FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Framework for Developing Interactive 360-Degree Video Adventure Games

Francisco Pinho

U. PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Rui Rodrigues

Co-Supervisor: Rui Nóbrega

July 20, 2019

Framework for Developing Interactive 360-Degree Video Adventure Games

Francisco Pinho

Mestrado Integrado em Engenharia Informática e Computação

July 20, 2019

Abstract

Making game environments, be it in 2D or 3D is an extremely laborious and skill intensive task. 360^{o} content is a visual medium known for its increased spatial immersion and potential enhancement of the users' emotional response to content [EEW18]. The possibility of using interactive 360^{o} video or images as the game environments will allow talent to invest their creativity into other crucial aspects of game design such as narrative, sound and game mechanics.

Adventure games belong in a diverse genre comprising many different types of experiences and sub-genres, from text-based adventures to puzzle-oriented point-and-click games, a cinematic experience with player choices or even a mixture of crime investigation with courtroom drama. In essence, games with simpler means of user input and a bigger focus on interactive narratives and storytelling - a very relevant field at the cutting edge of virtual reality research [RTGG17]. Furthermore these games feature entirely non reflex based mechanics making the genre of "Adventure" the most fitting for integration with 360-degree visual media.

This dissertation presents a framework that will allow the streamlining of the creative process of these experiences by giving creators the tools to make a fully-fledged virtual reality adventure game with 360 visual media as an interactive setting.

The framework was implemented and usability tests were conducted to validate the viability of 360^{o} video based adventure games as an entertainment and storytelling medium.

Results displayed the potential of this format of game as a fully fledged experience that was appreciated for the adapted mechanics from the family of genres derived from adventure games, ease of control and storytelling possibilities.

Resumo

Criar ambientes de jogo seja em 2D ou em 3D é bastante laborioso e requer um nível alto de aptidão. Entretenha-se a possibilidade de usar video ou imagens 360^{o} como uma experiência interactiva dado que são um meio visual conhecido pela imersão espacial oferecida e um aumento da resposta emocional do utilizador ao conteúdo [EEW18]. Utilizar videos ou imagens 360^{o} possibilitará então que o talento dos criadores de jogos possa ser investido em outros aspetos cruciais do desenho do jogo como a narrativa, som e mecânicas.

Jogos de aventura fazem parte de um género heterogéneo que contém diferentes tipos de experiências e sub-géneros, desde aventuras em texto a jogos orientados a puzzle do estilo "point-and-click", uma experiência cinemática com escolhas de jogador ou até mesmo uma mistura de investigação de crime com drama de tribunal. Ou seja, jogos com interação mais simples e mais focada em narrativa. Sendo esta um campo de investigação bastante relevante no contexto de realidade virtual [RTGG17]. Para além disso as mecânicas de jogo não são baseadas em reflexos o que torna o género de "Aventura" um dos mais apropriados para integração com media visual em 360^{o}.

Esta dissertação apresenta uma framework que possibilitará a simplificação do processo criativo deste tipo de experiências, proporcionando aos criadores as ferramentas necessárias para desenvolver jogos de aventura completos, utilizando videos 360^{o} como o ambiente de jogo e jogáveis em realidade virtual. A framework foi devidamente implementada e testes de usabilidade foram conduzidos para validar a viabilidade de jogos de aventura baseados em videos 360^{o} tanto num foco de jogabilidade como de narrativa. Os resultados mostraram o potencial deste formato de jogo como uma experiência de jogo completa que foi apreciada pelos utilizadores tanto pelas mecânicas adaptadas da família de géneros derivados de jogos de aventura como pela usabilidade dos controlos do jogo e as possibilidades de narrativa.

Acknowledgements

I would like to thank my supervisors, Rui Rodrigues and Rui Nóbrega, as well as the people of GIG for their continued advice and for giving me the opportunity to see this project through till the end by providing all the support and resources I needed.

I want to express the utmost gratitude to my significant other, Maria de Abreu, my family and friends for their continued support and for the simple fact that they exist.

Last but not least a big thank you to António Baía Reis for essentially volunteering as the actor in the proof of concept game on such short notice without having time to practice and for still doing a great job regardless.

Francisco Pinho

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Abbreviations

HMD	Head-mounted display
NPC	Non player character
VR	Virtual Reality
FOV	Field of view
HUD	Heads-up display
PSR	Parasocial Relationship
GPU	Graphics Processing Unit
CPU	Central Processing Unit
UI	User Interface
GUI	Graphical User Interface
DOF	Degrees of Freedom
JSON	Javascript Object Notation
HTML	Hypertext Markup Language
DOM	Document Object Model
API	Application Programming Interface
ECS	Entity-Component-System

Chapter 1

Introduction

Virtual Reality is still in early stages of commercial viability and mainstream adoption, with the most popular of headsets having a user base of over 3 million users [Yee18]. But research has began as far back as 1968 with the creation of the first computer-connected HMD (head-mounted display)[VKP10].

Synonymous with VR is immersive 360^{o} video, as when it is experienced through an HMD it may increase the user's spatial immersion and potentially even their emotional response to the content [EEW18].

There is a high level of interest in exploring the artistic, educational and academic possibilities of VR. Zyda argued that in order for researchers in the field to have impact, they would have to focus their efforts on game research as it would affect not only the entertainment-centric video game industry but also "the government and corporate organizations that could benefit from the training, simulation, and education opportunities that serious games provide" [Zyd05] (p. 25).

Game development is more accessible than ever. Enterprise-level game engines like Unity[1] and Unreal[2] have been made freely available to the public, and massive storefronts such as Steam[3] and PlayStation Network[4] have opened the floodgates for independent developers to self-publish their games.

However there are still many technical barriers and challenges that game developers have to overcome if they want to create a presentable and commercially viable product. Game development remains a complex and multidisciplinary undertaking that requires a wide array of skills and in many cases a high degree of specialization (for example within the software development component of game making there is a need for trained individuals in the field of graphics, systems, artificial intelligence, optimization and user interface to name a few).

[1] Unity 3D, https://unity.com/, last access 2019

[2] Unreal Engine, https://www.unrealengine.com/en-US/, last access 2019

[3] Steam, https://store.steampowered.com/, last access 2019

[4] PSN, https://store.playstation.com, last access 2019

1.1 Motivation

Considering the possibility of using live action 360º videos as the game world it will be possible to reduce significantly the amount of work necessary to create a game, which includes animation, 3D modelling, 2D graphic design, graphics and rendering programming (scene creation, post processing...) and even a portion of the audio design.

Adventure games are the perfect fit for this idea as they typically feature limited interaction and focus on storytelling and non reflex-based mechanics like puzzle solving and exploration. There are only 3 DOF (degrees of freedom) in a static 360º video or image, associated with rotation of the head, meaning that action games would be severely limited under these conditions.

Therefore the focus of the dissertation will be to create a framework to test the concept of 360º video adventure games that will also streamline the design and implementation of many game mechanics common in adventure games such as dialogue trees and object interaction and other non genre specific features such as the insertion of virtual game objects into the game world. This would allow artists, storytellers and filmmakers with limited technical knowledge or even technologically oriented people with lower artistic prowess to create fully featured virtual reality adventure games.

Content could also be produced for educational purposes and there is a precedent for the use of adventure games and immersive virtual reality experiences for learning and training [JW97, ANVA99, Dic05, Dic11, FO15, RKM$^+$16]. Furthermore, the framework itself could be used as a tool for students developing projects in subjects involving game development, interactive narratives and even filmmaking.

Other applications would involve producing content using the framework for research on interactive narratives with 360º video or even miscellaneous topics such as parasocial relationships as discussed in section 2.1.4.

1.2 Research Questions

To guide the work, the following research questions are defined:

- What are well suited game mechanics to this format of adventure games?

There are many types of adventure games and not all of the mechanics will translate properly to this format. It will be necessary to analyze the core mechanics of adventure games and adapt those that fit the constraints of 360º video based adventure games.

- What would be an adequate VR control scheme for this type of game?

Taking into account the controllers available in common VR headsets, this question relates to finding a comfortable and intuitive control scheme for users.

- Do 360º video based adventure games have the potential to satisfy users both as a game and as a means of storytelling?

Finally, the intent behind this question is to assess the potential of interactive 360^{o} video based adventure games as a new type of adventure game through the two core strengths of the genre.

1.3 Research Goals

The framework's design will be informed by the following research goals:

- Assess the viability and the advantages of creating 360 video based adventure games
- Pinpoint the core adventure game mechanics that should be integrated in 360 video based adventure games
- Assess the viability of 360 video based adventure games as an entertainment and storytelling medium
- Reach an adequate technological solution to streamline and facilitate the design and development of 360 video based adventure games

As to achieve the described goals usability tests to qualitatively measure end user engagement and immersion will be conducted. With the main goal being investigating the viability of 360^{o} video based adventure games. The evaluation of the research is described in more detail in chapter 4.

1.4 Contributions

The contributions of this dissertation include the creation of a low-coding framework that will allow people with cursory coding skills to create fully fledged adventure games with 360^{o} videos as the game environments, thus streamlining the game development process considerably.

Additionally a proof of concept game will be created and evaluated to test the usability of the game systems and player engagement of interactive 360^{o} video based adventure games in virtual reality. Through those usability tests it is also expected that some game design guidelines will be extracted for the creation of interactive 360^{o} video based adventure games.

1.5 Document Structure

The remainder of this document will include four more chapters. The next chapter is an overview of the state of the art in various relevant topics to this dissertation such as the adventure game genre, virtual reality and interactive 360^{o} video.

The third chapter focuses on the conception and development of the framework followed by a fourth chapter that details the usability tests that were done to evaluate 360 video based adventure games with content created with the framework.

To wrap the whole document the fifth chapter features conclusions regarding the whole dissertation project including answers to the research questions and future work that can be done.

Chapter 2

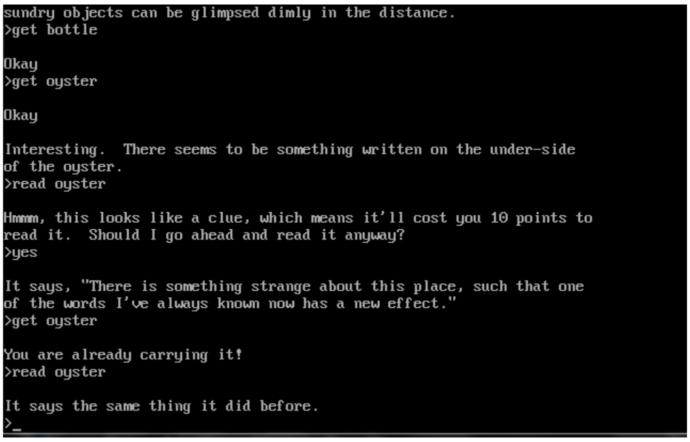
State of the Art

The main focus of this dissertation is looking into the use of 360^{o} videos and immersive environments as means for creating adventure games, and therefore the literature review will be focused in these key topics: adventure games, virtual reality and immersive 360^{o} content.

2.1 Adventure Games

There have been several attempts at defining what is an Adventure game. Fernández Vara defined the genre succinctly as one of story-driven games with particular focus on exploration and puzzle solving, where players interact with the game primarily through object manipulation and by navigating the game world itself. Puzzles represent the challenges of the game and are integrated into the fictional world [Fer09] (p.13).

The first adventure game can be traced back to 1976 with the release of "Colossal Cave Adventure" [And15], also known simply as "Adventure". It was the first known game to have incorporated a form of interactive fiction where players would try to progress through the game's story whose focus was on exploring an unknown cave by issuing text commands (Figure 2.1) thus birthing the first generation of adventure games in the form of so called "Text Adventures".



```
sundry objects can be glimpsed dimly in the distance.
>get bottle

Okay
>get oyster

Okay

Interesting.  There seems to be something written on the under-side
of the oyster.
>read oyster

Hmmm, this looks like a clue, which means it'll cost you 10 points to
read it.  Should I go ahead and read it anyway?
>yes

It says, "There is something strange about this place, such that one
of the words I've always known now has a new effect."
>get oyster

You are already carrying it!
>read oyster

It says the same thing it did before.
>_
```

Figure 2.1: "Colossal Cave Adventure" - First Text Adventure (1976)

"Adventure" by Crowder and Woods would go on to inspire and influence several innovators and pioneers of adventure games, namely the team over at Infocom[1] that developed the "Zork" series, games with more sophisticated text parsing that supported commands larger than two words. Another example of more sophisticated parsing can be found in the 1982 ZX Spectrum[2] game "The Hobbit" as it allowed players to insert entire sentences, fully complete with pronouns, adverbs, punctuation and prepositions. Also building on top of "Adventure" and subsequent efforts, Roberta and Ken Williams went on to create a new variation of adventure game, the graphical text adventure (Figure 2.2) with the release of "Mystery House" [And15] and founded one of the most prolific and influential game companies for the genre, Sierra Online[3].

Years following the release of "Mystery House", The Williams would disrupt the genre again with the release of "King's Quest" in 1984 with the development of a new game engine, completely revamping the way players interfaced with the game. The new engine allowed the player character to be directly controllable with the arrow keys in the keyboard, players could not only move from room to room but also within an individual room. However some of the inputs still needed to be typed out, particularly the ones related to interactive objects in the game world [FV08].

The 1990's would end up being the most prolific decade for adventure games, both in terms of popularity but also in innovation and it were the titles released in this era that solidified many of the genre's tropes and presentation style. In an analysis on the shifting levels of abstraction in adventure games, Fernández Vara argues that contrary to most other video game genres where the complexity of interaction has been going up, adventure games were instead being more and more streamlined as they evolved as to make their interfaces clearer and easier to use to avoid player frustration [FV11].

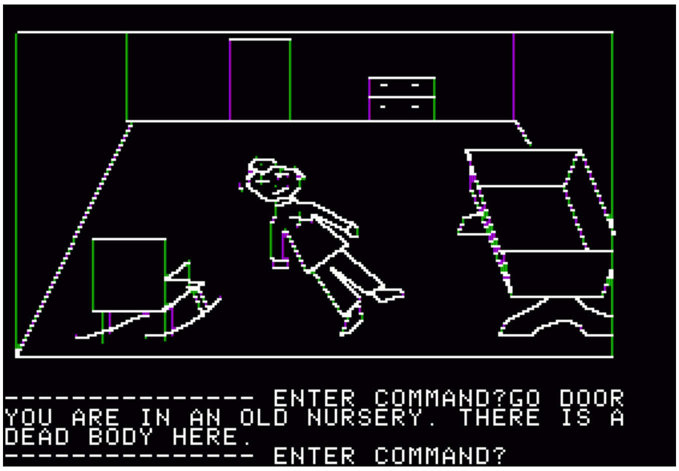
To that effect, following the introduction of graphical text adventures, LucasArts[4] came out

[1] Infocom, https://en.wikipedia.org/wiki/Infocom, last access 2019

[2] https://en.wikipedia.org/wiki/ZX$_{spectrum}$

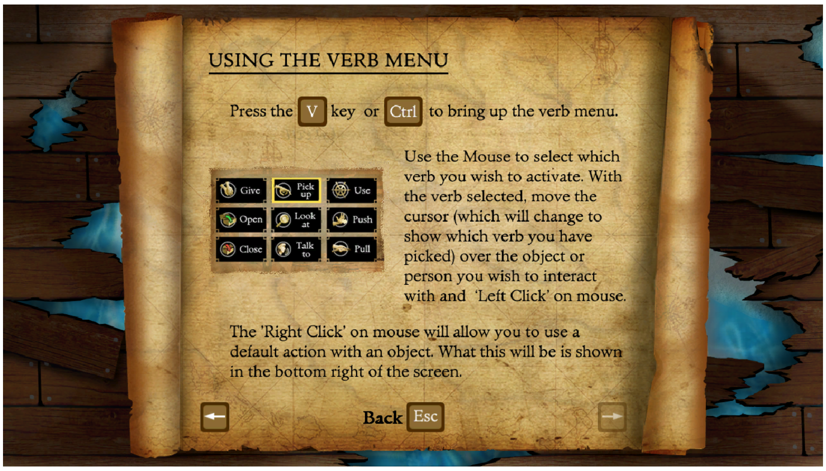
[3] Sierra, https://en.wikipedia.org/wiki/Sierra_Entertainment, last access 2019

[4] LucasArts, https://en.wikipedia.org/wiki/LucasArts, last access 2019



```
--------------- ENTER COMMAND?GO DOOR
YOU ARE IN AN OLD NURSERY. THERE IS A
DEAD BODY HERE.
--------------- ENTER COMMAND?
```

Figure 2.2: "Mystery House" - First Graphical Adventure (1980)



USING THE VERB MENU

Press the V key or Ctrl to bring up the verb menu.

Use the Mouse to select which verb you wish to activate. With the verb selected, move the cursor (which will change to show which verb you have picked) over the object or person you wish to interact with and 'Left Click' on mouse.

The 'Right Click' on mouse will allow you to use a default action with an object. What this will be is shown in the bottom right of the screen.

