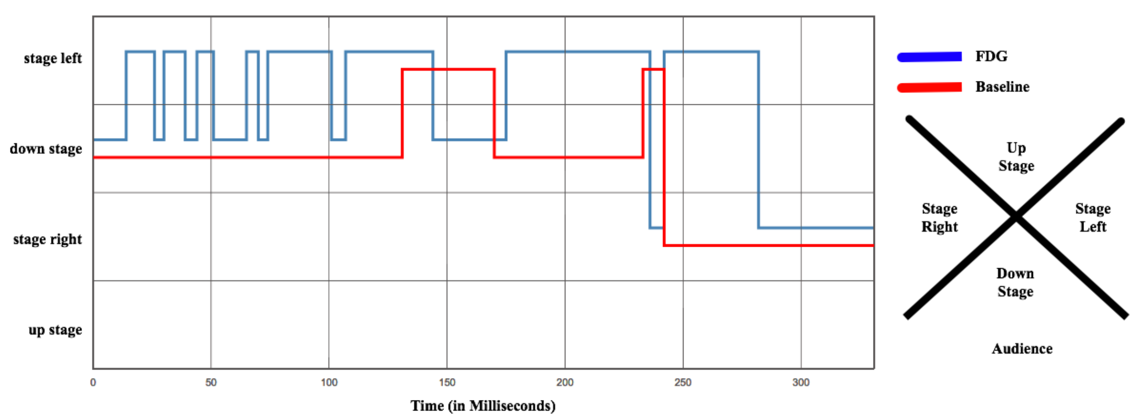


Figure 7.16: Character Gaze Traces for Bernard in *Death of a Salesman*

7.3.1.3 *Noises Off* Quantitative Analysis

With the *Noises Off* play-script, we had four characters (Garry, Lloyd, Dotty, and Brooke), and one pawn in the scene. The prioritization of the characters for this scene were defined to be:

$$Garry > Lloyd > Dotty > Brooke$$

We took the logged position and gaze traces for each of the characters, for both the baseline scene and our FDG scene. We normalized the time for both scenes, and mapped the positions and gazes to our grid locations, as before. These grid locations and directions were then compared for a match or not, incorporating the "close enough" criteria by leveraging the generalized locations and directions.

We found that, with the forces, we were able to position the characters correctly on average 62.76% of the time, and their gazes 24.55%, as can be seen in Table 7.5. Detailed character trace information can be seen in Figures 7.17, 7.18, 7.19, 7.20, 7.21, 7.22, 7.23, and 7.24. Some of these discrepancies appear to be related to slight shifts in position location, such as Brooke being down left in the Baseline video, but stage left in the FDG video. Also, you will see that some of the rules incorporated extra gaze directions, which were not captured in the Baseline video. One example is Dotty, where she oscillates from looking stage left versus stage right in the FDG video because she is looking at the current speaker the whole time. Many of these gaze changes were missed in the handmapping of the scene.

Table 7.5: *Noises Off* Character Traces Match for Baseline vs. FDG

Character Name	Gaze Match	Position Match
BROOKE	7.76%	5.97%
DOTTY	34.63%	97.31%
GARRY	31.04%	84.18%
LLOYD	24.78%	63.58%
Overall	24.55%	62.76%

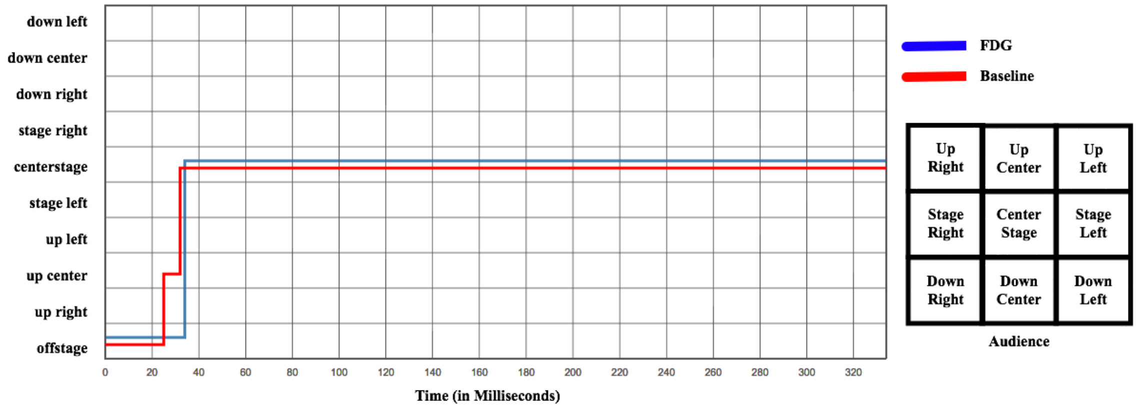


Figure 7.19: Character Position Traces for Dotty in *Noises Off*

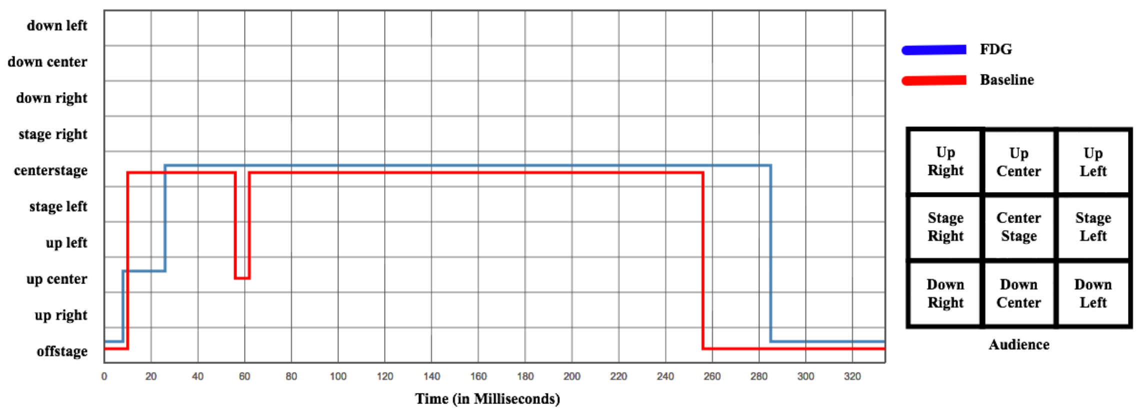


Figure 7.17: Character Position Traces for Garry in *Noises Off*

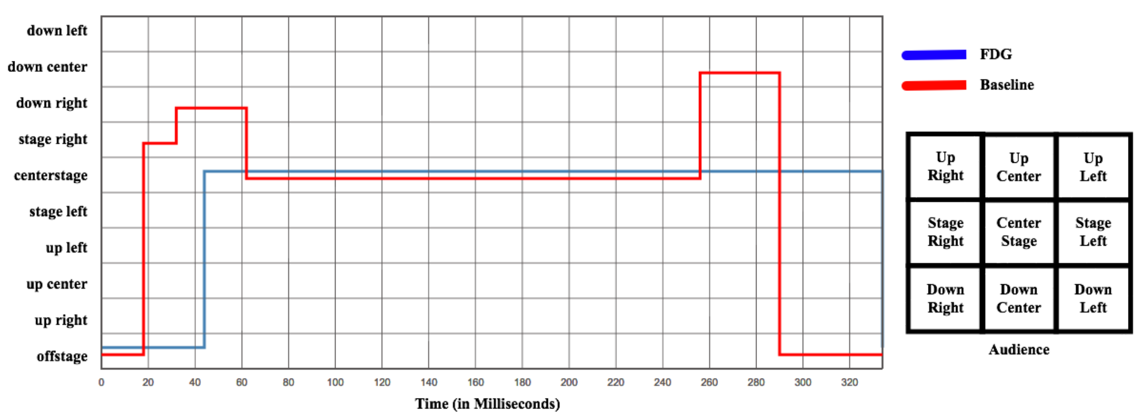


Figure 7.18: Character Position Traces for Lloyd in *Noises Off*

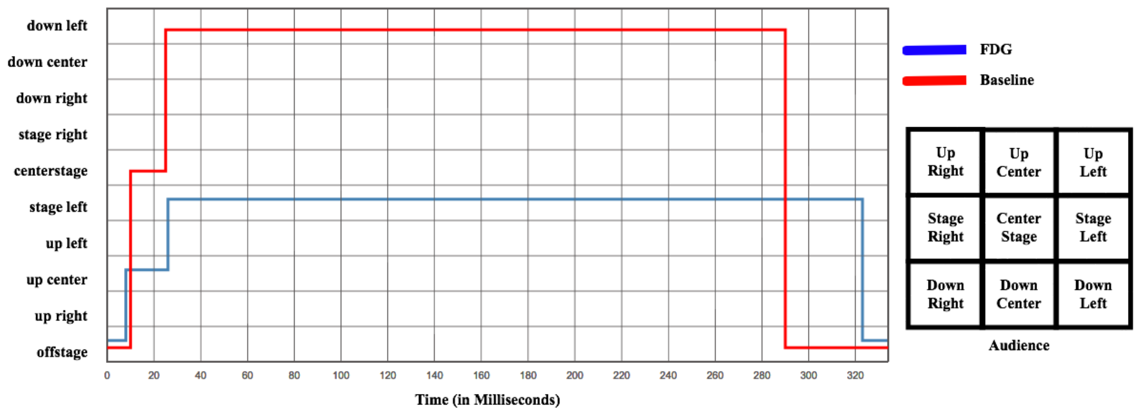


Figure 7.20: Character Position Traces for Brooke in *Noises Off*

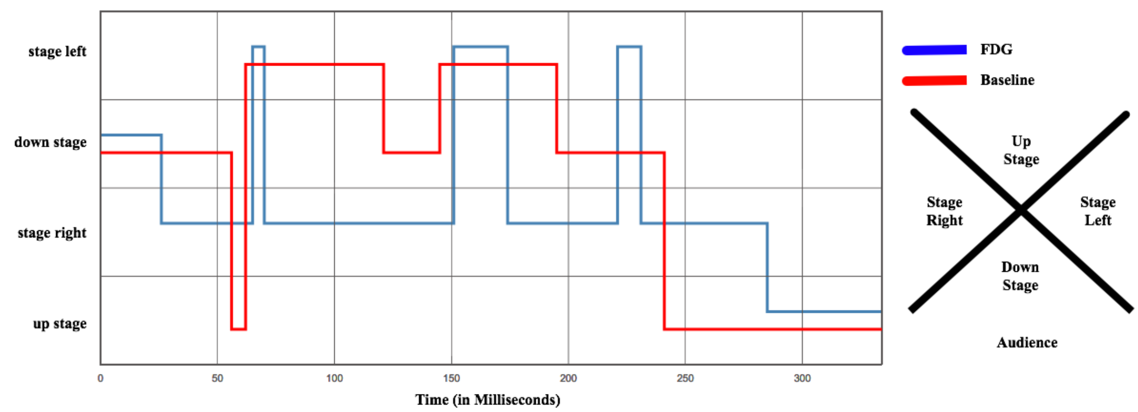


Figure 7.21: Character Gaze Traces for Garry in *Noises Off*

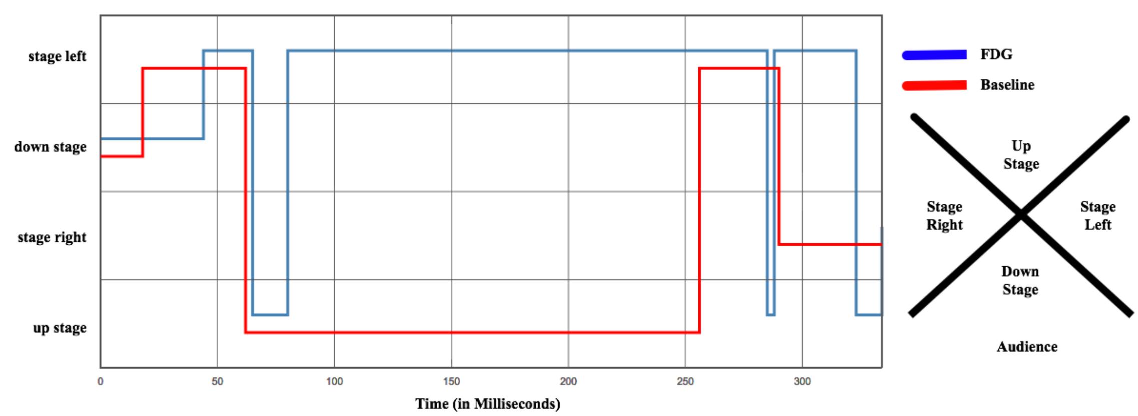


Figure 7.22: Character Gaze Traces for Lloyd in *Noises Off*

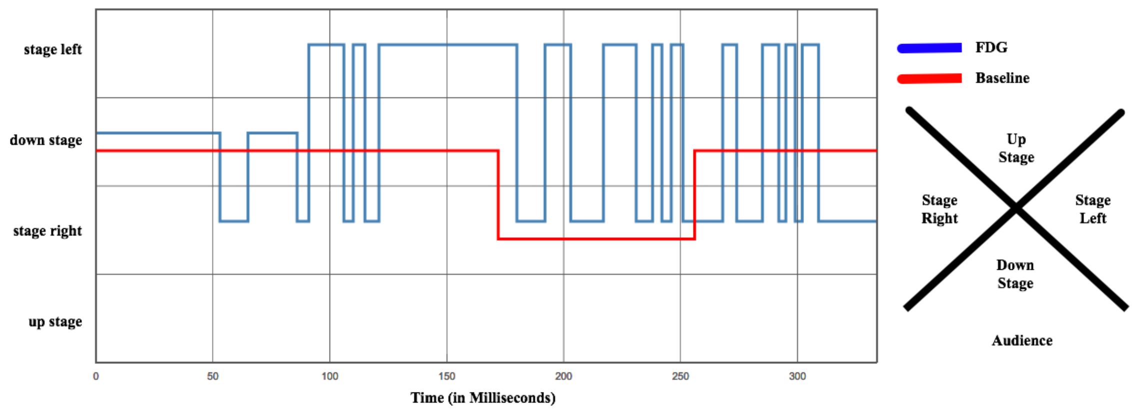


Figure 7.23: Character Gaze Traces for Dotty in *Noises Off*

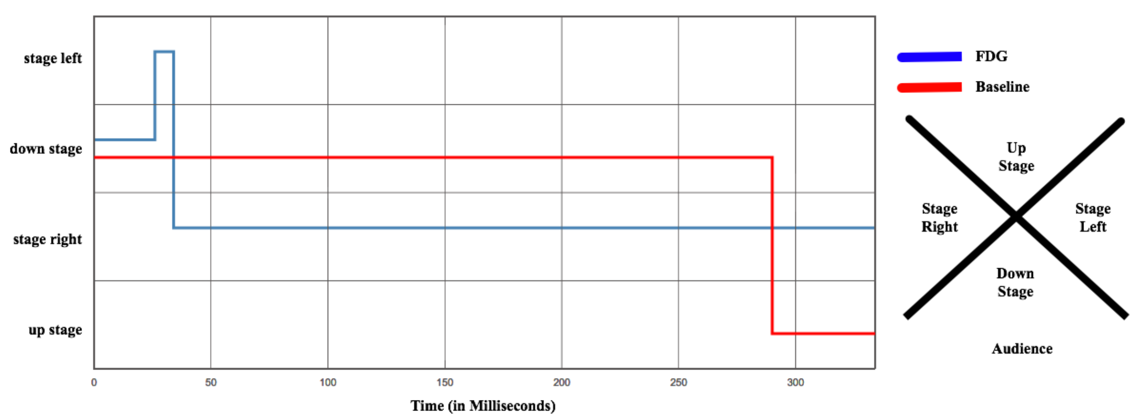


Figure 7.24: Character Gaze Traces for Brooke in *Noises Off*

7.3.1.4 *Krapp's Last Tape* Quantitative Analysis

With the *Krapp's Last Tape* play-script, we had one character (Krapp), and eight pawns in the scene. This particular scene required us to incorporate a feature for determining which object of a set of objects with the same name is being referred to (i.e., Box 1 versus Box 2). We determined this by which one was closest to the character. A similar feature was utilized in *Hamlet* to differentiate the two skulls in the "Graveyard" scene.

We took the logged position and gaze traces for each of the characters, for both the baseline scene and our FDG scene. We normalized the time for both scenes, and mapped the positions and gazes to our grid locations, as before. These grid locations and directions were then compared for a match or not, incorporating the "close enough" criteria by leveraging the generalized locations and directions.

We found that, with the forces, we were able to position the character correctly 100.00% of the time, and his gaze 43.62%, as can be seen in Table 7.6. Detailed character trace information can be seen in Figures 7.25 and 7.26.

Table 7.6: *Krapp's Last Tape* Character Traces Match for Baseline vs. FDG

Character Name	Gaze Match	Position Match
KRAPP	43.62%	100.00%
Overall	43.62%	100.00%

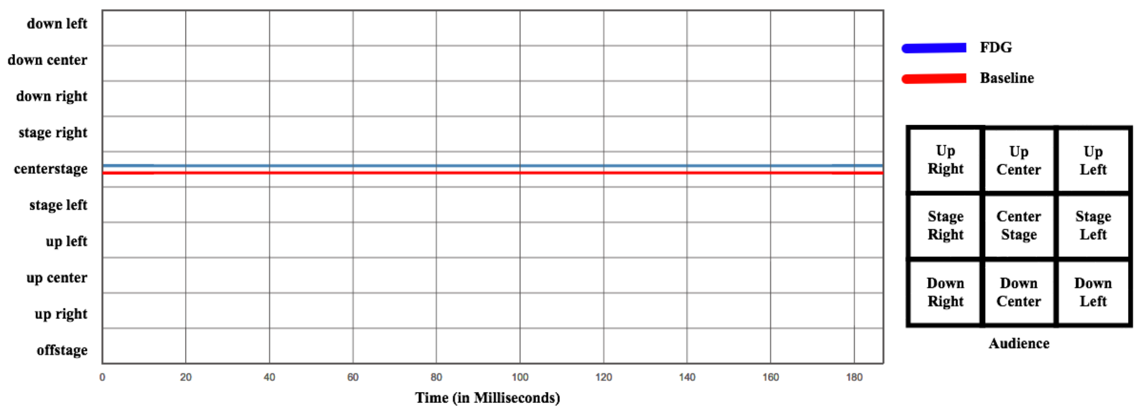


Figure 7.25: Character Position Traces for Krapp in *Krapp's Last Tape*

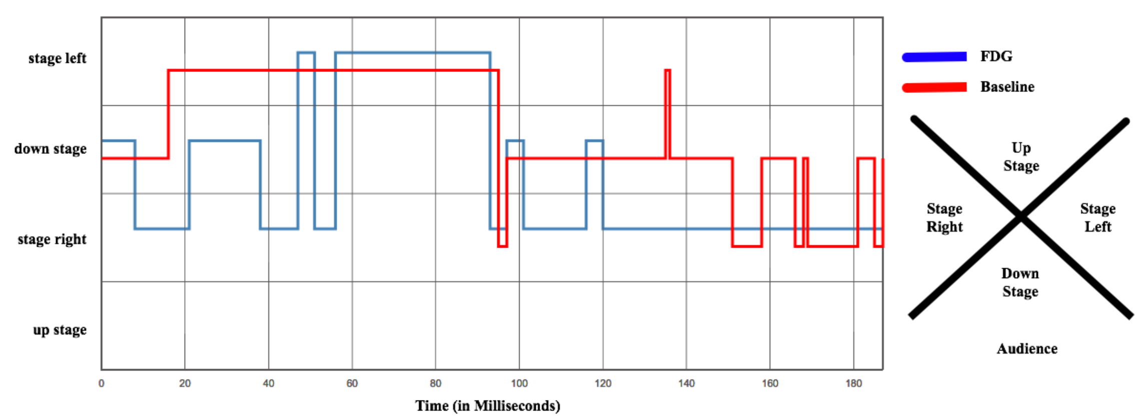


Figure 7.26: Character Gaze Traces for Krapp in *Krapp's Last Tape*

7.3.1.5 *The Cherry Orchard* Quantitative Analysis

With *The Cherry Orchard* play-script, we had six characters (Anya, Yasha, Epikhodov, Trofimov, Lopakhin, and Varya), and two pawns in the scene. The prioritization of the characters for this scene were defined to be:

$$Anya > Yasha > Epikhodov > Trofimov > Lopakhin > Varya$$

We took the logged position and gaze traces for each of the characters, for both the baseline scene and our FDG scene. We normalized the time for both scenes, and mapped the positions and gazes to our grid locations, as before. These grid locations and directions were then compared for a match or not, incorporating the "close enough" criteria by leveraging the generalized locations and directions.

We found that, with the forces, we were able to position the characters correctly on average 71.98% of the time, and their gazes 60.94%, as can be seen in Table 7.7. Detailed character trace information can be seen in Figures 7.27, 7.28, 7.29, 7.30, 7.31, 7.32, 7.33, 7.34, 7.35, 7.36, 7.37, and 7.38. Here again, we see a few slight shifts of location between the Baseline and FDG videos, such as being down right versus stageright with Anya, Trofimov, and Lopakhin.

Table 7.7: *The Cherry Orchard* Character Traces Match for Baseline vs. FDG

Character Name	Gaze Match	Position Match
ANYA	44.38%	23.13%
EPIKHODOV	54.38%	86.25%
LOPAKHIN	19.38%	68.13%
TROFIMOV	85.63%	70.00%
VARYA	84.38%	100.00%
YASHA	77.50%	84.38%
Overall	60.94%	71.98%

