INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Stories, Agents and Videotapes: Agents that make up stories

Joana Flipa Gouveia de Almeida

Dissertation submitted to obtain the Master Degree in
Information Systems and Computer Engineering

Jury

Chairman:	Prof. António Manuel Ferreira Rito da Silva
Supervisor:	Prof[a] Ana Maria Severino de Almeida e Paiva
Member:	Prof[a] Isabel Machado Alexandre

May 2012

Acknowledgements

I would like to thank Prof. Ana Paiva for starting me on this work and for providing encouragement throughout its development.

A huge thank you to Guilherme Raimundo, who was an invaluable source of critique and ideas on the later stages of development and testing. Without him, this work would not be here today as it is now. Also, a big thank you for João Dias and Samuel Mascarenhas for helping me out with the characters' Artificial Intelligence.

Also, my thanks go out to Hélio Mascarenhas for providing me with the 3D models for the characters used in our game.

Last, but not least, a shout out to my fellow colleagues at GAIPS, which supported me, endured the many questions and doubts I had, and helped me develop this work. Special mention goes out to Henrique Campos, Henrique Reis and André Carvalho.

Porto Salvo, May 14th 2012
Joana Filipa Gouveia de Almeida

To my parents, who supported me through all my life and gave up on a lot of things to get me where I am. For my grandparents, who were just as instrumental in my education (I hope you can see me now grandma! This one is for you!).
To my childhood friends, Sílvia and Joana, whom I kept close to my heart, and hope to continue doing so for a long time.
To all the great friends I made while studying at IST. Academic life without you guys would not have been the same!
I hope I made you all proud.

Resumo

Com o nosso trabalho queremos estudar câmeras virtuais e como elas providenciam aos jogadores uma maneira significativa para se ligarem aos muitos e diversos tipos de mundos virtuais e jogos existentes.

Explicaremos como pesquisámos câmeras virtuais em muitos domínios e com muitos propósitos, com o intuito de construir um sistema nosso onde podemos incorporar essas ideias.

Tudo isto para estudar o impacto e influência que a câmera tem nas percepções das pessoas sobre estes mundos virtuais, desde a consciência da ocorrência de eventos à compreensão das relações entre as personagens.

Pelos nosso resultados, conseguimos identificar a câmera em Terceira Pessoa como uma forte candidata para dar aos jogadores a melhor experiência possível em sistemas de narrativa interactiva centrada em personagens e jogos e com personagem controlável, ao providenciar o melhor compromisso entre a sensibilização ao mundo que o rodeia e a ligação que tem com a personagem que controla.

Acreditamos que este é um começo para compreender o envolvimento que as pessoas têm com estes sistemas e um primeiro passo para lhes providenciar uma melhor experiência que os deixe envolverem-se na história com maior facilidade.

Abstract

With our work, we aim to study virtual cameras and how they provide a meaningful way for players to connect to many and diverse types of virtual worlds and games alike.

We will explain how we researched virtual cameras in many fields and with many purposes, with the intent of building a system of our own and incorporate these ideas.

All this to study the impact and influence the camera has on people's perceptions of these virtual worlds, from event awareness to understanding inter-character relationships.

Through our results we were able to pinpoint the Third Person Camera as a strong candidate in leading the best possible experience on character-controlling character-based interactive storytelling systems and games, by providing the best compromise between world awareness and player attachment to the character she controls.

We believe this is a start to understanding people's engagement with these systems and a first step to providing them with a better experience that lets them get immersed in the story with greater ease.

Palavras Chave
Keywords

Palavras Chave

Câmera Virtual
Narrativa Interactiva
Narrativa Centrada em Personagens
Jogos
Compreensão de História
Controlo de Personagens

Keywords

Virtual Camera
Interactive Storytelling
Character-Based Storytelling
Games
Story Comprehension
Character Control

Contents

Bibliography **ii**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Problem . . . 2
1.3 Document Outline . . . 3

2 Related Work **5**
2.1 Camera in 3D Virtual Environments . . . 5
2.2 Interactive Storytelling . . . 7
2.2.1 Core of Progression . . . 8
2.2.2 Planning . . . 11
2.3 Camera in Interactive Storytelling . . . 16
2.4 Camera in Games . . . 17
2.5 Camera and Types of User Control . . . 20
2.6 Critical Analysis . . . 23
2.6.1 Cameras . . . 24
2.6.2 Player Control . . . 25
2.6.3 Interactive Storytelling Building . . . 26

3 The Game and Camera Model **29**
3.1 The World . . . 29
3.2 The Story . . . 30
3.3 The Characters . . . 30
3.4 The Camera . . . 32
3.4.1 Camera Basics . . . 32
3.4.2 First Person Camera . . . 35
3.4.3 Third Person Camera . . . 35
3.4.4 Godlike or Top-Down Camera . . . 36

4 Implementation of our System **39**
4.1 The Game's Architecture . . . 39
4.1.1 Creating the Game World . . . 39
4.1.2 The Camera and User Interaction . . . 42

4.1.3 The Characters' Minds . . . 43
4.1.4 Bringing it all together . . . 47
4.2 An Example Scenario . . . 49

5 Conceal **53**
5.1 The Setting . . . 53
5.1.1 The Characters and their Relationships and Goals . . . 53
5.1.2 The Objects . . . 56
5.2 The scene . . . 57

6 Evaluation **61**
6.1 Preliminary Tests . . . 61
6.1.1 Feedback and Changes . . . 62
6.2 Evaluation of the Camera Model . . . 62
6.2.1 Evaluation Procedure . . . 62
6.2.2 Analyzing the Answers . . . 63
6.2.3 Results . . . 64
6.3 Conclusions . . . 69

7 Conclusion **73**
7.1 Future Work . . . 74

Bibliography **75**

A Experiment's Questionnaire **79**

List of Figures

2.1 Storytelling Systems and Games that Feature the First or Third Person Camera. From left to right we have *Mass Effect 3* (Third Person), *Dark Messiah of Might and Magic* (First Person) and *Façade* (First Person). 21
2.2 Storytelling Systems and Games that Feature the Fixed Camera. From left to right, top to bottom we have *The Secret of Monkey Island*, *Grim Fandango*, *Alone in the Dark* and *Resident Evil 3: Nemesis*. 22
2.3 Storytelling Systems and Games that Feature the Godlike Camera. From left to right, top to bottom we have *The Sims*, *Icewind Dale 2*, *Black & White* and *Medieval: Total War*. 23
2.4 Storytelling Systems and Games that feature more than one type of Camera. From left to right we have *SpellForce: The Order of Dawn* (Godlike), *Dragon Age: Origins* (Godlike) and *The Elder Scrolls V: Skyrim* (Third Person). 24
2.5 Relation between Camera Types and the kind of engagement players get from them. 25

3.1 A decomposition of the model of our characters' planning. 31

4.1 The main components in a Unity3D game's architecture. 40
4.2 The main components of Conceal's architecture. 40
4.3 The main components of a Conceal character. 41
4.4 The main components of a Conceal interactable item. 42
4.5 The main components of a Conceal decorative item. 42
4.6 The main components of the Conceal camera. 43
4.7 The three main components of our system's architecture: the world in Unity3D, the world simulation in ION and the agents' minds in FAtiMA. 48
4.8 ION's update loop. 48
4.9 How the the three main components of Conceal's architecture exchange information. 50

5.1 The models of the characters in the final game: Arndís the Bride, Ragi the Groom and Vilmundr, the Bride's Father. 54
5.2 Relationships between the characters in Conceal. 55
5.3 The models of the gift items in the final game: the Dress, the Box of Jewels and the Land Titles. 56
5.4 The models of the prize items in the final game: the Crest, the Furs and the Sword. 57
5.5 Some of the models of hiders in the final game: the Chest and the Trash. 57
5.6 Some of the models of destroyers in the final game: the Acid and the Fireplace. . 58

5.7 Model of the only Tamperer in the final game: the Ink. 58
5.8 The House we built for our characters to interact in (seen fom a Godmode Camera). 59

6.1 Statistics for the population that tested the First Person Camera 71

List of Tables

2.1 Some of our analysed storytelling systems and information about them in terms of areas relevant to our study. 27

4.1 FAtiMA's goals from a character role's perspective. 44
4.2 A FAtiMA goal's properties. 45
4.3 A FAtiMAs action's properties. 46

6.1 Descriptives for the first and second parts of the questionnaire. 64
6.2 Homogeneity of Variances table for the first and second parts of the questionnaire. 64
6.3 ANOVA table for the first and second parts of the questionnaire. 64
6.4 Games-Howell post hoc test table for the first and second parts of the questionnaire. 65
6.5 Descriptives for the third part of the questionnaire. 66
6.6 Homogeneity of Variances table for the third part of the questionnaire. 66
6.7 ANOVA table for the third part of the questionnaire. 67
6.8 Tukey post hoc test table for the relevant questions of the third part of the questionnaire. 68
6.9 Tukey post hoc test analysis for the third part of the questionnaire. 68
6.10 Games-Howell post hoc test table for the third part of the questionnaire. 69
6.11 Games-Howell post hoc test analysis for the third part of the questionnaire. . . . 70
6.12 Most common camera types based on a game's genre. 70

Chapter 1

Introduction

Many storytelling systems and games have graced both the academic and the mainstream worlds at large. These systems allow for players to express themselves while experiencing a story that weighs in their decisions to determine its outcomes through their actions, some more than others. Many times these stories involve other characters that either have Artificial Intelligence, called Agents, that are controlled solely by computer algorithms, or are controlled by other human players. Normally both the players and agents must interact with each other other and/or with items in various ways in order to move the story along.

The controlling of the main character by the player sets the mood for the whole game. A good control scheme will enforce the player's agency, while a weak one might put him off the game altogether. To further enhance this idea of agency, it is very important to keep the user visually focused on what is going on in the game and the events happening within its world. Camera work in games is a focal component to set properly in these cases, because it dictates how the game's story will be experienced.

Given this importance, we aim to study how camera placement can influence how a player perceives and understands a story from within a storytelling system or game while they play it.

1.1 Motivation

Storytelling is at the core of human experience. It has been a defining attribute of many cultures, and used in various ways to portray a wide array of messages. Fairytales, Novels, Comic Books, Movies, Videogames, all of them share the need and will to guide the user into a different world, narrating and putting us in the shoes of the lives and trials of many and diverse characters, in several and sometimes unpredictable situations.

Stories have, until a few years ago, been told from one perspective, with one reality in mind. But that paradigm is changing. People want more interactive and adapted experiences. They want to experience the story by *their* own terms, meet characters who will react to what *they* do. They want to explore, to feel, to *live* and breathe these worlds.

And so emerged the concept of *Interactive Storytelling* (IS), where a story involves its users, centering them at the heart of story progression. Users may influence the world where other characters live, and even live among them as a character themselves.

Most recent storytelling research focuses on the more literal meaning of storytelling. There

are still many open questions as to how to create a story by chaining different events, creating believable characters that react those story's events or even how to make the processes of creation of these stories and characters available to authors at large by the way of designing tools to help them achieve those goals.

Some of these storytelling systems have little when it comes to a graphical interface. Most of them are trying to prove the correctness of their new or improved approach. However, many recent storytelling systems feature a graphical interface and let the user experience the story at their own pace.

Camera placement in these systems has been a focus of many a research, especially when inter-character interactions are concerned. When the players engage in conversations, for example, it is a good idea to follow the already established rules of cinematography in order to introduce the speakers and keep the flow of the conversation as smooth as possible.

Sometimes the complexity of a game will require several camera modes to work interchangeably, depending on the type of action or interaction that needs to be performed. A camera that works well for a conversation is probably not as engaging when trying to move around in the world. The camera must then become a dynamic entity, changing its parameters and working differently depending on the state of the story, interactions being performed or even at the user's request.

1.2 Problem

But even this dynamism needs to account for the player's ability to understand what is going on around her.

One issue rarely discussed when a game is being presented is why a certain camera placement was chosen over all the many other possibilities. What could be the difference (or differences) that make developers choose one version over another?

As we stated previously, camera placement can set the whole mood for the story about to be told. A more personal story might want to make use of the first person perspective so the player can see through the eyes of the character she's embodying (or even becoming the character herself). A story where several characters are important might want to adopt a looser camera so the player can follow several of them at a time and see what they are doing.

But this approach to choosing a camera mode for a game or storytelling system poses a question:

> *Given a camera mode, will the player be able to perceive the events of the story? Will she understand the overall arc of the story? The relationship between the characters? Can she understand their motivations and goals?*

This question is actually two-fold. A player needs to see the events to understand the story, but one does implicate the other directly: a player might see the events, but ultimately fail to understand what they mean in the overall story. Thus, this question needs to be preceded by a much more broad and important one:

> *Can the camera affect the way a player experiences a story?*

Our research aims to answer this question. For that to happen, we will evaluate how the camera plays a role in the existing storytelling systems and games, and especially the design choices that the camera influenced. These design choices can be varied, from including/excluding characters to changing the topology of the game world to better suit the visualization technique chosen.

We will then develop several camera models and incorporate them in a storytelling environment populated with characters. Both the main character, embodied by the player, and all other characters can interact with each other and many objects spread out over the world. These interactions make up the story as they occur (meaning there is not a definitive time line or specific way they should happen). By combining these factors we will try to validate the hypothesis that:

> *In a storytelling environment, having a camera too close or too far away to the main character is detrimental to understanding what is happening around her. By having the camera focus on the main character but also let the player see her immediate surroundings, the player will be able to place herself and connect better to what her character is experiencing.*

1.3 Document Outline

In order to put our work into perspective according to other academic works, we provide a section were we study and analyse **Related Work** in chapter 2. In it, we provide much of the groundwork and definition that we will use throughout the document, as well as point out many works of research that benefit ours. These works were categorized in terms of **Virtual Cameras in 3D environments**, **Storytelling Systems** and **Games**, **Virtual Cameras and Character Control**, and the foundations of **Interactive Storytelling** itself.

Using this information, we present the **Model** for our system, where we explain how all our previous research comes together in order to build it. We break this model in its most important components: The **World**, its **Story** and **Characters** and, of course, the **Camera**. We then materialize this model by **Implementing** it using several tools and libraries. We explain how these sub systems work between each other in order to make our designed model a reality. Finally, our System and its **Interactive Story** is introduced, including the **Characters**, their roles in the story and their **Goals**.

We then **Evaluate** this created system in terms of our research questions posed on the previous section (1.2). This entails many phases, where we **Create a Testing Environment**, **Gather Results**, and then **Study and Compare** them using statistical data.

Finally, we come to our own **Conclusions** and present points that cold be explored to develop **Future Work**.

Chapter 2

Related Work

Studying camera in virtual systems is not a new field. It has been applied to many types of systems with different goals in mind. In this chapter, we will study these systems and leverage their importance for the remainder of our work. These systems have become increasingly specialized over the years, and require cameras that play well with their strengths. As such, we will present several camera work done over the years. Starting with **3D Virtual Environments**, we explore the core principles and concepts of cameras in general. We then introduce **Interactive Storytelling** and the ways one can develop one such system. This will be important in order both to understand where our work stands within the research community, but also to rationalize some of the **Camera** decisions made in those systems. We then talk about cameras in more specialized virtual storytelling systems: **Games**. After analysing these systems, we will round up some of the more used camera patterns over all these systems and explain how they fit with **User Control**. Finally, we will perform a **Critical Evaluation** over the work we evaluated and see where it can benefit ours.

2.1 Camera in 3D Virtual Environments

Many camera studies have been done in a more broad approach, not committing to any type of system. These studies allow us to view the other, more specific ones in a simpler light, since they are founded in the very principles these broad systems introduce.

As early as computer systems with graphic capabilities were created, the inherent question of how to best view the visual information these systems showed us followed. Although much of the initial research was initially turned towards how to make the representation more accurate, more and more researchers have started to realize that the way we see the information is just as important. If you don't see the information, or misinterpret it, its purpose is clearly downplayed in the eyes of the user.

But this is not the only problem. These systems, as much as they need to show the information, need to also let the user explore the information at their own pace. This means that a new variable emerges that will complicate these camera studies even further: **user control**.

Phillips et al. argued that providing good control of the viewpoint is essential to give users a sense of three dimensionality of the objects they are manipulating. Users need to see what they are doing, they need to feel the relationship between the objects and the input devices, in

order to manipulate them better[31].

Initial research on the subject of object placement in 3D worlds identified 6 degrees of freedom: three degrees for its position on the axes (commonly referred to the tuple (x, y, z)) and three for its rotation along those same axes (pitch, yaw, roll). As Gleicher and Witkin stated, directly controlling all these degrees for an object with 2D mapped input devices (such as the keyboard or mouse) is extremely difficult to the point that it becomes confusing and frustrating[17]. Brooks and Mackinlay et al. also came upon this problem in their research but noted that, in specific cases and approaches, some of these values might be defaulted and thus removed from the control of the user[5][22]. Drucker and Zeltzer went further in his assessment by stating that in order to achieve an intelligent camera control scheme, and maintain task focus, direct controls should be abstracted into primitives that allowed higher level control[16].

In order to simplify the creation of these primitives, metaphors were introduced to let the users map real world concepts to camera control. Ware and Osborne introduced three such metaphors and mapped them to real world objects for an easier understanding of their concept, as well as providing a tangible interface[50]:

- **Eyeball in hand**: In this metaphor, the user moves a camera in her hand that serves as her viewpoint in the 3D world. By moving and tilting this physical camera, so does the virtual camera move and tilt accordingly.
- **Scene in hand**: In this metaphor, the viewpoint control is indirect. The user moves and rotates a bat that represents the virtual scene, while the camera remains stationary.
- **Flying Vehicle Control**: In this metaphor, the user operates a bat that, when moved, will propel the camera in that direction by means of a velocity change. Also, by rotating the bat, the camera will rotate as well.

However, the results they came to after evaluation corroborates what we have stated earlier: these metaphors do not work everywhere and for every case, and must instead be used on a case by case basis. Even though each metaphor was easy to learn by itself, they reached the conclusion that each one was suited for specific cases of interaction:

- **Eyeball in hand** was received with mixed results. Because it was tied directly to the physical realm, it forced users to move around and sometimes go into weird positions in order to get the visualization they wished. Also, it was very poor at capturing small details due to hand shaking when in these awkward positions and confused users when they wanted to see the back of the scene, disorienting them. However, users rated this as the easiest of the metaphors to learn.
- **Scene in hand** was rated the most useful when inspecting an object in detail in the centre of the camera. But problems begin to occur when the centre is focused too far, making the rotations too abrupt.
- **Flying Vehicle Control** was rated the most useful when trying to navigate a vast 3D scene. However, there needs to be a special consideration as to how speedy the control is, or users end up getting disoriented. Also, when close to the ground, this metaphor gave

the impression of walking instead of flying, which made it easier for the users to grasp this concept. It is also the metaphor that allows the biggest freedom of movement.

User disorientation is actually one of the top priority issues when dealing with camera control[22]. If the image jumps or is discontinuous, it might break the viewers **willing suspension of disbelief** (a viewers ability to perceive something as realistic and physically present, even though it is not). This is especially important because people need time to expose themselves to these virtual worlds in order to build a world model that allows them to navigate them better. If the image is discontinuous, it will undoubtedly damage the world model people create. Brooks observed that users tend to create these models in a top-down approach: they "start out doing map navigation and then progress to scene navigation" [5].

More recent research tried to abstract camera placement even further by automating it using computer algorithms. One of the first ways researchers tried to implement such methods was by using what cinematographers call idioms. These idioms are used to establish the several viewpoints in a scene and how to transition between them to present events better to the audience. In 3D virtual worlds, He et al. postulated that establishing similar idioms could be beneficial to establish comprehension and help navigate those worlds, locating characters with which to communicate with and also setting up the characters so they can all see each other when interacting between themselves[18].

The problem with idioms is that there is no direct translation between the language used in cinema and the one used by computer algorithms. By letting users explore at their own will, cameras in 3D virtual worlds will be invariably harder to place properly. This is because of the random nature of these systems: characters won't always be in the same place or doing the same actions all the time, and the system needs to take this into account in real time so as not to break the users' immersion by simply stopping the system to make these calculations, which might be jarring. The introduction of this new constraint complicates matters even further. Bares and Lester identified this very problem when trying to make a dynamic camera to incorporate in a 3D animated explanation system for educational purposes[4].

2.2 Interactive Storytelling

With the development of recent technology, Interactive Storytelling (IS) systems have grown steadily, and attracted the attention of an even more mainstream audience. Some people see Interactive Storytelling as a divergent path from the book, offering a distinct non-linear experience in contrast to a book's straight-forward approach. But this development in technology is always branching, always finding new ways to do the same things.

As there are numerous ways to tell the same story, so there are ways that an Interactive Storytelling system can be implemented to tell a story. Before focusing on the camera work they apply, it is important to give a small introduction to the differences between these various systems in order to justify their camera related decisions. Mainly, Interactive Storytelling systems differentiate themselves in two distinct factors: **Core of Progression** and **Planning**.

In the next section, we will provide a more camera-centric research approach that fits closer to our research goals and how the third variable, **User Control**, influences the camera.

2.2.1 Core of Progression

Stories vary wildly in terms of their focus. Sometimes the author will want to highlight the trials of a certain character (along with his feeling towards his predicament, etc.), while on the other hand some stories are better told when framed around other events that precede and follow it (even if the characters within it are as important as the story itself), in very specific time frames. These types of storytelling focus respectively on characters and story.

In Interactive Storytelling, it is no different: different stories must be told by different methods, and with different focus as well. But this fact is more clear cut when developing a system for Interactive Storytelling. Researchers have studied both these two types of focus in story telling, and analysing how both characters and story help influence the overall narrative. Such systems are therefore commonly named **Character-Based** and **Story-Based** Interactive Storytelling systems.

Story-Based Interactive Storytelling

Story-Based systems tend to favour an overall story structure. They clearly define the events that occur within a story, the connection between them and the events that both lead to and follow a specific plot point. This gives a very tight control over story development, providing a better cohesion between all events within the story, thus providing better continuity.

On the downside, the story becomes rigid, and user decisions alter the plot only at a very high level, since the story already has a clear chain of tight-knit plot-points. This means that characters (including the user) that take part in these events have clear-cut goals, and must achieve them through very specific means.

This is also a common problem present in Propp's narrative functions[40]. Since the *Dramatis Personae* (or character roles) are interchangeable (meaning a character of a type of *Dramatis Personnae* can be switched by another character as long as it has the same type), the story must be written taking this aspect into account. Therefore, characters with distinguishing feats and personalities would be undesirable, since this could lead characters to perform actions that did not conform to its personality for the sake of story progression. This in turn would make them unbelievable in the way that they would contradict their personalities with the actions they would perform. Paiva et al. identified this problem, when, for example, a child playing as a villain did not want to cause harm to other characters[29]. Still, the system allowed such discrepancies, but this led to some disappointment in the users, because character development suffered as a result. Not having any personality, however, leads to the problem that the audience cannot bond with the character, and thus not establish any emotional connection with her.

Other researchers have tried to minimize the down-sides of story-based systems, by incorporating constraints that help build the narrative as it develops, instead of setting it in stone from the very beginning[34][35]. These constraints also mean that a story might progress through different chains if there are many actions available to set the facts that will influence the predicates of other following constraints. Still, in order for characters to have any characterization, it must be provided by events in the story itself, such as the example given in a Spy story, where famous agent James Bond seduces another character only because there is a constraint that defines this action[34].