Figure 2.3: All menu actions, usage guide in "The Secret of Monkey Island Special Edition" (2009), updated re-release of original (1991)

with a new variety of adventure games known as the graphic adventures or point-and-click games as they could be fully completed with mouse clicks as the only input method. These games ditched the text commands for a purely graphical interface with several commands in the form of verbs (Figure 2.3). There were no additional or hidden commands besides the ones presented in the menu. All interactions then took the form of [verb] -> [object] with one example interaction being [Talk to] -> [Little green man]. Another common feature in graphic adventures is an inventory of objects visible in the interface. The inventory not only contained objects collected throughout the game to be used on other objects in the environments but also to combine with other objects within the inventory itself to create an object with new characteristics. This new object in turn can be used to fix a problem or to further the completion of a puzzle.

Dialogue systems were also made explicit through information presented in dialogue boxes instead of the player having to guess the proper text commands to initiate or to build upon an existing conversation. Players could now choose between a different number of pre-written dialogue options to get all the information possible from the game character with a system known as a "dialogue tree". Options in the "dialogue tree" could be choices that influenced the advancement of the story or simply topics of conversation that could be selected one at a time. Notable graphic adventures are games like "The Secret of Monkey Island", "Full Throttle" and "Grim Fandango" all developed by LucasArts.

Some games released in the same decade, notably Cyan Worlds's "Myst" in 1993 branched into different design philosophies by removing the typical graphical interface entirely. Choosing instead to fill the entire screen space of the player with the environments of the game. The camera perspective shifted to first person as see in Figure 2.4 and there were no characters in the world to interact with in contrast to the LucasArts games previously mentioned. Instead players would use mouse clicks to do all the possible actions in the game by clicking directly on the relevant entity of the game scene without selecting a menu command first, a form of contextual interaction [FV08]. This type of interaction design avoids the need of error messages such as "you can't do

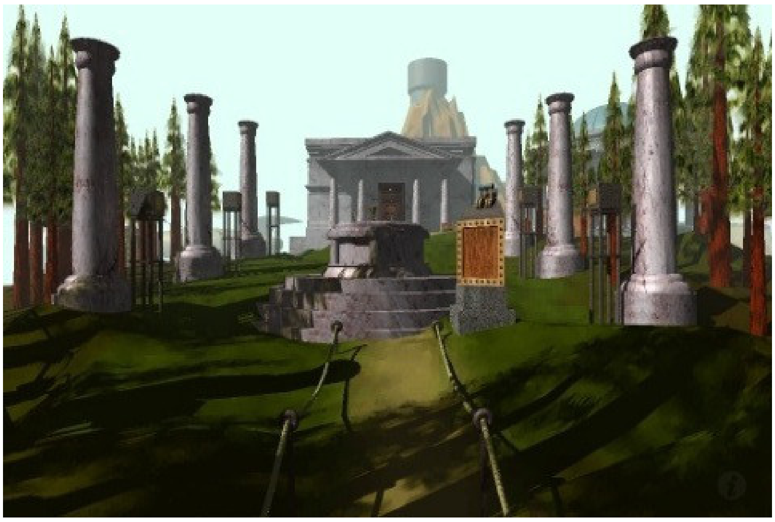


Figure 2.4: First person perspective in "Myst" (1993)

that" [FV08] because objects in the game world are directly manipulated [Shn83] by the player (for example clicking buttons in the game world or dragging objects around).

At present adventure games have fully embraced contextual interaction with the environments versus menu-based commands. With context-sensitive systems players can try to use or interact with objects directly in the environment and receive immediate feedback on the effect of their actions. For example in the 2009 video game "Machinarium" the player can select a cone object from his inventory and click on a barrel full of paint in the game scene. By doing so an animation where the character dips the cone in the barrel will play out and as a result the cone will come out with a different color. This is one of the steps needed to solve one of the first puzzles in the game. To elaborate further, why should a player have to select the action verb "Talk to" from a menu and then select the character in the game world to initiate the action when they can simply click on the character and the game will correctly assume what the player can do in that interaction or context. The idea of contextual interactions is to give the game engine responsibility over deciding what to do with player input based on the semantics of the object that was targeted by said input.

Additionally, 3D movement exploration of the environments has also been prevalent especially in the VR space and games whose story presentation is more cinematic and feature a lesser emphasis on puzzle solving in general. Another important characteristic of contemporary adventure games is that puzzles are deeply integrated in the context of the narrative instead of challenges that the player needs to overcome in order to get to the next story event. The narrative informs the design of the puzzles.

Modern adventure games have embraced dialogue trees and player choice (Figure 2.5) that lead to changes in the narrative (Figure 2.6a) and in some cases even entirely different narrative paths akin to "choose your own adventure" books (Figure 2.6b), in short they are branching narratives. Notable examples of this can be found in games like "9 Hours 9 Persons 9 Doors" by Spike

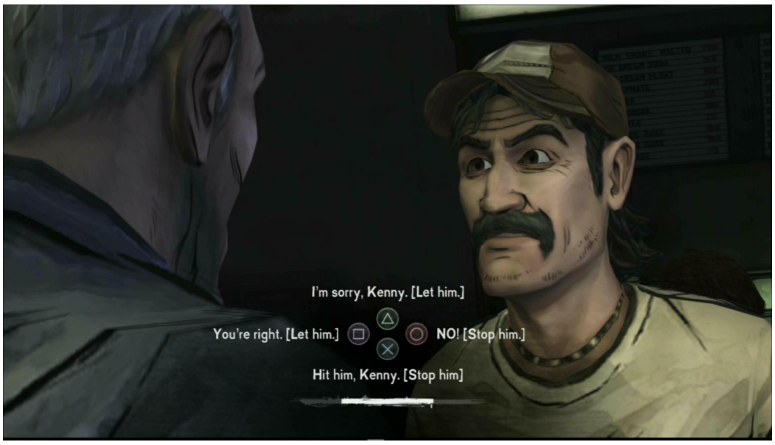


Figure 2.5: Player choice in "The Walking Dead" (2012)

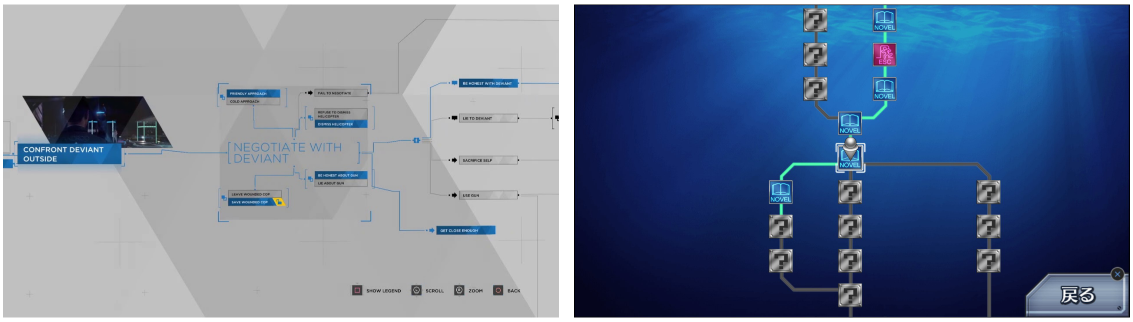
Chunsoft[5], "The Walking Dead" by Telltale games[6] and "Detroit: Become Human" by Quantic Dream[7].

Some contemporary adventure games have also mixed with other genres. Examples of this are series like "Ace Attorney" (2001-2017) and "Danganronpa" (2010-2017), both having embraced visual novel elements to tell compelling stories whilst integrating gameplay mechanics traditional to adventure games. These mechanics include point-and-click investigation sections in the case of "Ace Attorney", and a mix of first person 3D environment exploration and point-and-click rooms in the case of "Danganronpa" (not only for investigations but also for socialization with NPC's) (Figure 2.7). Both series feature puzzles in the form of murder trials (Figure 2.8) that are heavily contextualized within the narrative.

[5] Spike Chunsoft, https://www.spike-chunsoft.com/, last access 2019

[6] Telltale Games, https://telltale.com/, last access 2019

[7] Quantic Dream, http://www.quanticdream.com/en/, last access 2019



(a) Player choice and consequence chart in "Detroit: Become Human" (2018)

(b) Story flowchart in "9 Hours 9 Persons 9 Doors" (2009)

Figure 2.6: Examples of Branching Narrative

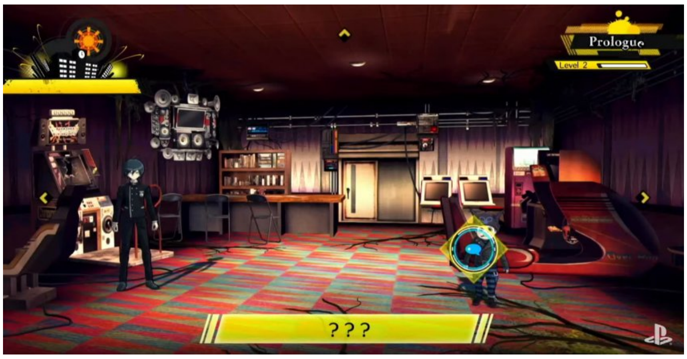


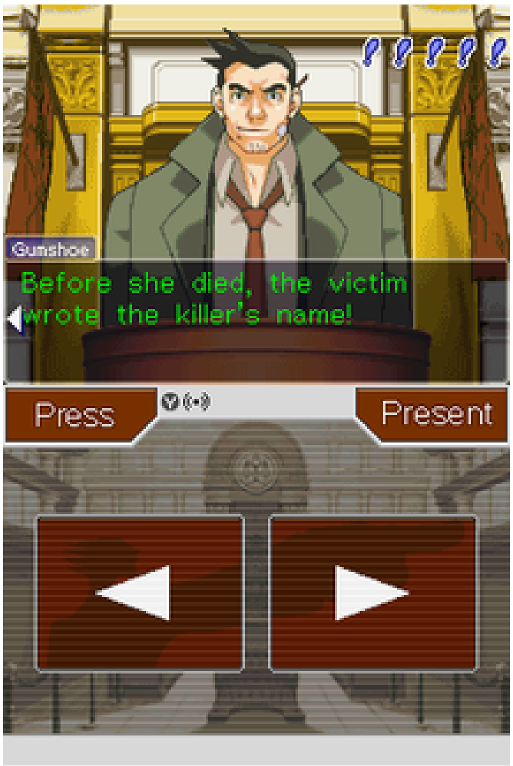
Figure 2.7: Point-and-click Screen Example in "Dangaronpa V3" (2017)

Currently, with the renaissance of Virtual Reality there is an undergoing effort by a few dedicated developers in figuring out how to evolve and integrate the genre into this medium. Notable examples can be found in "Obduction" (2016) from the creators of "Myst" and in "I Expect You to Die" (2016). These games take advantage of virtual reality by exploiting the increased sense of presence produced in players and by moving towards a true to life style of direct manipulation made possible by tracking of the hands through dedicated input devices.

Despite recent design trends that gravitate towards more cinematic games, puzzle and gameplay centric adventure games are still being made and some of them have been quite successful. Notably "Machinarium" with fully point-and-click based interaction and a minimalistic storyline that unfolded as players go from puzzle room to puzzle room.

Furthermore, 2018's critically acclaimed "Return of the Obra Dinn" puts the player in the shoes of an insurance agent that must investigate the return of a ship - "the Obra Dinn" - having disappeared five years earlier in the game world's fiction. Players have to uncover the fates of every member in the crew through deductive reasoning with hints based on flashbacks of several incidents that happened in the ship. The entire game is a large logic puzzle comprised of over 60 smaller puzzles.

Several old games have been remastered and updated to be brought over on modern platforms. "Myst" has received multiple releases and updates throughout the years, adding features like the option to play with full 3D locomotion versus navigation in point-and-click style. Another example is "Secret of Monkey Island Special Edition" (2009) that featured a complete graphical overhaul, full voice-over, re-mastered soundtrack and other improvements like a new hint system and revamped controls. Therefore there is still an audience for these titles even if in a smaller scale compared to the peak of popularity the genre enjoyed in days past.



Before she died, the victim wrote the killer's name!

Figure 2.8: Trial Mechanics in "Ace Attorney" (2001)

2.1.1 Design Challenges and Variations

As previously established, adventure games are varied in their structure and style. In this section their most common formats will be described in more detail alongside design problems associated with adventure games. These problems may be specific to some individual format of adventure game and not general to the entire genre.

General Design Challenges

- *Difficulty/Sign-Posting/Hint Systems* - A situation where a player gets stuck on a puzzle and there is no in-game hint system to nudge him into the right direction could lead to increased levels of frustration and having the player look up the solution online. In extreme cases that player might quit the game entirely. On the other hand, if the game is too generous with its hints there is a risk of making the game too easy and less engaging as a result. Balance is key when designing the challenges of the game. One strategy that designers employ is by using "sign-posting". It can be described as giving hints that are organic to the game's world and fiction as they are typically given to the player in the form of NPC dialogue or through interactions with objects [Bro15]. There are also strategies that involve giving players more explicit and direct hints. These strategies are usually available as a last resort to players through a layer external to that of the game's internal world. For example an actual help button in the menu interface or built-in walkthroughs [8]. There are several ways to work around this issue and game designers have come up with a variety of hint systems. These will be discussed in more detail in section 2.1.3.

- *Puzzle Integration* - Appealing puzzles have to be fun and challenging but puzzle contextualization is also very important. Players will appreciate a puzzle that has a direct connection to the narrative. An example is finding contradictions during cross examinations in "Ace Attorney". It may be difficult to design puzzles that relate to certain story contexts but designers should take both story context and puzzle design into account and design them hand in hand for a more satisfying experience. In a video series about game design, Game Maker's Toolkit, Mark Brown remarks on the design of point-and-click puzzles that one of the fundamental rules of design should be providing clear goals to the player. By establishing the narrative context with short and long term goals for the main character, players will know which puzzles they have to solve and why they need to solve them. This grants players motivation to engage with the challenges of the game [Bro15].

- *Inconsistent Logic* - In the same video essay, Mark Brown lists a few notable examples of inconsistent or nonsensical logic in adventure game puzzles. Notably in the recent "Broken Age: Act II" (2015) where a hand creature in the game world is obsessed with shoes but using the main character's shoes on it has no effect and in "Gabriel Knight 3" (1999) where the player is expected to use masking tape and maple syrup to turn cat hair into a moustache to disguise the main character as another character that does not even have a moustache [Bro15]. In short, if the game's internal rules are not consistent, players will feel that the challenge is unfair and as such unworthy of their time. As such this should be one of the main concerns on a designer's mind when making puzzles for adventure games.

- *Low Replayability* - Adventure games primarily feature a structure alternating between narrative focused scenes and player controlled segments with exploration and puzzle solving. Both the puzzles and the narrative are designed in a static manner. This means that there is little incentive to replay these games as players who finish them once will already know the contents of both the storyline and the solutions to the puzzles. 1997's "Blade Runner" tried to tackle this issue by making certain elements of the narrative itself random. In the game's story players are tasked with investigating which characters are "replicants"[9]. The game randomly determines which characters will be "replicants" upon initialization. Additionally, research has been made into procedural generation of puzzles for adventure games. One of the approaches by Fernandéz and Alec was the development of a tool that would allow designers to create puzzle maps, in essence structural patterns. After defining possible patterns, puzzles would be generated through an algorithm that would lookup possible entities in a database whose relationships and attributes were compatible with the previously defined pattern. [FVT12]. A different approach by Isaac and Mark focused on the idea of

[8] A walkthrough is a step-by-step guide to completing the levels, puzzles or challenges in a game

[9] A replicant is a possibly dangerous robot posing as a human in the Blade Runner universe

game items viewed from the perspective of "smart terrain" [DN12]. Through this lens, game objects are characterized by their physical properties, what problems they can solve, their behaviour and reactions to external forces like temperature or the presence of smoke. To be brief, it is an approach of generating valid cause and effect based puzzles by making use of objects with properties that may affect each other. This implies the design of a physics simulation system in contrast to typically hard coded approaches to object relationships in most adventure games. With a physics simulation it is possible to bring to life the properties of objects. An example of this would be making a paper object burn if the temperature is risen above a certain threshold or in contact with a fire object. While an interesting problem, it is beyond the scope of this dissertation and will not be addressed in the design of the framework.

Text Adventures

Text adventures are games with no graphics, instead they operate in a pure text interface. Players input text commands to navigate the environment, solve puzzles, talk to NPC's (non-player characters), make choices and advance through the story.

There are many issues related to designing a good text adventure. Some of these problems are simply innate to the limitations of the format:

- *Text parsing* - Designers need to develop a sophisticated natural language processing engine or be prepared for multiple syntactic and semantic variations of possible text commands if they want players to have more choices and freedom in the way they interact. If the text parsing capabilities are too limited the game will be frustrating due to excessive trial-and-error and less immersive as a consequence [FV08]. As a way to mitigate this issue, designers provide hints as to what the player is supposed to input next in the responses to the player and the game's own narration.