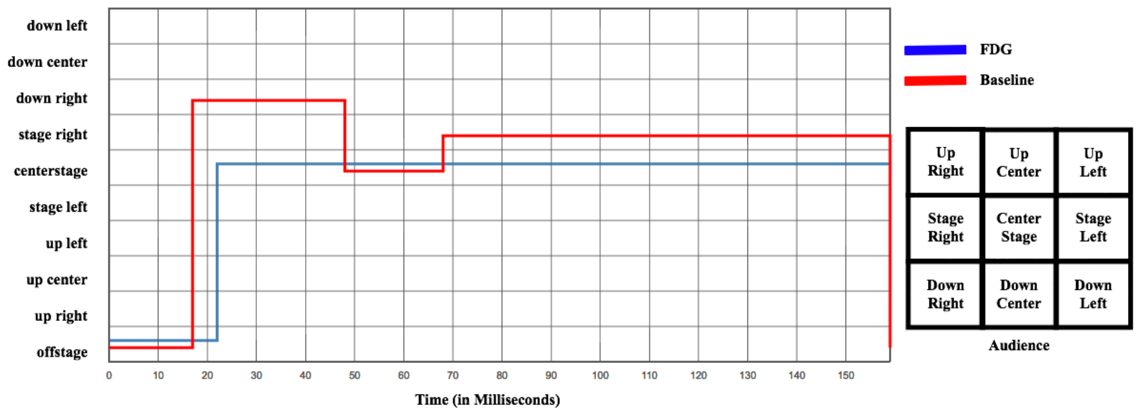


Figure 7.27: Character Position Traces for Anya in *The Cherry Orchard*

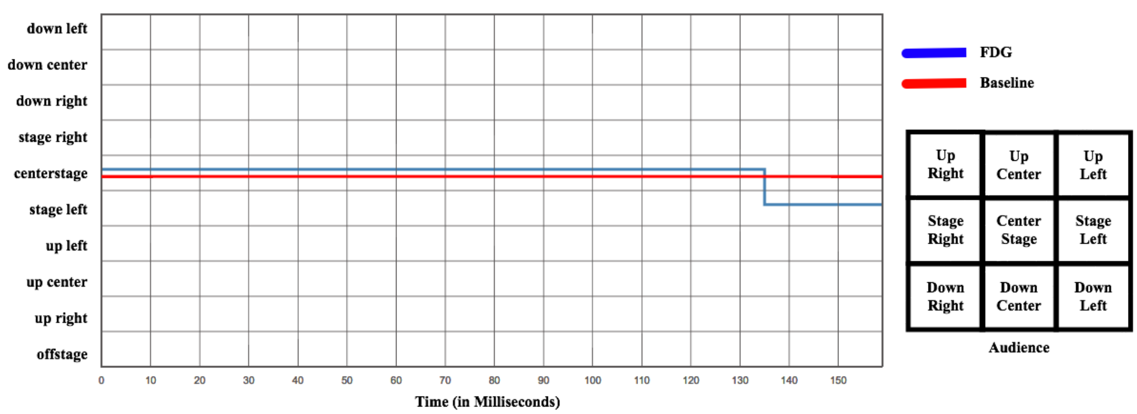


Figure 7.28: Character Position Traces for Yasha in *The Cherry Orchard*

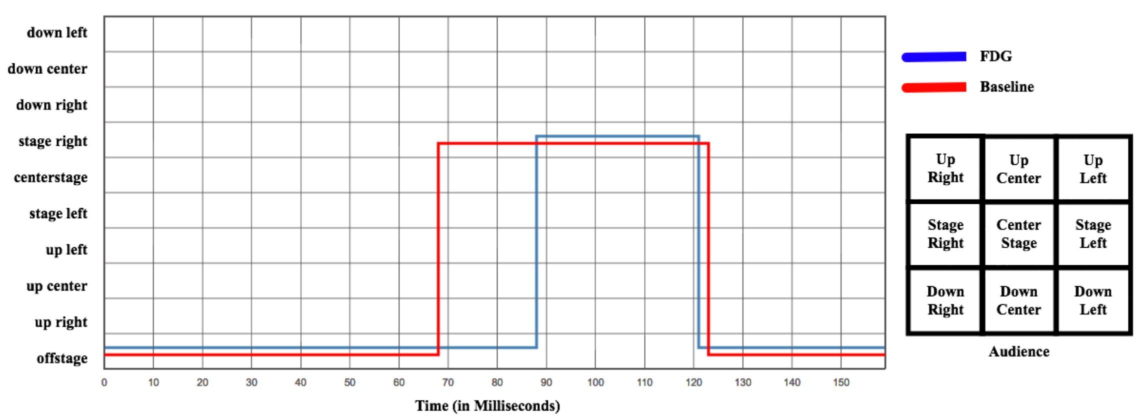


Figure 7.29: Character Position Traces for Epikhodov in *The Cherry Orchard*

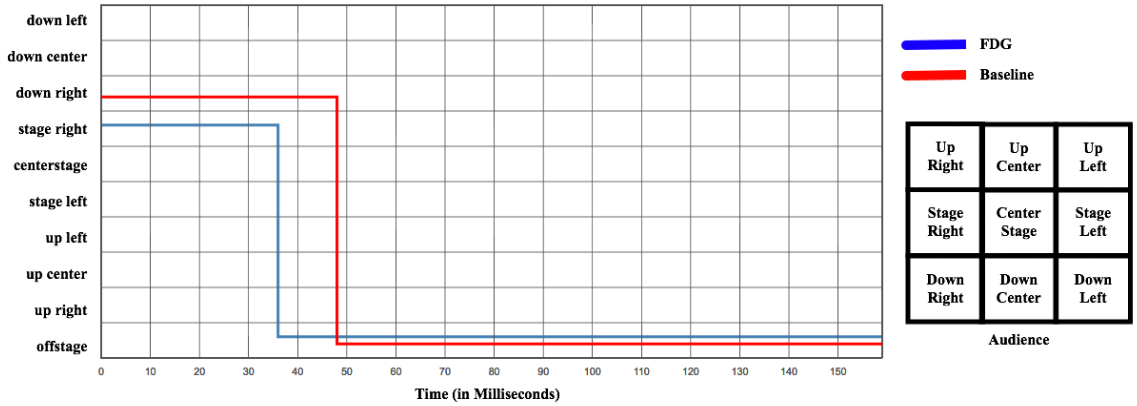


Figure 7.30: Character Position Traces for Trofimov in *The Cherry Orchard*

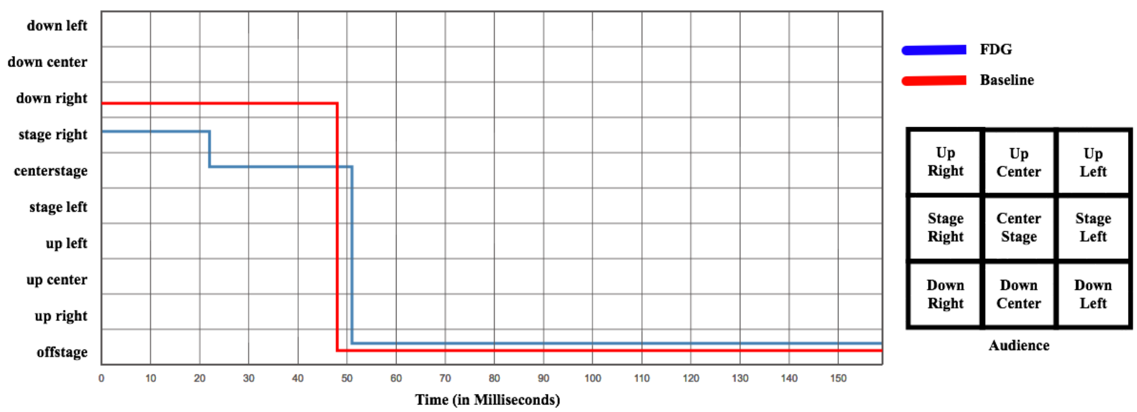


Figure 7.31: Character Position Traces for Lopakhin in *The Cherry Orchard*

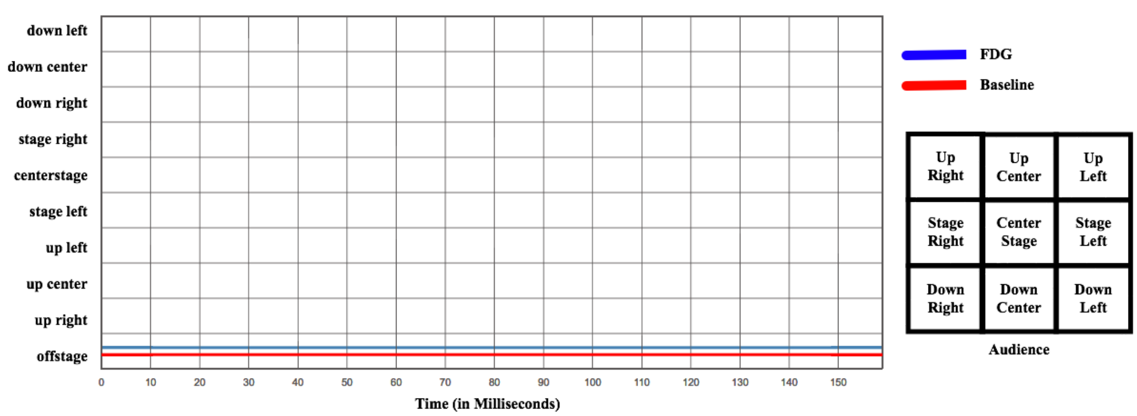


Figure 7.32: Character Position Traces for Varya in *The Cherry Orchard*

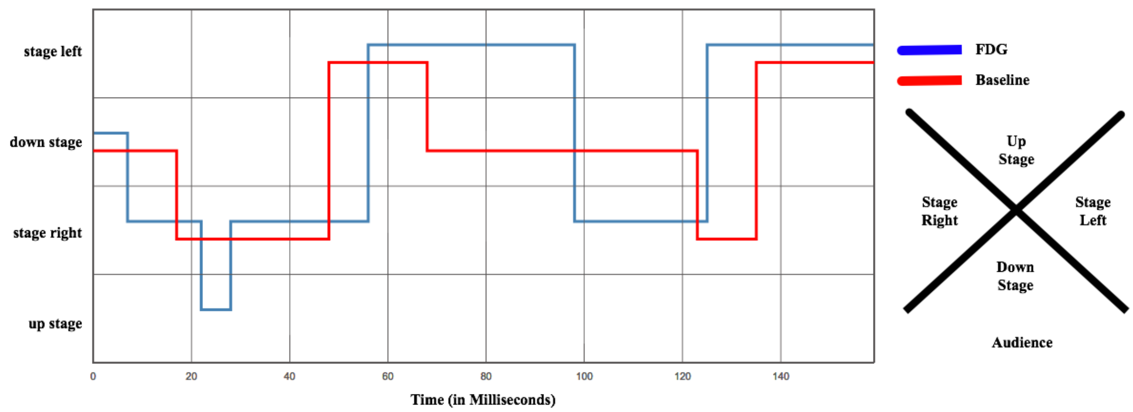


Figure 7.33: Character Gaze Traces for Anya in *The Cherry Orchard*

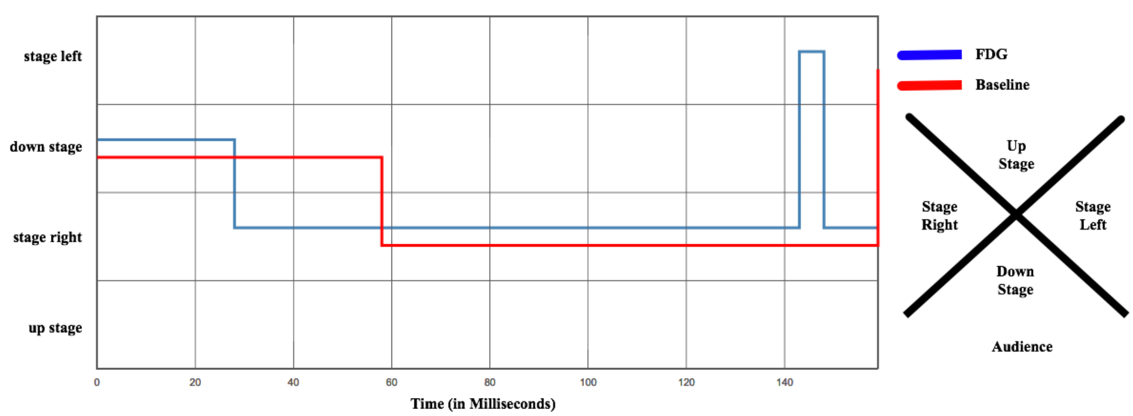


Figure 7.34: Character Gaze Traces for Yasha in *The Cherry Orchard*

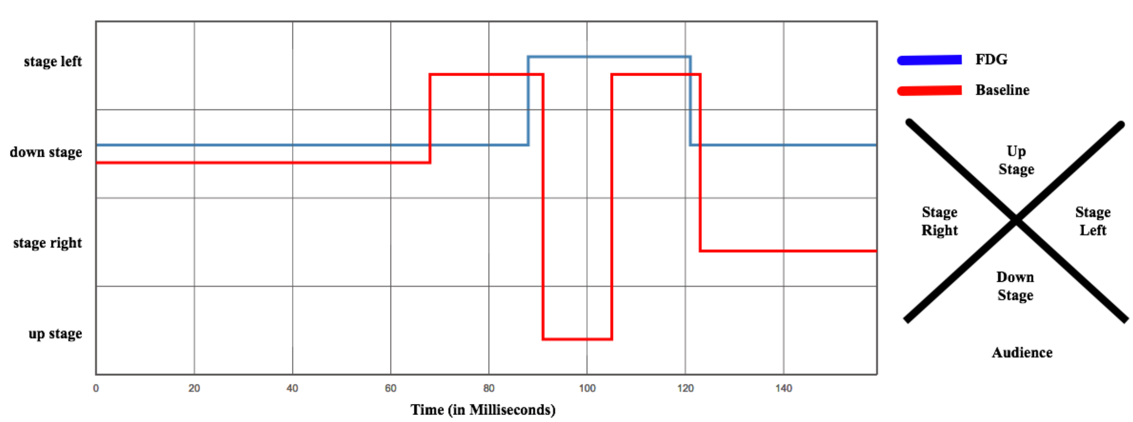


Figure 7.35: Character Gaze Traces for Epikhodov in *The Cherry Orchard*

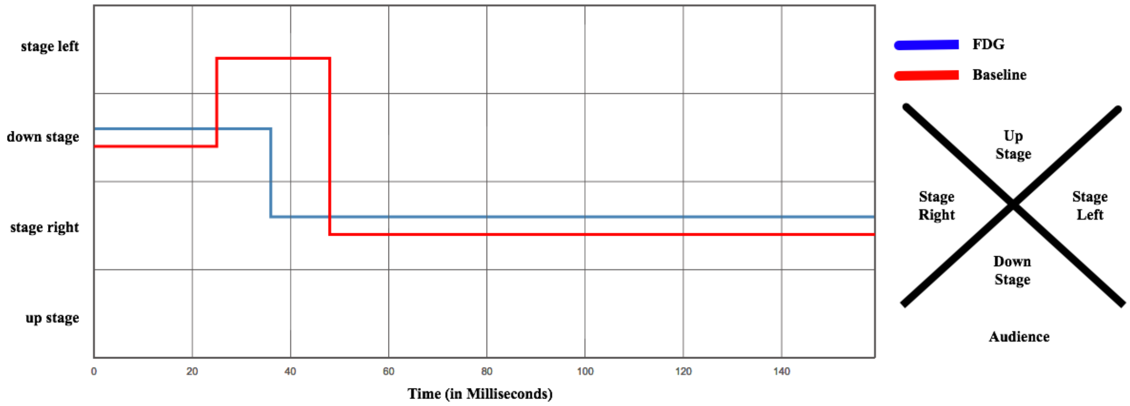


Figure 7.36: Character Gaze Traces for Trofimov in *The Cherry Orchard*

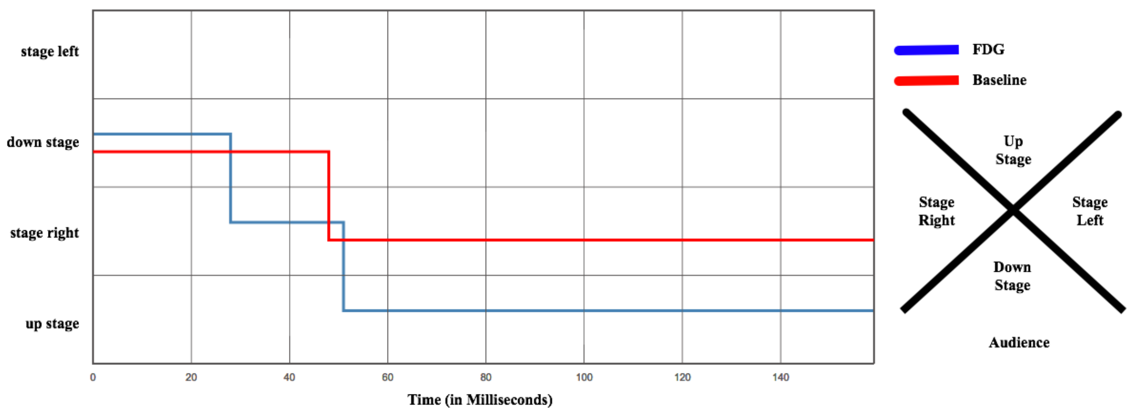


Figure 7.37: Character Gaze Traces for Lopakhin in *The Cherry Orchard*

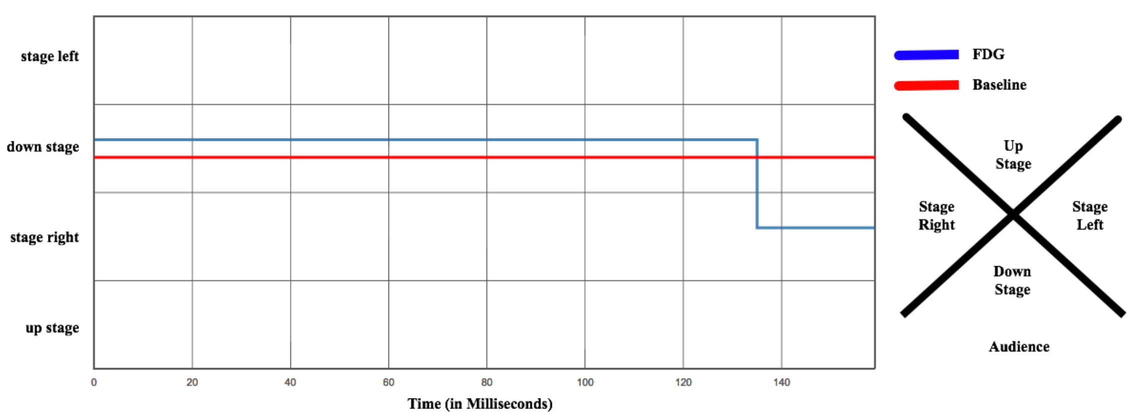


Figure 7.38: Character Gaze Traces for Varya in *The Cherry Orchard*

7.3.1.6 *The Importance of Being Earnest* Quantitative Analysis

With *The Importance of Being Earnest* play-script, we had seven characters (Jack, Lady Bracknell, Miss Prism, Gwendolen, Chasuble, Algernon, and Cecily), and two pawns in the scene. The prioritization of the characters for this scene were defined to be:

$$Jack > LadyBracknell > MissPrism > Gwendolen$$
$$> Chasuble > Algernon > Cecily$$

We took the logged position and gaze traces for each of the characters, for both the baseline scene and our FDG scene. We normalized the time for both scenes, and mapped the positions and gazes to our grid locations, as before. These grid locations and directions were then compared for a match or not, incorporating the "close enough" criteria by leveraging the generalized locations and directions.

We found that, with the forces, we were able to position the characters correctly on average 26.73% of the time, and their gazes 27.58%, as can be seen in Table 7.8. Detailed character trace information can be seen in Figures 7.39, 7.40, 7.41, 7.42, 7.43, 7.44, 7.45, 7.46, 7.47, 7.48, 7.49, 7.50, 7.51, and 7.52. Some of these discrepancies appear to be related to movements that were not annotated in the play-script.

Table 7.8: *The Importance of Being Earnest* Character Traces Match for Baseline vs. FDG

Character Name	Gaze Match	Position Match
CECILY	25.88%	0.00%
CHASUBLE	10.61%	60.96%
GWENDOLEN	41.87%	13.86%
JACK	36.49%	50.64%
LADY BRACKNELL	19.66%	2.55%
MISS PRISM	30.98%	32.39%
Overall	27.58%	26.73%

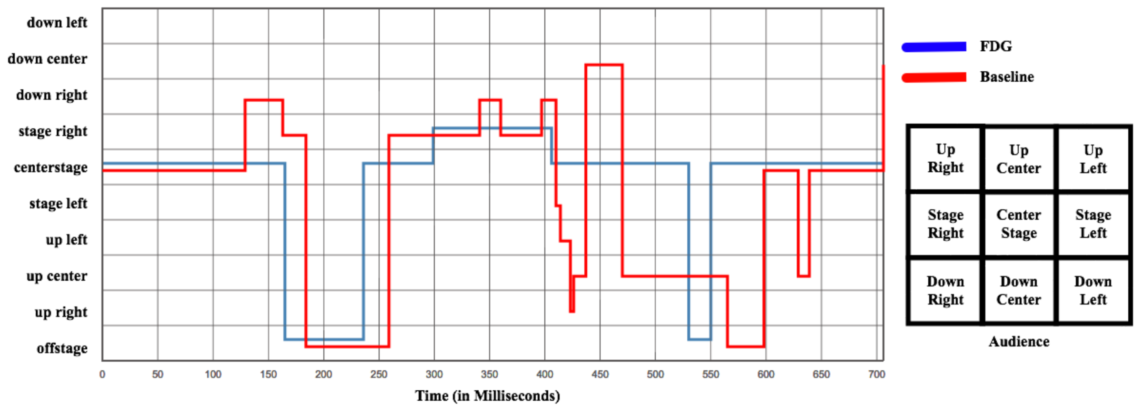


Figure 7.39: Character Position Traces for Jack in *The Importance of Being Earnest*

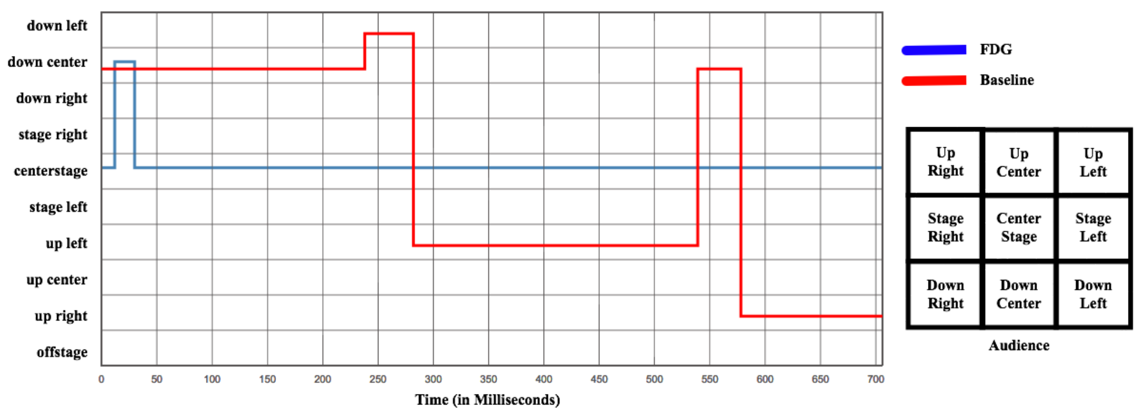


Figure 7.40: Character Position Traces for Lady Bracknell in *The Importance of Being Earnest*

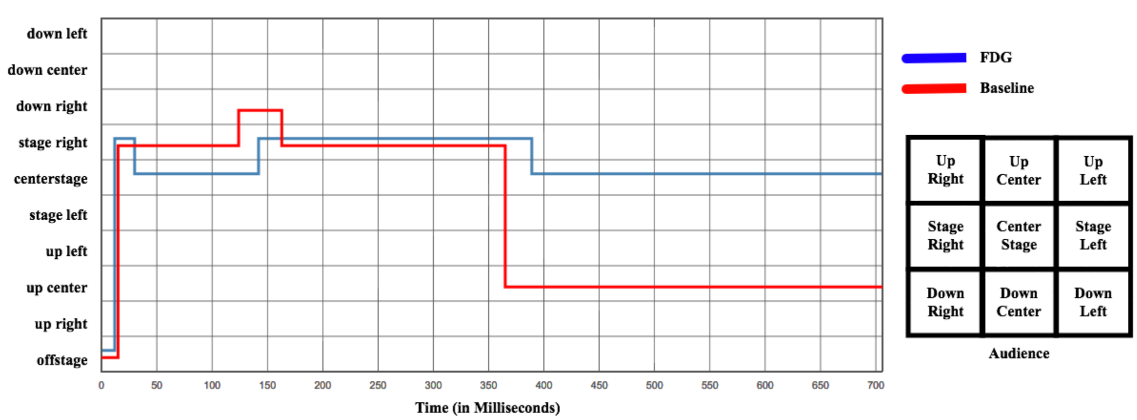


Figure 7.41: Character Position Traces for Miss Prism in *The Importance of Being Earnest*

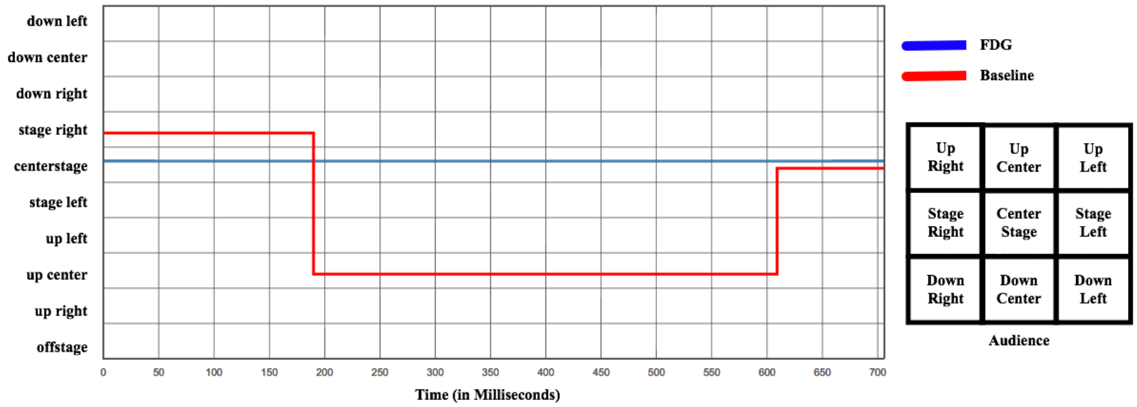


Figure 7.42: Character Position Traces for Gwendolen *The Importance of Being Earnest*

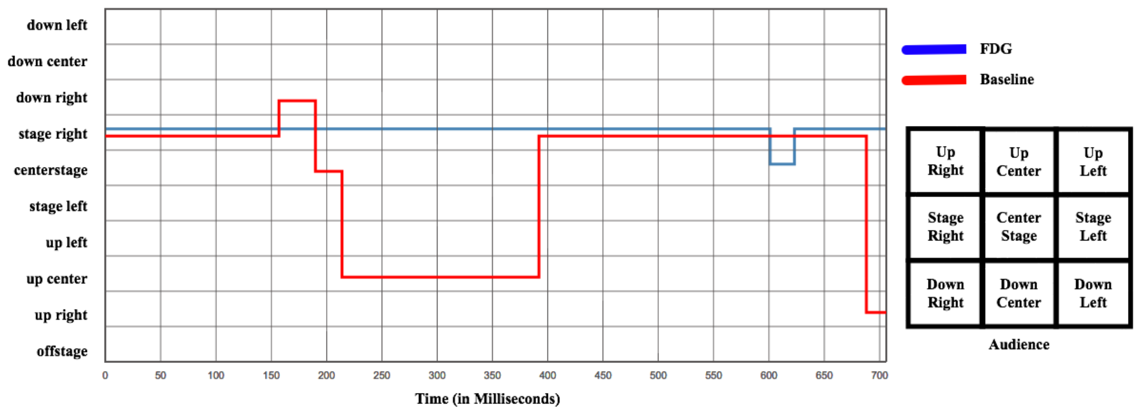


Figure 7.43: Character Position Traces for Chasuble in *The Importance of Being Earnest*

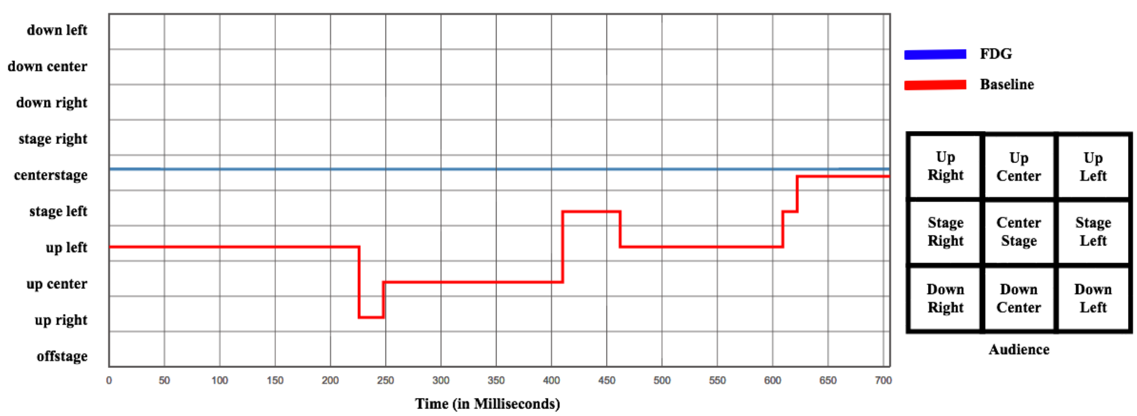


Figure 7.44: Character Position Traces for Algernon in *The Importance of Being Earnest*

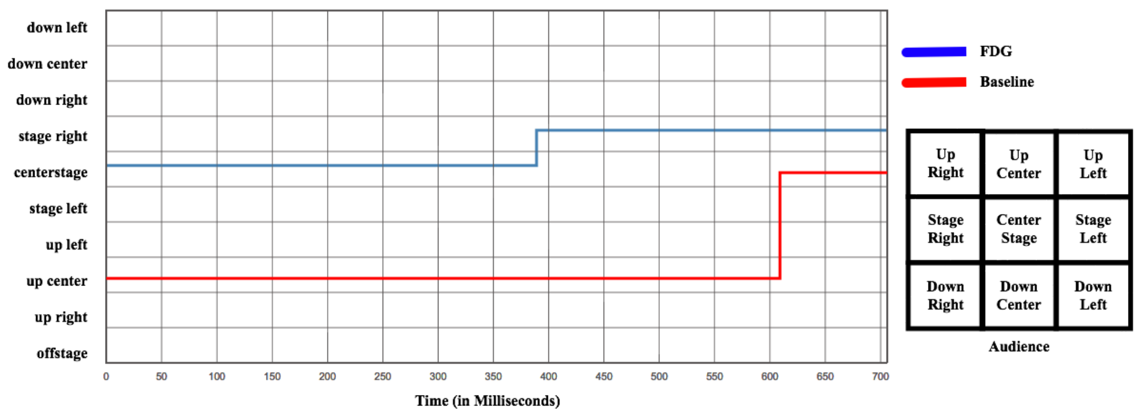


Figure 7.45: Character Position Traces for Cecily in *The Importance of Being Earnest*

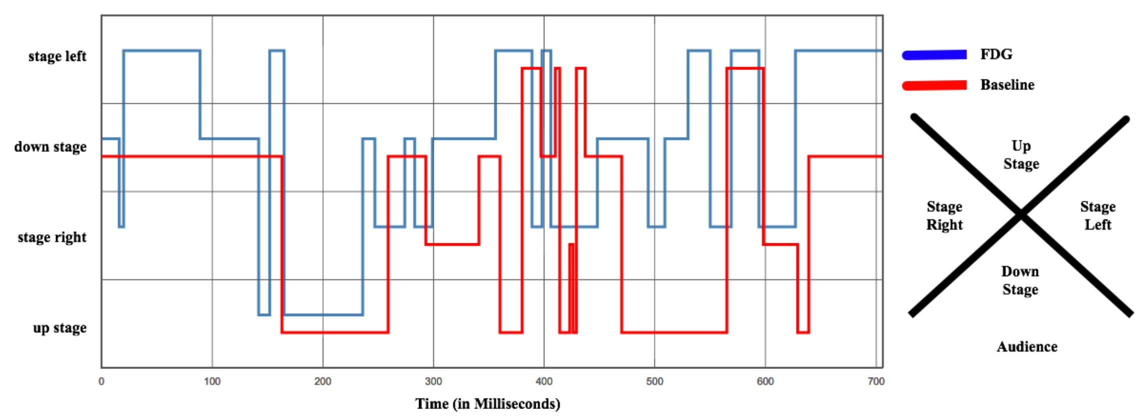


Figure 7.46: Character Gaze Traces for Jack in *The Importance of Being Earnest*

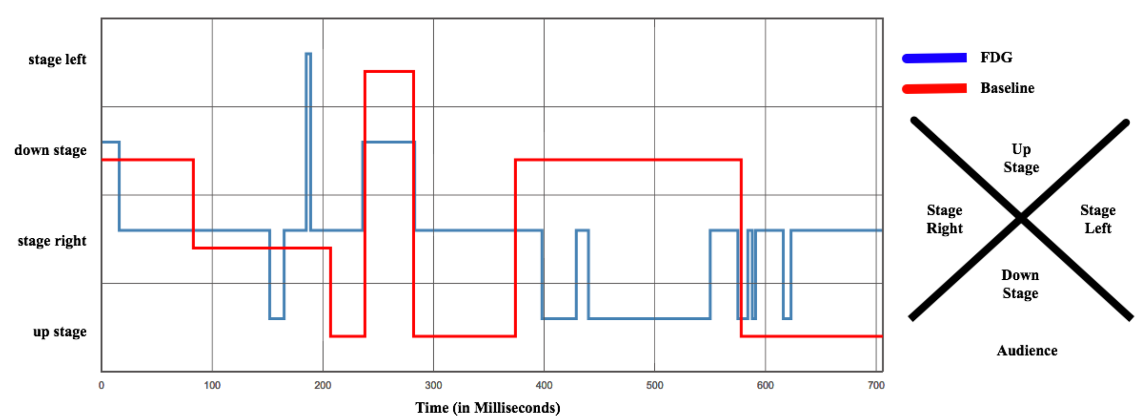


Figure 7.47: Character Gaze Traces for Lady Bracknell in *The Importance of Being Earnest*

