DIRECTING VIRTUAL HUMANS USING PLAY-SCRIPT SPATIOTEMPORAL 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

2015

Approved by:

__________________________
Dr. G. Michael Youngblood

__________________________
Dr. Richard Souvenir

__________________________
Dr. Jing Xiao

__________________________
Dr. Jamie Payton

__________________________
Dr. Andrew Hartley
ABSTRACT

CHRISTINE TALBOT. Directing Virtual Humans Using Play-Script Spatiotemporal 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 spatiotemporal 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 spatiotemporal 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 and cohesion 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 natural language processing, rules engines, and force-directed graphs. With these methods, we are able to similarly map any play-scripts 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. The goal of this research is to perform dynamic spatiotemporal reasoning with virtual characters following a play-script.

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.
ACKNOWLEDGMENTS

Will be added later.
TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: BACKGROUND 10
  2.1. Human Directions 10
  2.2. Play-Scripts 11
  2.3. Theatre 13
  2.4. Shakespeare 16

CHAPTER 3: RELATED WORK 17
  3.1. Motion Capture Files 17
  3.2. Markup Languages 18
    3.2.1. Behavior Markup Language 18
    3.2.2. Functional Markup Language 20
    3.2.3. Perception Markup Language 21
  3.3. Robotics 21
  3.4. Virtual Agents 22
  3.5. Natural Language Processing 23
  3.6. Psychology and Spatial Cognition 25
  3.7. Force-Directed Graphs 26
  3.8. Judging Criteria 29
  3.9. Planners 30
3.10 Crowd Modelling
3.11 Other

CHAPTER 4: METHODOLOGY

4.1 Baseline

4.2 Natural Language Processing to BML

4.3 Rules
   4.3.1 Grouping Spatial Rules
   4.3.2 Conversational Spatial Rules
   4.3.3 Theatre Rules
   4.3.4 General Rules
   4.3.5 Architecture

4.4 Force Directed Graphs
   4.4.1 Features
   4.4.2 Graph Structure
   4.4.3 Application of Graph Structure
   4.4.4 Algorithms
   4.4.5 Forces
   4.4.6 Validations

4.5 User Studies
   4.5.1 NLP versus BML
   4.5.2 NLP versus BML versus Rules versus FDGs
   4.5.3 Baseline versus FDGs for Human-Interaction

4.6 Generalization
LIST OF FIGURES

FIGURE 1: Interview with Ada and Grace, MOS Boston’s Virtual Human Museum Guides 2

FIGURE 2: Left to Right: Light Stage 1s 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 HDRI map; Light Stage 5 uses high-speed photography to record an actors reflectance with time-multiplexed illumination; Light Stage 6, at 8m in diameter, allows performance relighting for the whole human body. 2

FIGURE 3: SmartBody is a BML Realizer that provides locomotion, steering, object manipulation, lip syncing, gazing, nonverbal behavior, and retargeting in real-time. 4

FIGURE 4: Scene Directions Formatting Example 11

FIGURE 5: Stage Directions Formatting Example 12

FIGURE 6: Character Stage Directions Formatting Example 12

FIGURE 7: Play-Script Excerpt from Sir Gielgud’s Hamlet on Broadway 1964 13

FIGURE 8: Stage Layout 14

FIGURE 9: Example of a Counter-Cross: Actor B could move to Center Stage or to Right Stage to Counter Being Upstaged by Actor A 15

FIGURE 10: Example of a BML Request 19

FIGURE 11: Example of an FML and BML Request 20

FIGURE 12: A theoretical model of personal space as a function of interpersonal relationship and expectation of interaction 26

FIGURE 13: Hand-mapped blocking of Richard Burton’s Hamlet play from 1964 33

FIGURE 14: Work Required to Write BML 34
FIGURE 15: 2D model using hand-written BML commands

FIGURE 16: 3D Enactment of Hamlet in Unity Using the SmartBody BML Realizer

FIGURE 17: Sentence Parsing Structure

FIGURE 18: Rules Engine Architecture

FIGURE 19: Logic in the Rules Engine

FIGURE 20: Rules Engine Architecture with Force-Directed Graph Adjustments for Human-Controlled Character(s)

FIGURE 21: Force-Directed Graph Structure

FIGURE 22: Adjustments from Force-Directed Graphs

FIGURE 23: Block World Representation Utilized in 3D Videos

FIGURE 24: Character Traces for Hamlet (first row), Horatio (second row), Gravedigger1 (third row), Gravedigger2 (fourth row); Column 1 are the baseline traces, Column 2 are traces from our method

FIGURE 25: Comparison of Character Traces for Position Over Time (ms) and Stage Grid Diagram

FIGURE 26: Comparison of Character Traces for Gaze Direction Over Time (ms) and View Angle Diagram

FIGURE 27: Character Positioning Using Forces Red=Human; Green=AI Character; Black=Target; Blue=Center Point

FIGURE 28: Character Positioning Using Force-Directed Graphs—Red=Human; Green=AI Character; Black=Pawn; Blue=Center Pt

FIGURE 29: Counts of Distinct Movements Within Hamlet with at Least 40 Instances out of 3477 Lines of Script The Asterisk (*) Indicates Grouped Categories

FIGURE 30: Bundled List of Actions Captured within Hamlet The Asterisk (*) Indicates Items Considered as Big Actions
FIGURE 31: ROC Curves Samples for Techniques Utilized; Red=SVM; Green=Maximum Entropy; Blue=Boosting; Magenta=Random Forests
LIST OF TABLES

| TABLE 1: Attractive and Repellent Forces $|\delta| = \text{Separation Distance}, \alpha$ = Desired Separation Distance | 63 |
|---------------------------------------------------------------|-----|
| TABLE 2: Correlation of Spatial-Temporal Questions            | 68 |
| TABLE 3: Average Distances (in Feet) Between Characters and Pawns | 80 |
| TABLE 4: Experiment Results of Occlusion and Clustering Averaged Over Entire Scene | 85 |
| TABLE 5: 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. | 118 |
| TABLE 6: 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. | 121 |
| TABLE 7: Best Results Per Feature Set, Training on a 2:1 Negative:Positive Ratio Data-Set and Any Movement. Bolded Performed Better Than Random. | 122 |
CHAPTER 1: INTRODUCTION

Currently, most virtual 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 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. 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).

In addition, the team utilizes high fidelity characters, generated by Paul Debevec and their Light Stage technology, as seen in Figure 2. Even with these possibilities to make realistic looking and acting characters, we are unable to 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.

The question then becomes, how can we both realistically, and easily position
Figure 1: Interview with Ada and Grace, MOS Boston’s Virtual Human Museum Guides

Figure 2: Left to Right: Light Stage 1s 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 HDRI map; Light Stage 5 uses high-speed photography to record an actors reflectance with time-multiplexed illumination; Light Stage 6, at 8m in diameter, allows performance relighting for the whole human body.
characters in a virtual environment?

Today, there are limited capabilities for automatically positioning characters in a 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 non-verbal behaviors and interacting with humans to make the characters seem more realistic.

The movie 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 3). These also require some lower-level coding, but begin to abstract and parameterize the motion of the characters. It creates a more dynamic and repeatable motion for the characters.
Figure 3: SmartBody is a BML Realizer that provides locomotion, steering, object manipulation, lip syncing, gazing, nonverbal behavior, and retargeting in real-time.

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.

So we can move the characters, but how do we impart our 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 movement 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 assumed that they are easier and faster to create than BML for specifying positional information.

**Hypothesis 1.1** *Play-scripts can provide similar positioning of virtual characters within a scene as a real actor.*

This hypothesis can be evaluated by comparing the positioning of characters from a real performance versus the use of natural language processing on the same play-script.

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 and when/how. Because play-scripts typically consist of short and to the point directions from the director (often as sentence fragments), actors are required to apply their interpretation for the gaps. They do this, just as humans infer details from vague directions. Therefore, it is assumed 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** *Play-scripts can provide better positioning of virtual characters within a scene if additional rules are applied to the annotations.*
This hypothesis can be evaluated by comparing the positioning of characters from a real performance to both the natural language processing and a rules engine applied to the same play-script.

We realize that actors may also take it upon themselves to improvise with a script. We conjectured that perhaps there is something in what the character is saying which caused the actors to perform this extra, unannotated movement. In addition, it is assumed that what the actor(s) are saying may also impact what movement is performed within a scene.

**Hypothesis 1.3** *Additional movement within a scene can be inferred by what is said by the characters.*

This hypothesis can be evaluated by utilizing a real performance’s movements and speech lines to learn, then apply that learning to new speech lines.

Force directed graphs have been used to provide a clean layout for large and complex graphs. 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.

In a play, actors arrange themselves on the stage according to both basic rules of the theater, 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.

**Hypothesis 1.4** *Force-directed graphs can position characters onstage with similar locations as a play-script with additional rules.*

This hypothesis can be evaluated by comparing a performance utilizing a rules engine to one applying force-directed graphs techniques.

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, 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 characters on-stage are assuming that the human followed the script. If the agent-controlled characters do not adjust, they could create unrealistic positionings 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 1.5 *Force-directed graphs can incorporate human-controlled characters with a set of virtual characters, adjusting the virtual character movements around the human’s motion.*

This hypothesis can be evaluated by comparing the clustering and occlusion of characters utilizing force-directed graphs versus other techniques when a human-controlled character randomizes some or all of their movements.

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 imperceptible to the typical user, and perceived as a good performance.