Character-Based Interactive Storytelling

As the name implies, Character-Based Interactive Storytelling systems favour characters and their diversity over overall story structure. It also tends to give greater emphasis on inter-character relationships, and how they affect characters' decisions and thoughts. This allows for greater variety in the cast, and makes characters that are much more identifiable and, as an extent, likeable and believable.

These characters come across as more intelligent, because their decisions are made without a rigid story structure in mind. As such, they have more freedom over what they can accomplish and how. They have a goal and they will find the best way to achieve it by their own terms. But this leads to some problems as well.

When the characters are given "free reign" over their actions (free reign being quoted because there is still, of course, a limit imposed by the system itself), many events might occur, and while some of them might be good (in terms of pacing, tension, etc.) for the story, others might be completely irrelevant or even undesired. As a result, story structure suffers in the process.

Several methods have been implemented in order for this problem to be alleviated, such as heuristic values for story states (preferring certain actions in detriment of others).

"Emergence" is a high selling point, as well as a key feature in these systems. Emergence deals with how the story and its events might branch into different paths even though they were not specifically introduced, but rather occur as a result of the characters' though patterns and actions. Cavazza et al. identified several key aspects for a system to be highly emergent in two of his works: a *Friends* TV Show re-enactment[8] and a Pink Panther cartoon generator[9]. The aspects that he referenced were:

- **Initial Character Position, Action Duration and Speed**: If characters are in different positions each time a simulation starts, they have a different probability of running into each-other during the course of the simulation. This can be improved further by assigning tasks a certain amount of time to perform or even have characters move at different speeds. This gives more or less potential for character-to-character interaction (since they may or may not run into each-other), and may serve as a good starting point for variance in the story.

- **Random Outcomes for Actions**: When an action does not have a 100% probability of success and fails, the character is forced to look for other alternatives to fulfil its plan. This makes the characters plans much more diverse. Of course, this randomness must be backed up with additional ways to solve a problem or task.

- **User intervention**: "User intervention is by Nature non-deterministic"[8], making it a potential factor for story derivation. User intervention might change the world the characters are in, often resulting in the failure of some part of their plans. Like random outcomes, Characters need some way to react to this interference, often through re-planning their strategies.

- **Character-to-Character Interaction**: Characters' interactions with each-other are basically a competition for action resources. These might be objects in the world or even

other characters in the set. These competitions can cause "causal chain reactions"[8], where a character is constantly trying to focus his attention on different resources but is impeded by other characters that are using them.

- **Action Failure**: This is different from failure induced by resource competition, in the way that a character will be able to perform a given task, but will not be able to complete it because some factors outside his knowledge were not in agreement with his desire (a character may want to buy sweets and have the money and a shop that sells them nearby, but he might fail in his task if the shop does not have them in stock). This gives the characters a sort of rationality, separating the conditions that must be met for a task to be considered (or pre-conditions, in the example having money and access to a shop) from the conditions that must be met for the task to succeed (or executability conditions, in the example the store having the sweets for sale)[9].

Other works that include Character-based approaches are a sitcom based around the organization of a party [12], and a re-enactment of the XIX Century novel *Madamme Bovary*[32].

Mix of Story-Based and Character-Based Interactive Storytelling

Some systems tried to expand on this clear distinction between both previous types by merging them in a way that would be beneficial, combining their strengths.

Often, these types of systems will be story-based, meaning their story will be planned in order to have stronger coherence, but these plans will be made with the characters' abilities and personality in mind (thus incorporating a character-based type of planning), reducing the sometimes thoughtless and schizophrenic behaviour that characters tend to have in story-based systems. This also limits the unneeded scenes or redundant interactions recurrent in character-based systems, giving the characters more focus.

Façade *Façade* employs a Drama Manager that is charged with sequencing what Matheas and Stern call beats[25]. These establish a high-level direction for the narrative's goals (and thus structure the narrative), where the characters (a married couple) fix drinks for the user, discuss their state as a couple and many other themes about their life and their marriage. These can be interrupted by the user in what is called a Mix-In: the bringing up of a satellite topic (either by talking directly about it or by touching a relevant object). The story beats are sequenced by choosing an unused beat using several factors like pre-conditions, weights, priorities from the beat itself[25] and also overall story tension[24], and then apply it to the sequence of beats that are performed. These beats are authored with several variables in mind, having several lines of dialogue that pertain to factors such as coming back from a mix-in (re-establishing context), different tension levels of the story, characters' level of affinity with the user, etc. These *nuances* allow for context aware interactions, increasing believability and richness in the interaction, giving the character's actions and motivations more depth.

Actor Conference Riedl and Young presented another concept, that of the Actor Conference or **Blackboard**[41]. In this type of distributed story planner, an overarching goal is introduced

and then decomposed into the main story events that need to take place. These decompositions are then distributed by the "Blackboard" among the Actors. The set of actors chosen for each sub-goal are the ones the system believes can achieve or contribute to the solution of that particular problem. These actors take turns and refine the hypothesis, or unfinished solutions, of the problems present in the "Blackboard" until a solution is found. For example, if a character needs to know about a particular topic to progress the story, she will post in the blackboard a solution that has another character knowledgeable about the subject talk to her. Then, this character slot can be filled with any character that has the necessary knowledge to impart.

This means that a problem should have all the necessary information needed for an actor to take part in it (so that all actors have the same knowledge about it in order to solve it). It is worthy of note that these actors represent characters in the story, with different skill-sets and characteristics (like the aforementioned knowledge about certain topics), and thus the hypothesis and solutions that the Actors propose are always in accordance with what the related character can do. In this way, we combine a story-based approach by causally linking the steps in the story plan, but also a character-based one, in which the characters that will contribute on a specific step will do so because they have what it takes to do it.

Story Director Riedl and Stern introduced the concept of different behaviours that can change at real-time during the simulation to address specific needs of the story[43]. At the top-level is a narrative manager (called the Automated Story Director), that has the responsibilities of generating and managing a representation of the events of a scenario, monitoring the simulation and looking out for inconsistencies. It is from this plan that the characters of the set are given their goals, based on the conditions needed to be met for the story to progress. If a story requires a bomb explosion in a certain location (most likely to heighten tension in the narrative), it will give a character in the set the goal to plant the required bomb. When characters receive these goals to achieve, they trigger a scenario-specific behaviour, the Narrative Directive Behaviour, that makes sure that the character tries to accomplish the goals it has been given (in this case, the character must acquire the bomb from somewhere or fabricate it by her own means). When a character has no such goals, a Local Autonomous Behaviour is established, in which the characters will go out on their daily lives and perform generic activities (such as cleaning her house).

2.2.2 Planning

Planning, much like the core of progression of a story, can be applied to both Story Structure (defining plot points and their order) as well as Characters (the plans to achieve their goals). Indeed, planning – be it story or character related – always needs to be present no matter the type, and it will be the characters that will make sure it is followed.

Planning in Interactive Storytelling systems has been a key focus for its developers. It basically dictates how characters react to the situations they are in and, as such, is a very important factor on the story progression.

The first part of the plan is, of course, its goal, or what we want to achieve. Characters will want to act towards a goal, while antagonists will want to oppose it (in what is called a

"counter-plan"[13]). As discussed above, plan failure is a natural part of storytelling: it adds to the drama and believability of the story. As an extent, some re-planning will be necessary in order for the story to continue afterwards, setting a new goal and thus keeping the characters focused on an objective.

The Hierarquical Task Network (HTN)

The Hierarchical Task Network was adapted as a formalism for Interactive Storytelling in many systems due to its appropriateness for use in domains that are very demanding knowledge-wise, a characteristic that Interactive Storytelling clearly has. The HTN is also a formalism with good representational abilities, as it supports interleaving planning and execution (a characteristic highly desired due to the need to re-plan strategies mid-simulations).

A HTN is composed of, as the name would imply, tasks, which can be of various types: primitive (simple tasks that can be translated into actions), compound (sets of primitive tasks), and goal tasks (defines certain conditions that must be made true once the task is over).

Tasks are thus composed of other, simpler tasks, meaning that problems will be solved on a top-to-bottom fashion, decomposing bigger problems into smaller ones until a problem is simple enough to be solved on its own (through an action). It is from this very decomposition that the solution plan is then extracted and executed.

In this formalism, the ultimate goal of the story will be the root of an (usually) AND/OR graph. Its children nodes are then the scenes of the story, which have a temporal arrangement. These are then split into various nodes that will represent the primitive actions needed to accomplish the goals of a certain scene. AND nodes will represent a group of tasks that must be completed in the same order as they appear for the parent task to succeed, whilst a OR node will represent several tasks that are alternative ways to solve the parent task, and from which only one can be chosen. Each node under an OR node specifies different pre-conditions so that different tasks will be chosen according to the specific execution of a simulation so far.

In order for the HTN to represent a story, it relies on two key requirements. The first, and most obvious, is that any action can be partitioned into sub-tasks, and that these bear no dependency between each-other. This excludes story factors like negative or positive outcomes in actions, which harden or facilitate, respectively, the following tasks. Also, these tasks must be totally ordered, meaning that every task (that is not the first or last task) must have a clear previous task and a following task.

Several Projects have used this kind of planning formalism, such as an enactment of the popular "Friends" TV show[11], a generic sitcom situation about organizing a party[12] and even a military training scenario application (IN-TALE)[44].

The Heuristic Search Planner (HSP)

The Heuristic Search Planner was initially adopted in order to overcome the main problem of the Hierarchical Task Network of not having a way of incorporating long-distance interactions between actions in a plan. These interactions state that actions in the narrative can influence (positively or negatively) other actions that occur after it (making them easier or harder, respectively).

An HSP usually uses a STRIPS-like description of the problem from which an heuristic can be directly extracted. This heuristic is then used to search over the space of states of the world, determining thus the likelihood of reaching a goal.

The operators (or actions) used in this kind of planning have, beyond the common pre-conditions and executability conditions described above, two types of lists: the add-list, one or more new facts and goals given by finishing the operator, and the delete-list, one or more facts and goals that are irrelevant or completed after the operator ends. These are then used to lock/unlock new operators, by providing/removing their pre and executability conditions. The add and delete lists are the main drive for long term dependencies between operators, updating information that may not necessarily be used immediately.

Of course, since there is no direct connection between any set of operators, narrative drive will be maintained through the narrative's (or the character's) ultimate goal, which the character(s) will try to pursue. This has two side effects. On one hand, this allows for greater story versatility, since the order of actions can vary depending on the current heuristic evaluation. On the other hand, this planning method has greater potential for dramatic or comical situations, since it does not try to attain a correct plan from the start like HTN (where the state of the world is known at all times), but rather builds from the knowledge that it acquires over time. This plan might sometimes be flawed, as seen in [9], where the Pink Panther needs a bath after going to the cellar to cut the water, but cannot complete this task because it cut the house's water moments before.

This is further elaborated on by the pre and executability conditions. As explained above, separating these two concepts allows for operators to have more varied outcomes and give planning a more organic approach in the way that an action might seem to be doable from a theoretical standpoint (the pre-conditions holding true), but be ultimately impossible due to some unforeseen event (the executability conditions failing).

HSP has been used in several projects, mostly when a character-centric approach to Interactive Storytelling was taken. Examples of this are the recreation of Pink Panther cartoons[9] and a new take at the XIX century's "Madame Bovary" Novel, by Flaubert[32].

Beats

Façade introduced in its work the concept of beats and how they could be used for planning[25]. The simulation features a Drama Manager that contains beats, which are composed of procedural content, also called Joint Dialogue Behaviours or JDBs. These are dramatic exchanges between the characters, focused around a narrative goal, and can change based on the values of the story's state such as story tension, character affinity with the user, etc. This procedural content can be one of three types: Beat Goals, Beat Mix-ins and Global Mix-ins.

Beat Goals represent the beat's canonical sequence. Also, to incorporate reactivity to the user's interruptions, it features information such as resuming points, in case the beat is interrupted, and several re-establishing dialogue sequences.

The **Canonical Sequence** is a guideline around how a beat should perform in case of no response from the player. They are composed of several goals, including the goals that establish the tone of the beat and the conflicts within it. The Canonical Sequence is not static though,

since the user's actions need to shape up the experience. In order for them to be acknowledged, there's the **Handlers**, whose job is to listen to user interaction and change the Canonical Sequence accordingly, by adding, removing or reordering Beat Goals and also Interjecting Mix-ins.

Mix-ins, either beat or global, are interruptions to a beat caused by the user. The difference lies, as the names imply, in their scope. Beat Mix-ins pertain to interruptions that are in context with the beat's current theme, while Global Mix-ins focus on satellite topics, objects in the scene, generic reactions to praise or rebuttals to unrecognised or confusing input.

These JDBs are then sequenced within the beat according to its specific logic, using pre-conditions, weights, priorities and other factors, such as story tension, etc. A beat is chosen to continue the story, based on these values, from an unused set of beats, which are sequenced as the story progresses.

Beats were also adapted from *Façade* and used in a military training scenario application[43] when using Narrative Directive Behaviours.

Constraints

Porteous and Cavazza introduced their work with planning of a story using constraints when trying to recreate a James Bond Novel, namely *Goldfinger*[34]. These constraints are used to order events of a narrative in time based on other events, giving them a partial temporal order (since events do not depend on every other but a small subset, or an empty one). The constraints are defined through the operators:

- **(sometime-before a b)**: b must be true for a to occur.
- **(sometime a)**: a must be true at some part of the narrative.
- **(at-end a)**: a must be true by the end of the narrative.

As the name suggests, *at-end* constraints are the story's goals, that is, what we want the narrative to end in. As such, they are the planning goals as well, and must be featured at the end of the temporal order.

The planning method for this kind of structure uses a modified version of a planning method that uses landmarks[34]. The difference lies in the fact that, unlike landmarks, not all events need to occur in order to achieve an end goal. However, just like landmarks, events must be achieved to satisfy the constraints of other events.

The planning phase needs a tuple of information so it can succeed: **Facts** about the world, **Initial State**, **Goal Condition**, **Constraints** and **Operators** (each having an add, delete and pre-condition list). A tree of constraints is produced and, for each constraint present, a plan is formed (if possible) and attached to a master-plan. This plan takes into account the facts of the world and pre-conditions of the operators that it will perform. Any constraints that the new plan makes true (through the operators it applies) are removed from the tree of constraints.

For example, a constraint that runs at the beginning of the story might have the character go to a certain location. After this happens, a new fact about the character's location now

exists, replacing the old one. After this is done, a constraint that needs the character to be at this location can then be chosen to be performed.

This process repeats until there are no more constraints present or it becomes impossible to formulate a plan, in which case, a narrative has been assembled or not at all (respectively).

This plan is more focused on story, and not so much in characters. Still, there are ways to incorporate characterization in this type of planning, by providing specific events that correspond to characters' actions, such as seducing another character. Also, this type of planning, with the correct constraints, can lead to phases in the story where conflict and suspense can arise.

Another work that uses this kind of planning is the enactment of the Merchant of Venice, which uses Points of View to delve deeper into a character's motivations and actions. Here, Points of View are Constraints that depend on a characters mindset that is defined at the beginning of the story and never changes[35].

The Actors Conference or Blackboard

On this kind of planning proposed by Riedl and Young, the core of the plan is given by Author Goals[41]. Author Goals are states of the story ordered in time of appearance. These can be conflicting, since they are completed in order, thus creating narrative interest through "Reversal of Fortune", a term referring to the transition of a character's circumstance from a positive one to a negative one or vice-versa.

With the Author's Goals in place, it is up to the Actors to establish and fulfil the abstract plans that correspond to these goals. The board initially contains an incomplete plan and annotations. This incompleteness is due to the fact that the plan has flaws which are either actions with its pre-conditions unsatisfied or an abstract action that is not developed enough to be put in motion (not split into concrete actions). Actors are expert systems representing each character of the story. The term expert means that the actor is fully committed to this character, and knows exactly how this character should act in any situation. For that, it has the set of actions his character can perform in the story world, and also an heuristic function to customize plan searching based on the character's attributes. This enables the actor to serve as a mediator for the character within the narrative: he will only perform actions in agreement to his character's traits, in the form of preconditions and effects. Actions can be either abstract or concrete, and are used by the actor to further develop abstract actions that he finds in the blackboard. The key difference is that abstract actions can be further developed by him or other actors, and concrete actions are specific approaches by a character to solve a part of the plan.

The blackboard will summon each actor in turn so he can try to further develop an abstract action in the plan, based on the character's attributes. This development can be either a further refined abstract action or a concrete action. Abstract actions can take the form of interaction with other characters, and these are considered separately, because characters cannot decompose actions that do not belong to them. This forces abstract actions in the plan to refer a character role instead of a specific character (unless the character is forced to participate due to a story constraint or was the one to propose it), based upon the action's pre-conditions and constraints. These role abstractions are called Cast Calls, in the way that all the cast can decide to perform a certain action if they are able to.

2.3 Camera in Interactive Storytelling

Whenever Interactive Storytelling is concerned, camera work needs to account for a very important variable: **User Control**. As we have stated before, this is the key difference between Interactive Storytelling and other storytelling media.

User control and virtual cinematography stand at odds with each other: by having greater control, the cinematography will become its "servant", and the reverse is true as well.

This also applies to narrative: greater control means that a narrative will have to become less focused in order to account for every possible interaction. Researchers commonly refer to this problem as the dichotomy between story coherence and user control, or author control versus user control[33][46][52]. It is a very important balance to keep because the story should not be very derivative, leading to the user's boredom, or be very inflexible with its control over the story's progression, thus risking removing the player from the experience altogether by undermining her sense of presence and agency[42].

Also, giving more control to the user (on both narrative and cinematographic terms) might mean she will never see certain events of the story unfold. There's also the possibility of she seeing them but not completely understanding them, or even get distracted by other parts of the environment and miss events that would otherwise be visible[46].

Nack argues in the opposite direction: that the perception of information is important to establish how much knowledge a user has over the story being told. By not showing every event that happens within a story, an Interactive Storytelling system can effectively create a dichotomy of objectivity vs. subjectivity that might engage the user more and provide further variety in temporal, spatial and chronological placement of events[27]. There have been some works that try to adapt this concept to storytelling systems[51][2], but this is a relatively unexplored possibility as of yet. due to its complexity.

Camera placement in Interactive Storytelling systems needs to account for a lot of variables, and the characters are just a part of it. The environment that they live in can also provide information that might be relevant to the plot, such as lighting, objects and how they are arranged in the set. Tomlinson et al., when developing an Autonomous Cinematographer, identified several elements that make up a camera shot when dealing with a world filled with autonomous characters (a Character-Based Interactive Storytelling system)[47]:

- **Actor**: the most important part of the decision making when establishing a shot. When a user is controlling a certain character, she becomes the obvious choice for the camera to focus on. However, there might come a time when other characters are doing actions that are relevant to the story and should also be focused. This, however, should be used with caution, as to not break the user's experience.

- **Angle**: normally decided based on the actions being performed. In world establishing shots, for example, a bigger angle is a better choice, while close-ups are better to determine the character's expressions and reactions to world events.

- **Motion**: is determined by the overall emotion of the focused character. When a character is sad, for example, the camera might move slowly, whereas an angry camera movement might be fast and abrupt.

- **Transition**: defines how shots mesh with each other. A cut, being an abrupt change, might confuse users as we have discussed on 2.1, while a pan from the current position to the new location of interest gives users a better sense of motion and tracking of their location.

- **Occlusion Detection**: when walls or other obstacles are in the way, the camera should readjust its position and other parameters so as to remove the obstruction.

- **Lighting Elements**: light might be a good way to divert the user's attention towards or away from something. It can also convey a series of emotions. As such, lighting placement on the shot should be considered carefully.

These, combined with the type and extent of user control we want to give the target, will decide the type of camera the system will use. We discuss these types of cameras and their use in section 2.5

2.4 Camera in Games

Games share many similarities with Interactive Storytelling systems, especially the genres that prominently rely on story to engage their audience. One such genre is the Role Playing Game (RPG). However, unlike some Interactive Storytelling systems, games engage their audience with more than story, they also do it through *gameplay*. Gameplay describes the many types of different actions (or *mechanics*) players might do to move the story along (or moving from story point to story point): from combat, to exploration, even puzzle solving. Combined with mechanics that make up story progression, such as character conversations, most games nowadays feature more than one mechanic as they get progressively more complex. From our previous analysis, this means that the camera will need to change based on the current type of interaction taking place.

For example, in various games where players have to travel between areas that are very far apart, the game shifts from its normal perspective into one where the whole world can be seen so the player can select their destination more easily. Games normally call this the "world map". This happens due to the fact that it is often far too cumbersome to recreate the whole world (and its connections) with the same level of detail that main areas have. This dichotomy is what Švelch calls **Illusionism** vs. **Illustrationism**[49]:

- **Illusionism** is the technique applied by games where the world representation is meant to be final and representative of the world as it truly is.

- **Illustrationism** is the technique where game's world is simplified and the player is shown a mere representation or part of the world.

In this way, in our *world map* example, each main area is considered an "illusion", meaning, the game's world is meant to work this way for every part of the world. However, due to constraints, developers decided to make the connections between these "illusory" areas, our *world map*, an "illustration". But, in order for this change to work, a change of perspective is

also needed, so that players know how to interact with it accordingly, and distinguish the types of actions they can perform[49]. Many, if not most commercial games resort to this type of approach, from BioWare's *Dragon Age: Origins*, to Interplay's *Fallout*. Even games that feature a whole seamless world, such as Bethesda's more recent entries in the *The Elder Scrolls* series, often feature a world map to keep players localized and provide them with means to fast travel between locations of interest.

Much like Interactive Storytelling systems, games want the players to feel connected to their representing character. By putting them at the centre of events, and making them believe they are achieving their goals, they reinforce the player's feeling of agency, and thus motivate them to continue. This connection needs to be made at both the visual and control level: only by working together will the player feel immersed in the experience. As Mallon's test subjects said while playing the action/adventure game *Might and Magic IX*[23]:

- "Might and Magic is terrible: You are just clicking and you can't see what you are attacking."
- "I didn't like the way you couldn't see your sword. It was hard to know whether it was working or not."

Developing skills is a very important factor in gaming experience, and the responsiveness of the input helps in this development. As we can see, this response lies in the visualization of what the input did in the world. The camera is pivotal in this visualization and also sets the mood for the interactions the payer will perform. Games have, until now, presented us with several types of camera control, all of which tend to favour distinct control schemes (more on that in the next section):

- **First Person Camera**: This type of camera places itself on the "eyes" of the character that the player is controlling. It serves to sell the player on the fact that she *is* the character, that the actions the character is performing is indeed the player's own. It builds a relation of proximity and, in horror games, sets up the tension by putting the player closer to the dangers of the world[21].
- **Fixed Camera Angles**: In this type of camera, the user's visibility is at the whims of the camera, which is out of control of the user: it changes based on the place that the character is. This makes for a more cinematic experience, but at the cost of interactivity, proving to be inflexible to players[47].
- **Third Person Camera**: This camera follows the character closely, as if the player was behind him. This type of camera is normally used to establish close proximity (though not as much as the First Person camera[21]), but lets the player see the immediate surroundings of her character. In games where precise character placement is necessary, such as platform games, this camera type is used more frequently, since it lets players keep up with the character, but also position themselves better, which can be difficult with a First Person Camera.