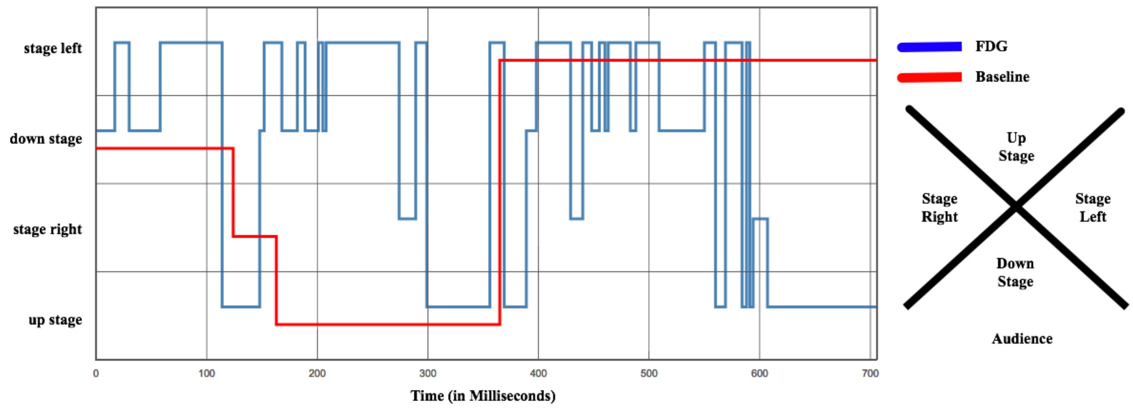


Figure 7.48: Character Gaze Traces for Miss Prism in *The Importance of Being Earnest*

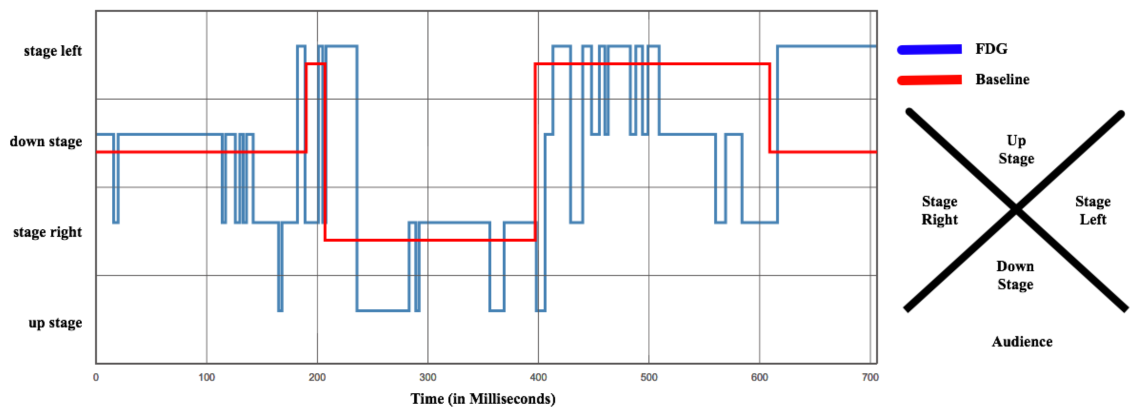


Figure 7.49: Character Gaze Traces for Gwendolen *The Importance of Being Earnest*

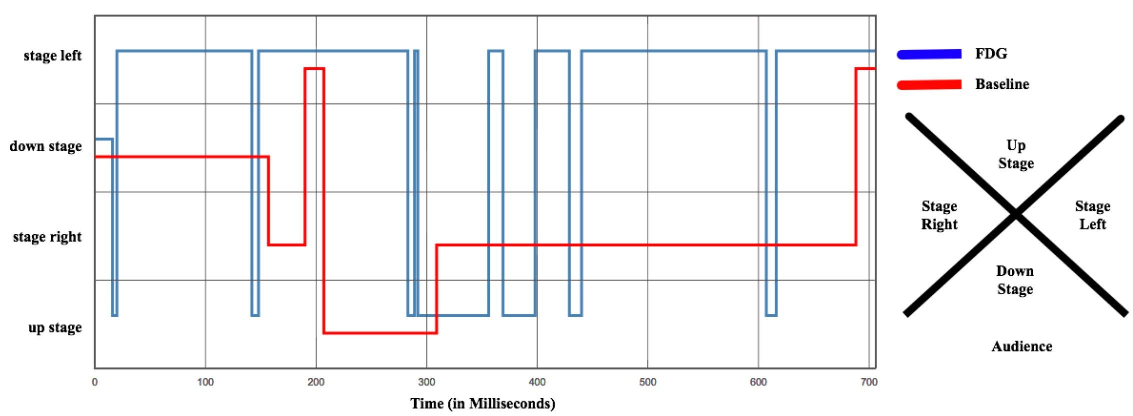


Figure 7.50: Character Gaze Traces for Chasuble in *The Importance of Being Earnest*

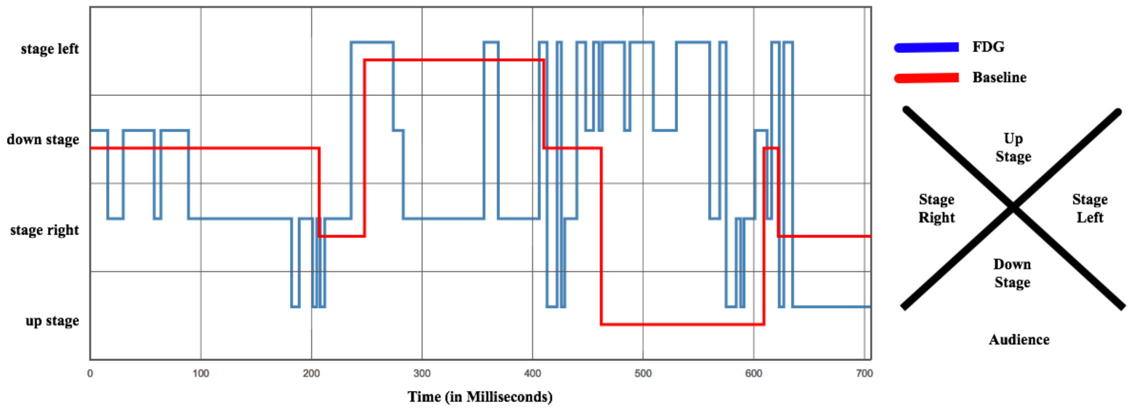


Figure 7.51: Character Gaze Traces for Algernon in *The Importance of Being Earnest*

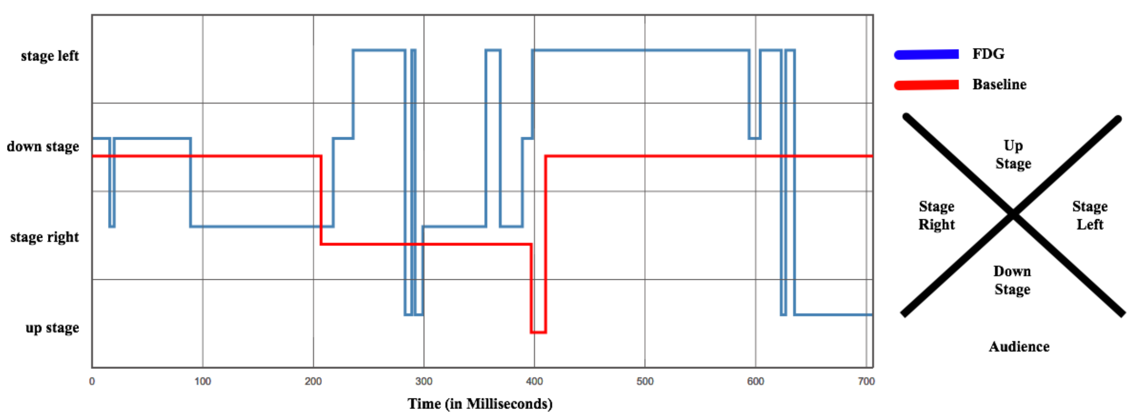


Figure 7.52: Character Gaze Traces for Cecily in *The Importance of Being Earnest*

Table 7.9: Overall Play Character Traces Match for Baseline vs. FDG

Play Title	Average Gaze Match	Average Position Match
The Importance of Being Earnest	27.58%	26.73%
Krapp's Last Tape	43.62%	100.00%
Death of a Salesman	41.57%	54.76%
Noises Off	24.55%	62.76%
The Cherry Orchard	60.94%	71.98%
Tartuffe	38.36%	31.75%
Overall	44.03%	58.00%

7.3.1.7 Quantitative Summary

Overall, we were able to position characters to match the baseline videos 58.00% of the time with their positions, and 44.03% of the time with their gazes, as seen in Table 7.9. Some scripts did better than others, and were different mostly because of missing annotations in the play-scripts or being slightly shifted outside the stage grid of where they should have been because of the forces being applied. Many of the scripts show higher gaze oscillations in the FDG videos due to the rule that has onstage characters look at the current speaker, and the Baseline videos may have missed some gaze changes during the handmapping.

7.3.2 Qualitative Analysis

We found actual recorded performances of each of the scripts identified in the Generalization Coverage section and hand-mapped the movements and gazes to BML to produce our Baseline video. We then took the same play-script (with performance-specific annotations, where available), and ran it through our NLP+Rules+FDG engine to produce a second video, which we referred to as the FDG video. Participants viewed both videos in a randomized order to avoid any order effects on the results. After each video, they were asked the questions in Figure 7.53 about what they just viewed, with the available responses being "Very Good," "Good," "Acceptable," "Poor," "Very Poor."

- **Character Positioning** Rate the quality of the Character Positioning within the performance. Ex: Were the characters too close together? Too far apart? Did the arrangement of the characters make sense?
- **Character Movement** Rate the quality of the Character Movements within the performance. Ex: Did the movements appear to be in-sync with the script? Did the characters move at unusual times? Did they move too much? Too little?
- **Stage Space** Rate the quality of the use of the Stage Space within the performance. Ex: Did the characters cover the whole stage? Only one small part of the stage? Did the use of the space make sense with respect to the scene?
- **Character Visibility** Rate the quality of the overall Character Visibility within the performance. Ex: Were the characters frequently blocking your view to another character? Were all characters visible throughout the entire scene?
- **Pace** Rate the quality of the Pace of the scene within the performance. Ex: Did it move too slow? Did it move too fast? Did the scene progress in-line with expectations for the script?
- **Overall Performance** Rate the quality of the Overall Performance. Ex: Considering the entire scene, was it pleasing or believable to you?

Figure 7.53: Likert-Scale Spatio-Temporal Questions for Generalization Study

We leveraged this questionnaire to perform a qualitative analysis on our system for each of the play-scripts defined for our generalization coverage. The goal of these studies were to determine whether we are able to provide a realistic performance which is similar to a human-performed scene from the viewer's perspective. We also asked for basic demographic information, like before, with questions in Figure 7.54.

7.3.2.1 *Tartuffe* Qualitative Analysis

With this study, each group viewed both of the videos (a handmapped Baseline video https://www.youtube.com/embed/aECjAkITzmk, and a FDG-generated video https://www.youtube.com/embed/vOkYFwKuang) and answered the questions in Figure 7.53 (within groups experiment). Both the order of the videos and the order of the questions for each video were presented in randomized order to the users.

The survey was posted on Mechanical Turk (MTurk) with criteria to enforce participants were from the United States, to avoid cultural differences of opinion regarding space and performances. Additional controls were put in place to ensure participants viewed the entire video by including a timer on the video viewing page, and including an intermission screen of a particular color mid-way in the video. If the participant did not remain on the video page long enough to watch the videos, or if they did not know what color the intermission screens were, they were disqualified from participating.

The study for this play-script included 61 participants that viewed both videos, as seen in Figure 7.56. 140 people attempted the study, but were unable to complete the study due to the controls in place to ensure proper participation. Figure 7.55 shows the breakdown of the participants by the different demographics. As you can see, it represents a reasonable sampling of the population.

This experiment provided an estimated power to detect a medium effect (f=0.25) of 0.96. We performed a Wilcoxon Signed-Rank test that showed there was a statistically significant difference in rankings for three questions, “Character Positioning,” “Stage

- In what state or U.S. territory do you live?
  - US States and Territories
  - Other - Outside the U.S.
- Which category below includes your age?
  - 17 or younger
  - 18-20
  - 21-29
  - 30-39
  - 40-49
  - 50-59
  - 60 or older
- What is your gender?
  - Male
  - Female
  - Non-binary
  - Prefer not to answer
- What is your employment status?
  - Employed, Full-time
  - Employed, Part-time
  - Student
  - Retired
  - Unemployed
  - Other
- What culture do you relate most to?
  - American
  - Arabic
  - Chinese
  - English
  - French
  - German
  - Italian
  - Japanese
  - Korean
  - Portuguese
  - Russian
  - Spanish
  - Other
- Over the last 12 months, roughly how many times have you been to see a theatre performance (including opera, musical, play, dance)?
  - 0
  - 1-3
  - 4-10
  - 11+
- In the past 7 days, roughly how many hours have you spent playing video games (e.g., gaming consoles, mobile phones, computers, etc.) involving virtual characters?
  - None
  - 1 to 3 hours
  - 4 to 6 hours
  - 7 to 9 hours
  - 10 hours or more
- How familiar are you with theatre, performances, and theatre terminology?
  - Very Familiar
  - Familiar
  - Somewhat Familiar
  - Not Familiar
- Are you familiar with the play and / or the scene prior to today's showing?
  - Read / seen it multiple times
  - Read / saw it once
  - Heard of it
  - Never read, seen, or heard of it
  - Other

Figure 7.54: Demographic Questions in the User Survey

(a) Gender

(b) Age

(c) Residence

(d) Employment

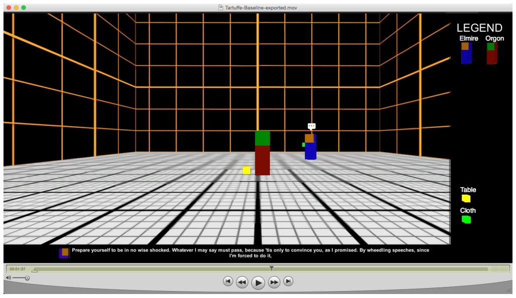
(e) Culture

(f) Games Participation

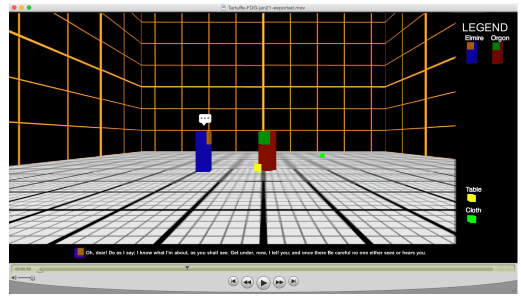
(g) Theatre Performances

(h) Scene Familiarity

Figure 7.55: Demographic Breakdown of Participants for the *Tartuffe* Within Subjects Study



(a) Baseline Video



(b) FDG Video

Figure 7.56: Screenshots of the *Tartuffe* Videos Participants Viewed

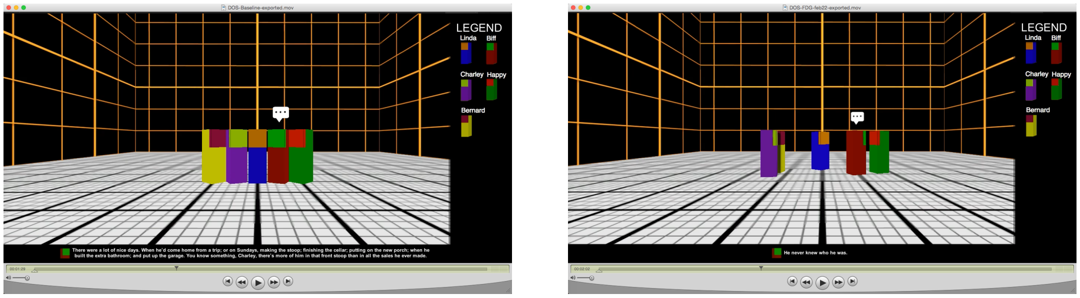
Space," and "Character Visibility." It showed that Character Positioning for the FDG video was considered better, Z=-2.519 p=0.012. It also showed that the Character Visibility for the FDG video was also considered better, Z=-4.688 p<0.001. For the last one, Stage Space, it showed that the Baseline video was considered better, although this was just barely a statistically significant finding, Z=-1.966 p=0.049.

These findings show that the FDG video was as good as, or better than, the human performance of the same scene. The one area where the Baseline was seen as better was from the use of the stage space, which was more widely used in the Baseline because Elmire does some adhoc pacing from one side of the stage to the other, as seen in our quantitative analysis section.

7.3.2.2 *Death of a Salesman* Qualitative Analysis

With this study, each group viewed both of the videos (a handmapped Baseline video https://www.youtube.com/embed/i9sQIf9ezas, and a FDG-generated video https://www.youtube.com/embed/P2C8SJrGKKk) and answered the questions in Figure 7.53 (within groups experiment). Both the order of the videos and the order of the questions for each video were presented in randomized order to the users.

The survey was posted on Mechanical Turk (MTurk) with criteria to enforce participants were from the United States, to avoid cultural differences of opinion regarding space and performances. Additional controls were put in place to ensure partici-



(a) Baseline Video (b) FDG Video

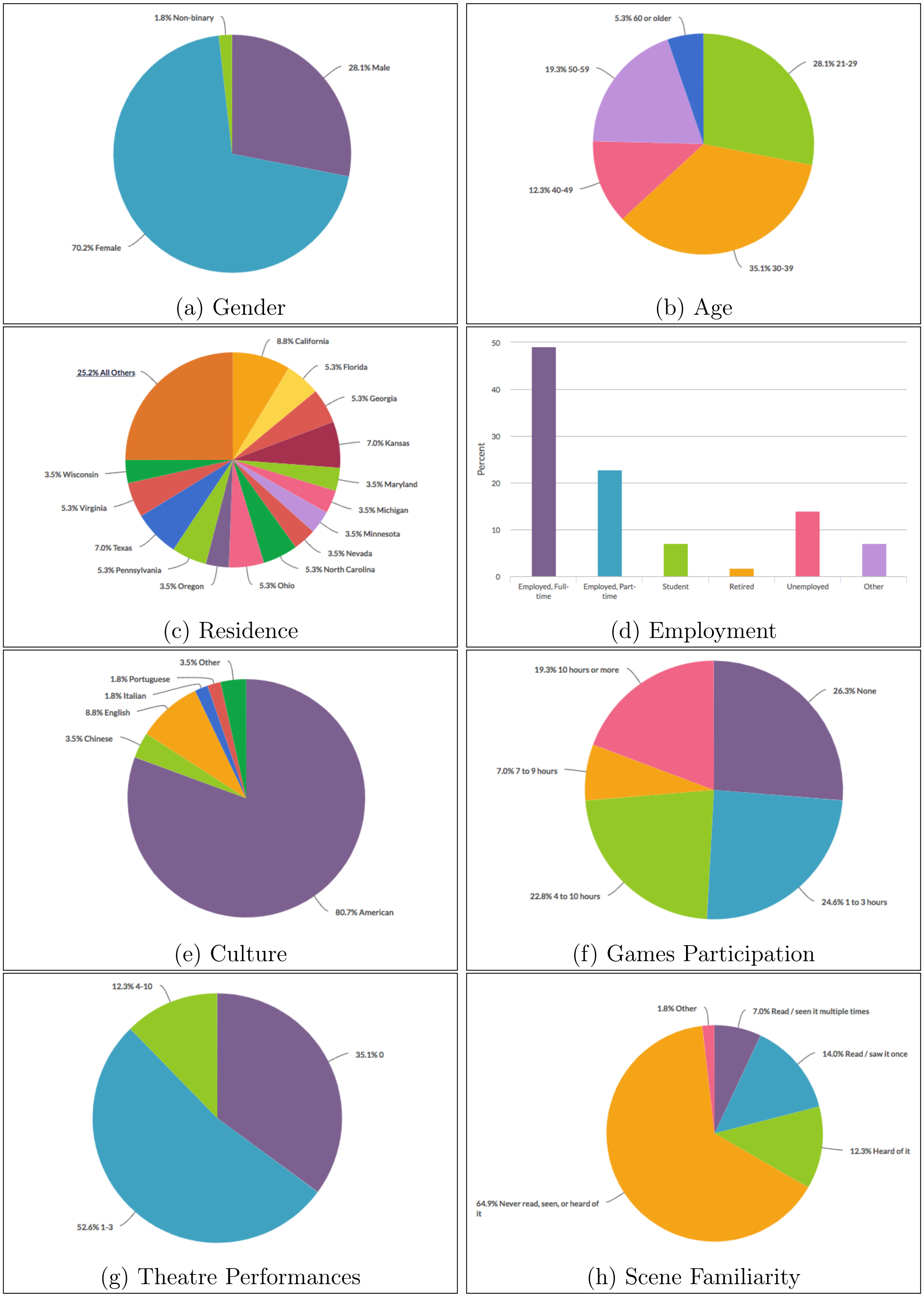
Figure 7.57: Screenshots of the *Death of a Salesman* Videos Participants Viewed

pants viewed the entire video by including a timer on the video viewing page, and including an intermission screen of a particular color mid-way in the video. If the participant did not remain on the video page long enough to watch the videos, or if they did not know what color the intermission screens were, they were disqualified from participating.

The study for this play-script included 57 participants that viewed both videos, as seen in Figure 7.57. 177 people attempted the study, but were unable to complete the study due to the controls in place to ensure proper participation. Figure 7.58 shows the breakdown of the participants by the different demographics. As you can see, it represents a reasonable sampling of the population.

This experiment provided an estimated power to detect a medium effect (f=0.25) of 0.95. We performed a Wilcoxon Signed-Rank test that showed there was a statistically significant difference in the rankings for three questions, “Character Positioning,” “Stage Space,” and “Character Visibility.” It showed that Character Positioning for the FDG video was considered better, Z=-3.017 p=0.003. It also showed that the Stage Space for the FDG video was considered better, Z=-3.481 p<0.001. For the last one, Character Visibility, it showed that the Baseline video was considered better, Z=-3.855 p<0.001.

These findings show that the FDG video was as good as, or better than, the human performance of the same scene. The one area where the Baseline was seen as better



(a) Gender (b) Age (c) Residence (d) Employment (e) Culture (f) Games Participation (g) Theatre Performances (h) Scene Familiarity

Figure 7.58: Demographic Breakdown of Participants for the *Death of a Salesman* Within Subjects Study

was from the character visibility, because of the semi-circular arrangement of the characters by the forces. This caused Bernard to be slightly behind Charley, whereas the Baseline video had all the characters side-by-side.

7.3.2.3 *Noises Off* Qualitative Analysis

With this study, each group viewed both of the videos (a handmapped Baseline video https://www.youtube.com/embed/MES_AW9MVI8, and a FDG-generated video https://www.youtube.com/embed/UtjpzK8SJI4) and answered the questions in Figure 7.53 (within groups experiment). Both the order of the videos and the order of the questions for each video were presented in randomized order to the users.

The survey was posted on Mechanical Turk (MTurk) with criteria to enforce participants were from the United States, to avoid cultural differences of opinion regarding space and performances. Additional controls were put in place to ensure participants viewed the entire video by including a timer on the video viewing page, and including an intermission screen of a particular color mid-way in the video. If the participant did not remain on the video page long enough to watch the videos, or if they did not know what color the intermission screens were, they were disqualified from participating.

The study for this play-script included 59 participants that viewed both videos, as seen in Figure 7.60. 201 people attempted the study, but were unable to complete the study due to the controls in place to ensure proper participation. Figure 7.59 shows the breakdown of the participants by the different demographics. As you can see, it represents a reasonable sampling of the population.

This experiment provided an estimated power to detect a medium effect (f=0.25) of 0.96. We performed a Wilcoxon Signed-Rank test that showed there was a statistically significant difference in the rankings for two questions, “Character Positioning” and “Character Movement.” It showed that Character Positioning for the FDG video was considered better, Z=-2.774 p=0.006. It also showed that the Character Movement

(a) Gender

(b) Age

(c) Residence

(d) Employment

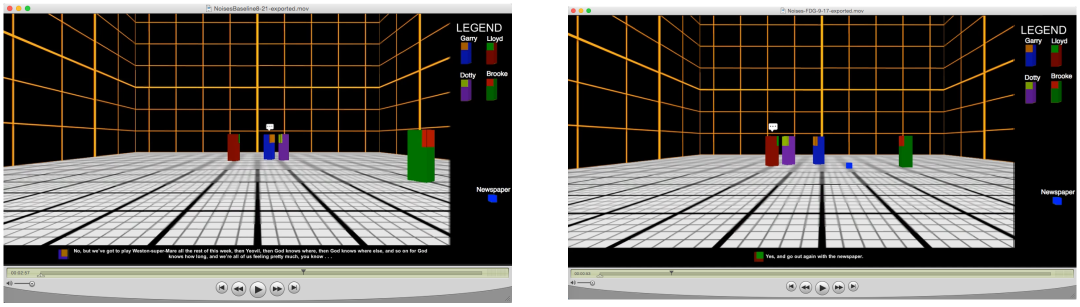
(e) Culture

(f) Games Participation

(g) Theatre Performances

(h) Scene Familiarity

Figure 7.59: Demographic Breakdown of Participants for the *Noises Off* Within Subjects Study



(a) Baseline Video (b) FDG Video

Figure 7.60: Screenshots of the *Noises Off* Videos Participants Viewed

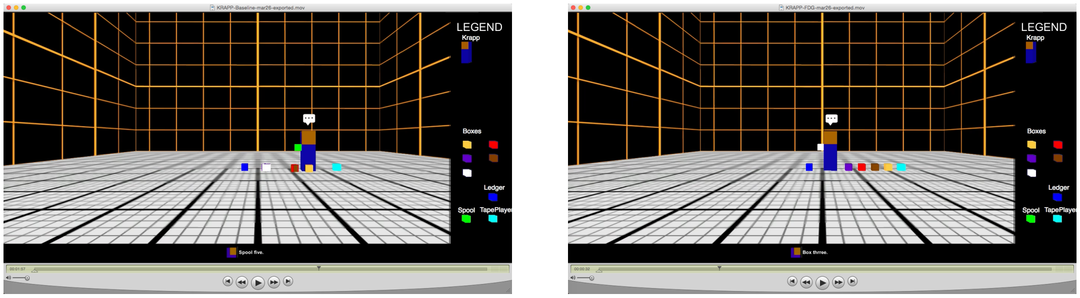
for the FDG video was also considered better, Z=-2.157 p=0.031. These findings show that the FDG video was as good as, or better than, the human performance of the same scene.