- *Error Handling* - When the player input is unrecognizable by the text parsing engine there needs to be feedback. This is a big design issue because if a game keeps responding with syntactic error messages, even to commands that may make sense to the player but that the system is not prepared for, will lead to players being taken out of the game as they are constantly reminded that they are not interfacing with a living breathing world but with limited software instead [FV08].

- *Indirect Manipulation* - Players do not control their character or interact with the game's environments and characters directly, they have to input text commands for everything, an unintuitive and indirect way of interaction [FV08]. In 1983, Shneiderman defined direct manipulation as a capability of systems that offer a satisfying user experience by providing direct and instantaneous feedback to user actions on visible components of the interface, making applications more transparent in the process as the users do not have to think about the inner workings of the software to properly interact with it [Shn83] . Adventure games evolved from indirect manipulation to increasing levels of direct manipulation, culminating in games that try to mimic real life handling of objects with more sophisticated means of interaction such as gesture and motion controls.

Graphical Text Adventures

These are adventure games with graphics but they are still controlled through text inputs and at most a few limited menu options. These menu options never represent the entire subset of possible commands in each situation. Essentially they are an extension of text adventures.

The design challenges carry over from text adventures but perhaps to a lesser degree. Since the game has a graphical representation of what is happening, it is easier for the player to visualize the consequences of their actions thus bringing the game closer to the direct manipulation concept. The existence of a few menu options also lessens the burden from the players by giving them direct access to a few of the commands that the game supports.

Point-and-click/Graphical Adventures

Games that can be controlled exclusively with mouse clicks, a movable cursor and button presses if played with a console game-pad or touch inputs (adventure games on the Nintendo DS handheld system for example). They may have a first person perspective or some form of non first person perspective focused on the main character, commonly 2D side views.

Additionally these games have fully featured dialogue systems in the form of "dialogue trees" as previously mentioned. It is not uncommon for these games to be fully or partially voice acted.

3D Adventure Games

These games can be presented in a first or third person camera perspective. They feature full locomotion systems by allowing the player to explore the environments freely by directly controlling the character through controller analog sticks or computer "WASD" [10] controls.

The requirements for designing a world in this format are more complex as there is a need to support physics, collision detection and advanced animation systems to make the world believable and properly interactive. Therefore there is a need for the production of a high volume of assets such as 3D models, animations and more.

Virtual Reality Adventure Games

VR Adventure games are typically an extension of 3D Adventure games. The component of virtual reality gives the player a sense of presence in the game world and thus an enhanced level of immersion. Developers can also make use of motion controllers and the 6 degrees of freedom (rotational and positional tracking) provided in high end VR headsets like the Oculus Rift, HTC Vive

[10] "WASD" refers to the traditional way of controlling characters on PC, W moves the character forward, S backwards, A to the left and D to the right

and Playstation VR to achieve a greater level of interaction. To elaborate, with these controllers it is possible to simulate controlling the character's limbs and interacting with objects in a more complex and true to life way. Case in point, interactions such as reaching out into a drawer and pulling the drawer open or grabbing a key and manually inserting the key in the keyhole are made possible through one to one translation of movements in real life into movements in the game.

Movement can be achieved through traditional controls found in 3D adventure games but at the risk of causing motion sickness to some players. As an alternative these games implement comfort functionalities like teleportation [11] and snap turning [12].

With virtual reality comes a new set of design challenges. Namely in figuring out the ideal control scheme, offering multiple locomotion options for users with varying degrees of immunity to motion sickness[13]. Knowing how to structure the game's environments to make relevant content naturally attractive to players is also a problem since there is a bigger amount of information to process in a 360-degree display versus a flat one.

Finally, it is also possible to create more traditional experiences with static camera viewpoints (no movement or positional tracking) by replacing the mouse as the control interface with a laser pointer in VR. This is the style of game that will be the target of the projected framework. Many design challenges inherent to this style remain from those of adventure games in general but with the introduction of the VR element and the use of 360^{o} video as the environments of the game world new design challenges arise to be discussed in section 2.2.

2.1.2 Spin-off Genres

The adventure game has birthed sub genres. It may be debatable whether some of these sub genres are still considered adventure games but that debate is not very relevant in the context of this dissertation. By implementing the core game mechanics necessary to make an adventure game most of the listed spin-off genres will be possible creations in the projected framework. Adventure games can also be multi-genre games by mixing elements of these spin-off genres or the other way around.

Visual Novel

Heavily narrative focused games with very little interaction and commonly with branching paths. Player input is only necessary to advance the dialogue and to make the occasional choice. Visually, game scenes are composed of a text, characters (typically in sprite forms) and 2D images as the background environment.

[11] Teleportation movement is achieved by moving a cursor around the 3D environments as one would a game character and then click a button, mouse or keyboard key to teleport to the current location of the cursor

[12] Snap turning is a technique that implies rotating the camera perspective in the 360-degree environments instantly in incremental angles instead of rotating the camera perspective smoothly

[13] May also be referred as simulator sickness in the context of virtual reality

Escape-the-Room

Players need to solve a series of puzzles to escape from imprisonment. The degree of focus in narrative may vary from title to title. The types of puzzles can be very similar to that of adventure games. It is a genre that differentiates itself through its setting and context more than anything else.

Interactive Drama

Games mostly composed of a series of cinematics as opposed to a more literary style of narration commonplace in visual novels. They may feature a few interactive moments in between story scenes where players may walk around the environments and interact in basic ways. Another common form of interaction in this genre is player choice in dialogue that may have consequences in the narrative.

Walking Simulator/First Person Exploration

Games where first person exploration and examination of game environments comprise the core mechanics. The narrative will unfold as players explore the environment either through traditional cutscenes or more commonly by story content integrated into objects in the environment. As an example players may find a diary in a house that belongs to a character in the story and by interacting with it they will get a piece of story content embedded in the object itself.

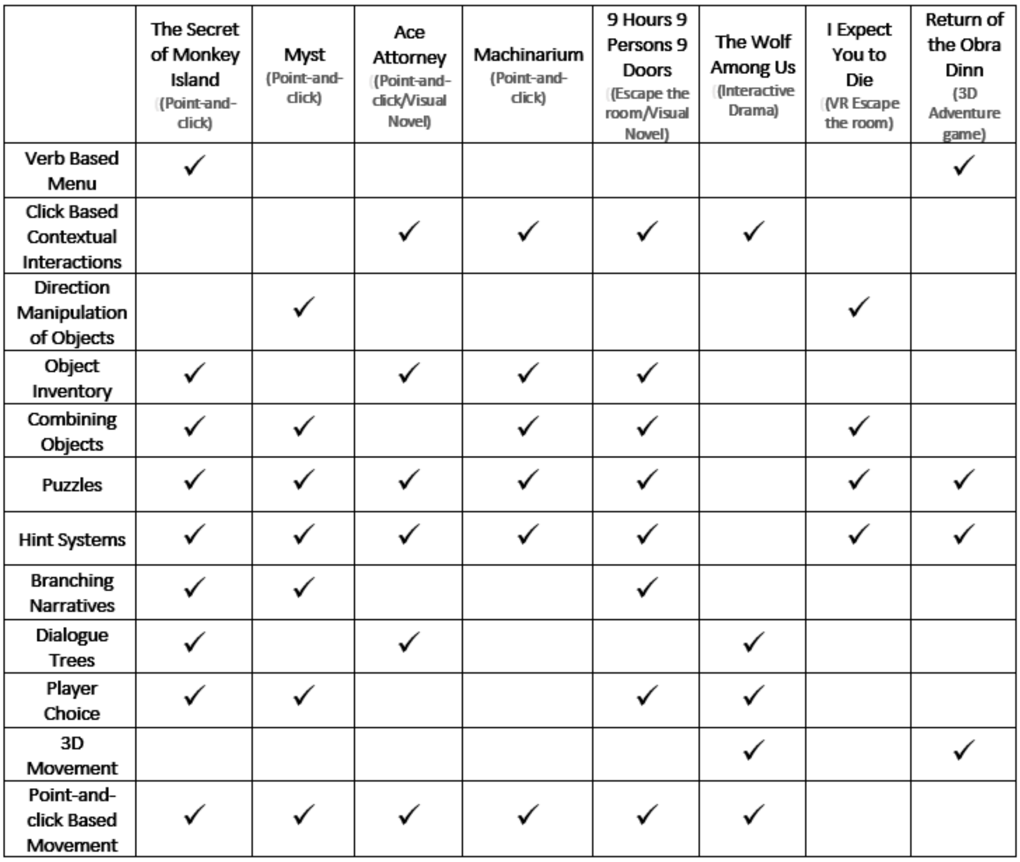
Full Motion Video

Games composed entirely of pre-recorded video (live action or animation). Player input is done through a dedicated game interface or prompts to make choices or quick time events [14] during an ongoing video. The game world is expressed entirely through the pre-recorded video.

2.1.3 Core Game Mechanics

The core mechanics of the adventure game are those that are found across most games in the genre, as visible in Figure 2.9 and while some games may have unique mechanics of their own these are the mechanics that define the genre and that are most prevalent in its DNA. These mechanics will be supported in the projected framework in some way as they have several variants. Mechanics like 3D exploration of environments and physics-based object manipulation for example are not appropriate given the limitations of the static environments of a recorded 360° video.

[14] Quick Time Events are timed prompts for a specific button or keyboard press



	The Secret of Monkey Island (Point-and-click)	Myst (Point-and-click)	Ace Attorney (Point-and-click/Visual Novel)	Machinarium (Point-and-click)	9 Hours 9 Persons 9 Doors (Escape the room/Visual Novel)	The Wolf Among Us (Interactive Drama)	I Expect You to Die (VR Escape the room)	Return of the Obra Dinn (3D Adventure game)
Verb Based Menu	✓							✓
Click Based Contextual Interactions			✓	✓	✓	✓		
Direction Manipulation of Objects		✓					✓	
Object Inventory	✓		✓	✓	✓			
Combining Objects	✓	✓		✓	✓		✓	
Puzzles	✓	✓	✓	✓	✓		✓	✓
Hint Systems	✓	✓	✓	✓	✓		✓	✓
Branching Narratives	✓	✓			✓			
Dialogue Trees	✓		✓			✓		
Player Choice	✓	✓			✓	✓		
3D Movement						✓		✓
Point-and-click Based Movement	✓	✓	✓	✓	✓	✓		

Figure 2.9: Common Mechanics in Adventure Games - Mechanics matrix in hallmark titles. Games are on the top row and the mechanics are featured in the first column.

Puzzles

Puzzles are usually the main gameplay mechanic of an adventure game. They constitute the challenges that the player has to overcome to experience the entire narrative and see the game to its completion.

Puzzles are thinking challenges with one or multiple solutions and they are designed to be fun to solve and typically require out of the box thinking. A puzzle needs to be interesting to be worthwhile. Easy puzzles are uninteresting and puzzles that are too difficult from the outset will be discouraging [Kim06].

Some puzzles require solving a multitude of smaller puzzles while others are standalone. Each step in a bigger puzzle can be a different variety of puzzle in itself.

In adventure games puzzles can take a multitude of formats and contexts. To illustrate the possibilities and varieties here are a few examples:

- Choose the correct piece of evidence that will show a contradiction in a witness' statement
- Decode a mathematical operation that is written in hexadecimal into a decimal format in order to use the result of the operation as a code to unlock a safe

- Achieve a checkmate state from a chess board in a given state in X amount of plays
- Find a way to unfreeze a door handle with objects present in the environment to leave the room
- Find a way to turn cat hair into a moustache to disguise yourself as another character
- Give the correct item to a certain character that is blocking the entrance of a building
- Deduce a character's cause of death and corresponding murder weapon

Hint Systems

Hint systems are complementary systems to the puzzles/challenges present in the game. Such systems can be classified as diegetic or non-diegetic [Gal06]. Diegetic hint systems are those that are organically integrated in the game world and its fiction [Gal06]. One such example is in the game "9 Hours 9 Persons 9 Doors" (2009), in the escape-the-room sections where a companion character will give an increasing number of suggestions as the player fails at reaching the correct solution to a puzzle. Non-diegetic systems are those that are external to the game's fiction and that the in-game characters cannot perceive [Gal06]. A game's HUD [15] is an example of a non-diegetic system [ICK+15].

"Machinarium" has two non diegetic hint systems. One of them is a button that will give the player a more subtle hint about one important step to solve the current puzzle, the hint is portrayed through a short animation of the in-game character executing an action. The second system is another button that will give players access to a in-game walkthrough with all the steps necessary to solve the puzzle. The latter more overt hint system is not accessible before the player goes through a side scrolling shooting mini game as a way to discourage players from overusing it. Basically the second system is designed to be a truly last resort help to the player with the intent of making it unnecessary to browse the internet for solutions.

Exploration

Exploration can be achieved through direct manipulation of the player character's movement or a point-and-click interface to move the character around or shift the camera perspective into different portions of the screen in the case of a first person perspective.

Exploration also implies the existence of multiple environments or a single large scale and interconnected environment.

There may be loading screen transitions between different portions or the environment or seamless transitions depending on how the game is implemented.

[15] A heads-up display (HUD) is essentially a GUI with elements such as a map to help with player's navigation or buttons that can be clicked on to access some menu

Object Examination

This mechanic refers to interacting with an object in the environment. Depending on the type of game this can be achieved by directly clicking on the object in the case of point-and-click style games or by moving the character to the desired object and then inputting the correct button to trigger the interaction. There are two types of outcomes that can result from this interaction, as these outcomes are not mutually exclusive both can happen in sequence:

- The player is given a dialogue blurb containing the character's impressions, thoughts or description of the examined object. If the examined object is part of a future step in a puzzle the character might make a statement about how they have no use for that object at that moment as a way to signpost future utility of said object.

- The object will be added to the player's inventory

Inventory

Players may collect objects in the environment and an inventory system is needed to manage the collection of objects available at any given time.

Use Objects

Objects present in the inventory can be used on the environment by being given to some NPC or used on other objects to produce some effect like using a key to unlock a door or using a lighter to burn a paper. In the case of detective or murder mystery games objects can be used as evidence of some deduction.

Combine Objects

Objects in the inventory can be combined to create a new object that will have some use as described above.

Dialogue Trees and Player Choice

Dialogue is usually presented in the form of a dialogue tree. On gamasutra[16], a blog about video game development by actual developers, Alexander Freed detailed the process of a dialogue tree as such [Fre14]:

1. The NPC says something.

2. The Player is presented with a limited set of options indicating ways to respond.

3. The Player chooses a response

[16] Gamasutra, https://www.gamasutra.com/, last access 2019

4. The NPC replies according to the Player's chosen response, this will result in the conversation branching into a new direction

5. The Player is presented with a new set of response options distinct from the previous set, response options can be repeated from the previous set

This is generally an apt description of how these systems work but sometimes the choices are presented before the NPC even says anything, as conversation starters. Choices may be inconsequential if they are a simple choice of the next topic of conversation or consequential in a variety of non mutually exclusive ways:

- Player choice may define personality traits of the controlled character

- Player choice may affect the way other NPCs treat and react to the player character

- Player choice may affect the plot (branching narratives)

2.1.4 Narrative and Storytelling

There are several theories and lenses through which to analyze and discuss the nature and structure of narrative. As a useful guiding light for this discussion structuralist theory will be used as a basis. In short it is a theory that proposes that narrative is divided into two distinct parts, the story and its discourse [AL04]. Chatman succinctly describes the story as the "what" in the narrative [Cha78]. Essentially the main events that comprise the plot, characters and the setting. The second part - the discourse - is the "how" [Cha78]. It is the means of expression through which the story is told (verbally or through cinematics for example).

Another important concept when talking about interactive storytelling is that of emergent narratives [Ayl99]. They can be described as narratives that result from players interacting with the game world through simple game mechanics.

In the context of an adventure game, an example can be found in "I Expect You to Die", a virtual reality escape-the-room game where most of the story is emergent. In one scenario players are put in the shoes of a spy that needs to hijack a car inside a plane and send it flying. There are multiple ways to solve each step in the puzzle and emergent narrative is created through the actions that the player enacted at each step of the process. While one player might have constructed the narrative "I defused the bomb inside the car by smashing a champagne bottle and cutting the wires" another player might have constructed another alternative such as "I found a switchblade in the car and used it to defuse the bomb by cutting the wires".

In adventure games the story can essentially fall into two different categories structure wise: Non-branching and branching narratives.

Non-branching Narratives

These may include linear or non-linear scenarios. In this case linearity refers to the chronological order of the story events. Linear narratives are those that follow a rigid sequence of events from beginning to end with no deviations in the chronological events of the story.

Non-linear narratives may feature out of order sequence of events. A technique that is commonly used to make a non linear story is to begin the events "in medias res" and then have chronological flashbacks to fill in the gaps of the backstory.

Branching Narratives

Branching narratives feature alternative story paths with branches resulting from user choice. Therefore they can be non-linear as users navigate back and forth between the chronology in the story. These narratives may diverge and converge at any point but the defining characteristic is that they diverge at all. Typically in branching systems alternative paths are instances and variations of the same base story [LA05].

