Interactive Story Visualization

MARTIN BJÖRKMAN

KTH Computer Science and Communication

Master of Science Thesis
Stockholm, Sweden 2011
Interactive Story Visualization

MARTIN BJÖRKMAN

Master’s Thesis in Computer Science (30 ECTS credits) at the School of Computer Science and Engineering Royal Institute of Technology year 2011
Supervisor at CSC was Gustav Taxén
Examiner was Lars Kjelldahl

TRITA-CSC-E 2011:087
ISRN-KTH/CSC/E--11/087--SE
ISSN-1653-5715

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden
URL: www.kth.se/csc
Abstract

The purpose of this report is to map the current means of visualizing stories in video games. The focus lies on how the camera is used as a storytelling component.

All games that have been analyzed, have a strong storytelling component and are categorized based on a few common parameters. The result seems to indicate that there are a clear structure to how games in different genres are being visualized, but a study of much greater magnitude needs to be conducted before anything close to a defined structure can be presented as fact.
Referat

Visualisering av interaktiva berättelser

Syftet med denna rapport är att kartlägga hur man idag visualiserar berättelser i TV-spel. Fokuset ligger på hur kameras används för att förmedla berättelsen.

Alla spel, som har blivit analyserade, har en tydlig berättelsekomponent. Dessa har sedan blivit kategoriserade efter ett antal gemensamma parametrar. Resultatet verkar indikera att det finns en tydlig struktur i hur man visuellt skildrar berättelsen i de olika genrarna, men en mer utförlig studie behöver genomföras, innan man kan presentera detta som entydiga fakta.
Contents

1 Introduction
   1.1 Problem statement ......................................... 1
   1.2 The scope of the problem .................................. 2

I Execution

2 Background
   2.1 Interactive storytelling .................................... 6
   2.2 Remediation .................................................. 6
   2.3 Story visualization .......................................... 8

3 Process
   3.1 Data collection ............................................... 9
      3.1.1 Literature ............................................. 9
      3.1.2 Game analysis ......................................... 10
   3.2 Data interpretation ......................................... 10
   3.3 Data revision and composition ................................ 11

II Analysis

4 The concept of story and discourse ........................... 15

5 Discourse
   5.1 Perspective .................................................. 17
      5.1.1 2D ...................................................... 17
      5.1.2 2.5D .................................................... 19
      5.1.3 3D ...................................................... 20
   5.2 Camera type ................................................ 21
      5.2.1 First person .......................................... 21
      5.2.2 Third person ......................................... 22
      5.2.3 Cinematic ............................................ 26
   5.3 Tools ........................................................ 28
Chapter 1

Introduction

Stories and storytelling have always played a big part in the culture of our world. We use it to teach our young, to enrich our everyday life and to give meaning to a sometimes confusing existence. Storytelling has always been a medium for conveying hope, even in the darkest of times and even through the darkest of stories. It has been told in a spoken format, a painted format, a written format, a moving visual format and now in an interactive format. Through the years, the way of storytelling has changed, but the core has remained solid.

The purpose of this thesis, is to dive into the world of storytelling, while still focusing on the subject of interactive visual storytelling, i.e. video games, and to bring forth a taxonomy of how these stories are told and which tools are at our disposal. This is a subject that has been named, in the context of this thesis, Story visualization.

Story visualization is in its essence how, and with which tools, we choose to use to visualize and convey a story to the end user, in this case the player of the game. The way of visualization extends from the mere way of displaying important information, to how the camera is used to visualize components of the story and emphasize the game-play.

The main reason for writing this thesis is to emphasize the importances of Story visualization and to create a clearer picture of how it is being used today. Writers tell stories through the words that they choose, directors tell it through what they let their audience see and game designers tell it through what they let the player experience.

1.1 Problem statement

How is the story being visualized in video games today, which tools are available and what is specific for each genre?
CHAPTER 1. INTRODUCTION

1.2 The scope of the problem

The scope of the problem is to evaluate how story based video games are being visualized today, with a focus on general camera techniques, interaction and visualization. This does not include games without a strong storytelling component, such as simulators, sports and party games. It is only the single player sections of the games, that has been analyzed.
Part I

Execution
Chapter 2

Background

The medium of video games did not simply pop into existence, but rather evolved through earlier media. It was a remediation of many different media, among which some of the most prominent were books, movies and interactive storytelling. This remediation occurred over time, starting with interactive adventures in the text based format. One of these adventures were Zork (fig 2.1), where the player is experiencing an adventure through the text printed on the screen and through the commands entered on the keyboard. An example might be that the game tells the player “You are in a forest” to which the player responds “go north”. The game then moves the player north, if possible, and prints the description for the new area.

These text based games then got more and more advanced graphics and ways of interactions added to them. This then became two and three dimensional games, which in turn evolved over the years to the visually stimulating experience it is today.

The following sections will explain the concepts of: Interactive storytelling, Remediation and Story visualization. This is done to further enhance the understand-
ing of video games as a media, since these three components affect how the media has been used and how it is being used today. With these components in mind one can likely find aspects of games, which one has not noticed before.

### 2.1 Interactive storytelling

The concept of interactive storytelling is, as the names imply, a story told in an interactive fashion. It is a story experienced rather than a story told, since it allows the player to act out a part of the story, and through his or her actions affect the outcome. One might say that with the birth of interactive storytelling, the digital role-playing games were born and with them the story based video games.

Interactive storytelling have taken many shapes during the years, including rudimentary conversations with an AI, multi user dungeons\(^1\), multiform stories and the fantastic concept of the Holodeck [Mur97]. Each and everyone of these ideas has been an attempt to create a concept for a fully interactive story, where the user can interact almost limitlessly with the environment, within the boundaries of the story.

The multiform stories are stories told using a plot-line similar to a web, rather than a straight line. Each significant choice takes the user down a different path, affecting the order and outcome of the story [Mur97]. Asking a certain kind of question in a role-playing game, might cause a new “world” to unlock, which might not have been the case, if a different question had been asked. This does not necessarily mean that the new “world” is inaccessible for all time, that depends on the structure of the story and the significants of the question.

The Holodeck is the ultimate interactive story, where the player takes on the role of one of the characters in a completely immersive environment with full sensory input. The concept is an imaginary one, from the television series *Star Trek: Voyager*, but it serves well as a goal for interactive storytelling [Mur97].

Interactive storytelling is what really separates video games from other story driven media such as books, plays and movies. The interactive adds a new dimension, but also demands new methods and new ways of thinking.

### 2.2 Remediation

Remediation is in its essence “...the way in which, one medium is seen by our culture as reforming or improving upon another.” [JDB00]. An example of this might be the way the newspaper has merged the printed word and photography into a “new” media, which in turn is an improvement of its components. In the context of remediation there are no new media, since everything is based on prior media, the word “new” will however be used with this in mind.

The process of creating new media through the act of remediation, is not only an act of improvement, but also an act of recognition. We accept and can interact with

\(^1\)A text based multiplayer adventure.
new forms of media, since in their essence they are not really new. We recognize the presence of earlier media and we can through that recognition use the new, without much confusion. It is hard to imagine what a completely new media would look like, since every media we know of, has evolved from something that came before.

As new media evolves, there is a steady exchange of improvements between the new and the old, as they both struggle against the absorption of one another [JDB00].

The information communicated through the media, can be perceived by the recipient in two very different ways. It can either be seen as information presented by a medium, for example a newspaper, where the awareness of the medium itself is high, or it can be presented seamlessly through the medium, where the awareness is very low or nonexistent. The former is what is called *Hypermediacy* while the latter is called *Immediacy*.

**Hypermediacy**

Hypermediacy is, as mentioned above, when the awareness of that, which we perceive is communicated through a medium, is high. This is not sought after, when telling an interactive story through a medium, such as video games. An example of a medium with high hypermediacy might be the regular newspaper, where what is being depicted is not visualized in anything close to its true form. The text and photographs does not put the reader in the scene with the content.

**Immediacy**

In its essence immediacy means that the media erases itself and only leaves the viewer in the presence of the object represented [JDB00]. In other words it is a transparent media, the opposite of hypermediacy. This disappearance or transparency of the media, can be acquired either by really making the barrier between media and end user disappear, or by making the user believe that it has disappeared. This belief can be enhanced by adding interactive components to the medium, since that causes the user to feel actively involved in that, which is unfolding, or through the repeated use of the medium, to such an extent that we do not acknowledge its presence any more. As Douglas Adams once said [HH09]:

“We notice things that don’t work. We don’t notice things that do. We notice computers, we don’t notice pennies. We notice e-book readers, we don’t notice books.”

**Remediation in the scope of this thesis**

What needs to be realized is that even though video games are a remediation of television, film, novels and many other media, it is in itself a “new” media, that should not rely to heavily on the rules and regulations of others. The medium of video games needs to find its own identity, strengths and weaknesses. The fact that
it is an interactive media, separates it from its predecessors and makes it all but impossible to tell the story, using their exact rules and regulations.

2.3 Story visualization

Story visualization is, as mentioned in the introduction, a new term coined during this master thesis. It is meant to bridge the fields of general information visualization and interactive storytelling, told in a visual medium. The reason behind this “new” concept is to emphasize the importance of the act of visualization, both through what the camera depicts and through the information shown to the player (usually in the form of a Heads Up Display, presenting information in the form of text, images and meters/charts), as a storytelling component.

One might not think that how the camera system behaves during navigation affects the way the player perceives the story, but that is not the case. Every detail can be seen as a part of the story. The camera controls how we view the environment and how we connect to the character, be it in first person, third person, close to or far away from. To emphasize that importance is the purpose of Story visualization and this entire master thesis.
Chapter 3

Process

The method used during this master thesis to collect, analyze and categorize data, has been split into three different stages, namely data collection, data interpretation and data revision and composition. The first stage of data collection was performed to build a foundation for further evaluation.

3.1 Data collection

The data collection entailed gathering data from various different fields, related to storytelling, both in video games and in film. Those fields were Interactive storytelling and Remediation. A study of camera techniques in film and camera techniques in video games, was also conducted. This stage consists of a literature section and a game analysis section.

3.1.1 Literature

During the literature section of the data collection, books were read on the subjects of: Remediation, Interactive storytelling, Grammar of the film language and Real-time camera systems in video games. This was done to create a theoretical foundation on which the structure and taxonomy could be based. The books used were:

- *Real-Time Cameras - A guide for game designers and developers* by Mark Haigh-Hutchinson
- *Grammar of the film language* by Daniel Arijon
- *Hamlet on the Holodeck* by Janet H. Murray
- *Interaktiv manuskriptskrivning - Principer, begreber och redskaber* by Michael Valeur
- *Remediation* by Jay David Boltner, Richard Grusin
These books were selected based on their prominence in their respective field and a recommendation made by the staff at VIC Sthlm. Hutchinson’s book on the subject of camera design is the only real documentation on the subject that is published in printed form.

Remediation and Interactive storytelling were studied to get a proper background to the field of video games, why they are constructed the way they are and what has been inherited from previous media.

3.1.2 Game analysis

The game analysis section of the data collection, was performed to get a basic understanding of how the camera system actually works in video games, which later could be used to build a foundation for a possible breakdown and taxonomy. It was conducted with a few key aspects in mind, to get a fairly equal assessment of the otherwise quite different game mechanics. The aspects were as follows:

**Perspective** - which perspective is used to visualize the game?

**Control** - which camera controls are available to position and reorient the camera and will it reset upon movement?

**Effects** - are there any special effects present in the game and how are they connected to the game-play?

“**Cut”-scenes** - how is the story told through “cut”-scenes?

**Unique behavior** - is there any unique camera behavior present in the game that distinguishes it from every other game?

A few of the games analyzed were: *Mirror’s Edge* by EA DICE, *Little Big Planet* by SCE, *Mass Effect 2* by Bioware, *Starcraft 2* by Blizzard, *Killzone 2* by Guerrilla Games and *Metal Gear Solid 3: Snake Eater* by Konami (for a complete list of games used for evaluation see appendix A).

3.2 Data interpretation

The data interpretation stage, was used to assemble and analyze all the data, that was previously gathered. The data was divided into groups, based upon when and how the technique was used. This categorization was done, using a mind-map structure arranging the information in different levels, based on the subject of *Story visualization*.

The usage of the concept of Story and Discourse (explained in chapter 4), as a wrapper for the different parts of Story visualization, was first discovered during a discussion with Björn Thuresson, a teacher working at VIC-Stockholm, who has a background in film science. It was the last piece missing, in the basic structure of Story visualization.
3.3 Data revision and composition

The data revision and composition stage, was used to enhance the previously structured data. All the components of the Discourse section were finished and this stage was used to assemble the Story components, which are Character, Sequence and Genre. These components are all essential to the core of every story.