- **Top-Down/Free Roam/Godlike Camera**: In this type of camera, the player views the world and its characters from a high ground. This grants the user a bigger chunk of the world to see at one time, but at the cost of the player's connection to the character he controls[47] and details in the character's looks and expressions[49] because of this very distance. Usually, the types of games that feature this camera mode rely on tactical positioning of the characters, such as strategy games, simulations or recreations of tabletop RPGs. They use this camera not only due to gameplay reasons, but also because it would be either very cumbersome or ineffective to organize and command groups of people from a personal perspective. Sometimes, the game might not even have a physical character to ground the player into the world at all.

These types of cameras are nowadays considered "standard" across all virtual systems, and are used in many types of systems. There is, however, research into more refined possibilities for systems with more specific needs[6], most of them dealing with automatized camera placement based on the type of information to be transmitted. Since we're aiming for a camera fit for general gameplay, we will not discuss this topic in depth.

As can be seen, using the correct camera type in a game is very important, as it allows the players to better see the consequences of their actions.

The game *Mirror's Edge*, for example, tried to make platforming a core aspect of the game and used a First Person Perspective, which led to mixed results and to its criticism on many game review sites over its "trial and error" gameplay[3][48][19]. This was mainly due to the fact that people have a hard time translating body movement from the real to the virtual world. By not being able to see of feel the character's feet at all times, they imagined the character was actually further back than she really was. When playing any game on any platform, the player can't be exactly in the place of the character unless she delves into virtual reality. The player will always sit distanced from her point of view (the screen), and thus think she is (or rather, the character is) further back than what she really is. This ended up breaking the connection between the player and the character, even though that's the core strength of the First Person Camera.

In another example how the camera can break the users' immersion is *God of War III*. In its final moments, the camera shifts from the usual Third Person Perspective to the First Person Perspective for the first and only time in the series, confusing players as to whether this shift was actually meaningful in any way[36].

Cutscenes are also a big part of user disconnect in games, mainly due to their misuse. Although cutscenes can be used effectively to set the tone of the game, most of them are used in between play sections, "breaking the flow of play" and "contradicting the rules of gameplay"[37]. Most of the time, they also create a gap in the user's perception of the game, by employing different camera modes and focus than what they are accustomed to seeing throughout the gameplay segments.

As we have also discussed in the previous section, object positioning is very hard when considering a single viewpoint. This is very true when we're talking about a character that wants to see what is happening around her. The information should be present at a glance, but this is not always true or possible. This creates a disconnection where events are happening and

the player can't help feeling lost or confused, especially if too many things are happening at once or too quickly. This obviously begs the player to question: "Why can't I see it if it obviously is happening "in the reality of the game world"?"[49]

The Elder Scrolls V: Skyrim was a recent game that suffered from this problem. In the introductory sequence, the player is bound and cannot move. She's surrounded by several characters that try to engage in conversation with her, but they can't all be visible in the screen at the same time. This forces the player to change her perspective a lot and very quickly if she wants to keep track of the conversation[38].

This is the kind of problems camera placement needs to consider when dealing with a world that features a lot of interaction. In order to keep players from not knowing what to do next or where to look at, level designers employ a multitude of ways to maintain their focus. Milam and El Nasr studied and categorized these patterns with the help of several level designers from several mainstream games[26]:

- **Collection**: gathering items throughout the level.
- **Path Target**: diverts the attention of the player to a target in the level.
- **Pursue AI**: move towards friendly or hostile character to interact with her.
- **Path Movement and Resistance**: the general narrative goal that the player pursues, that can include resistance and/or environmental barriers. This pattern usually layers itself with other patterns.
- **Player is Vulnerable**: threats to the player character's safety.

All these patterns have two associated goals: **Design Goal**, where the game actively makes this goal mandatory, and **Non-Goal**, where the goal can be pursued by the player but is not mandatory. Also, these patterns can be used in combination with each other. The first three are arguably very influenced by the player's ability to see the world, so the camera should be taken into consideration when dealing with these patterns.

Nnadi et al. also studied the effects of camera positioning on players' ability to read a game's virtual space. By employing different camera parameters and making them consistent with each type of scene present in the game (in this case each scene was directly linked to a Chinese element), the players were able to understand the transitions between these scenes faster when compared to a traditional regular third person camera[28].

2.5 Camera and Types of User Control

There are many different ways for a user to interact with a computer system and the one used very much depends on the type of system that the control is destined for. After our introduction to camera work over various types of systems, it is no surprise that camera placement will directly influence the type of control that is expected from the system. The reverse is also true.

In this section, we will introduce some of the more common types of control schemes used with the camera types we introduced earlier, and introduce games that feature these types of controls.

The **First Person Camera** and **Third Person Camera** tend to be very similar in terms of how they control. This is due to the proximity the player has to the character she's controlling, and focus on the character's actions. The most used scheme for this type of visualization is often Mouse + Keyboard or Joystick approach, using keys or an analog stick to direct the character's movement and use keys or other buttons to have the character interact with the world. Recent games have started experimenting with motion controls for these games, but the connection between the user' actions and the games' interpretation is still at a point where it's not completely immersive. This is because these systems cannot account for the wide range of body motions a human can do, creating a jarring sensation in the user[39].

Many modern games and Interactive Storytelling systems sport this style of interaction, such as Bioware's RPG series *Mass Effect*, Arkane Studios' *Dark Messiah of Might and Magic* action game and the Interactive Storytelling system *Façade*[25], in which the player becomes a character controlled through mouse and keyboard (figure 2.1). *Mass Effect* and *Dark Messiah of Might and Magic* also provide joystick control in their console versions. The only real big difference between them is the type of camera used: **First Person Camera** for *Dark Messiah of Might and Magic* and *Façade* and **Third Person Camera** for *Mass Effect*.

Figure 2.1: Storytelling Systems and Games that Feature the First or Third Person Camera. From left to right we have *Mass Effect 3* (Third Person), *Dark Messiah of Might and Magic* (First Person) and *Façade* (First Person).

The **Fixed Camera** is commonly used in more cinematic experiences, such as Adventure Games. Since the camera does not have a discernible reference point at all times, incorporating a type of character control similar to a First Person or Third Person game tends to confuse the user. Most of these types of games tend to implement a "click to move" (or "point and click") approach, where a cursor, normally controlled with the mouse, directs the controlled character to where the player wants the character to go or what she wants her to interact with.

This type of camera has fallen in disuse over the years, since graphical processors can now render full three dimensional environments in real time, but it was common practise in earlier games, such as LucasArts' adventure games *The Secret of Monkey Island* and *Grim Fandango*, and horror games *Alone in the Dark Series* by Infogrames and *Resident Evil 3: Nemesis* by Capcom (figure 2.2). The first two support a point and click style of interaction, while the last two use a direct character control scheme, much like the ones given in the First and Third Person Camera examples. A more recent example in the Interactive Storytelling field is the Teatrix project[29].

Figure 2.2: Storytelling Systems and Games that Feature the Fixed Camera. From left to right, top to bottom we have *The Secret of Monkey Island*, *Grim Fandango*, *Alone in the Dark* and *Resident Evil 3: Nemesis*.

The **Top-Down or Godlike Camera** normally features control schemes more focused on gestures and sweeps. The world is being seen from a higher vantage point and, as such, precise control becomes harder to do for a specific character, depending on how far the camera is. These controls become more generalising and broad than character focused systems, most of the time because there is no character to control directly. These types of systems normally fare better at controlling groups of people and positioning them tactically.

Plenty of games and Interactive Storytelling systems feature this type of camera, such as Maxis' simulation game *The Sims*, Black Isle Studios' *Icewind Dale* RPG series, Lionhead's simulation/strategy game *Black & White* (which embodies the name of the camera completely by letting the player become a god), The Creative Assembly's strategy game *Medieval: Total War* and an Interactive Storytelling system developed by Charles et al., where the player observes and can interfere with the characters' plans by manipulating the environment[11] (figure 2.3). All these games use the mouse to manipulate the world and its objects and characters. It's worthy to note that motion based controls fare better in this kind of interaction due to its sweeping properties, since it is not as physical or as precise as other types of controls.

Figure 2.3: Storytelling Systems and Games that Feature the Godlike Camera. From left to right, top to bottom we have *The Sims*, *Icewind Dale 2*, *Black & White* and *Medieval: Total War*.

Also, as we have stated, many games use combinations of these types of cameras. Most of them do it so that the users can use the one they are most conformable with. Examples of this type of game are Phenomic's Strategy/RPG hybrid *SpellForce: The Order of Dawn* and BioWare's RPG *Dragon Age: Origins*, which provide a way to change the camera from Third Person to Top Down camera during gameplay, and Bethesda's *The Elder Scrolls V: Skyrim* RPG, which lets the player choose between controlling the character with a First Person or Third Person camera (figure 2.4).

2.6 Critical Analysis

We will now present a critical analysis of the work reviewed so far.

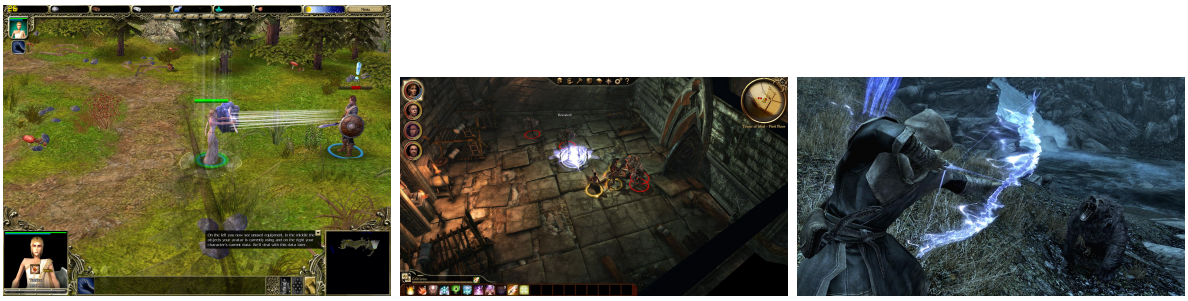


Figure 2.4: Storytelling Systems and Games that feature more than one type of Camera. From left to right we have *SpellForce: The Order of Dawn* (Godlike), *Dragon Age: Origins* (Godlike) and *The Elder Scrolls V: Skyrim* (Third Person).

2.6.1 Cameras

Visualization of a virtual world has been a constant point of thought and research ever since computer systems became capable of rendering data. From the very first research done on this subject, it was clear that this problem would not be solved by a single approach. In the end, the type of camera to use is highly dependent of the system that one tries to make. The problem lies when a system can use a multitude of camera types to view the data. Is one more viable than the other? Will it influence the users' actions in any way?

Here lies our very first problem. Most systems and games barely discuss the camera and its advantages and disadvantages at all unless the system was specifically designed to test the camera, instead focusing on other parts of the work, such as story building or character planning. While these are key components of interactive storytelling, very little work actually devotes itself to seeing how the camera can influence these system's ability to deliver a solid story that users can comprehend.

Most systems assume a camera type to be the best to use, according to its characteristics, and use it. But they rarely explore the possibilities of using another type of camera to suit its needs.

From the presented research, we have seen how a character can establish a connection to the world and its characters.

In the **First Person Camera**, the player trades the ability to position himself with precision in order to increase his proximity to the character (often times *becoming* that character). Most often in this case, the game's main character tends to be a blank slate, a vessel for the player to project onto. In this aspect, this type of camera allows for the greater immersion and impact to the player, by bringing the world closer to her. However, it also has the potential to break immersion by not allowing the player to see what she's doing. The examples stated before, as when the player could not tell if attacking another character was working, are clear cases of this. In another example, in *The Elder Scrolls V: Skyrim*, when moving objects, the object positions itself at the center of the camera and suspended in mid air without anything to hold it in place. This can be fixed to a certain degree by showing the character's body on the screen, but this new feature creates another disconnection: it breaks the illusion that the player is the character.

Fewer and fewer games use **Fixed Camera Angles** as a recurrent type of camera during

gameplay: its use quickly decayed as graphical capabilities of computer graphics evolved and three dimensional environments could be rendered as fast as a static image. Its main strength is to provide a more cinematic experience to the players, but at the cost of visibility of the world and positioning. It is by far the most restricting camera type analysed and, as such, should be used carefully. Most games nowadays feature this type of camera in small segments, such as conversations or other interactions with characters, using any of the other types for the rest of the gameplay.

In the **Third Person Camera**, the character is visible to the player, letting her see what the character looks like and does in greater detail. This lets the player connect better to the actions her character performs on the world, at the cost of distancing both from each other. However, it also lets the player see more of the world at once, and allows the player to analyse the character's surroundings faster when compared to the First Person Camera. This camera is most often used in games where the character is fairly established in the world's lore, but the developer still wants the player to connect to her. In this way, the player knows that she's not the character, but by controlling her actions, will inevitably feel connected to what she does (though potentially not as much as a First Person Camera would allow).

The **Top-Down or Godlike Camera** is more commonly used in games or simulations where the player either does not have a physical character to control in the world or he has to control/maintain more than one character. This camera is the best to use when trying to view the world as a whole, but ineffective to generate player attachment to it as well as the other types of cameras do. The characters will lack detail and expressivity, since the player will see them from far away.

In short, all these cameras can be summarized in the terms of User Attachment versus World Awareness, where the First Person Camera occupies the User Attachment end of the spectrum and the Top-Down or Godlike Camera occupies the World Awareness Spectrum, as can be seen in figure 2.5.

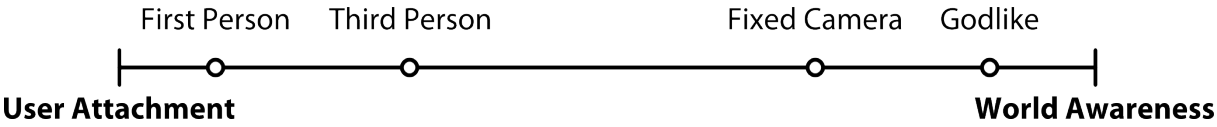


Figure 2.5: Relation between Camera Types and the kind of engagement players get from them.

2.6.2 Player Control

Player control is a very important aspect when discussing the camera: not every type of control will work with a certain type of camera.

All types of user controlled cameras we have discussed (save for the Fixed Camera, which is not controllable) tend to favour a **Flying Vehicle** type of control (which we talked about in in Section 2.1). Depending on the type of camera however, certain parameters will be defaulted or severely limited in its value:

- The **First Person Camera** sticks its position with the character's eyes, and follows the character's movement.

- The **Third Person Camera** is more relaxed when it comes to its position and movement. It can be located either very closely to the character (by her shoulder) or far enough to see the character's whole body. Normally, this type of camera does not allow the height to change. When the character moves, the camera can follow her at a precise distance or have a small fallout in which it does not move, making the transition between following and stopped states slower and more fluid.

- The **Top-Down or Godlike Camera** is usually the one with most freedom. Depending on the game, it can zoom in and out of the world, move around in all directions and see from a wide range of angles.

Most of these cameras support a wide array of controllers. Usually, the keyboard and mouse combination is by far the most used, but some games can also go by with just one or the other (though this will invariably have to limit the control of either the camera or the character). Console games tend to use joysticks, but these offer less buttons and thus need to either cut on controls or make controls context sensitive.

The use of motion controls is still very jarring to the user, but can achieve reasonable results with games that feature simpler control schemes. These normally encompass Top-Down or Godlike cameras, where the gestures are more broad and sweeping and thus are easier to recognise.

2.6.3 Interactive Storytelling Building

On the development of Interactive Storytelling, there are many ways one can go about to tell a story. In this section, we will provide an analysis of the methods presented so far. These will be important when we go about to develop our own work.

Core of Progression

When developing a system, it is useful to analyse what the focus of the work is. In this regard, Story-Centric and Character-Centric approaches are best for respectively representing a coherent plot structure or a strong characterization of the characters that populate the world.

As for mixed approaches, although they feature some level of distinction between characters, they are also more plot-focused than Character-based approaches, not having an equally high degree of generation in comparison. Of course, this comparison is based on the assumption that each system is comparable in number of actions a character can perform and possible story states are fairly equivalent.

Still, mixed approaches might be the best of two worlds, by establishing a greater cohesion between plot and characters.

Planning

Planning depends greatly on the core of progression. When going for a more story-based approach, a more structured planning approach is normally used, like the HTN. Constraints are also a possibility, and can allow for a greater variety in story decomposition. This is due to the

more loose approach when choosing the next action by using pre-conditions, while the HTN has a clear set of action sequences that must be performed for a plan to be successful.

When doing a Character-based approach, a more flexible planning tactic is normally a better option. For this purpose, HSP is widely used, since it allows a greater generation of rational actions by incorporating long-distance relations between them.

When using a mixed approach, it depends on the focus of the story. In *Façade*, since the focus is on the characters and their relationship with the user, a beats approach works well, because it focuses on character interactions and the subtlety of character discourse to convey their feelings to the user, while also advancing the story.

On other systems where the focus is the story as a whole, a more plot-centric planning choice is best. The blackboard is a good example of this: creating a backbone for the story but letting the characters decide how to reach the goals by applying their own skills.

Combinations in Interactive Storytelling Systems

Now that we have established the needed components to create an interactive storytelling system, we should see what other storytelling systems are providing in order to put our research into context.

From some the systems we analysed, we collected information based on the evaluation parameters we specified in this chapter. The information can be consulted in table 2.1.

System	Core of Progression	Planning	Has a Controllable Character?	Type of Camera
Teatrix[29]	Story-Based	Propp's Formalism	Yes	Fixed
Façade[25]	Mix	Beats	Yes	First Person
Friends TV Show[8]	Character-Based	HTN	No	Godlike
Pink Panther[9]	Character-Based	HSP	No	Godlike
Organizing a Party[12]	Character-Based	HTN	No	Godlike
Madame Bovary[32]	Character-Based	HSP	Yes/No	Third Person/Godlike
Merchant of Venice[35]	Story-Based	Constraints	No	Godlike
INTALE[43]	Mix	HTN / Beats	Yes	Third Person

Table 2.1: Some of our analysed storytelling systems and information about them in terms of areas relevant to our study.

One aspect worthy of note is that whenever a Godlike Camera is featured, there is no character to control whatsoever. Some of these systems provide a way to interact with the world, but this is normally done through a disembodied entity that acts like a "god" of sorts: moving, taking and putting objects on the scene and even influencing and following the character's actions in the world. The Pink Panther and Merchant of Venice examples are more proofs of concept than a "game" on their own and only allow the user to view the story as it occurs, even though the latter allows the user to choose several points of view from which to experience the story.

Most of these systems are also character-based. This can be explained by the fact that developers want the users to connect to the characters. The story can take a secondary place in this case. Even tough a mixed approach is a best case scenario, it is still under heavy development and has no distinct default approach.

The Godlike camera is the most used in our sample, followed closely by the third person camera. This might be due to the impact that having a free-moving camera can make it easier

to traverse the world and see many branches of the story happening at the same time. But, as we have stated, these systems do not normally involve a controllable character and, taking that into account, the Third Person camera is actually the most used type.

Chapter 3

The Game and Camera Model

After our analysis on Games and Interactive Storytelling in the previous chapter, it is now time to pool all that information in order to establish the base of our work.

In order to do so, we need to review what we set out to do in he first place. In section 1.2, we introduced our initial questions that led to the beginning of our research:

> *Given a camera mode, will the player be able to perceive the events of the story? Will he understand the overall arc of the story? The relationship between the characters? Can he understand their motivations and goals?*
>
> *Can the camera affect the way a player experiences a story?*

Just by this small set of questions, we have several needs for our system:

- A **virtual world**, where everything else will live.
- A **story** that occurs in that world.
- **Characters** to populate the world and help with the story driving process. These characters will need to:
  - **Interact** between them based on their **relationships**.
  - **Act** independently, based on their **motivations** and **goals**.
- The **camera** that allows this world to be seen, and the main focus of our research.

In this chapter, we will describe how to bring this world to life based on our previous research, and what technologies, methods and algorithms we'll use to make our solution a reality. We'll begin to describe how each of the previous mentioned components will fit into the overall model.

3.1 The World

The world is the base for everything else. Is is where the characters will live, and the story will unfold. In keeping up with the most recent storytelling systems and games, it is important our world be fully realized in three dimensions. This allows the users to experience a world that

functions closer to the real world, increasing their immersion and understanding of space, and also create a world with greater depth for the characters to live in. More importantly, this is the only way we can fully realize the camera types we have discussed earlier.

3.2 The Story

Considering that so much emphasis is put on the characters, including their relations and independence, it is inevitable that we should choose a character-based approach to telling our story, making it far more flexible and unpredictable by relying on the characters to tell the story through their own means and actions.

In this type of storytelling, there usually exists a high level goal that is not explicit, but can be derived from each characters' goals. By advancing their personal goals, each character makes the story progress. Sometimes, these character goals will collide with each other, and only one will be allowed to succeed. In these cases, it is important to have fallbacks that allow these characters to continue with their own stories despite failures.

This of course, presents some notable drawbacks, as we have discussed previously. These types of stories do provide better character development and characterization, but this increased free will the characters are given comes at the cost of a looser story structure. This can have consequences in the impact the story has on the viewer, leading to the possibility that events might not even happen at all, thus making the story bland.

Competition of resources, such as characters and other objects, might also lead to this problem: a character with good sense of timing will have every resource available to him, leaving another character to do nothing because what he needs is always out of reach. This is why it is very important to consider character and object placement when creating the world: when two characters want the same object, the one furthest away will most likely end up being outplayed. Making sure the items and characters are located so that they have similar chances of obtaining and/or interacting with them is key.

3.3 The Characters

After establishing our main story focus as focusing on the characters, it is important we delve deep into how those characters will drive the story forward and make it happen. As presented in the previous section, our characters will do this by means of their planning abilities and actions. We decided that using HSP (introduced in section 2.2.2) or a close approximation might be the best solution for our system in this case. This lets the character choose between the possibilities at runtime during the story's progress.

Each character will have an **ultimate goal**: what the character will strive to accomplish during the course of the story. Any phase of this goal might include the gathering of resources or the interaction with other characters, potentiating resource conflicts that can lead to a more interesting tale and a way to increase story variability.

This ultimate goal will be in turn composed of **abstract sub-goals**: the moment to moment decisions about what the character should do next, but without commitment to a specific course of action. They have pre-requisites for triggering and effects that must be obeyed further down

the chain (meaning, these effects must occur/ be true in order for this goal to succeed). Also, these goals can have conditions that make it fail instantaneously upon their verification. Abstract sub-goals, once triggered, are refined into **concrete sub-goals**, where the character decides and commits to a specific idea that will satisfy the abstract goal.

Concrete goals will in turn be completed by chaining **actions** in a logical way, based on their **pre-requisites** and **effects**. Pre-requisites state the properties that must be true in order for that action to begin, while effects are the properties and their change in values when an action completes, either successfully or unsuccessfully. Actions might start by being partially abstract, by not having certain parameters defined, and these must be refined by taking the other actions that either precede and follow it into account.

When an action fails to complete for some reason (lack of resources, resource in use, inability to reach target, etc), the concrete sub-goal that action spawned from fails automatically as well. However, this does not mean that the abstract sub-goal nor ultimate goal will fail automatically: it is important to give the character several options to continue, so she can form a new either abstract or concrete goal that she can comply with. This will both make the character more believable and intelligent, as well as more flexible.