7.3.2.4 *Krapp's Last Tape* Qualitative Analysis

With this study, each group viewed both of the videos (a handmapped Baseline video https://www.youtube.com/embed/CRl9R5_IAj8, and a FDG-generated video https://www.youtube.com/embed/UMFEpH_kP-M) and answered the questions in Figure 7.53 (within groups experiment). Both the order of the videos and the order of the questions for each video were presented in randomized order to the users.

The survey was posted on Mechanical Turk (MTurk) with criteria to enforce participants were from the United States, to avoid cultural differences of opinion regarding space and performances. Additional controls were put in place to ensure participants viewed the entire video by including a timer on the video viewing page, and including an intermission screen of a particular color mid-way in the video. If the participant did not remain on the video page long enough to watch the videos, or if they did not know what color the intermission screens were, they were disqualified from participating.

The study for this play-script included 62 participants that viewed both videos, as seen in Figure 7.61. 199 people attempted the study, but were unable to complete the



(a) Baseline Video (b) FDG Video

Figure 7.61: Screenshots of the *Krapp's Last Tape* Videos Participants Viewed

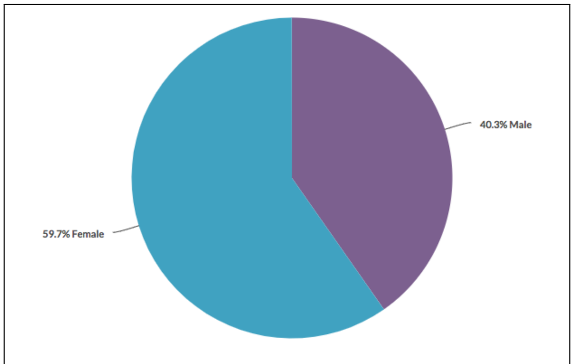
study due to the controls in place to ensure proper participation. Figure 7.62 shows the breakdown of the participants by the different demographics. As you can see, it represents a reasonable sampling of the population.

This experiment provided an estimated power to detect a medium effect (f=0.25) of 0.97. We performed a Wilcoxon Signed-Rank test that showed there was a statistically significant difference in the rankings for only one question, "Character Visibility." It showed that the Character Visibility for the FDG video was considered better, $Z=$ -2.855 p=0.004. These findings show that the FDG video was as good as, or better than, the human performance of the same scene.

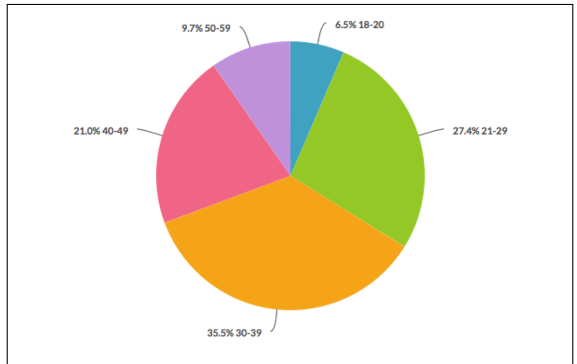
7.3.2.5 *The Cherry Orchard* Qualitative Analysis

With this study, each group viewed both of the videos (a handmapped Baseline video https://www.youtube.com/embed/RPOmPSW6eqs, and a FDG-generated video https://www.youtube.com/embed/hW316DYm69E) and answered the questions in Figure 7.53 (within groups experiment). Both the order of the videos and the order of the questions for each video were presented in randomized order to the users.

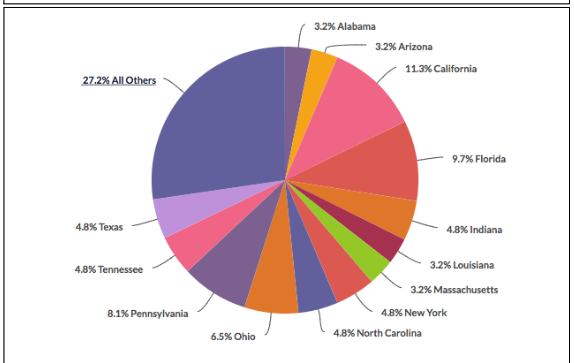
The survey was posted on Mechanical Turk (MTurk) with criteria to enforce participants were from the United States, to avoid cultural differences of opinion regarding space and performances. Additional controls were put in place to ensure partici-



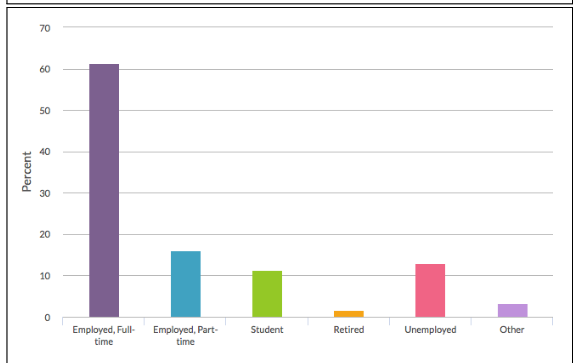
(a) Gender



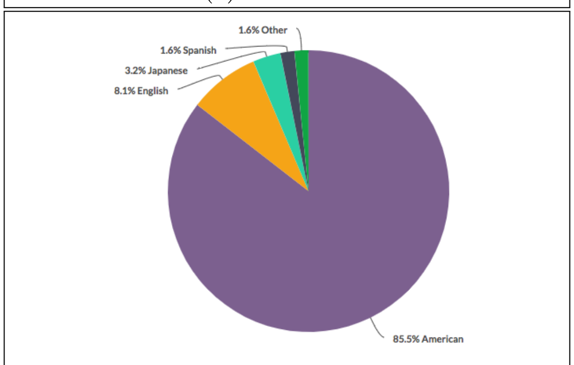
(b) Age



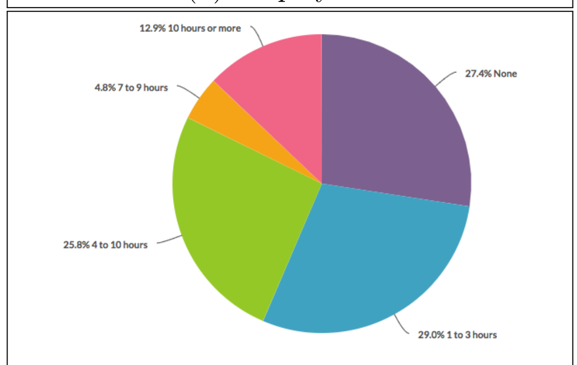
(c) Residence



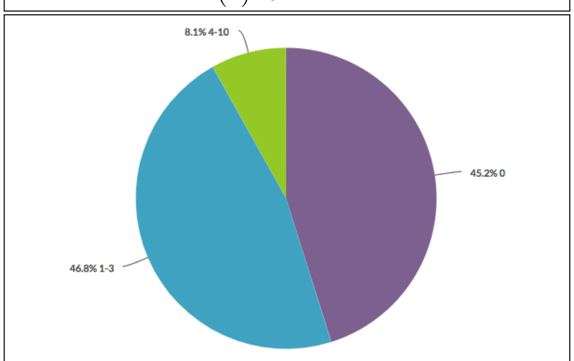
(d) Employment



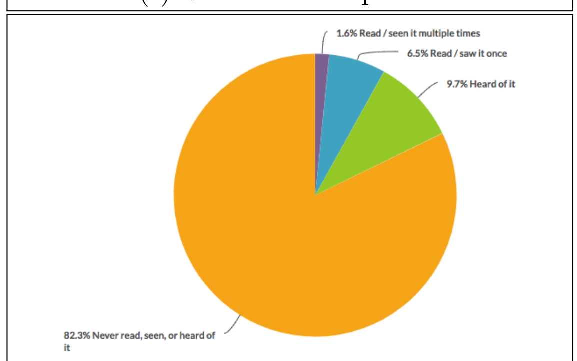
(e) Culture



(f) Games Participation

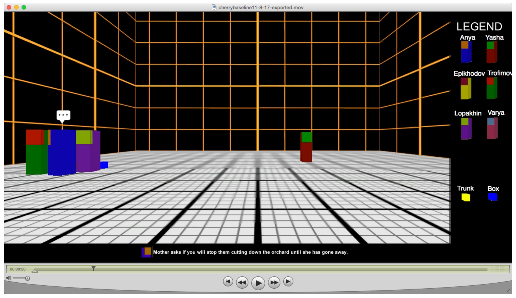


(g) Theatre Performances

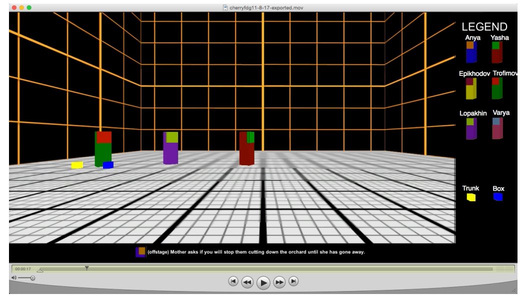


(h) Scene Familiarity

Figure 7.62: Demographic Breakdown of Participants for the *Krapp's Last Tape* Within Subjects Study



(a) Baseline Video



(b) FDG Video

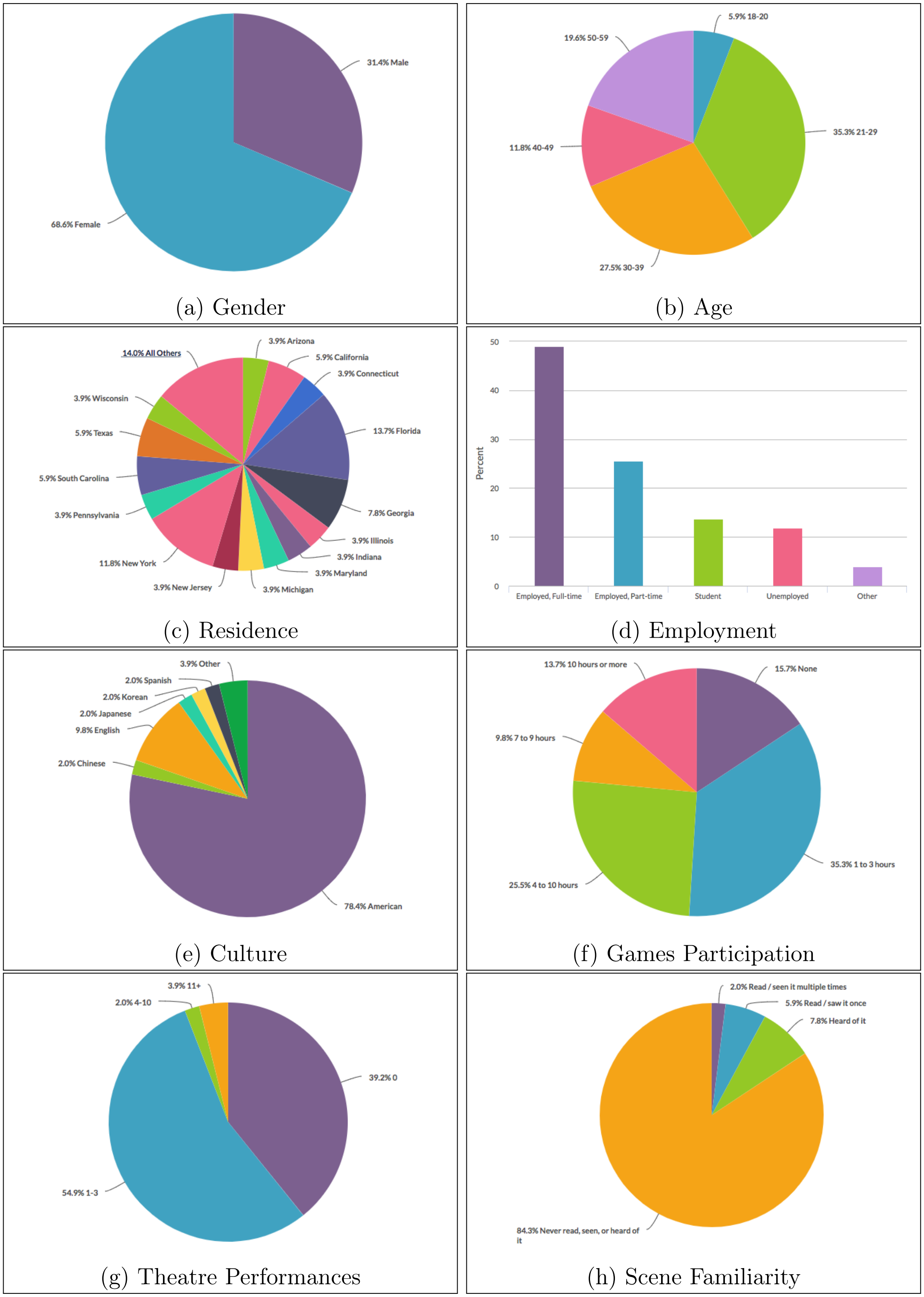
Figure 7.63: Screenshots of *The Cherry Orchard* Videos Participants Viewed

pants viewed the entire video by including a timer on the video viewing page, and including an intermission screen of a particular color mid-way in the video. If the participant did not remain on the video page long enough to watch the videos, or if they did not know what color the intermission screens were, they were disqualified from participating.

The study for this play-script included 51 participants that viewed both videos, as seen in Figure 7.63. 155 people attempted the study, but were unable to complete the study due to the controls in place to ensure proper participation. Figure 7.64 shows the breakdown of the participants by the different demographics. As you can see, it represents a reasonable sampling of the population.

This experiment provided an estimated power to detect a medium effect (f=0.25) of 0.93. We performed a Wilcoxon Signed-Rank test that showed there was a statistically significant difference in the rankings for only one question, "Character Movement." It showed that the Character Movement for the Baseline video was considered better, Z=-2.452 p=0.014.

We attempted to follow-up with these participants to determine why they rated the FDG video worse for the "Character Movement" question. We had them view both videos again and answer the more detailed questions from our original between-subjects study for *Hamlet* (Figure 7.65), with the response options of "Strongly Agree," "Agree," "Disagree," "Strongly Disagree," and "I Don't Know."



(a) Gender (b) Age (c) Residence (d) Employment (e) Culture (f) Games Participation (g) Theatre Performances (h) Scene Familiarity

Figure 7.64: Demographic Breakdown of Participants for *The Cherry Orchard* Within Subjects Study

1. Characters showed evidence of engaged listening
2. Characters appeared to perform suitable movements on cue
3. The characters' movement onstage during the performance was believable in the context of the performance
4. The performance is free from distracting behavior that does not contribute to the scene
5. The character movements provide appropriate dramatic emphasis
6. All visible behaviors appear to be motivated and coordinated within the scene
7. The movements of the characters were consistent with the play
8. There was a great deal of random movement
9. The characters' reactions to other characters were believable
10. Characters showed a lack of engagement when listening

Figure 7.65: Likert-Scale Spatio-Temporal Questions for *The Cherry Orchard* Follow-up Study

We also asked them why they rated the Baseline video better than the FDG video. Only two participants responded to the follow-up survey, and their general comment on why they rated the Baseline video (video #1) higher than the FDG video (video #2) was: "The movements in video number one seem to be more fluid than video two." Since question number eight and question number ten are worded negatively, so Agreeing more on Baseline would mean the FDG video was better. As far as the other questions, Table 7.10 shows which video was rated higher for each question. The highlighted cells show the questions where one of the participants rated the Baseline video better than the FDG video, and the red text shows the questions that were worded negatively. All other questions they liked the FDG video the same or more than the Baseline video.

Therefore, it appears in *The Cherry Orchard* videos, the Baseline video's characters showed more engaged listening and coordinated movements within the scene, which caused it to be rated higher. However, comparing the two scenes, there were minimal differences in the general positioning of the characters with respect to each other. The FDG technique placed the characters closer to the center of the stage than the handmapped baseline performance did.

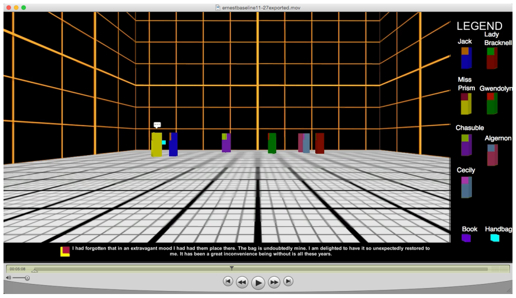
7.3.2.6 *The Importance of Being Earnest* Qualitative Analysis

With this study, each group viewed both of the videos (a handmapped Baseline video https://www.youtube.com/embed/g3Y4UvBcDCw, and a FDG-generated video https://www.youtube.com/embed/GVlYGH53730) and answered the questions in Figure 7.53 (within groups experiment). Both the order of the videos and the order of the questions for each video were presented in randomized order to the users.

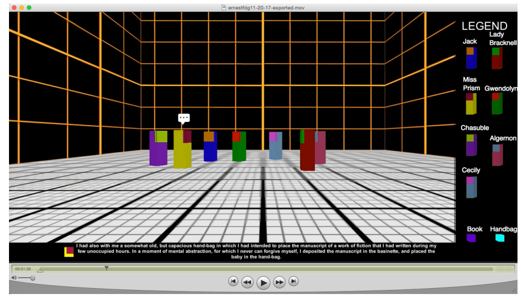
The survey was posted on Mechanical Turk (MTurk) with criteria to enforce participants were from the United States, to avoid cultural differences of opinion regarding space and performances. Additional controls were put in place to ensure participants viewed the entire video by including a timer on the video viewing page, and

Table 7.10: *The Cherry Orchard* Follow-up Questions

Question	Rating1	Rating2
The character movements provide appropriate dramatic emphasis	Agreed More on FDG	Agreed More on FDG
The characters' reactions to other characters were believable	Agreed More on FDG	Agreed More on FDG
Characters showed a lack of engagement when listening	Agreed More on FDG	Agreed More on FDG
Characters showed evidence of engaged listening	Agreed More on FDG	Same Rating
Characters appeared to perform suitable movements on cue	Agreed More on FDG	Same Rating
The characters' movement onstage during the performance was believable in the context of the performance	Same Rating	Same Rating
The performance is free from distracting behavior that does not contribute to the scene	Same Rating	Same Rating
The movements of the characters were consistent with the play	Agreed More on FDG	Agreed More on Baseline
There was a great deal of random movement	Agreed More on Baseline	Same Rating
All visible behaviors appear to be motivated and coordinated within the scene	Agreed More on Baseline	Agreed More on Baseline



(a) Baseline Video



(b) FDG Video

Figure 7.66: Screenshots of *The Importance of Being Earnest* Videos Participants Viewed

including an intermission screen of a particular color mid-way in the video. If the participant did not remain on the video page long enough to watch the videos, or if they did not know what color the intermission screens were, they were disqualified from participating.

The study for this play-script included 51 participants that viewed both videos, as seen in Figure 7.66. 419 people attempted the study, but were unable to complete the study due to the controls in place to ensure proper participation. Figure 7.67 shows the breakdown of the participants by the different demographics. As you can see, it represents a reasonable sampling of the population.

This experiment provided an estimated power to detect a medium effect (f=0.25) of 0.93. We performed a Wilcoxon Signed-Rank test that showed there was a statistically significant difference in the rankings for one question, "Character Visibility." It showed that Character Visibility for the Baseline video was considered better, Z=-2.312 p=0.021. For this video, this was expected, since we saw that with the larger number of characters in the scene, the characters were forced into more of a circle than a semi-circle, which caused some extra occlusion of the characters. Parameter adjustments could be made to increase the conversation space of the characters, and thereby allow for more characters side-by-side in a semi-circle without occluding others.

(a) Gender (b) Age (c) Residence (d) Employment (e) Culture (f) Games Participation (g) Theatre Performances (h) Scene Familiarity

Figure 7.67: Demographic Breakdown of Participants for *The Importance of Being Earnest* Within Subjects Study

7.3.2.7 Qualitative Summary

Overall, the NLP+Rules+FDG videos performed as good as, or better than the hand-mapped Baseline videos, which represented an actual performance by real actors. There were a few instances where the participants thought the Baseline video was better—mostly around "Stage Space" and "Character Movement." This shows that to a viewer, our techniques meet the expectations for a quality performance. One viewer that viewed the FDG *The Cherry Orchard* video first, the Baseline video second, said: "The second scene was a strong favorite above the first although I liked aspects of both." Another participant who viewed *The Importance of Being Earnest* FDG video first, and the Baseline video second, stated: "The first video seemed more like a movie, with the characters talking to themselves, while the second seemed more theatrical, with asides to the audience." Both highlight how these techniques meet or exceed what humans perceive as "good."

Yet another participant who viewed the *Noises Off* FDG video first, then the Baseline video second, stated: "The first run I was very bored, the players were clustered together, it was all hard to understand. The second time I was much more entertained and didn't feel the same confusion and boredom." This highlights some of the ordering effect that may have been in play with these studies. A summary of the statistically significant differences between the FDG and Baseline videos can be seen in Table 7.11.

Table 7.11: Significant Differences Between Baseline and FDG Videos, Z Values and Which is Better (Empty Cells Mean No Statistically Significant Differences Found)

Play Title	Char Postn	Char Mvmt	Stage Space	Char Vis	Pace	Overall
The Importance of Being Earnest				Z = -2.312 $p = 0.021$ Baseline		
Death of a Salesman	Z = -3.017 $p = 0.003$ FDG		Z = -3.481 $p < 0.001$ FDG	Z = -3.855 $p < 0.001$ Baseline		
Krapp's Last Tape				Z = -2.855 $p = 0.004$ FDG		
Noises Off	Z = -2.774 $p = 0.006$ FDG	Z = -2.157 $p = 0.031$ FDG				
Tartuffe	Z = -2.519 $p = 0.012$ FDG		Z = -1.966 $p = 0.049$ Baseline	Z = -4.688 $p < 0.001$ FDG		
The Cherry Orchard		Z = -2.452 $p = 0.014$ Baseline				

7.4 Generalization Conclusions

With being able to cover over 71% of all spatio-temporal dimension combinations with a proscenium (single) audience point, we were able to show that our technique of NLP+Rules+FDG can provide as good or better spatio-temporal blocking as the same human-performed scenes from a qualitative perspective. Also, we are able to see that quantitatively, we are able to maintain a similar blocking as the human-performed scene 58.00% of the time for position, while maintaining the integrity of the play-script.

CHAPTER 8: INTERACTIVE EXPERIMENTATION

In this chapter, we explore the additional features that force-directed graphs (FDGs) provide through our engine. We perform an interactive within subjects study that compares our natural language processor plus rules (NLP+Rules) engine versus our natural language processor, rules, and force-directed graphs (NLP+Rules+FDG) engine. We measured the overall performance experience, spatio-temporal aspects of the performance, and perceived workload for performing the scenes. Overall, we found that there was no statistically significant differences between the NLP+Rules versus NLP+Rules+FDG performances with respect to the experience or spatio-temporal aspects. However, we did find that the FDG version did provide greater clustering of the characters to the human-controlled character than just the Rules alone. This shows that our technique does provide a more spatially-inclusive experience for participants.

8.1 Methodology

For our interactive study, we wanted to understand whether our techniques with force-directed graphs would produce a more inclusive scene for the human-controlled character. To study this, we utilized three different questionnaires:

1. Spatio-temporal questions: Our previously utilized short questionnaire around space and timing within the scene.

2. Experience questions: A new questionnaire to define the user's experience, based on theatrical experiential studies.

3. Task Load questions: A standardized questionnaire to define the user's subjective mental workload.

- **Character Positioning** Rate the quality of the Character Positioning within the performance. Ex: Were the characters too close together? Too far apart? Did the arrangement of the characters make sense?
- **Character Movement** Rate the quality of the Character Movements within the performance. Ex: Did the movements appear to be in-sync with the script? Did the characters move at unusual times? Did they move too much? Too little?
- **Stage Space** Rate the quality of the use of the Stage Space within the performance. Ex: Did the characters cover the whole stage? Only one small part of the stage? Did the use of the space make sense with respect to the scene?
- **Character Visibility** Rate the quality of the overall Character Visibility within the performance. Ex: Were the characters frequently blocking your view to another character? Were all characters visible throughout the entire scene?
- **Pace** Rate the quality of the Pace of the scene within the performance. Ex: Did it move too slow? Did it move too fast? Did the scene progress in-line with expectations for the script?
- **Overall Performance** Rate the quality of the Overall Performance. Ex: Considering the entire scene, was it pleasing or believable to you?

Figure 8.1: Likert-Scale Spatio-Temporal Questions for Interactive Study

More information on our spatio-temporal questions can be found in Chapter 6: EXPERIMENTATION AND DISCUSSION, Section 6.4.4: Multiple Component-Based Within Subjects Evaluation and Chapter 7: GENERALIZATION, Section 7.3.2: Qualitative Analysis. These questions can also be seen in Figure 8.1, and were Likert-scaled questions with possible responses being: “Very Good,” “Good,” “Acceptable,” “Poor,” and “Very Poor.”

For our Experiential questions, we leveraged the document “Capturing the audience experience: A handbook for the theatre” [69], which provided sample templates and questions for evaluating audience experience. We leveraged sample questions from each of the five areas for audience experience: engagement and concentration, learning

- I felt engaged in the scene.
- There were aspects of the performance that I found difficult or challenging.
- I feel that I shared the experience with the other characters in the scene.
- It was difficult to determine what to do / where to go / what to say.
- I felt I could identify with the characters / story.
- I thought this was fun.

Figure 8.2: Likert-Scale Experience Questions for Interactive Study

and challenge, energy and tension, shared experience and atmosphere, and personal resonance and emotional connection. The questions in Figure 8.2 were asked with a Likert scale of “Strongly Agree,” “Agree,” “Disagree,” “Strongly Disagree,” and “I Don’t Know.”

We also included our standard demographic questions to ensure appropriate sampling from the population, as seen in Figure 8.3.

Lastly, we wanted to evaluate the mental workload for the participant, so we leveraged the National Aeronautics and Space Administration Task Load Index (NASA-TLX) [10]. This tool measures the subjective mental workload of a participant while they are performing a task. It evaluates performance across six dimensions to determine an overall workload rating, on a scale from 0 to 100. These include:

1. **Mental demand**—how much thinking, deciding, or calculating was required to perform the task. Rated “low” to “high.”

2. **Physical demand**—the amount and intensity of physical activity required to complete the task. Rated “low” to “high.”

3. **Temporal demand**—the amount of time pressure involved in completing the task. Rated “low” to “high.”

- In what state or U.S. territory do you live?
  - US States and Territories
  - Other - Outside the U.S.
- Which category below includes your age?
  - 17 or younger
  - 18-20
  - 21-29
  - 30-39
  - 40-49
  - 50-59
  - 60 or older
- What is your gender?
  - Male
  - Female
  - Non-binary
  - Prefer not to answer
- What is your employment status?
  - Employed, Full-time
  - Employed, Part-time
  - Student
  - Retired
  - Unemployed
  - Other
- What culture do you relate most to?
  - American
  - Arabic
  - Chinese
  - English
  - French
  - German
  - Italian
  - Japanese
  - Korean
  - Portuguese
  - Russian
  - Spanish
  - Other
- Over the last 12 months, roughly how many times have you been to see a theatre performance (including opera, musical, play, dance)?
  - 0
  - 1-3
  - 4-10
  - 11+
- In the past 7 days, roughly how many hours have you spent playing video games (e.g., gaming consoles, mobile phones, computers, etc.) involving virtual characters?
  - None
  - 1 to 3 hours
  - 4 to 6 hours
  - 7 to 9 hours
  - 10 hours or more
- How familiar are you with theatre, performances, and theatre terminology?
  - Very Familiar
  - Familiar
  - Somewhat Familiar
  - Not Familiar
- Are you familiar with the play *The Importance of Being Earnest* by Oscar Wilde?
  - Read / seen it multiple times
  - Read / saw it once
  - Heard of it
  - Never read, seen, or heard of it
  - Other

Figure 8.3: Demographic Questions in the User Survey

4. **Effort**—how hard does the participant have to work to maintain their level of performance? Rated "low" to "high."