Parasocial Relationships

An interesting phenomena in the realm of psychology that could be applied to enhance storytelling in the creation of fictional characters is that of parasocial relationships (PSR) with virtual entities or personas [17]. The concept of parasocial interaction was first introduced by Horton and Wohl as a one-sided psychological relationship created by an audience in relation to the mediatic performers on mass media. It is described as an illusory relationship that makes audiences experience feelings of friendship and of an existing authentic reciprocal relationship [HW56].

Research into this phenomena has extended to interactive and virtual mediums [Har08]. Particularly in the realm of video games there have been studies and surveys about the formation of PSRs with the player's own avatar [JP09], the player character and non-player characters [Kav12]. Dating simulators, a subset of adventure games and visual novels were also examined to determine the effect of their consumption and how it could translate into idealized romantic beliefs and the development of parasocial relationships with the player's avatar and virtual characters [SF16].

Determining the effects of using an immersive live action 360º video game world as a setting for such games could be an interesting branch of research. To be brief, if and how using immersive environments or real actors to portray characters could affect the development of PSRs with virtual characters as players would be experiencing their presence in a closer representation of the real world. This topic is out of this dissertation's scope but the proposed framework could certainly facilitate research in this field as a tool to create the necessary content.

[17] personas are the social mask or characters expressed by the likes of celebrities, tv hosts and influencers to an audience

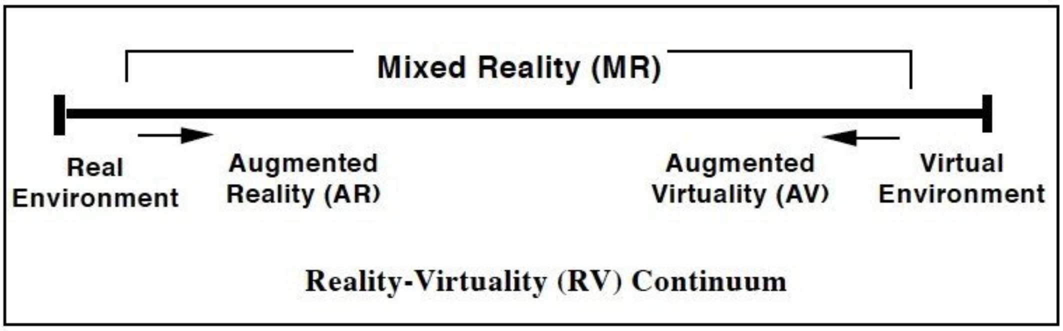


Figure 2.10: Reality-Virtuality Continuum [MK94]

2.2 Virtual Reality and 360-Degree Video

In 1994 Milgram et al. described the concept of the reality-virtuality continuum as a spectrum that describes experiences through their pure or mixed use of virtual environments or the real world [MK94]. An approach of including real world content in virtual reality games is one proposed by Valente et al., the concept of "Live-action Virtual Reality Games" where the virtual is constructed by using the real world architecture as the basic geometry. The idea is that players would interact with real world "smart objects" as a way to produce inputs in the game. It is through the use of sensors and tracking technology that objects in the real world could be mapped into 3D objects in the virtual environment and function as "smart objects" [VCSF16] by reacting in sync with user interactions on their respective real world counterparts.

360^{o} video goes hand in hand with virtual reality headsets. They provide the optimal way to experience the full panoramic view offered by the video as it allows users to be immersed in the content.

The most common type of 360^{o} available is monoscopic which is a flat 360^{o} video with no 3D depth and consequently offers 3 degrees-of-freedom associated with rotational freedom. Stereoscopic 360^{o} videos exist but they require cameras that are too expensive at the moment for regular consumers. Additionally technology to convert monoscopic videos into stereoscopic is in its infancy and not readily available, as such the focus of this dissertation will be on monoscopic videos.

The remainder of literature review will be focused on three key subtopics, the concepts of immersion and presence, focusing viewer attention in 360^{o} content and interactive 360^{o} video.

2.2.1 Presence and Immersion

Immersion and presence are key concepts in the virtual reality experience. Slater and Wilbur defined immersion as "the extent to which the computer displays are capable of delivering an inclusive, extensive, surrounding, and vivid illusion of reality to the senses of a human participant." [SW97] and presence "as the psychological state of being in a virtual environment" [SW97].

Scholars have identified six important factors that influence the user's sense of presence. These include vividness, use of a virtual body, interactivity, internal factors, realism and distraction factors.

Steuer associates the concept of vividness with the ability of the technology to provide a rich sensorial experience. The sensorial experience needs to offer both breadth and depth. Breadth is concerned with the variety of senses that are being stimulated and depth relates to the resolution of the sensorial experience in each of the senses [Ste92].

Two of the senses that will be stimulated by the interactive experience discussed in this dissertation are sight and sound. On the topic of sight, technological factors such as resolution, field of view and the general quality of the display severely influence the fidelity of the experience. Regarding audio, music and sound effects will need to be appropriately integrated to make the experience more believable.

One of the most relevant factors, interactivity, defined as the extent to which users can modify and affect a mediated environment with their input in real time [Ste92] is not only something that enhances the user's presence but it is also at the core of making the gameplay satisfying and engaging to the user. Steuer discusses three characteristics that are necessary to provide adequate interactivity, namely speed, range and mapping. Speed is equated to response time of the system, range refers to the number of interactions made possible and finally mapping is concerned with properly associating actions with controls in response to changes in the system's state [Ste92]. These characteristics are definitely applicable in the context of adventure games and game usability in general as they relate closely to concepts such as direct manipulation and contextual interactions previously discussed in section 2.1.

The use of a virtual body refers to the existence of an accurate representation of the user's body in virtual reality and it implies positional tracking to mirror the movements of the real person [SU93]. As these games will be played with monoscopic 360^{o} videos there is little point in taking advantage of positional tracking because the game world lacks perceptual depth. Research has been made with the intent of converting monoscopic 360^{o} videos obtained through a single camera into a stereoscopic format as to provide 6 degree-of-freedom tracking. The results are still limited to the quality of the 3D reconstruction process employed to the monoscopic video and may vary significantly from video to video. From the obtained 3D geometry of the scene new views can be synthethized by warping the content on a perspective basis. This entire process may also cause image artifacts and noise and it requires high-end consumer grade GPUs and CPUs due to requirements of heavy real time processing power [HCCJ17].

One of the six previously identified factors, internal factors, are concerned with the individual differences between users and how these factors influence the way a user cognitively experiences the virtual environment [SS02]. Realism factors are not connected with real-world content but refer instead to "the connectedness and continuity of the stimuli being experienced" [WS98]. Distraction factors refer to the isolation of the user from stimuli from the outside world, which can be achieved with the HMD and other equipment such as headphones. Other distraction factors include the observer's willingness to focus on the VR stimuli [WS98] and the existence of intrusive interface elements that may appear clumsy and unnatural to users [HD92]. The distraction and realism factors will thus be highly correlated with the quality of the game's user interface, interaction design and VR display hardware.

More recent literature reviews on sense of presence and immersion in virtual reality [NN16, RTGG17] have also used the theoretical foundations outlined by the likes of Steuer, Witmer and Slater.

2.2.2 Focusing Viewer Attention

Cinematic virtual reality refers to the use of 360° video as a means for immersive passive or interactive storytelling. One of the biggest issues associated with this format of storytelling is that of guiding the viewer's attention into relevant elements as there is a lack of natural directorial control over what the viewer is experiencing [MS17]. Lack of directorial control over what the viewer is seeing may lead the user to miss important moments in the narrative. However if the user is properly immersed in the experience and the video features proper visual and acoustic cues there is a greater emotional response from viewers to the content compared to traditional video formats [EEW18].

Brown et al. found that the most effective and non obtrusive way of guiding the viewers attention was a combination of both visual and audio cues, even without spatialized audio [BSEW16]. Additionally participants in experiments with sample monoscopic 360° videos also reacted positively to being directly addressed or acknowledged in the narrative [BSEW16]. Special concerns regarding the distance between the user's point of view and the contents of the 360° video need to be taken into account [BSEW16]. Evidence suggests that people may feel discomfort at an invasion of their personal space in virtual reality just as they would in the natural world [WAEG06]. The context of the video scene also needs to be taken into consideration: non threatening scenes will not require as much of a perceived distance between the viewer and the action and too large a distance may lead to a decrease in the immersive qualities of the video [BSEW16].

Several alternative approaches have been proposed towards fixing the issue of guiding the viewers attention. In the context of cinematic virtual reality Nielsen et al. tested the use of a firefly that highlights relevant portions of the video as well as forced rotation and they found that the latter could lead to a decrease in the user's sense of presence [NMH+16] as it strips away control and user freedom. Another approach by Matos et al. is a system with different types of annotations that not only help guide the user's attention but also may present additional information about certain elements present in the video [MNRP18] making it more tailored towards informative and journalistic content. The proposed project will use the work developed by Matos where appropriate, to give developers basic tools for guiding the player's attention. An example application of this work would be using annotations that change their position over time in the video to make static content interactive.

2.2.3 Interactive 360° Video

Adventure games with 360° video backgrounds are a form of interactive 360° video. Related work to interactive 360° video include the previously mentioned dynamic annotation system where users may look at an annotation leading it to expand as to display additional information about a certain element in the scene [MNRP18]. This functionality is similar to that of a mouse hover event in traditional web browsing.

A different approach to interactive 360^o video and one that has some overlapping similarities with the proposal of adventure games with 360^o videos is that of gamified 360^o videos. Gamification is the process of introducing game like elements to non gaming media. Overlapping functionality in this project include the integration of 3D objects in the environment and the ability to make choices and thus branch the narrative. Other functions include time-based challenges, scores and levels [AEBD16]. While this research was more focused on gamification of 360^o video with generic game elements, this dissertation will be focused on bringing adventure games to the format and on developing a framework that will allow creation of content with a low/no code approach.

Argyriou et al. highlighted several design challenges inherent to making 360^o videos interactive. These challenges can be classified as either technical or as general design challenges [AEBD16]. The technical challenges include the smooth transition between the video resources, creation of natural looking environments and reality-based navigation [AEBD16]. The design challenges relate to previously discussed topics such as the creation of a non intrusive user interface, a usability guideline that can be connected back to the distraction factors highlighted by Witmer [WS98]. The other noteworthy design challenge is emphasizing important elements of the panoramic view which is related to the earlier topic of focusing viewer attention on relevant elements in the scene. The technical challenges highlighted by Argryiou et al. very relevant to this dissertation so they will be described in more detail.

Smooth transition between different resources is described as a challenge because there may be delays and blinking effects when transitioning between different video resources and that could affect the immersion factor of the experience [AEBD16]. The solution applied to this problem was employing fade in and fade out effects from the game engine to avoid blinking effects and keeping the video assets preloaded as to avoid delays between the transitions [AEBD16].

The second technical challenge - the creation of a natural looking world - concerns the integration of virtual objects into the 360^o video environment. Argyriou et al. suggest the use of photogammetry techniques to reconstruct real world objects into 3D models as to make them as

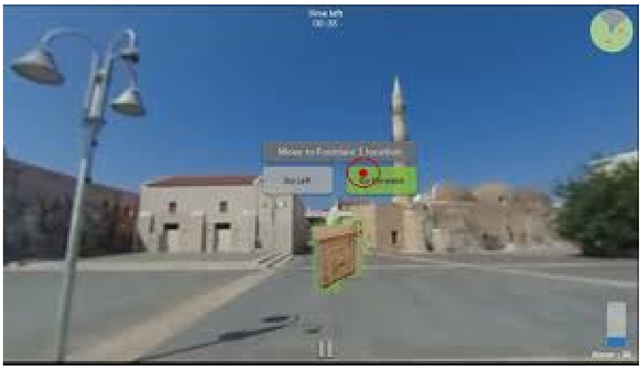


Figure 2.11: Image of gamified 360^o video from Argyriou et al. [AEBD16]

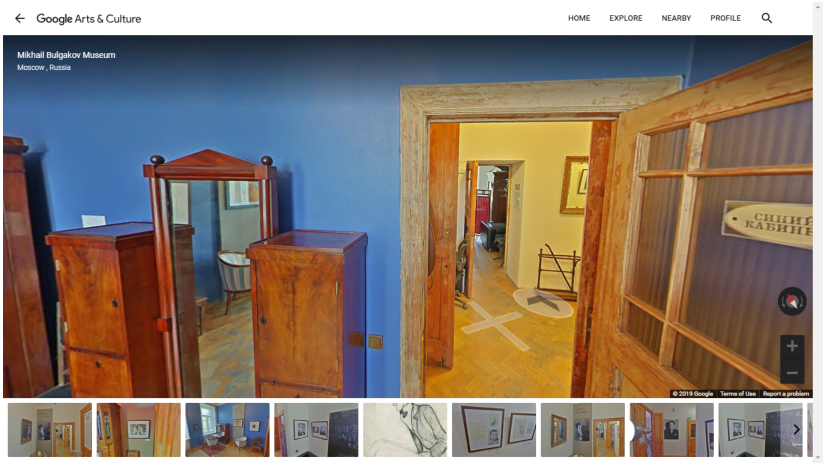


Figure 2.12: Google Arts and Culture - Mikhail Bulgakov Museum navigable through streetview

realistic as possible and thus more naturally fitting with the rest of the environments. Another important suggestion is adaptive scaling applied to the 3D objects in case of perspective changes in the video [AEBD16]. The main takeaway is that in the instance of using 360º videos with camera movement it is important to take into consideration maintaining consistency in the presentation of the 3D objects by changing the transforms applied to said objects over time.

Lastly, reality-based navigation is more of a technical problem related to the filming process or capture of 360º content. It refers to making smart use of camera techniques such as panning, tilts and zoom to emulate movement and more natural interaction with the environments [AEBD16]. Overall this research into gamification of 360º video has provided instrumental design guidelines for application in similar problems that will be tackled throughout the development phase of this dissertation.

Another example of a noteworthy interactive 360º project is the Google Arts and Cultures' initiative to make use of street views[18] to allow the exploration of cultural spots such as museums through a point-and-click navigation system [Dou18]. Finally, in the same vein Kwiatek developed an application that makes use of interactive 360º video to preserve the cultural memory of Charles Causley's poetry [Kwi12]. This application allows users to explore several environments of import in the life of the poet accompanied by an interface and the integration of virtual objects that contain relevant information on the poet's biography and body of work.

2.3 Summary

Through a brief analysis of the past and present landscape of adventure games it was possible to ascertain that they are a multi-format genre. They evolved from text-based interfaces to verb menus to modern context-based interaction or direct manipulation interfaces.

Design challenges from different types of adventure games were highlighted and discussed as well as general design concerns related to the whole of the genre.

[18] Street views are a collection of 360º images that depict the real world - See https://mapstreetview.com/

Adventure games also feature a variety of sub genres, many of which will be possible creations through the proposed framework as their mechanics overlap with those of adventure games in general. Some of these sub genres are characterized more by their presentation style and story context than differing interactive systems.

The core game mechanics were identified as puzzles, hint systems, exploration, object examination, inventory systems, object use, combining of objects and dialogue systems (dialogue tree), player choice and consequently branching narratives.

The virtual reality component introduces new design concerns such as the need to support comfort options like snap turning to combat simulator sickness, giving proper cues and guidance towards relevant content in the scene and concerns related to preserving and enhancing player immersion and sense of presence.

From the analysis on the literature about presence in virtual reality video many useful design guidelines were internalized. Such guidelines include the design of non intrusive interfaces, betting on the vividness of the experience by using high quality HMD's and stimulating as many senses as possible with proper use of visuals and sound. Interactivity was highlighted as a key component of presence and as such the software needs to be responsive, support a range of engaging interactive mechanics and be mapped to controls in input devices that make sense.

Furthermore from the investigation on existing work for interactive 360^{o} video it was decided that the work on dynamic annotations by Matos [MNRP18] may be used as a foundation to provide basic functionality for guiding the player's attention to certain points of interest in the environment. Finally, technical challenges highlighted and discussed in [AEBD16] provided valuable insight into solving issues related to transitioning between different 360^{o} videos, creating an environment that mixes recorded 360^{o} visual media with virtual 3D objects and use of proper camera techniques when capturing 360^{o} content to enhance the interactivity of the games.

Chapter 3

A Framework for Interactive 360 Video Adventure Games

From the literature review it was possible to understand which were the core game mechanics that make up the DNA of adventure games and how immersive 360 video has the potential to enhance the user's sense of presence when coupled with adequate virtual reality technology. With that in mind, this chapter moves into conceptualizing a framework for the creation of interactive 360^o video based adventure games with a more high-level view of the framework goals and interactions that it should support. The second half of the chapter focuses on outlining the implemented solution for the framework, aptly named "Real Adventure". Throughout the chapter there will be images of a game produced with "Real Adventure". It is important to consider that the proportions of the various scene elements is different when seen from a proper VR headset in contrast with the 2D screenshots present here.