During this assembly it was discovered that a revision of the game analysis was needed, to touch upon subjects missed in the last sweep. The new attributes, based upon the sequence structure, that every game genre consists of, were:

**Navigation** - the way the camera is used to navigate the levels (including climbing, jumping and puzzle solving).

**Combat** - the way the camera behaves during hostile and violent situations.

**Dialogue** - how the interactive dialogue sequences work.

**Cinematic** - which kind of “cut”-scenes there are.

**Visualization** - the information displayed on screen.
Part II

Analysis
Chapter 4

The concept of story and discourse

The concept of story and discourse is a narrative structure inherited from ancestral media such as fiction and film. The concept is based on an observation that every narration consists of two different segments: story and discourse (figure 4.1). The story is defined by the events that occur (actions and happenings) and what might be called existents (characters, settings and objects). The discourse is how that story is communicated to the audience [Cha89], in this case the players of the game. In its essence the story is the core of the narrative and the discourse is the directors/authors/designers interpretation and communication of that core. If you change a component in the story, you change the narration, but if you change a component in the discourse you only change how the story is told.

![Figure 4.1: The building blocks of a narrative.](image)

In this thesis the concept of story and discourse will be used as a basis for categorizing the components of Story visualization. The story section consists of a genre specific analysis of how stories are told (i.e. which components define and bind the story) and through the use of which tools that is accomplished. The discourse section consists of a list of available tools for visual and interactive storytelling and how they are “defined”.

15
Chapter 5

Discourse

The discourse represents how the narrative is told and in this case stands for the tools available to tell a visual and interactive story.

This chapter consists of six sections: Perspective, Camera type, Motion, Transition, Visualization and Interaction. These sections describe and define different tools for Story visualization; some are more loosely connected to the actual game camera than others.

The reason for listing and explaining these tools lie in the fundamental understanding of how the camera system works and which resources are available for the storyteller. These tools are used to build the structure of each genre in the Story chapter and it is therefore important to have a basic understanding of their functionality.

5.1 Perspective

The perspective is the way the camera view of the game world is rendered. This can be done in three different ways namely in two dimensions (2D), two and a half dimensions (2.5D or fake 3D) or three dimensions (3D). Each perspective has its own functionality and is used in different situations. The following sections will present each perspective.

5.1.1 2D

The camera we call a two dimensional (2D) has its name from the projection method used by the rendering system, namely the parallel orthographic projection. Even though this camera presents the viewer with a two dimensional image, it still has the functionality of a regular 3D camera [HH09].

The two dimensional camera was forced on the developers of early video games due to hardware limitations, but there are still those who favour the technique thanks to its simplistic style and sometimes innovative game-play [HH09]. For
example it is very well suited for more simplistic games such as puzzles, flash games and mini-games in bigger three dimensional games.

![Orthographic Projections](image)

Figure 5.1: Typical orthographic projections.

The presentation style is what makes this camera quite unique since it displays a very flat image, which is shown in Figure 5.6. This can be done either from a side view or a top view.

To increase the feeling of depth and movement, in an otherwise flat projection, a moving background is used (otherwise known as a *scrolling* background). This can be done in a few different ways, namely:

- **Continuous** background scrolling is when the background continuously moves independent of the characters position. The character can move freely in the portion of the world, that is currently being displayed, but the camera is continuously moving forward, preventing the character from going back. The speed of the background often varies depending on the speed of the character, to prevent them from moving off-screen.

- **Character-relative** background scrolling is when the background is "locked" to the position of the character i.e. the background moves (reversed direction of the character) when the character moves. The character has got a fixed relative position on the display device, often a bit behind the center of the screen. This "lock" can be used both in the horizontal and the vertical direction of movement.

- **Directional lag** is when the camera no longer is following the exact movement of the character, but lags behind like a rubber band causing non instantaneous camera movement.

- **Burst** is when the camera position is fixed on a section of the game world and only moves when the character reaches an edge in order to reveal a new section of the world. This is often done by freezing the character input and moving the camera to it’s new position over a short period of time.
5.1. PERSPECTIVE

**Screen-relative** background scrolling is when the camera movement is locked to a border around the screen edge, resulting in movement once the character enters that particular section of the screen. The speed of the camera is locked to the speed of the character, preventing him/her to move outside the screen. Once the character no longer is in that part of the screen, the camera stops moving.

**Region based** is when the camera continuously moves between two designated triggers in the game world. Once the character is outside those triggers, the camera is fixed, until a new set of triggers is encountered.

5.1.2 2.5D

The difference between 2D and 2.5D is that even though almost all objects are rendered in an orthographic view, there are also objects rendered to simulate a three dimensional view. The number of objects rendered in this way may vary greatly from game to game. The camera motion is typically parallel to the orthographic projection lines to further increase the sensation of three dimensions [HH09].

The 2.5D presentation style sometimes still uses the parallax effect\(^1\) to further enhance the sensation of three dimensions. If so the camera is often tightly controlled [HH09].

One of the most frequently used parallel projection types is isometric projection (figure 5.2a) [HH09], which projects the world in a slightly angled fashion (the angle between all axes is an even 120 degrees). This gives the illusion of depth and is often combined with pre-rendered tiles and backgrounds.

The term 2.5D is also used to define a new type of game, where the environment is displayed in 3D, but the navigation is locked to two dimensions (figure 5.2b, typically along the x axis and y axis [Cro09].

(a) Starcraft by Blizzard Entertainment  (b) Trine by Frozenbyte

*Figure 5.2: Classic 2.5D and “new” 2.5D.*

---

\(^1\)The apparent displacement of an object viewed from two different lines of sight. The effect occurs when you view an object in front of you and alternate by closing your left and right eye (the object will seem to shift its position).
5.1.3 3D

Perspective presentation or 3D is the interpretation of how the human eye perceives the world, i.e. objects further back look smaller than the ones that are close. This is achieved through the use of vanishing points, which are points in space to which all lines of the objects converge. The resulting image created by those lines is then projection on a 2D surface, corresponding to the view of the players’ “eye” [HH09].

![Diagram of vanishing point](image)

Figure 5.3: A simple example drawn with a vanishing point.

This presentation style is used by most games on the market today, since it simulates how we perceive the world [HH09]. This in turn promotes immediacy and if done right, can gift the player with an increased sense of freedom.

The use of a perspective projection both has its benefits and drawbacks. The biggest benefit of using 3D cameras is that you get something that closely resembles how we perceive the world. You can also use the camera much more freely, similar to what can be done in film. The biggest drawback with 3D cameras is that you have to put more effort into the camera system, both in regard to collision and to which sections of the environment are most important to visualize.
5.2 Camera type

There are three different camera types: first person, third person and cinematic. The first and third person camera systems are connected to the players direct view of the game world and can usually be controlled. The cinematic camera system is used, by the developers, as a pure storytelling component and often, if not always, limits the players control of both character and camera.

5.2.1 First person

The first person perspective (FP), or camera, features a camera with a vantage point coinciding with that of the protagonist\(^2\). For that reason it is sometimes also called a point of view camera. First person cameras offer the closest experience in regard to player presence inside a game and can be extremely immersive [HH09].

There is an entire genre associated with this particular camera namely first person shooter (FPS) or first person adventure (FPA), when the use of weapons is a small part of the game-play. A few examples of the former are *Half Life 2* [Val04b], *Battlefield 2* [Dic05] and *Killzone 2* [Eur09] and of the latter *Mirror’s Edge*, *Myst* and *Minecraft*.

![Examples of the first person perspective. (a) FPS, (b) FPA](image)

(a) Killzone 2 by Guerrilla Games (ign.com)  (b) Mirror’s Edge by EA Dice (on-mirrors-edge.com)

The only attachment a player has to the protagonist in a first person game, is either through the visible body parts on the screen or through nearby objects, such as a helmet, HUD or the cockpit of a plane. The body parts are usually represented by the hands, forearms and (a lot of the time) some sort of weapon. These parts create the illusion of control over an actual body and not just a floating being. The scale and position of these items are designed to fit the view-port and not to be proportionate, in relation to the environment. This has to be done since these parts usually are not visible in our field of view, unless the head is tilted down. The

\(^2\)The protagonist is the main character, the one you are controlling, also known as the avatar.
position of the camera is usually not locked to the vertical position of the eyes of
the protagonist (on the 3D model) since that affect how the gamer perceives the
scale of the game world and the travelling speed of the character. [HH09]

The first person camera system comes with a few limitations in regard to how
the player can perceive the environment. The fact that the camera is so tightly
bound to the character, limits the players field of view, which puts a high demand
on the camera control system. This is the case since the player needs to aim the
camera to clearly see close objects out of regular view. This makes it difficult to
perform high precision actions (like jumps) without demanding camera adjustments
performed by the player.

### 5.2.2 Third person

The third person perspective (TP), or camera, features a camera with a vantage
point outside the character’s direct field of vision. The camera is typically positioned
slightly above and a bit behind the character. One could say that the earliest games
might be considered as third person perspective, even though they were presented
in 2D, since their view was “detached” from the character [HH09].

The third person perspective is not considered quite as immersive as the first
person perspective, since it is not considered natural to see oneself from a distance.
Even if this is the case it still is the most widely used perspective (outside the
realm of FPS) both in games and films. The fact that it is detached from the
character makes it perfect for those games, where you need to have a greater view
of - (in relation to the character) and improved interaction with - the environment.
It makes it easier to navigate complex levels and is therefore used in games, where
it is a key aspect of the game-play [HH09]. The detachment also makes it possible
to move the camera in ways that would not be practical or even possible in the real
world. An example of this is “the over the shoulder view”, utilized by almost all
third person shooter games.

This type of camera is not only practical but also esthetically pleasing, since
you get a clear picture of the character, both in regard to graphical detail and
animation.

Third person game cameras often allow the player to dynamically control the
angular position and distance of the camera, relative to the direction of motion.
These changes can either happen instantaneously (usually toggle-able) or be inter-
polated for a smooth motion (input control). This solution is almost always used
in all sorts of third person games..

In some situations it is important to add restrictions to the camera manipulation
to prevent breaching the barrier between the game world and the “outside”. This is
usually the case with cramped environments, such as tunnels and hallways, where
letting the camera pass through a wall (unintentionally) can completely break the
illusion of the game. A solution to this problem was presented in *Metal Gear Solid*,
where the entire perspective changed from third person to first person upon entering
confined spaces such as tunnels [Kon98].
5.2. CAMERA TYPE

Even though the third person camera is a great choice, it still has a few drawbacks. The main problems with the system are collision and occlusion. Since most games that use the TP camera has a relatively open world solution with dynamic navigation, it is hard to predict where the camera is going to be. It is therefore important to try to prevent any and all kinds of unintentional camera behaviors such as; passing through walls, getting stuck behind objects and intense shaking (due to camera interpolation). It is also important to try to prevent total occlusion of the game character. Even though it might present a desired effect for a short period of time, it might cause the player to feel disoriented or frustrated, if it happens at key segments of the game, such as combat.

There are four types of behavior, in regard to positioning, of the third person camera: Fixed, Behind, Character relative, Path and Free-flying. These will be explained in the following sections.

Fixed cameras

Fixed cameras are placed in (and locked to) key positions in the environment and can be used to visualize a variety of different scenarios. Some of the most common applications are: when there is limited space for camera movement; when a key object or objects need to be presented to the player or as CCTV cameras to create the illusion that you are being watched.

They can also be used to mimic the traits of a film-camera or the fourth wall of a play\(^3\), in which it is often only possible to view the scene from a particular angle. This is used in the game Heavy Rain [Dre10] by Quantic Dream. In Heavy Rain almost all scenes are presented in this fashion (fig 5.5) but you have the ability to toggle between two or three of these predefined views.

\[\text{Figure 5.5: Two different fixed camera views in Heavy Rain by Quantic Dream (ign.com)}\]

---

\(^3\)The fourth wall of a play is the “wall” represented by the audience. The other three walls are usually fixed objects on the stage.
CHAPTER 5. DISCOURSE

**Behind the character**

Behind the character is the most commonly used view in third person perspective games (and movies). This is done since the character usually walks, runs and jumps, which makes it suitable to put the camera behind and slightly above the direction of movement [HH09].

There are a few subsets to the classical “behind the character” view namely:

**Visible feet** entails when the camera is positioned a bit further back than usual, which results in the visibility of the characters feet and with them the full body animation. This can be practical when the feet are used to judge jumping or other character positioning such as balancing on a ledge or rope.