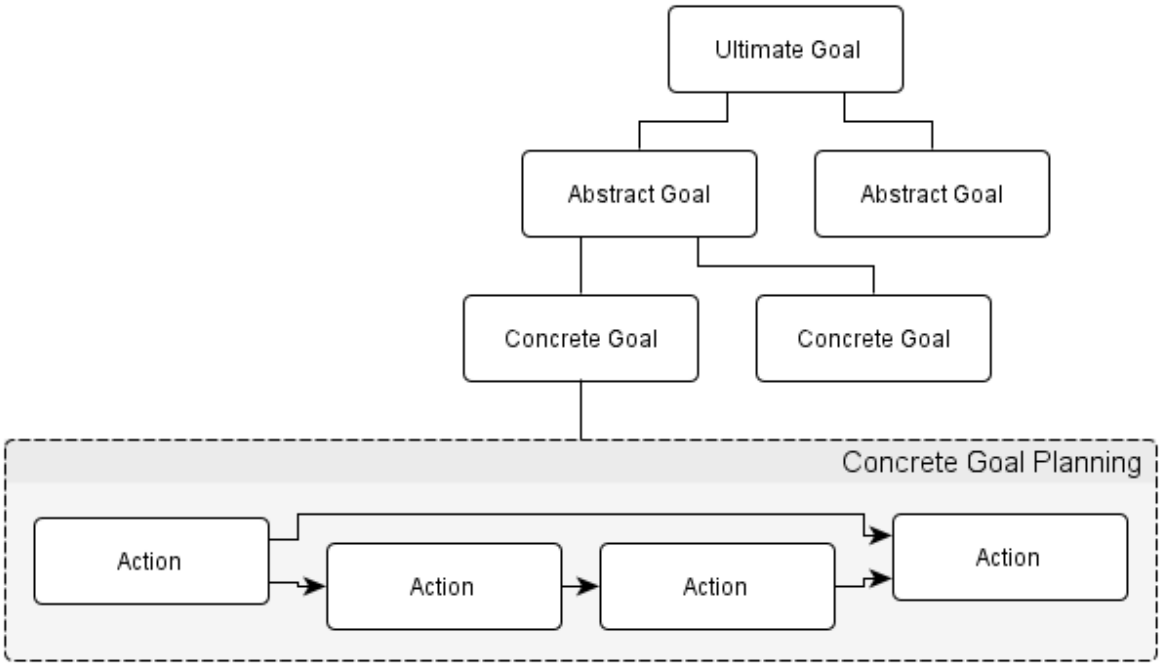


Figure 3.1: A decomposition of the model of our characters' planning.

As an example, consider a character whose ultimate goal is to go about and fulfil her basic daily needs. An abstract goal in this scenario will trigger whenever her hunger property goes above a certain threshold or attains a certain value: she must eat. But this "eat" goal does not commit to any course of action just yet: the character can complete this goal in a multitude of ways depending on the available resources around her. She can get an ice cream from the freezer just as likely as to go out to the forest and hunt for an animal: it all depends on the world and what it provides.

Remembering she had made soup just the day before stored on the fridge, the character

decides this will be her meal, thus refining her abstract goal into a concrete one: eat the soup.

But to eat the soup, she must do a series of actions. The most obvious one is that she must eat the soup, thus suppressing her hunger and completing the goal successfully. But, to eat the soup, she must have it with her: since it is in the fridge, she must go and get it. But the soup is cold too, and she'd rather enjoy a hot meal: she must heat it using a nearby appliance. This appliance can be any of the ones that can heat food, such as the stove or the microwave.

These actions are then sequenced and the character tries to follow their logical order. To either eat or reheat the soup, she must have it in her possession, thus making it the obvious first. She can't eat it immediately, since the soup is cold, so she must reheat it before eating it. Since there is no real difference between heating it in one place or another, she can shoose either of her options and opts for the microwave. By defining these dependencies we now have an order for our actions: get, heat and eat.

Now imagine she gets to the fridge and the soup is not there anymore. This makes the action and the eat soup goal fail. However, she is still hungry, and decides she still needs to eat something else. She remembers other food she might eat and the cycle of planning begins anew.

3.4 The Camera

The camera will be our focal point, since it is on it that hinges much of our research. Our idea is to provide a way to evaluate how common camera patterns used all over games and IS systems fare in giving users the context needed to understand an interactive story.

We identified these types of camera in section 2.4 as being the **First Person Camera**, the **Third Person Camera** and the **Top-Down or Godlike Camera**. But before delving into the specifics of each camera, it is a good idea to review how all cameras operate. We will continue by explaining the basics of how any camera operates.

3.4.1 Camera Basics

The camera is a special object in any virtual system, because it declares parameters and functionality that define much of how a user sees a virtual system. The camera can be set up in any number of ways and provide different looks and feel depending on its configuration. In order to operate, the camera needs to have the following attributes correctly defined:

- **The Projection Type**, chosen from two possible types:
  - **Perspective**: this type of projection features a viewing volume of a 4-sided pyramid frustum (with its apex cut off) and whose angle of visualization is controlled by the **Filed of View**. Objects farther from the camera will appear smaller and angles can be slightly distorted, much like peoples' eyes function. As such, this type is the most commonly used in 3D applications, since it provides a more realistic visualization for the viewer. We used this type on all our cameras for this very reason.
  - **Orthographic**: this projection uses a viewing volume in the shape of a box, whose size determines the visibility of the world. Objects' size and angles maintain their

values in this kind of perspective, making it useful for object analysis, but becomes hard when trying to traverse a 3D world since the depth of the world is lost.

- **Clipping Planes**: The near and far planes represent respectively the minimum and maximum distance that the viewing volume will contain. Anything behind the near plane and in front of the far plane will not be drawn.

Tu use the camera, it also needs a variety of functions that help with its management, including one of the most important: the way to position and rotate itself so it focuses on a target, which we will describe in the next section.

The *LookAt* Matix

Positioning and rotating the camera so it aligns to a certain view is a problem that all camera types share. This is done by applying a *LookAt* matrix to the camera's own transformations. The *lookAt* matrix is calculated through performing a series of operations with vectors and matrices[45], as we will explain here.

Consider the eye vector, that represents the camera's initial position, and is defined as:

$$\overrightarrow{eye} = \begin{bmatrix} eye_x \\ eye_y \\ eye_z \end{bmatrix} \tag{3.1}$$

We also need the center vector, which tells us where the camera will point to:

$$\overrightarrow{center} = \begin{bmatrix} center_x \\ center_y \\ center_z \end{bmatrix} \tag{3.2}$$

By subtracting both these vectors, we get the forward vector, that begins in the eye position and points to the center position:

$$\overrightarrow{forward} = \overrightarrow{center} - \overrightarrow{eye} \tag{3.3}$$

$$\begin{bmatrix} fwd_x \\ fwd_y \\ fwd_z \end{bmatrix} = \begin{bmatrix} center_x \\ center_y \\ center_z \end{bmatrix} - \begin{bmatrix} eye_x \\ eye_y \\ eye_z \end{bmatrix} \tag{3.4}$$

The forward vector is then normalized (we will assume this from now on in our equations), and proceed to calculate the right vector, an auxiliary helper calculated through the cross product between the normalized forward vector and generic up vector (usually a vector pointing

towards the y axis). This will help us determine the true up vector, used in the final transformation matrix applied to the camera:

$$\overrightarrow{right} = \overrightarrow{forward} \times \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{3.5}$$

To keep this explanation as simple as possible, we present solely the result of the cross product without engaging in mathematical proof. Thus, the right vector will end up as:

$$\overrightarrow{right} = \begin{bmatrix} fwd_z \\ 0 \\ fwd_x \end{bmatrix} \tag{3.6}$$

Since both vectors are unit vectors, the right vector is already normalized, and can be used to calculate the up vector:

$$\overrightarrow{up} = right \times forward \tag{3.7}$$

$$\overrightarrow{up} = \begin{bmatrix} side_y fwd_z - side_z fwd_y \\ side_z fwd_x - side_x fwd_z \\ side_x fwd_y - side_y fwd_x \end{bmatrix} \tag{3.8}$$

For the same reasons as before, the up vector is already normalized as well. We can now compose the rotation matrix, that is given by:

$$rotMat = \begin{bmatrix} side_x & side_y & side_z & 0 \\ up_x & up_y & up_z & 0 \\ -fwd_x & -fwd_y & -fwd_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

With this, the final camera transformation matrix will be the combination of this rotation matrix with a translation matrix defined by the eye vector given earlier:

$$transMat = \begin{bmatrix} 1 & 0 & 0 & eye_x \\ 0 & 1 & 0 & eye_y \\ 0 & 0 & 1 & eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

We must position the camera and then rotate it. Taking into account that the last matrix will be the first to be applied to an object, the combination of matrices (our LookAt matrix) will be:

$$lookAtMat = rotMat * transMat \tag{3.11}$$

$$lookAtMat = \begin{bmatrix} side_x & side_y & side_z & 0 \\ up_x & up_y & up_z & 0 \\ -fwd_x & -fwd_y & -fwd_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & eye_x \\ 0 & 1 & 0 & eye_y \\ 0 & 0 & 1 & eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

$$lookAtMat = \begin{bmatrix} side_x & side_y & side_z & side_x eye_x + side_y eye_y + side_z eye_z \\ up_x & up_y & up_z & up_x eye_x + up_y eye_y + up_z eye_z \\ -fwd_x & -fwd_y & -fwd_z & -fwd_x eye_x - fwd_y eye_y - fwd_z eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

3.4.2 First Person Camera

The **First Person Camera** will be intimately connected to the main character, by sharing its position with the position of the character's eyes, and looking to the character's front. Despite this type of camera being a proxy for the character's (and the player's) head, tilting (rotation along the z axis) is often disabled in order to reduce controls and confusion for the user. A normal person can do this in real life this by tilting her head, as the name would imply, but doing so in a virtual world would introduce a far less needed freedom that would be difficult to map to a 2d mapped input device (such as the mouse or a joystick thumbstick), already used to looking left and right (rotation along the y axis) and up and down (rotating along the x axis).

This camera functions quite similarly to **flying vehicle control** metaphor talked about on section 2.1, which was rated by users as the easiest present metaphor to traverse 3D worlds.

3.4.3 Third Person Camera

The **Third Person Camera** is also connected to the main character, but with less emphasis than the First Person Camera. This camera hangs a certain distance from the character, far away enough to view her but close enough to still maintain a personal view and perspective. Also, this type of camera usually does not sit exactly behind the character, but instead looks over her shoulder in order to let the player view a bigger portion of what is in font of the character.

To place this camera, there are more calculations required than in the First Person Camera. In this one, the character's front and right vectors (the exact same concept presented in section 3.4.1 but applied to the character instead of the camera) are needed. They are then used to determine the point from where the camera should look over the character's shoulder by calculating three vectors:

- The **horizontal distance vector**, calculated by the character's front vector multiplied by the negative distance we want the camera to be away from the character (since we want the camera to be behind him).
- The **vertical distance vector**, which is obtained by multiplying the character's up vector with the distance we want the camera to go up from the character's position, in order to look down on the character and be able to see more of her body (and the floor if possible or desirable).
- The **side distance vector**, calculated using the side vector of the character and multiplying it by the distance we want the camera to shift to her side, so the camera looks over her shoulder.

These are all added up to determine the point where the camera should be placed. Also, we define an extra 3D point with the help of the character's forward vector and multiplying it by a distance to get our **LookAt** point, which will become the center of our camera perspective.

The most common type of control provided for this type of camera is to move each time the character moves or turns around, being more constrained, but more consistent. There is, however, another wasy to control the camera in this mode: the user can rotate the camera around the character at will and use it to set the character's direction when moving. In this case, the camera orbits around the character at will, and movement depends on the direction the current camera is facing (as opposed to the camera being dictated by movement). This can be potentially more confusing, since the player is controlling two different entities at the same time instead of just one. On the flipside, it provides greater freedom of control. Because of this, we decided to employ a direct relationship between the character's and the camera's direction when rotating the camera.

Much like the First Person Camera, this camera draws great similarities with the **flying vehicle control** metaphor talked about on section 2.1, although in this case, the player can see the object rooting them in the world.

3.4.4 Godlike or Top-Down Camera

The **Godlike or Top-Down Camera**, unlike the previous two types, has little to no attachment to any characters in the game whatsoever. In this type of camera, its position and rotation provide much more freedom than any other type. The player can move it around (sometimes in all directions), zoom in and out to get a more detailed or broad view, even rotate the camera around to see the same scene from different angles. Some of these can still be constrained (such as preventing the camera from moving outside the world's borders, or rotate the camera to view the sky/ceiling), but its lack of focal point and anchor in the world (which was the character in the previous two) allows the player much more freedom to see the scene how she wants and from where she wants to.

Since this camera also uses the **flying vehicle control** metaphor when moving around (referenced in section 2.1), it provides an interesting dynamic. It can still be used similarly to the other types of cameras (as in the player can follow a specific character in a similar fashion as the previous camera types), restrictions willing, but most of the time it fails in this regard

due to its intended use with long and sweeping pans that allow the view of areas in a broader perspective, forcing the camera to lose precision during movement when closer to the ground.

There are two types of rotation used by these cameras:

- **Rotation in place**, where the camera stays in the same place and circles around itself, providing a 360 degree view of its surroundings. This type of rotation makes it easier to explore the world, but difficult to study one object in detail from several angles, since the camera has to be repositioned each time a rotation happens.
- **Orbit**, where the camera positions itself in a circle with the center on its focal point (what it is looking at). This makes it easier to study a point of interest from various angles but creates confusion when trying to navigate the world, since its position changes each time a rotation is performed (much like Ware and Osborne identified when using the **scene in hand** metaphor introduced on section 2.1).

Although this is not a very used type of camera when a single character is involved, we deemed important to compare its results against the other two most common used types. This, however, means that a different control scheme needed to be developed to play to this camera's strengths and weaknesses. The camera will focus on the main character at first, in order to let the player make the connection that he will control her during the experience, but will lend the player full control beyond that point.

Chapter 4

Implementation of our System

After establishing the theory behind an Interactive Storytelling system, and ways the player can control both the characters and camera, it is now time to delve into how our system was developed.

Our game took inspiration out of one of Charles et al.'s earlier work in their Interactive Storytelling research[11][8][13][10][7]. In it, the characters are all autonomous, and the player is a disembodied agent that can influence the world by manipulating its objects and thus mess or help the characters' plans. In this case, the characters in the set are a group of friends, and one of the characters wants to take another on a date. In order to do that, he needs to know what she likes and woo her with gifts and other things she might like. By interacting with other characters and objects in the environment, he can collect clues that can lead him to success. The setting and specific characters is further developed in the next chapter.

In this chapter, we will discuss how we implemented their artificial intelligence (AI), using the FAtiMA agent architecture, how we created our virtual world using Unity3D and finally how we brought the whole system together by linking them using the ION simulation framework.

4.1 The Game's Architecture

In this section, we will describe how we brought the Conceal world to life as a 3D application. This was a very important step in the process of our research, since it will be in this representation that we'll add the camera and implement user control to let players experience the story.

4.1.1 Creating the Game World

In order to make our Game a reality, we decided that using a game development engine was the best approach. To make our game, we used the Unity3D engine/game development tool, already used in several commercial titles[1] and that has proven to be an easy to use and fast prototyping tool.

In Unity3D, a game is an hierarchy of Scenes, Game Objects and Components (figure 4.1):

- **Scenes** are the levels or virtual spaces in the game.

[1] http://unity3d.com/gallery/made-with-unity/game-list

- **Game Objects** are generic containers that represent entities in the game. These can be organized into an hierarchy by putting Game Objects as children of other Game Objects.
- **Components** are small sets of data or behaviour that can be attached to a GameObject to change its functionality, visibility and appearance, and even behaviour.

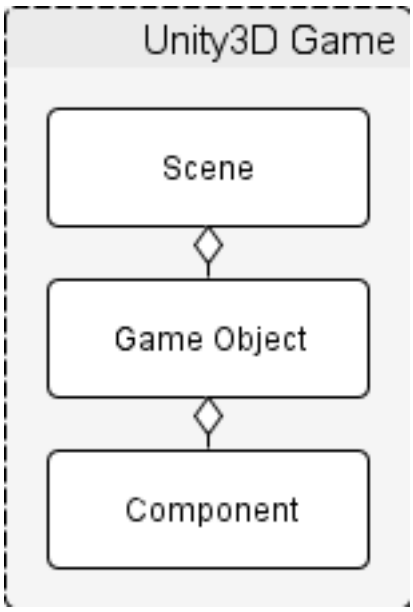


Figure 4.1: The main components in a Unity3D game's architecture.

A Game Object starts with a single required component that defines its position, rotation and scale in the Scene, the *Transformation Component*. From there, all other components that can be added are optional and can be grouped as needed for a specific type of entity. For example, any entity with a 3D representation will need a 3D model to represent it on the scene (a *Model Component*), a texture to place on the model (a *Material Component*) and a way to combine both (a *Render Component*).

The Architecture of our system features several types of Game Objects: The **Manager**, the **Camera**, the **Pathfinding Grid**, the **Characters**, the **Interactable Items and Placeables**, and the **Decorative Items** (figure 4.2).

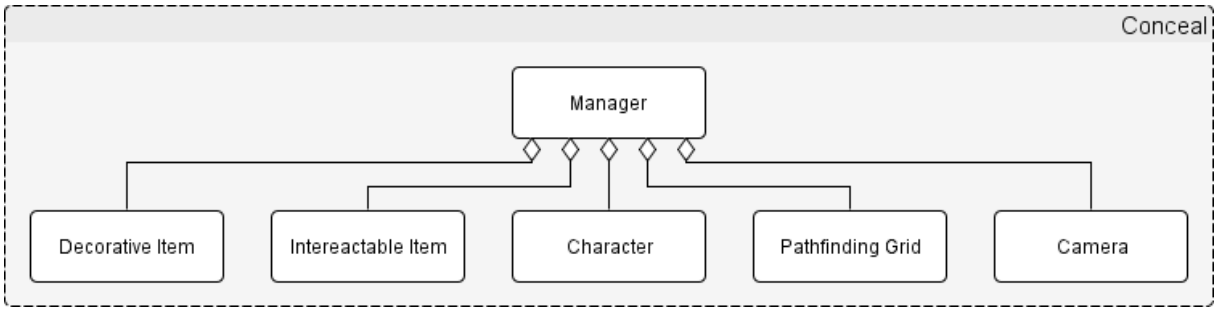


Figure 4.2: The main components of Conceal's architecture.

The **Manager** is an invisible entity responsible for organizing and maintaining all other Game Objects. It places (and tweaks as necessary) all the characters, items and camera at the

beginning of the game, keeps track of them throughout the simulation, and is also responsible for the information seen in the screen through the Graphical User Interface (GUI).

The **Camera** is our main focus in this research, and will be discussed in more detail in section 4.1.2. It houses a *Camera Component* that Unity3D already features and that we extended to make our camera able to change between presets in the beginning or even during the simulation.

The **Pathfinding Grid** was introduced in order to help our characters traverse through the scenario. It uses the AStar Path library[2], that creates a grid of the scene in real-time and determines paths point-to-point using the A* algorithm. Characters can then use these paths to get to their objectives.

The **Characters** are composed of three types of Game Objects, each with a specific function (figure 4.3):

- The **Body** is responsible for all of the Physical variables attached to the character, such as her position and orientation, the speed at which she should move through the world and even what she's carrying at the moment (her inventory).
- The **Mind** houses all of the intelligence the character has, including the actions she can perform and variables used in planning. These are directly tied to our Agent Framework FAtiMA, which we explain further on sections 4.1.3 and 4.1.4.
- The **Model** houses all the needed resources and variables that pertain to the character's visual representation: its 3D mesh, materials, textures and animations. This was done in order to keep the characters as modular as possible, and be able to create many similar characters by changing the model and/or textures alone.

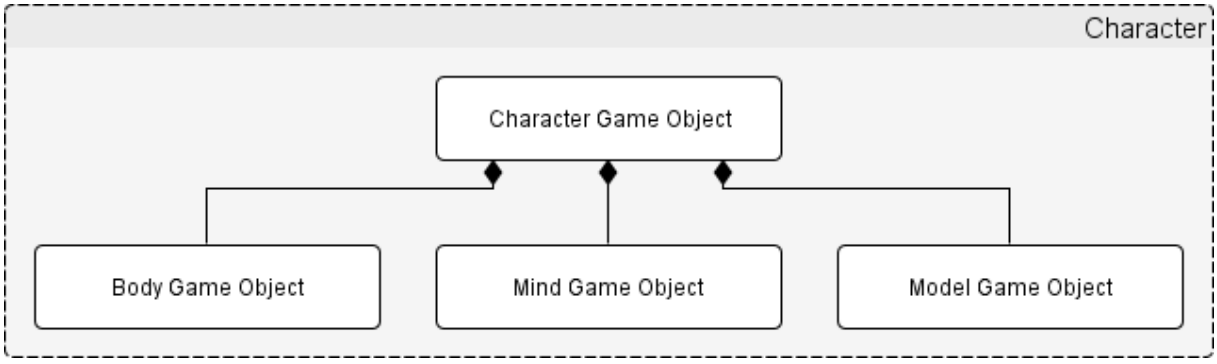


Figure 4.3: The main components of a Conceal character.

Interactable Items and Placeables are the main type of objects seen in the world. Characters can interact with them in order to complete their plans and advance the story. These objects feature a similar composition to the characters, the only difference being the fact that these items do not have a "Mind", but instead have an **Entity** Game Object that is directly tied to its ION representation (more on that in section 4.1.4) and houses all the variables needed for the characters to use that item in order to formulate their plans(figure 4.4).

Decorative items are never used by the characters. Instead, they fill the world to make the scene more recognisable, provide natural ways of restricting the playing space and offer greater

[2]http://www.arongranberg.com/unity/a-pathfinding/

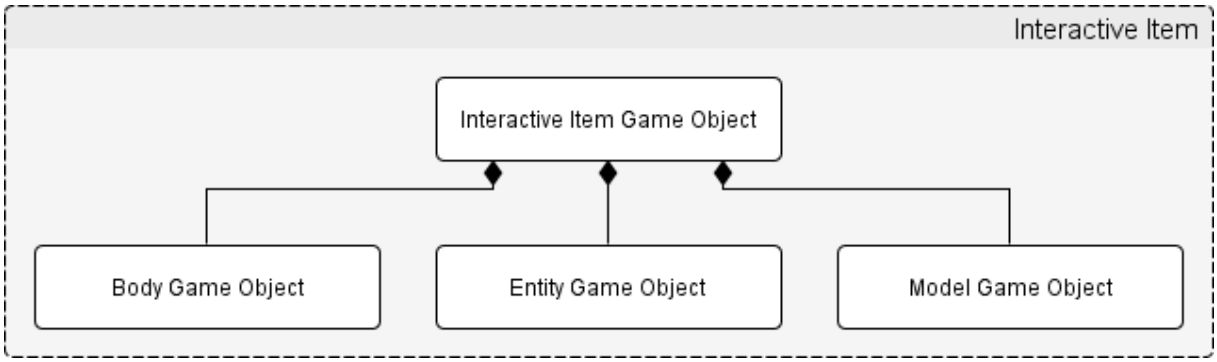


Figure 4.4: The main components of a Conceal interactable item.

visual variety to rooms. Because of this, they do not feature the "Entity" Game Object like their Interactable counterparts (figure 4.5). Examples of this kind of objects are the walls, the floor and ceiling, chairs, tables, desks and counters.