5. **Performance**—the level of success in completing the task. Rated "good" to "poor."

6. **Frustration level**—how insecure, discouraged, or secure or content the participant felt during the task. Rated "low" to "high."

It also utilizes a paired comparisons method, which presents the fifteen pairwise combinations of the above six dimensions to the participants and ask them to select which one of each pair had the most effect on the workload during the task. Both sets of responses are utilized to generate a single score of their subjective workload for the given task, and can be used to compare effort between multiple tasks.

8.2 Qualitative Evaluation

Participants were asked to perform three scenes as part of this within subjects experiment. The first scene was leveraged as a practice scene, where they learned how to interact with the application and none of the AI characters moved. The application included a performance stage, a legend on the right, and the play-script on the bottom, which highlighted the current block of the script being performed, as seen in Figure 8.5. Participants performed all scenes as Miss Prism, and leveraged the "Next" button to progress through the scene at their own pace. They were able to move to, pick up, put down, point to, and gaze at objects and locations by clicking on the target on the screen, as can be seen in the instructions in Figure 8.4. They were also able to make Miss Prism speak her lines via a "Speak" button near the script, and below the legend.

After the first scene, the participant was then asked to complete the TLX questionnaire for both the pairwise combinations, as well as the scaled effort rating for the practice scene.

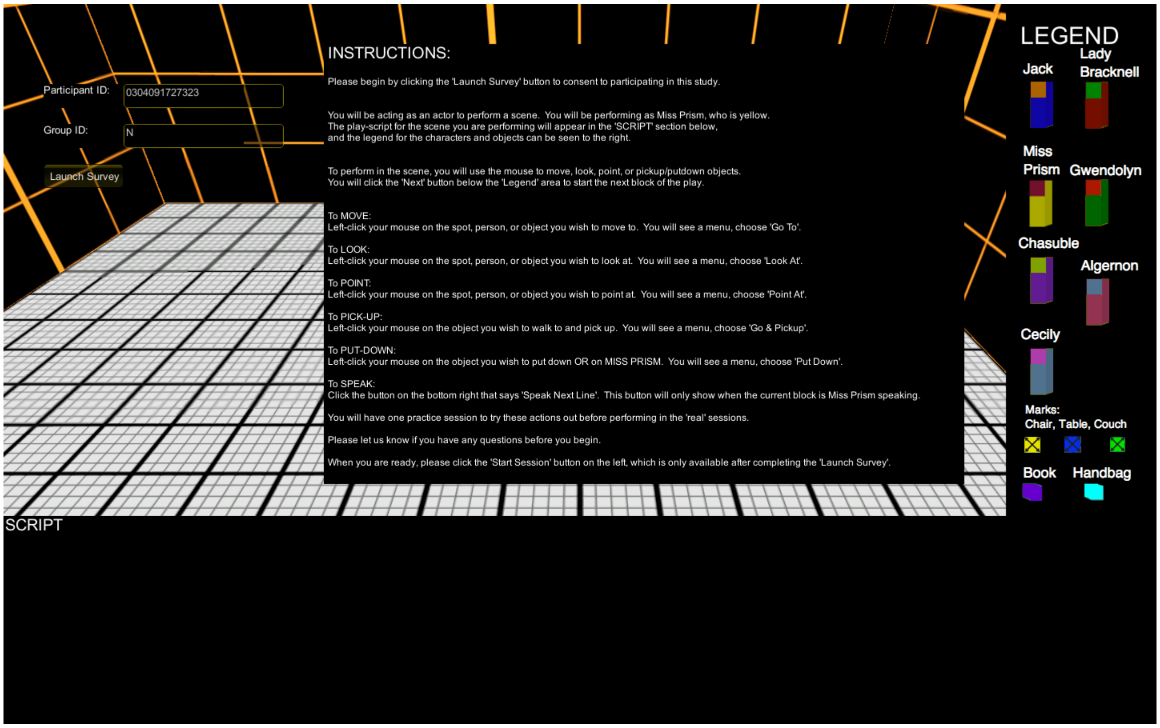


Figure 8.4: Screenshot of the Interactive Application's Instructions Utilized by Participants

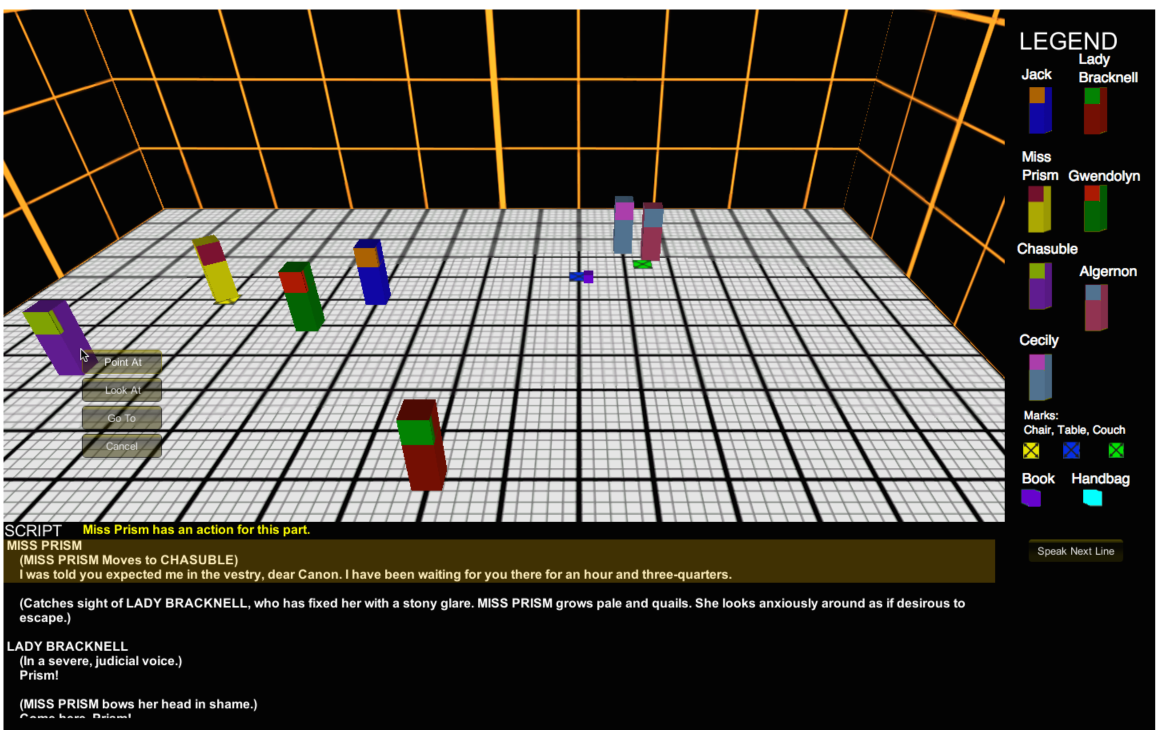


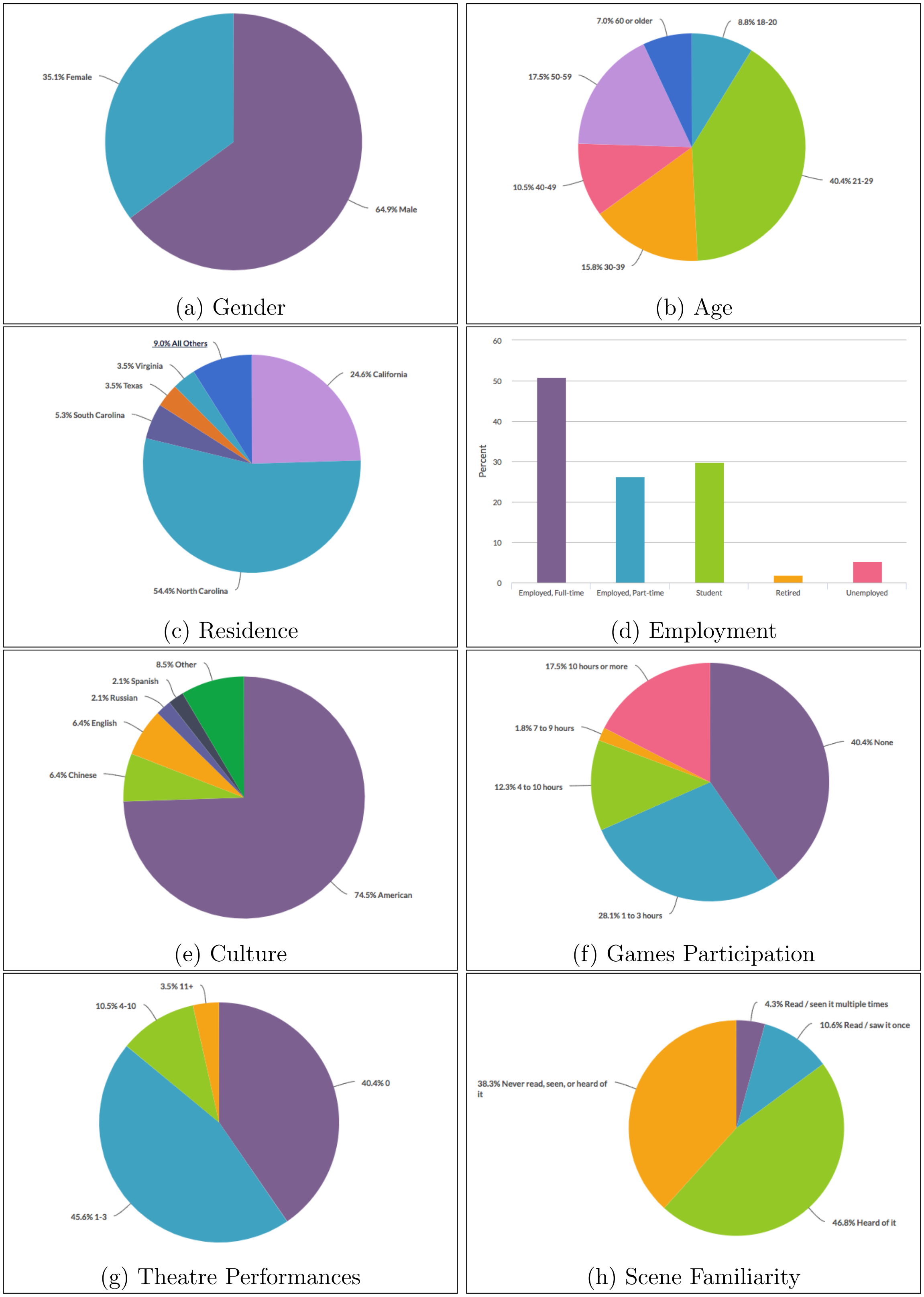
Figure 8.5: Screenshot of the Interactive Application Utilized by Participants

The participants were then presented (randomly) either the FDG or the Rules scene to perform. The only difference between these two scenes was the incorporation of the force-directed graphs component of our techniques in the FDG scene, which would adjust the characters to incorporate the human-controlled character. The Rules scene would only perform the scene as transcribed, incorporating only the rules component. This scene was the same scene's play-script for both the FDG and the Rules performances, however the practice scene was a different scene from *The Importance of Being Earnest* play-script.

After performing this scene, participants were asked both the Spatio-temporal questions from Figure 8.1, and the Experience questions from Figure 8.2. They also completed the scaled effort ratings for the TLX questionnaire.

The participants were then asked to perform their third scene, which was the remaining un-performed scene (randomly the Rules or FDG scene, respectively). After completing this third, and last, scene, participants were once again asked both the Spatio-temporal questions from Figure 8.1, and the Experience questions from Figure 8.2. They also completed the scaled effort ratings for the TLX questionnaire. A trimmed video that shows the Rules and FDG scenes can be seen at https://www.youtube.com/embed/kYdR-EKThH4.

This study included 57 participants that performed all three scenes, with a power to detect a medium effect (f=0.25) of 0.95. The breakdown of demographics for this study can be seen in Figure 8.6. We performed a Wilcoxon Signed-Rank test that showed there was no statistically significant difference in the rankings for any of the Experience or Spatio-temporal questions. There appear to be some differences related to ordering effects, but due to the short scenes used in this study, we were unable to overcome the novelty effect of the study to determine real differences. We plan to perform more longitudinal studies in the future to further explore the inclusiveness of our techniques. It is also believed that the wording of the inclusiveness question



(a) Gender (b) Age (c) Residence (d) Employment (e) Culture (f) Games Participation (g) Theatre Performances (h) Scene Familiarity

Figure 8.6: Demographic Breakdown of Participants for *The Importance of Being Earnest* Interactive Within-Subjects Study

may have been misleading. It appears that participants interpreted it as engagement to the overall scene, not positioning of where their character was with respect to the other characters.

Also, this shows that the Rules and FDG videos were qualitatively the same, but not necessarily "good." So, we reviewed the histograms for the questions, as seen in Figure 8.7 and Figure 8.8. When looking at the spatio-temporal questions, we see a definite skew towards "Good" and "Very Good" reponses. This clarifies that these videos are generally seen as good performances, from a spatio-temporal perspective. Most participants seemed to think this study was fun, and it was a bit of a wash with respect to participants feeling they shared the experience.

We also completed a repeated measures ANOVA with a Greenhouse-Geisser correction, which determined that the mean TLX Score differed statistically significantly between sessions ($F(1.772, 99.223)=37.892$, $p<0.001$). Post hoc tests using the Bonferroni correction revealed that the practice session was statistically significantly different than both the FDG and the Baseline sessions, $p<0.001$ for both. It showed no statistically significant difference between the FDG and Baseline sessions, $p = 0.954$. This shows that there was a learning curve to overcome during the practice session, but the subsequent sessions were not any more difficult to perform, regardless of which session they performed first.

Verbal discussions with the participants after the study generally stated that the FDG scene did make them feel more included in the scene with the other characters than the Rules scene. Several statements in the questionnaire confirmed this opinion, such as "I liked the how the characters were positioned more in this scene. Miss Prism felt like the focal point with everyone focused on her" for the FDG scene.

Some had mixed feelings about the extra movement in the FDG scene: "Characters were more bunched to the right and did not constantly adjust, which was good and bad. Good for allowing the ability to work around the characters without too much

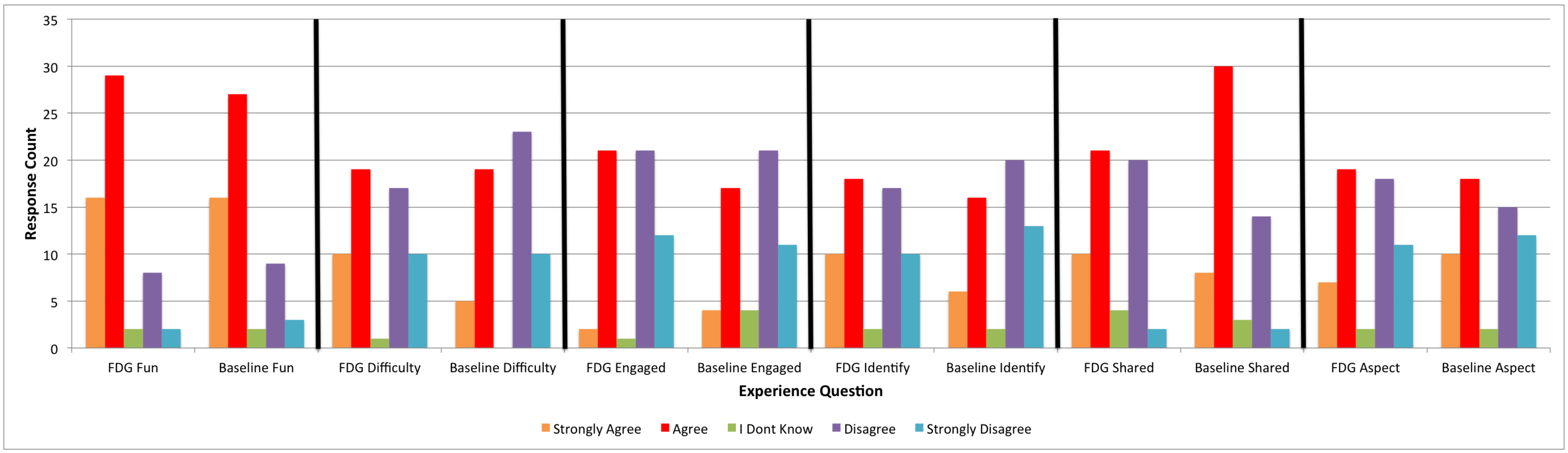


Figure 8.7: Histogram of Experience Questions

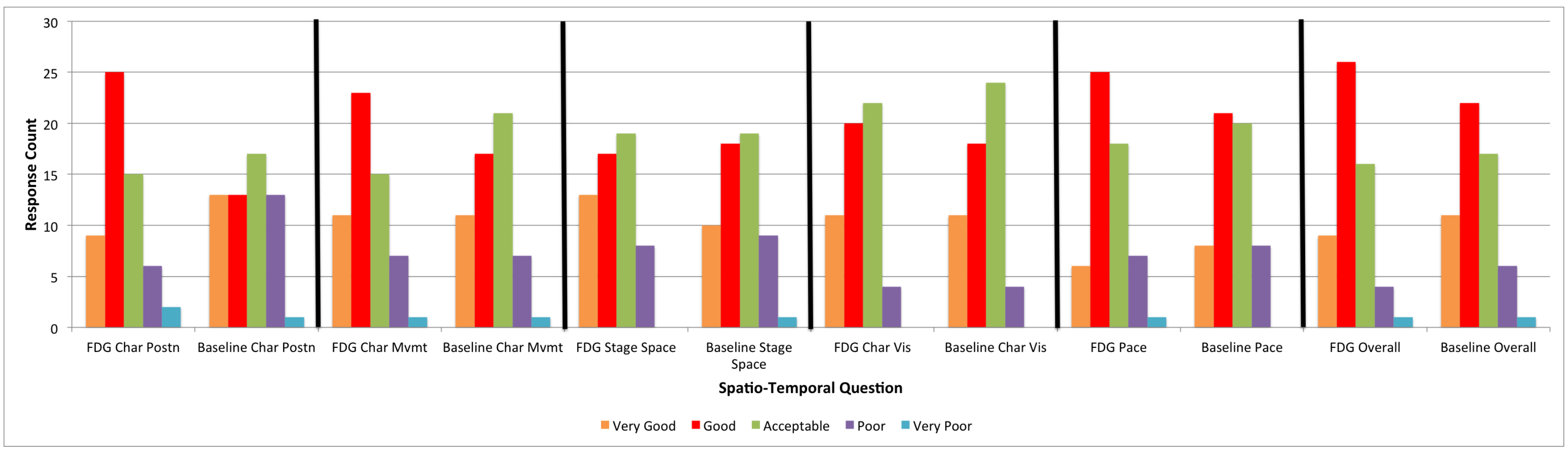


Figure 8.8: Histogram of Spatio-Temporal Questions

movement, but bad in that they were too planted." However, some also commented that the extra movement made it more stressful to complete the FDG scene over the Rules scene. Others commented on the overall flow, such as "This session seemed to flow better than the last, but the character interactions with the background characters of the scene were less engaging" in the Rules scene.

The questions might not have been appropriately worded, since verbal discussions exposed more information supporting our hypothesis than the questionnaire did. Some stated that the "inclusion" concept felt more like a question about being engaged, not about virtual character positioning for group inclusion.

8.3 Quantitative

We captured the locations of all the characters within the scene during each participant's performance. We then compared the FDG scene to the Rules scene to determine how close the AI characters were to the human-controlled character at each logged point in time. If one scene is statistically significantly different and shows the AI characters to be closer to the human-controlled character, then we can infer that the human-controlled character is more included in the scene.

This study included the same 57 participants that performed all three scenes, with a power to detect a medium effect (f=0.25) of 0.981. We completed a repeated measures ANOVA with a Greenhouse-Geisser correction, which compared the average distance between the human-controlled character (Miss Prism) and all other characters onstage. The test showed that this distance differed statistically significantly between sessions ($F(1.000, 56.000)=16.874$, $p<0.001$). This indicates that the AI-controlled characters were closer, on average, to the human-controlled character in the FDG scene than in the Rules scene, regardless of which came first. The descriptive statistics for this study can be seen in Table 8.1.

Table 8.1: Overall Average Distances of AI Characters from the Human-Controlled Character (Miss Prism)

Descriptive Statistics	Mean Distance	Standard Deviation	N
FDG Overall Average	21.46	4.095	57
Rules Overall Average	26.39	8.462	57

8.4 Interactive Conclusions

With our qualitative analysis, we found that there were no statistically significant differences in the perception of the two scenes (Rules and FDG), although the participants verbally stated that the FDG one felt like the character was more included in the scene. The quantitative analysis shows that, from a pure spatial perspective, the human-controlled character was closer to all of the AI-controlled characters in the FDG scene than in the Rules scene. This confirms our hypothesis that the FDG component would make a scene more inclusive spatially to a human-controlled character.

CHAPTER 9: CONCLUSIONS

Throughout this work, we have put together techniques for positioning characters in a virtual environment, which leverages play-scripts, natural language processing (NLP), Behavior Markup Language (BML), rules engines, and force-directed graphs (FDGs). We have also evaluated each component against our hypotheses using both quantitative and qualitative measures.

Our approach is based upon the ability to pre-block a play objectively, however real theatre blocking is based more upon chemistry and make-up of a cast. The overall arrangement of how the ensemble looks onstage is more important than being on the right mark or knowing ahead of time where to go. However, our approach brings us one step closer to being able to block a play in an automated fashion for virtual environments, and provide a reasonably good performance from the viewer's perspective.

9.1 Annotation Extraction

To begin, we needed to find a simple way of telling our AI director how to position the characters within the scene. We were inspired by theatre and their use of play-scripts to help actors perform a scene. We decided to utilize this format for instructing our AI director, and extracted movements from the annotations within the play-scripts. In doing this, we hypothesized that:

Hypothesis 1.1. *A computational algorithm using annotations in a play-script can provide similar positioning of virtual characters as a real actor directed by a human.*

To validate our hypothesis, we handmapped the "Graveyard" scene from *Hamlet*, Act V, Scene I, which was a famous Broadway performance from 1964. We calculated the time required to write the Behavior Markup Language (BML) for each of the movements within the 14 minute scene. We built a part of speech tagging and named entity recognition component to extract the movements from the annotations within the play-script.

We found we were able to reduce both the technical expertise and the time required to write the script by over four hours. We also measured the characters' positions throughout the scene for both the handmapped baseline scene and our NLP-generated scene. We found we were able to accomplish similar (~78% position match and ~34% gaze match) movements without requiring any technical BML expertise from the author. [91]

This confirms our Hypothesis 1.1, which states that we can provide similar positioning of virtual characters within a scene as a real actor.

9.2 Rules Application

In our pursuit of creating a director that can mimic a human actor, we realized that actors apply certain rules to their movements within the theatre and life. We identified four basic groups of rules:

1. Grouping Spatial Rules
2. Conversational Spatial Rules
3. Theatre Rules
4. General Rules

These rules incorporated things like distances between characters when they are speaking to each other, not upstaging/turning your back on the audience, and looking at what is being pointed at. With these rules, we hypothesized:

Hypothesis 1.2. *An algorithm-based director can improve character positioning of virtual characters within a scene if rules are applied to the movements defined in the annotations.*

We took our previous work, which extracted the movements from the annotations, and applied these new rules to the characters. We measured the resultant positioning of the characters versus the handmapped baseline again, and found that we increased our matching of both position and gaze by $\sim$11% and $\sim$20%, respectively. This brought us to an accuracy of 89.8% for position and 52.7% for gaze, just by applying these rules. [95]

This again confirms our Hypothesis 1.2 that states that adding rules will improve our positioning of characters, and more closely mimic the decisions made by actors in a performance. This work has focused on the theatre; however many of these rules are also applicable to other applications of spatial positioning, such as games and virtual worlds.

9.3 Implied Movement

We noticed that despite doing so well mimicing the positions of the characters in a scene, we were still missing approximately 11% of the movements. As we dug deeper, it looked that the actor took their own initiative to improvise additional movements, which were not annotated within the scene. With this knowledge, we wondered if there were some triggers in the actual speech of the performance that may have prompted the actor to add these extra movements. We conjectured that there may be something in either what the actors are saying, or something about the timing that may infer some movement in the scene.

We attempted to infer these movements based on the speech within the scene through several attributes, such as bag of words, number of lines before or after the current line, number of annotations between movements, word repetition, and punctuation counts. Even as we explored numerous learning methodologies (Max Entropy, Boosting, Random Forests, etc.), we were unable to learn any of these implied movements. We believe that due to the nature of Shakespeare's iambic pentameter and creation of new words, we were unable to extract additional movements from the speech within the "Graveyard" scene within *Hamlet*. Additional research should be pursued for other play-scripts, as well as other techniques including parse trees. [93]

For this conjecture, we were not able to find any supporting evidence to infer any movement in the scene, although we believe additional work may still uncover results here.

9.4 Applied Forces

Next, we attempted to better incorporate the semi-circular arrangement of characters within a scene and reduce occlusion of characters. To do this, we introduced some force-directed graph visualization algorithms, based on Fruchterman and Reingold's algorithm. The graph consisted of nodes for the characters, audience points, pawns, marks, and centralizing points to arrange the characters. With these algorithms, we hypothesized:

Hypothesis 1.3. *Force-directed graphs can position characters onstage with typical conversational arrangements, avoiding character occlusion.*

We then measured the distances between the characters on the stage and their related targets, audience, and other characters. We found the characters maintained an approximately three foot distance, which is supported by Sundstrom's research for comfortable conversational space [88]. We also measured the overall clustering of the characters and occlusion. We found that these FDG-based algorithms provided similar clustering on the stage as the handmapped baseline performance, but provided 1% less occlusion of characters from the audience's viewpoint. We also confirmed the spatial arrangement tendencies formed a semi-circle for the characters, facing the audience.

This Hypothesis 1.3 was confirmed to show that we can position characters with typical conversational arrangements and reduce overall character occlusion from even the human-performed scene.

9.5 Human-Controlled Characters

When humans are involved in a scene, they do not always follow the play-script as they should. They may miss their cue or their mark. When this happens, the character usually becomes very visibly "incorrect" from the audience perspective. Actors tend to adjust their performance to compensate, so we hypothesized:

Hypothesis 1.4. *Force-directed graphs can better incorporate human-controlled characters with a set of virtual characters, adjusting the virtual character movements around the human's motion, than a performance done only with the play-script and applied rules.*

We believe that the forces defined for Hypothesis 1.3 could be leveraged to adjust the AI characters when the human-controlled character does the wrong thing, but still maintain the integrity of the play-script. To assess this, we introduced a "human-controlled character" that would randomly do the wrong thing $x\%$ of the time. We then measured the character clustering and occlusion to confirm we were able to include the human character's incorrect movement within the scene. We compared these measurements against the same measurements for when the AI characters performed the scene exactly as the handmapped play-script indicated. We found that as the human-controlled character's accuracy decreased, the occlusion increased by 1% for the handmapped scene, but decreased by 1% for the FDG-controlled scene. We also found that the clustering of the characters was 10% closer for the FDG-controlled scene, since the characters were shifted to compensate for the human-controlled character.

We then performed a qualitative analysis where we allowed participants to perform the scenes themselves, both with and without the FDG-control component. We wanted to verify whether the forces would allow for a more spatially-inclusive arrangement of characters, even if the human-controlled character did not do the right thing. Statistically, we found that both performances were not different as far as spatio-temporal aspects, performance experience, or perceived workload. Verbally, participants confirmed that the one which adjusted based on their movements (the FDG performance) made the character appear more included in the scene than the scene without the FDG-component. We also found quantitatively that the characters were closer to the human-controlled character in the FDG-controlled scene than in the rules-only scene.