Hypothesis 1.6 *The combination of play-scripts, rules, and force-directed graphs can surpass the human-perceived threshold of a quality performance.*

This hypothesis can be evaluated by user studies that compare similar scenes with each of the techniques for their spatial positioning.

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

Hypothesis 1.7 *These tools and algorithms can be generalized to a set of play types, grouped by their typical organization.*
This hypothesis can be evaluated by either quantitative analysis of positioning for each of the play types or user studies for a qualitative analysis.

In the remaining sections of this document, we will discuss the background, related work, methodology, and experimentation to prove or disprove our 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:

<table>
<thead>
<tr>
<th>A: Excuse me...</th>
</tr>
</thead>
<tbody>
<tr>
<td>B: Can I help you?</td>
</tr>
<tr>
<td>A: Where is the conference room?</td>
</tr>
<tr>
<td>B: Go down the hall and take the elevator to the fourth floor.</td>
</tr>
</tbody>
</table>

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 4.

```
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 4: 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). See the example in Figure 5.

<table>
<thead>
<tr>
<th>GRAVEDIGGER1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Digs and sings)</td>
</tr>
<tr>
<td>In youth when I did love, did love, Methought it was very sweet,</td>
</tr>
<tr>
<td>(HAMLET and HORATIO enter center, cross to the side steps, and watch him, amused)</td>
</tr>
<tr>
<td>GRAVEDIGGER1</td>
</tr>
<tr>
<td>(Digs and sings)</td>
</tr>
<tr>
<td>To contract, oh the time for-a my behove, O me thought there-a was nothing a-meet.</td>
</tr>
</tbody>
</table>

Figure 5: 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. See the example in Figure 6.

<table>
<thead>
<tr>
<th>HORATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Laughing)</td>
</tr>
<tr>
<td>Aye, my lord.</td>
</tr>
</tbody>
</table>

Figure 6: 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 7 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 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. Assumptions are also made that the actors understand 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 of each as shown in Figure 8.

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 $45^\circ$ 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 [47].

![Figure 9: Example of a Counter-Cross: Actor B could move to Center Stage or to Right Stage to Counter Being Upstaged by Actor A](image)

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 opposite direction of the other actor—see Figure [9]). 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

Shakespeare 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 [52]. 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 [34].

We focused 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 [76]. It was filmed during three successive stage performances in June/July 1964 by Electronovision, Inc. [13]. 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 [76].
CHAPTER 3: RELATED WORK

We focus on creating an 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.

3.1 Motion Capture Files

Canned/explicit cut scenes are very common in games, films, and virtual environments. This is often accomplished via 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 [28].

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 Markup Languages

Current methods such as Behavior Markup Language (BML) [55], Functional Markup Language (FML) [90], and BML Realizers like SmartBody [22] and Elckerlyc [99] 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.

3.2.1 Behavior Markup Language

Markup languages, such as Behavior Markup Language (BML) [55], are making it possible to abstract the control of virtual characters. BML abstracts the physical realization of behaviors and movements, along with their constraints. It is not concerned with the intent behind the movements [55]. BML is structured like a typical XML message, as seen in Figure [10]. One can control what is done, when it is done, and what runs concurrently with other commands. However, it is often at such 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).
Figure 10: Example of a BML Request

Plus, 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.

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.\(^{[55]}\)

BML consists of a block of XML which 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 \(^{[22]}\) or Elckerlyc \(^{[99]}\), which execute behaviors specified by BML on the character in the environment. However, BML realizers are still their early stages of development.
3.2.2 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 [90]”. The Non-Verbal Behavior Generator (NVBG) module in the Virtual Human Toolkit (VHToolkit) [38] utilizes these FML commands along with rules to generate BML with nonverbal behaviors inserted into the speech text [43]. 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 11.

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 NPCEditor [44] component of the VHToolkit provides a utility to translate the questions to the

![Figure 11: Example of an FML and BML Request](image)
answers for passing on to the NVBG module and may be useful in this effort if expanded to encompass translations to spatial movements as well.

3.2.3 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 SAIBA framework, splits out the behavior scheduling and the behavior execution. It is able to interrupt the robot’s behavior plan based on the perceptions that are sent to it via PML. This enables the support of on the fly changes to behaviors.

3.3 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 [8]. 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 [51]. 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.

Also, Langley, Schermerhorn, and Scheutz provide an approach to human-robot interaction which allows for communicating complex tasks, which can be translated into procedures for the robot [92]. Matuszek and Herbst take natural language and robotic perceptions and translate it into a robot control language for following route directions [57]. Dzifcak, Scheutz, and Baral also utilize natural language to determine actions and goals for the robot [21]. All of these incorporate telling a robot what to do or where to go.

3.4 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 [20]. The FAtiMA architecture was built to create autonomous believable characters that allowed the establishment of empathetic relationships with other characters in the FearNot! system [19].

Then there are things like the Virtual Storyteller, which enables characters to
tell a story with the appropriate gestures, prosody, and so forth [88]. 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 [41]. 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 [72].

Other research utilizing virtual agents focuses primarily on the conversational and nonverbal domains, such as Thespian [73], Virtual Storyteller [88], and Stability and Support Operations (SASO) [38]. 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.

3.5 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” [31]. Kelleher and Costello [36] and Regier and Carlson [66] built learned models for the meanings of static spatial prepositions such as “in front of” and “above” while Tellex focused on “across” [87].

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 [42]. Kelleher also proposed a framework for understanding prepositions primarily around the closeness of objects and the visual representation of those objects [36]. His research explores how humans describe where objects are within space, which is 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 [17].

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 [16]. 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 [45]. 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 [24]. Other methods include the use of spatial templates to identify acceptable locations with respect to a given object for a particular preposition [49], and vector sum models [66] to formalize spatial relationships.

3.6 Psychology and Spatial Cognition

Conversational space, spatial prepositions, and group dynamics have been studied for years in psychology.

A lot of the work done by groups 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 [32]

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 [78], as seen in Figure 12. People also prefer to be across from one another than next to each other in most situations, but there is importance to the environment for determining what distance is comfortable [75]. 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 [78]. There is also a discussion around the effects of spatial invasion on character behaviors and movements within Sundstrom’s review.

3.7 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 [93]. 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 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 [39], 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.

Next, 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. [25]

Then there is the algorithm by Kamada and Kawai which tries to minimize the distance of vertices from their corresponding underlying graph distances [35]. This method requires more computation and storage space since it requires a shortest
distance calculation on every vertex before running its minimization function \cite{39}. 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 which 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 out there that can accommodate tens and hundreds of thousands of vertices. These attempt to break down the graph into simpler structures, like Hadany and Harel \cite{29} or Gajer, Goodrich, and Kobourov \cite{27}. 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, such as social networks, such as Bannister et al’s work. Their work attempts to centralize vertices that are more theoretically central in the graph \cite{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 [70].

3.8 Judging Criteria

There is a lack of existing tools to qualitatively evaluate the spatiotemporal rea-
soning within a performance. However, one-act play competitions are often critiqued
by judges and include spatial aspects of the performance in their evaluations. There-
fore, we reviewed these evaluation criteria which 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 concern
of the variety and balance in the use of the 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 in-
cluded 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. [59]

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 spatial-temporal questions are included in Pavis’s questionnaire, such as: space organization, relationships between actors, and pacing. [62]

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 5 star scale. [60]

3.9 Planners

To expand on this, there are many planners out there, such as the one by Vidal which attempts to enable a planner and plan execution system to run concurrently [96]. Also, we may look at better movement target predictions utilizing the stage directions and details from Frank’s work [24] on logic for geographic locations with
respect to known object locations. This will require some basic 9-cell grids to divide up the stage (center, upstage, downstage, stageleft, stageright). Better thoughts on translating the he/she pronouns, as well as ordering of phrases such as “follows” vs. “followed by”. Also, focus on learning about certain phrases equating to certain actions or movements utilizing some Bayesian reasoning.

3.10 Crowd Modelling

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.

3.11 Other

Additional related areas include: waypoints, gradient fields, mobility techniques used in games, robot mobility planning, automated cinematography, and Craig Reynolds’s work on boids. When combining automated cinematography, SmartBody, and this work, it has the potential of providing an end-to-end set of techniques for creating movies.
CHAPTER 4: METHODOLOGY

4.1 Baseline

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 5 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 [76], 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 [13] 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 [13] and annotated play-script [76] 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 our ground-truth to compare a basic natural language translation of the same annotated scene.

We manually mapped out about ten minutes of Act V, Scene I from Hamlet, as produced by John Gieguld in 1964 (Figure [13]). This happens to be the graveyard scene where Hamlet reminisces about a skull that may have been Yorick, an old friend. 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 BML (and FML where appropriate for speech), as seen in Figure 14.

Next, a simple GameSoupJS and NodeJS application was built to visualize the results of the BML and FML. 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 this particular scene, only a Lantern (L),
Spade (S), and two Skulls (X) are 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 15. This could also have been done with a BML Realizer, like SmartBody, in conjunction with a game engine like Unity, which can be seen in Figure 16.

This application and BML script became our ground truth 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).

4.2 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
Figure 15: 2D model using hand-written BML commands

Figure 16: 3D Enactment of Hamlet in Unity Using the SmartBody BML Realizer
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 so forth.

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.

Next, 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, 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.” We then generate and send the BML to our simulator to perform the action.

Assumptions 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. Other assumptions were made about the timing of these spatial events. All sentences, or sentence fragments, within a single set of parentheses were assumed 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 assumption 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. These annotations were assumed to take on a structure like Figure 17 and were parsed using Algorithm 1.

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

Figure 17: Sentence Parsing Structure
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] = wordtranslated
        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 function

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

These techniques can 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 setups of any scene for any play.

**Algorithm 2** Pseudo-Code for Natural Language Parse Sentence Algorithm