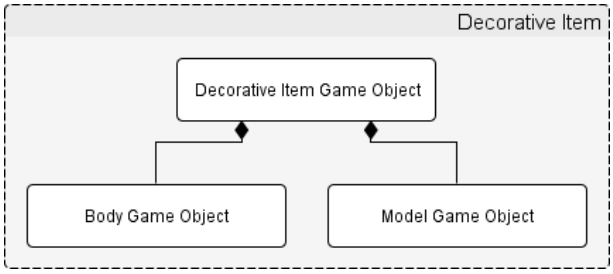


Figure 4.5: The main components of a Conceal decorative item.

By combining all these types of Game Objects we are able to then populate our virtual world. But this is not the only step needed to complete it. We need to focus on the camera and user control and imbue our characters with intelligence.

4.1.2 The Camera and User Interaction

In our project, we needed to ground the player's experience in a character from the world, in order to test how each camera would affect the player's understanding of the situations around that character. To do so, we took advantage of the already existing camera specific components present in Unity3D, but decided to add functionality to them to make it work like the camera types we mentioned in our Model.

Unity3D already has a pre-made *Camera Component*, that includes all the basic variables and functionality that a camera needs in order to function properly, and that we introduced in section 3.4.1.

After this, we expanded this component with one of our own: a *Script Component* (figure 4.6). *Script Components* allow the introduction of new behaviour for the Game Object they are attached to, the manipulation of other components, and even of other Game Objects, through code. The script uses the manager to track which type of camera it should use and any other information it needs about the character to update the camera, like we described in section 3.4.

The control scheme used in each camera is the scheme commonly used on most recent games and interactive storytelling systems that feature that type of camera, in order to provide some

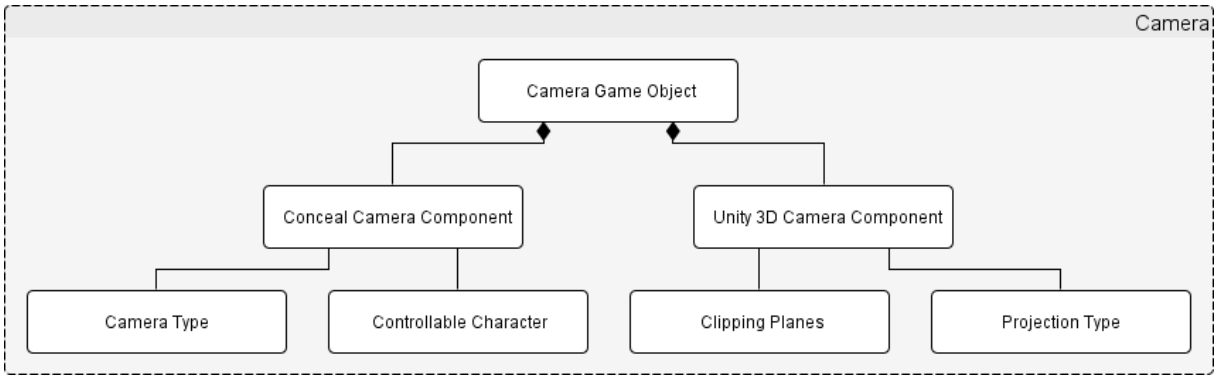


Figure 4.6: The main components of the Conceal camera.

familiarity. These schemes were done by attaching a script object to the playable character's "Body" Game Object, much like the camera itself.

In First and Third Person Camera, by moving the mouse away or towards her, the player can have the character look up or down. By moving the mouse sideways, the character rotates in that direction. Also, the left mouse button is used to interact with close objects that are targeted by the mouse pointer. The character's movement is controlled by the WASD keys or the directional keys.

In the Godlike or Top-Down Camera, we had to change this scheme in order to provide for a more fine-tuned control. Since the camera can be placed and orientated freely, we decided that using a point-and-click style of interaction was better suited in order to keep the player from becoming confused and still be able to perform all actions with a high degree of accuracy. By clicking on the floor, the character is instructed to move to that position using the pathfinding grid, while clicking an object will instruct to move the character to it and use it as soon as she reaches a comfortable distance. The player can move the camera in all directions using the WASD or arrow keys, zoom in and out using the mouse wheel and rotate it by right-clicking and dragging the mouse.

4.1.3 The Characters' Minds

As we have introduced in our model, our characters will be the main focus of our system when it comes to story progression. In this section, we present how we developed these characters in terms of characterization and action: the basis for their intelligence.

In order to focus on building the world, we decided that using an existing architecture for our agents' minds was the best solution. FAtiMA[14][20][15] is a fully mature model for character artificial intelligence with a good track record. It is used in several systems where characters are at the center of the experience, such as FearNot![30] and ORIENT[1], and uses a very similar method for planning and character action that we described in our model (section 3.3). But before going into detail on how our characters reason, we'll introduce how FAtiMA's architecture functions in a general sense.

FAtiMA employs a empty core template that defines how the whole architecture works. This template can then be extended by layers (or components) that add up to form progressively more complex lines of reasoning for the characters. This is done by running each layer in turn,

executing its main functions. If new events occurred, they are processed into the character's memory and fed once again to all layers to update them on the new status. Finally, it chooses an action to perform based on its Action Selection components. This process is repeated each update.

We will now explain how the main component we used, the Deliberative Component, works to bring our characters to life.

FAtiMA's Deliberative Component

The **Deliberative Layer**, like the name implies, is the component that is responsible for the character's reasoning and planning.

FAtiMA does not have a direct definition of an **ultimate goal** like we described in our model, but instead each character (represented by a role in FAtiMA) is imbued with a list of her goals, ordered by the importance that the character gives to their success and failure, as table 4.1 shows. Listing 4.1 provides an example of one such role.

Attribute	Description
name	The goal's name.
importanceOfSuccess	A number from 1 to 10 that symbolizes the importance this character places on this goal's success (a higher number will make the character focus on this goal before the ones with a lower number).
importanceOfFailure	A number from 1 to 10 that symbolizes the importance this character places on this goal's failure (a higher number will make the character more afraid to pursue this goal. Higher numbers might put him off going after this goal altogether).

Table 4.1: FAtiMA's goals from a character role's perspective.

```
<Character role="Bride">
  [...]
  <Goals>
    <Goal name="HamperGifting([gift])" importanceOfSuccess="10"
        importanceOfFailure="1" />
    [...]
  </Goals>
  [...]
</Character>
```

Listing 4.1: An example of how a goal fits within a character's role.

Goals are then defined in a separate file that all characters have access to, called the **Goal Library**. However, each character will only use the goals listed explicitly in their character role. Goals have several properties that are described in table 4.2.

```
    <ActivePursuitGoal name="HamperGifting([gift])">
        <PreConditions>
      [...]
            <Property name="[gift](type,gift)" operator="=" value="True" />
            <Property name="[gift](hidden)" operator="!=" value="True" />
            <Property name="[hider](hider)" operator="=" value="True" />
    </PreConditions>
        <SucessConditions>
            <Property name="[gift](hidden)" operator="=" value="True" />
```

```
        </SucessConditions>
        <FailureConditions>
            <Property name="[gift](validOwner)" operator="!=" value="True" />
            <Property name="[anyone](impressedGift,[gift],[impressed])" operator="
                =" value="True" />
        </FailureConditions>
    </ActivePursuitGoal>
```

Listing 4.2: An example of an active pursuit goal in the goal library

Listing 4.2 shows one such goal, where our Bride will try to hide any gifts that she comes across that haven't been hidden yet. The preconditions state that:

- The item to hide must be a gift (the property "(type,gift)" must be True).
- The item must not be already hidden (the property "(hidden)" must not be True).
- There must be a hiding place somewhere in the world (its "(hider)" property must be True).

The goal will succeed when the item has been hidden (the property "(hidden)" becomes True), and will fail if either:

- The item has been picked up by someone else and is no longer possible to obtain (the (validOwner) property is not True) (we will see how this is done shortly)
- Someone is already using this object to impress her father (in which case she will no longer seek it, as hiding it will no longer benefit her plan)

Attribute	**Description**
name	The goal's name.
PreConditions	A list of the conditions (either properties or events) that need to be met in order for the goal to be pursued.
SucessConditions	A conjunction of the conditions (properties or events) that need to be met in order for the goal to succeed.
FailureConditions	A disjunction of the conditions (properties or events) that make the goal fail when met.

Table 4.2: A FAtiMA goal's properties.

If the goal is actually pursuable, then FAtiMA's planner creates a sequence of **actions** to accomplish it. Actions, like goals, are all compiled under one file and have several parameters described in table 4.3.

```
        <Action name="Hide([item],[hider])" >
        <PreConditions>
            <Property name="[AGENT](person)" operator="=" value="True" />
            <Property name="[AGENT](has,[item])" operator="=" value="True" />
            <Property name="[hider](hider)" operator="=" value="True" />
            <Property name="[hider](canHide)" operator="!=" value="False" />
        </PreConditions>
        <Effects>
            <Effect probability="1.0">
```

```
10                  <Property [...] name="[item](hidden)" operator="=" value="True" />
11          [...]
12                  <Property [...]  name="[place](canHide)" operator="=" value="False
                        " />
13                  <Property [...]  name="[place](has,[item])" operator="=" value="
                        True" />
14                  <Property [...]  name="[AGENT](hasItem)" operator="=" value="False
                        " />
15                  <Property [...]  name="[AGENT](has,[item])" operator="=" value="
                        False" />
16              </Effect>
17          </Effects>
18      </Action>
```

Listing 4.3: An example of an action in the action library

As an example, the action shown in listing 4.3 will only take place if:

- The agent is a person (the property "(person)" must be True), and has the item it wants to hide in her possession (the property "(has,[item])" must be True, [item] being the input given to this action during the planning phase).
- There exists a hiding place (an object must have the "(hider)" property as True), and that hiding place isn't already filled with something else (the property "(canHide)" must not be False).

If this action succeeds, the following effects take place:

- The item becomes hidden (the property "(hidden)" becomes True).
- The hiding place now houses that item (the property "(has,[item])" becomes True) and thus can no longer be used to hide any other object (the property "(canHide)" becomes False).
- The agent no longer carries that item (the property "(has,[item])" becomes False), and thus no longer has anything in her inventory (the property "(hasItem)" becomes False).

Attribute	**Description**
name	The action's name, including the input parameters it receives.
PreConditions	A list of the conditions (either properties or events) that need to be met in order for the action to be performable.
Effects	A list of the effects (properties or events), including the probability of that effect happening, that take effect when and if the action succeeds.

Table 4.3: A FAtiMAs action's properties.

In order to build a plan, actions have to be sequenced so that an action's effects meet future actions' pre-conditions. This dependency can be extended as much as the first action having direct influence in the last action of the plan. FAtiMA begins by pairing the goal's success conditions with the actions that provide such effects. From then, it keeps building backwards

by chaining the last action's pre-requisites with the current action's effects until it reaches an action that the character can perform in her current state. If no action or an inconsistency is found, the current action considered is discarded and new actions are considered. If there are no more actions to consider, then the last action is dropped in favour of other alternatives. In the end, if a the chain of actions is able to be built, it becomes the plan the character will pursue.

When an action fails, the plan it belongs to fails as well, and the planner for that character is forced to either come up with a new goal to pursue (according to the order of interest the character has on each goal) or try to find another way to pursue that very goal (by coming up with a new plan for that goal that is able to comply with its requirements). If no plan is able to be devised, then the character focuses on the next goal, in order of importance.

However, if a goal fails, that does not mean it can't be pursued later, when circumstances change. Characters in FAtiMA can retry goals after their failure, provided they can generate a goal towards their completion again (be it the exact same plan or any other variant, depending on circumstances).

Finally, there is a special kind of action, called an **inference**. These actions are not used in planning directly, instead they are run every cycle of a character's thought process before the formal planning procedure and, if pre-conditions are met, carry out its effects instantly. These are distinguished from normal actions by starting their name with "Inference". One such action is described in listing 4.4.

```
<Action name="InferenceOperatorInvalidOwner([object])" >
      <PreConditions>
          <Property name="[object](owner)" operator="!=" value="SELF" />
          <Property name="[object](owner)" operator="!=" value="NoOne" />
      </PreConditions>
      <Effects>
          <Effect probability="1.0">
              <Property name="[object](validOwner)" operator="=" value="False" />
          </Effect>
      </Effects>
  </Action>
```

Listing 4.4: An example of an inference action

In this case, the agent will analyse an item's owner. If it's not either her ("SELF") or no one has it ("NoOne"), then the object has an invalid owner (the property "validOwner" becomes False). This change is then fed to the memory, where it will affect future plans involving this item.

4.1.4 Bringing it all together

After we established and developed both parts of our work: the characters' intelligence and the game, there was the need of bringing these two together to make our whole, final system. In this section, we will describe the process of linking these two modules. This is where the ION Simulation Framework we have discussed previously makes its appearance, and will make for our third and final module in our overall architecture (figure 4.7).

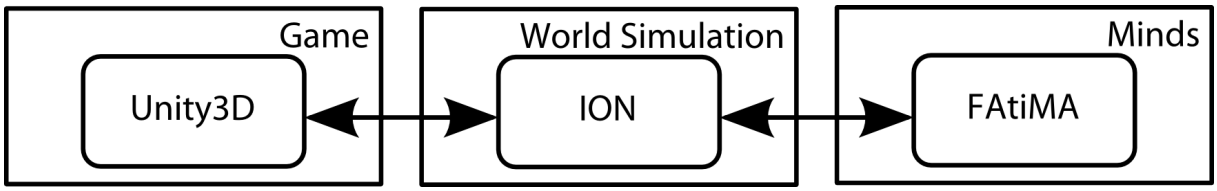


Figure 4.7: The three main components of our system's architecture: the world in Unity3D, the world simulation in ION and the agents' minds in FAtiMA.

The ION Simulation Framework

The ION simulation framework, as the names implies, simulates virtual environments, allowing the bi-directional contact between the agent's minds and the world properties they need to manipulate. This allows the minds to become detached from the system they are used and able to be re-purposed to any other system that satisfies its requirements in terms of properties.

The framework is composed of four key components: **Entities**, **Properties**, **Actions**, **Events** and **Groups**. Entities represent Objects in the world, which in turn have Properties and can perform Actions, thus generating Events. Groups are used to link Entities together, but were not used in our work.

The core aspect of this framework is to be able to keep an eye on concurrency when performing changes to the world. Instead of directly manipulating values during access, ION puts them on a queue that is processed each update, after which they take full effect after taking care of any conflicting changes by a priority discriminator that can be customized. By doing this, we guarantee that any Entity receives the same information that another has just accessed as well. When the update ends, and changes are made, an Event is created symbolizing this change, and is propagated to any Entity that subscribes to it (figure 4.8).

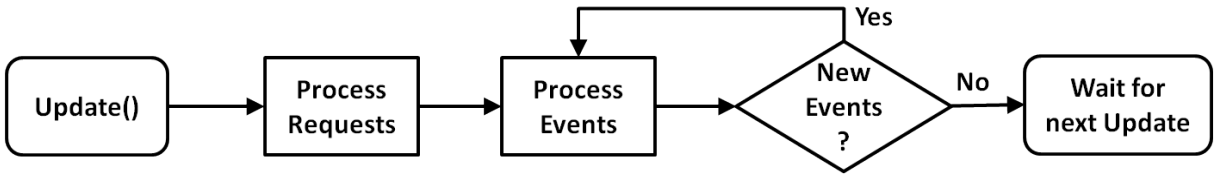


Figure 4.8: ION's update loop.

Integrating the Agents' minds in the Game

All that we need now to finish our system is to integrate this middle layer with the remaining two.

The integration of ION in Unity3D is done trough *Script Components*. The main component that needs to be present somewhere in the scene (and that we placed in the Manager Game Object) is the IONSimulation. This component is used to issue direct control over the updates carried over in the ION Layer.

For characters and other interactable objects, the components are stored in the "Mind" or "Entity" sub-Object of each and represent the various components of the ION Framework:

- **IONEntity** creates and adds an Entity to the simulation. It also scans the Game Object

it is attached to and adds all Properties and Actions present as part of the Entity, also registering all Events related to them. Only one Entity can be placed in a Game Object.

- **IONProperty** creates a Property and associates it with the Entity present in the same GameObject, defining Event handlers related to its changes. There can be more than one Property in the same Game Object.
- **IONAction** creates an Action and associates it to the Entity present in the same GameObject. There can be more than one Action in the same Game Object. Defines three event handlers:
  - **OnStart** is raised when the Action starts its execution.
  - **OnStepped** is called each update while the Action is being performed.
  - **OnStopped** is raised when the Action terminates, either due to success or failure.

Characters need one extra component: the connection to their specified role in FAtiMA. This is another *Script Component* that needs to be attached to the "Mind" called **FAtiMAMind**. This script launches a FAtiMA process and creates a representation of the Character's mind in ION, a RemoteMind, linking them both afterwards.

The RemoteMind then opens a socket between itself and the FAtiMA process it was attached to in order to exchange information. Events that occur in the Simulation are translated to FAtiMA Events and sent to the FAtiMA process to update all the parameters necessary for planning, while the FAtiMA process is responsible for planning and then telling the RemoteMind which action to perform next in conformity to the generated plan, in the form of Action OnStart requests.

In order for this exchange of information to occur, the RemoteMind also registers to all Events pertaining to the character Entity it is attached to and all other Entities present in the Simulation. However, this proved problematic in our case, since we did not want the characters knowing what was going on in the other rooms of the house that they did not have direct visibility towards. To counteract this problem, we decided to filter the Events sent by the RemoteMind to the FAtiMA process based on the proximity between the character's Entity and event location involved. New Properties that a character introduces to an Entity are also only visible to the character that places them. This means, for example, that a character will look for an object in the last place she saw it and if it was there the last time she was close to it, even if it is no longer there due to some other character moving it away while she was not around. Only when the character reaches the last place she saw the object will she update her information on it, and this will reflect on both her own internal knowledge base and plans. This not only increased the characters' overall intelligence and believability, but also served to add some tension to the game itself. Figure 4.9 shows the whole process.

4.2 An Example Scenario

Now that we have presented our system as a whole, including its various layers and how they communicate with each other, we can no present how they all work together with an example scenario. Consider the following scenario which can happen in our final system:

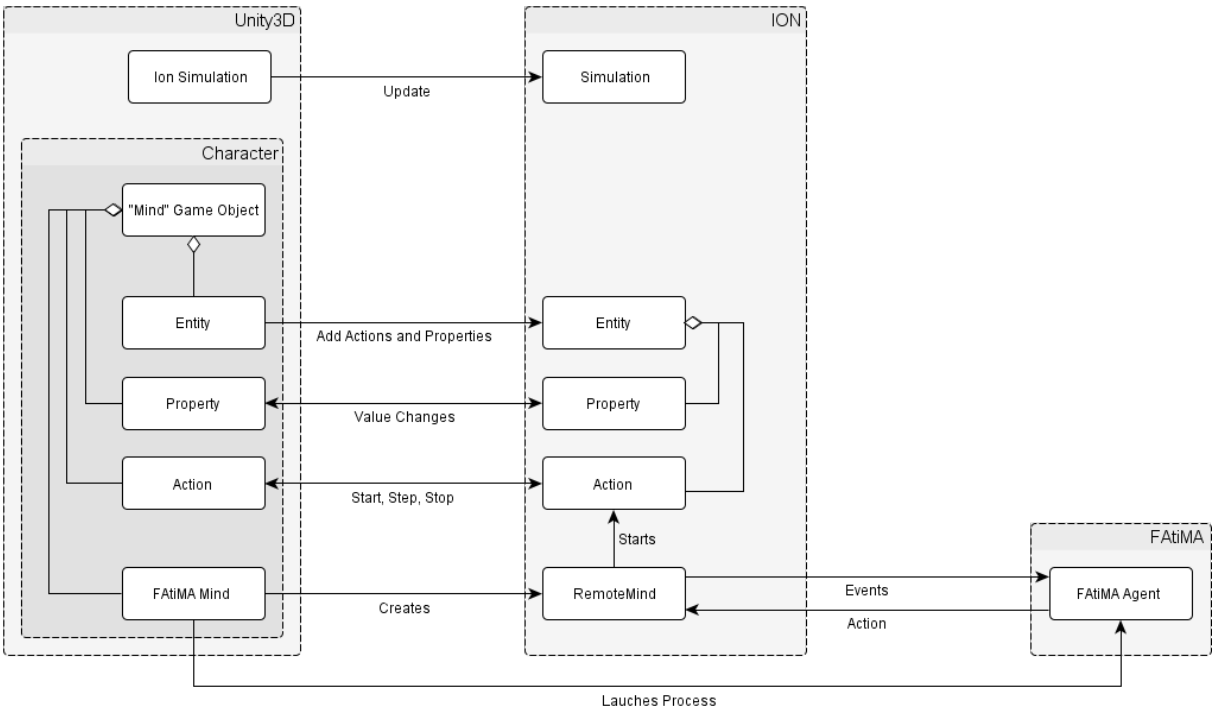


Figure 4.9: How the the three main components of Conceal's architecture exchange information.

At the beginning of the game, the player is presented with the point of view (let us admit the Third Person Camera) centered on her character. The player is then free to move about and start interacting with the world.

Imagine the payer controls the Bride, and the Groom is left to his own devices by means of Artificial Intelligence. The Groom's "mind" checks his role for his intended goals and decides to pick the highest ranking one: Gift an item. FAtiMA checks this goal from the library and then starts creating a plan for this, by checking in succession the actions needed for it to happen.

The item must be gifted, so a Gift action must occur. The pre-requisites for this action demand that he have an item capable of being gifted on him and having a person able to receive it. FAtiMA then checks for these properties and finds a person able to receive this gift: the Bride's Father. But he does not have the item with him yet, so he needs to get one. For that, he needs to perform a PickUp action. But this pick up needs to have certain restrictions based on the actions that follow it: it can't be just *any* item, it needs to be a giftable item. He then checks his knowledge base for items that can be gifted (by having the "gift" property). He chooses one, the box of Jewels, and the plan is now formed: he must pick up the Box of Jewels and then proceed to gift it to the Bride's Father. FAtiMa's planner reports this to the ION RemoteMind, and in turn it sets out to start the PickUp Action. The character's "Mind" Game Object then recalls the last place where he saw the Box and uses the pathfinding grid to check which nodes he should walk through in order to get to it. After the A* algorithm gives him the list of nodes he needs, he sets out to find the Box.

Meanwhile, in another room, the player moves and picks a nearby Sword. This change is reflected on the Sword's properties: it is no longer on the ground to be picked and its owner changed from "No one" to "Bride". The player decides to destroy the Sword using a vial of Acid nearby. She moves towards it and uses the sword with it, destroying it permanently. The

sword leaves the game, no longer has any owner and receives the "destroyed" property. This is only visible to the Bride so far, since she is the only one nearby.

While this is happening, the Groom managed to pick up the Box of Jewels (thus the Action completed successfully) and this is reported to the ION Mind, which in turn reports it to the FAtiMA planner. Since it was successful, the rest of the plan can be carried out, and a new request for an action to move out and gift the Box to the Bride's Father is sent. When reaching him, the Groom speaks about the item and gives it to him (the owner and possessor of the item is now the Bride's Father). The Father then analyses the item and, since it is a gesture of good faith, reacts positively. The Groom, after hearing his reaction, sets out to find the next item to show or gift to the Father. By checking his goals again, he finds that Impressing the Father with a Prize also has a very high priority. FAtiMA plans for the Groom to pick up the sword and then show it to the Bride's Father, since he yet thinks that the Sword is available. An Action to move out and pick up the sword is sent and the Groom sets out.