This data supports our Hypothesis 1.4, which states that the FDG-component can better incorporate the human-controlled characters than just utilizing the play-script and rules.

9.6 Overall Quality and Generalization

Next, we needed to show that these techniques were generalizable to other playscripts, not just *Hamlet*. We defined five different spatio-temporal dimensions to categorize play-scripts: Speed, Number of Characters, Space, Audience, and Dynamics. These dimensions were reduced down to 54 combinations, with a constraint on the audience due to recording availability. We found eight play-scripts that covered 100% of our dimensional criteria, and 71% of our pairwise dimensional space. We hypothesized that:

Hypothesis 1.5. *An algorithm-based director, using a combination of play-scripts, rules, and force-directed graphs, can equal or surpass the human-perceived threshold of a quality performance for a variety of spatio-temporal play types.*

We then performed several qualitative assessments of these scenes, comparing a handmapped human-performed version to our NLP+Rules+FDG-generated version. With each of the six new scenes, we found that the two scenes were either not statistically significantly different, or our NLP+Rules+FDG-generated video was perceived as better, from a spatio-temporal perspective. Only a couple of scenarios showed one or two questions stating the handmapped version was better perceived, as seen

Table 9.1: Statistically Significant Differences Between Baseline and FDG Videos (Empty Cells Mean No Statistically Significant Differences Found)

Play Title	Char Postn	Char Mvmt	Stage Space	Char Vis	Pace	Overall
The Importance of Being Earnest				Baseline		
Death of a Salesman	FDG		FDG	Baseline		
Krapp's Last Tape				FDG		
Noises Off	FDG	FDG				
Tartuffe	FDG		Baseline	FDG		
The Cherry Orchard		Baseline				
Hamlet						

in Table 9.1. Quantitative comparisons showed a 58% match in position across all scenes, with most discrepancies being due to slight grid position shifts from the FDG arrangements.

This allows us to confirm Hypothesis 1.5 as true, with the constraint of a proscenium-style theatre/audience configuration.

9.7 Summary

In summary, our system has been shown to reduce the amount of time to author character positioning within a scene. It is able to match a real production's blocking 89.8% of the time, and incorporate human-controlled characters within the scene more consistently, regardless of their accuracy. We also see that despite our system's blocking not being exact for 10.2% of the scene, it does not appear to degrade a viewer's experience of the scene. This shows that perhaps an exact match is not required for a realistic performance for the viewer, but can be explicitly written into a play-script if required. Our engine is generalizable across 71% of the pairwise combinations of our scene dimensions, and incorporates a human-controlled character well, even if they do not follow the play-script. We were also able to confirm all five of our hypotheses formed when starting this research, and provided a key dynamic spatio-temporal tool through our NLP+Rules+FDG engine. This tool allows designers to utilize natural language formatted as a play-script and arrange the characters on a stage dynamically, adjusting for any human-controlled characters.

CHAPTER 10: FUTURE WORK

This work will be useful in assisting directors, game writers, and other virtual environment authors with placing virtual characters within their environment, whether it is a stage, or a more general virtual environment. It is also complementary to dialogue trees, which are often used with dynamic speech in a scene. Future work will pursue the use of these techniques in combination with dialogue trees for play-script-like branches to control AI characters.

There are many other areas of expansion for this work, which we look to complete in the future. For instance, this work does not apply the optimizations of audience seating visibility (similar to multiple camera angles in television, movies, and games) at this time, but could be considered for future work. Another great expansion would be to pursue the adjustments to our force-directed graphs to support multiple audience viewpoints. Also, we would like to pursue the use of these techniques as a control mechanism within a virtual reality environment.

Some interesting force-related expansions that will be considered include time-based attraction degradation for joint entrances (where two or more characters enter at the same time and have stronger forces pulling them together), target points (stronger forces initially, but degrades over time), and importance to the scene (stronger forces pulling the priority characters towards the audience). Also, incorporating an exception for certain types of AI characters that should not have forces applied, such as larger groups. Plus, a deeper dive into the potential for inferring additional movement based on the play-script is on our radar.

Lastly, some deeper perusal into longer scenes, both for exploration of forces over longer periods of time, as well as to overcome the "newness" effect that was found with the interactive study. Additional modularization of the provided research code will be provided as well.

REFERENCES

[1] Abigail-Nicole. How to Format a Stage Play–Script Frenzy. http://www.scriptfrenzy.org/howtoformatastageplay, July 2012.

[2] P. Aggarwal and D. Traum. The BML Sequencer: A Tool for Authoring Multi-character Animations. *Intelligent Virtual Agents*, pages 428–430, 2011.

[3] M. Alderson. Theatre Types. http://www.ia470.com/primer/theatres.htm, July 2012.

[4] G. H. S. Association. One Act Play - Judge's Evaluation Sheet. http://www.ghsa.net/sites/default/files/documents/OAP%20Evaluation%20Sheet.pdf, 2014.

[5] K. Baker. Stage Movement and Acting Rules. http://campuses.fortbendisd.com/campuses/documents/teacher/2010/teacher_20100525_1349.pdf, Jan. 2002.

[6] M. J. Bannister, D. Eppstein, M. T. Goodrich, and L. Trott. Force-Directed Graph Drawing Using Social Gravity and Scaling. *CoRR*, abs/1209.0748, 2012.

[7] S. Bml. Wiki - BML 1.0 Standard - Mindmakers. http://www.mindmakers.org/projects/bml-1-0/wiki/Wiki, July 2012.

[8] M. N. Borazjany, L. Yu, Y. Lei, R. Kacker, and R. Kuhn. Combinatorial testing of ACTS: A case study. *Proceedings - IEEE 5th International Conference on Software Testing, Verification and Validation, ICST 2012*, pages 591–600, 2012.

[9] A. G. Brooks. *Coordinating Human-Robot Communication*. PhD thesis, MIT, Jan. 2006.

[10] A. Cao, K. Chintamani, A. K Pandya, and R. Ellis. NASA TLX: Software for assessing subjective mental workload. *Behavior research methods*, 41:113–7, 03 2009.

[11] T. Causey. Where is the best place to sit in the theater. http://theater.about.com/od/faqs/f/faqbestseat.htm, July 2012.

[12] M. Cavazza and F. Charles. Agents' interaction in virtual storytelling. *Intelligent Virtual Agents*, pages 1–15, 2001.

[13] M.-W. Chang, L. Ratinov, D. Roth, and V. Srikumar. Importance of Semantic Representation: Dataless Classification. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 830–835. AAAI Press, 2008.

[14] W. T. Chuang and J. Yang. Text Summarization by Sentence Segment Extraction Using Machine Learning Algorithms. *Knowledge Discovery and Data Mining. Current Issues and New Applications*, 2000.

[15] B. Colleran, J. Gielgud, W. Shakespeare, R. Burton, H. Cronyn, A. Drake, and E. Herlie. *Hamlet*, Electronovision, Inc., 1964.

[16] M. Collins and N. Duffy. Convolution Kernels for Natural Language. In *Advances in Neural Information Processing Systems 14*, pages 625–632. MIT Press, 2001.

[17] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537, Aug 2011.

[18] K. R. Coventry and S. C. Garrod. *Saying, Seeing and Acting: The Psychological Semantics of Spatial Prepositions (Essays in Cognitive Psychology)*. The Psychological Semantics of Spatial Prepositions. Psychology Press, 1 edition, Mar. 2004.

[19] B. Coyne and R. Sproat. WordsEye: An Automatic Text-to-Scene Conversion System. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 487–496, New York, NY, USA, 2001. ACM.

[20] A. Culotta and J. Sorensen. Dependency Tree Kernels for Relation Extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[21] P. Debevec. The Light Stages and Their Applications to Photoreal Digital Actors. In *SIGGRAPH Asia*, Singapore, Nov. 2012.

[22] W. Despain. *Writing for Video Game Genres: From FPS to RPG*. CRC Press, 2009.

[23] J. Dias and A. Paiva. Feeling and Reasoning: a Computational Model. *EPIA 2005*, pages 127–140, 2005.

[24] J. Dias and A. Paiva. Agents with Emotional Intelligence for Storytelling. *Affective Computing and Intelligent Interaction*, pages 77–86, 2011.

[25] J. Dzifcak, M. Scheutz, C. Baral, and P. Schermerhorn. What to do and How to do it: Translating Natural Language Directives into Temporal and Dynamic Logic Representation for Goal Management and Action Execution. *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, pages 3768–3773, 2009.

[26] A. W. Feng, Y. Xu, and A. Shapiro. An Example-Based Motion Synthesis Technique for Locomotion and Object Manipulation. *I3D 2012*, 2012.

[27] G. Forman. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *J. Mach. Learn. Res.*, 3:1289–1305, Mar. 2003.

[28] A. U. Frank. Qualitative Spatial Reasoning: Cardinal Directions as an Example. *International Journal of Geographical Information*, 10(February 2012):269–290, 1996.

[29] T. M. J. Fruchterman, Edward, and E. M. Reingold. Graph Drawing by Force-Directed Placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[30] J. Furnkranz. A Study Using n-gram Features for Text Categorization, Nov. 1998.

[31] P. Gajer, M. T. Goodrich, and S. G. Kobourov. A Fast Multi-Dimensional Algorithm for Drawing Large Graphs. *8th Symposium on Graph Drawing (GD)*, pages 211–221, 2000.

[32] R. Games. Rockstar Games Presents: L.A. Noire. https://www.rockstargames.com/lanoire/, 2012.

[33] R. Hadany and D. Harel. A Multi-Scale Algorithm for Drawing Graphs Nicely. In *Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science*, WG '99, pages 262–277, London, UK, UK, 1999. Springer-Verlag.

[34] E. T. Hall, R. L. Birdwhistell, B. Bock, P. Bohannan, J. Diebold, A. Richard, M. Durbin, M. S. Edmonson, J. L. Fischer, D. Hymes, S. T. Kimball, W. L. Barre, S. J. Lynch, Frank, J. E. McClellan, D. S. Marshall, G. B. Milner, H. B. Sarles, G. L. Trager, and A. P. Vayda. Proxemics [and Comments and Replies]. *Current Anthropology*, 9(2/3):pp. 83–108, 1968.

[35] A. Hartholt, D. Traum, S. C. Marsella, A. Shapiro, G. Stratou, A. Leuski, L.-P. Morency, and J. Gratch. All Together Now: Introducing the Virtual Human Toolkit. In *13th International Conference on Intelligent Virtual Agents*, Edinburgh, UK, Aug. 2013.

[36] International Computer Science Institute and T. Regier. The Acquisition of Lexical Semantics for Spatial Terms: A Connectionist Model of Perceptual Categorization, 1992.

[37] D. Jan and D. R. Traum. Dynamic Movement and Positioning of Embodied Agents in Multiparty Conversations. In *Proceedings of the Workshop on Embodied Language Processing*, Embodied NLP '07, pages 59–66, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[38] T. P. Jurka, L. Collingwood, A. E. Boydstun, and E. Grossman. RTextTools: A Supervised Learning Package for Text Classification, 2011.

[39] B. D. Kabialis. When Shakespeare Dependence Hits Point of Diminishing Returns. http://www.berkeleybeacon.com/arts-and-entertainment/2012/2/2/when-shakespeare-dependence-hits-point-of-diminishing-returns, 2012.

[40] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letter*, 31(1):7–15, Apr. 1989.

[41] J. D. Kelleher and F. J. Costello. Applying Computational Models of Spatial Prepositions to Visually Situated Dialog. *Multiple values selected*, 35(2):271–306, Jan. 2012.

[42] A. Kendon. *Conducting Interaction: Patterns of Behavior in Focused Encounters*. Studies in Interactional Sociolinguistis. Cambridge University Press, Nov. 1990.

[43] P. Kenny, A. Hartholt, J. Gratch, W. Swartout, D. Traum, S. Marsella, and D. Piepol. Building Interactive Virtual Humans for Training Environments. *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, 2007(-1), 2007.

[44] H. Kobayashi and C. Rinnert. Effects of First Language on Second Language Writing: Translation versus Direct Composition*. *Language Learning*, 42(2):183–209, 1992.

[45] S. G. Kobourov. Spring Embedders and Force Directed Graph Drawing Algorithms. *CoRR*, abs/1201.3011, 2012.

[46] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward Understanding Natural Language Directions. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*, HRI '10, pages 259–266, Piscataway, NJ, USA, 2010. IEEE Press.

[47] M. Kriegel, R. Aylett, J. Dias, and A. Paiva. An authoring tool for an emergent narrative storytelling system. *AAAI Fall Symposium on Intelligent Narrative Technologies, Technical Report FS-07-05*, pages 55–62, 2007.

[48] D. R. Kuhn, R. N. Kacker, D. R. Kuhn, and R. N. Kacker. NISTIR 7878 Combinatorial Coverage Measurement NISTIR 7878 Combinatorial Coverage Measurement. *Proceedings of the IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops (ICSTW 2013)*, 2009.

[49] D. R. Kuhn, R. N. Kacker, and Y. Lei. SP 800-142. Practical Combinatorial Testing. Technical report, National Institute of Standards & Technology, Gaithersburg, MD, United States, 2010.

[50] B. Landau and R. Jackendoff. "What" and "Where" in Spatial Language and Spatial Cognition. *Behavioral and Brain Sciences*, 16(02):217–238, 1993.

[51] J. Lee and S. Marsella. Nonverbal behavior generator for embodied conversational agents. *Intelligent Virtual Agents*, pages 243–255, 2006.

[52] A. Leuski, J. Pair, D. Traum, P. J. McNerney, P. Georgiou, and R. Patel. How to talk to a hologram. *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 360–362, 2006.

[53] S. C. Levinson. Frames of Reference and Molyneux's Question: Crosslinguistic Evidence. *Language and Space*, pages pp. 109–169, 1996.

[54] C. Liao, S. Alpha, and P. Dixon. Feature Preparation in Text Categorization, 2002.

[55] P. B. Lile. Drama Stages and Stage Movements. http://www.graves.k12.ky.us/schools/gcms/plile/Drama%20Stages%20and%20Stage%20Movements.ppt, 2012.

[56] H. Liu and P. Singh. Commonsense Reasoning in and over Natural Language. In *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES-2004*, pages 293–306. Springer, 2004.

[57] G. D. Logan and D. D. Sadler. A Computational Analysis of the Apprehension of Spatial Relations. *Language and Space*, pages pp. 493–529, 1996.

[58] E. Loper and S. Bird. NLTK: The natural language toolkit. *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, 1:63–70, 2002.

[59] D. V. Lu and W. D. Smart. Human-Robot Interactions as Theatre. In *RO-MAN 2011*, pages 473–478. IEEE, 2011.

[60] A. Mabillard. The Chronology of Shakespeare's Plays. http://www.shakespeare-online.com/keydates/playchron.html, 2000.

[61] A. Mabillard. Shakespearean Sonnet Basics: Iambic Pentameter and the English Sonnet Style. http://www.shakespeare-online.com/sonnets/sonnetstyle.html, 2012.

[62] C. Manning. Force Layout. https://github.com/mbostock/d3/wiki/Force-Layout, 2013.

[63] A. Marshall, H. Vilhjalmsson, S. Kopp, M. Kipp, M. Krieger, M. Wissner, P. Tepper, J. Homer, H. V. Welbergen, A. Hill, T. Bickmore, and J. Gruber. Behavior Markup Language (BML) Version 1.0 (Proposal). http://www.mindmakers.org/projects/saiba/wiki/Wiki/, 2011.

[64] M. Mateas and A. Stern. Integrating Plot , Character and Natural Language Processing in the Interactive Drama Faccade. *Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE-03)*, 2, 2003.

[65] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to Parse Natural Language Commands to a Robot Control System. In *Proceedings of the 13th International Symposium on Experimental Robotics (ISER)*, June 2012.

[66] G. A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[67] A. Modeling. Anchor Modeling | An agile modeling technique for evolving information. http://www.anchormodeling.com/, 2013.

[68] L. Muñoz. Theatre Judges Packet: One-Act Play Contest Questionnaire. http://www.uiltexas.org/files/academics/theatre/Theatre-Judges-Packet.pdf, 2014.

[69] NEF (the New Economics Foundation). Capturing the Audience Experience: A Handbook for the Theatre. http://itc-arts-s3.studiocoucou.com/uploads/helpsheet_attachment/file/23/Theatre_handbook.pdf, 2010.

[70] C. Ouzouni and K. Nakakis. Measurement Tool. *NIST IR 7878*, 4(May):222–231, 2009.

[71] P. Pavis. Theatre Analysis: Some Questions and a Questionnaire. *New Theatre Quarterly*, 1:208–212, 5 1985.

[72] J. C. Platt, N. Cristianini, and J. Shawe-Taylor. Large Margin DAGs for Multiclass Classification. *Advances in Neural Information Processing Systems*, 12(3):547–553, 2000.

[73] J. Primrose. Theatrecrafts - Entertainment Technology Resources - Home Page. http://www.theatrecrafts.com/, July 2012.

[74] A. Quigley. Large Scale Force Directed Layout (Spring Algorithm): For Information Visualization: Clustering and Abstraction. http://rp-www.cs.usyd.edu.au/~aquigley/3dfade/, 2003.

[75] T. Regier and L. A. Carlson. Grounding Spatial Language in Perception: An Empirical and Computational Investigation. *Journal of Experimental Psychology: General*, 130:273–298, 2001.

[76] T. Ribeiro, M. Vala, and A. Paiva. Thalamus: Closing the Mind-Body Loop in Interactive Embodied Characters. In Y. Nakano, M. Neff, A. Paiva, and M. Walker, editors, *Intelligent Virtual Agents*, pages 189–195. Springer, 2012.

[77] C. Rich and C. Sidner. Using Collaborative Discourse Theory to Partially Automate Dialogue Tree Authoring. In Y. Nakano, M. Neff, A. Paiva, and M. Walker, editors, *Intelligent Virtual Agents*, volume 7502 of *Lecture Notes in Computer Science*, pages 327–340. Springer Berlin Heidelberg, 2012.

[78] R. E. Schapire and Y. Singer. BoosTexter: A Boosting-Based System for Text Categorization. *Machine Learning*, 39(2):135–168, 2000.

[79] W. Shakespeare. *Hamlet*. Horace Howard Furness, 1905.

[80] R. Shannon, A. Quigley, H. Australia, and P. Nixon. Graphemes: Self-Organizing Shape-Based Clustered Structures for Network Visualisations. In *Proceedings of the 28th of the International Conference Extended Abstracts on Human Factors in Computing Systems*, pages 4195–4200, 2010.

[81] D. Shen, G. J. M. Kruijff, and D. Klakow. Studying Feature Generation from Various Data Representations for Answer Extraction. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP*, pages 65–72, 2005.

[82] M. Si, S. C. Marsella, and D. V. Pynadath. Thespian: Using multi-agent fitting to craft interactive drama. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 21–28. ACM, 2005.

[83] M. Si, S. C. S. Marsella, D. V. Pynadath, and M. Rey. Evaluating Directorial Control in a Character-Centric Interactive Narrative Framework. *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 1289–1296, 2010.

[84] T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer. ROCR: Visualizing Classifier Performance in R. *Bioinformatics*, 21(20):3940–3941, 2005.

[85] R. Sommer. The Distance for Comfortable Conversation: A Further Study. *Sociometry*, 25(1):111–116, 1962.

[86] R. L. Sterne. *John Gielgud Directs Richard Burton in Hamlet by Richard L. Sterne*. Random House, 5th edition, 1967.

[87] J. Sun, M. Zhang, and C. L. Tan. Tree Sequence Kernel for Natural Language. In *AAAI Conference on Artificial Intelligence*, 2011.

[88] E. Sundstrom and I. Altman. Interpersonal Relationships and Personal Space: Research Review and Theoretical Model. *Human Ecology*, 4(1):47–67, 1976.

[89] C. Talbot. Virtual Companions and Friends. In *Proceedings of the 49th Annual Southeast Regional Conference 2011*, pages 356–357, New York, NY, USA, 2011. ACM.

[90] C. Talbot. Creating an Artificially Intelligent Director (AID) for Theatre and Virtual Environments. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '13, pages 1457–1458, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[91] C. Talbot and G. M. Youngblood. Spatial Cues in Hamlet. In *Proceedings of the 12th International Conference on Intelligent Virtual Agents*, IVA '12, pages 252–259, Berlin, Heidelberg, 2012. Springer-Verlag.

[92] C. Talbot and G. M. Youngblood. Application of Force-Directed Graphs on Character Positioning. In *Proceedings of the Spatial Computing Workshop (SCW 2013) collocated with AAMAS (W09)*, pages 53–58. IFAMAAS (International Foundation for Autonomous Agents and Multiagent Systems), 2013.

[93] C. Talbot and G. M. Youngblood. Lack of Spatial Indicators in Hamlet. In *Florida Artificial Intelligence Research Society Conference*, FLAIRS '13, pages 154–159. Association for the Advancement of Artificial Intelligence, 2013.

[94] C. Talbot and G. M. Youngblood. Positioning Characters Using Forces. In *Proceedings of the Cognitive Agents for Virtual Environments Workshop (CAVE 2013) collocated with AAMAS (W08)*. IFAMAAS (International Foundation for Autonomous Agents and Multiagent Systems), 2013.

[95] C. Talbot and G. M. Youngblood. Shakespearean Spatial Rules. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '13, pages 587–594, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[96] C. Talbot and G. M. Youngblood. Scene Blocking Utilizing Forces. In *Florida Artificial Intelligence Research Society Conference*, FLAIRS '14, pages 91–96. Association for the Advancement of Artificial Intelligence, 2014.

[97] C. Talbot and G. M. Youngblood. Virtual Environment Positioning Utilizing Play-Script Spatiotemporal Reasoning. *IEEE Transactions on Games*, 2018.

[98] S. Tellex and Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences. *Natural language and spatial reasoning.* PhD thesis, Massachusetts Institute of Technology, 2010.

[99] M. Theune, S. Faas, and D. Heylen. The virtual storyteller: Story creation by intelligent agents. *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment TIDSE Conference (2003)*, 2003.

[100] M. Thiebaux, M. Rey, A. N. A. N. Marshall, S. Marsella, and M. Kallmann. SmartBody : Behavior Realization for Embodied Conversational Agents. *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 179(AAMAS):151–158, 2008.

[101] K. R. Thorisson and H. H. Vilhjalmsson. Functional Description of Multimodal Acts: A Proposal. *AAMAS 2009 Workshop Towards a Standard Markup Language for Embodied Dialogue Acts*, pages 1–4, Mar. 2009.

[102] Z. Tomaszewski. On the Use of Reincorporation in Interactive Drama. *Workshops at the Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.

[103] D. Traum, P. Aggarwal, R. Artstein, S. Foutz, J. Gerten, A. Katsamanis, A. Leuski, D. Noren, and W. Swartout. Ada and Grace: Direct Interaction with Museum Visitors. In Y. Nakano, M. Neff, A. Paiva, and M. Walker, editors, *Intelligent Virtual Agents*, pages 245–251. Springer, 2012.

[104] N. Trivedi, P. Langley, P. Schermerhorn, and M. Scheutz. Communicating, Interpreting, and Executing High-Level Instructions for Human-Robot Interaction. In *Proceedings of the 2011 AAAI Fall Symposium on Advances in Cognitive Systems*, Arlington, VA, November 2011.

[105] W. T. Tutte. How to Draw a Graph. *Proceedings London Mathematical Society*, 13(3):743–768, 1963.

[106] Unknown. AVATAR. http://avatarblog.typepad.com/avatar-blog/2010/05/behind-the-scenes-look-at-the-motion-capture-technology-used-in-avatar.html, 2012.

[107] Unknown. GA Judging. https://www.ghsa.net/sites/default/files/documents/forms/OAP-Judges-Evaluation-Sheet.pdf, 2012.

[108] Unknown. Force-directed graph drawing. *Wikipedia, the free encyclopedia*, jan 2013. Page Version ID: 535828764.

[109] H. van Welbergen, Y. Xu, M. Thiebaux, W. W. Feng, J. Fu, D. Reidsma, and A. Shapiro. Demonstrating and testing the BML compliance of BML realizers. *Intelligent Virtual Agents*, pages 269–281, 2011.

[110] E. Vidal Jr and A. Nareyek. A Real-Time Concurrent Planning and Execution Framework for Automated Story Planning for Games. *Workshops at the Seventh Artificial Intelligence ...*, 2011.

[111] H. Vilhjalmsson, N. Cantelmo, J. Cassell, N. E. Chafai, M. Kipp, S. Kopp, M. Mancini, S. Marsella, A. Marshall, C. Pelachaud, and Others. The Behavior Markup Language: Recent Developments and Challenges. *Intelligent Virtual Agents*, pages 99–111, 2007.

[112] Y. Wei, E. Brunskill, T. Kollar, and N. Roy. Where to Go: Interpreting Natural Directions Using Global Inference. *IEEE International Conference on Robotics and Automation*, pages 1–7, Mar. 2009.

[113] H. Welbergen, D. Reidsma, Z. o. f. M. Ruttkay, and J. Zwiers. Elckerlyc. *Journal on Multimodal User Interfaces*, 3(4):271–284, 2009.

[114] D. Zelenko, C. Aone, and A. Richardella. Kernel Methods for Relation Extraction. *The Journal of Machine Learning Research*, 3:1083–1106, 2003.

[115] M. Zhang, G. D. Zhou, and A. Aw. Exploring Syntactic Structured Features over Parse Trees for Relation Extraction using Kernel Methods. *Information Processing & Management*, 44(2):687–701, 2008.

APPENDIX A: LACK OF SPATIAL INDICATORS IN *HAMLET*

A.1 Introduction

When actors perform on stage, they are provided with specific directions on where and how to perform their lines. The director provides these directions via a play-script's annotations. Beyond these annotations, the actors are provided some freedom in performing their lines, although certain guidelines for theatre acting are always in play. Intuition and characterization help the actor to identify other movements that are in-character and appropriate in the different parts of the play for their character.

We look to realistically capture the spatial movements of actors on stage, so we started by translating the spatial movements found within the annotations from the director, as can be seen in our prior work [91]. Basic parts of speech (POS), sentence structure parsing, and entity recognition provided us with key movements detailed from the annotations in the play-script with about 78% accuracy for character positions.

Next, we targeted the basic rules and guidelines that actors and directors use to control movement on the stage [95]. These included conversational space, group space, theatre rules, and general common-sense rules. This got us to 89% accuracy for position and 53% accuracy for gaze. After capturing these movements, there were still some movements in the play that the actors performed, but were not captured by the annotations and rules we encoded. One good example is in Act V in *Hamlet* where the gravedigger walks towards the audience, then turns around and walks back towards the grave. These are the kinds of movements that the actor decides upon based on their intuition.

Therefore, we thought about what might help a system to learn these sort of movements by the actors. We came to the hypothesis that perhaps what the actor is saying could imply certain types of movement. Now, these are not the same kinds of movements as one actor telling the other actor to do something, but more of an implied movement, such as moving towards the audience for a monologue, gesturing to help emphasize what they are saying, or even a movement to keep the audience's attention during a rather long scene that has little to no movement involved with it. We are not focused on what is explicitly stated in the language, but more on the hidden movement that is likely to be performed by the actor on stage.

The context of the speech and the characters were identified as two key components to interpreting the implied movement, in addition to what the character is saying. We pursued several existing natural language processing and machine learning approaches to learn these implied movements within one particular play, *Hamlet*, as produced by Sir John Gielgud in 1964 on Broadway [15]. We utilized the script as written by Shakespeare [79], as well as the Electronovision video [15] of Richard Burton in Sir John Gielgud's production of *Hamlet* as our baseline. Each line's related movement was captured for the play and categorized into a standardized set of motions, such as walking, jumping, fighting, and so forth. We fed this information into machine learning algorithms, such as Maximum Entropy (MaxEnt) and Support Vector Machines (SVM), to help learn about the implied movements within the play. Our intent is to be able to identify that a movement should occur because of the speech being said, as well as specifically what type of motion for the speaker to perform.