3.1 Conceptualization

The conceptualization section contains the general system requirements, the intended creation workflow and use cases. Finally, it also features the types of objects that can exist in the game as well as the interactions that need to be implemented based on the core mechanics that were discussed in section 2.1.3.

3.1.1 System Requirements

In this section a list of the general system requirements of the framework will be presented.

- *Low code* - The framework must be accessible to people with a cursory understanding of coding or game development
- *Usability* - The experiences created through the framework should provide clear and understandable interactions

- *Compatibility* - The experiences created through the framework will be compatible with browsers that support the WebVR standard and compatible with headsets with 3 or 6 DOF controllers such as the Oculus Quest or the Rift/Vive
- *Accessibility* - The framework needs to support common accessibility features such as subtitles and snap turning
- *Persistence* - The framework needs to account for longer experiences and provide automatic game saving and loading capabilities

From the perspective of potential content creators, the low code requirement will be very important and provide an advantage in terms of ease and speed of game development. The remaining requirements are necessary to provide satisfactory playing experiences to users.

3.1.2 Target Creation Workflow and Output

In order to visualize the expected output of the projected framework, a mockup was created with some of the intended core features visible. In Figure 3.1 it is possible to observe a screenshot of a live action environment taken from a 360º video and the integration of interactive virtual 3D objects in the form of the popcorn and the orange lamp. The small white sphere at the door would allow players to move into a different environment, serving as a navigational portal. Additionally, the sprite of a virtual 2D character from the game "Hotel Dusk: Room 215" (2005) is displayed alongside an example of the interface for player choice. Dialogue trees would be displayed in a similar manner to that of the player choice UI.

Since it is a 360º interactive environment players would be able to look around through rotational head movements or through alternative input methods in case the physical space poses an inconvenience to full 360º head turning to the player. This alternative rotation method will be doable through snap turning, meaning sharp and instantaneous rotations in fixed angles.

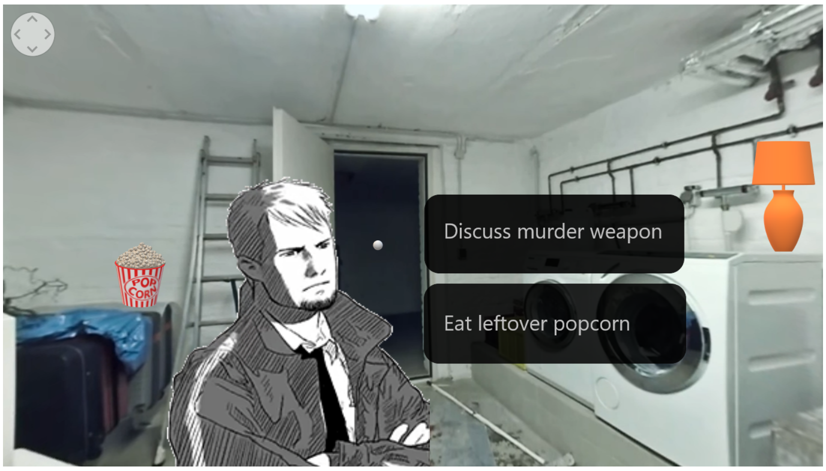


Figure 3.1: Mockup of an hypothetical adventure game produced through the proposed framework

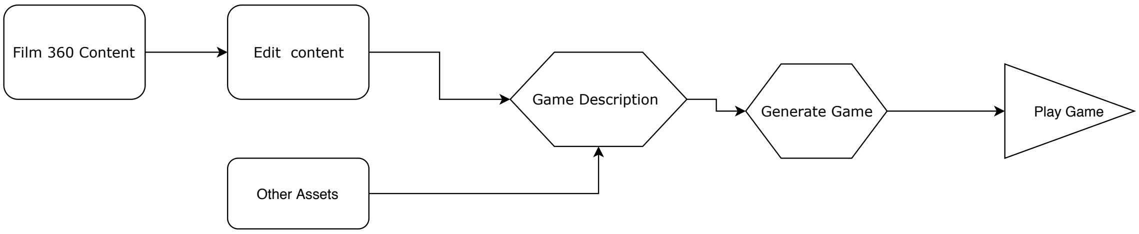


Figure 3.2: Expected game creation workflow

The target content creator workflow can be seen in Figure 3.2. Content creators are expected to obtain the necessary assets before starting development with the planned framework. Filming and editing of 360^{o} content will be necessary to set the groundwork for the game environments. Other assets include standard format subtitle files for the 360^{o} videos, music, sound effects, 3D models of objects, 2D images of objects or characters and icons to represent actions such as picking up objects or examining as well as icons for each object in the player inventory.

The next step is describing each scene, dialogues, scripted transitions and object inventory combinations in the game through a custom descriptive language. Following the description of the entire game through various files the framework will automatically generate the game at runtime by parsing the descriptions and convert everything to scene elements.

The proposed framework could be used in several contexts in addition to the creation of games for entertainment.

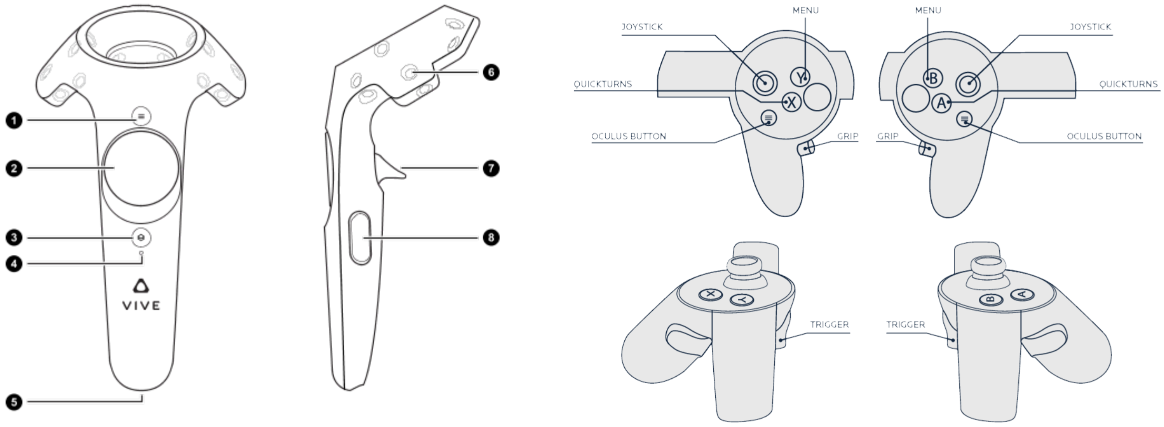
3.1.3 Virtual Objects and Interactions

This section is comprised of a comprehensive list of game mechanics/interactions intended to be supported and implemented into the framework. This list can also be seen as the framework's functional requirements. Some features such as physics based 3D objects, voice command controls as well as a drag-and-drop GUI for game creation would be interesting inclusions in the framework but they are out of the feature scope of this thesis and as such were not included.

As a frame of reference the general control scheme for any type of supported VR controller is composed of a laser that comes out of a 3D model representation of the input device in order to point at interactive entities in the environment (it is only necessary to have one controller, not two unlike a typical virtual reality game).

Additionally it is also necessary to use two buttons for the core functions. A general action button, generally mapped to the main trigger button of a VR controller (button 7 in Figure 3.3a and button "trigger" in Figure 3.3b) and the inventory button, a button to open and close the inventory (button 1 in Figure 3.3a and button "grip" in Figure 3.3b).

In the case of controllers with a greater number of input options like the Vive Wands and the Oculus Touch controllers (See figures 3.3a and 3.3b respectively) there is also a way to snap turn the camera to the left or right with the big touchpad in the case of the Vive (button 2 in Figure 3.3a and in the analog stick in case of the Oculus Touch controller ("joystick" in Figure 3.3b).



(a) Vive Wand Controller

(b) Oculus Touch Controller

Figure 3.3: Common VR controllers - The most prominent VR controllers and their input configurations

Before going into detail about the possible interactions it is first necessary to contextualize the types of objects that can be present in an interactive 360^o video based adventure game and how these objects are classified and interacted with.

Table 3.1: Virtual Object Types

Object Type	Description
Aesthetic Objects	These objects can be any 3D or 2D asset placed in the scene, visible and with no interactivity
Examinable Objects	By pointing with the laser and clicking on this type of object a dialogue sequence will be triggered. Examinable objects may be an invisible geometry that will allow game creators to make static recorded/captured visual content an examinable object in the game
Pickable Objects	By pointing with the laser and clicking on this type of object, the object will be collected and added to the player's inventory
Code Puzzle Objects	By pointing with the laser and clicking on this type of object, a numpad-like interface (though it may contain any other character besides numbers) will be displayed that allows players to input a specific sequence of characters for secret-code type puzzles
Navigation Spheres	By pointing with the laser and clicking on navigational spheres, the player will be transported into a different 360^o environment after a fade out and fade in transition

The interactions represent all the game mechanics that the player can explicitly play with and as a result change the state of the game world or even influence the outcomes of the storytelling. Table 3.3 will contain a listing of all the possible interactions conceptualized for the Real Adventure framework, a description of how the interaction is achieved and the possible outcomes from each interaction.

Table 3.3: Interactions

Interaction	Description	Outcome
Navigation	Point and click at a navigation sphere to change the current background environment be it a 360^o video or image	The background will change into a new one after a fade-out and fade-in effect.
Snap turn	By using the correct input controls either snap the camera to the left or the right in a fixed angle increment. Useful to allow players to rotate the camera without rotating physically with their body	Camera rotates to either the left or right in a fixed angle
Examine object	By pointing with the laser and clicking on examinable objects a dialogue sequence will trigger. Actions can be seen in sequence from Figure 3.13 to Figure 3.15	A dialogue sequence with text will trigger, may offer clues to the current puzzle in the room or include some descriptive text related to that object and the overall narrative the game is trying to present.
Pickup object	By pointing with the laser and clicking on a collectable object will make said object fly into the player if the animation is enabled and will be added to the player's inventory	New object is added to player inventory
Solve code puzzle	By pointing with the laser and clicking on this type of object, a numpad-like interface is displayed such as the one in Figure 3.4 for players to input the correct solution based on clues gathered in the environment. This mechanic requires interaction with others to be engaging and fun	By inputting the correct solution the puzzle will be solved
Open inventory	Click the dedicated inventory button to bring up the inventory	Opens inventory
Combine objects	Hold the main action button (trigger) pointed at an inventory object to drag it into another inventory object to try to combine those objects. See Figure 3.5 for reference	Creates a new object out of the two combined objects if that combination is valid, removes the used objects from the inventory and adds a new one

Use objects	Drag and drop an inventory object into another object in the environment. See Figure 3.6 for reference	There are three possible outcomes. In any of the outcomes the object that was used in the inventory may or may not be removed from the inventory, it is all up to the content creator. The outcomes are as follows: • A new object is added to the inventory (Example: dragging a key from the inventory into a locker in the environment to get what is inside) • The object changes its appearance (Example: dragging a lighter into a newspaper 3D object may update the model into a burned 3D object) • Some dialogue is triggered (Example: Dragging a missing part in the inventory into a machine in the environment triggers a dialogue to signify the machine has been successfully fixed)
Close inventory	Click the dedicated inventory button to close the inventory	Closes inventory
Advance Dialogue	Click the main action button (trigger) to advance the to the next piece of dialogue or close the current dialogue if there is not anymore text to be displayed	Advances the text box to the next piece of dialogue or makes the text box disappear if no more dialogue
Make choice	When prompted for making a choice, either for conversational topics in dialogue trees or a singular narrative decision, point to the desired option and choose. See Figure 3.14 for reference	A record is made in the game state that the player made a certain choice and advances to the next piece of dialogue or scripted event in the game

Present item	When prompted with the inventory interface choose the correct item based on the story/puzzle context of the game	Proceed to the next scripted event in the game or have the player retry if he chooses the wrong item

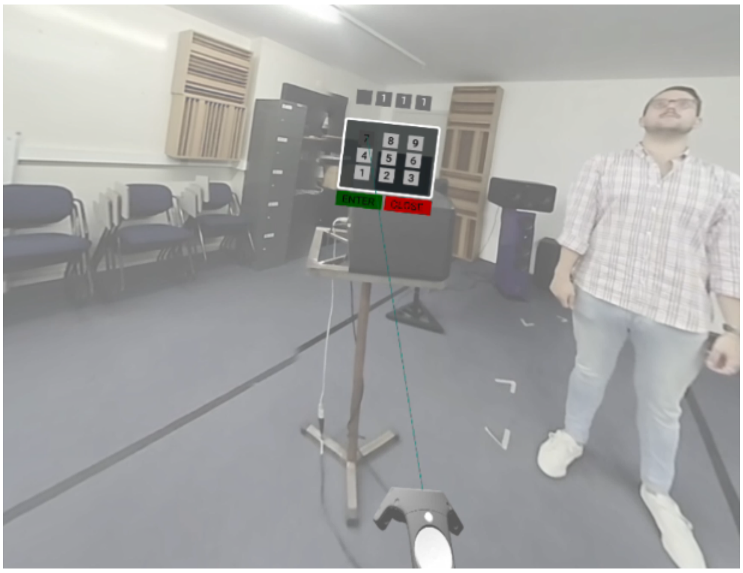


Figure 3.4: Code Puzzle Interaction - Input the correct combination of numbers or characters to open a locked object

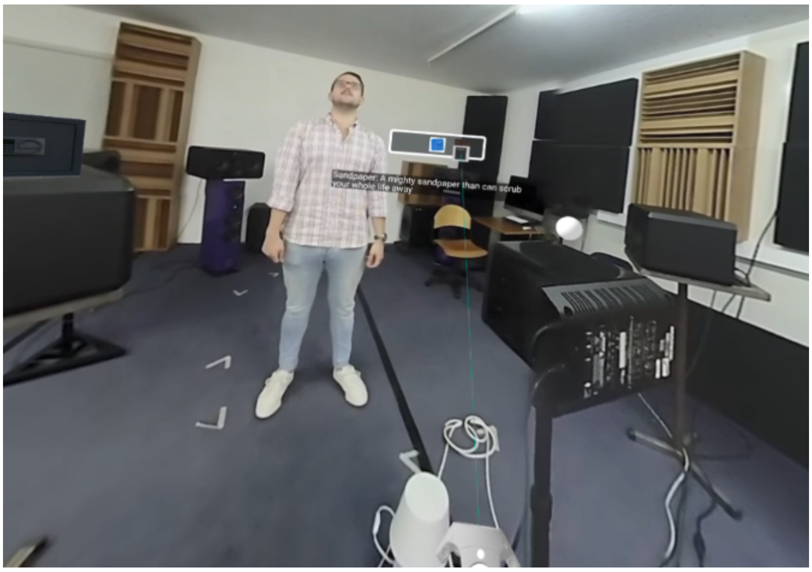

Sandpaper: A mighty sandpaper than can scrub your whole life away

Figure 3.5: Combine Objects Interaction - By dragging and dropping the inventory object into another inventory object it is possible to combine them to make a new object

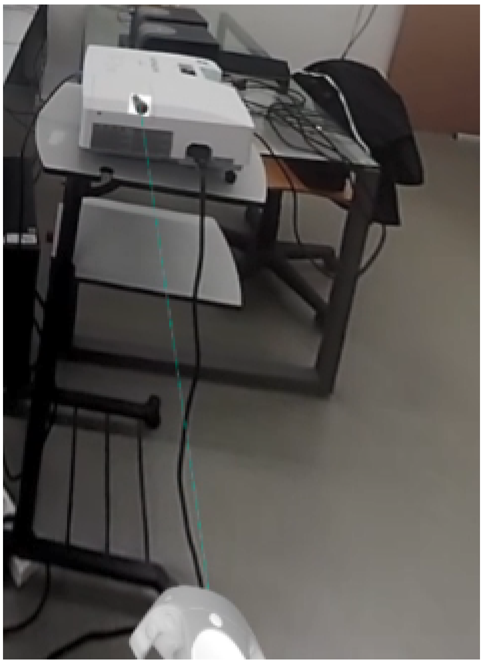


Figure 3.6: Use object Interaction - Use objects by dragging and dropping the inventory object in another interactive object in the environment

3.2 Development

This section will feature a more technical look on the development of the framework starting with the chosen development tools and technologies, system overview and the general architecture of Real Adventure followed by the scene elements present in a game made with the framework and how they were implemented.

In succession to the basis for understanding the general structure of the framework it is important to go into detail about the implemented components that provide behaviour and functionality to scene entities. Finally, there will be an overview of the descriptive language used in Real Adventure and the development methodology that was adopted given the characteristics of the implemented solution.

3.2.1 Development Tools and Technologies

The technologies used to develop the framework were evaluated during the conception phase of the project and the goals of the framework were taken into consideration. Those goals were mainly creating easily accessible and cross device WebVR experiences as well as having low-code game creation.