```plaintext
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 towhat object or postn
end function
```

### 4.3 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.

4.3.1 Grouping Spatial Rules

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, or proxemics [32]. Hall describes four different zones that personal space is divided into: intimate, personal, social, and public zones [30]. 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 the participant’s comfort area, the participant will move toward the speaker. Similarly, if someone invades the personal space of a participant, the participant will move away [32]. 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 [37].

4.3.2 Conversational Spatial Rules

Older 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 [75]. Also, friendship and at-
traction can affect the spatial distances between people by decreasing them, while negative attitudes may not have much affect on the spatial distances [78].

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 [78]. He also discusses the effects of spatial invasion for character behaviors and movements and provides a nice overview of multiple research efforts looking at conversational space for both sitting and standing positions [78].

4.3.3 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. 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 of each as shown on the bottom right of Figure 8.

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.\[47\].

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 opposite direction of the other actor—see Figure 9). 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.\[5\].

4.3.4 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.

4.3.5 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 combined the use of a standard annotated play-script with a natural language processor, which utilizes a part of speech tagging and named entity recognition module to extract the high-level movements of the characters.

These movements were fed into our rules engine (as seen in Figure [18]) 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
Figure 18: Rules Engine Architecture

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 19.

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

\[ r_7: \text{Characters should be next to an item they wish to pick up} \]
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 which 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.

4.4 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 20). This will help us to accommodate the unpredictable actions of the human on-stage with respect to the overall production of the play.

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 these different approaches to spatially displaying different types of graphs, we decided upon basing our character positioning adjustments on the algorithm by Fruchterman and Reingold. 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.
4.4.1 Features

4.4.1.1 Even Vertex Distribution

In theater, 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 are needed to accomplish this.

The distance between each character on-stage must be measured and compared. The initial run 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, which can be introduced in a secondary run. 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.

4.4.1.2 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 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.

4.4.1.3 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 would be able to accommodate this if we can guarantee we have a 3-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.
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.

4.4.1.4 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.

4.4.1.5 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.

4.4.1.6 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.

4.4.1.7 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.

4.4.1.8 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.

4.4.2 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 21. 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 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).
Figure 21: Force-Directed Graph Structure

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.

4.4.3 Application of Graph Structure

The algorithms described in our previous paper [83, 81] are then utilized to determine better target position(s) for the onstage characters. These algorithms include a force-directed graph drawing algorithm based 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 [25]. 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 [83].

4.4.4 Algorithms

To accomplish the positioning of characters using force-directed graphs, we introduced the following functions (based primarily on Fruchterman and Reingold’s algorithm) which are shown in the pseudo-code in Algorithm 3. We focus on how we want to combine the above requirements to create our own algorithm.

- Add Character(s) (Algorithm 4)
• Character Move to Position (Algorithm 5)

• Human Moves (Algorithm 5)

• Character(s) Leave (Algorithm 6)

• Time Step ( Algorithms 3 & 5)

**Algorithm 3** Pseudo-code for Force-Directed Graph Adjustments within Virtual Stage Environments

function \texttt{AdjustAll} \\
G \leftarrow (V, E); \quad \triangleright \text{ the vertices are assigned initial positions based on annotations} \\
function \texttt{f} \texttt{a}(x) \leftarrow \texttt{return} \texttt{AttractiveForce} \\
function \texttt{f} \texttt{r}(x) \leftarrow \texttt{return} \texttt{RepellentForce} \\
for \( i = 1 \rightarrow \text{iterations} \) do \quad \triangleright \text{ calculate repulsive forces} \\
\text{for} \ v \ \text{in} \ V \ \text{do} \quad \triangleright \text{ each vertex has two vectors: \texttt{.pos} and \texttt{.disp}} \\
\quad v.\text{disp} \leftarrow (0, 0); \\
\text{for} \ u \ \text{in} \ V \ \text{do} \\
\qquad \text{if} \ (u \neq v) \ \text{then} \quad \triangleright \delta \text{ is the difference vector between} \\
\qquad \quad \delta \leftarrow v.\text{pos} - u.\text{pos}; \quad \triangleright \text{ the positions of the two vertices} \\
\qquad \quad v.\text{disp} \leftarrow v.\text{disp} + (\delta/|\delta|) \ast f_r(|\delta|); \\
\text{end if} \\
\text{end for} \\
\text{end for} \\
\text{for} \ e \ \text{in} \ E \ \text{do} \quad \triangleright \text{ calculate attractive forces} \\
\quad \delta \leftarrow e.v.\text{pos} - e.u.\text{pos}; \\
\quad e.v.\text{disp} \leftarrow e.v.\text{disp} - (\delta/|\delta|) \ast f_a(|\delta|); \\
\quad e.u.\text{disp} \leftarrow e.u.\text{disp} + (\delta/|\delta|) \ast f_a(|\delta|); \\
\text{end for} \quad \triangleright \text{ limit max displacement to temperature } t \\
\text{for} \ v \ \text{in} \ V \ \text{do} \quad \triangleright \text{ Only update non-human characters} \\
\quad v.\text{pos} \leftarrow v.\text{pos} + (v.\text{disp}/|v.\text{disp}|) \ast \text{min}(v.\text{disp}, t); \\
\quad v.\text{pos}.x \leftarrow \text{min}(W, \text{max}(0, v.\text{pos}.x)); \\
\quad v.\text{pos}.y \leftarrow \text{min}(L, \text{max}(0, v.\text{pos}.y)); \quad \triangleright \text{ Prevent displacement off the stage} \\
\quad v_{\text{aud}.\text{pos}.x} \leftarrow v.\text{pos}.x \\
\text{end for} \quad \triangleright \text{ reduce the temperature over iterations as} \\
\quad t \leftarrow \text{cool}(t); \quad \triangleright \text{ layout approaches a better configuration} \\
\text{end for} \\
\text{end function}

When adding a character on the stage (as seen in Algorithm 4), if they are the only one on the stage, we will introduce vertices for their targeted position, the audience
Algorithm 4 Pseudo-code for Adding Characters Method

\[ G \leftarrow (V,E); \]  \,
\[ \text{The graph contains all vertices and edges for onstage characters} \]
\[ \text{audience.y} \leftarrow \text{position} \]  \,
\[ \text{Default y position for audience front row} \]
\[ \text{function AddCharacters(charlist, targetlist)} \]
\[ \text{for char in charlist do} \]  \,
\[ V \leftarrow v_{\text{char}} \leftarrow (\text{target.x, target.y}) \]
\[ V \leftarrow v_{\text{char.aud}} \leftarrow (v_{\text{char.x}}, \text{audience.y}) \]  \,
\[ \text{only add when not already in } G \]
\[ E \leftarrow e_{\text{char.target}} \]  \,
\[ \text{strength based on character importance} \]
\[ E \leftarrow e_{\text{char.target@timestamp}} \]  \,
\[ \text{save temperature information for connection} \]
\[ \text{if OnStageCount} = 1 \text{ then} \]  \,
\[ \text{Adding the second character onstage} \]
\[ V \leftarrow v_{\text{centerpt}} \]
\[ \text{end if} \]
\[ E \leftarrow e_{\text{char.centerpt}} \]  \,
\[ \text{Only review the character vertices} \]
\[ \text{for } v \in V_{\text{char do}} \]  \,
\[ E \leftarrow e_{\text{char.v}} \]  \,
\[ \text{give stronger strength to edges if } v \in \text{charlist} \]
\[ \text{end for} \]  \,
\[ \text{also give stronger strength based on character relationship} \]
\[ \text{end for} \]
\[ \text{CHARACTERMOVE(all)} \]  \,
\[ \text{update strength of degrading edges} \]
\[ \text{end function} \]

(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 more than one character is 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 of people. It is only removed if either only one character or no charac-
ters remain on-stage. An additional edge will be included whose strength will be
directly proportional to the closeness of the relationship between the two characters
with respect to the play.

Also, if multiple characters are entering the stage at the same time, their connec-
tion 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.

Algorithm 5 Pseudo-code for Moving Character Method

function CHARACTERMOVE(char)
  \( v_{\text{char aud}} \leftarrow (\text{char}.x, \text{char}.y) \)
  for \( e_{\text{char, char}} \in E \) do
    \( \triangleright \) Only review edges between two chars
    if \( e_{\text{char, char}}.\text{strength} > \text{charRelationship} \) then
      \( \triangleright \) Cool if entered together
      \( e_{\text{char, char}}.\text{strength} \leftarrow \text{cool}(); \)
    end if
  end for
  for \( e_{\text{char, target}} \in E \) do