A.2 Background and Related Work

Naturally, while pursuing an appropriate approach for our work, we started with the natural language processing that is used for giving directions to robots. This incorporates both natural language and spatial reasoning. However, the key difference with what we were looking to do is that we were not trying to give explicit directions for someone to do something. We want to understand the hidden movement. So looking at work like Wei et al.'s[112], Brooks'[9], and Kollar et al.'s [46] only provided techniques that assumed a set of predefined keywords, phrases, or corpus to be extracted and utilized for further processing. These focused on the meaning of different prepositions in order to interpret a spatial location.

Next, we looked into text categorization and summarization. The main focus of most text categorization is around known keywords or phrases to identify if the text contains that concept. The more similar the strings or synonyms are, the more similar they are considered to the entity being matched. The summarization techniques, like those used in Chuang and Yang's[14] paper, focus on segmentation of the text and the extraction of important sentence segments via machine learning on a feature vector. This is closer to what we want to do, but still is based on keywords and phrases, with little to no implied meaning involved.

A main exception to the patterns of text classification was with the data-less categorization done by Chang et al.[13]. They focused on the semantic meaning of the category to determine how to classify text without labeling and training the system. Also, classifying text into multiple categories is still not completely solved, as discussed in Platt, Cristianini, and Shawe-Taylor's[72]. This is key as we look at our data where one line can imply more than one motion. Some researchers, such as Schapire and Singer [78], have pursued multiple class classifications by using Boosting and text classification where you do not turn the problem into multiple binary classification problems, as is typical for this problem.

Other work with ConceptNet [56] also is closer extracting the meaning of words; however is still very similar to a synonym retriever. Similarly, relation extraction utilizes phrases and parse-trees for determining relationships between entities (again pre-defined entities and relationships), such as Culotta and Sorensen's[20], Zhang, Zhou, and Aw's[115], and Sun, Zhang, and Tan's [87] papers. Here we start to get to the capturing of features, especially contextual or sequential types of features. Others have pursued the use of tree kernels to help with machine learning on text, such as Collins and Duffy's[16] and Shen, Kruijff, and Klakow's[81] papers. Each of these papers discuss the use of tree kernels to try to better capture a parse-tree and its dependencies for use in machine learning. This is important with the type of natural language classification we are planning to do, since we hypothesize that the context of the words is just as important, if not more so, than the words themselves.

Since most traditional learning machine learning algorithms rely on feature-based representations of objects, we explored the different types of features that could be used to learn classifications within natural language. Liao [54] describes features as being local or global. They can be as simple as a single token, a phrase, or something much more complex. Selecting useful and relevant features, as well as deciding how to encode them, can greatly impact the machine learning algorithm's ability to learn good models [58]. Therefore, a lot of time is spent on identifying appropriate features, and many people start with everything they can think of. However most of these end up being local representations of the objects [114], such as just the words themselves.

Ultimately, we are transforming a document from one set of tokens to another, which is prone to loss of information, such as word sequence. Collobert et al.[17] discusses common feature categories, such as parts of speech (POS), voice of the sentence, and stemmed root words, while Culotta and Sorensen[20] mention word n-grams, capitalization, and conjunctions (or merging) of many features. Furnkranz[30] found that using n-grams of length two or three improved classification over longer n-grams. Forman[27] suggests the removal of common words (stop words), removal of rare words, and the use of booleans instead of counts for bag of words features. None discuss the appropriateness of features that represent spatial information, such as character positions. Kernels have also been utilized in place of traditional feature vectors, but were not pursued in our work at this time.

A.3 Approach

In order to have a baseline to train against, we took the Electronovision video [15] of the production of *Hamlet* on Broadway in 1964 and mapped all the movement of the characters for each line of the play-script [79]. We kept the "sentences" as the way Shakespeare originally divided up his lines of text. Shakespeare nearly always wrote in iambic pentameter (ten syllables per line, with alternating unstressed and stressed syllables) [61]. This meant that a speech like:

> Last night of all,
> When yond same star that's westward from the pole
> Had made his course to illume that part of heaven
> Where now it burns, Marcellus and myself,
> The bell then beating one,– [79]

was broken up into five sentences. An alternate approach could have been used where each real sentence was used to determine implied movement or not. This may have helped with the training ratio for movement versus no movement, which will be discussed further in the Experimentation section. However, we chose the phrase-approach because of the frequency of the change in actions being performed within the play. By splitting the sentences to this size, we had a more consistent line-length, were able to more precisely capture a single phrase that might imply movements, and could capture more movements than we could with full sentences.

The main two challenges with mapping this three hour play were in carefully identifying only one movement per line, as well as accurately capturing all the desired movements throughout such a long play, with standardized movement names as seen in Figure A.1. Many lines involved multiple movements. To keep things simple, we decided to capture the biggest movement performed by the speaker whenever there were more than one movement for the line. The longest line in *Hamlet* was only fifteen words long, with the average being seven words in length. We also wanted to capture the locations of each character on stage to see if this would help in identifying when a movement would occur, (more from learning a rules-based approach); however we were unable to capture that level of detail due to time constraints.

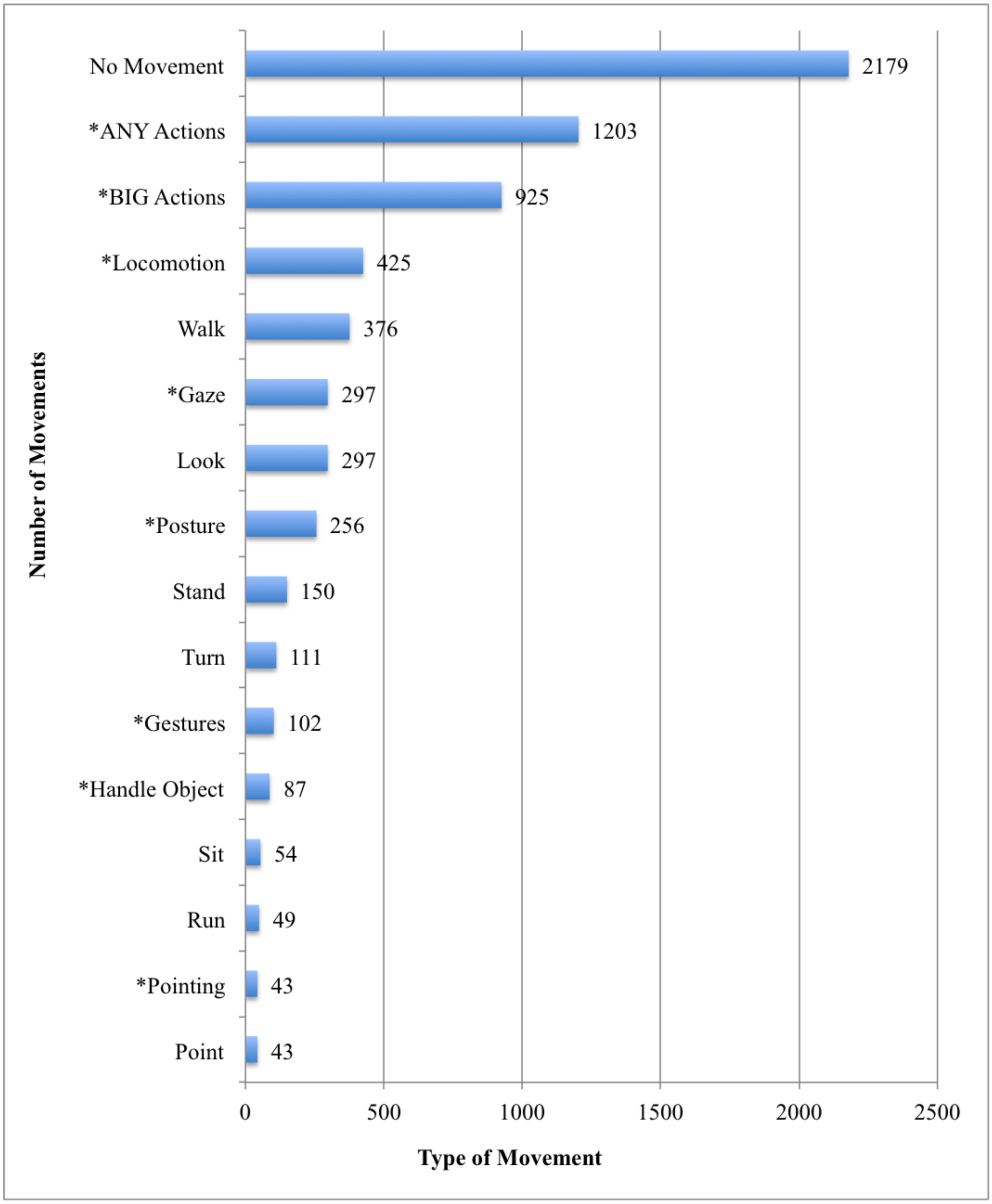


Figure A.1: Counts of Distinct Movements Within *Hamlet* with at Least 40 Instances out of 3477 Lines of Script
The Asterisk ($*$) Indicates Grouped Categories

The key movement types we captured within the *Hamlet* play can be seen in the list in Figure A.2. These movements are for both the speaker and the other characters onstage, and includes how we grouped them for better training capabilities (as will be discussed further in the Experimentation section). As you can see, the majority of movements were captured very few times within the dataset, with the majority being less than 100 instances out of 3477 instances possible.

Fighting

- Fighting∗
- Pushing∗

Handle Object

- Hand Object∗
- Pickup Object∗
- Throw Object∗

Change Posture

- Jump∗
- Lie Down∗
- Sit∗
- Stand∗
- Kneel∗

Gaze

Gestures

- Point∗
- Gesture∗
- Nod∗
- Raise Arm∗
- Wave∗

Locomotion

- Walk∗
- Run∗

Other

- Dig∗
- Turn∗
- Climb∗
- Kick∗

Figure A.2: Bundled List of Actions Captured within *Hamlet*
The Asterisk (∗) Indicates Items Considered as *Big* Actions

Each line of the play's speech was then used to create features for training. We utilized the openNLP package, tied to the Java openNLP implementation, within R to tag each word with its part of speech, along with the RTextTools package [38] for creating our n-grams and bag of words for our text. This information was then chunked into a bag of words approach, which used counts of each type of part of speech as a feature. Other features we incorporated into the training included:

- Number of lines for the speaker before this line
- Number of lines for the speaker after this line
- Number of annotations before this line
- Number of annotations after this line
- Number of speech lines since the last movement
- Maximum number of times a word is repeated
- Number of uppercase words in this line of speech
- Count of each punctuation mark within this line

Our hypothesis was that the length of the speech could trigger a movement for the character, such as moving towards the audience due to the start of a monologue. Another assumption was that movements might not occur really close together, to prevent excessive attention and confusion from the audience. Therefore, understanding how long it had been since the last movement was deemed important and a potential aide for learning implied movements. Knowing that there is an annotation coming up (which usually means an actor will perform some sort of movement), seemed to be useful for determining if a movement should occur now, or would be explicitly provided in the annotation later. Adding the features for punctuation, repeated words, and uppercase words was thought to help with identifying movement that might cause an actor to emphasize what they were saying, such as pointing or gesturing.

We pursued both a part of speech “sentence" and an n-gram bag of words approach for the speech due to Shakespeare’s known inclination to make up words and not repeat phrases a lot. We hoped this would help to find patterns in the sequence and frequency of “words," despite being unable to properly turn a parse-tree into a feature vector for training. We are confident that the sequence and dependency tree of the words in conjunction with the words themselves are key in being able to identify implied movement, except with Shakespeare’s work due to his jumbling of phrases to fit iambic pentameter. Several options utilized kernels and/or dynamic programming to learn off of parse trees and subtrees. This was not utilized here, but may be useful for future work. Ideally, also including the number of characters and their positions onstage for each line would be used to help capture the movements related to being upstaged, along with other theatre rule-guided movements.

A.4 Experimentation

Once we generated our features for all the lines in the play-script, we fed them into several machine learning algorithms: Maximum Entropy (MaxEnt), Support Vector Machine (SVM), Boosting, and Random Forests (RF). We focused only on the actions the speaker performed during their speech lines, and learning a specific movement or movement type one at a time. Initially, we took a random half of the lines (1739 lines) from the play-script for training the classifiers, and tested on the other half (1738 lines).

However, we found very poor results (same as a random classifier), as can be seen in the Table A.1. This was due to having such a large portion of the training set being classified as “no movement," due to often having much worse than a 10:1 ratio of movement to no movement (as can be seen in Figure A.1). Forman [27] discusses the issue of having a substantial class distribution skew (like we see with our Hamlet movement dataset), which worsens as the problem size scales upwards. Forman mentions the example of having many more news articles that do not meet a person’s personalization profile when looking at all news articles posted on the Internet world-wide. Most machine learning research does not consider such extreme skews as Forman saw (1:31 on average). Just as we saw with our dataset, we found it very difficult to beat the high accuracy that can be achieved by classifying everything negatively. Forman also mentions that feature selection becomes much more important in these types of situations where the training data is highly skewed.

We first attempted to address this by shrinking down our training set to a more

Table A.1: Best Results Per Movement Type While Training on Half of the Data-Set vs. a 2:1 Negative:Positive Ratio Data-Set, With All Features, Best Machine Learning Algorithms, and Unigrams. Fighting Did Not Have Enough Instances to Train On. Bolded Ones Performed Better Than Random.

	Even Split of Data-Set for Training						**2:1 Ratio Split of Data-Set for Training**					
Movement	**Ratio of Neg:Pos Instances**	**Recall**	**Precision**	**F_1-score**	**$F_{0.5}$-score**	**Matthews Correlation Coefficient**	**Training Size**	**Recall**	**Precision**	**F_1-score**	**$F_{0.5}$-score**	**Matthews Correlation Coefficient**
Any	1.8:1	0.017	0.455	0.033	0.075	**0.029**	1806	0.917	0.338	0.493	0.386	**0.038**
Big	2.4:1	0.000	0.000	0.000	0.000	0.000	825	0.000	0.000	0.000	0.000	0.000
Gestures	21.4:1	0.000	0.000	0.000	0.000	0.000	153	0.000	0.000	0.000	0.000	0.000
Object	25.1:1	0.000	0.000	0.000	0.000	0.000	134	0.907	0.013	0.026	0.016	**0.007**
Locomot.	5.1:1	0.000	0.000	0.000	0.000	0.000	639	0.953	0.079	0.146	0.097	**0.048**
Gaze	7.3:1	0.000	0.000	0.000	0.000	0.000	447	0.027	0.125	0.044	0.072	**0.037**
Pointing	50.7:1	0.000	0.000	0.000	0.000	0.000	66	0.048	0.026	0.034	0.029	**0.027**
Posture	8.5:1	0.000	0.000	0.000	0.000	0.000	384	0.000	0.000	0.000	0.000	0.000
Fighting	198.1:1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

specific set of lines where the ratio of "movement" to "no movement" was closer to a 2:1 ratio, while ensuring we did not use more than half of the annotated movement lines we were trying to classify. This performed marginally better, but still really did not get us past the performance of guessing "no movement" for everything or even a random classification, as can be seen in Table A.1.

We also found that we do not have enough examples of detailed movements in *Hamlet* to be able to classify all movements at a detailed level, such as hand fighting or lying down. Therefore, we were forced to look at the problem more generically than would be useful for actually predicting specific movements. We tried grouping the movements into buckets, as described in the Approach section; however only the posture, gaze, and locomotion came close to a 10:1 ratio, and even learning on those datasets ended up classifying almost everything as "no movement". The main two buckets that could give us almost reasonable results were the ones for any movement and any big movement.

We then looked at the different n-gram approaches to see what would work best to incorporate more of the relationships of the words in the phrases as seen in Table A.2. Bigrams appear to have done better than just a plain bag of words (BoW), with trigrams doing slightly worse than the bigrams, but still performing pretty well. 4-grams and 5-grams dropped performance to be closer to unigram performance in most instances. This correlates well with what Furnkranz[30] mentioned in their work with different n-grams for classifications.

As Forman[27] discussed, having such skewed training datasets puts more emphasis on the feature sets. Therefore, we pursued several different feature sets and combinations. We began initially with the sentences themselves turned into a BoW of different ngram lengths, along with the other features mentioned in the Approach section.

We then decided to take advantage of Shakespeare's iambic pentameter, which produced the majority of the lines as ten syllables, and a maximum of fifteen words

Table A.2: Highlights the Performance of Different N-Grams on Classifying the Different Movement Types on a 2:1 Negative:Positive Ratio Data-Set. Bolded Performed Better Than Random.

Movement Type	**n-gram**	**Machine Learning Algorithm**	**tp**	**fn**	**fp**	**tn**	**Accuracy**	**Recall**	**Precision**	**F_1-score**	**$F_{.5}$-score**	**Matthews Correlation Coefficient**	**Direction**
Any Mvmt	1	MaxEnt	508	46	997	120	0.376	0.917	0.338	0.493	0.386	**0.038**	
Any Mvmt	2	MaxEnt	505	49	988	129	0.379	0.912	0.338	0.493	0.387	**0.041**	⇑
Any Mvmt	3	MaxEnt	502	52	987	130	0.378	0.906	0.337	0.491	0.386	**0.034**	⇓
Any Mvmt	4	MaxEnt	503	51	987	130	0.379	0.908	0.338	0.492	0.386	**0.037**	⇑
Any Mvmt	5	MaxEnt	501	53	984	133	0.379	0.904	0.337	0.491	0.386	**0.035**	⇓
Gestures	1	Rand Forest	0	51	0	3273	0.985	0.000	0.000	0.000	0.000	0.000	
Gestures	2	Boosting	1	50	36	3237	0.974	0.020	0.027	0.023	0.025	**0.010**	⇑
Gestures	3	Boosting	6	45	229	3044	0.918	0.118	0.026	0.042	0.030	**0.023**	⇑
Gestures	4	MaxEnt	5	46	114	3159	0.952	0.098	0.042	0.059	0.047	**0.042**	⇑
Gestures	5	MaxEnt	5	46	111	3162	0.953	0.098	0.043	0.060	0.049	**0.043**	⇑
Locomotion	1	MaxEnt	202	10	2359	267	0.165	0.953	0.079	0.146	0.097	**0.048**	
Locomotion	2	MaxEnt	202	10	2351	275	0.168	0.953	0.079	0.146	0.097	**0.050**	⇑
Locomotion	3	MaxEnt	202	10	2379	247	0.158	0.953	0.078	0.145	0.096	**0.043**	⇓
Locomotion	4	MaxEnt	201	11	2347	279	0.169	0.948	0.079	0.146	0.097	**0.047**	⇑
Locomotion	5	MaxEnt	201	11	2346	280	0.169	0.948	0.079	0.146	0.097	**0.047**	

Table A.3: Best Results Per Feature Set, Training on a 2:1 Negative:Positive Ratio Data-Set and Any Movement. Bolded Performed Better Than Random.

Feature Set	**n-gram**	**Machine Learning Algorithm**	**tp**	**fn**	**fp**	**tn**	**Accuracy**	**Recall**	**Precision**	**F_1-score**	**$F_{0.5}$-score**	**Matthews Correlation Coefficient**
Text Only	1	RF	178	376	120	997	0.703	0.321	0.597	0.418	0.510	**0.263**
POS BoW Only	1	RF	158	396	137	980	0.681	0.285	0.536	0.372	0.456	**0.201**
All Features	3	RF	4	550	5	1112	0.668	0.007	0.444	0.014	0.034	**0.018**
POS BoW & Text	3	RF	4	550	5	1112	0.668	0.007	0.444	0.014	0.034	**0.018**
POS BoW & Other	2	Boost	0	554	0	1117	0.668	0	0	0	0	0
POS BoW, Text, Other	1	SVM	0	554	0	1117	0.668	0	0	0	0	0

per line. We decided to break these sentences into just the parts of speech (POS) tags as a sentence. This was intended to help with the issue of Shakespeare's writing not including much repetition. With the real sentences broken into BoWs, if we removed sparse words or stop words, we ended up with no words left. However, using the POS tags as sentences, we could get a similar concept, but were able to trim out sparse n-grams. This appeared to perform about the same as just counting the parts of speech and punctuation in the sentences, as can be seen in Table A.3.

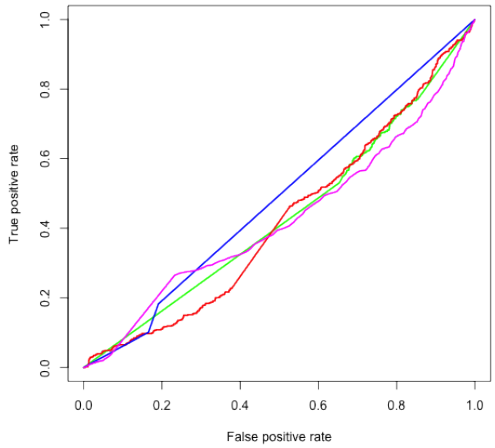
Finally, we combined the best feature sets described above (in different combinations) to see how it would perform. We chose to use the smaller training set, geared towards a 2:1 ratio of "no movement" to "movement," and focused primarily on classifying any movement within the play. The best classifications were obtained on just the unigrams of the actual speech text, although on average, the part of speech (POS) sentences with the speech sentences as bigrams and the other features did better.

To analyze these statistics, we used the ROCR package within R [84] to generate the ROC Curves for the better techniques. We also looked at the overall accuracy, precision, recall, F_1-score, $F_{0.5}$-score, and the Matthews correlation coefficient for each method. We were able to achieve high accuracy, but this was shown to be achievable with just a blind guess of everything to be "no movement". Therefore, the accuracy scores were not useful in determining the goodness of any of our methods.

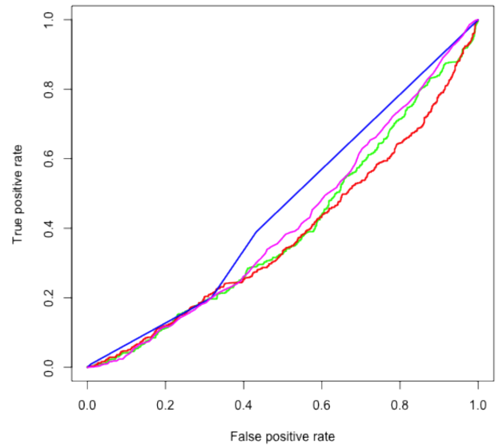
Looking at precision and recall, we often found we could do reasonably well with one, but very poorly with the other. Recall is focused on being able to classify as many positive examples as possible, whereas precision focuses on being more certain of classifying positive examples that really are positive classes. In our case, we are more concerned with making sure that if we identify a line as an implied movement, then there really should be an implied movement with that line. Therefore, precision was more important to us.

Trying to balance these two measures, we looked at the F_1-scores; however this put equal emphasis on both precision and recall. The $F_{0.5}$-score was better since it put more emphasis on the precision than the recall.

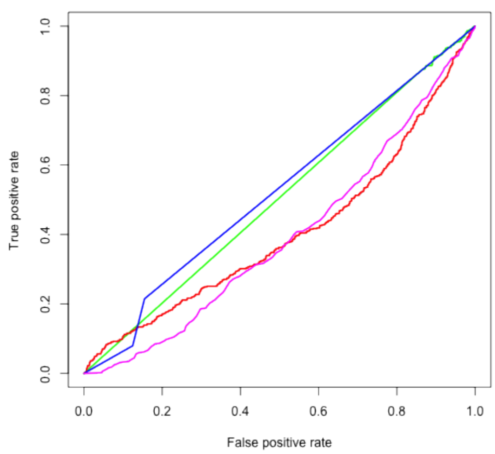
However, those approaches still left us uncertain to what degree we were able to outperform the random classifier and the guess "no movement" classifier. Therefore, we focused primarily on the Matthews Correlation Coefficient (MCC) measurement, as this takes into account true and false positives and negatives, and is generally regarded as a balanced measure, which can be used even if the classes are very skewed like ours. This measure returns a value between -1 and +1. A result of +1 represents a perfect prediction; 0 represents the same as a random classifier; -1 represents 100% incorrect classifications. Using this measure, we found that we were able to do better than the random classifier in many of our tests, as can be seen in the previous tables and in the ROC Curves in Figure A.3.



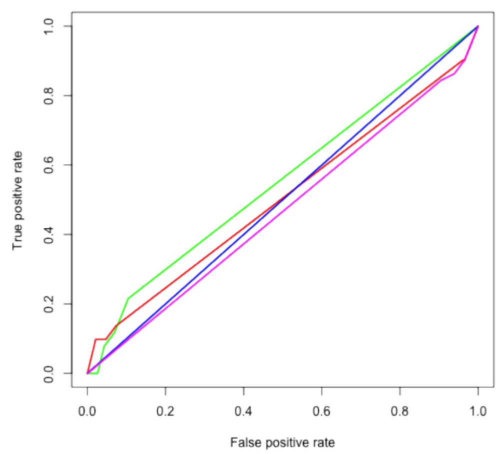
(a) No Text Features
Any Movement
1807 Training Cases
Unigrams



(b) POS BoW Features Only
Any Movements
1807 Training Cases
Unigrams



(c) Text Features Only
Any Movements
1807 Training Cases
Unigrams



(d) All Features
Gesture Movements Only
154 Training Cases
4-grams

Figure A.3: ROC Curves Samples for Techniques Utilized;
Red=SVM; Green=Maximum Entropy; Blue=Boosting; Magenta=Random Forests

A.5 Conclusions

Ultimately, Shakespeare is a more difficult context to use than typical play-scripts due to his tendency to make up words and rephrase things to fit into iambic pentameter. We were able to reasonably tell when some movement should occur, which should at least give us a sanity check for use with our previous work to ensure the characters are moving enough or not. However, the more specific movement types were more difficult to classify due to the limited number of test cases available in *Hamlet*.

Humans are able to do this with no prior examples, so there must be a way to learn these implied movements. Therefore, future work should include further analysis into tree kernels for machine learning, classifying more detailed movements using additional datasets, and an ability to classify more than one type of movement for a single line. Finally, incorporation of other features may be useful, such as number of characters onstage, locations of all the characters onstage, and other contextual features not included here.

APPENDIX B: DEFINITIONS

Below are some key definitions to help the reader better understand some of the terms used within the paper.

Annotation The director's directions for objects and actors in the play to perform during or around different speech acts within the play

Blocking The process of arranging moves to be made by the actors during the play [73]

BML Behavior Markup Language—An XML description language for controlling the verbal and nonverbal behavior of virtual characters [63]

Cue Signal or command given to indicate another action should follow [73]

Director The role responsible for the overall artistic vision of a production or play [73]

DownStage Part of the stage that is closest to the audience.

FDG Force-Directed Graph

FML Functional Markup Language—An XML description language for describing the effect that an intended action or plan should have on the environment, most obviously the agent itself [101]

M Mean

Marks The correct position on the stage for the actor to be at a given point in the play

Mocap A form of motion capture that captures motion data by a real person's movement and is applied to virtual characters to perform the same motion

MTurk Mechanical Turk—A site where jobs can be posted for users to complete for money, such as surveys and audio translations.

NLP Natural Language Processing

Non-Verbal Behaviors Actions that are performed by humans or characters that do not include speech, but may portray some sort of communication to others

Play-Script A written version of a play with annotations from a director to be followed during a performance

PML Perception Markup Language–An XML description language for describing a percept ,such as vision, touch, or sound, in order to provide input to a character or robot.

SD Standard Deviation

Stage Left The side of the stage to the actor's left when standing on the stage, facing the audience.

Stage Right The side of the stage to the actor's right when standing on the stage, facing the audience.