A-Frame[1] is a Javascript framework for the creation of WebVR content. It was built on top of HTML and three.js[2] and provides unlimited access to DOM APIs, WebVR, and WebGL through

[1] A-Frame, https://aframe.io/docs/0.9.0/introduction/, last access 2019

[2] Three.js, https://threejs.org/, last access 2019

an entity component structure. This combination of HTML/three.js with an entity component structure allows for easy translation between a JSON game description to A-frame entities with plugged-in functionalities through the declarative component system. Moreover by using DOM operations it is possible to select, add and remove singular or groups of entities to the scene and similarly through the DOM API's functions *setAttribute* and *removeAttribute* it is possible to add, alter or remove components from said entities to enable, change or disable a specific functionality at any point during the game's runtime. To optimize for performance in many cases it is recommended to access the underlying three.js *object3D* API directly and to throttle behaviour that would otherwise run on every frame whenever possible. In general best practices recommended in the A-Frame documentation[3] were employed in the development of the Real Adventure framework

The entity-component-system architecture follows the composition over inheritance principle. Entities are the base objects and components attach functionality, data, appearance and behaviour to said entities. This allows for a very modular design where components can be re-used and attached across different kinds of entities. Essentially, a single intended functionality such as making an object in the environment collectable can be implemented as its own component.

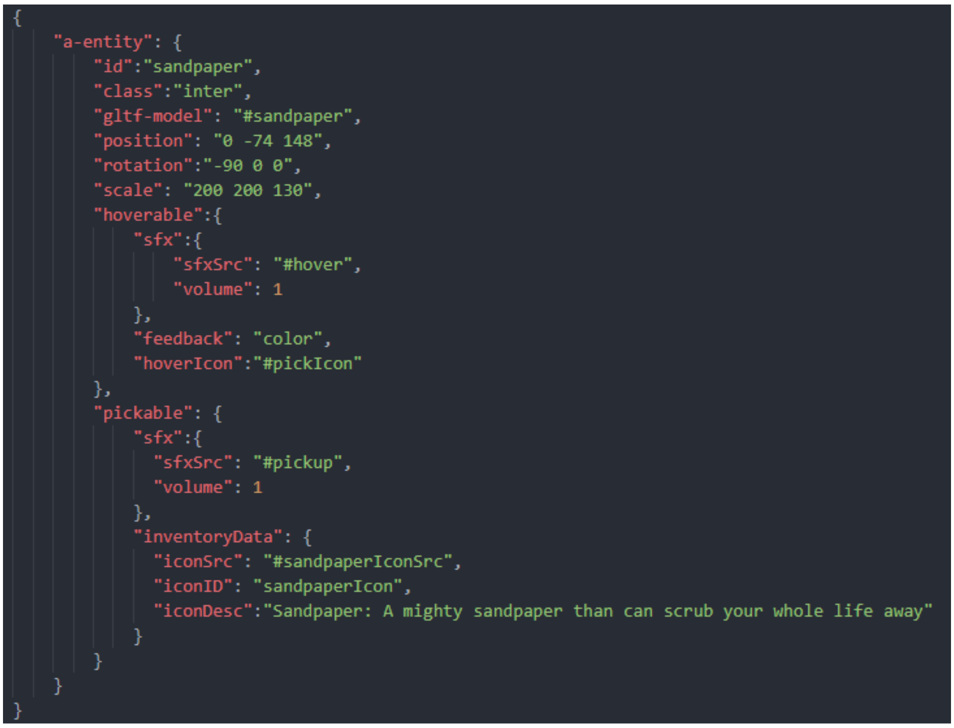
Figure 3.7 shows an example of the JSON representation utilized by Real Adventure of an entity that will be inserted into the DOM with a given *id* and *class*. The *id* and *class* are standard DOM attributes but the remaining ones are A-Frame components. Some of these A-Frame components may be out-of-the-box components that come with the A-Frame itself while others were components implemented from scratch for the purposes of the Real Adventure framework.

An example of an out-of-the-box A-Frame component would be the *gltf-model* component. It receives a selector for a 3D asset to be rendered as the entities' visual representation. This means that if a game creator wants to have an entity in the scene that is a tree, he will need to add a *gltf-model* component to that entity. This very same *gltf-model* component receives a selector for the 3D model that will be used to render the entity (a tree in this case) in-game. On the other hand, *pickable* is an example of a custom-made component. This component allows an in-game object to be collected by the player upon point-and-click interaction.

Each key-value pair in the *a-entity* object represents either a regular DOM attribute and its value or an A-Frame component (custom or out-of-the-box) and its corresponding data attributes. A-Frame components can be specified with a custom schema to represent data attributes that are allowed to be passed to that component. This data can be freely used in the programming of the intended functionality of the component. In the case of the previously mentioned *pickable* component the data schema can be seen in Figure 3.8. In essence, entities are neutral and empty until components are attached in order to provide appearance, purpose and functionality.

The last concept in the ECS architecture is that of systems, these are optional and may be used to implement more global-scope management functionalities such as a global store. Incidentally that very same system was used in the Real Adventure framework. The global store allows the existence of a player inventory, management of various application states, storing of flags based on player interactions in-game and also to handle persistence (save and load game) by interfacing

[3] A-Frame Performance Tips, https://aframe.io/docs/0.9.0/introduction/best-practices.html, last access 2019

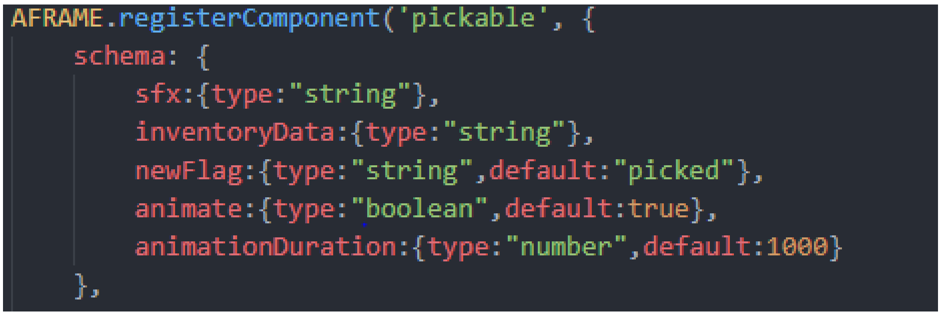


```
{
    "a-entity": {
        "id":"sandpaper",
        "class":"inter",
        "gltf-model": "#sandpaper",
        "position": "0 -74 148",
        "rotation":"-90 0 0",
        "scale": "200 200 130",
        "hoverable":{
            "sfx":{
                "sfxSrc": "#hover",
                "volume": 1
            },
            "feedback": "color",
            "hoverIcon":"#pickIcon"
        },
        "pickable": {
            "sfx":{
              "sfxSrc": "#pickup",
              "volume": 1
            },
            "inventoryData": {
              "iconSrc": "#sandpaperIconSrc",
              "iconID": "sandpaperIcon",
              "iconDesc":"Sandpaper: A mighty sandpaper than can scrub your whole life away"
            }
        }
    }
}
```

Figure 3.7: Virtual Object JSON Representation Example - An example of a JSON representation of an entity. Describes a sandpaper object represented as a 3D model in the game environment that can be collected by the player

with the *localStorage* API in the browser. By constantly updating the browser's *localStorage* with the current state of the game it is possible to close and re-open the game at any point and it will automatically load the necessary saved states to resume the game, as long as it is done in the same computer-browser combination.

In particular, a very important use of the global store is keeping track of an array of flags related to actions performed by the player. If the player examines an object, watches a cutscene, makes some story choice or solves a puzzle and basically every other possible event in this type of game will lead to a flag being automatically added to that array in the global state. These flags will



```
AFRAME.registerComponent('pickable', {
    schema: {
        sfx:{type:"string"},
        inventoryData:{type:"string"},
        newFlag:{type:"string",default:"picked"},
        animate:{type:"boolean",default:true},
        animationDuration:{type:"number",default:1000}
    },
```

Figure 3.8: *Pickable* Component Data Schema - An example data schema for the data attributes that may passed to the component *pickable*. Entire Javascript objects can be received as a string (*inventoryData* attribute is an example of this)

allow content creators to freely script transitions in the game such as transitioning from opening a locked safe to adding an object in the inventory and by having that object in the inventory we can transition into triggering a dialogue or a cutscene as an example.

In essence this is the heart of the framework: it allows content creators to easily setup a chain of events and plan out the entire flow and sequence of the game and ultimately to also create branching and player-choice driven narratives. The way to setup these transitions through the Real Adventure JSON and the possibilities afforded by them will be explained in more detail in section 3.2.5.

This global store system was available as an external A-frame package/component[4] as were others used in the framework. Other external components include a custom 9-slice geometry[5] needed for the kind of texture intended to adorn the inventory interface and a *look-at*[6] component that when attached to an entity makes it face the camera in-game at all times.

As a Javascript framework, A-Frame can also interface with non A-Frame related modules, which allowed the use of an already existing Javascript parser for existing and commonplace subtitling specifications to more easily implement subtitles for 360^{o} videos. Another example of external tech is the one related to helping users focus their attention on relevant content, for this purpose an arrow annotation was adapted from the previously mentioned work on dynamic annotations [MNRP18] into an A-Frame component. Since the arrow annotation technology was implemented using three.js the work needed to port it over to A-Frame was practically non-existent. In conclusion, A-Frame was an adequate tool when taking into consideration the main goals of this thesis as it offered cross browser and device support out-of-the-box as well as a programmatic structure that can be easily parsed, manipulated and interfaced with through a descriptive language. JSON in particular was chosen as the descriptive language because it is the native to Javascript object notation and A-Frame is programmed in Javascript thus facilitating the encoding and parsing of data needed by Real Adventure. While A-Frame provides the low-level facilities needed to create Virtual Reality games (components for controller mappings and tracking, 360^{o} media playback and 3D graphics), the intended interactions for interactive 360^{o} video adventure games had to be implemented from scratch. Additionally, the Real Adventure framework had to be engineered as to allow the descriptive-based creation of content with an architecture and project structure of its own, detailed in section 3.2.2.

An additional tool is required to run the games developed with the framework, which is a generic HTTP server such as WAMP or Google Chrome's Web Server application. An HTTP server is needed to serve the static assets of the game to the browser through the HTTP protocol.

[4] aframe-state-component, https://www.npmjs.com/package/aframe-state-component, last access 2019

[5] Slice-9 Component, https://github.com/fernandojsg/aframe-slice9-component, last access 2019

[6] Look-at Component, https://github.com/supermedium/superframe/tree/master/components/look-at, last access 2019

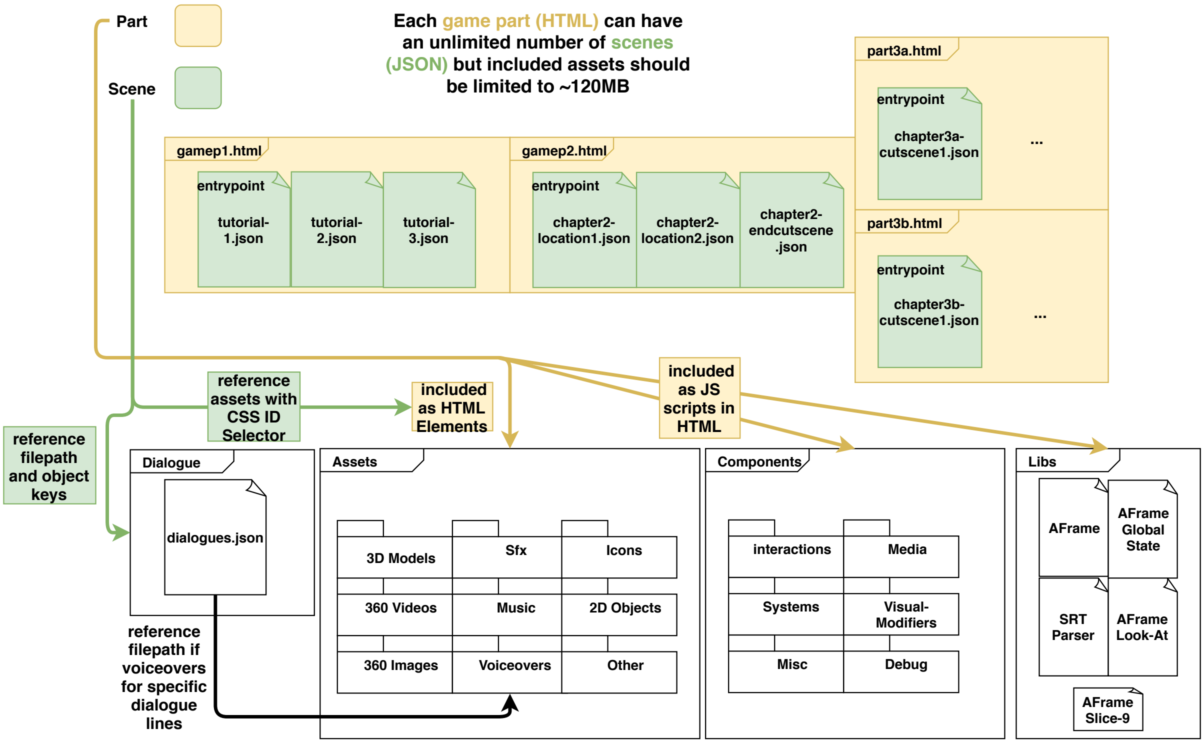


Figure 3.9: Real Adventure Project Anatomy - An overview of the structure of a game project made in the Real Adventure framework

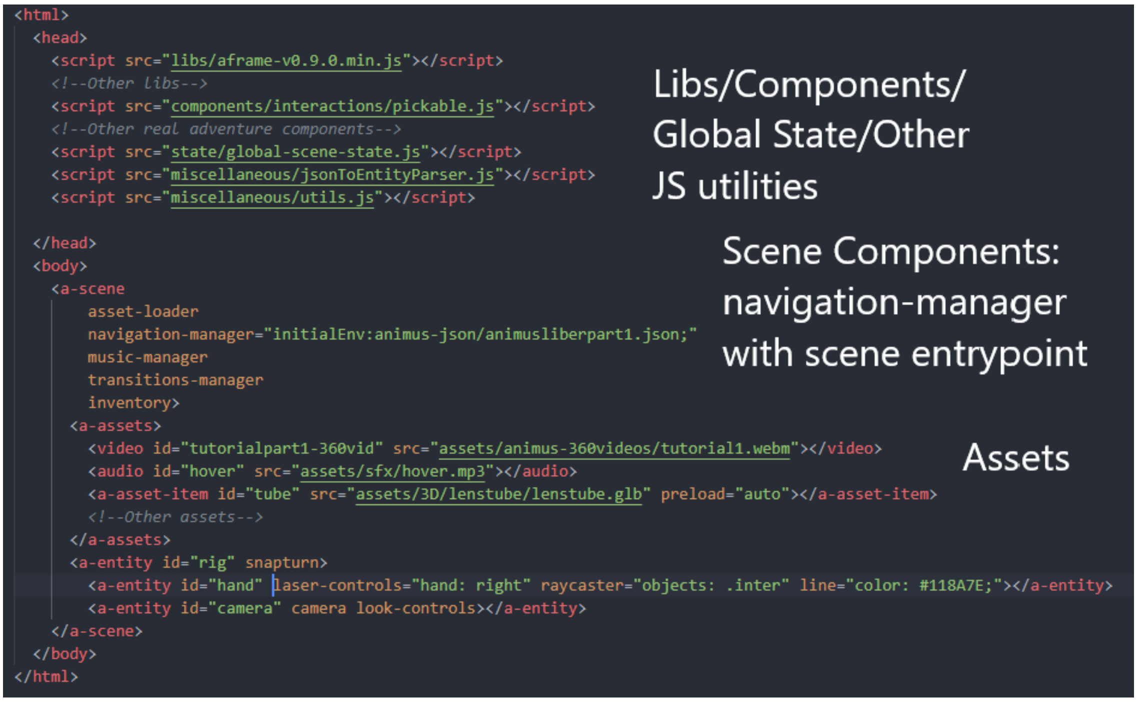
3.2.2 System Overview and Architecture

In order to grasp the inner workings of the Real Adventure framework it is first necessary to understand the project structure of an hypothetical game developed with the framework. For that purpose a careful examination of Figure 3.9 is needed.

A project in Real Adventure is primarily composed of the following:

- **Parts** - A game part is the HTML page through which one can access the game. Theoretically an entire game could be made using only one part but given that the intended purpose is to make games with 360 videos, due to their size it is mostly not possible to keep the game relegated to one part/HTML page only. Assets need to be included in this HTML page as media/asset elements and to ensure a smooth playing experience these assets need to be pre-loaded and cached on scene initialization.