In the meanwhile, the player moves to another room to get the Groom's Crest and finds a can of Ink nearby. She gets closer and clicks on the Crest to pick it up and then approaches the ink and clicks on it to use it with the Crest. The Crest now has the "tampered" property. The player then gets back to the place where the Crest was and uses the waypoint to put it back in the same spot.

The Groom finally reaches the place where the Sword was, but it is no longer there! This makes him fail the Action, and when this information reaches FAtiMA, it drops this plan completely. FAtiMA decides to use another prize this time: the Crest. After forming the plan, the Groom sets out and goes to the Crest. When he gets there, he finds it is there and picks it up. Since he was not the one to tamper with it, he does not see the subtle changes the Bride made to it. When reaching the Bride's Father and gifting it to him, his meticulous eye notices the ink splattered everywhere, and thus reprimands the Groom for treating his items in such a callous manner.

Chapter 5

Conceal

On top of the systems presented in the last chapter, we devised a world that would let us use all these components to create a story. That world is *Conceal*, where you play one of the characters involved in a wedding proposition: the bride or the groom.

5.1 The Setting

A village in the Northern Continent is under serious pressure from the surrounding lands. Several warring enemies see it as a stepping stone to conquer their enemies and their leaders. So far, they have kept their neutrality and have not made their move. But the uncertainty of an attack is big.

Seeing a alliance with another village as a potential to win the respect of his neighbours, the leader of the village, Vilmundr, decides to arrange a strong oath to keep the alliance in check: a marriage. His youngest child, his daughter Arndís, would make for an excellent candidate.

Little did he know, however, that her adamant will to not get married was as strong as she is young. No matter what it takes, she will try to outdo this wedding. But she cannot afford to look selfish. She must expose and shame her husband in the eyes of her father, so that he deems him unworthy and sends him off and cancels the wedding.

5.1.1 The Characters and their Relationships and Goals

We have three characters in our story, each serving a different function (figure 5.1). In this section, we will present their individual stories and goals in the context of our system.

Characters

Arndís, the Bride Arndís is the youngest child and only daughter of Vilmundr, the village's chief. She is a young and spunky girl, with her knees continuously scraped and always getting into trouble. At 15 years old, she is young and inexperienced, but compliments that with her determination and attitude.

But all that is about to change when her father announces that she is to be married to a son of another village's chief. Outraged by this attack on her freedom, she is determined to foil his plan. But she knows better than to confront him directly, as he would never allow it.

Figure 5.1: The models of the characters in the final game: Arndís the Bride, Ragi the Groom and Vilmundr, the Bride's Father.

But she also knows he will not stand for an unworthy husband, no matter how many problems it will solve, and therein lies the solution to her predicament. She must try to do whatever it takes to shame and poison her to-be husband's opinion in the eyes of her father so that he cancels the wedding.

Vilmundr, the Bride's Father Vilmundr is a stern father and a skilful leader. His tribe looks to him for guidance, although he is sometimes seen as too harsh.

His arrangement to get his daughter married is a big investment to him, a way out of having to deal with petty warfare. In his eyes, it is a necessary sacrifice, no matter what his daughter or anyone else thinks.

His relationship with his daughter is sometimes strained by her rebellious spirit, even more so now that the marriage has been called for. Despite this, he regards his family highly, and carries his name proudly. Besmirching it would be a crime in his eyes.

Ragi, the Groom Ragi is a boisterous and eager young warrior. Being a son to a village's chief has given him plenty of room to indulge in his sometimes rash behaviour. Hanging out with his village's daughters and causing mischief has given him a reputation that was not easy to appease in the eyes of his father.

It is at his behest that he is now in the house of strangers to conquer the hand of another chief's daughter. As much as he sees this as a mere convenience for him and his father, he revels in the challenge to show off in order to win a girl's hand.

But this is much more than a personal challenge or a event where he can feed his ego. It is a way to gain prestige and reputation for his family and village. He is aware that his village benefits as much from this allegiance as her bride's and, as such, must tread carefully and refrain from embarrassing himself.

Relationships

Relationships are very important in order to establish what each character will accomplish and how, since character interaction is the key for story progression. These relationships are

summarized in figure 5.2.

The Bride and Groom share a relation of **Antagonism**. The Bride will try everything in her power to stop or hinder the Groom's plans by intercepting what he is doing in order to please her father and increase his opinion of him. She does not actively despise the Groom as a person, but more what he represents for her life. The Groom does not know this at the start, however. He will continue to follow his plans independently of the Bride's sneaky interruptions.

Due to this, the Bride and her Father are at **Tension** with each-other. This is a implicit relation, and will not manifest itself directly. Their strenuous relation will be put to the test with the marriage arrangements.

Since both the Bride's Father and Groom want the marriage proposal to succeed, they share a relation of indirect **Complicity**.

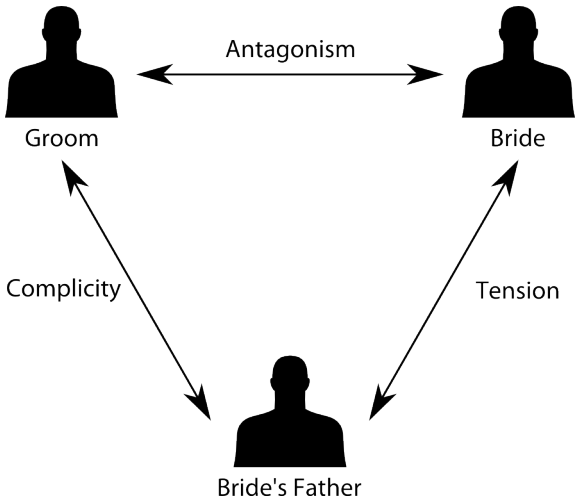


Figure 5.2: Relationships between the characters in Conceal.

Goals

The Bride As described above, the Bride's ultimate goal will be to not marry the Groom. In order to do so, she will employ several tactics, which may compliment each-other or be used on their own, in order to discredit her husband-to-be. These tactics are related to Gifts and Prizes and, as can be surmised, are directly tied to the Groom's plans (further defining the Antagonism relation that was introduced earlier). The Bride can steal and hide or destroy these items, or exchange gifts with less desirable ones.

The Groom The Groom's ultimate goal, as we have stated, is to gain favour with the Bride's Father in order for him to accept him as a worthy husband for his daughter. He will employ two main tactics to be successful: giving gifts and boast about his accomplishments through prizes.

The Bride's Father The Father's ultimate goal will be to and maintain his family's good name, with a secondary goal to increase his prestige. The marriage is the primary source for the increase in prestige he seeks. Marrying her daughter with the Groom, provided he is a worthy husband, will satisfy both his goals. If the Groom is later seen as a less than worthy candidate

for his daughter's hand, both these goals clash, since he cannot maintain his family's good name, which takes priority over the rest.

5.1.2 The Objects

When approaching our game, we wanted the interaction mechanic to remain similar to our inspiring system (as stated in chapter 4) in the way it engages the player, mainly with the repercussions the taking of objects from the scene might cause to the story's structure.

To keep the interactions between our characters a bit more interesting, we decided to include several types of objects that can work in combination with each other to produce different results.

In our story, the Groom brought several objects with him to impress the Bride's father. These items fall into two categories:

- **Gifts** that the Groom may give the Bride's father (figure 5.3).
- **Prizes** that the Groom collected during his life and can show the Bride's father to boast about his accomplishments (figure 5.4).

Figure 5.3: The models of the gift items in the final game: the Dress, the Box of Jewels and the Land Titles.

Both these types of items increase the Father's opinion of the Groom when gifted or shown. However, we devised other types of items that can help the Bride turn the tide to her favour:

- **Hiders** can be used to hide **Gifts** so that the Groom does not find them, and thus not gift them (figure 5.5).
- **Destroyers** can be used to destroy **Prizes** so that the Groom does not have a chance to show them (figure 5.6).
- **Tamperers** can be used with **Prizes** to alter them (figure 5.7). If the Bride tampers with a Prize and drops it at the same spot it was before for the Groom to pick up and show her father, the father will be negatively impressed with him. This is currently the only way the Bride has to decrease her father's opinion on the Groom.

Figure 5.4: The models of the prize items in the final game: the Crest, the Furs and the Sword.

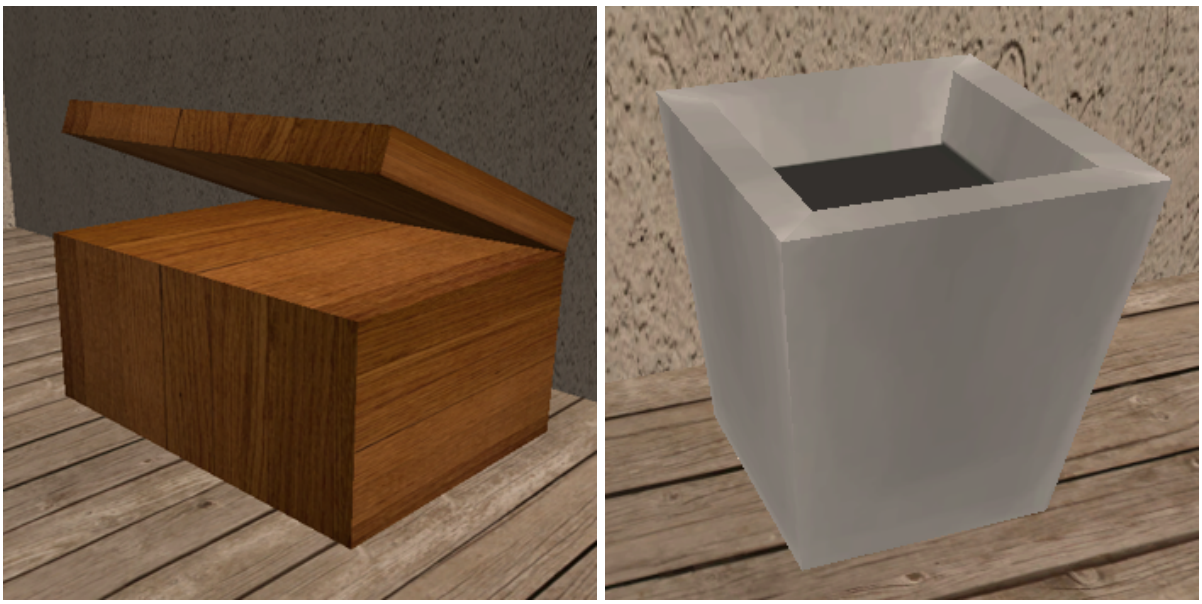


Figure 5.5: Some of the models of hiders in the final game: the Chest and the Trash.

By placing these items around the world, we can increase the complexity of both our story and characters. The interactions the characters have through these items serve as both a way to make them interact with the world in general, but also to further establish their relationships, described in section 5.1.1, even without having direct interactions between them. For example, by stealing, destroying and tampering with the Groom's belongings, it should become clear that the Bride's attitude towards the Groom is nothing short of negative.

5.2 The scene

In order to provide a relative familiar environment for the player, we decided to set our story inside the Bride's house, with decorative objects the player could easily associate with the real world. The house is a collection of rooms with various configurations that resemble common

Figure 5.6: Some of the models of destroyers in the final game: the Acid and the Fireplace.

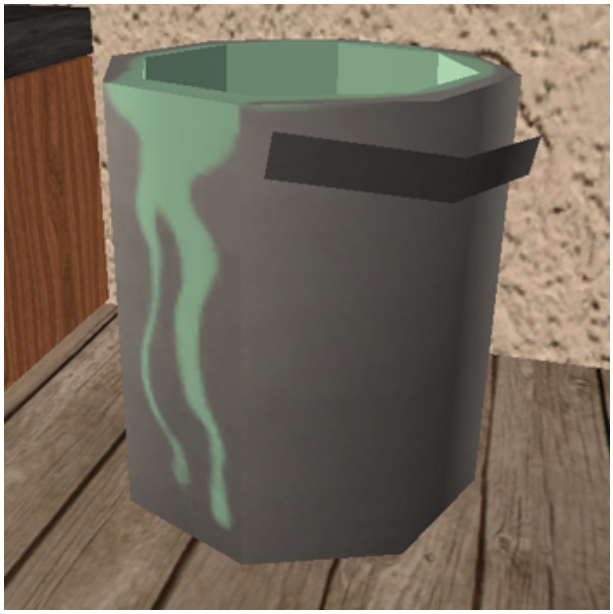


Figure 5.7: Model of the only Tamperer in the final game: the Ink.

household rooms, such as bedrooms, a study, dining room, workshop and dressing room.

All items and characters were carefully placed and split across all rooms in order to encourage exploration and mitigate conflicts that could arise from both characters approaching the same object at the same time. The final house can be partially seen in figure 5.8.

Figure 5.8: The House we built for our characters to interact in (seen fom a Godmode Camera).

Chapter 6

Evaluation

In this research, we set out to investigate the relation between the player and the character she portrays in various games and storytelling systems. We proposed that this connection can be made stronger or weaker depending on the type of camera that the game or system used to ground the player in the world. Going further, we hypothesized that a camera that sets a balance between world awareness and player immersion might help the player to better understand the story that unfolds in these types of systems and its events.

After making our system, as we presented in the previous chapter, and implementing the types of cameras introduced in our research and model, it was time to test the scenario.

We begin by describing the preliminary tests we conducted. These were done with a small set of people in order to test the players' reaction to each type of camera and also discover problems we did not during the development stage. After these initial tests, we corrected some problems pointed out by our testers and moved on to the bulk of the testing phase. We begin by describing the type of tests we did and the method for analysing the results to our tests. We then present those results and draw some conclusions from them.

6.1 Preliminary Tests

In order to evaluate our system in full, we needed to have a small set of users to test it before committing to a larger audience. These preliminary tests were done to discover and iron out the main problems that might arise when people play or watch the game that we were not aware at first.

Our preliminary tests consisted of letting people play the full game and asking them to comment as they played, while we collected their comments. Also, the game also collected logs that allowed us to later analyse what actions the player did, where they clicked, where they were and how long it took them to perform each action. The test was conducted on six people, ages ranging from 21 to 26 years old, all with some form of experience playing games similar to the ones we based our interactions out of.

6.1.1 Feedback and Changes

From the players' comments, as well as well as checking the logs we realized that our initial house was to little and had too few rooms, which clamped up the objects too much and did not allow for player exploration. Also, the items littered the floor and were hard to spot with cameras at eye level. To correct this, we ended up increasing the size of the house by four times the size of the original, created several new rooms and spread the items throughout them. Also, we created new decorative items to place interactable objects on top of, such as tables, desks and counters. In order to keep the new rooms from being too empty, we added new purely decorative items such as beds, bookcases and sofas.

We also realized that the main character's controls were inadequate. Based on feedback, we ended up fine-tuning much of the control attributes, such as speed of the character, ending up having two distinct speeds whether she's carrying an item or not. The Third Person Camera was the one most extensibility changed, pulling it a bit further from the character, allowing the player to see more around her, and also by increasing the side distance and allow the player to see more over the character's shoulder.

We also received some feedback based on the Camera angles and the player's understanding of what was going on in the game world. One of our Godlike Camera testers ended up confused since so many things were happening at once. This was less of a problem related to our system, and more of a drawback of the camera type itself, which bode well for our future tests.

6.2 Evaluation of the Camera Model

Making the corrections we deemed necessary after our first feedback loop, we decided to finally entail with our main batch of feedback to make our final evaluation and validate our main research hypothesis:

> *In a storytelling environment, having a camera too close or too far away to the main character is detrimental to understanding what is happening around her. By having the camera focus on the main character but also let the player see her immediate surroundings, the player will be able to place himself and connect better to what her character is experiencing.*

This means that we are placing our focus on the **Third Person Camera** which, as we analysed on section 2.6.1, has the best chance of achieving the balance between world awareness and player attachment to the character.

In this section, we begin by introducing the way we conducted our tests, analyse the answers and study the resulting comparison.

6.2.1 Evaluation Procedure

In this second evaluation, we decided to open up and broaden our audience. Since our focus was not so much the integrity of our controls or camera but rather peoples' perception of the world based on it, we decided to employ a combination of video and a questionnaire about it.

In the video, people can see a normal playthrough of the game from the perspective of the Bride, which goes about and interacts with several objects. We had three versions of the questionnaire, with three different videos based on the camera types we have developed. We made sure that the actions performed by the characters were fairly equivalent and that their results were visible during the video, so people could potentially understand their impact.

The questionnaire itself can be seen in Appendix A. It contained four main batches of questions that touched on the subjects of **what the Bride did**, **what the Groom did**, the **perceived relationship between the characters** and **general information**.

The first part's main goal was to test how easy it was to understand what the main character was doing. This is especially important because we want the player to understand when the character is doing something, and what, in order to surmise the consequences of her actions. In this part, we were hoping that the Third Person and Godlike cameras would provide better context, since the First Person camera is not clear when the character interacts with objects, as we've seen in section 2.4.

The second part was designed to test the ability of the person to infer or understand the consequences of the Bride's actions. Also, we wanted to test how easy it was to perceive what the other characters in the set were doing.

In the third part, we wanted to test how the characters' relations transpired depending on the player's vantage point, since we discussed that cameras closer to the character tended to bring the character and the world closer to the player. By testing the impact the actions during the game had on the relationships of the characters by the end, we were hoping that First and Third Person Perspectives would receive a more definite set of responses.

By general information, we wanted to capture a glimpse of the person's age, gender and gaming ability. With it we were hoping to understand their background and analyse their responses based on their previous experience if the results proved too disparate. People already playing games similar to the ones we developed should prove to have a better understanding of the game and its conventions than people with less experience.

6.2.2 Analyzing the Answers

After completing the three questionnaires, we published them online. People were randomly assigned to one of them, and we stopped taking answers after we reached the ninety people cap (thirty people for each camera type).

Most of the questions asked were regarding what was presented in the video. Mainly, we were testing what people could remember or retain after the video was over. Combining the number of events they remembered and the perceived relationships they took from the experience constituted our definition for understanding the story.

In parts one and two, we asked several questions that pertained to what had transpired in the video. These were absolute questions, which had a clear correct answer. When analysing the answer to these parts, we considered each correct and incorrect question. We calculated the percentage of correct responses for each person on each part.

As for the third part, we merely took the values that we were given and compared to our initial assessments. With this, we wanted to see if the feelings and relationships of the characters

were perceivable by the audience.

The fourth part was a mere statistical curiosity, and served to put our results into perspective. Like the first and second part, some questions were treated before being evaluated.

6.2.3 Results

We will now analyse the answers to our questionnaire. We will focus on each part separately where appropriate and examine it in detail.

First and Second Parts

The First and Second parts of our questionnaire dealt with what had happened in the world. We measured the people's understanding of what had transpired by calculating the percentage of correct answers. Our results are summarized in table 6.1.

		N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Min	Max
						Lower Bound	Upper Bound		
Correct Answers (%) (Bride)	First Person	30	.666667	.3632782	.0663252	.531016	.802317	.2222	1.0000
	Third Person	30	.911111	.1660910	.0303239	.849092	.973130	.4444	1.0000
	Godlike	30	.744444	.2562958	.0467930	.648742	.840147	.3333	1.0000
	Total	90	.774074	.2896705	.0305340	.713404	.834744	.2222	1.0000
Correct Answers (%) (Groom)	First Person	30	.694444	.2822438	.0515304	.589053	.799836	.3333	1.0000
	Third Person	30	.916667	.1313064	.0239732	.867636	.965697	.5000	1.0000
	Godlike	30	.702778	.2195905	.0400916	.620781	.784774	.4167	1.0000
	Total	90	.771296	.2407986	.0253824	.720862	.821731	.3333	1.0000

Table 6.1: Descriptives for the first and second parts of the questionnaire.

After determining the percentage of correct responses per person, we then conducted tests to see if this percentage significantly changed based on the type of test done. To do so, we used the Test of Homogeneity of Variances (Table 6.2) and the one-way ANOVA test (Table 6.3), since we had three main cases to test between. Note that all ANOVA tests performed in the present research had a Significance level of .05.

	Levene Statistic	df1	df2	Sig.
Correct Answers (%) (Bride)	34.394	2	87	.000
Correct Answers (%) (Groom)	22.775	2	87	.000

Table 6.2: Homogeneity of Variances table for the first and second parts of the questionnaire.

		Sum of Squares	df	Mean Square	F	Sig.
Correct Answers (%) (Bride)	Between Groups	.936	2	.468	6.232	.003
	Within Groups	6.532	87	.075		
	Total	7.468	89			
Correct Answers (%) (Groom)	Between Groups	.952	2	.476	9.840	.000
	Within Groups	4.209	87	.048		
	Total	5.161	89			

Table 6.3: ANOVA table for the first and second parts of the questionnaire.

In both cases, Levene's test indicates that the assumption of homogeneity of variance has been violated, ($F(2, 87) = 34.394, p < .001$ for the Bride and $F(2, 87) = 22.775, p < .001$ for

the Groom). Transforming the data did not rectify the problem and so F-tests are reported nevertheless.

In both cases, there is a statistically significant difference in the average number of correct responses between camera types (as evidenced by our Significances of .003 and $< .001$). But this is rather vague: we need a direct comparison between all three. In order to do that, we applied the Games-Howell post hoc test, since our assumption of homogeneity of variance has been violated. Table 6.4 shows the results of this test when applied to each part.

Dependent Variable	(I) Camera Type	(J) Camera Type	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
						Lower Bound	Upper Bound
Correct Answers (%) (Bride)	First Person	Third Person	−.2444444	.0729286	.005	−.421844	−.067045
		Godlike	−.0777778	.0811703	.606	−.273594	.118038
	Third Person	First Person	.2444444	.0729286	.005	.067045	.421844
		Godlike	.1666667	.0557595	.012	.031960	.301374
	Godlike	First Person	.0777778	.0811703	.606	−.118038	.273594
		Third Person	−.1666667	.0557595	.012	−.301374	−.031960
Correct Answers (%) (Groom)	First Person	Third Person	−.2222222	.0568340	.001	−.360424	−.084021
		Godlike	−.0083333	.0652895	.991	−.165624	.148957
	Third Person	First Person	.2222222	.0568340	.001	.084021	.360424
		Godlike	.2138889	.0467124	.000	.100869	.326909
	Godlike	First Person	.0083333	.0652895	.991	−.148957	.165624
		Third Person	−.2138889	.0467124	.000	−.326909	−.100869

Table 6.4: Games-Howell post hoc test table for the first and second parts of the questionnaire.

As the table shows, there was a statistically significant difference between groups when it comes to the percentage of correct responses in the first part of the questionnaire, as determined by one-way ANOVA ($F(2, 87) = 6.232, p = .003$). A Games-Howell post hoc test revealed that the percentage of correct responses was statistically significantly higher in the Third Person Camera opposed to the First Person Camera ($p = .005$) and the Godlike Camera ($p = .012$). There were no statistically significant differences between the First Person and Godlike Camera ($p = .606$).

Much like the first part of the questionnaire, there was a statistically significant difference between groups when it comes to the percentage of correct responses in the second part of the questionnaire, as determined by one-way ANOVA ($F(2, 87) = 9.840, p < .001$).Also like the first part of the questionnaire, a Games-Howell post hoc test revealed that the percentage of correct responses was statistically significantly higher in the Third Person Camera opposed to the First Person Camera ($p = .001$) and the Godlike Camera($p < .001$) There were no statistically significant differences between the First Person and Godlike Camera ($p = .991$).