Theatre Configurations Setup of the stage area that can be in one of seven different arrangements [3]:

Proscenium Stage Typical "theatre"setup—contains a picture frame placed around the front of the playing area of an end stage

Thrust Theatre Stage surrounded by audience on three sides

End Stage A thrust stage extended wall to wall, with audience on only one side

Arena Theatre A central stage surrounded by audience on all sides

Flexible Theatre Big empty boxes painted black inside where the stage and seating are not fixed

Profile Theatre Audience is placed on risers on either side of the playing space, with no audience on either end of the "stage"

Sports Arenas Resemble large arena stages, but with a rectangular floorplan

UpStage Part of the stage that is behind the actor when they are facing the audience.

APPENDIX C: TOOLS

Here is where we will refer to all of the tools we built for this research and placed on GitHub.

C.1 2D Model

All code can be found here http://github.com/UNCCPhDResearchTalbot/IVA2012.

This includes all code from IVA 2012 conference paper, with the 2D model found in the 2DDemo folder, and the charting tools in the D3js folder.

All instructions are written with the assumption of running on a Mac.

This toolkit provides the 2D BML Realizer for the *Hamlet* "Graveyard Scene" utilizing only natural language processing (named entity recognition and part of speech tagging).

It utilizes jsGameSoup for the UI components and Node.js for processing, with natural, socket.io, and xml2js modules, and with a javascript and HTML front-end, which can be seen in Figure C.1.

- Install jsGameSoup
- Install node.js npm natural npm socket.io npm xml2js

In main.js, change line:

```
var BML = false;
```

to true if you want to use the BML baseline file, false if you want to use the natural language processing of the actual play-script.

Files provided:

- InputFile.txt ==> Hand-mapped BML code with some "triggers" for coinciding movements based on the 1964 *Hamlet* video

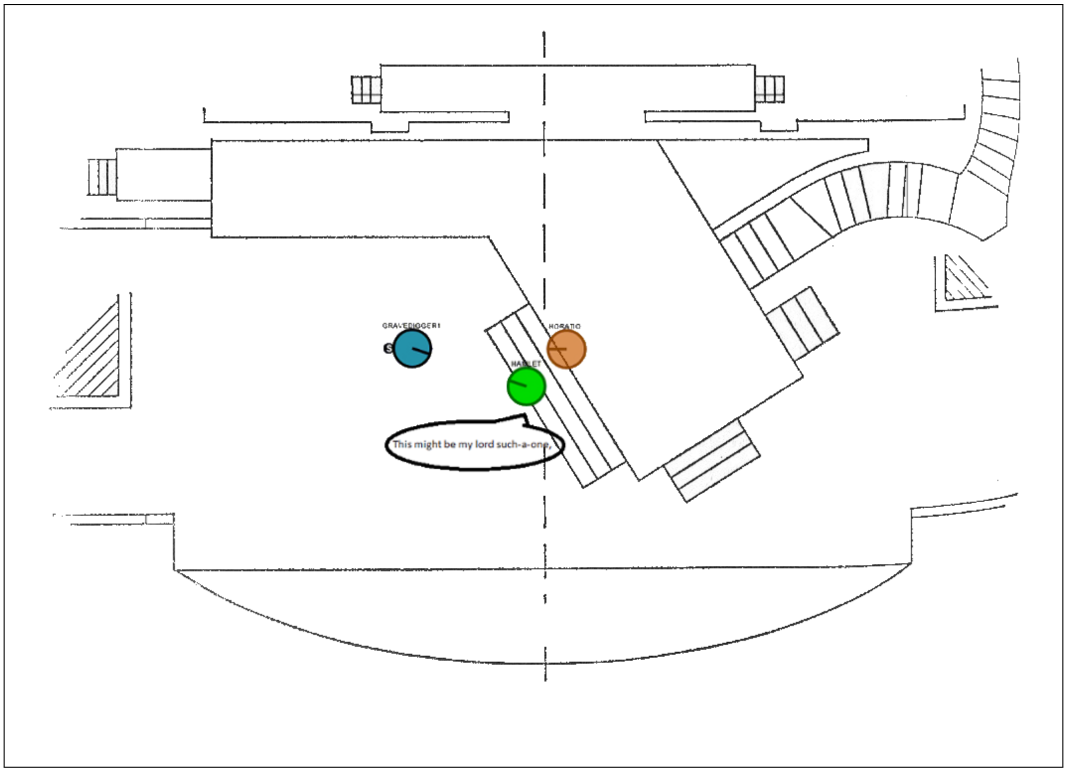


Figure C.1: 2D Simulation Screenshot

- InputScript.txt ==> Play-script from 1964 *Hamlet* video in natural language and formatted to play-script standards

To run: start python for page hosting

```
python -m SimpleHTTPServer 8888
```

start NodeJS module by running

```
node server
```

Then, open index.html file to begin running the scene and logging the character traces: http://localhost:8888/index.html

---

Charting utilizes output log files from the 2D BML Realizer to create character traces with a D3js component. One sample file is included for initial load in this directory (GRAVEDIGGER1.csv), with a sample output visible in Figure C.2

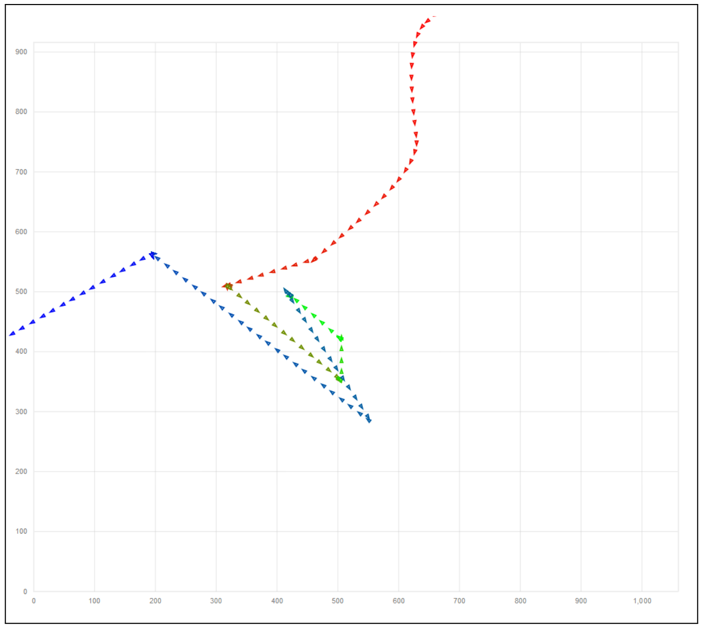


Figure C.2: Character Position Over Time Trace Chart Output

To run: start python for page hosting

```
python -m SimpleHTTPServer 8888
```

Open chartrace.html file to display the character trace: http://localhost:8888/chartrace.html

Change the filename (placed in the D3js folder) to whatever log file you want to show from running the GameSoup application & click Generate Chart button to see the new trace. Colors go from red to blue and show arrows pointing in the direction that the character was facing at each point.

C.2 Forces

All code can be found here https://github.com/UNCCPhDResearchTalbot/AAMAS2013Workshops.

This includes all code from the AAMAS 2013 Conference Workshop Code (both CAVE & SCW).

This code provides a 2D layout of characters, targets, and audience connections for random scenes and applies force-directed graphs for adjusting non-human characters

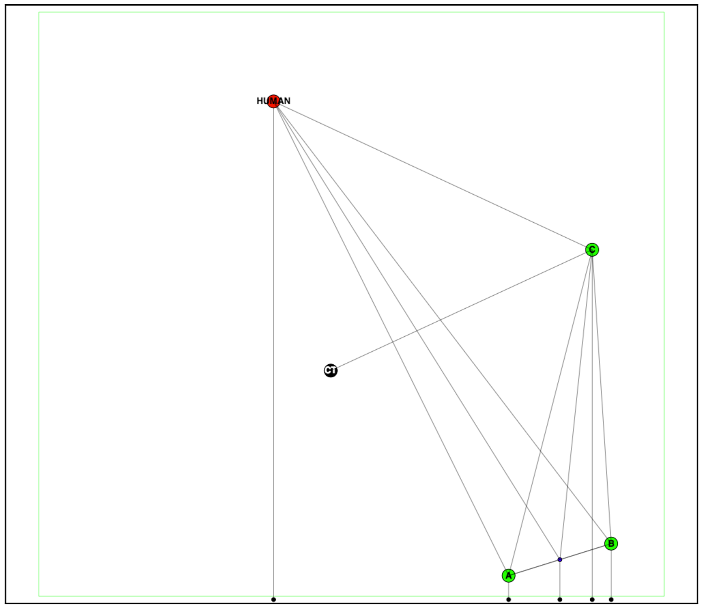


Figure C.3: 2D Forces Simulation

within the scene for optimal/pleasing positioning. A sample of the output can be seen in Figure C.3.

To run: start python for page hosting

```
python -m SimpleHTTPServer 8888
```

Open index.html file to display the scene http://localhost:8888/index.html. Click the button "CreateRandomFile" to randomly create a set of character positions, targets, etc. for a single moment within a scene. Click the button "Calculate Positions" to use force-directed graphs to adjust the non-human characters on the screen.

The file import button is limited due to browser limitations with retrieving/storing files on the desktop.

C.3 2D Rules

All code can be found here:

https://github.com/UNCCPhDResearchTalbot/AAMAS-2013

This includes all code from the AAMAS 2013 Conference Paper Code.

The RulesEngine folder contains the code to run the scene in either baseline or rules-applied mode and generate log files.

It utilizes jsGameSoup for the UI components and Node.js for processing, with natural, socket.io, and xml2js modules, and with a javascript and HTML front-end.

- Install jsGameSoup
- Install node.js npm natural npm socket.io npm xml2js

In main.js, change line:

```
var BML = false;
```

to true if you want to use the BML baseline file, false if you want to use the natural language processing of the actual play-script.

Files provided:

- InputFile.txt ==> Hand-mapped BML code with some "triggers" for coinciding movements based on the 1964 *Hamlet* video
- InputScript.txt ==> Play-script from 1964 *Hamlet* video in natural language and formatted to play-script standards

To run: start python for page hosting

```
python -m SimpleHTTPServer 8888
```

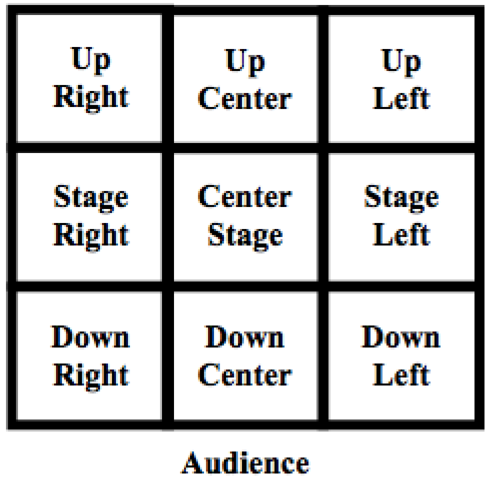
start NodeJS module by running

```
node server
```

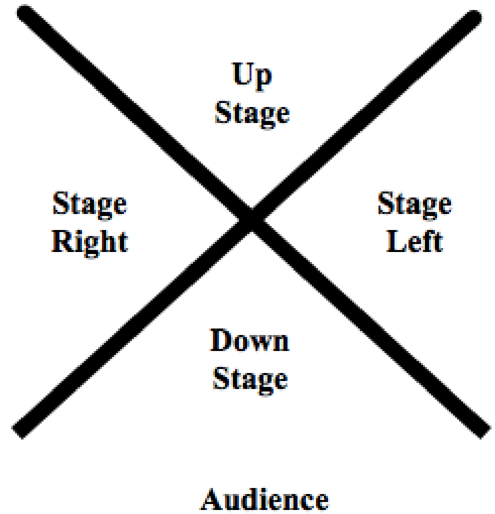
Then, open index.html file to begin running the scene and logging the character traces: http://localhost:8888/index.html

Sample output files are in the logs/bmllogs and logs/ruleslogs

Applied rules include:



(a) Stage Positions



(b) Stage Gaze Directions

Figure C.4: Stage Area Breakdown for Position and Gaze Table Generation

- looking where someone is pointing
- looking at speaker
- not upstaging higher importance characters
- look at what picking up
- move to what want to pick up
- see paper for full details

The Charts folder contains the D3js code to generate the charts to compare gaze direction for baseline only and rules blocking. Chart labels aren't correct, but follow (from top to bottom) for the y axis: stageright, audience, stageleft, backstage. These directions can be seen in Figure C.4b.

The x-axis represents time. Upper lines will represent logs from the baseline / bmllogs folder, with the Lower lines representing the logs from the ruleslogs folder.

To chart for position instead of gaze direction, modify the "generateData" function in the d3linecharts.js file to change the postns.forEach loop to map the stagegrid column instead of the rotation column. This will result in the y-axis displaying the following (from top to bottom): downstage, stageleft downstage, center downstage, stageright center, stageleft centerstage center, stageright upstage, stageleft upstage, center upstage, stageright offstage. These grid locations can be seen in Figure C.4a.

To run: start python for page hosting

```
python -m SimpleHTTPServer 8888
```

Then, open chartraces.html file to begin running the scene and logging the character traces: http://localhost:8888/chartraces.html. Enter the filename that exists in both the bmllogs/ and ruleslogs/ folders for the character you want to compare gaze direction or position for. Then, click the button. Figure C.5 shows what you will see, depending on whether you are running the gaze or the position traces.

Folders contain sample files for running.

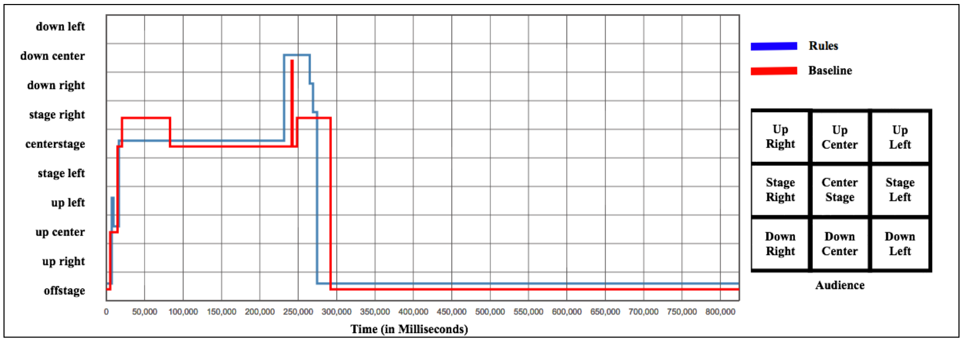
The CharTraces folder contains the D3js code to generate character traces for all characters during a scene for either baseline or rules blocking. Each character has a different shape, colors go from red to blue to indicate time progression, and each shape points in the direction the character was facing at each timestep.

To run: start python for page hosting

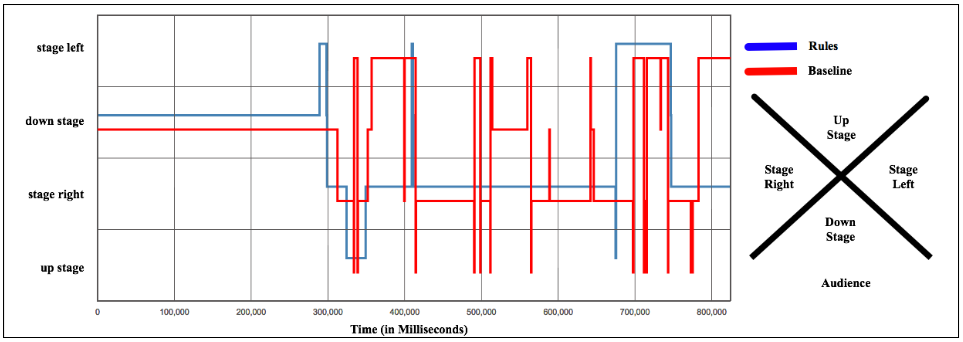
```
python -m SimpleHTTPServer 8888
```

Then, open chartraces.html file to begin running the scene and logging the character traces: http://localhost:8888/chartraces.html. Enter the folder name that contains the log files to be plotted, such as ruleslogs/ or bmllogs/, then click the button.

Folders contain sample files for running.



(a) Stage Positions



(b) Gaze Directions

Figure C.5: Charting Code Output

C.4 Implied Movement R Code

All code can be found here https://github.com/UNCCPhDResearchTalbot/FLAIRS2013.

This includes all code from the FLAIRS 2013 Conference Paper Code.

run file by typing:

```
source("~/Dropbox/FLAIRS-FINALFILES/runclassify.R");
```

within R.app which has NLTK installed & dependent libraries too. Different .R files run against different featuresets which are stored in the .txt files. The master file is in the .xlsx file for all statistics/features used.

Several files are included which contain details on the *Hamlet* scene for different types of movements and are used in the below script. Outputs are printed to .png files for the diagrams of the ROC curves, scriptoutput.txt (monitor run status), and testing123.csv (results) for the output of each learning session.

Charts for PDF.xlsx has summary information from all the runs for extraction into the paper. Movement Counts.xlsx has summary movement count information for the annotations within the scene.

Actual R code used is available in the gitHub repository.

C.5 2D Forces

All code can be found here https://github.com/UNCCPhDResearchTalbot/FLAIRS2014.

This includes all code used for the FLAIRS 2014 conference paper.

The BML-Rules Simulation folder contains the code to run the scene in either baseline or rules-applied mode and generate log files.

It utilizes jsGameSoup for UI components and Node.js for processing, with natural, socket.io, and xml2js modules, and with javascript and HTML front-end.

- Install jsGameSoup
- install node.js npm natural npm socket.io npm xml2js

In main.js, change line:

```
var BML = false;
```

to true if you want to use the BML baseline file, false if you want to use the natural language processing of the actual play-script.

Files provided:

- InputFile.txt ==> Hand-mapped BML code with some "triggers" for coinciding movements based on the 1964 *Hamlet* video
- InputScript.txt ==> Play-script from 1964 *Hamlet* video in natural language and formatted to play-script standards

To run: start python for page hosting

```
python -m SimpleHTTPServer 8888
```

start NodeJS module by running

```
node server
```

Then, open index.html file to begin running the scene and logging the character traces: http://localhost:8888/index.html

Sample output files are in the logs/bmllogs and logs/ruleslogs

Applied rules include:

- looking where someone is pointing
- looking at speaker
- not upstaging higher importance characters
- look at what picking up
- move to what want to pick up
- see paper for full details

To run: start python for page hosting

```
python -m SimpleHTTPServer 8888
```

Then, open chartraces.html file to begin running the scene and logging the character traces. http://localhost:8888/chartraces.html. Enter the filename that exists in both the bmllogs/ and ruleslogs/ folders for the character you want to compare gaze direction or position for. Then, click the button.

Forces Simulation folder contains the code to run the scene while applying forces to the characters and generate log files.

To change the randomness of the human, modify the ACCURACY variable to the % accuracy desired. To change which character is acting like the human, change the HUMAN variable to the name of the character to use as the human character.

C.6 3D Model-Hamlet

All code can be found here https://github.com/UNCCPhDResearchTalbot/BlockWorld

BlockWorld Unity 3d implementation of a Block World with block characters and pawns to validate spatio-temporal algorithms for positioning characters in virtual environments. Focuses on Baseline and NLP, and can be seen in Figure C.6.

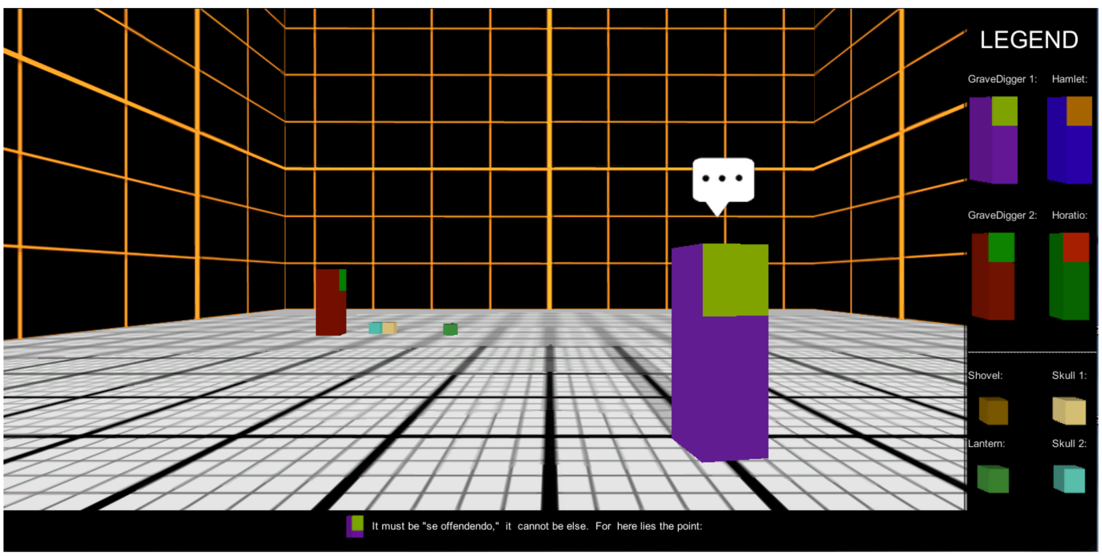


Figure C.6: 3D Simulation Screenshot

Additional adjustments and all code can be found here: https://github.com/UNCCPhDResearchTalbot/2ndUserStudyBlockWorld

2ndUserStudyBlockWorld Code used for within subjects study for FDG, rules, NLP, baseline, and random—not generalized. Can utilize screen dropdowns and buttons to test and run as desired.

C.7 BML Extractor

All code can be found here https://github.com/UNCCPhDResearchTalbot/FINALNLPBMLExport

GeneralizedExportBML = generalized code for converting play-scripts to pseudo-bml.

Replace the file called InputScript.txt with a formatted play-script & modify characters & pawns info for script.

Details on what to change: main.js—three sections:

- top of file—arrays of pawns & characters & marks, filenames, and additional movement words
- checkposition function—any special position calculations like skull (closest of two skull objects) or non-viewable pawns like coin
- bottom of file—creation of all marks, pawns, and characters

client.js—one section:

- bottom of file—creation of visible pawns & characters (same positions as main.js)

To run: start python for page hosting

```
python -m SimpleHTTPServer 8888
```

start NodeJS module by running

```
node server
```

Then, open index.html file to begin running the scene and logging the character traces: http://localhost:8888/index.html

Results will be in the log directory.

C.8 Generalized 3D Model

All code can be found here https://github.com/UNCCPhDResearchTalbot/FINALUnityBlockWorld

GeneralizationBlockWorld Code used for within subjects study for FDG, rules, NLP, baseline, and random. Can utilize screen dropdowns and buttons to test and run as desired. Has been generalized and accepts files with the following formats:

- Initialization file—tab separated columns
  - Type: C or P or M for type of object—S for speed
  - Speed: Slow, Med, Fast (only for S)
  - Name: uppercase name with no spaces
  - Start X Position
  - Start Z Position
  - Rotation 4 components
  - Holding Object: define prior to the character!!
  - Color: blue, purple, red, green, yellow, orange, brown, white
  - Importance: 1 to 8 for chars only saying 1 = highest priority char to lowest–only chars
  - Voice: Alex, Ralph, Bruce, Fred, more? Kathy, Vicki, Victoria, Agnes, Princess, Junior
- BML file—tab separated columns
  - 1—Y or N—this line should be executed at the same time as the next line
  - 2—SPEAK or MOVE—determines whether it is a speech line or a movement line (easier parsing)

  - 3—<character name in CAPS>—who the current actor of this line of BML is
  - 4—<target name in CAPS>—what object or person is targeted by this BML (pickup what object? follow what person?), XXXXX is ok if no target
  - 5—<bml>—the actual bml for the command

- Legend image

C.9 Interactive Application

All code can be found here https://github.com/UNCCPhDResearchTalbot/InteractiveUnity

This is the code used for the interactive study, with an example seen in Figure C.7.

Inputs from the Generalized 3D Model still apply, but additional files are required. Additional files:

- Three Init Files (one per scene)
- Three BML files (one per scene)
- Three Play-script files (one per scene)
  - Formatted for scrolling on the screen for the user
- Three Count files (one per scene)—tab delimited
  - Number of lines in section
  - Is speech? Y/N
  - Is movement? Y/N
  - Speech to say

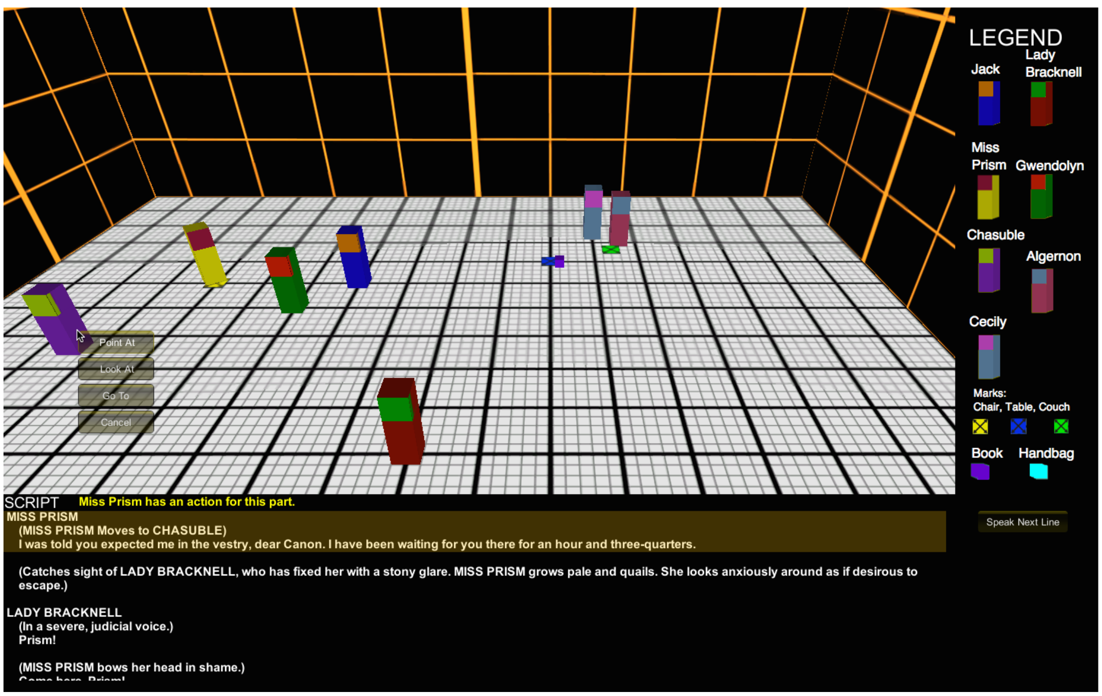


Figure C.7: Interactive Simulation Screenshot

VITA

Christine Talbot was born in 1977 in Tucson, Arizona. She completed two Bachelor of Science degrees (Math and Computer Science) in a total of three years from the University of Missouri at Columbia in 1998 as the Salutatorian. She was a member of the Golden Key Honor Society and the Phi Kappa Phi Honor Society, as well as made Dean's list every semester.

Christine began working with Lucent Technologies upon graduation, and started her graduate studies at Washington University in Saint Louis, Missouri. She continued her career working with Siebel Systems, Oracle, TIAA-CREF, and Salesforce.

She restarted her Masters degree in 2009 at the University of North Carolina at Charlotte, completing it in 2011 while working for TIAA-CREF. Christine continued her studies towards a PhD at the University of North Carolina at Charlotte. She now works for Salesforce as a Program Architect, and plans to complete her degree in 2018. She is a member of the Association for Computing Machinery, Golden Key Honor Society, Phi Kappa Phi Honor Society, and National Society of Leadership and Success.