**Far distant** pulls the camera further back to present a wider view of the environment. This is usually used in vehicle games [HH09], as an establishing shot *Enslaved: Odyssey to the West* [The10] or to further enhance the feeling of open space *Shadow of the Colossus* [Ico06] (fig 5.7).

**Over the shoulder** is the most common camera presentation for third person shooter games (or games with aiming requirements). This entails moving the game character slightly to the side and positioning the camera right behind the character. This usually enables the player to see the upper part of the torso and the head of the character, while allowing for more accurate aiming.

![Figure 5.6: Behind the character camera positions](https://ign.com/a) inFamous by Sucker Punch Productions(b) Gears of War 2 by Epic Games (ign.com)

Character relative

Character relative cameras are usually used for replay cameras, side views and fender views, which are all used in racing and simulator games.
5.2. CAMERA TYPE

Path-based

Path based cameras uses a pre-determined path through the environment relative to the character [HH09]. This makes it possible for the level designer to decide where the camera should be positioned as well as what the camera should be looking at as the player navigates the level. This can be useful both in regard to preventing occlusion and collision and in regard to player focus.

Path based cameras are used in the God of War series, by Sony Computer Entertainment, where it removes the hassle of camera navigation entirely from the player (fig 5.8). This is an example of well executed planning and designing of both the levels and the camera. The correct angles are almost always depicted and the fighting flows very smoothly. The camera varies between different views when necessary. Top down, side views and perspective views are all being used depending on the situation and are switched through jump cuts.
Free-flying

Free-flying cameras are widely used in Real-Time Strategy games and allows the player to relocate the camera to any position inside the level. The camera is usually positioned in a top down fashion, giving a strategical overview of the battlefield, but can on occasion be moved into a far distant behind view [Ent10, Starcraft 2].

5.2.3 Cinematic

A cinematic camera is by definition (in this context) a camera view, in which the game world is displayed in a non-interactive fashion. It is used to tell a part of the story during which the player has no control what so ever over the unfolding of events. These cinematic events can either be pre-rendered and just replayed by the game or they can be rendered in real-time by utilizing the game engine. The former is most often used for playback of high quality films not rendered by the game engine, such as intro films. The latter is mostly used for pushing the story forward between segments of game play.

Cinematic “cut”-scenes

A cinematic “cut”-scene is (as mentioned above) a film segment of the game that has been pre-rendered (usually by a different rendering engine than the game engine) and then displayed in game. These sequences are basically real movie segments that can utilize all the effects and techniques of real movies. They are usually produced to tell parts of the story, that the player must see to understand it, or to set the mood for the rest of the game.

Cinematic “cut”-scenes are usually displayed in a different aspect ratio than the rest of the game to further signal to the player that they are non-interactive. These segments can usually be skipped by the press of a button if the player has seen them before.

Real-time “cut”-scenes

A real-time “cut”-scene is a cinematic sequence that is rendered in real-time by the game engine. These segments are often presented between different “chapters” of game play to tell more intricate parts of the story, which can not be told during continuous game play, and therefore take away most or all of the player’s control. The length of these sequences can vary from just a few seconds to a couple of minutes. In Assassin’s Creed Brotherhood they are used both to tell the story and to visualize reactions to player actions. If you for example pulls a switch to a crane a short sequence displaying the character being pulled skyward is presented during which no player input is processed. Since these sequences are so short the lack of control is barely noticeable [Ubi10].

To emphasize that a non-interactive sequence is being played the HUD (if there is one) is usually removed and restored upon completion.
Interactive “cut”-scenes

There are games that present the story based “cut”-scenes during continuous game play, which is not technically using cinematic cameras. This is often done to enhance the feeling that the events take place in the “real” world. These are scripted events that use the game engine for “playback”, which present the user and creators with a dilemma. If the user wants to get the complete story, he or she needs to “watch” these segments, which the creators probably want them to (otherwise why put them there?), and not just continue on. Since this is the case, the player has both the option of skipping them and is somewhat forced not to.

This is the case of *Half Life 2*, where most “cut”-scenes are interactive. They allow you to move on and leave the “cut”-scenes behind. When this is done you, on occasion, end up having to wait for the characters left behind, since they are needed to move forward [Val04b].

In *Assassin’s Creed Brotherhood* you encounter another interpretation of this method, where they use what you might call a faded replay visualization of events that occurred in the same place 500 years earlier. When you navigate the level you get ghostly glimpses of what happened the last time someone visited this area. They basically act as storytellers. Since the timing and execution of the ghost performance is of such a high quality, you can really take in the story as you flow through the level [Ubi10].
5.3 Tools

This section explains the basis of camera operation in the form of definitions and rules regarding camera movement, orientation and effects.

5.3.1 Movement

Motion in the form of general camera displacement usually occurs either through crab or dolly motion (fig 5.9), which are combined to create tracking.

Crab is when you move the camera sideways along with the camera axis (crab left or crab right), also known as strafing in gaming circles.

Dolly is the movement of the camera along the camera forward axis. Moving forward along the camera orientation is called dollying in and moving backwards is called dollying out.

Tracking refers to the motion of the camera relative to another object. Examples of tracking cameras can be a static surveillance like camera, that tracks the game character as it moves into frame or the third person camera, tracking the character continuously. The object, the camera tracks must not necessarily be the game character.

5.3.2 Rotation

Rotation is the angular displacement of the camera around the world axis. It can either be done separately for each axis by using yaw, pitch and tilt (fig 5.10) or through a combination of all of them.

Yaw is the horizontal orientation of the camera (also called Pan in classical cinematography). This is usually defined as the angular rotation around the world up axis, i.e. player turns the camera to look left or right.

Pitch, also know as tilt, is the rotation around the side axis i.e. player looks up or down.
5.3. TOOLS

**Roll** is the rotation around the forward axis. This is seldom used due to the fact that it can disorient the player. When used, it is to simulate vehicle movement, like the roll of an airplane or to emphasis small movements, like turning the head on its side when looking around a corner.

### 5.3.3 Effects

Effects are not only used by the film industry or in “cut”-scenes, but actually often are present during actual game-play. An example of this might be the slight camera bobbing that occurs in some games when the character is running or when objects blow up.

**Shaking/bobbing**

Refers to the irregular movement of the camera that can occur when an outside force acts on the mechanism holding the camera, such as vehicle motion, vibration or moving the camera by hand. This results in an uneven and shaky movement which can both be useful and nauseating (if used too much and/or with to high intensity). The effect is often used to simulate explosions, vehicles, hand-held cameras [(Ed10, Kane & Lynch 2] or head-bobbing from intense character movement [DIC08, Mirror’s Edge].

**Depth of field**

Refers to a phenomenon that occurs naturally in optical systems, but not in computer generated ones. This phenomenon puts different objects in focus depending on where we are looking. If we focus on an object in the center of the room, objects closer to and further away from us will be out of focus (i.e. details on those objects will be blurry). This “effect” has to be simulated using rendering effects in computer graphics, since virtual cameras have no lens. This effect is often used to direct the player’s attention towards a specific object [HH09].
CHAPTER 5. DISCOURSE

Motion blur

This is an effect that occurs in optical cameras, when an object moves faster than the exposure time of the film. The resulting image has the moving object on multiple locations, which causes the object to leave a blurry trace. This effect is often used to simulate fast player movement, which is the case in Mirror’s Edge where the edges of the screen are blurred, when running really fast [DIC08]. The problem with this effect is that it does not occur naturally in simulated systems and must therefore be added in by the rendering engine at a cost of extra computational time [HH09].

Lens flares

Lens flares are artifacts created, when the camera is directed towards intense light, like the setting sun, causing the film to overexpose and the iris to generate silhouettes of itself. They are usually unwanted in film, since they break the illusion by telling the viewers that this is filmed by a camera. Many games simulate this effect to try and add some realism to the game [HH09]. This can be quite misleading, since the presence of a camera promotes hypermediacy, unless it is a part of the actual game play and should therefore only be used in such cases.

Forced perspective

This is a camera technique used to alter the perception of relative size. It was frequently used in The Lord of the Rings (by Peter Jackson [Jac03]) to make the actors playing the Hobbits look smaller than they actually were [Jac03]. This is done by physically displacing the people or objects, without the viewers noticing, that they are not standing next to each other. This is not used very often in games, since we have the ability to scale models, which is not possible with actual people.

Zoom

The act of zooming can either be done optically by the camera or by moving the camera closer to the object of interest. The latter is the case used in video games, since there is no optical system present.

5.3.4 Rules

There are a few basic rules that should not be broken, when it comes to camera movement (interpolated or jumps), since it will cause unnecessary confusion for the player. Ever felt disoriented when the direction of movement is suddenly altered, when moving through a doorway in a video game? That is the result of breaking one of these rules.
5.3. TOOLS

30 degree rule

The 30 degree rule defines the minimum number of degree a camera must be moved between two cuts (showing the same frame) to 30 degrees. A change using a lesser amount than 30 degrees might cause the viewer to experience a shift in the objects position within the screen, rather than that of the camera [HH09]. This gives the effect that the entire scene jumps a tat during the transition. An example of a violation of this rule was presented in the film Rose Hobart (1963), see figure 5.11 [Cor36].

Figure 5.11: Violation of the 30 degree rule causes the character to jump sideways when the cut is made.

180 degree rule

The 180 degree rule (also known as the line of action) tells us that the imaginary line between two characters in a scene must not be crossed with a jump-cut (explained in section 5.4), since that will cause the characters to swap positions [HH09]. This is also applicable to moving objects, where switching between camera angles on different sides of the line, will switch the direction the object is moving in [Ari91].

Screen matching

Screen matching is the notion that characters need to be positioned on matching parts of the screen, when switching between different camera angles/cuts. If this is not the case, the player gets distracted and has to switch his or her point of focus, which can be annoying [Ari91].

The same notion applies when the character is in movement. The result of breaking that position on the screen during movement, does not only cause the player to have to shift focus, but also causes a perceived change of direction.
CHAPTER 5. DISCOURSE

This is used by Peter Jackson in the second *The Lord of the Rings* film to create the illusion of dialogue between two characters when in fact there is only one (in the scene where Sméagol talks to his alter ego Gollum) [Jac03].

Figure 5.12: 180 degree rule.

Figure 5.13: The dialogue between Gollum’s two alter egos.
5.4 Transition

Transition is the way the camera switches between two different angles, locations or points in time. It can for example be done by instantaneously switching to another view (Jump-cut), slowly fading to another view (Dissolve) or by showing multiple views before focusing on a new one (Viewport transition). The following sections will explain the basis of the most common ones.

Jump-cut

Jump-cuts are one of the most simple transitional techniques, both in games and in films, since it instantaneously switches from one point of view to another [HH09]. Everything in view can change within a jump-cut and is therefore often used to move through time [Ari91]. The pace of the cuts can also be used to increase the pace of the sequence. It is important not to break the 30 and 180 degree rules, when using jump-cuts, to not confuse the player/viewer.

Cut-away shot

Cut-away shots are used to show glimpses of events occurring on another location either at the same time as the current one or as a result of an action just performed [HH09]. This is often used to emphasize the reaction to a player action such as pulling a lever resulting in the opening of a door.

Insert shot

Insert shots are used to emphasize a certain element in the current scene. It can be used to get a closer look on a clue or the facial reaction of one of the characters [HH09].

Color fade

A color fade is a transition, during which the screen is faded to/from a solid. These are often used at the conclusion of a scene or to move between different scenes [HH09].

Point of view shot

Point of view shot is a shot, where the point of view is transferred to the first person perspective of one of the characters. This somewhat resembles the first person camera [HH09].
Crane shot
A crane shot is when the camera is detached from the view of the characters. It is often used to provide a better vantage point of the surrounding area and resembles the classic third person camera [HH09].

Reaction shot
A reaction shot is used to emphasize a character's reaction to an outside source, such as another character talking. It often precedes or follows a point of view shot [HH09] since jumping in and out of a point of view shot, can be confusing if the reason for it has not been shown (action demands reaction).

Reverse shot
A reverse shot is a shot, where the movement of the camera is reversed in relation to the previous shot [HH09].

Cross fade/dissolve
Cross fade is probably one of the most desirable transitions, since it gives a smooth transition between two scenes, by fading out the previous one on top of the new one. This is expensive due to the fact that two scenes need to be rendered on the same time. An alternative to the actual fading is to dissolve the old scene and thus revealing the new one (this can be done by saving a previously rendered image of the old scene) [HH09]. The old scene can be dissolved in any way you might like, an example might be to shatter it into small pieces that fly off screen.

Wipe(s)
A wipe literally wipes the current scene off the screen. This can either be done to wipe the screen to a solid color or to reveal a new scene [HH09]. The shape and direction of the wipe can take on any form; it can be a wipe to the side (like turning the page of a book); it can be a checker wipe that removes one or more segments of the screen at one time or it can take on the shape of a star, that expands from the center of the screen.