    \( \triangleright \) Only review edges between chars and targets
    if \( e_{\text{char, target}}.\text{char} = \text{char} \) then
      remove \( e_{\text{char, target}} \)
      \( \triangleright \) remove edge if char has new target
    else
      \( e_{\text{char, target}}.\text{strength} \leftarrow \text{cool}(); \)
      if \( e_{\text{char, target}}.\text{strength} = 0 \) then
        remove \( e_{\text{char, target}} \)
        \( \triangleright \) remove edge if has been there too long
      end if
    end if
  end for
  ADJUSTALL()
end function
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 5 and Figure 22). 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 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 6).

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
Figure 22: Adjustments from Force-Directed Graphs
Algorithm 6 Pseudo-code for Remove Character Method

function REMOVECHARACTER(char)
    for $e_{\text{char} \times \epsilon E}$ do ▷ Find edges tied to char being removed
        remove $e_{\text{char} \times \epsilon}$
    end for
    remove $v_{\text{char aud}}$
    remove $v_{\text{char}}$
    if $v_{\text{char target}.edges}$ is empty then ▷ If nothing else tied to this, remove it
        remove $v_{\text{char target}}$
    end if
    if $\text{OnStageCount} = 1$ then ▷ If leaving only one char onstage
        remove $v_{\text{centerpt}}$
    end if
end function

re-arrangement request (as seen in Algorithms 3 and 5). The strength of the connections 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 optimally based on the human-controlled character’s position and movements.

4.4.5 Forces

The main algorithm utilized 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 unmoveable, 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 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 strong
attraction to the audience, it forces the group of characters to form a semi-circle
facing the audience.

4.4.6 Validations

Validating the accomplishment of these requirements, we will need to measure
performance in three separate, but related, threads. The first is to validate the
positioning from randomized states of the play, which is done in this paper. Most of
the requirements can be measured during this experimentation. Next, the positioning
of the characters must be validated over the duration of an entire scene or play. This
test focuses on the oscillation-free arrangements and the strength of relationships over
Table 1: Attractive and Repellent Forces

<table>
<thead>
<tr>
<th>Force Type</th>
<th>Attractive</th>
<th>Repellent</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI Character</td>
<td>$</td>
<td>\delta</td>
</tr>
<tr>
<td>Human Character</td>
<td>$</td>
<td>\delta</td>
</tr>
<tr>
<td>Audience</td>
<td>$</td>
<td>\delta</td>
</tr>
<tr>
<td>Center Point</td>
<td>$</td>
<td>\delta</td>
</tr>
<tr>
<td>Target / Pawn</td>
<td>$\beta</td>
<td>\delta</td>
</tr>
</tbody>
</table>

$|\delta| = $ Separation Distance, $\alpha = $ Desired Separation Distance

time, in conjunction with human-controlled character movements both correct and incorrect. Finally, it is important to validate these positionings with respect to a human audience. Is the play reasonably blocked, and is it a believable arrangement of the characters?

This approach can be applied to predefined trajectories of characters, if the target destination is considered the current location of the character during the adjustment calculations. It will not address the orientation of the characters at each of these positions. However, our prior work with a rules engine can be utilized to adjust gazes appropriately.

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 20. The rules around facing direction are re-applied once the movements are completed since the force-directed graph approach does not handle facing directions.

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 \( 1 \) where we sum the overlaps of each character and divide by the length of the stage.

The second criteria, clustering, is used to ensure we are not clumping everyone too
\[
\sum_{i=0}^{\text{count}} \sum_{j:i+1}^{\text{count}} \frac{\text{char}[i].maxX - \text{char}[j].minX}{\text{stageLength}} \begin{cases} 
> 0; & \text{char}[i].maxX - \text{char}[j].minX \\
\leq 0; & 0 
\end{cases}
\]

(1)

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 \[2\] and \[3\] 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.

\[
\frac{\text{Max}_{\forall i}(\text{char}[i].x) - \text{Min}_{\forall i}(\text{char}[i].x)}{\text{stageLength}_X} \quad (2)
\]

\[
\frac{\text{Max}_{\forall i}(\text{char}[i].y) - \text{Min}_{\forall i}(\text{char}[i].y)}{\text{stageLength}_Y} \quad (3)
\]

We will look to minimize the occlusion equation and maximize the two clustering equations to determine quality of the spatial positioning.

4.5 User Studies

4.5.1 NLP versus BML

We presented 3D videos of both the hand-mapped production from Broadway in 1964 \[13\] \[76\] 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 \[23\]. 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.

1. Characters showed evidence of engaged listening
2. Characters appeared to perform suitable movements on cue
3. The pace of the performance was too fast
4. The pace of the performance was too slow
5. The use of the space on stage was appropriate
6. The blocking (positioning and timing of the characters) was appropriate
7. There was adequate variety in the staging positions of the characters
8. The characters’ movement onstage during the performance was believable in the context of the performance
9. The performance is free from distracting behavior that does not contribute to the scene

10. The arrangement of the performers appropriately conveys the mood of the scene

11. The character movements provide appropriate dramatic emphasis

12. There is adequate variety and balance in the use of the performance space

13. All visible behaviors appear to be motivated and coordinated within the scene

14. The characters were grouped to give proper emphasis to the right characters at the right time

15. The characters frequently covered or blocked each other from your point of view

16. The movements of the characters were consistent with the play

17. There was a great deal of random movement

18. The characters’ reactions to other characters were believable

19. Characters showed a lack of engagement when listening

20. The arrangement of the performers contradicts the mood of the scene

21. The more prominent characters in the scene were hidden or masked from your view

22. The characters were too close together

23. The characters were too far apart
Table 2: Correlation of Spatial-Temporal Questions

<table>
<thead>
<tr>
<th>Question #s</th>
<th>Spatial Component Covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ≈ ¬ 19</td>
<td>Engaged Listening</td>
</tr>
<tr>
<td>2 ≈ 6</td>
<td>Pace of Performance</td>
</tr>
<tr>
<td>16 ≈ ¬ 17</td>
<td>Appropriate Movement and Timing</td>
</tr>
<tr>
<td>3 ≈ ¬ 4</td>
<td>Consistent Movement</td>
</tr>
<tr>
<td>5 ≈ ¬ 24 ≈ 12</td>
<td>Space Usage</td>
</tr>
<tr>
<td>7 ≈ ¬ 22 ≈ ¬ 23</td>
<td>Variety and Closeness</td>
</tr>
<tr>
<td>9 ≈ ¬ 13</td>
<td>Motivated Movement</td>
</tr>
<tr>
<td>10 ≈ ¬ 20</td>
<td>Scene Mood</td>
</tr>
<tr>
<td>14 ≈ ¬ 21 ≈ 11</td>
<td>Character Emphasis</td>
</tr>
<tr>
<td>15 ≈ ¬ 25</td>
<td>Visible Characters</td>
</tr>
<tr>
<td>18 ≈ 8</td>
<td>Believable</td>
</tr>
</tbody>
</table>

24. The stage space was not utilized to its full potential

25. All characters were visible from your point of view throughout the scene

Each group viewed only one of the videos and answered the questions about the spatiotemporal reasoning included within the video (between groups experiment). The questions were presented in randomized order to the users after viewing the video. Above, the spatial-temporal related questions are listed that were presented to the study participants. The questions within the survey were intentionally asked two or more times with different phrasing to alleviate any bias presented in the wording of the question. The expected correlation between the questions can be seen in Table 2. 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 [62], and can be seen below.

- 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)
• Did anything in the production not make sense? What was it and why?

• Were there any special problems that need examining? What were they and why?

• Were there any particular strong, weak, or boring moments in the scene? What were they and why?

• Any other comments?

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.

4.5.2 NLP versus BML versus Rules versus FDGs

Additional user studies to qualitatively evaluate each component: Baseline, Natural Language Processor, Rules Engine, and Force-Directed Graphs. Studies will utilize the BlockWorld and shortened scenes to avoid user viewing fatigue. The previously created spatiotemporal questionnaire will be utilized to evaluate and compare each of the techniques.
4.5.3 Baseline versus FDGs for Human-Interaction

Additional user studies to qualitatively evaluate Human-interaction with Force-Directed Graphs. Studies will utilize the BlockWorld and users controlling one character within the scene for each of the techniques (Baseline, Natural Language Processor, Rules Engine, and Force-Directed Graphs). The previously created spatiotemporal questionnaire, with modifications, will be used to evaluate the scene and the incorporation of the human-controlled character.

4.6 Generalization

Describe approach for validating this approach will be generalizable to other play-scripts. We will identify the X types of plays by their typical organization. We will demonstrate the applicability of these techniques (Natural Language Processor, Rules Engine, and Force-Directed Graphs) to up to 10 of these types. Additional user studies or quantitative evaluations will be performed to prove generalization.
CHAPTER 5: EXPERIMENTATION AND DISCUSSION

5.1 BML Generation