  Additionally from experience during the development and trying to target lower-end systems like the Oculus Go it was surmised that asset sizes should not go over a limit of around 120 megabytes for a single HTML page to avoid browser crashes and other issues. As such, for full sized games it is advisable or even necessary to divide the game across multiple parts/HTML pages even if this separation has no meaning besides asset size limitations. It is possible to navigate to and keep the game state across different pages because Real Adventure saves the necessary global state attributes in the browser's *localStorage* and allows the



```html
<html>
  <head>
    <script src="libs/aframe-v0.9.0.min.js"></script>
    <!--Other libs-->
    <script src="components/interactions/pickable.js"></script>
    <!--Other real adventure components-->
    <script src="state/global-scene-state.js"></script>
    <script src="miscellaneous/jsonToEntityParser.js"></script>
    <script src="miscellaneous/utils.js"></script>

  </head>
  <body>
    <a-scene
        asset-loader
        navigation-manager="initialEnv:animus-json/animusliberpart1.json;"
        music-manager
        transitions-manager
        inventory>
      <a-assets>
        <video id="tutorialpart1-360vid" src="assets/animus-360videos/tutorial1.webm"></video>
        <audio id="hover" src="assets/sfx/hover.mp3"></audio>
        <a-asset-item id="tube" src="assets/3D/lenstube/lenstube.glb" preload="auto"></a-asset-item>
        <!--Other assets-->
      </a-assets>
      <a-entity id="rig" snapturn>
        <a-entity id="hand" laser-controls="hand: right" raycaster="objects: .inter" line="color: #118A7E;"></a-entity>
        <a-entity id="camera" camera look-controls></a-entity>
    </a-scene>
  </body>
</html>
```

Figure 3.10: Mock HTML game part of a game in Real Adventure

scripting of flag-based navigation to different URLs through the JSON descriptive language, more on that in section 3.2.5.

A simplified mock file of a game part can be seen in Figure 3.10. Of note is the HTML attribute/component[7] of the <*a-scene*> element called *navigation-manager*. By passing the *initialEnv* data-attribute to *navigation-manager* a particular game scene can be defined as the page's entry point. Scenes will be described below in detail. While <*a-scene*> is the literal HTML entity that represents the root node of the scene graph, when the text refers to scenes it is in relation to the actual content of the game that is injected into the scene graph and not the previously mentioned root element of the scene graph.

- **Scenes** - A game scene is described by a JSON file that delineates the main background be it a 360° image or video and its children elements (virtual objects of the scene). Virtual objects may also have their own child elements, which may include other virtual objects (child elements are positioned relative to their parent). However, it is probably simpler and easier for content creators to keep the scene graph as shallow as possible. Aside from the background and the virtual objects a scene JSON description may also contain possible combinations of inventory objects and flag-based transitions that will be added to global state whenever that particular scene comes into play. The scenes refer to the declared assets in the HTML page in which they will be rendered and played.

[7] Reminder: A-Frame components are DOM HTML attributes with their own behaviour and scripting

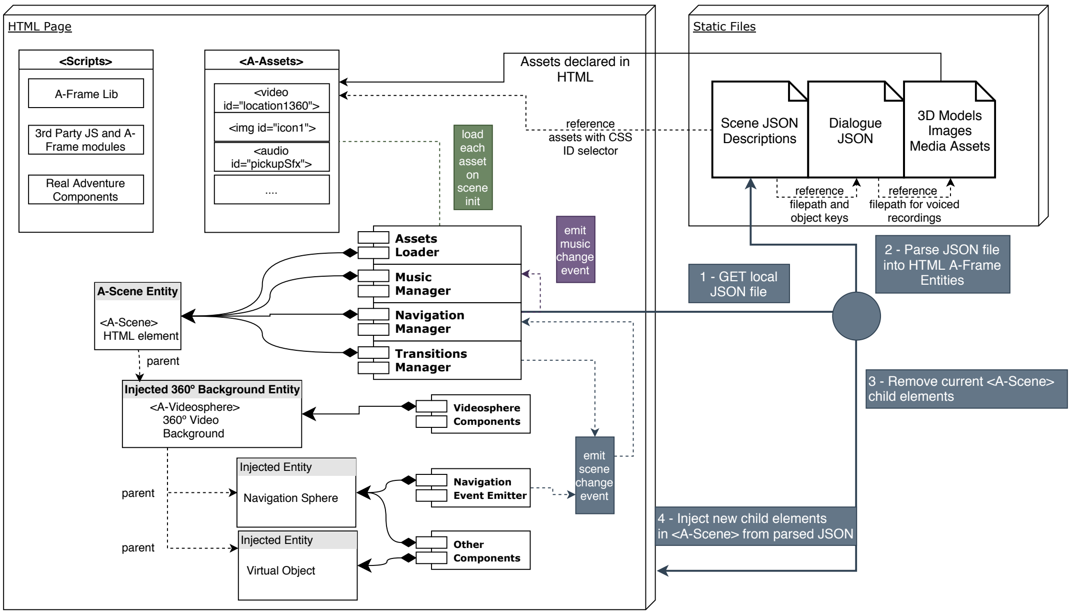

HTML Page
<Scripts>
A-Frame Lib
3rd Party JS and A-Frame modules
Real Adventure Components
<A-Assets>
<video id="location1360">
<img id="icon1">
<audio id="pickupSfx">
....
Assets declared in HTML
reference assets with CSS ID selector
load each asset on scene init
emit music change event
Assets Loader
Music Manager
Navigation Manager
Transitions Manager
A-Scene Entity
<A-Scene> HTML element
parent
Injected 360º Background Entity
<A-Videosphere> 360º Video Background
Videosphere Components
Injected Entity
Navigation Sphere
Navigation Event Emitter
emit scene change event
parent
Injected Entity
Virtual Object
Other Components
Static Files
Scene JSON Descriptions
Dialogue JSON
3D Models Images Media Assets
reference filepath and object keys
reference filepath for voiced recordings
1 - GET local JSON file
2 - Parse JSON file into HTML A-Frame Entities
3 - Remove current <A-Scene> child elements
4 - Inject new child elements in <A-Scene> from parsed JSON

Figure 3.11: General Architecture of the Real Adventure - Illustrates how game scenes are parsed, rendered and organized. Components such as *transitions-manager* will be detailed in section 3.2.4

In Figure 3.11 it is possible to see how a scene comes into play. There are three ways to trigger a new scene injection into the game even if the process of injecting the elements of the new scene into the actual scene graph remains the same. These include:

- The scene being the initial environment of a particular game part/HTML
- Through an event emitted by navigation spheres when interacted with by the player
- Through content creator defined flag-based transitions

The *navigation-manager* component of the <*a-scene*> entity handles all of these possible cases by firstly requesting the necessary JSON file through an asynchronous HTTP GET request. Whenever this request is fulfilled the received JSON will be systematically parsed starting with root objects in the JSON (main background, combinations and transitions) and ending with the child elements of the main background and if necessary any other n-th level descendants. All current children of <*a-scene*> are removed from the DOM to make way for the newly parsed scene elements. The new scene elements will be automatically rendered by A-Frame as soon as their DOM injection concludes.

In essence the game scenes are the logical divisions of the game. A single cutscene will be a scene. A playable segment of the game may be a number *N* of scenes depending on how many different *N* locations it spans.

This means that if a single environment is needed for a playable segment, such a playable segment along with all the needed interactions can be defined in a single scene. If there are cutscenes in between a playable scene, it is possible through the flag-based transitions to go back and forth between a cutscene and the same playable scene, whilst keeping the game state consistent. If a scene has been previously parsed, Real Adventure will keep track of the changes that were already made to the scene through player interaction. Picked up objects will not re-appear and examined objects will treated as such when going back and forth between a set of scenes.

It is also possible to change the current video background without transitioning into an entirely different scene per the *navigation-manager* apparatus detailed earlier. This is made possible by flag-based transitions and it is useful if we want to make a more reactive game. An example would be having the player use a lighter from the inventory into some recorded real world object turned interactive and changing the background video into the same recorded environment but with that real world object burning in real life without having to remove and reintroduce the same virtual objects but described in a different JSON file with a different 360^{o} background. A more detailed look at flag-based transitions will be present in section 3.2.5.

- **Components** - These refer to the actual A-Frame components implemented as part of the Real Adventure framework that enable all the interactions and functionalities listed thus far. By simply including these components in different entities through the JSON description and passing the required data-attributes (if any) it is possible to add complex behaviour and functionality and create fully featured 360^{o} video based adventure games. More detail on the implemented components in section 3.2.4. Any game made in the framework would need to have these files in their project and include them in the game parts/HTML pages in order to use them in the descriptions of the game scenes.
- **Dialogue** - These are also described through JSON files and are referenced in the game scene files through their object keys and file path. More detail on how dialogue is described and used as well as all the possible functionality in section 3.2.5.
- **Assets** - Assets need to be stored in the directory of the game in whichever way the content creator decides as long as they are included in the assets of their corresponding game part/HTML page with the correct file path as exemplified in Figure 3.10.
- **Libs** - These are the external Javascript frameworks/modules that are required and used by the Real Adventure framework.

3.2.3 Scene Elements

The scene elements are those that belong in the game's actual scene graph, Figure 3.12 with a generic scene graph will be used as a basis for this discussion. Scene elements include typical visual elements such as 3D objects but may also refer to other more abstract elements such as a dialogue tree choices or even invisible elements that contain voice-overs that play either automatically or upon some sort of flag-based scripted activation. By understanding the disposition of scene elements in the scene graph it is possible to get a better general understanding on how a game made with Real Adventure works and is structured from a graphical point of view.

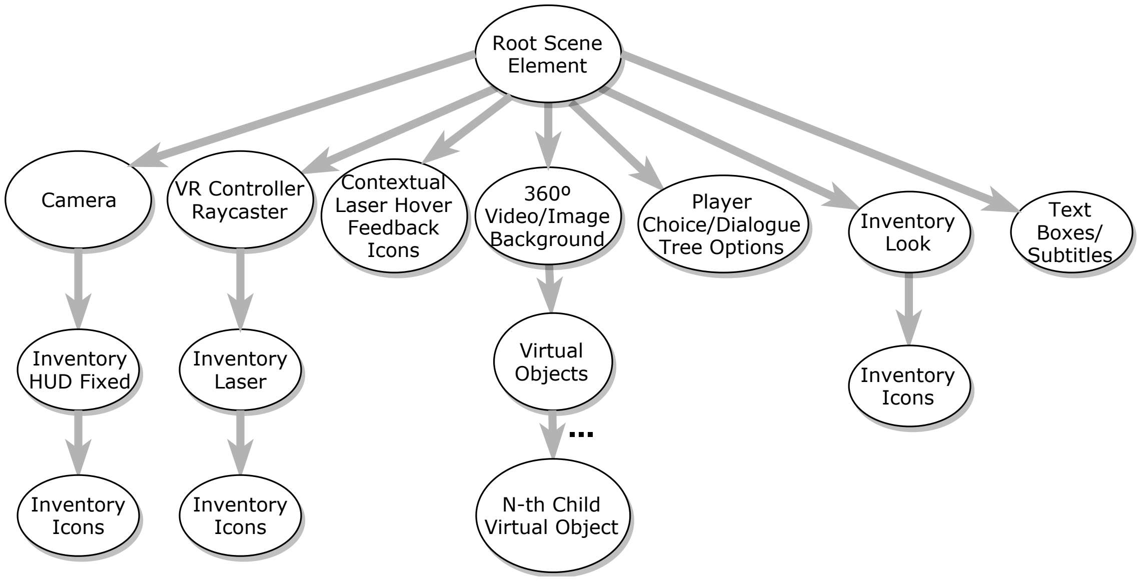


Figure 3.12: Generic Scene Graph

Camera

This scene element is a scene camera that is controlled by the rotational movement of the headset, with A-frame it is a camera entity with the component *look-controls*.

VR Raycaster Controller

Entity with A-Frame's *laser-controls* component that represents a VR controller and will allow any compatible input device to cast a laser out of a 3D model representation of itself in the game scene. This laser detects collisions with entities that fit a specified CSS selector.

360º Video/Image Background

This scene element is an entity that maps a 360º video or image into a sphere that will become the background of the game.

Virtual Objects

Virtual objects are entities that represent some object in the scene. These objects may be 3D models, 2D images or even invisible geometries to allow for interactivity with static video content. In the context of the Real Adventure framework, virtual objects have been previously categorized in table 3.1.

Contextual Laser Hover Feedback Icons

These icons are children of the scene element but they are only added if an interactive virtual object exists with a *hoverable* component, a component that provides visual feedback on laser hover. The icon remains invisible outside of laser collisions. There are five different possible icons, content creators are free to use whatever asset they want to represent the icons but they must be defined if the content creator wants contextual feedback on different types of virtual objects. Those five icons apply to the following:

- Pickable[8] objects
- Examinable objects
- Previously examined objects
- Code puzzle objects
- Previously solved code puzzle objects

Player Choice/Dialogue Tree

A graphical representation of story choices players may have to do throughout the game. Could also be a dialogue tree that instead of a singular one-time choice display a series of conversational topics for players to select and go through.

Text Boxes/Subtitles

Text boxes to display written for any reason (after interacting with an object for example) or subtitles over dialogue during recorded 360^{o} video.

Inventory

The inventory may appear at different locations depending on the chosen configuration by the content creator. Real adventure supports three different ways of displaying the inventory to the user and it all depends on the parent node of the inventory in the scene graph. All configurations of the inventory will decrease the opacity of the background to make the user focus on the inventory as other interactions besides the ones related to the inventory are disabled while it is opened. The possible inventory configurations are as follows:

- If the inventory is a child of the scene, it means the inventory will be displayed where the player is looking in the game environment the moment he clicks the dedicated inventory button. This is the default configuration of the framework
- If the inventory is a child of the camera, the inventory will be displayed in front of the player at all times no matter where he looks, as a HUD element as if glued to the player's field of view. This configuration may cause discomfort to some users in VR. A-Frame's own documentation advises against placing HUD elements glued to the camera

[8] collectable or pickable are interchangeable

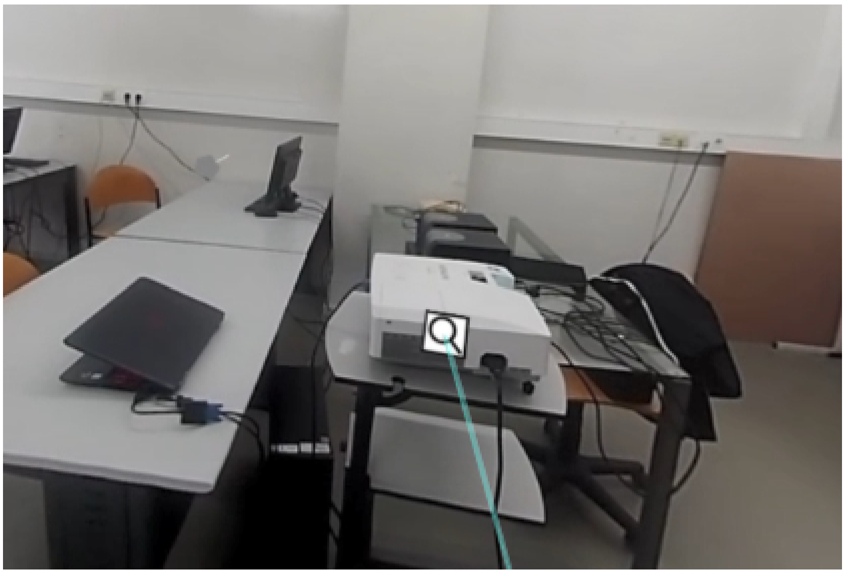
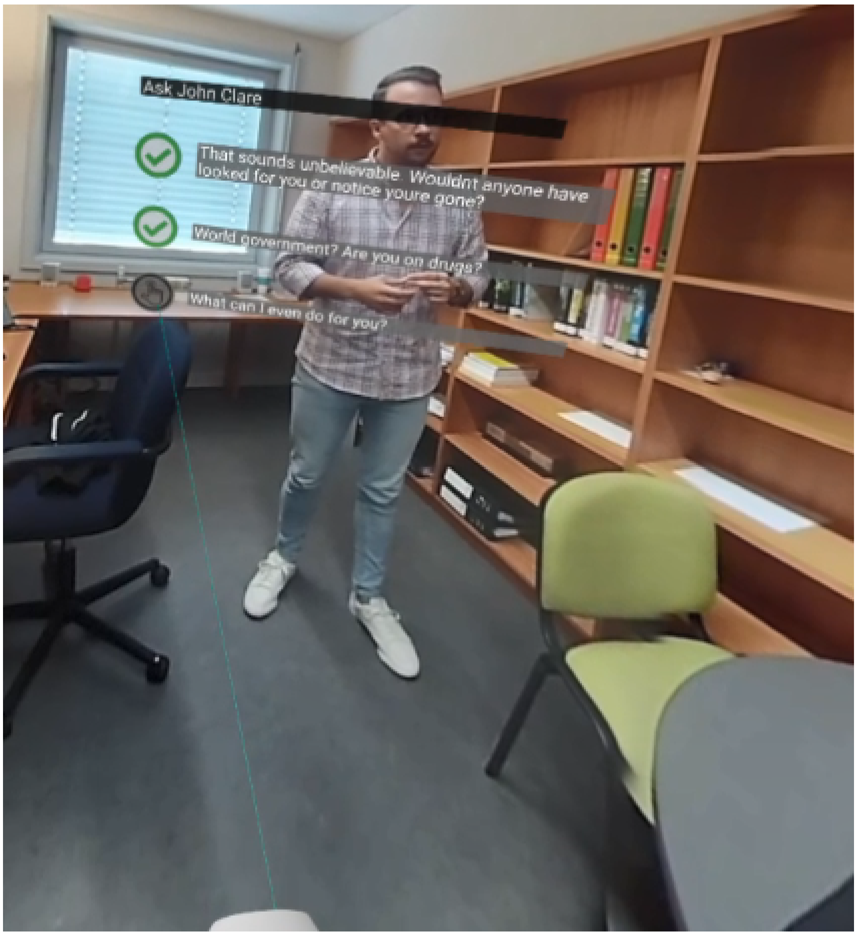


Figure 3.13: Examinable Object Hover Icon - Laser pointed at an examinable object in the environment, the hover icon feedback represented by a magnifying glass can be seen



Ask John Clare

✓ That sounds unbelievable. Wouldnt anyone have looked for you or notice youre gone?