An elegant way of using the wipe transition is to show the character, walking in behind an object, that spans the full height of the screen and then swiping that scene in the reverse direction of movement, revealing the character entering the new scene as it comes out on the other side of the object.

Viewport transition
A viewport transition allows the player to see multiple events at the same time, by dividing the screen into different sections. This can be used to show the reactions of multiple characters to a singular event, even though they might be positioned off
screen in relation to each other. It can also be used to display multiple time-lines, unfolding alongside each other. After a short period of time, when the “message” has been presented to the player, one of the sections of the screen is chosen and presented in full-screen, either by expanding the view or by switching it instantaneously [HH09]. This is used in *Lost Odyssey* during certain “cut”-scenes that involve dialogue and reactions between multiple character [Stu08b].
5.5 Visualization

When telling an interactive story, it is not only important to make it visually pleasing, to the players but also make it functional and fun to play. This means that you sometimes have to present information to the player that could break the illusion of immediacy. An example of this could be the use of a heads up display to present the player with the current position of the character (in relation to the game world), on a mini-map or to show how much life the character has remaining.

There are multiple ways of displaying this kind of information to the player and the upcoming sections will present a few of them in regards to how they work, when they can be used and possible alternative ways of relaying the same information to the player.

5.5.1 Heads up display

A heads up display or HUD is a transparent overlay of information, presented on the screen to allow the user to focus on the task at hand, while still being able to see crucial “hidden” information. A HUD is a collection of on-screen elements, the purpose of which is to indicate player status. The information displayed greatly varies from game to game, but usually consists of information regarding player health status and/or location [Wil06].

HUD elements have been used for a very long time, in the video games industry, to present the player with information that “might” be necessary during game-play. With the evolution of the computer technology, a lot of these elements have stopped being useful, since they can be visualized through purely graphical components. More and more HUD elements are removed with the release of new titles and new innovative ideas. *Mirror’s Edge*, is one example where the HUD has been “completely” removed. The only thing that is left is the dot on the center of the screen, to simplify the navigation [DIC08].

With the release of *Killzone 2*, Guerrilla games made an update to their HUD, which no longer displays a player’s health level (fig 5.14). This has instead been replaced by a visual cue, in the form of color fading. When a player has full health the colors on the screen are vibrant, just as in real life, but when the character takes damage, the colors fade to black and white [Eur09]. The color level depends on the amount of life the player has left. This is a less precise way of visualizing health, but it feels like a more natural one.

If still it is necessary to incorporate a HUD into a game, one can either make the best of it and try to make the distinction between the HUD and the rest of the game as small as possible, while still making the important information visible, or one can “cheat” and introduce a reason for the existence of a HUD in the story, as was made in *Halo Reach* and *Enslaved*, where the HUD is explained by a futuristic helmet and a headband containing a neural interface [Stu10],[The10].
5.5. VISUALIZATION

Figure 5.14: Almost HUD-less action in Guerrilla Games Killzone 2 (ign.com).

5.5.2 Mini-map

The mini-map is a part of the HUD and is usually positioned in a corner of the screen, where it does not block important aspects of the view. It typically presents the player with a more or less detailed version of the surrounding area of the current level. The mini-map can be used to present a variety of information to the player, such as locations of quests, key objects and enemies as well as how the level can be navigated.

There are two possibilities used to prevent the player from obtaining too much information about the level, especially places he or she not yet has visited. The first one is to use fog of war, which essentially hides information from the player in these areas. This can either be done by completely blocking it out or by showing the level structure, but omitting all other information. The second one is to present the player with only a segment of the map, usually a small radius around the player position. This alternative is often used in conjunction with a bigger map, displaying the entire level, with or without fog of war, usually presented over the entire screen.

The first one is generally used in RTS games, where an overview of the entire map is important. The second one is used in a variety of games (for example action, adventure and role playing games), where displaying the above mentioned information is important.

An alternative to the mini-map (in more linear games) might be the establishing shot. This is common practice in film and television to present the viewer with a context and environment for the current scene. This is done by using a distant shot to show the spacial relations between important objects and characters in the scene [MP02]. In video games, where this is used, they usually present the player with an overview of the new level or sequence. This is done by showing the player the general structure of the level and where it ends. Key reference objects (like buildings or statues) are often highlighted in the shot, to present the player with a reference point, while navigating the level (Enslaved).
The mini-map is not just a component made for visualization, but also for storytelling. The choice whether or not to use a mini-map, tells the player about the characters knowledge of the game world. The level of detail, can be used to indicate that level of knowledge.

In a game like *inFamous* the presence of the map is explained by the fact that the main character Cole, is a bicycle messenger and urban explorer and therefore knows the city quite well. The dark areas of the map use fog of war, to tell us that they have been conquered by the enemy and is now considered as unfamiliar and unexplored territory [(SC09)].

### 5.5.3 Graphical user interface

A graphical user interface (GUI) is a common element in role playing and real time strategy games, other than just the information displayed on a HUD and/or on mini-map. The GUI consists of a variety of buttons and extra information that makes game-play a lot easier, since one usually has more abilities than is comfortable of possible to map to key/button-combinations. These extra elements can often be hidden by the player or even customized, which is the case in *World of Warcraft* [Bli10].
5.6 Interaction

There are three kinds of interaction in video games, according to Michael Valeur, the local interaction, the interaction that comes from making decisions and the hidden interaction.

The local interaction is the most common one, we perform an action and get an immediate reaction. This occurs on all player input, be it navigation or combat.

The decision based interaction takes place when we make a decision, which directly or indirectly affects the outcome of the story. The response is not immediate, instead it can take quite a while, before we see the result of our actions. An example of this might be, when you have to choose whether or not to save the life of a villain, rather than letting him die (in *Mass Effect 2*), which has different effect later on in the game [Bio10].

The hidden interaction takes place in the background, as the system gathers data as you play, evaluating your performance and adapting the difficulty to match it [Val04a].

5.6.1 Controllers

The controller is a key aspect of the interaction between the game and the player. If the controller and the key-mapping of that controller is comfortable and easy to use, the physical object "vanishes" and becomes an extension of the player inside the virtual world. The controller used, does not only affect in which physical way the game is played, but also how the game can be played. This affects and puts restrictions on the camera system.

Alongside the evolution of the video games themselves, lies the evolution of the input devices. Through the years these devices have taken a lot of different shapes and sizes and have suited different types of game play. Today the most commonly used input devices for "regular" games are game-pads and the keyboard and mouse combo. Alongside these regular devices are the motion controllers, which allow a different kind of user input, and touch screens.
Chapter 6

Story

The story consists of the core elements, that define it. A removal or drastic change in one of these elements, might change the story entirely.

This chapter consists of three sections, that in turn defines and explains the building blocks of the story: Character, Sequence and Genre.

6.1 Character

“We watch a film, and find ourselves becoming attached to a particular character or characters, on the basis of values of qualities roughly congruent with those we possess, or those that we wish to possess, and experience vicariously the emotional experience of the character: we identify with the character.”

- Murray Smith, Engaging Characters

One of the most important things in a video game, as in film, is that you relate in some way to the protagonist, just as the quote suggests. You need to feel some sort of bond to achieve a certain level of immersion. This is not only done through graphics, sound and story, but also through how the protagonist moves and interacts with the environment or more importantly, how the player perceives the protagonist’s movements and interaction’s.

6.1.1 Character specific camera considerations

There are a few things to take into consideration when designing a camera system, that affects how the character is perceived namely:

The character type defines how the player behaves and moves, which defines where the camera should be placed and how it reacts to the environment. A slow moving character has a different setup, than a fast paced flying character [HH09].
The camera needs to behave in such a way as to both improve the field of view and to enhance the impression of that particular type of movement.

The abilities of a character also defines, which camera modes that need to be present in the game. A character that utilizes weapons or ranged powers, needs to have some form of aiming mode and a close combat character needs to have an action camera, that can handle fighting enemies in various ways and locations.

Interaction with other characters or objects in the environment, might demand different camera setups for improved visualization and realism. There might for example need to be a special camera setup for dialogue.

The evolution of the protagonist during the length of the game (if there are any) might affect the camera system. There might be new or improved abilities, that give the character flight, vehicles or new weapons that demand new ways of handling the camera [HH09].
6.2 Sequence

Every story is divided into different sequences, depending on what is being told during each specific sequence. Be it dialogue between characters, environmental descriptions, character presentation or epic battles. The same theory applies to video games, where sequences are used to decide the design of the level and its pacing.

In the context of this thesis a sequence can be defined as one of the following: Cinematic, Combat, Dialogue and Navigation. Each sequence entails different interactivity, story structure and camera functionality. The following sections will define and describe each sequence.

6.2.1 Cinematic

The cinematic sequence is defined, in this context, as a completely story driving sequence. It is a section of the game that clearly tells a part of the story. It can come in many shapes and sizes, including everything from completely interactive “cut”-scenes to full blown pre-rendered animations.

Cinematic sequences are usually used throughout the entire game to drive the story forward through events and animations, that are too complex to execute through regular AI interaction. There are three main categories of cinematic sub-sequences namely: interactive “cut”-scenes, regular “cut”-scenes and pre-rendered “cut”-scenes.

Interactive “cut”-scenes are sequences where the story is being told, but you still have full control of your characters movements and actions. These sections do not usually demand that the player remains in the vicinity of the storytelling event, even though it might improve the players comprehension of the story. An example might be taken from the intro section of Killzone 2, where two of the main characters discuss the upcoming assualt of the enemy base [Eur09].

Regular “cut”-scenes are sequences, where all control of the character is lost and the game engine renders scripted events to tell the story. The level of complexity for these sequences can vary from something very simple, such as depicting the opening of a door as a result of pulling a lever to dialogue and combat sections, that go on for minutes. An indication of such a sequence is the fading of the HUD, if there is one, or the change in aspect ratio.

Pre-rendered “cut”-scenes are sequences, that has been rendered and composited beforehand using render farms. Since these sequences are so expensive to make they are usually only used to tell very important sections of the story, such as the intro and ending.
6.2.2 Combat

The combat sequence is defined, in this context, as a segment of game-play, where the player uses hostile abilities to eliminate enemies. This sequence demands a certain level of “physical” violence and does not include, when Mario jumps on or fires at turtles.

Combat can be conducted in a multitude of ways, not all mentioned in this report. The most common method is through direct player input, where the actions are carried out by character abilities. This is used in games like Bayonetta, God of War, Killzone 2, Mass Effect and Starcraft 2. A less common method is used in Heavy Rain, where the combat is conducted solely through quick-time events (the player needs to quickly press the button that appears on screen). Even though combat comes in many different forms the main focus still lies in the level of violence present, that is what defines this sequence.

The way combat is conducted, affects how the camera system is implemented and what kind of functionality is required, to give the player a fluent experience. Games containing range weapons, demand some form of aiming system and close combat, requires target selection. Both usable when engaging varying degrees of enemy numbers and locations.

6.2.3 Dialogue

The dialogue section is defined, in this context, as a segment of game-play, where an interactive dialogue between the player and some form of scripted non player character (NPC), is being conducted. The dialogue is usually conducted through an interface, where the player is able to choose his or her response from a pre-defined list of alternatives. This exchange is made in the form of a semi-interactive “cut”-scene, where the player only has the ability to choose responses.

![Figure 6.1: Dialogue in Bioware’s Dragon Age.](image-url)
6.2. SEQUENCE

Dialogue sequences are almost only used in role playing games, where choices and comments have the most impact on the story being told. Other dialogues, taking place between characters without any form of interaction, are defined as a part of the cinematic sequence and will not be presented in the dialogue sections of each genre.

6.2.4 Navigation

The navigation sequence is defined, in this context, as the segment of game-play, where the player traverses the level. It entails everything from walking, running, climbing to puzzle solving. It is the only sequence, that has a clear presence in all games.

The navigation usually defines the majority of the camera decisions, since the majority of the player interaction is done through navigation. The navigation also affects how the game world and the protagonist are perceived, since we “live” through the navigation.

The genre with the strongest navigation sequences is the platform/puzzle genre, in which it is the core game-play mechanic.
6.3 Genre

A genre in this context is a way to categorize games, based on their game-play mechanics, which in turn convey a certain kind of story. This is done to try to map the most common ways of telling a certain story in a certain kind of game. A categorization must be made, if a comparison between games should be possible.

The genre is defined, as mentioned earlier, by the game-play mechanics that put it aside from other games. An example might be platform games, that focus more strongly on navigation and jumping, than any other game. Another might be role-playing games, which focus strongly on character development and involvement.