For a 14 minute heavily annotated scene cut for Hamlet, 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 took a less than 100 line script and turned it into about 400 speech and spatial commands in BML. Even with this amount of time and effort, it probably only covered about 75% of the movement assumptions that an actor would utilize in performing the script, and no non-verbal behaviours. 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 with spatial implications.

This resulted in 400 BML and FML commands with similar spacing between character speech and act lines as exist in the play-script. 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 for a ten minute performance. 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. All of this effort required four hours and 12 minutes just for writing the BML and FML to support the four actions that were focused on.

5.2 Natural Language Processing to BML

We took the character traces from both our ground truth (hand-coded BML) and our new method and compared them. We want our new method to result in character positioning as close to our baseline as possible; however we do not want to penalize for being “close enough”. As you can see in Figure 24 we accomplished a relatively similar character trace over time with our new method. Digging deeper and comparing actual numerics behind these traces, we can see that we are typically very close to the baseline for our characters. We see that 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 ten minutes of script.

Issues include the fact that sequence does not always indicate the actor vs 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 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. Also, due to the lookups, most of this algorithm lends itself to being a pre-processor for the script and may not be the best of choices
Figure 24: Character Traces for Hamlet (first row), Horatio (second row), Gravedigger1 (third row), Gravedigger2 (fourth row); Column 1 are the baseline traces, Column 2 are traces from our method.
for a realtime script.

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.

Some 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). Also, 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, most of this algorithm lends itself to being a pre-processor for the script and may not be the best of choices for a real-time script.

5.3 Rules

We took the character traces from both our ground truth (hand-coded BML based on the Electronovision video [13]) 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 Figure 25, 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, downstage, center-stage, stage-right, and stage-left (as seen in the lower-right of Figure 25).

We found that our method was able to position the characters within 0.12 squares
Figure 25: Comparison of Character Traces for Position Over Time (ms) and Stage Grid Diagram

(Euclidean distance) of our baseline BML method and placed them correctly 88.9% of the time on the stage. 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 as can be seen in the lower-right of Figure 26. Here we found our results did not match as well (as seen
in Figure 26, 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 ground-truth vs our method. The ground-truth 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 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 15% for position and approximately 30% for gaze, even though it still incurred similar issues around duality of word meanings and pronouns found in our first experiment.

5.4 Force directed graphs

To validate our algorithm’s effectiveness, 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; 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 positionings within the play, and their realism.

5.4.1 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 27.

![Figure 27: Character Positioning Using Forces](image)

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 ($\beta$) 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 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 1 in the previous chapter.

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 3.
Table 3: Average Distances (in Feet) Between Characters and Pawns

<table>
<thead>
<tr>
<th>Number of Characters</th>
<th>Character Connected To:</th>
<th>Average Distance (in Drawing Units)</th>
<th>Average Distance (in Feet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>audience</td>
<td>370.63</td>
<td>2.47</td>
</tr>
<tr>
<td>1</td>
<td>human</td>
<td>526.30</td>
<td>3.51</td>
</tr>
<tr>
<td>1</td>
<td>target</td>
<td>496.80</td>
<td>3.31</td>
</tr>
<tr>
<td>2</td>
<td>audience</td>
<td>515.92</td>
<td>3.44</td>
</tr>
<tr>
<td>2</td>
<td>center</td>
<td>270.80</td>
<td>1.81</td>
</tr>
<tr>
<td>2</td>
<td>char</td>
<td>400.74</td>
<td>2.67</td>
</tr>
<tr>
<td>2</td>
<td>human</td>
<td>488.35</td>
<td>3.26</td>
</tr>
<tr>
<td>2</td>
<td>target</td>
<td>456.22</td>
<td>3.04</td>
</tr>
<tr>
<td>3</td>
<td>audience</td>
<td>449.66</td>
<td>3.00</td>
</tr>
<tr>
<td>3</td>
<td>center</td>
<td>355.34</td>
<td>2.37</td>
</tr>
<tr>
<td>3</td>
<td>char</td>
<td>414.75</td>
<td>2.77</td>
</tr>
<tr>
<td>3</td>
<td>human</td>
<td>662.09</td>
<td>4.41</td>
</tr>
<tr>
<td>3</td>
<td>target</td>
<td>448.98</td>
<td>2.99</td>
</tr>
<tr>
<td>4</td>
<td>audience</td>
<td>528.01</td>
<td>3.52</td>
</tr>
<tr>
<td>4</td>
<td>center</td>
<td>296.00</td>
<td>1.97</td>
</tr>
<tr>
<td>4</td>
<td>char</td>
<td>424.55</td>
<td>2.83</td>
</tr>
<tr>
<td>4</td>
<td>human</td>
<td>468.30</td>
<td>3.12</td>
</tr>
<tr>
<td>4</td>
<td>target</td>
<td>555.88</td>
<td>3.71</td>
</tr>
<tr>
<td>5</td>
<td>audience</td>
<td>419.66</td>
<td>2.80</td>
</tr>
<tr>
<td>5</td>
<td>center</td>
<td>378.33</td>
<td>2.52</td>
</tr>
<tr>
<td>5</td>
<td>char</td>
<td>529.04</td>
<td>3.53</td>
</tr>
<tr>
<td>5</td>
<td>human</td>
<td>466.43</td>
<td>3.11</td>
</tr>
<tr>
<td>5</td>
<td>target</td>
<td>611.72</td>
<td>4.08</td>
</tr>
<tr>
<td>6</td>
<td>audience</td>
<td>425.97</td>
<td>2.84</td>
</tr>
<tr>
<td>6</td>
<td>center</td>
<td>368.86</td>
<td>2.46</td>
</tr>
<tr>
<td>6</td>
<td>char</td>
<td>482.32</td>
<td>3.22</td>
</tr>
<tr>
<td>6</td>
<td>human</td>
<td>484.72</td>
<td>3.23</td>
</tr>
<tr>
<td>6</td>
<td>target</td>
<td>464.94</td>
<td>3.10</td>
</tr>
<tr>
<td>7</td>
<td>audience</td>
<td>436.75</td>
<td>2.91</td>
</tr>
<tr>
<td>7</td>
<td>center</td>
<td>348.79</td>
<td>2.33</td>
</tr>
<tr>
<td>7</td>
<td>char</td>
<td>469.19</td>
<td>3.13</td>
</tr>
<tr>
<td>7</td>
<td>human</td>
<td>502.33</td>
<td>3.35</td>
</tr>
<tr>
<td>7</td>
<td>target</td>
<td>428.91</td>
<td>2.86</td>
</tr>
<tr>
<td>8</td>
<td>audience</td>
<td>415.04</td>
<td>2.77</td>
</tr>
<tr>
<td>8</td>
<td>center</td>
<td>401.58</td>
<td>2.68</td>
</tr>
<tr>
<td>8</td>
<td>char</td>
<td>564.30</td>
<td>3.76</td>
</tr>
<tr>
<td>8</td>
<td>human</td>
<td>462.93</td>
<td>3.09</td>
</tr>
<tr>
<td>8</td>
<td>target</td>
<td>487.51</td>
<td>3.25</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Number of Characters</th>
<th>Character Connected To:</th>
<th>Average Distance (in Drawing Units)</th>
<th>Average Distance (in Feet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>audience</td>
<td>498.57</td>
<td>3.32</td>
</tr>
<tr>
<td>9</td>
<td>center</td>
<td>439.40</td>
<td>2.93</td>
</tr>
<tr>
<td>9</td>
<td>char</td>
<td>511.48</td>
<td>3.41</td>
</tr>
<tr>
<td>9</td>
<td>human</td>
<td>502.24</td>
<td>3.35</td>
</tr>
<tr>
<td>9</td>
<td>target</td>
<td>489.51</td>
<td>3.26</td>
</tr>
<tr>
<td>10</td>
<td>audience</td>
<td>482.81</td>
<td>3.22</td>
</tr>
<tr>
<td>10</td>
<td>center</td>
<td>376.99</td>
<td>2.51</td>
</tr>
<tr>
<td>10</td>
<td>char</td>
<td>483.21</td>
<td>3.22</td>
</tr>
<tr>
<td>10</td>
<td>human</td>
<td>480.38</td>
<td>3.20</td>
</tr>
<tr>
<td>10</td>
<td>target</td>
<td>521.40</td>
<td>3.48</td>
</tr>
<tr>
<td>11</td>
<td>audience</td>
<td>496.43</td>
<td>3.31</td>
</tr>
<tr>
<td>11</td>
<td>center</td>
<td>394.77</td>
<td>2.63</td>
</tr>
<tr>
<td>11</td>
<td>char</td>
<td>478.84</td>
<td>3.19</td>
</tr>
<tr>
<td>11</td>
<td>human</td>
<td>495.07</td>
<td>3.30</td>
</tr>
<tr>
<td>11</td>
<td>target</td>
<td>480.68</td>
<td>3.20</td>
</tr>
<tr>
<td>12</td>
<td>audience</td>
<td>476.52</td>
<td>3.18</td>
</tr>
<tr>
<td>12</td>
<td>center</td>
<td>403.22</td>
<td>2.69</td>
</tr>
<tr>
<td>12</td>
<td>char</td>
<td>477.78</td>
<td>3.19</td>
</tr>
<tr>
<td>12</td>
<td>human</td>
<td>498.51</td>
<td>3.32</td>
</tr>
<tr>
<td>12</td>
<td>target</td>
<td>493.62</td>
<td>3.29</td>
</tr>
</tbody>
</table>

5.4.2 Point in Time Evaluation

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 28. 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 28d. More work may be required to better balance the grouping arrangements of the characters.

More work needs to 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. Additional force manipulations may achieve better results than found during this particular experiment.

5.4.3 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

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 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
Figure 28: Character Positioning Using Force-Directed Graphs—Red=Human; Green=AI Character; Black=Pawn; Blue=Center Pt
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 for the human to move to during that moment, which does not coincide with the play-script.

To evaluate our approach, we ran numerous experiments as described above. We started with a baseline reading which utilized the hand-mapped blocking from the 1964 Hamlet production on Broadway. As can be seen in Table 4, we have some minor occlusions of the characters on the stage with that 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 15% along the depth of the stage).

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
Table 4: Experiment Results of Occlusion and Clustering Averaged Over Entire Scene

<table>
<thead>
<tr>
<th>Case #</th>
<th>Case Description (Including Accuracy of Human)</th>
<th>Avg Occlusion Frequency</th>
<th>Average Clustering X</th>
<th>Average Clustering Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Baseline All AI</td>
<td>3.6%</td>
<td>19.5%</td>
<td>14.6%</td>
</tr>
<tr>
<td>1</td>
<td>Baseline Human 90%</td>
<td>3.6%</td>
<td>19.1%</td>
<td>15.4%</td>