Third Part

The Third part of the questionnaire dealt with how the characters related and felt towards one another. This was a more subjective measure, but one that we deemed important to analyse nonetheless. We evaluated how the interviewees perceived the relationship between characters by providing a scale from 1 to 5 (*Disagree Completely* to *Completely Agree*) on several questions about their behaviour and internal thoughts. The results are shown in table 6.5.

One thing of note is that the averages of these values pretty much correspond to the ideas we were trying to convey. The Bride did not like (2.13) nor help the Groom (1.76), due to not wanting this marriage (2.16). The Groom seems interested in her though (4.19), and wants

		N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Min	Max
						Lower Bound	Upper Bound		
Bride Helped Groom	First Person	30	1.83	1.117	.204	1.42	2.25	1	5
	Third Person	30	1.53	.819	.150	1.23	1.84	1	4
	Godlike	30	1.90	.885	.162	1.57	2.23	1	3
	Total	90	1.76	.952	.100	1.56	1.95	1	5
Groom Impressed Father	First Person	30	3.77	.626	.114	3.53	4.00	3	5
	Third Person	30	4.30	.466	.085	4.13	4.47	4	5
	Godlike	30	4.00	.455	.083	3.83	4.17	3	5
	Total	90	4.02	.560	.059	3.90	4.14	3	5
Bride Likes Groom	First Person	30	2.37	.809	.148	2.06	2.67	1	4
	Third Person	30	1.80	.847	.155	1.48	2.12	1	4
	Godlike	30	2.23	.898	.164	1.90	2.57	1	3
	Total	90	2.13	.877	.092	1.95	2.32	1	4
Groom Likes Bride	First Person	30	3.80	.961	.176	3.44	4.16	2	5
	Third Person	30	4.60	.621	.113	4.37	4.83	3	5
	Godlike	30	4.17	.699	.128	3.91	4.43	3	5
	Total	90	4.19	.833	.088	4.01	4.36	2	5
Father Likes Groom	First Person	30	3.67	.547	.100	3.46	3.87	3	5
	Third Person	30	4.17	.531	.097	3.97	4.36	3	5
	Godlike	30	3.73	.640	.117	3.49	3.97	2	5
	Total	90	3.86	.610	.064	3.73	3.98	2	5
Bride Wants Marriage	First Person	30	2.30	1.055	.193	1.91	2.69	1	4
	Third Person	30	1.73	1.112	.203	1.32	2.15	1	5
	Godlike	30	2.43	1.251	.228	1.97	2.90	1	4
	Total	90	2.16	1.170	.123	1.91	2.40	1	5
Groom Wants Marriage	First Person	30	4.23	.898	.164	3.90	4.57	3	5
	Third Person	30	4.80	.407	.074	4.65	4.95	4	5
	Godlike	30	4.40	.621	.113	4.17	4.63	3	5
	Total	90	4.48	.707	.074	4.33	4.63	3	5
Father Will Choose Groom	First Person	30	3.60	.814	.149	3.30	3.90	2	5
	Third Person	30	4.03	.669	.122	3.78	4.28	3	5
	Godlike	30	3.17	1.053	.192	2.77	3.56	1	5
	Total	90	3.60	.922	.097	3.41	3.79	1	5

Table 6.5: Descriptives for the third part of the questionnaire.

the marriage to happen (4.48). He got in the good graces of her father (3.86) and managed to impress him (4.02). As a result, he's likely to be chosen as his daughters fiancé (3.60).

After gathering these results, we needed to study them to determine if there was any significant change between camera types. Once again, we used both the test of Homogeneity of Variances (Table 6.6) and the one-way ANOVA test (Table 6.7).

	Levene Statistic	**df1**	**df2**	**Sig.**
Bride Helped Groom	1.309	2	87	.275
Groom Impressed Father	7.151	2	87	.001
Bride Likes Groom	1.125	2	87	.329
Groom Likes Bride	7.656	2	87	.001
Father Likes Groom	.754	2	87	.474
Bride Wants Marriage	1.928	2	87	.152
Groom Wants Marriage	23.703	2	87	.000
Father Will Choose Groom	9.816	2	87	.000

Table 6.6: Homogeneity of Variances table for the third part of the questionnaire.

Like the first two parts, in half of these questions' results, Levene's test indicates that the assumption of homogeneity of variance has been violated. Transforming the data did not rectify the problem and so F-tests are reported nevertheless.

As can be seen in the ANOVA test table, seven out of eight questions have significant

		Sum of Squares	df	Mean Square	F	Sig.
Bride Helped Groom	Between Groups	2.289	2	1.144	1.271	.286
	Within Groups	78.333	87	.900		
	Total	80.622	89			
Groom Impressed Father	Between Groups	4.289	2	2.144	7.883	.001
	Within Groups	23.667	87	.272		
	Total	27.956	89			
Bride Likes Groom	Between Groups	5.267	2	2.633	3.629	.031
	Within Groups	63.133	87	.726		
	Total	68.400	89			
Groom Likes Bride	Between Groups	9.622	2	4.811	8.024	.001
	Within Groups	52.167	87	.600		
	Total	61.789	89			
Father Likes Groom	Between Groups	4.422	2	2.211	6.703	.002
	Within Groups	28.700	87	.330		
	Total	33.122	89			
Bride Wants Marriage	Between Groups	8.289	2	4.144	3.176	.047
	Within Groups	113.533	87	1.305		
	Total	121.822	89			
Groom Wants Marriage	Between Groups	5.089	2	2.544	5.623	.005
	Within Groups	39.367	87	.452		
	Total	44.456	89			
Father Will Choose Groom	Between Groups	11.267	2	5.633	7.618	.001
	Within Groups	64.333	87	.739		
	Total	75.600	89			

Table 6.7: ANOVA table for the third part of the questionnaire.

differences to how people perceived the interactions between their characters and their feelings for one another based on the camera type.

For the questions that do not violate the assumption of homogeneity of variance, we applied an additional post hoc test that is more appropriate, the Tukey Test. Its results can be consulted in table 6.8, and their subsequent analysis in table 6.9.

To analyse the others, we once again recurred to a Games-Howell post hoc test. Table 6.10 shows the results of this test when applied to all parts. In order to avoid repetition, table 6.11 can be consulted to sumarize the analysis of these results.

These values clearly show us that the Third Person Camera stands out in most tests over the First Person Camera, even trumping the Godlike Camera in a few tests as well.

Fourth Part

The fourth part of the questionnaire was comprised of several general questions to help us analyse the background of our interviewees. It contained questions about gender, age and playing habits by game genre. These game genres helped us filter the types of cameras people are accustomed to playing with, based on the most common type of camera implemented in those types of games, and which can be seen on table 6.12.

After this, it was only a matter of counting the number of people in each category for each camera type. Figure 6.1 depicts the distribution of our population in the categories tested.

Here is what we learned about our population:

Dependent Variable	(I) Camera Type	(J) Camera Type	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Bride Helped Groom	First Person	Third Person	.300	.245	.442	−.28	.88
		Godlike	−.067	.245	.960	−.65	.52
	Third Person	First Person	−.300	.245	.442	−.88	.28
		Godlike	−.367	.245	.298	−.95	.22
	Godlike	First Person	.067	.245	.960	−.52	.65
		Third Person	.367	.245	.298	−.22	.95
Bride Likes Groom	First Person	Third Person	.567	.220	.031	.04	1.09
		Godlike	.133	.220	.817	−.39	.66
	Third Person	First Person	−.567	.220	.031	−1.09	−.04
		Godlike	−.433	.220	.126	−.96	.09
	Godlike	First Person	−.133	.220	.817	−.66	.39
		Third Person	.433	.220	.126	−.09	.96
Father Likes Groom	First Person	Third Person	−.500	.148	.003	−.85	−.15
		Godlike	−.067	.148	.895	−.42	.29
	Third Person	First Person	.500	.148	.003	.15	.85
		Godlike	.433	.148	.012	.08	.79
	Godlike	First Person	.067	.148	.895	−.29	.42
		Third Person	−.433	.148	.012	−.79	−.08
Bride Wants Marriage	First Person	Third Person	.567	.295	.139	−.14	1.27
		Godlike	−.133	.295	.894	−.84	.57
	Third Person	First Person	−.567	.295	.139	−1.27	.14
		Godlike	−.700	.295	.051	−1.40	.00
	Godlike	First Person	.133	.295	.894	−.57	.84
		Third Person	.700	.295	.051	.00	1.40

Table 6.8: Tukey post hoc test table for the relevant questions of the third part of the questionnaire.

Dependent Variable	Comparison Between Camera Type (I)	Camera Type(J)	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Bride Helped Groom	First Person	Third Person	.442	−.28	.88
		Godlike	.960	−.65	.52
	Third Person	Godlike	.298	−.95	.22
Bride Likes Groom	First Person	Third Person	.031	.04	1.09
		Godlike	.817	−.39	.66
	Third Person	Godlike	.126	−.96	.09
Father Likes Groom	First Person	Third Person	.003	−.85	−.15
		Godlike	.895	−.42	.29
	Third Person	Godlike	.012	.08	.79
Bride Wants Marriage	First Person	Third Person	.139	−.14	1.27
		Godlike	.894	−.84	.57
	Third Person	Godlike	.051	−1.40	.00

Table 6.9: Tukey post hoc test analysis for the third part of the questionnaire.

- They are closely tied together on gender, with males (58.9%) occupying a slightly larger portion than the females.
- They are young overall, with most people occupying the 18 to 25 years tier (78.9%).
- Most are avid gamers, playing at least once a week. The majority either play everyday (34.4%) or once every two days (22.2%).
- The majority played or plays games that usually feature the type of camera that they tested (78.9%).

The last point is especially important to note: using a type of camera regularly might be a good start to figure out how it works and helps in the creation of a more familiar environment,

Dependent Variable	(I) Camera Type	(J) Camera Type	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
						Lower Bound	Upper Bound
Bride Helped Groom	First Person	Third Person	.300	.253	.466	−.31	.91
		Godlike	−.067	.260	.964	−.69	.56
	Third Person	First Person	−.300	.253	.466	−.91	.31
		Godlike	−.367	.220	.227	−.90	.16
	Godlike	First Person	.067	.260	.964	−.56	.69
		Third Person	.367	.220	.227	−.16	.90
Groom Impressed Father	First Person	Third Person	−.533	.143	.001	−.88	−.19
		Godlike	−.233	.141	.233	−.57	.11
	Third Person	First Person	.533	.143	.001	.19	.88
		Godlike	.300	.119	.038	.01	.59
	Godlike	First Person	.233	.141	.233	−.11	.57
		Third Person	−.300	.119	.038	−.59	−.01
Bride Likes Groom	First Person	Third Person	.567	.214	.028	.05	1.08
		Godlike	.133	.221	.818	−.40	.66
	Third Person	First Person	−.567	.214	.028	−1.08	−.05
		Godlike	−.433	.225	.141	−.98	.11
	Godlike	First Person	−.133	.221	.818	−.66	.40
		Third Person	.433	.225	.141	−.11	.98
Groom Likes Bride	First Person	Third Person	−.800	.209	.001	−1.30	−.30
		Godlike	−.367	.217	.219	−.89	.16
	Third Person	First Person	.800	.209	.001	.30	1.30
		Godlike	.433	.171	.036	.02	.84
	Godlike	First Person	.367	.217	.219	−.16	.89
		Third Person	−.433	.171	.036	−.84	−.02
Father Likes Groom	First Person	Third Person	−.500	.139	.002	−.83	−.17
		Godlike	−.067	.154	.902	−.44	.30
	Third Person	First Person	.500	.139	.002	.17	.83
		Godlike	.433	.152	.016	.07	.80
	Godlike	First Person	.067	.154	.902	−.30	.44
		Third Person	−.433	.152	.016	−.80	−.07
Bride Wants Marriage	First Person	Third Person	.567	.280	.115	−.11	1.24
		Godlike	−.133	.299	.896	−.85	.59
	Third Person	First Person	−.567	.280	.115	−1.24	.11
		Godlike	−.700	.306	.065	−1.44	.04
	Godlike	First Person	.133	.299	.896	−.59	.85
		Third Person	.700	.306	.065	−.04	1.44
Groom Wants Marriage	First Person	Third Person	−.567	.180	.008	−1.00	−.13
		Godlike	−.167	.199	.682	−.65	.31
	Third Person	First Person	.567	.180	.008	.13	1.00
		Godlike	.400	.136	.013	.07	.73
	Godlike	First Person	.167	.199	.682	−.31	.65
		Third Person	−.400	.136	.013	−.73	−.07
Father Will Choose Groom	First Person	Third Person	−.433	.192	.071	−.90	.03
		Godlike	.433	.243	.185	−.15	1.02
	Third Person	First Person	.433	.192	.071	−.03	.90
		Godlike	.867	.228	.001	.32	1.42
	Godlike	First Person	−.433	.243	.185	−1.02	.15
		Third Person	−.867	.228	.001	−1.42	−.32

Table 6.10: Games-Howell post hoc test table for the third part of the questionnaire.

but ultimately it is not a instant defining factor to understand what is going on in the virtual world as events occur, as we proved in the previous section.

6.3 Conclusions

Measuring a person's story understanding can be quite a daunting task, mainly because there's no clear or assured way to evaluate such an abstract concept. On our research, we focused mainly on the three aspects that make character-driven interactive storytelling so compelling to their players: the main character, the supporting characters or Non-Playable Characters (NPCs) and the relationships between them. We tried to evaluate a person's understanding of the story by analysing what was retained after the story was over from the events and character interactions

Dependent Variable	Comparison Between Camera Type (I)	Camera Type(J)	Sig.	95% Confidence Interval Lower Bound	Upper Bound
Bride Helped Groom	First Person	Third Person	.466	−.31	.91
		Godlike	.964	−.69	.56
	Third Person	Godlike	.227	−.90	.16
Groom Impressed Father	First Person	Third Person	.001	−.88	−.19
		Godlike	.233	−.57	.11
	Third Person	Godlike	.038	.01	.59
Bride Likes Groom	First Person	Third Person	.028	.05	1.08
		Godlike	.818	−.40	.66
	Third Person	Godlike	.141	−.98	.11
Groom Likes Bride	First Person	Third Person	.001	−1.30	−.30
		Godlike	.219	−.89	.16
	Third Person	Godlike	.036	.02	.84
Father Likes Groom	First Person	Third Person	.002	−.83	−.17
		Godlike	.902	−.44	.30
	Third Person	Godlike	.016	.07	.80
Bride Wants Marriage	First Person	Third Person	.115	−.11	1.24
		Godlike	.896	−.85	.59
	Third Person	Godlike	.065	−1.44	.04
Groom Wants Marriage	First Person	Third Person	.008	−1.00	−.13
		Godlike	.682	−.65	.31
	Third Person	Godlike	.013	.07	.73
Father Will Choose Groom	First Person	Third Person	.071	−.90	.03
		Godlike	.185	−.15	1.02
	Third Person	Godlike	.001	.32	1.42

Table 6.11: Games-Howell post hoc test analysis for the third part of the questionnaire.

Game Genre	Common Camera Type(s)
Puzzle / Casual / Board	Godlike
Action / Adventure	Third Person
First Person Shooter	First Person
Role-Play	First Person, Third Person
Massive Multiplayer Role-Play	Third Person
Strategy	Godlike
Simulation	Godlike
Racing	First Person, Third Person
Other Sports (Soccer, Basketball, etc.)	Godlike

Table 6.12: Most common camera types based on a game's genre.

that occurred. This was ultimately done by asking the person what she remembered after seeing the story unfold before her eyes an her perceived character relations.

Most of our tests show that the Third Person Camera allows for a greater retention of transpired events and heightened insight on character relationships when compared to the First Person Camera. The line blurs a little more when compared to the Godlike Camera: the perception of events or character relationships does not have a discernible difference in most cases. Still, the Third Person Camera faired better against the Godlike Camera where character relations were concerned, proving that the Godlike Camera can distance the characters from the viewer more.

Regardless of the results' statistical significance, the Third Person Camera still gained the upper hand when considering the overall mean of the answers. All in all, the Third Person

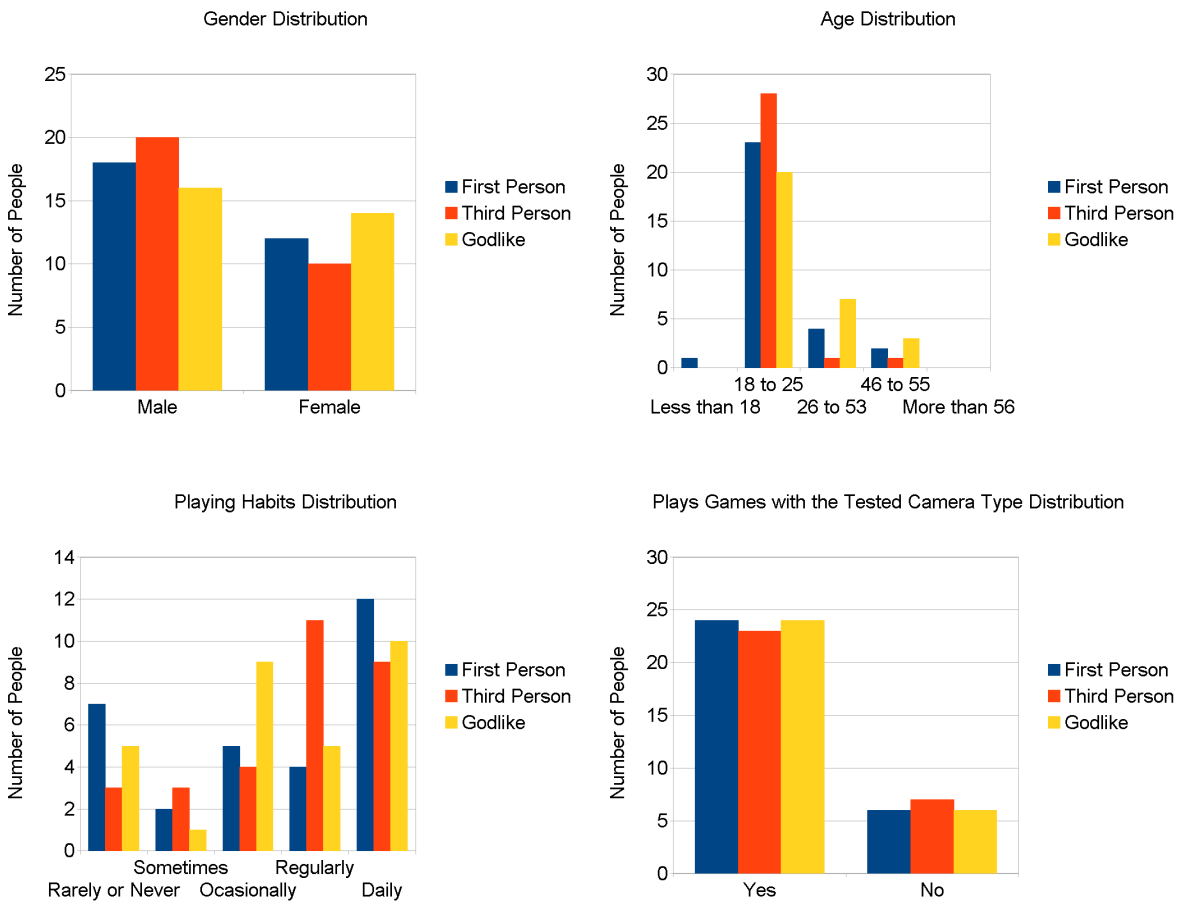


Figure 6.1: Statistics for the population that tested the First Person Camera

camera clearly shows the potential to overthrow the other two types when character-controlling character-based interactive storytelling systems are concerned.

This does not mean that people got very disparate takes on the characters, however: in the end everyone was able to find a common ground and share the same opinions overall about character relations. The Third Person Camera stood out merely for garnering more definitive answers.

Taking all of this into account, these tests are not meant to be taken as a definitive result. There are shortcomings with the type of test we did that could potentially skew the results. The first and most prominent is that we had no control over the number of times people could watch the video, even thought we discouraged multiple viewings. This could potentially mean that some of the people might have second guesses when it comes to what happened and the relationships between the characters.

Another problem is that we had no way to collect interviewees' thoughts during the test. We could have provided a comments box, but this would invariably either discourage them from writing in it or provide an overall opinion that would have little to no meaningful content.

This could potentially be solved by having interviewees actually test the prototype and having people supervise and collect information from the player while they play. Although this could be potentially a better type of test to conduct, it presents other challenges: we could never

attain the sheer number of tests we did in the same time period and, most importantly, people would invariably experience it at a different pace and commit to different plans, compromising the homogeneity of the tests. The last factor was what led us to conduct the tests we have showed. They allowed us to reach conclusions based on tests under the exact same conditions.

Despite all this, we still believe that the tests and the difference they showed are meaningful, and could be backed up with further and more diverse testing.

Chapter 7

Conclusion

With this work, we sought to study how the camera works with a game to provide the player with the information he needs to understand what is going on in the virtual world during play. We started by asking ourselves whether or not the camera actually impacts this exchange of information, and proposed that the camera should be far away from character enough that player sees her immediate surroundings, but also close enough to maintain a connection between them, as our research question showed:

> *In a storytelling environment, having a camera too close or too far away to the main character is detrimental to understanding what is happening around her. By having the camera focus on the main character but also let the player see her immediate surroundings, the player will be able to place himself and connect better to what her character is experiencing.*

This, we postulated, could be the key to making stories in virtual systems more approachable to the common person.

In order to pursue this research, we started by gathering related work from various sources and dealing with many topics. These would lay the foundations for our own system which we would develop.

We started by going over how cameras work in virtual settings in a general sense. This allowed us to put many of the current camera systems into perspective, by learning about the fundamentals that make or break a camera. It provided us with the basic knowledge to build these cameras in our own virtual environment later on. Also, there was a need to study how Interactive storytelling had progressed over the years, and how it was able to bring a world and its characters to life in order to tell a story. We focused especially on the relationship between story and characters in order to decide which path to take when developing our system. With this information, we were able to better analyse how today's cameras work in many and varied virtual systems. We studied many interactive stories and games to provide us with key information about choice of camera and design decisions that either influenced it or were made because of it. Finally, we delved into the relationship between camera and character control by analysing how different types of control could be mapped to certain camera types and vice versa, verifying the pros and cons of each approach.

After all this research, we started building the model for our interactive storytelling system based on the information we collected. We established it would be a character-based storytelling system, which would support the three main cameras used today in games and interactive storytelling systems: the First Person, Third Person and Godlike Cameras. We developed a 3D virtual world using the Unity3D Game Engine and populated it with characters and items that they could interact with. These interactions made the story flourish.

Done with its development, we finally tested our system by employing a combination of playtesting, used as a preliminary measure, and questionnaires, used as the main means of testing. Evaluating the responses to these questionnaires, we were able to verify our initial claim: indeed a camera like the Third Person can be beneficial to people's understanding of the world, its events and the relationships between the characters that live there, when compared to the other types of cameras. Although this difference was not statistically significant for all the parts of our test, we can still say with confidence that it was the camera which fared better in every tested aspect overall, from understanding the main character's actions to more accurately discerning characters' relationships and feelings towards others.

7.1 Future Work

Even though our results seem promising at first, many variables are still left to be tested. As examples, the following points can provide good starts for continued research that could potentially improve other areas related to gameplay cameras on interactive storytelling systems and games:

- When building our system, we had to scale down our initial idea of having a lot of characters on screen (somewhere around 5 to 7 characters to keep the system relatively stable and simple) to having just the main character plus the two supporting characters that were utterly essential to keep the story going forward. It would be interesting to test the values we proposed when more characters (and more items for them to interact with) join in, especially if those characters can also disrupt the other characters' plans as well.