With this as a foundation for categorizing games, six genres with a strong storytelling component were found. The categories are Shooter, Action/adventure, Stealth, Platform, Role-playing and Real-time strategy. The First person shooter genre has been merged with the third person shooter games into a new genre simply called Shooter. There are no real game mechanical aspects that separate them from third person shooters, other than the perspective used to visualize the story, which is not basis enough, in this context, for a separate genre.

As mentioned in the method description, each sequence has been structured based upon five parameters namely: camera position, orientation, transition, control and information visualization.

The conclusions and assumptions made, during the definition and composition of each genre, are based upon the games analyzed and should not be considered to be the complete and utter truth. A few genres have gotten more attention than others, due to availability of games and limitations in time. The ones that has gotten the least focus are: Stealth and Real-time strategy, which might result in a less comprehensive structure and analysis. There will be games that do not fit the description entirely and there will be those that fall outside the parameters of the structure outlined in this report.
6.3. GENRE

6.3.1 Shooter

Shooters are defined, in the context of this thesis, as games with a very strong weapon based action component. The majority of the game-play should be spent in combat, aiming or firing some sort of weapon. The weapons of choice are usually of long range, even though close combat on occasion is an alternative when stealth is key. Shooter games usually feature some sort of war scenario, where the player takes on the role of one (or a few) of the combatants. The mission objectives are usually pretty clear; get in, kill the enemy, get out and through your actions save the day.

A few examples of shooters include: *Bioshock*, *Gears of War 2*, *Half Life 2*, *Halo Reach*, *Killzone 2* and *Resident Evil 4*.

![Shooter genre structure](image)

Figure 6.2: Shooter genre structure (For a full view image see appendix B.1).

Sequences

Even though there can be a pretty strong story component in this genre, the presence of real dialogue sequences are quite unusual. A variation of the dialogues usually takes place as other forms of communication, either through radio or in person, between fellow soldiers or with the headquarters (HQ).
Navigation

Navigation and combat in shooter games is quite closely linked, since you navigate with the weapon at the ready. In first person games this is done in what would be called aim-mode in action/adventure games (weapon at the ready and a cross-hair on the center of the screen), while in third person games you still have to enter aim-mode.

First person

During regular navigation in first person shooter games, the camera is locked to what we perceive to be the head of the character, that we are controlling. We are usually able to orient it in the same way a human head can, but what is shown to us, as we tilt the camera down, differs from game to game. Some game allows us to view the bottom half of the character’s body, as one would in real life, while others simply show the ground, where the legs and feet ought to be.

The acts of running/sprinting or jumping does not affect the camera orientation as much as in third person games. Effects are instead used to distinguish these special moves from regular navigation. Running or sprinting will often cause the camera and/or weapon bobbing to increase, and a mild range of motion blur to be present at the edges of the screen [Stu10, Halo Reach], [Eur09, Killzon 2]. Jumping will most likely only jostle the camera upon landing if anything, but there are exceptions. In Killzone 2 the act of jumping is demanding, since the character gathers strength before the jump and hurls forward, as if jumping with both feet at once. The camera tilts down just before the jump is executed, then up and then down again, before it sets into default position.

As we look through the eyes of the protagonist we see different things in different games. Some games put the protagonist in a high-tech helmet, which in turn explains the presence of a HUD, while others leave your head bare and uses HUD elements only when necessary. The classical shooter HUD usually consists of a status meter, ammo meter and a cross-hair. All of the mentioned are often present in modern shooters as well, although their size and focus have been diminished. In some games the health meter has been replaced completely by other visual and auditory cues. An example would be Killzone 2, where the edges of your screen is splattered with blood and the colors start to fade, if you are low on health. This is emphasized by your team mates, scream at you to find cover [Eur09, Killzon 2].

There are two types of visualizations when it comes to gear. Those that use it to ensure realism and those that use it only as a decoration, which should not take precedence over game play features. In Bioshock 2 this is not consistent, since you on one hand is hindered by the weight, of the suite you are wearing and the edges of the helmet are shown to you (moving as you move), but on the other hand you can reorient the camera in such a way, that the suite should not allow you to. The fact that you can not see your legs, even though you can tilt the camera in that direction, is a dead give away [Gam10, Bioshock 2]. In Halo Reach you look through the helmet, wear the suite and are enhanced by it [Stu10, Halo Reach].
6.3. GENRE

**Third person**

Third person shooters tend to have the camera positioned slightly behind the character (showing everything above the waist in default view), which is positioned more to the side than to the center of the screen. The distance between the camera and the protagonist varies as with all games, but is now also used to limit the player’s field of view. This makes it harder to see enemies sneaking up from behind, keeping the player on his or her toes [Cap07, Resident Evil 4].

The camera can not be rotated around the character, as is the case with other third person games, but rather along with the character, since it should always be facing the center of the screen with its weapon. The same applies for vertical reorientation. There are of course other implementations like the one in *Kane and Lynch 2*, where the camera is not locked to the character per se but rather to the position behind the character. Rotating the camera around that position, makes it possible to almost fully exclude the character from the field of view, its back turned against us, on one edge of the screen [(Ed10, Kane and Lynch 2]. This is probably done to further enhance the feeling that you are an observer, rather than actually controlling the character.

Rapid movement often causes motion blur and camera bobbing. This is the case in *Gears of War 2*, where sprinting causes the character to huddle and run in the desired direction, accompanied by a lowered camera. This has a feeling of animation rather than direct player control. The same goes for jumping, which is performed almost solely through crawling over chest high walls [Stu08a, Gears of War 2].

In *Kane and Lynch 2* the camera not only behaves as a hand held cam corder, but also takes on the same characteristics when it comes to “effects”. The picture seems to be of low quality due to a grain effect, light sources get overexposed, which cause flares, and motion blur on rapid movement[(Ed10, Kane and Lynch 2].

The HUD is often minimized during regular navigation; in some cases only showing how much ammo is in the currently equipped weapon.

**Combat**

The combat is in its essence the same for both first and third person shooters. The only major difference is that you still see a small part of the protagonists body, besides the weapon and the arm wielding it, in third person games. The camera works the same in both types, once aim-mode has been entered in third person games. It can be oriented in every way the human head can be and it follows the center of the screen, where the cross-hair is located.

Different weapons have different sights, which result in different ways of zooming. On the other hand the trade off for zooming is almost always the same. You get an enhanced view of a limited area, while loosing your view of the immediate surroundings.

Cover based shooting is something that becomes increasingly common in shooter games, since taking direct damage often results in a quick death. As graphics get more realistic, so does the game play, at least in a lot of cases. The old “run and
gun” approach is no longer viable in every game. Moving from cover to cover is therefore common practice in all kinds of shooters, but it has almost been taken to the extreme in third person shooters. Every level is now conveniently covered with chest high walls for you to hide behind. Once a wall has been reached, you might not be fully in cover until you have snapped your character’s back to the wall. When this has been done the controls are limited, allowing you only to pop your head up to shoot at the enemy, until you disengage from the cover [Stu08a, Gears of War 2], [Ed10, Kane and Lynch 2]. The same control limitations seldom exist in first person games, where crouching behind cover is enough.

The HUD is often complemented with additional elements, such as the cross-hair, an ammo meter and on some occasions, additional elements such as a radar, shield gauge and locations of current objective. These extra elements either appear with the press of a button or are visible as a part of a high tech helmet. The use of health meters seem to be less and less common and have instead been replaced by visual cues, such as blood spatter on the edges of the screen or fading colors.

**Cinematic**

There are three kinds of cut-scenes in shooters; interactive, real-time and pre-rendered. Interactive cut-scenes bring the story forward without removing player control. They are performed with different level of complexity. The most basic one involves only voice communications, either between fellow soldiers or over the radio with HQ or an AI. The most advanced one includes all sorts of scripted behavior of the other characters present, like conversations, arguments or fist fights [Gam10, Bioshock 2], [Stu08a, Gears of War 2], [Stu10, Halo Reach], [Eur09, Killzon 2].

Real-time cut-scenes are rendered by the game engine in real time and portraits non-interactive scripted events. In first person perspective games these are usually viewed through the eyes of the protagonist. On rare occasions when that is not the case, an outside source is usually used, such as a security camera or a tactical perspective [Stu10, Halo Reach]. The higher the quality of the cut-scene the less
6.3. GENRE

likely is it that it is in first person [Stu10, Halo Reach], [Eur09, Killzon 2]).

Pre-rendered or cinematic cut-scenes are usually only used on major storytelling events, such as the introduction of the story and the ending.

Cut-scenes usually mimic the behavior of regular films, but there are a few that try harder than others. In *Kane and Lynch 2* the cut-scenes really give the impression of an amateur documentary, where the lead characters are followed by a guy with a hand held camcorder. Lens flares, grainy filters and irregular camera motion is a part of the experience [(Ed10, Kane and Lynch 2)].
6.3.2 Action/adventure

The genre of action and adventure cover a wide range of different game types, everything from pure adventure games like Myst to more combat oriented action games like Bayonetta. They do on the other hand have a few things in common, which defines the genre, namely; navigation and exploration of complex levels, fluent real-time combat; problem solving in the form of puzzles and a more or less linear story.

The most common way to visualize gameplay in an action/adventure game is through a three dimensional third person perspective, even though both 2.5D and the first person perspective are utilized on occasion.

As with the shooters most games in this genre have no real interactive dialogue sequences, but there are of course exceptions like Monkey Island. Most of the conversations are being conducted either in interactive “cut”-scenes or in regular real-time ones, where the dialogue is scripted.

A few examples of action/adventure games include: Assassin’s Creed: Brotherhood, Bayonetta, Heavy Rain, inFamous and Mirror’s Edge

Figure 6.4: Action/Adventure genre structure (For a full view image see appendix B.2).
Sequences

Like most other games the action/adventure genre is based upon the navigation, combat and cinematic sequences, but they almost never contain the dialogue sequence. An interactive dialogue between characters, does not seem to be necessary in this genre, since it rarely digresses from the linear storyline. Of course there are a few exceptions, where the player gets to make moral decisions, which effects the outcome of the story, but it is almost always done through action and not speech. This is, if anything, the most prominent factor that separates role playing games from action adventure ones.

Navigation

The most dominant means visualizing navigation in action adventure games, is through the use of the third person perspective. This is either done by positioning the camera behind the character, differences in distance are common both between games and between situations in the same game, or by locking the camera either to a fixed path or to fixed positions. Even though the third person perspective is the most dominant one, it does not mean that it is the only one. On occasion, games utilize the first person perspective, either consistently through out the entire game [DIC08, Mirror’s Edge] or when it is more helpful (mostly to deal with camera collision, occlusion or mini-games).

First person

The first person perspective is quite uncommon in the genre, since it makes it harder for the player to perform complex and precise maneuvers, when it comes to pure navigation like jumping. While it puts more strain on the player input, it also makes for a more realistic experience. With camera collision and occlusion out of the way, more time and effort can be put into making the experience as smooth and realistic as possible.

In the game Mirror’s Edge by EA DICE a lot of polish has gone into making the experience of running through and navigating complex environments with multiple paths, as smooth and fluent as possible. They have taken the first person perspective to a completely new level by utilizing small and subtle effects to make the entire journey more realistic [DIC08, Mirror’s Edge]. A few of the effects used are as follows: a slight camera shake to simulate head bobbing while running; motion blur around the edges of the screen while sprinting; jarring of the camera on impact with solid objects, such as doors and ledges and a slightly blurred view, when performing physically demanding moves, such as pulling one self up from a ledge [DIC08, Mirror’s Edge]. All of these things emphasize the reality and the fragility of the human character, something that is not common, but not that hard to implement. It shows, through the camera, that it is a game of finesse, motion and tactics and not a run and gun game.

Behind

The most common view in third person action/adventure games is behind the pro-
Figure 6.5: Mirror’s Edge by EA Dice (on-mirrors-edge.com).

tagionist. The distance to the character and the player’s influence in regard to camera control, differ from game to game and from situation to situation. A lot of games use the distance between the camera and the protagonist as a buffer zone, that can be increased or decreased, either to create an effect or to deal with collision and occlusion. The former can either be used to create an establishing shot, which limits the necessity of a mini-map, or to set the feeling of the surroundings (cramped, large or open). The latter seems to be a simple and functional way to deal with the problem. The camera is moved closer to the protagonist; the tighter the space between camera and surroundings gets, until it is either transferred to the first person perspective or until the protagonist is faded out. When fading out, the character one can do two different things; either make the character fully transparent [(TH10, Darksiders] or make only the outline of the character visible [(SC09, inFamous].