</tr>
<tr>
<td>2</td>
<td>Baseline Human 50%</td>
<td>2.9%</td>
<td>20%</td>
<td>14.7%</td>
</tr>
<tr>
<td>3</td>
<td>Baseline Human 10%</td>
<td>4.4%</td>
<td>30.9%</td>
<td>28.7%</td>
</tr>
<tr>
<td>4</td>
<td>Forces All AI</td>
<td>2.4%</td>
<td>16.8%</td>
<td>14.6%</td>
</tr>
<tr>
<td>5</td>
<td>Forces Human 90%</td>
<td>2.4%</td>
<td>16.8%</td>
<td>14.6%</td>
</tr>
<tr>
<td>6</td>
<td>Forces Human 50%</td>
<td>1.6%</td>
<td>20.4%</td>
<td>13.8%</td>
</tr>
<tr>
<td>7</td>
<td>Forces Human 10%</td>
<td>2.4%</td>
<td>20.8%</td>
<td>14%</td>
</tr>
</tbody>
</table>

on the stage. 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 remains excluded in the hard-coded blocking for the AI characters.

Considering the scene we utilized has at most 3 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 when we utilize the force-directed graphs. However, with the hard-coded AI character blocking, 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.

5.5 User Studies

Now that we have quantitatively evaluated each of our techniques, we must perform qualitative analyses on a user’s perception of the performance.

5.5.1 NLP versus BML

Over 748 participants attempted the survey, with only 214 completing it due to the checks put in place within the survey.

The study included 214 participants who were asked to evaluate the spatial aspects of a recorded video, as described in the Approach section. 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); the other viewing a natural language processing interpretation from the play-script (NLP). There were 108 participants viewing the Baseline video, and 106 participants viewing the NLP video.
We combined several questions into a composite question during our analysis. These were combined based on our expected correlation of the questions, which was intentionally done to avoid bias with the phrasing of the questions. These correlations can be seen in Table 2 in the Methodology chapter. All questions were answered by all 214 participants included in this study.

5.5.1.1 T-Tests

This experiment provided an estimated power to detect a medium effect (d=.5) of >.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=.97) and the NLP group (M=2.32, SD=1.07), t(212)=2.74, p=.007 for the question "There was a great deal of random movement." The 95% confidence interval for the difference between the means was .11 to .66, so the minimum expected difference would be about a tenth of a point on a 5 point scale. This reflects a significant difference, but is not significant from a practical perspective. 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=.80) and the NLP group (M=2.58, SD=1.16), t(186)=2.11, p=.036. The 95% confidence interval for the difference between the means was -.56 to -.02, so the minimum expected difference would be about two one-hundredths of a point on a 5 point scale. This reflects a significant difference, but is not significant from a practical perspective. 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=.71) and NLP group (M=3.58, SD=.68), t(212)=2.36, p=.019 for the combined question of "Consistent Movement." The 95% confidence interval for the difference between the means was -.41 to -.04, so the minimum expected difference would be about two one-hundredths of a point on a 5 point scale. This reflects a significant difference, but is not significant from a practical perspective. 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=.58) and the NLP group (M=3.22, SD=.75), t(198)=2.47, p=.014. The 95% confidence interval for the difference between the means was .05 to .41, so the minimum expected difference would be about five one-hundredths of a point on a 5 point scale. This reflects a significant difference, but is not significant from a practical perspective. 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.

5.5.1.2 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 2 in the previous 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=.3) of >.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=.98), r(212)=+.53, p<.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)=+.59, p<.001. "The arrangement of the performers appropriately conveys the mood of the scene" (M=3.26, SD=.98) is also shown as directly correlated to the CharMvmtBe-
lievable question, \( r(212) = +.51, p < .001 \).

Additionally, both the question "The movements of the characters were consistent with the play" (\( M=3.45, SD=.89 \)), \( r(212) = +.50, p < .001 \), and "The characters’ reactions to other characters were believable" (\( M=3.08, SD=1.11 \)), \( r(212) = +.505, p < .001 \), 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" (\( M=3.03, SD=1.06 \)), \( r(212) = +.542, p < .001 \). This shows that variety and balance are directly related to how the space on stage is used.

### 5.5.1.3 Multi-Factors

Using our demographic questions that we presented to our participants, we analyzed the effects of multiple factors to our results: region, age, gender, culture, theatre frequency, gaming frequency, theatre familiarity, scene familiarity, and employment. We consolidated the US states where the participants live into four regions based on the US Census’ region definitions of Northeast, South, Midwest, West, and Other. Similarly, we combined the employment status questions to provide groupings that included multiple choices, such as "Student and Full-Time Employed” instead of separate groupings on whether the participant was a student or not. Age was broken up by units of 10, starting at age 18-20, 21-29, 30-39, 40-49, 50-59, 60 or older. Culture
was represented loosely by differing countries to determine any effect of backgrounds with spatial evaluations.

5.5.1.4 Regions

When evaluating the regions, we had five groups, and a power to detect effects of a moderate size \( \eta^2 = .06 \) that is moderate \( (.70) \). The question "The character movements provide appropriate dramatic emphasis" had a significant interaction effect \( F(3, 205) = 2.66, p = .049, \partial \eta^2 = .037 \). To interpret the interaction, simple effects tests were computed for the Baseline group \( (M=3.14, SD=.24) \) versus the NLP group \( (M=2.40, SD=.25) \) independently. Although the means suggest that the Northeast \( F(1, 205) = 4.58, p = .034 \) region is significantly different in evaluating this question, these represent a difference of at least five one-hundredths of a point on a 5 point scale, and is not significant from a practical perspective.

Region also played an interaction effect with the question "There was a great deal of random movement" \( F(3, 205) = 3.13, p = .027, \partial \eta^2 = .044 \). Here, we found that the Mid-West \( F(1, 205) = 16.12, p < .001, \partial \eta^2 = .073 \) region is significantly different, resulting in at least half a point (.54) to 1.58 points on a 5 point scale difference, with the Baseline group receiving a higher amount of agreement with the question \( (M=3.06, SD=.18) \) than the NLP group \( (M=2.00, SD=.20) \). This is interesting to note, and may be due to the way that the Hamlet character turns frequently when speaking to the GraveDigger and Horatio at the end of the scene with the Baseline performance.
5.5.1.5 Age Groups

When reviewing the participants’ age groups, we had six groups and a power to detect effects of a moderate size ($\eta^2 = .06$) that is moderate (.65). We found that “The pace of the performance was too slow” had a significant interaction effect $F(5, 202)=3.38$, $p=.006$, partial $\eta^2 = .077$. To interpret this interaction, simple effects tests were computed for the Baseline group ($M=4.04, SD=.20$) versus the NLP group ($M=3.00, SD=.24$) independently. Looking at the means suggest that 30-39 year old participants $F(1, 202)=11.16, p=.001$, partial $\eta^2 = .052$ are significantly different in evaluating this question, with a difference of at least .42 to 1.65 points on a 5-point scale, with the Baseline group agreeing with this question more. We speculate that the performance appears slower for the Baseline group because more events occur during that video (17 minutes in length) than in the NLP video (14 minutes in length), however it is unclear why this age group would notice it more than the other groups.

We also saw that age played an interaction effect with the question “The characters were grouped to give proper emphasis to the right characters at the right time” $F(5, 202)=3.17$, $p=.009$, partial $\eta^2 = .073$. Here, we found that 40-49 year olds $F(1, 202)=9.81, p=.002$, partial $\eta^2 = .046$ is significantly different, resulting in at least .35 to 1.54 point difference on a 5 point scale. The Baseline group ($M=3.79, SD=.19$) agreed more than the NLP group ($M=2.85, SD=.23$), reflecting that this age group noticed the grouping to emphasize the correct characters more often in the Baseline video.
5.5.1.6 Gender

Next, we evaluated any effects due to gender on the qualitative analysis, with a power to detect effects of moderate size ($\eta^2 = .06$) that is high (.87). We found a significant interaction with the question "The performance is free from distracting behavior that does not contribute to the scene" $F(1, 210)=4.20$, $p=.042$, partial $\eta^2 = .020$. To interpret this interaction, simple effects tests were computed for the Baseline group and NLP group, finding that Males agreed with this question more when viewing the NLP video ($M=3.77$, $SD=.18$) than the Baseline video ($M=3.23$, $SD=.17$). This results in at least a .04 to 1.04 point difference on a 5 point scale.

Like with the regional difference with the random movement question, it may be due to Hamlet’s performance near the end of the Baseline video, yet is unclear why the males would pick up on this more than the females. The question "All visible behaviors appear to be motivated and coordinated within the scene" also provides a significant difference based on gender $F(1, 210)=4.05$, $p=.045$, partial $\eta^2 = .019$. Here, we see that Females $F(1, 210)=1.06$, $p=.303$, partial $\eta^2 = .005$, are more likely to agree with this question for the Baseline ($M=3.58$, $SD=.11$) than the NLP ($M=3.27$, $SD=.11$) videos.

5.5.1.7 Culture