- During development, and following ideas on our related work, we wanted to have a system where the player could choose the type of camera he wanted on the fly during gameplay. Our initial idea was to study how and when the players would change the type of the camera based on world events. Even though we had this functionality present in our system, we deemed it too complicated to test under such a short time frame. However, it would be interesting to figure out how players use the camera to their advantage and the reasons for changing to a certain type.

This kind of research would prove invaluable to developers of storytelling systems and games to provide better worlds and improve the interactions players need to perform in them. Also, having a set camera, or knowing the places where the probability players will switch to a certain type is big enough could lead to further improvements in level design, making a world's environments more fun to play and explore through and interactions more interesting.

Bibliography

[1] Ruth Aylett, Natalie Vannini, Elisabeth Andre, Ana Paiva, Sibylle Enz, and Lynne Hall. But that was in another country: agents and intercultural empathy. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, pages 329–336, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-6-1. URL `http://dl.acm.org/citation.cfm?id=1558013.1558058`.

[2] Byung-Chull Bae and R. Young. Suspense? surprise! or how to generate stories with surprise endings by exploiting the disparity of knowledge between a story's reader and its characters. In *Interactive Storytelling*. Springer Berlin / Heidelberg, 2009.

[3] Chris Baker. Review: Mirror's edge, the first-person parkour game you must play, November 2008. URL `http://www.wired.com/gamelife/2008/11/mirrors-edge-fi/`.

[4] William H. Bares and James C. Lester. Realtime generation of customized 3d animated explanations for knowledge-based learning environments. In *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, AAAI'97/IAAI'97, pages 347–354. AAAI Press, 1997. ISBN 0-262-51095-2. URL `http://dl.acm.org/citation.cfm?id=1867406.1867460`.

[5] F. P. Brooks. Grasping reality through illusion - interactive graphics serving science. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '88, pages 1–11, New York, NY, USA, 1988. ACM. ISBN 0-201-14237-6. doi: 10.1145/57167.57168. URL `http://doi.acm.org/10.1145/57167.57168`.

[6] Carlos Calderón, Nicholas Worley, and Karl Nyman. *Spatial Cinematic Mediation In Real-Time Architectural Walkthroughs*, volume 11, pages 343–360. ITcon, 2006.

[7] Marc Cavazza, Fred Charles, and Steven J Mead. Planning characters' behaviour in interactive storytelling. *The Journal of Visualization and Computer Animation*, 13:121–131, 2002.

[8] Marc Cavazza, Fred Charles, and Steven J. Mead. Emergent situations in interactive storytelling. In *Proceedings of the 2002 ACM symposium on Applied computing*, 2002. ISBN 1-58113-445-2.

[9] Marc Cavazza, Fred Charles, and Steven Mead. Intelligent virtual actors that plan ... to fail. In *Smart Graphics*. Springer Berlin / Heidelberg, 2003.

[10] F. Charles, M. Cavazza, and S.J. Mead. User intervention in virtual interactive storytelling. In *VRIC, Virtual Reality International Conference*, 2001.

[11] F. Charles, M. Cavazza, and S.J. Mead. Generating dynamic storylines through characters' interactions. *International Journal on Intelligent Games and Simulation*, 2002.

[12] Fred Charles and Marc Cavazza. Exploring the scalability of character-based storytelling. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, 2004. ISBN 1-58113-864-4.

[13] Fred Charles, Miguel Lozano, Steven J. Mead, Alicia Fornes Bisquerra, and Marc Cavazza. Planning formalisms and authoring in interactive storytelling. In *Proceedings of the 1 st International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 2003.

[14] J. Dias and Ana Paiva. Feeling and reasoning: a computational model for emotional characters. In *Proceedings of the 12th Portuguese conference on Progress in Artificial Intelligence*, 2005. ISBN 3-540-30737-0, 978-3-540-30737-2.

[15] J. Dias, I. Leite, M. Kriegel, M. Y.Lim, C. Martinho, S. Mascarenhas, A. Paiva, A. Pereira, R. Prada, and A. Wichert. Architecture development with action-selection mechanism. Unpublished Deliverable 5.3, 2010.

[16] Steven M. Drucker and David Zeltzer. Camdroid: a system for implementing intelligent camera control. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, I3D '95, pages 139–144, New York, NY, USA, 1995. ACM. ISBN 0-89791-736-7. doi: 10.1145/199404.199428. URL `http://doi.acm.org/10.1145/199404.199428`.

[17] Michael Gleicher and Andrew Witkin. Through-the-lens camera control. *SIGGRAPH Comput. Graph.*, 26(2):331–340, July 1992. ISSN 0097-8930. doi: 10.1145/142920.134088. URL `http://doi.acm.org/10.1145/142920.134088`.

[18] Li-wei He, Michael F. Cohen, and David H. Salesin. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 217–224, New York, NY, USA, 1996. ACM. ISBN 0-89791-746-4. doi: 10.1145/237170.237259. URL `http://doi.acm.org/10.1145/237170.237259`.

[19] Tom Hoggins. Mirror's edge review, December 2008. URL `http://www.telegraph.co.uk/technology/video-games/3690013/Mirrors-Edge-review.html`.

[20] S. Mascarenhas J. Dias and A. Paiva. Fatima modular: Towards an agent architecture with a generic appraisal framework. *Workshop in Standards in Emotion Modeling, Leiden, Netherlands*, 2011.

[21] Tanya Krzywinska. *Hands-On Horror*, pages 12–23. Harmony Wu, 2002.

[22] Jock D. Mackinlay, Stuart K. Card, and George G. Robertson. Rapid controlled movement through a virtual 3d workspace. *SIGGRAPH Comput. Graph.*, 24(4): 171–176, September 1990. ISSN 0097-8930. doi: 10.1145/97880.97898. URL `http://doi.acm.org/10.1145/97880.97898`.

[23] Bride Mallon. Towards a taxonomy of perceived agency in narrative game-play. *Comput. Entertain.*, 5(4):4:1–4:15, March 2008. ISSN 1544-3574. doi: 10.1145/1324198.1324202. URL `http://doi.acm.org/10.1145/1324198.1324202`.

[24] Michael Mateas and Andrew Stern. Integrating plot, character and natural language processing in the interactive drama façade, 2003.

[25] Michael Matheas and Andrew Stern. *Writing Façade: A Case Study in Procedural Authorship*, chapter II. Computational Fictions. MIT Press, 2007.

[26] David Milam and Magy Seif El Nasr. Design patterns to guide player movement in 3d games. In *Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games*, Sandbox '10, pages 37–42, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0097-1. doi: 10.1145/1836135.1836141. URL http://doi.acm.org/10.1145/1836135.1836141.

[27] Frank Nack. Event and story: an intricate relationship. In *Proceedings of the 2011 joint ACM workshop on Modeling and representing events*, J-MRE '11, pages 49–50, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0996-7. doi: 10.1145/2072508.2072520. URL http://doi.acm.org/10.1145/2072508.2072520.

[28] Ogechi Nnadi, Ute Fischer, Michael Boyce, and Michael Nitsche. Effect of dynamic camera control on spatial reasoning in 3d spaces. In *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, Sandbox '08, pages 157–162, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-173-6. doi: 10.1145/1401843.1401873. URL http://doi.acm.org/10.1145/1401843.1401873.

[29] Ana Paiva, Isabel Machado, and Rui Prada. Heroes, villains, magicians, ...: dramatis personae in a virtual story creation environment. In *Proceedings of the 6th international conference on Intelligent user interfaces*, 2001. ISBN 1-58113-325-1.

[30] Ana Paiva, Joao Dias, Daniel Sobral, Ruth Aylett, Sarah Woods, Lynne Hall, and Carsten Zoll. Learning by feeling: Evoking empathy with synthetic characters. *Applied Artificial Intelligence*, 19(3-4):235–266, 2005. doi: doi:10.1080/08839510590910165. URL http://www.ingentaconnect.com/content/tandf/uaai/2005/00000019/F0020003/art00003.

[31] Cary B. Phillips, Norman I. Badler, and John Granieri. Automatic viewing control for 3d direct manipulation. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, I3D '92, pages 71–74, New York, NY, USA, 1992. ACM. ISBN 0-89791-467-8. doi: 10.1145/147156.147167. URL http://doi.acm.org/10.1145/147156.147167.

[32] D. Pizzi and M. Cavazza. Affective storytelling based on characters' feelings. In *AAAI Fall Symposium on Intelligent Narrative Technologies*, 2007.

[33] Andrew Polaine. The flow principle in interactivity. In *Proceedings of the second Australasian conference on Interactive entertainment*, IE 2005, pages 151–158, Sydney, Australia, Australia, 2005. Creativity & Cognition Studios Press. ISBN 0-9751533-2-3. URL http://dl.acm.org/citation.cfm?id=1109180.1109204.

[34] Julie Porteous and Marc Cavazza. Controlling narrative generation with planning trajectories: The role of constraints. In *Interactive Storytelling*. Springer Berlin / Heidelberg, 2009.

[35] Julie Porteous, Marc Cavazza, and Fred Charles. Narrative generation through characters' point of view. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, 2010. ISBN 978-0-9826571-1-9.

[36] James Portnow. No redeeming value, August 2010. URL http://penny-arcade.com/patv/episode/no-redeeming-value.

[37] James Portnow. Cutscenes, September 2011. URL http://penny-arcade.com/patv/episode/cutscenes.

[38] James Portnow. Skyrim's opening, January 2012. URL http://penny-arcade.com/patv/episode/skyrims-opening.

[39] James Portnow. Kinect disconnect, April 2012. URL http://penny-arcade.com/patv/episode/kinect-disconnect.

[40] Vladimir Propp. *Morphology of the Folktale*. University of Texas Press, 1998.

[41] Mark Riedl and R. Michael Young. Character-focused narrative planning for execution in virtual worlds. In *Proceedings of the Second International Conference on Virtual Storytelling*, 2003.

[42] Mark Riedl, C. J. Saretto, and R. Michael Young. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, AAMAS '03, pages 741–748, New York, NY, USA, 2003. ACM. ISBN 1-58113-683-8. doi: 10.1145/860575.860694. URL `http://doi.acm.org/10.1145/860575.860694`.

[43] Mark O. Riedl and Andrew Stern. Believable agents and intelligent story adaptation for interactive storytelling. In *3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 2006.

[44] Mark O. Riedl and Andrew Stern. Failing believably: Toward drama management with autonomous actors in interactive narratives. In *3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 2006.

[45] Dave Shreiner. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison Wesley, 7th edition, 2010.

[46] Karl E. Steiner and Jay Tomkins. Narrative event adaptation in virtual environments. In *Proceedings of the 9th international conference on Intelligent user interfaces*, IUI '04, pages 46–53, New York, NY, USA, 2004. ACM. ISBN 1-58113-815-6. doi: 10.1145/964442.964453. URL `http://doi.acm.org/10.1145/964442.964453`.

[47] Bill Tomlinson, Bruce Blumberg, and Delphine Nain. Expressive autonomous cinematography for interactive virtual environments. In *Proceedings of the fourth international conference on Autonomous agents*, AGENTS '00, pages 317–324, New York, NY, USA, 2000. ACM. ISBN 1-58113-230-1. doi: 10.1145/336595.337513. URL `http://doi.acm.org/10.1145/336595.337513`.

[48] Kevin VanOrd. Mirror's edge review, November 2008. URL `http://www.gamespot.com/mirrors-edge/reviews/mirrors-edge-review-6200863/`.

[49] Jaroslav Švelch. What you can't see is what you don't get: paradigms of game world visualization. In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, Future Play '08, pages 212–215, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-218-4. doi: 10.1145/1496984.1497026. URL `http://doi.acm.org/10.1145/1496984.1497026`.

[50] Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. *SIGGRAPH Comput. Graph.*, 24(2):175–183, February 1990. ISSN 0097-8930. doi: 10.1145/91394.91442. URL `http://doi.acm.org/10.1145/91394.91442`.

[51] R. Michael Young. Notes on the use of plan structures in the creation of interactive plot. In *AAAI Fall Symposium on Narrative Intelligence*, 2000.

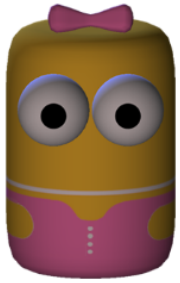
[52] R. Michael Young and Mark Riedl. Towards an architecture for intelligent control of narrative in interactive virtual worlds. In *Proceedings of the 8th international conference on Intelligent user interfaces*, IUI '03, pages 310–312, New York, NY, USA, 2003. ACM. ISBN 1-58113-586-6. doi: 10.1145/604045.604108. URL `http://doi.acm.org/10.1145/604045.604108`.

Appendix A

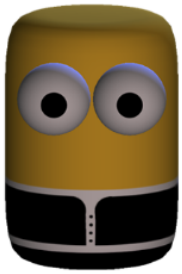
Experiment's Questionnaire

Conceal: Questionário

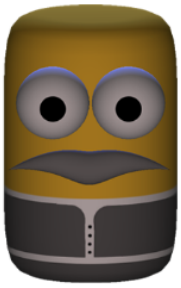
Bem vindo ao Mundo de Conceal. Neste mundo, vivem 3 personagens: Um pai (PAI DA NOIVA) que quer casar a filha (NOIVA). Um rapaz (NOIVO) de outra aldeia aceitou este desafio, mas isso significa que ele terá de o impressionar para ser o escolhido.



A Noiva



O Noivo



O Pai da Noiva

Para impressionar o PAI DA NOIVA, o rapaz trouxe vários presentes e prémios, para respetivamente oferecer e impressionar o pai da rapariga com os seus feitos.

Vestido

Jóias

Títulos de Terras

Símbolo da Acção "Dar"

O Vestido, as Jóias e os Títulos de Terras, que são Presentes que o NOIVO pode oferecer ao PAI DA NOIVA para o impressionar.

Brasão

Peles

Espada

Símbolo da Acção "Vangloriar"

O Brasão, as Peles e a Espada, que são prémios usados pelo NOIVO para se vangloriar ao PAI DA NOIVA.

À medida que cada um destes objectos é oferecido / mostrado, a opinião do PAI DA NOIVA sobre o NOIVO mudará.

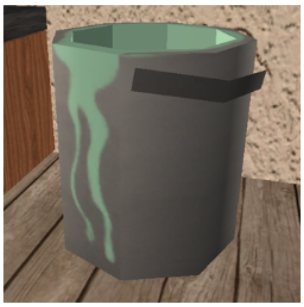
Existem ainda outras maneiras de interagir no mundo usando estes objectos:

Ácido

Lareira

Símbolo da Acção "Destruir"

O Ácido e a Lareira são destruidores que podem ser usados com prémios para os tirar do jogo permanentemente.



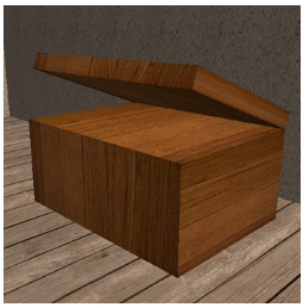
Tinta

Símbolo da Acção "Modificar"

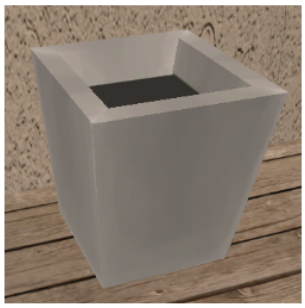
A Tinta é um modificador e pode ser usada com prémios para deturpar o seu propósito. Caso o NOIVO se vanglorie com um prémio alterado, em vez de impressionar o PAI DA NOIVA vai deixá-lo com uma pior impressão dele. Os prémios alterados não mudam de aparência. Apenas o seu ícone muda de cor para reflectir esta mudança:

Peles (Normais)

Peles (Modificadas)



Cofre



Lixo

Símbolo da Acção "Esconder"

O Cofre e o Lixo são esconderijos onde se podem guardar presentes para que o NOIVO não os encontre.

Nesta demonstração do jogo, o jogador encarna a personagem principal, a NOIVA. Na janela de jogo existem os seguintes elementos:

Tendo isto em consideração, veja o seguinte vídeo (tente não o ver mais do que uma vez) em que as personagens interagem com estes objectos e entre elas (aconselha-se a visualização em ecrã inteiro e em qualidade 720p). Preste especial atenção a estas interacções e depois responda às questões abaixo.

Parte 1: A Noiva

1.	O que fez a NOIVA às PELES? *	◯ Destruiu-as ◯ Modificou-as ◯ Escondeu-as ◯ Não Sei
2.	Caso as tenha respondido "Destruíu-as" ou "Modificou-as" na Questão 1: Com o quê?	◯ Ácido ◯ Martelo ◯ Lareira ◯ Tinta ◯ Não Sei
3.	Caso as tenha respondido "Escondeu-as" na Questão 1: Onde?	◯ Armário ◯ Cofre ◯ Cómoda ◯ Louceiro ◯ Lixo ◯ Não Sei
4.	O que fez a NOIVA ao VESTIDO? *	◯ Destruiu-o ◯ Modificou-o ◯ Escondeu-o ◯ Não Sei
5.	Caso as tenha respondido "Destruíu-o" ou "Modificou-o" na Questão 4: Com o quê?	◯ Ácido ◯ Martelo ◯ Lareira ◯ Tinta ◯ Não Sei
6.	Caso as tenha respondido "Escondeu-o" na Questão 4: Onde?	◯ Armário ◯ Cofre ◯ Cómoda ◯ Louceiro ◯ Lixo ◯ Não Sei
7.	O que fez a NOIVA à ESPADA? *	◯ Destruiu-a ◯ Modificou-a ◯ Escondeu-a ◯ Não Sei
8.	Caso as tenha respondido "Destruíu-a" ou "Modificou-a" na Questão 7: Com o quê?	◯ Ácido ◯ Martelo ◯ Lareira ◯ Tinta ◯ Não Sei
9.	Caso as tenha respondido "Escondeu-a" na Questão 7: Onde?	◯ Armário ◯ Cofre ◯ Cómoda ◯ Louceiro ◯ Lixo ◯ Não Sei

Parte 2: O Noivo

10.	O que fez o NOIVO ao ESCUDO? *	◯ Ofereceu-o ao PAI DA NOIVA ◯ Mostrou-o ao PAI DA NOIVA ◯ Nada, não o encontrou ◯ Não Sei
11.	Caso tenha respondido "Ofereceu-o" ou "Mostrou-o" na Questão 10: Qual a reacção do PAI DA NOIVA?	◯ Positiva ◯ Negativa ◯ Não sei
12.	O que fez o NOIVO às PELES? *	◯ Ofereceu-as ao PAI DA NOIVA ◯ Mostrou-as ao PAI DA NOIVA ◯ Nada, não as encontrou ◯ Não Sei
13.	Caso tenha respondido "Ofereceu-as" ou "Mostrou-as" na Questão 12: Qual a reacção do PAI DA NOIVA?	◯ Positiva ◯ Negativa ◯ Não sei
14.	O que fez o NOIVO ao VESTIDO? *	◯ Ofereceu-o ao PAI DA NOIVA ◯ Mostrou-o ao PAI DA NOIVA ◯ Nada, não o encontrou ◯ Não Sei
15.	Caso tenha respondido "Ofereceu-o" ou "Mostrou-o" na Questão 14: Qual a reacção do PAI DA NOIVA?	◯ Positiva ◯ Negativa ◯ Não sei
16.	O que fez o NOIVO às JÓIAS? *	◯ Ofereceu-as ao PAI DA NOIVA ◯ Mostrou-as ao PAI DA NOIVA ◯ Nada, não as encontrou ◯ Não Sei
17.	Caso tenha respondido "Ofereceu-as" ou "Mostrou-as" na Questão 16: Qual a reacção do PAI DA NOIVA?	◯ Positiva ◯ Negativa ◯ Não sei
18.	O que fez o NOIVO aos TÍTULOS DE TERRAS? *	◯ Ofereceu-os ao PAI DA NOIVA ◯ Mostrou-os ao PAI DA NOIVA ◯ Nada, não os encontrou ◯ Não Sei
19.	Caso tenha respondido "Ofereceu-as" ou "Mostrou-as" na Questão 18: Qual a reacção do PAI DA NOIVA?	◯ Positiva ◯ Negativa ◯ Não sei
20.	O que fez o NOIVO à ESPADA? *	◯ Ofereceu-a ao PAI DA NOIVA ◯ Mostrou-a ao PAI DA NOIVA ◯ Nada, não a encontrou ◯ Não Sei
21.	Caso tenha respondido "Ofereceu-a" ou "Mostrou-a" na Questão 20: Qual a reacção do PAI DA NOIVA?	◯ Positiva ◯ Negativa ◯ Não sei

Parte 3: Relações entre as Personagens

22.	A NOIVA ajudou o NOIVO. *	◯ Discordo Totalmente	◯ Discordo	◯ Não Concordo nem Discordo	◯ Concordo	◯ Concordo Totalmente
23.	O NOIVO conseguiu impressionar o PAI DA NOIVA. *	◯ Discordo Totalmente	◯ Discordo	◯ Não Concordo nem Discordo	◯ Concordo	◯ Concordo Totalmente
24.	A NOIVA gosta do NOIVO. *	◯ Discordo Totalmente	◯ Discordo	◯ Não Concordo nem Discordo	◯ Concordo	◯ Concordo Totalmente
25.	O NOIVO gosta da NOIVA. *	◯ Discordo Totalmente	◯ Discordo	◯ Não Concordo nem Discordo	◯ Concordo	◯ Concordo Totalmente
26.	O PAI DA NOIVA gosta do NOIVO. *	◯ Discordo Totalmente	◯ Discordo	◯ Não Concordo nem Discordo	◯ Concordo	◯ Concordo Totalmente
27.	A NOIVA quer que este casamento vá em frente. *	◯ Discordo Totalmente	◯ Discordo	◯ Não Concordo nem Discordo	◯ Concordo	◯ Concordo Totalmente
28.	O NOIVO quer que este casamento vá em frente. *	◯ Discordo Totalmente	◯ Discordo	◯ Não Concordo nem Discordo	◯ Concordo	◯ Concordo Totalmente
29.	O PAI DA NOIVA vai escolher o NOIVO para casar com a NOIVA. *	◯ Discordo Totalmente	◯ Discordo	◯ Não Concordo nem Discordo	◯ Concordo	◯ Concordo Totalmente

Parte 4: Perguntas Genéricas

30. Sexo: *	○ Feminino ○ Masculino
31. Idade: *	○ Menos de 18 ○ 18 a 25 ○ 26 a 35 ○ 36 a 45 ○ 46 a 55 ○ Mais de 55
32. Com que regularidade joga jogos? *	○ Raramente ou Nunca ○ Algumas vezes (1 vez a cada duas semanas) ○ Ocasionalmente (1 a 2 vezes por semana) ○ Regularmente (Dia sim, dia não) ○ Diariamente
33. Que tipos de jogos costuma jogar?	☐ Puzzle / Casual / Tabuleiro ☐ Acção / Aventura ☐ First Person Shooter ☐ Role-Play ☐ Massive Multiplayer Role-Play ☐ Estratégia ☐ Simulação ☐ Corridas ☐ Outros Desportos (Futebol, Basquet, etc.) ☐ Outros

* = Input is required

Submeter

This form was created at www.formdesk.com