Common practice (in the games analyzed) seems to indicate that the player should have full control over the camera orientation, in relation to the protagonist, at least when no movement occurs. This is where a slight difference of opinion seems to exist. Some games allow the orientation of the camera, as decided by the player settings to be constant when moving [Ubi10, Assassin’s Creed], [The10, Enslaved], while others seem to prefer to slowly reset the camera to its original position forcing, the player to constantly move the camera while running [(Se10, Bayonetta], [(TH10, Darksiders].

Two common aspects of the navigation in this genre is jumping and climbing, since most levels are more complex than just running through killing enemies. When it comes to jumping there are three ways of automatic camera control. The first one does completely nothing, it works in the same way as the regular camera controls, the second one is an adaptive one and the third one is a locked one. The adaptive one tilts the camera slightly downwards, showing the landing area for the player without its input. This is usually used only on jumps that are significantly longer or higher than just jumping up and down on location. The locked one only allows
6.3. GENRE

jumps between key locations and works much like the path based camera [The10, Enslaved].

When it comes to climbing, the camera generally works in the same way between games with a few smaller variations. It is almost always positioned directly behind the character and allows the player to rotate the camera almost 180 degrees around the character, since you cannot move the camera through the wall, you are climbing. The small variations mentioned all relate to the way you can pan the camera. Some games allow limited panning to help you see the surroundings, others do it for you [The10, Enslaved] while some does not allow it at all.

It gets more and more common to use special camera effects, triggered by moving the character close to certain objects or by performing certain actions. In Enslaved: Odyssey to the West, this is used quite frequently to simplify navigation and to highlight certain areas. In a section of the first level of the game, the camera is moved to a fixed position, tracking the protagonist from above (without breaking the line of motion) [The10]. This creates an effect that emphasizes the environment, without disturbing the players motion. The camera is also used to simplify navigation around corners and other sharp edges, such as climbing over ledges by moving it to a neutral, position showing both sides.

In some cases these effects do not always work out for the best. An example of this might be navigating through doorways in Bayonetta, where the camera jumps to a new location on the other side of the door, breaking the line of motion and confusing the players, by inverting the characters direction of movement [(Se10, Bayonetta].

The HUD has a strong presence among these games, as with most other games. It usually consists of at least two elements, namely some form of character status meter, displaying health and the like, and a mini-map for simplified navigation. This as everything else, varies from game to game. In some cases, where the level-design is very linear, the mini-map shines with its absence and in some cases the HUD is only visible, once hostile actions have been taken [(Se10, Bayonetta].

Path-based

A less common, but very useful way of handling the camera, is to use a path-based system. In short this means that the player has no control of the camera, what so ever, since its path and orientation has been set beforehand by the level designer. This makes it possible for the player to focus entirely on the task at hand, without having to worry about the camera angle, if done correctly that is. This also means that the camera usage can be pushed a step further, since all possible outcomes of camera movement can be tested prior to release.

An example where this has been executed well, in most part, is the God of War series. In God of War regular navigation works in much the same manner as regular third person games. The camera is positioned behind the main character and the distance between camera and target depends on the amount of space in the surroundings. What makes it different from other games is its automatic switching between camera views. Depending on the situation, the camera jumps between a
behind, side and top view in a very fluent manner, it seldom leaves the player out in the cold. This is also used to emphasize important objects, paths or as an effect [Ent05, God of War].

One problem that occurs from time to time is the breaking of the line of motion, when transitions between different areas of a level occur. This is especially the case when passing through doorways or moving between areas, with varying degrees of navigational space occurs instantaneously (such as moving from a cramped hallway to a ballroom sized area). Upon this breaking, the player can sometimes get confused and halt all motion, which causes the controls to invert on release since the direction of movement is opposite to what it was before the break, causing the character to move in the wrong direction upon continued movement [Ent05, God of War].

**Fixed**

Fixed cameras can be used from time to time, in sequences of games to emphasize certain objects or make navigation easier, but they are rarely used for entire games. One exception to that “rule” is the interactive drama/mystery/adventure *Heavy Rain* by Quantic Dream, which is more of a slow paced mystery, rather than an action game. *Heavy Rain* introduces a relatively new way of telling a story, both in regard to the camera system and the player interaction. The game is played in a third person perspective, viewed through a set of fixed cameras, located throughout the level (fig 6.6a), which you can toggle between to get complete coverage of the area. You can move freely in the levels and interact with objects, using different button combinations, to collect clues, which in turn drives the story forward [Dre10, Heavy Rain]. Since these views are fixed, much like the path-based version, they can be used to tell a very precise story (closely resembling the way a director tells a story in a regular film).

What is also quite new, is the way of interacting with objects using different button combinations. This is done by showing the player, which button(s) to press near different objects, to interact with them. These combinations move according to the emotional status of the protagonist. In most cases they are solid and unmoving, but when the character is experiencing an increased flow of emotion, such as fear or anxiety, the icons begin to move, blur and shake, resulting in increased difficulty in executing them. The same way of visualizing combinations on screen is used to show alternatives to decisions in dialogue and different character thoughts (fig 6.6b), which heightens the player’s awareness of the protagonist’s emotional state [Dre10, Heavy Rain].

What might be important to learn from *Heavy Rain* is, that it is possible to use fixed angles and jump cuts to enhance the visual storytelling, without confusing the player. It was done to some regard on occasion in *God of War* and in *Enslaved*, but it is rarely used as an effect, rather than basic emphasis.

**Combat**

Combat in action/adventure games generally tend to include fighting with both close and long range weapons.
6.3. GENRE

**First person**

Combat in the first person perspective usually closely resembles that of the regular navigation, since you can not move the camera. What might change is the way you target enemies.

**Third person**

The camera usage during close combat in third person action/adventure games usually resembles the system used during regular navigation, but with a few modifications. The camera can sometimes move slightly back and above, to show as much of the enemies as possible or it can focus on the target selected on occasion, occluding the other ones. The player should generally not have to move the camera during combat, since he most likely have his hands full already, but some times it is necessary. This is usually the result of glitches in the camera system, since everything can not be foreseen by the developers, when you have a dynamic system, and is something you can live with, if it does not occur that often.

In a lot of third person action adventure games the combat is not only limited to melee but also includes ranged combat. This demands the presence of an aim mode, which simplifies the process of aiming the various ranged weapons at the enemy. There are two major ways of doing this and the first one is target lock and the second one is free aim.

Target lock allows the player to choose a target and fire at it while moving. Selecting a target is usually done through toggling between the ones in range. This diminishes precision in favor of fluent combat. The target to, which the character is locked is usually focused on by the camera, while still framing the protagonist. This is used in *Bayonetta, Darksiders* and *Assassin’s Creed* in conjunction with free aim mode. ([Se10, Bayonetta],[TH10, Darksiders],[Ubi10, Assassin’s Creed].

Free aim favors precision over fluency (even tough combat can flow pretty well in this mode, which can be seen in *inFamous* and usually results in fever projectiles fired. Upon entering free aim mode, the camera usually changes position, moving closer to the protagonist and a bit to the side, and a cross-hair appears on the center
of the screen. To which degree the camera moves closer, depends on the game and on the level of zoom of the “weapon”. Some games also allow the player to toggle from which hand the weapon should be fired, moving the protagonist from side to side in the process [(SC09, inFamous).

In some games the HUD is hidden, until hostile action is taken [(Se10, Bayonetta], [Ent05, God of War]. In these games, much like the other ones (where it is always visible), it usually consists of a status meter and on occasion a component for weapon switching. Regular HUD elements are sometimes complemented by visual effects to emphasize important knowledge, such as low health or power. This is usually done by adding or draining color from the screen. In some cases, the colors fade to black and white, in others it blurs, the edges of the screen start to flash red or a combination of two or three are used. In inFamous the color fades and splotches of blood appear around the edges, while in Enslaved the colors blur and the edges pulsate in red [(SC09, inFamous],[The10, Enslaved].

Combat using path-based cameras work much in the same way as with regular third person combat. The only difference is that you have no control over the camera, which puts a high demand on its planning and execution.

**Fixed**

The combat in Heavy Rain is performed as an interactive cut-scene, where the only control the player has is through predetermined options viewed, as quick-time events. This makes the combat extremely cinematic and more nerve wrecking than regular combat. The quick time events are pretty lenient and if you miss one, you probably won’t die as a result, but you will take a beating, which makes the characters more believable.

**Mini games / finishing moves**

A common practice among action/adventure games is to have finishing moves that execute an enemy in a more “elegant” fashion than regular hack and slash. This can either take on the form of a regular or an interactive animation. The former are usually used to show off the cool effects and animation, while the latter is used as a mini-game-mode to test the players reactions. This mode usually removes all form of control from the player, besides pressing the buttons prompted for on the screen at the right time (quicktime events). During this combat scenario the camera follows the character, while it executes the devastating moves that will eventually bring down the enemy [(Se10, Bayonetta],[TH10, Darksiders],[Ent05, God of Wat].

**Cinematic**

There are three different kinds of cut-scenes used in action/adventure games; interactive, real-time and pre-rendered. The interactive cut-scenes are not that common, but they are used in Assassin’s Creed: Brotherhood, Heavy Rain and inFamous. In Assassin’s Creed they take the form of bleed through effects from earlier memories, which materialize as semi transparent ghosts, during the first section of the game, where they tell the story of what last happened at that location [Ubi10, Assassin’s
6.3. GENRE

Creed]. They are seamlessly integrated and cause no interruption to the regular navigation what so ever. In *Heavy Rain* the interactivity manifests as quick time events during “regular” cut-scenes. In *inFamous* the interactive cut-scenes come in two different forms. The first one is through television sets, spread around the game world, which on occasion broadcast news reports. These broadcasts can either be viewed, ignored or listened to, if one remains close enough. The second one is stolen memories from dead victims, which are materialized as ghosts you can follow, until they reach the villain responsible for their death [(SC09, inFamous).

Real-time cut-scenes are the most common kind off cut-scene, since they utilize the in game rendering engine to render pre-defined scenarios in real-time. This makes it possible to allow player actions to affect the cut-scenes without having to pre-render every variation possible.

Cinematic cut-scenes are only used to show of important aspects of the story and are pre-rendered using a real high end rendering engine and replayed as a video file.

There are a few different variations of cut-scenes used in action/adventure games that are not rendered in a standard “movie” fashion. These are usually used in conjunction with regular cut-scenes.

*Bayonetta* for example, uses a mix of live action and stills in their cut-scenes. This is done by showing a strip of film, containing images in multiple frames and then focusing on different ones, to tell the story in the correct order [(Se10, Bayonetta).

*Enslaved* uses collectible flashbacks, consisting of a series of distorted images, which as a hole tells a story [The10, Enslaved].

Both *inFamous* and *Mirror’s Edge* use cinematic cut-scenes rendered not as photo-realistic but as a cartoon. *InFamous* compose them as pictures in a comic book, moving between semi static shoots, while *Mirror’s Edge* uses a more live action approach [(SC09, inFamous],[DIC08, Mirror’s Edge].
6.3.3 Stealth

Stealth games are very similar to action/adventure games in every aspect but one and that is that Combat and navigation should be performed with minimal detection. If you are discovered during infiltration of the enemy territory, the likelihood of you being killed is high, since you can not survive against multiple enemies at once.

A few examples of stealth games include: *Hitman Blood Money*, *Metal Gear Solid 3: Snake Eater*, and *Splinter Cell Double Agent*.

Figure 6.7: Stealth genre structure (For a full view image see appendix B.3).

Sequences

Stealth games consists of navigation, combat and cinematic sequences that all are focused on avoiding combat.
6.3. GENRE

Navigation
Most stealth games are played in a third person perspective, since an overview of
the surroundings is very important, if you are going to avoid enemy detection. The
camera is either positioned behind or above the character, and the player has a
varying degree of control of its position and orientation. In *Metal Gear Solid 3: Snake Eater* the camera is locked to the character, who is centered on screen, and
the only reorientation available, is the slight shifting of the camera position in every
direction, which resets upon release [Kon04, MGS3]. In *Hitman: Blood Money* on
the other hand, the player has full control of the camera orientation, which will not
reset when moving [(Ei06, Hitman: Blood Money].

What many stealth games have in common is the ability to enter a first person
view, either used for scouting areas, not visible through the regular camera control
or to get a more detailed view of nearby objects.

There are a few automatic camera controls used in stealth games, that usually
involves camera adaptations while jumping, climbing or pushing up against walls.
Another adaptation is the way the camera moves closer to the protagonist to avoid
collision with other objects. This is used both when standing and crawling.

Figure 6.8: Metal Gear Solid 3 navigation (ign.com).