Looking at the participants’ cultural background also provided some key areas of differentiation, with a power to detect effects of moderate size ($\eta^2 = .06$) that is moderate(.70). The question "The characters’ movement onstage during the performance was believable in the context of the performance" had a significant interac-
tion $F(3, 205)=2.83$, $p=.040$, partial $\eta^2=.040$. It was found English-backgrounds (not to be confused with American backgrounds) had a significant difference from other groups $F(1, 205)=5.11$, $p=.025$, partial $\eta^2=.024$, with the Baseline videos being rated higher (greater agreement with this question) ($M=3.69$, $SD=.29$) than the NLP videos ($M=2.63$, $SD=.37$). Once again, this is an expected result that is only surprising in that the English-background participants noticed more than the other participants.

5.5.1.8 Gaming Frequency

When looking at how frequently a participant spent in playing video games, we saw some differences in the evaluation of the performances. We had a power to detect effects of moderate size ($\eta^2=.06$) that is moderate (.75). The question regarding “The pace of the performance was too fast” had a significant interaction effect $F(4, 204)=2.57$, $p=.039$, partial $\eta^2=.048$. Participants that spent 1 to 3 hours a week playing games found that the Baseline video was faster ($M=2.31$, $SD=.15$) compared to the NLP video ($M=1.75$, $SD=.16$). This corresponds with the expectation that the Baseline video had more action and therefore had a faster pace than the NLP video, despite the Baseline video lasting longer.

It was also noticed that the question ”The characters frequently covered or blocked each other from your point of view” resulted in a significant interaction $F(4, 204)=3.76$, $p=.006$, partial $\eta^2=.069$. This time, the 4 to 10 hour group of gamers were significantly different $F(1, 204)=14.91$, $p<.001$, partial $\eta^2=.068$. These gamers said that the NLP video ($M=3.55$, $SD=.21$) had more characters blocking each other than the
Baseline video (M=2.38, SD=.23). This is another expected result, although why this particular group of gamers noticed it over the others is unclear.

5.5.1.9 Theatre Familiarity

When asked how familiar a participant was with theatre, there were some differences in the evaluation of the performances. We had a power to detect effects of moderate size (eta$^2$=.06) that is moderate (.70). The question regarding "The characters were too close together" had a significant interaction effect $F(3, 205)=3.47$, $p=.017$, partial eta$^2$=.048. Participants that were very familiar with this scene in Hamlet were significantly different $F(1, 206)=8.41$, $p=.004$, partial eta$^2$=.039. These participants said that the NLP video (M=3.83, SD=1.17) had characters closer together than the Baseline video (M=2.25, SD=.89). This corresponds with the expectation that the NLP video did not space characters as well as the Baseline video.

It was also noticed that the question "The stage space was not utilized to its full potential" resulted in a significant interaction $F(3, 205)=2.93$, $p=.035$, partial eta$^2$=.041. The participants that were very familiar with this scene in Hamlet were significantly different $F(1, 206)=8.36$, $p=.004$, partial eta$^2$=.039. These participants said that the NLP video (M=5.00, SD=0.00) did not utilize the stage to its full potential versus the Baseline video viewers (M=3.25, SD=1.04). This also highlights the fact that the NLP video did little to maximize the use of the space on stage.

5.5.1.10 Hamlet Familiarity

When asked how familiar a participant was with theatre, there were some differences in the evaluation of the performances. We had a power to detect effects of
moderate size \((\eta^2 = .06)\) that is moderate \((.70)\). The question regarding "R" had a significant interaction effect \(F(3, 205) = 3.46, p = .017\), partial \(\eta^2 = .048\). Participants that spent

5.5.2 NLP versus BML versus Rules versus FDGs

Additional user studies to qualitatively evaluate each component: Natural Language Processor, Rules Engine, and Force-Directed Graphs. Studies will utilize the BlockWorld and shortened scenes to avoid user viewing fatigue. The previously created spatiotemporal questionnaire will be utilized to evaluate and compare each of the techniques.

5.5.3 Baseline versus FDGs for Human-Interaction

Additional user studies to qualitatively evaluate Human-interaction with Force-Directed Graphs. Studies will utilize the BlockWorld and users controlling one character within the scene for each of the techniques (Baseline, Natural Language Processor, Rules Engine, and Force-Directed Graphs). The previously created spatiotemporal questionnaire, with modifications, will be used to evaluate the scene and the incorporation of the human-controlled character.

5.6 Generalization

Describe approach for validating this approach will be generalizable to other play-scripts. We will identify the X types of plays by their typical organization. We will demonstrate the applicability of these techniques (Natural Language Processor, Rules Engine, and Force-Directed Graphs) to up to 10 of these types. Additional user studies or quantitative evaluations will be performed to prove generalization.
CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

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

This 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.

We showed that we can retrieve spatial directions from a play-script. We were able to reduce the technical expertise required to write the script by over four hours. We were able to accomplish similar, albeit not exact, movements without requiring any technical BML or FML expertise from the author. We reduced our authoring time by four hours from our prior work by utilizing the natural language processing of the annotated play-script.

The results of our experiments show that adding rules helps with a better blocking of the play from a spatial perspective. It confirms our hypothesis that adding rules
to control the spatial movements of our characters can more fully encapsulate the decisions actors and directors make when performing a play. We improved our blocking accuracy for both position and gaze with our rules-based approach. This work has focused on the theatre; however many of the rules are also applicable to other applications of spatial positioning, such as games and virtual worlds.

We showed the ability to use of force-directed graphs to arrange characters on a stage based on multiple criteria—concurrent entrances, audience visibility, and so forth. We showed that we can realistically position characters through a single scene, utilizing time-based relationships, such as target relationships and paired character entrances.

These methods better incorporated the unplanned movements of a human-controlled character by providing an on-the-fly adjustment module for our existing rules engine process. We showed that force-directed graphs can create a balanced and centralized grouping of characters on the stage, and maintain relatively consistent distances between characters versus other connections on the stage. The human-controlled character is more tightly integrated with the AI characters onstage, despite its incorrect movements, yet maintains its play-script integrity.

We evaluated user perception of each of these components and found...

We showed that these techniques can be applied to other play-scripts...

6.2 Future Work

This work does not apply the optimizations of theatre seating visibility (similar to multiple camera angles in television and movies) at this time, but could be considered
for future work.

Put future work here that is not part of the final dissertation document
REFERENCES


APPENDIX A: LACK OF SPATIAL INDICATORS IN HAMLET

A 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 [80]. 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 75% accuracy for character positions.

Next, we targeted the basic rules and guidelines that actors and directors use to control movement on the stage [85]. 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 which 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 Gielgud in 1964 on Broadway [13]. We utilized the script as written by Shakespeare [69], as well as the Electronovision video [13] of Richard Burton in Sir 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.
B 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[98], Brooks’[8], and Kollar et al.’s[40] 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[12] 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.[11]. 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 dis-
cussed in Platt, Cristianini, and Shawe-Taylor’s[63]. 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 [68], 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 [48] 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[18], Zhang, Zhou, and Aw’s[104], and Sun, Zhang, and Tan’s[77] 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[14] and Shen, Kruijff, and Klakow’s[71] 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[46] 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 [50]. 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 [100], 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. [15] discusses common feature categories, such as parts of speech (POS), voice of the sentence, and stemmed root words, while Culotta and Sorensen [18] mention word n-grams, capitalization, and conjunctions (or merging) of many features. Furnkranz [26] found that using n-grams of length 2 or 3 improved classification over longer n-grams. Forman [23] 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.

C Approach

In order to have a baseline to train against, we took the Electronovision video [13] of the production of Hamlet on Broadway in 1964 and mapped all the movement of the characters for each line of the play-script [69]. 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) [53]. 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,— 69