The HUD usually displays a status bar, mission objective and/or action buttons,
this differs from game to game. A game that does not display a status frame, usually
use other visual queues, such as fading colors to visualize health loss. Some games
allow the player to toggle the visibility of a mini-map on screen [Ubi08, Splinter
Cell Double Agent], while others demand that the player pauses the game to view
it [Kon04, MGS3].

Combat
The combat in stealth games is very similar to that of action/adventure games, when
it comes to control. The aim-mode is present and works in the same way; moves
the camera closer and places a cross-hair on the screen, in most cases. An exception
is the controls used in *Metal Gear Solid 3*, where the camera is unchanged and one
reorients the protagonist’s body instead, using it to aim in the general direction of the enemy [Kon04, MGS3].

Once combat is initiated, the HUD is supplemented with an element, showing how much ammo is left in the currently equipped weapon.

**Cinematic**

A common trait among stealth games is the use of mission briefings and radio communication to tell the story. Both of these features are used in *Metal Gear Solid 3*, *Splinter Cell Double Agent* and *Hitman Blood Money*. In *Metal Gear* the introduction is an extensive mission briefing, with actual real world footage mixed in with real-time rendered “cut”-scenes and the radio communication with the headquarters is extensive. It is performed using a visual representation of both parties mixed with voice over communication and text transcription. In *Hitman* and *Splinter Cell* the radio communication is conducted during actual game-play and a lot of the mission briefing is done before the level is loaded [(Ei06, Hitman: Blood Money), (Ubi08, Splinter Cell Double Agent)]. *Hitman* uses a quite interesting technique to present the target during game-play. This is done by closing of a part of the screen, in which a “live” stream of the target is depicted, and is similar to a viewport transition [(Ei06, Hitman: Blood Money).

The non-interactive “cut”-scenes usually change the aspect ratio of the screen, showing black strips, to let the player know that this is a “cut”-scene.
6.3. GENRE

6.3.4 Platform

A platform game is a game, where the core gameplay component is traversing the level. The purpose of each level is to move from point A to point B without dying, at least not to many times, by jumping between platforms. It is often combined with finding the right path, collecting objects or solving puzzles. Some games are more focused on puzzles than others, like Portal 2.

The most common way of visualizing platform games is through the use of the third person perspective; this can be done in all previously mentioned dimensions (2D, 2.5D and 3D). The original way was to use 2D, since it was the only real option available at the time, combined with a side view. This is still fairly common in smaller games such as Braid, but also in bigger games honoring the “old ways”, like New Super Mario Bros for Nintendo DS. Another common way is to use the same “camera style”, but using the “new” 2.5D, where everything is rendered in 3D, but the player can only move along the up and side axis.

A few examples of platform games include: Braid, Little Big Planet, New Super Mario Bros and Trine

Figure 6.9: Platform genre structure (For a full view image see appendix B.4).
**Sequences**

Platform games mainly consist of navigation and cinematic sequences, which on occasion are complemented with combat sequences.

**Navigation**

The majority of the platform games are produced using the third person perspective, in either 2D, new 2.5D or 3D. The navigation is generally the same in 2D as in new 2.5D, since the only difference is the way objects are rendered and not the number of dimensions available for movement.

The common traits of camera control for 2D/2.5D platform games are as follows. The character is almost always positioned slightly behind the center of the screen, relative to the direction of motion (fig 6.10a). If the direction changes, the camera lags behind to reposition itself. If the character reaches an edge of the level, the camera locks to that position, allowing the character to move freely within the screen (fig 6.10b) [NN09, Braid].

The player has no direct control of the camera position and orientation. It is either character relative, locked to the character, or screen-relative, which allows for free movement within an area of the screen, without affecting the camera. It is also common to use directional lag, to add a rubber band effect to the camera motion, to prevent it from performing to sharp turns [Fro09, Trine][Eur08, Little Big Planet].

(a) Trine by Frozenbyte  
(b) Braid by Number None

Figure 6.10: Camera control in third person platform games.

Third person platform games rendered in 3D usually have the same camera controls as action/adventure games, using the same perspective. The camera is positioned behind and above the character, and the player has full control of the camera. It only moves on its own, to improve the field of view, either by avoiding collisions and occlusion or to follow the character as it edges close to the screen, which can occur when climbing/jumping to new elevations [Ins03, Ratchet and Clank 2 Going Commando].

As mentioned above there do exist a few first person platform games, such as the Portal-series by Valve, which focus quite highly on puzzle solving. They use
6.3. GENRE

the same camera controls, as every other first person game, which means that the player has full control of the orientation of the camera and that it can move in every direction a human head can [Val11, Portal 2].

Many platform games have a HUD, but what it displays varies from game to game. The most common item displayed on the HUD is the number of objects/points collected, which tend to fade only to reappear when new ones have been collected. Some games show lives [Nin06, New Super Mario Bros] or status frames containing the health level, among other things [Fro09, Trine][Ins03, Ratchet and Clank 2 Going Commando]. Others contain cross-hairs for aiming [Val11, Portal 2][Fro09, Trine] or timers, which indicate the amount of time left, to complete the level or challenge [Eur08, Little Big Planet][Nin06, New Super Mario Bros].

Cinematic

Cinematic sequences in platform games are usually not as extensive as, in say role playing games or action/adventure games, at least not in 2D and 2.5D games. In these games the focus remains on interactivity, which is why there are so many interactive “cut”-scenes. An example is Braid, where the story is told through text displayed on screen and is activated by interacting with books placed in “story levels” (fig 6.11)[NN09, Braid]. Another one is Trine, where a lot of the story is told during actual game-play by a narrator [Fro09, Trine].

Interactive “cut”-scenes are not only used in 2D and 2.5D games, but also in 3D games. In Portal 2 the majority of the “cut”-scenes are interactive. They mostly entail one-way communication between different non player characters and the protagonist, either over a distance (radio/monitor) or in person.

![Figure 6.11: In Braid the story is told through text.](image)

Real-time (scripted) “cut”-scenes are also quite common. They are in fact the only kind of “cut”-scene there is, in Ratchet and Clank 2 [Ins03]. In Trine they take the shape of static images, bundled together with a few effects and a voice telling the story [Fro09, Trine].
6.3.5 Role-playing

A Role Playing Game, or RPG, is in its essence a game, where you take the role of another character, much like in the theater, of whom you have almost complete control. The character is then put into a context, usually a grand adventure, with a varied collection of other characters, either computer or player controlled. During the adventure, the player or players are usually faced with moral decisions, that affects the outcome of the story itself and (in many cases) the fate of their entire world. In accordance with the decisions they make, the characters evolve (gain levels), learn new things (abilities) and discover their true potential. This evolution is often somewhat player controlled, which makes the experience quite unique for each player and makes it re-playable, since the story can go in different directions, depending on the decisions you make.

The role-playing genre is largely based upon the pen and paper version of the game and is therefore often turned based, story driven and involves a lot of ability points. The original role-playing games, were as mentioned, played with pen and paper. The story was told by a dungeon master, which tells the story and acts out all the non-player characters. Each player in turn had his or her own character, with an accompanying character sheet, which they played during the adventure. The setting of the story can greatly vary, both in regard to time and place, but the character composition of the party almost always contains the same stereotypes.

A few examples of role-playing games include: Dragon Age Origins, Final Fantasy XIII, Mass Effect 2 and World of Warcraft.

![RPG genre structure](image)

Figure 6.12: RPG genre structure (For a full view image see appendix B.5).
6.3. GENRE

Sequences

Role playing games, as with almost all games, are based on the four sequence components, but some are more prominent than others. Both the dialogue and cinematic components get a lot more focus and game time, than they would in a regular first person shooter. This is due to the fact that role playing games are often more story driven than other games.

Navigation

There are two different kinds of navigational input that affect the way the camera is used in role playing games. Those are continuous and point and click. Continuous player input is defined, in this context, as direct control of the player character through continuous button or joystick input. An example of this might be how you move the character in *Final Fantasy XIII*, you press and hold left and it moves to the left. Point and click on the other hand is defined, in this context, as abrupt character control [Eni10, Final Fantasy XIII]. It is often done by pointing at and clicking on a location in the level, after which the character moves to that location. An example of this is the navigation in *Neverwinter Nights* [Bio02].

Continuous

Continuous navigation is the most common kind of navigation in role playing games, since it gives the player complete control of the character and makes navigation easier in a three dimensional environment.

The camera is most often placed behind the main character in a third person perspective. How far behind, usually depends on how cramped the surrounding area is; narrow spaces often require a more tightly bound camera than open ones. In *White Knight Chronicles* the camera moves from being four or five meters behind the character to first person perspective depending on camera location and distance to obstacles [(SC10, White Knight Chronicles].

The player is often left with varying degrees of control over the camera. In a lot of cases, one can rotate the camera freely around the main characters up and right axis (even while moving). This allows the player to inspect the character from almost all angles (it is usually not possible to view the underside of a character) and makes it possible to inspect the scenery while navigating. In some cases it is even possible to zoom in on the character or the surrounding areas either by using the aim mode of a weapon [Bio10, Mass Effect 2], the inspect mode [EAD06, Zelda: Twilight Princess] or a build in zooming function [(SC10, White Knight Chronicles].

There are on occasion, exceptions made to the regular camera control, usually on sections of a level, that is more complex or where certain important objects need to be emphasized. These exceptions often take the form of fixed cameras or short sections of path based cameras, both of which are not changed into abruptly but rather smoothly. This occurs from time to time, during regular level navigation in *Final Fantasy XIII* [Eni10].

The camera is often aided in the storytelling by the HUD, which usually presents the player with a mini-map and/or party frames, to ease the game-play (especially
the navigation of large non-linear levels).

**Point and click**

The method of navigating through point and click is more common in the earlier role playing games and in the new versions, honoring the classics. It was used in conjunction with a slightly tilted top down camera, to present the player with a strategic view, simple navigation and a three dimensional experience.

The camera is often detached from the character, much like in a RTS game, and is moved either via the use of buttons or mouse motion (moving the cursor close to an edge of the screen, moves the camera in that direction). You can also rotate the camera around the character.

Sometimes it is even possible to switch between the top down view and a regular third person view, presenting the player with the choice to have the best possible view, depending on the situation. This, on the other hand, puts a higher strain on both the level design and the camera system, since both have to be functional and aesthetic for two different views at the same time. An example, where this has been well executed, is *Dragon Age: Origins*, where you can switch view with the press of a button [Bio09]. This is also used to prevent the camera from colliding with very tall objects (the camera is interpolated from the top view to the regular behind view).

Like the continuous navigation there is a HUD present at all times, which gives you quick access to the mini-map, abilities and the party frames. All of this is visible, since the combat is always continuous.

**Dialogue**

Dialogues in role playing games, tend to depict how “real” conversations are carried out in film. This means that the focus generally lies on the character, currently speaking and that it alternates as the participants take turns in the conversation. Basic cinematographic rules are often used, such as the 30 and 180 degree rules and screen matching. The camera setup usually mimics one form of the triangle principle and the cuts between angles are often jump cuts.

A lot of role playing games feature dialogues only in the form of a non interactive “cut”-scene, but there are a few cases, where the player gets the opportunity to choose what his or her character says to non player controlled characters. This is usually done by letting the player choose from a set of predefined options, which are displayed on the HUD. This can either be done in complete sentence form [Bio09, Dragon Age: Origins] or as paraphrases [Bio10, Mass Effect 2]. The response can either be read out loud to the player by the character [Bio10, Mass Effect 2] or be delivered in silence [Bio09, Dragon Age: Origins].

**Combat**

The combat system is generally separated from the navigational system in respect to overall view. A lot of role playing games have a separate combat mode, which one enters upon initiating combat. This mode can either take place in a remote
6.3. GENRE

combat “dimension”, where the combat usually is more or less turn based, or on
the current location in the level, where the combat is played out in “real-time”.
Real-time combat is implemented to be more or less seamless with the rest of the
game play while turn based combat is more focused on strategy and execution.

**Real-time**

Real-time combat is defined in this context, as combat that takes place, on location,
in the level, with minor changes to the camera behavior, and that is fought in more
or less real-time. A few examples of this might be the combat system in *World of
Warcraft*, *Mass Effect 2* and *White Knight Chronicles*.

In *World of Warcraft* the only changes made to the system, are the added in-
formation on the HUD, regarding enemy health and combat information (such as
damage done, abilities used and experience gained) [Bli10, WoW]. In *Mass Effect
2* the camera enters aim mode, i.e. it moves closer to the character and looks over
the shoulder to improve aiming, upon drawing a weapon [Bio10].

A common practice is to add short animations to special abilities or combos
to make them stand out and make the combat segments more interesting. During
these animations, the player usually loses control of both the character and the
camera, since they both are used in the animation.

**Turn based**

Turn based combat usually takes place in a “combat dimension” between the en-
emies attacking and the player controlled characters. The transition from regular
navigation to the combat dimension, is usually hidden behind a short animation.

The general camera control is usually limited, since the camera tracks each
characters actions, as it takes action (once it is its’ time). This gives the player
enough freedom to concentrate on the ability selection and tactics, without having
to worry about getting the right camera angle to see what he or she is doing. It
also allows the animators to show off their work, as each attack gets direct camera
attention [Eni10, Final Fantasy XIII].

The HUD elements visible are usually the same as the ones used in real-time
combat situations, i.e. party frames, enemy frames and ability selection. On the
other hand you rarely see the mini-map during these sequences, most likely since it
has no real function and takes up space on the screen.

**Cinematic**

As with almost every genre the cinematic sequences in role-playing games are com-
prised of interactive, real-time and pre-rendered “cut”-scenes. Games with a high
number of dialogue sequences tend to have less regular “cut”-scenes compared to
those that do not utilize dialogues at all. An example comparison might be *Mass
Effect 2* and *White knight chronicles* [Bio10], [SC10]. A common indicator among
most role-playing games that a “cut”-scene is starting is the fading of the HUD.

Short Real-time or interactive “cut”-scenes or animations are often used to show
off finishing moves or special abilities during combat. During these segments the
only player input possible occurs through pressing the buttons prompted for on
screen.

One of the few games where something different is presented during a “cut”-
scene is *Lost Odyssey*. The game used viewport transitions during multiple real-
time “cut”-scenes where it is important to show multiple characters reaction to the
same event.
6.3. GENRE

6.3.6 Real-time strategy

Real-time strategy (RTS) games are games that allow the player to experience different strategical scenarios, during which the player is in command over a base, used for harvesting resources and deploying multiple tactical units. The scenarios can include everything perimeter defense to pure offense, but almost everyone of them demands that the player first establishes a base.

The story is told a bit differently in RTS games. Instead of telling it through the character you are controlling, as with almost every other genre, it is told through a few specific units, which can not always be controlled in all situations. If you take Starcraft 2 as an example, you have full control of a great number of units, but you only control the characters driving the story forward during a few missions. These characters, whom you play, are usually in command of their forces and only communicate through radio/video transmissions [Ent10].

A few examples of real-time strategy games include: Command & Conquer, Starcraft 2 and Warcraft 3

![RTS genre structure](image)

**Figure 6.13:** RTS genre structure (For a full view image see appendix B.6).

**Sequences**

Real-time strategy games tell their story through three sequence components, namely navigation, combat and cinematic. The two major components are navigation and
combat, which from time to time merges into one.

**Navigation**

The navigation works a bit differently in RTS games, since you can control over 100 different characters/units/assets. This means that the player needs to have a layout of the land and an overview of all available units. For this reason the camera is usually placed in a top down aerial view (fig 6.14a), with the ability to move the camera closer into a more tilted top down view (fig 6.14b) [Ent10, Starcraft 2].

![Camera control in Starcraft II by Blizzard.](image)

The position of the camera can be controlled either by dragging the mouse cursor to the edge of the screen, by direct button input or by clicking on the desired location on the mini-map. As mentioned above, the camera can be zoomed into a small degree, but it can not be rotated or reoriented in any other way. The same camera controls apply for both regular navigation and combat [Ent10, Starcraft 2][Ent02, Warcraft 3].

The different maps/levels on which the game is played, is all covered in “fog of war”, which hides everything that is not in range of player units. The same technique is usually used for areas outside the edges of the map, where the only visible elements are the contours of what might lie beyond [Ent10, Starcraft 2][Ent02, Warcraft 3].

A big part of the game-play is focused on establishing a base of operations for ones army. This is done by building structures near available resources such as wood, mineral deposits or gas pockets. One creates new structures by telling certain units, who’s only purpose is to gather materials and building structures, to build them at the desired location. When selecting a location for the building a semi-transparent version of the structure is displayed on screen, which is then moved into position by the player [Ent10, Starcraft 2][Ent02, Warcraft 3].

As you might imagine RTS games are focused quite heavily on unit control and are therefore usually accompanied by a quite extensive HUD. It usually contains a mini-map, command buttons for the selected units, unit portrait, mission objectives and available resources. The HUD can also indicate when constructions are complete...
or new units have finished their training.

**Combat**
The forces at your disposal can be controlled by selecting them and then clicking on the desired location. Different units have different abilities, which can be controlled through the HUD. Troops can be divided into different squads, which one can toggle between. This is very practical since battles can occur at multiple fronts at once. When a unit is engaged by the enemy, that location is highlighted on the mini-map.

**Cinematic**
The cinematic sequences in RTS games, tend to work a bit differently than in “regular” games such as RPG:s. There are of course pre-rendered and real-time “cut”-scenes, but there are also mission briefings and interactive “cut”-scenes, during which the unit portraits are used for communication.

Interactive “cut”-scenes, as mentioned above, usually take advantage of the portraits for the different units, as a means of communication, since the units themselves are too small to be used, as a good means of communication. This makes it possible to have detailed images of the character talking present, during regular game-play. The same technique is used during mission briefings.

Real-time “cut-scenes” usually depict events occurring in-game and are rendered in the same fashion. The only difference is that you do not have any camera control and that the number of HUD elements are limited. An example might be the conversation between Arthas and Uther in Warcraft III depicted in fig 6.15a. There are of course, other ways of visualizing real-time “cut”-scenes, which you can see in Starcraft II. These “cut”-scenes are still rendered, using the game engine, but tell the story in a more classical film fashion (fig 6.15b).

![Warcraft III by Blizzard](a)

![Starcraft II by Blizzard](b)

Figure 6.15: Different real-time “cut”-scenes.

Pre-rendered “cut”-scenes are limited, as with all other games, since they are expensive to make, and is therefore only used during major storytelling events.
Chapter 7

Conclusions

The entire purpose of this thesis has been to analyze and catalog different traits and aspects of video games to try and find a common structure, depicting how video games are being visualized today. The results described in this report could function as a reference for those wanting to create their own story based video games. It could enhance their understanding of Story visualization and hopefully help create better games. While this has been done, there are a few things that need mentioning in regard to results, structure, rules and the future of Story visualization.

7.1 What can this be used for?

The results presented in this report consists of two main components. The first one is the taxonomy containing all the tools and rules used to visualize video games today. This taxonomy can be used by game and level designers to plan the storytelling aspects of their games. Hopefully it will also serve as an inspiration to find new ways and new combinations of telling a visual story.

The second component is the analysis of the games and their categorization into different genres. This is also far from complete but will hopefully inspire others to pick up the torch and add their own contributions to the results presented in this report. The results could act both as an inspiration and as a guideline for everyone interested in developing video games. Hopefully it will inspire others to “think outside the box” and try something that is not commonly used in their genre.

7.2 Common structure in video games

As seen in the result section, there are quite a few common traits among video games in the same genre. A lot of them so defining that adding or removing one, might cause the game to jump into a completely different genre. This is especially the case when it comes to the presence of the different sequences the games consist of, which in itself is not strange, since they are core components of the story. For
example if one added strong dialogue sequences to *BioShock 2*, one might cause it to switch genre from shooter to role-playing game.

Each genre has its own trait that separates them from one another. Action/adventure games focus on navigation, puzzle solving and action. Platform games focus on pure navigation. Shooter games focus on action through the use of long range weapons. Stealth focus on action and navigation that occurs without the enemy detection. Real-time strategy games focus on action, planning and gathering in strategical scenarios. Role-playing games focus on storytelling, character evolution and player involvement. These are all clear cut genres, but there are aspects in them that are similar. Without the dialogue *Mass Effect 2* could be a shooter game and if one were to add more complex navigation to *Killzone 2* it could be an action/adventure game. Each genre consists of navigation and cinematic sequences and almost all of them have some form of combat. What is limited is the number of games outside the RPG genre that have real interactive dialogue sequences, but they do exist.

### 7.2.1 Outliers

As with every categorization there are elements, that is harder than others to place into the right category. One of these games is *Portal*, both one and two, which technically could fit into more than one category, depending on how you interpret the main game-play mechanic. It could be an adventure game, much like *Mirror’s Edge*, or a platform game, since a big part of the game revolves around navigation levels, or it could be a puzzle game. Some even go so far as to say that it is a first person shooter game, since it is played in the first person perspective and you “shoot” portals.

This factor makes it hard to define proper genres, but allows for cross genre game-play. This is not necessarily a bad thing, since breaking boundaries and new ground, is what brings the entertainment industry forward. It is just one thing that one should keep in mind, when analyzing games.

### 7.3 Exceptions to each rule

As the section title implies there are exceptions to almost every rule when it comes to Story visualization. The only requirement is that common sense is used and that the pros outweighs the cons. For example if one simply must break the 180 degree rule, due to some unchangeable aspect, one can do it if it is done a limited number of times and if it effects the player as little as possible.

There are of course other rules that apply more to the believability of the storytelling than that of the technical one. For example one might want to use the lens flare effect to emphasize the the setting sun. Lens flares are usually an indicator that a camera is present, capturing the moment, which breaks the illusion of immediacy. But since they are common in most films, due to the construction of the camera, the player is most likely used to their presence and might even miss them.
if they are not there. As mentioned in the background chapter immediacy can be achieved through extended use, in this case its constant presence in film.

7.4 Future work

The field of video games is a vast one and even though a lot of ground has been covered, both in earlier research and in this report, there is still a lot of games out there, that might change the concept of Story visualization. The fact that a structure has been found, for each genre, even though a limited number of games have been analyzed, tells us that there are conventions used, when creating certain kinds of games. A further analysis might conclude that there are vast gaps in the structure presented in this report or it might conclude that only small adjustments are needed. To prove one or the other an analysis on a much larger scale needs to be conducted, which was not possible in the allotted time for this master thesis. An alternative might be to present this mighty challenge to the gaming community and do the analysis together.

There are also quite a few fields, that has not been touched in this report, a few of which are: multi-player games, games for mobile devices, games using stereoscopic 3D and a lot of genres with a smaller storytelling component. All of which, deserves their place in a more complete structure and mapping of Story visualization.
Bibliography


BIBLIOGRAPHY


[Stu08a] Epic Games (Microsoft Game Studios). Gears of war 2. Xbox360, 2008.


BIBLIOGRAPHY


Appendix A

Complete list of games evaluated

- Assassin’s Creed: Brotherhood
- Bayonetta
- Bioshock 2
- Braid
- Darksiders
- Dragon Age: Origins
- Final Fantasy XIII
- Enslaved: Odyssey to the West
- Gears of War 2
- God of War
- Half Life
- Halo Reach
- Heavy Rain
- Hitman: Blood Money
- inFamous
- Little Big Planet
- Lost Odyssey
- Kane & Lynch 2: Dog Days
- Killzone 2

83
APPENDIX A. COMPLETE LIST OF GAMES EVALUATED

- Mass Effect 2
- Metal Gear Solid 3: Snake Eater
- Mirror’s Edge
- Neverwinter Nights
- New Super Mario Bros
- Portal 2
- Ratchet and Clank 2 Going Commando
- Resident Evil 4
- Shadow of the Colossus
- The Legend of Zelda: Twilight Princess
- Tom Clancy’s Splinter Cell: Double Agent
- Starcraft II: Wings of Liberty
- Trine
- Warcraft III: Reign of Chaos
- White Knight Chronicles
- World of Warcraft: Cataclysm
Appendix B

Genre structure
Figure B.1: Shooter genre structure (explained in section 6.3.1)
Figure B.2: Action/Adventure genre structure (explained in section 6.3.2)
APPENDIX B. GENRE STRUCTURE

Figure B.3: Stealth genre structure (explained in section 6.3.3)
Figure B.4: Platform genre structure (explained in section 6.3.4)
Figure B.5: Role-playing genre structure (explained in section 6.3.5)
Real-Time Strategy

Sequences

Cinematic
- Interactive
  - Video communication
- Real-Time
  - Portraits
- Pre-rendered

Navigation
- Third person
  - Position
    - Behind
    - Free-flying
  - Orientation
  - Zoom
  - Transition
  - Controls
  - Point and click
  - Visualization
  - HUD
    - Mini-map
    - Objectives
    - Portraits
    - Resources
    - Fog of war

Combat
- Third person
  - Position
    - Behind
    - Free-flying
  - Orientation
  - Zoom
  - Transition
  - Controls
  - Point and click
  - Visualization
  - HUD
    - Mini-map
    - Objectives
    - Portraits
    - Resources
    - Commands
    - Fog of war

Figure B.6: Real-time strategy genre structure (explained in section 6.3.6)