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 29. 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 29: 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 30. 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.

<table>
<thead>
<tr>
<th>Fighting</th>
<th>Gestures</th>
<th>Locomotion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fighting*</td>
<td>Point*</td>
<td>Walk*</td>
</tr>
<tr>
<td>Pushing*</td>
<td>Gesture*</td>
<td>Run*</td>
</tr>
<tr>
<td>Handle Object</td>
<td>Nod*</td>
<td></td>
</tr>
<tr>
<td>Hand Object*</td>
<td>Raise Arm*</td>
<td></td>
</tr>
<tr>
<td>Pickup Object*</td>
<td>Wave*</td>
<td></td>
</tr>
<tr>
<td>Throw Object*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Posture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jump*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lie Down*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sit*</td>
<td>Dig*</td>
<td></td>
</tr>
<tr>
<td>Stand*</td>
<td>Turn*</td>
<td></td>
</tr>
<tr>
<td>Kneel*</td>
<td>Climb*</td>
<td></td>
</tr>
<tr>
<td>Gaze</td>
<td>Kick*</td>
<td></td>
</tr>
</tbody>
</table>

Figure 30: 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 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.

D 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 5. This was due to having such a large portion of the training set be-
Table 5: 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.

<table>
<thead>
<tr>
<th>Movement</th>
<th>Ratio of Neg:Pos In- stances</th>
<th>Recall</th>
<th>Precision</th>
<th>F1-score</th>
<th>F0.5-score</th>
<th>Matthews Correlation Coefficient</th>
<th>Training Size</th>
<th>Recall</th>
<th>Precision</th>
<th>F1-score</th>
<th>F0.5-score</th>
<th>Matthews Correlation Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>1.8:1</td>
<td>0.017</td>
<td>0.455</td>
<td>0.033</td>
<td>0.075</td>
<td><strong>0.029</strong></td>
<td>1806</td>
<td>0.917</td>
<td>0.338</td>
<td>0.493</td>
<td>0.386</td>
<td>0.038</td>
</tr>
<tr>
<td>Big</td>
<td>2.4:1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1806</td>
<td>0.917</td>
<td>0.338</td>
<td>0.493</td>
<td>0.386</td>
<td>0.038</td>
</tr>
<tr>
<td>Gestures</td>
<td>21.4:1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>153</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Object</td>
<td>25.1:1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>153</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Locomot.</td>
<td>5.1:1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>153</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Gaze</td>
<td>7.3:1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>153</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Pointing</td>
<td>50.7:1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>153</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Posture</td>
<td>8.5:1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>153</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Fighting</td>
<td>198.1:1</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
ing 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 29). Forman [23] 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 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 5.

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 6. 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 mentioned in their work with different n-grams for classifications.

As Forman 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 n-gram 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 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 7.
Table 6: Highlights the Performance of Different N-Grams on Classifying the Different Movement Types on a 2:1

<table>
<thead>
<tr>
<th>Movement Type</th>
<th>n-gram</th>
<th>Machine Learning Algorithm</th>
<th>tp</th>
<th>fn</th>
<th>fp</th>
<th>tn</th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F1-score</th>
<th>F.5-score</th>
<th>Matthews Correlation Coefficient</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any Mvmt</td>
<td>1</td>
<td>MaxEnt</td>
<td>508</td>
<td>46</td>
<td>979</td>
<td>120</td>
<td>0.376</td>
<td>0.917</td>
<td>0.338</td>
<td>0.493</td>
<td>0.386</td>
<td>0.038</td>
<td></td>
</tr>
<tr>
<td>Any Mvmt</td>
<td>2</td>
<td>MaxEnt</td>
<td>505</td>
<td>49</td>
<td>988</td>
<td>129</td>
<td>0.379</td>
<td>0.912</td>
<td>0.338</td>
<td>0.493</td>
<td>0.387</td>
<td>0.041</td>
<td>↑</td>
</tr>
<tr>
<td>Any Mvmt</td>
<td>3</td>
<td>MaxEnt</td>
<td>502</td>
<td>52</td>
<td>987</td>
<td>130</td>
<td>0.378</td>
<td>0.906</td>
<td>0.337</td>
<td>0.491</td>
<td>0.386</td>
<td>0.034</td>
<td>↓</td>
</tr>
<tr>
<td>Any Mvmt</td>
<td>4</td>
<td>MaxEnt</td>
<td>503</td>
<td>51</td>
<td>987</td>
<td>130</td>
<td>0.379</td>
<td>0.908</td>
<td>0.338</td>
<td>0.492</td>
<td>0.386</td>
<td>0.037</td>
<td>↑</td>
</tr>
<tr>
<td>Any Mvmt</td>
<td>5</td>
<td>MaxEnt</td>
<td>501</td>
<td>53</td>
<td>984</td>
<td>133</td>
<td>0.379</td>
<td>0.904</td>
<td>0.337</td>
<td>0.491</td>
<td>0.386</td>
<td>0.035</td>
<td>↓</td>
</tr>
<tr>
<td>Gestures</td>
<td>1</td>
<td>Rand Forest</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>3273</td>
<td>0.985</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Gestures</td>
<td>2</td>
<td>Boosting</td>
<td>1</td>
<td>50</td>
<td>36</td>
<td>3237</td>
<td>0.974</td>
<td>0.020</td>
<td>0.027</td>
<td>0.023</td>
<td>0.025</td>
<td>0.010</td>
<td>↑</td>
</tr>
<tr>
<td>Gestures</td>
<td>3</td>
<td>Boosting</td>
<td>6</td>
<td>45</td>
<td>229</td>
<td>3044</td>
<td>0.918</td>
<td>0.118</td>
<td>0.026</td>
<td>0.042</td>
<td>0.030</td>
<td>0.023</td>
<td>↑</td>
</tr>
<tr>
<td>Gestures</td>
<td>4</td>
<td>MaxEnt</td>
<td>5</td>
<td>46</td>
<td>114</td>
<td>3159</td>
<td>0.952</td>
<td>0.098</td>
<td>0.042</td>
<td>0.059</td>
<td>0.047</td>
<td>0.042</td>
<td>↑</td>
</tr>
<tr>
<td>Gestures</td>
<td>5</td>
<td>MaxEnt</td>
<td>5</td>
<td>46</td>
<td>111</td>
<td>3162</td>
<td>0.953</td>
<td>0.098</td>
<td>0.043</td>
<td>0.060</td>
<td>0.049</td>
<td>0.043</td>
<td>↑</td>
</tr>
<tr>
<td>Locomotion</td>
<td>1</td>
<td>MaxEnt</td>
<td>202</td>
<td>10</td>
<td>2359</td>
<td>267</td>
<td>0.165</td>
<td>0.953</td>
<td>0.079</td>
<td>0.146</td>
<td>0.097</td>
<td>0.048</td>
<td></td>
</tr>
<tr>
<td>Locomotion</td>
<td>2</td>
<td>MaxEnt</td>
<td>202</td>
<td>10</td>
<td>2351</td>
<td>275</td>
<td>0.168</td>
<td>0.953</td>
<td>0.079</td>
<td>0.146</td>
<td>0.097</td>
<td>0.050</td>
<td>↑</td>
</tr>
<tr>
<td>Locomotion</td>
<td>3</td>
<td>MaxEnt</td>
<td>202</td>
<td>10</td>
<td>2379</td>
<td>247</td>
<td>0.158</td>
<td>0.953</td>
<td>0.078</td>
<td>0.145</td>
<td>0.096</td>
<td>0.043</td>
<td>↓</td>
</tr>
<tr>
<td>Locomotion</td>
<td>4</td>
<td>MaxEnt</td>
<td>201</td>
<td>11</td>
<td>2347</td>
<td>279</td>
<td>0.169</td>
<td>0.948</td>
<td>0.079</td>
<td>0.146</td>
<td>0.097</td>
<td>0.047</td>
<td>↑</td>
</tr>
<tr>
<td>Locomotion</td>
<td>5</td>
<td>MaxEnt</td>
<td>201</td>
<td>11</td>
<td>2346</td>
<td>280</td>
<td>0.169</td>
<td>0.948</td>
<td>0.079</td>
<td>0.146</td>
<td>0.097</td>
<td>0.047</td>
<td>↑</td>
</tr>
<tr>
<td>Feature Set</td>
<td>n-gram</td>
<td>Machine Learning Algorithm</td>
<td>tp</td>
<td>fn</td>
<td>fp</td>
<td>tn</td>
<td>Accuracy</td>
<td>Recall</td>
<td>Precision</td>
<td>F1-score</td>
<td>P0.5-score</td>
<td>Matthews Correlation Coefficient</td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------</td>
<td>----------------------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----------</td>
<td>--------</td>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
<td>--------------------------------</td>
<td></td>
</tr>
<tr>
<td>Text Only</td>
<td>1</td>
<td>RF</td>
<td>376</td>
<td>997</td>
<td>178</td>
<td>3</td>
<td>0.703</td>
<td>0.321</td>
<td>0.597</td>
<td>0.418</td>
<td>0.510</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>POS BoW Only</td>
<td>1</td>
<td>RF</td>
<td>396</td>
<td>980</td>
<td>158</td>
<td>3</td>
<td>0.681</td>
<td>0.285</td>
<td>0.536</td>
<td>0.372</td>
<td>0.456</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>All Features</td>
<td>3</td>
<td>RF</td>
<td>550</td>
<td>1112</td>
<td>4</td>
<td>3</td>
<td>0.668</td>
<td>0.007</td>
<td>0.444</td>
<td>0.014</td>
<td>0.034</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>POS BoW &amp; Text</td>
<td>3</td>
<td>RF</td>
<td>550</td>
<td>1112</td>
<td>4</td>
<td>3</td>
<td>0.668</td>
<td>0.007</td>
<td>0.444</td>
<td>0.014</td>
<td>0.034</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>POS BoW &amp; Other</td>
<td>2</td>
<td>Boost</td>
<td>554</td>
<td>1117</td>
<td>0</td>
<td>1</td>
<td>0.668</td>
<td>0.007</td>
<td>0.444</td>
<td>0.014</td>
<td>0.034</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>POS BoW, Text, Other</td>
<td>1</td>
<td>SVM</td>
<td>554</td>
<td>1117</td>
<td>0</td>
<td>1</td>
<td>0.668</td>
<td>0.007</td>
<td>0.444</td>
<td>0.014</td>
<td>0.034</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
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 [74] 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 31.

E 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
(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 31: ROC Curves Samples for Techniques Utilized;  
Red=SVM; Green=Maximum Entropy; Blue=Boosting; Magenta=Random Forests
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

**BML** Behavior Markup Language—An XML description language for controlling the verbal and nonverbal behavior of virtual characters

**Cue** Signal or command given to indicate another action should follow

**Director** The role responsible for the overall artistic vision of a production or play

**DownStage** Part of the stage that is closest to the audience.

**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

**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 which 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 which can be in one of seven different arrangements. 


**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.
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, and TIAA-CREF.

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.com, and plans to complete her degree in 2015. She is a member of the Association for Computing Machinery, Golden Key Honor Society, and Phi Kappa Phi Honor Society.