✓ World government? Are you on drugs?

What can I even do for you?

Figure 3.14: A dialogue tree with several conversation topics

- If the inventory is a child of the raycaster, it will spawn where the laser is pointing in the world at the moment of clicking the dedicated inventory button at a configurable distance from the player's point of view

Inventory Icons

Each item in the inventory is represented by an icon. These icons are positioned relative the overall inventory container, as child elements of the inventory node in the scene graph. The overall look of the inventory can be seen in Figure 3.16. The inventory and corresponding icons look bigger when seen from a proper VR headset.

3.2.4 Framework Components

Several components were developed for the Real Adventure framework, which when attached to entities provide all sorts of functionalities. These components can be organized in several categories: Systems, Interactions, Media, Visual Modifiers and Miscellaneous.

Components are generally implemented as modular and completely independent software units but some of them may need to send events to be handled in other components (for example, send a navigation event to be handled by the *navigation-manager* as previously discussed) or coordinate through regularly updated data on the global state (as an example, the *dialogue* component will not trigger any text box or player choice interface if the inventory is flagged as open in the global state).

All of the components detailed in this section were implemented from scratch for the purposes of Real Adventure and were not available as existing A-Frame components,

Systems

System-types refer to more global scoped components, all of them are attached to the main scene entity (though not all components attached to the main scene entity are of this category) and they function as centralized systems that react to events from multiple sources and manage important functionalities such as navigation, flag-based transitions and music.

These could have been also implemented as actual systems but in terms of development it was more intuitive to see them as components of the scene instead of invisible background systems. There are also no performance gains from implementing them as systems vs components either. System components are detailed in table 3.4.

Table 3.4: System Components

Component	Description

navigation-manager	• Responsible for injecting new scenes into the DOM whilst removing any previous scene content if any • Listens to a custom DOM event for transitions into new scenes, receives new scene file path through event • Loads any existing saved game state from browser's *localStorage* happens on initialization • Receives an *initialEnv* as the entry point for that particular game part and it parses and renders that scene on initialization • It is possible to customize the duration of the transition fade effect (1000ms as a default)
transitions-manager	• Runs periodically while the game is running • By accessing the global state it checks if any of the content creator defined • transitions have met all the needed flag requirements to execute • Transitions are not executed if the inventory is open or if a dialogue is active (including player choice) among other conditions in the global state • These transitions are varied in nature. They can range from adding an object to the inventory to going into another game part/different HTML page. More detail on what transitions are possible in section 3.2.5
music-manager	• Manages the background music of the game, listens for change, pause and resume events • Fade effect for music transitions • Can cache current times of any music track and resume from the saved cached value if content creator desires to do so

Interactions

Interaction components are those that implement actual interactions such as hovering feedback on laser collision, making an object collectable or examinable, among others. The interaction components are detailed in table 3.5.

Table 3.5: Interaction Components

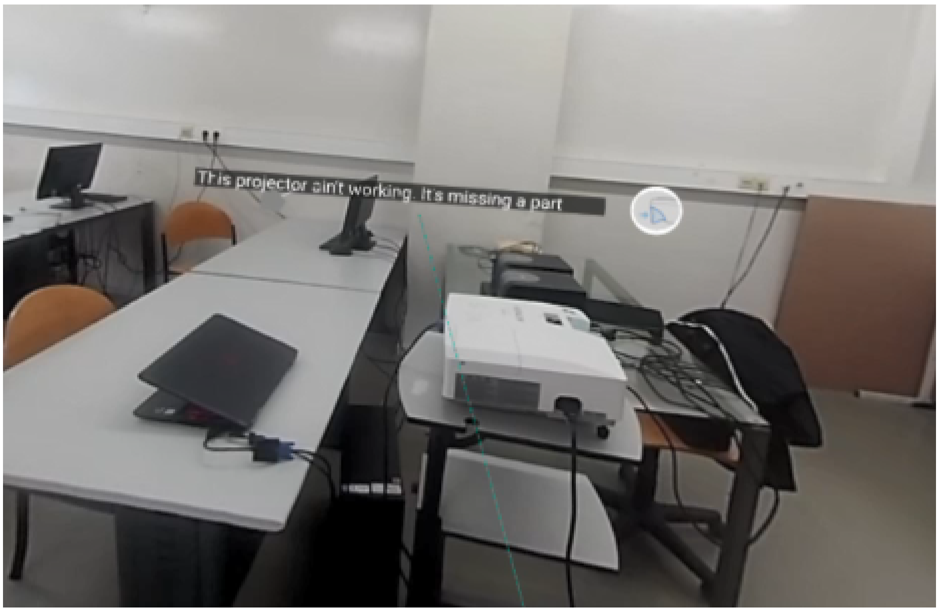
Component	Description

hoverable	• Provides feedback on laser collision events with the entity the component is present on • Default feedback type is halving the RGB values of the material's color so the object will look darker when collided with the laser. Content creators may opt to choose rotation or scaling as the hover feedback instead • May also have an icon in addition to the previous feedback by setting the *hoverIcon* attribute with a selector the desired icon asset in the HTML • Content creators may pass a sound effect for additional feedback on hover. Sound effects are passed through the *sfx* data attribute (a Javascript object with a source and volume key-value pairs) • Hover feedback is deactivated during cutscenes
pickable	• Allows an object in the game environment to be picked up by the player, adds object to player inventory stored in global state • Animation is active by default but may be deactivated by content creators. The animation makes the object fly into the camera before finally disappearing • Content creator needs to pass Javascript Object *inventoryData* composed of the DOM *id* of the new inventory object, a css selector to the image asset to be used as icon and a description for the inventory icon to be displayed in game • Content creators may also pass a sound effect through the data attribute *sfx* as explained earlier • Animation duration is customizable but 1000ms by default
codepuzzle	• Displays the code input interface seen in Figure 3.4 when entity it is attached to is interacted with • Content creators can customize the characters that appear on the numpad, the number of solution spaces displayed above the numpad and obviously the solution itself • Similarly to the *dialogue* component, updates the global state with a flag to restrict interactions until the code puzzle is either successfully solved or closed by the player
navigation-event-emitter	• Should be attached to navigation spheres but optionally can be attached to any other virtual object • Sends a navigation event that will be handled by *navigation-manager*. Event contains file path for the new scene to be transitioned into

dialogue/cutscene-player-choice	• Component that powers both examinable objects and dialogue trees/player choice • By default the dialogue interface is displayed where the player is looking, alternatively it could be a glued hud element that follows the player's POV, but as explained earlier that is not as comfortable for VR users • Receives the dialogue file path and a reference to the correct dialogue in that file (a dialogue file may define multiple dialogues) • A dialogue may be a simple sequence of texts or they may exclusively contain choices for the player/dialogue trees to navigate or any combination of all of these elements • If a specific dialogue line has a voice-over associated with it, the component will play it • Can receive multiple sound effects for dialogue advancement, player choice, hover player choice and spawning of the dialogue interface • Can be autoplayed (*autoplay* data attribute passed as true) or triggered on normal trigger button/click events (*examinableObject* attribute set to true). Dialogue autoplay is useful to display dialogue automatically when the component is attached to the background entity after a flag-based transition • The *cutscene-player-choice* component is essential a high-order wrapper of the *dialogue* component that allows player choice to be triggered at a specific timestamp during a 360° video cutscene • Content creators may pause the background music when dialogue is triggered with *pauseBackgroundSong* data attribute • Updates global state *dialogueOn* flag to restrict other interactions besides advancing the dialogue or making a choice if necessary until the dialogue ends and disappears from the scene

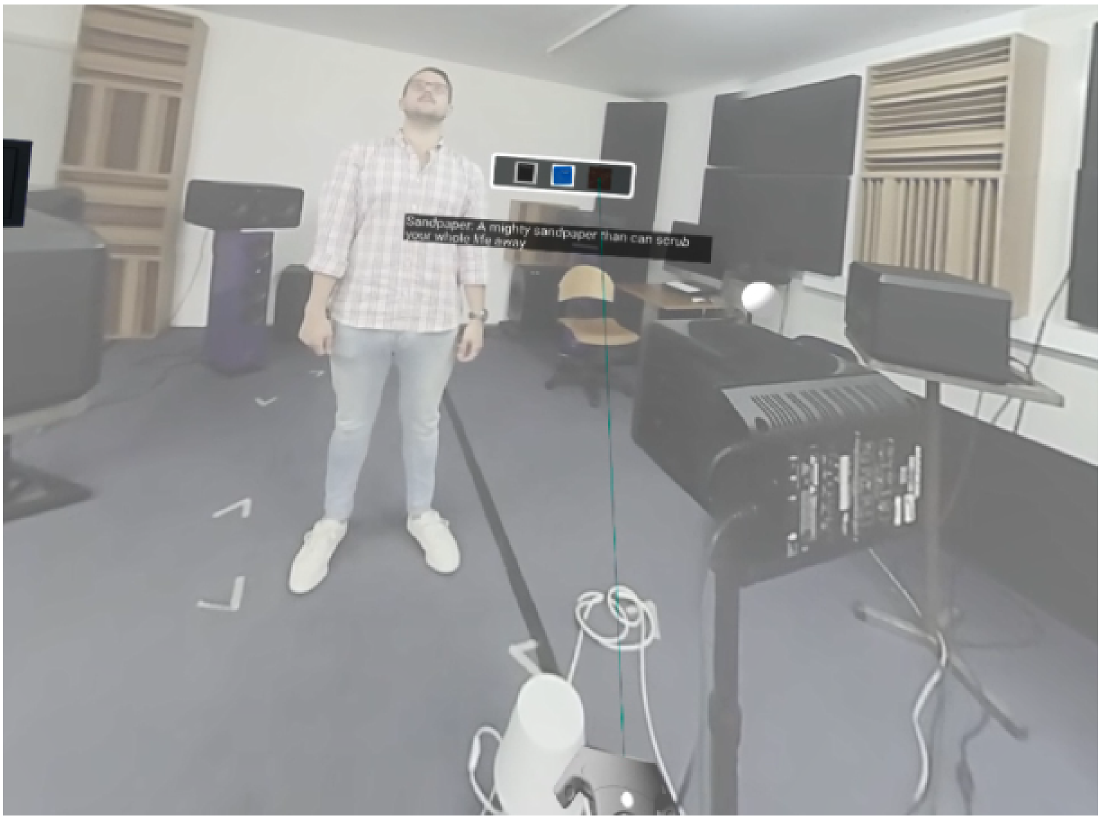
inventory	• Component attached to the root scene entity by default. Allows players to bring up inventory during gameplay by clicking the assigned button • Updates the global state of the game to restrict interactions outside of those with the inventory objects while open • Creates a modal like effect when open to focus player attention on the inventory. This effect is temporarily disrupted while inventory objects are being grabbed/dragged by the player • Does not need to be explicitly used by content creators. It should be present in the game parts/HTML pages template by default
grabbable	• Allows inventory objects to be dragged and dropped • When inventory objects are dropped into other inventory objects an attempt is made at combining those objects by checking for valid combinations in the global state • When inventory objects are dropped into an interactive object in the environment an event is emitted to that interactive object and if it has a *use-target* component an effect may occur • Does not need to be explicitly used by content creators. It is automatically used internally in the framework
present-item/cutscene-present-item	• Displays the inventory interface to prompt players to choose the correct item • If players choose the incorrect item the prompt will display again • Can be passed sound effects for wrong and correct choices • The *present-item* component can be activated by flag-based transitions while the cutscene variant triggers on a content creator defined timestamp of the 360^{o} video cutscene

use-target	• Listens to events sent by grabbable inventory objects • Receives a list of possible stimulus. Each stimulus is an object composed of a *targeted-by* attribute to validate that the correct inventory object was used and alongside some other exclusive or non-exclusive attributes Non-exclusive attributes include: • A new flag to add to the global state, content creator defined • Boolean *removeUsedObject* that will either keep or remove the used inventory object from the inventory itself • A sound effect for when the interaction happens and the intended volume for that sound effect Exclusive attributes include (meaning only one of these can be passed): • An *inventoryData* object similar to that of component *pickable* to add some object to the inventory as a result of the interaction. An example of this would be dragging a key inventory object to a safe and the contents of the safe would be added to the inventory as a result • A new asset source for the targeted environment object (a new 3d model or 2d asset) to represent that object alongside any other DOM attributes like the *id* to differentiate between the old and the new transformed object. An example for this use case would be burning a newspaper by dragging a lighter into it and thus a new object burned-newspaper replaces the old one • A dialogue to be dynamically attached to the background entity and autoplayed as a result of the interaction. An example use case for this would be letting the players know a certain interaction is invalid, such as when the player attempts to drag a lighter into a character in the game and a dialogue would appear to inform the player of the impossibility of the action with some humor thrown in for entertainment value
snapturn	• Automatically included in the game part/HTML page template, allows players to use dedicated buttons to turn the camera in fixed angles • It is possible to customize the angle increments by passing the angle value inline in the HTML pages



This projector ain't working. It's missing a part

Figure 3.15: Text Box Example - After pointing and clicking on the projector a text box appears, result of the player examining the projector. To advance the dialogue of the text boxes players need to click the trigger button as seen in the circle to the right.



Sandpaper: A mighty sandpaper than can scrub your whole life away

Figure 3.16: Inventory - Upon hovering an inventory object with the laser a description of the item will appear below the inventory.

Media

Media components are those that handle functionalities related to media such as 360^o video or audio playback. The implemented media components are detailed in table 3.6.

Table 3.6: Media Components

Component	Description
asset-loader	• On scene initialization it loads the music and video files from by iterating through the corresponding HTML media tags
video-player	• Plays the video (360^o background video or flat video inserted in the game environment) as soon as the scene initializes • If the video is supposed to be a *cutscene* or a *flatCutscene* (2d video injected in the game environment) there are boolean attributes that must be set when including the component in an entity • If it is a cutscene/flatCutscene the component will add a flag to the global state to reflect that the cutscene was seen by the player • If it is a cutscene/flatCutscene the component will update the global state in order to restrict interactions while the video is playing • It is also possible to pause the background song while the video plays, end the video earlier than the actual duration of the video and adjust the volume through specific component data attributes
video-looper	• Loops the video at passed timestamps for the beginning and end of the loop. By default loops the entirety of the video if no data attributes are passed
subtitles	• Receives the file path for a .srt file and when attached to the 360^o video background entity displays in text boxes the spoken dialogue according to the defined timestamps in the file
music-change-emitter	• Sends a music change event to be handled by the music-manager. Within the event it contains a selector for the new song to be played, the intended volume and whether the duration for that new song should be cached

scripted-audio-player	• Similar to the dialogue component, this may be attached to an object and triggered upon interaction or alternatively it could be added to the game's background entity and autoplay either in the beginning or later through flag-based transitions • Similar to the video-player element it adds a flag to the global state when over to represent that the audio was heard by the player • The playback of the audio can be delayed in seconds and the volume is adjustable through their corresponding data attributes • A data attribute called *exclusive* will make that audio have priority over any other. If a non priority audio is playing when an exclusive audio is triggered, the non exclusive audio will be interrupted. However if any exclusive audio is playing, it will finish its playback even if another exclusive audio is triggered in the meantime

Visual Modifiers

Visual Modifiers are those that relate to the appearance of the object. This includes simple animations, transformations and visibility management. The implemented visual modifier components are detailed in table 3.7.

Table 3.7: Visual Modifier Components

Component	Description
scripted-invisibility	• A component that allows us to control the visibility of virtual objects through DOM events • Through the *initialVisibility* data attribute it is possible to set the initial visibility state • When making the object visible it is possible to also change its position over time by defining the initial position, the target position and the duration of the animation. In the proof of concept of this dissertation this functionality was useful to make a newspaper appear from above and fall on the ground
invisible	• If we want to make a moving target in recorded video an interactive object we need to have a geometry that is only visible when the game is not in VR mode as to adjust its position correctly in the JSON description • When the game is actually running that geometry is no longer visible

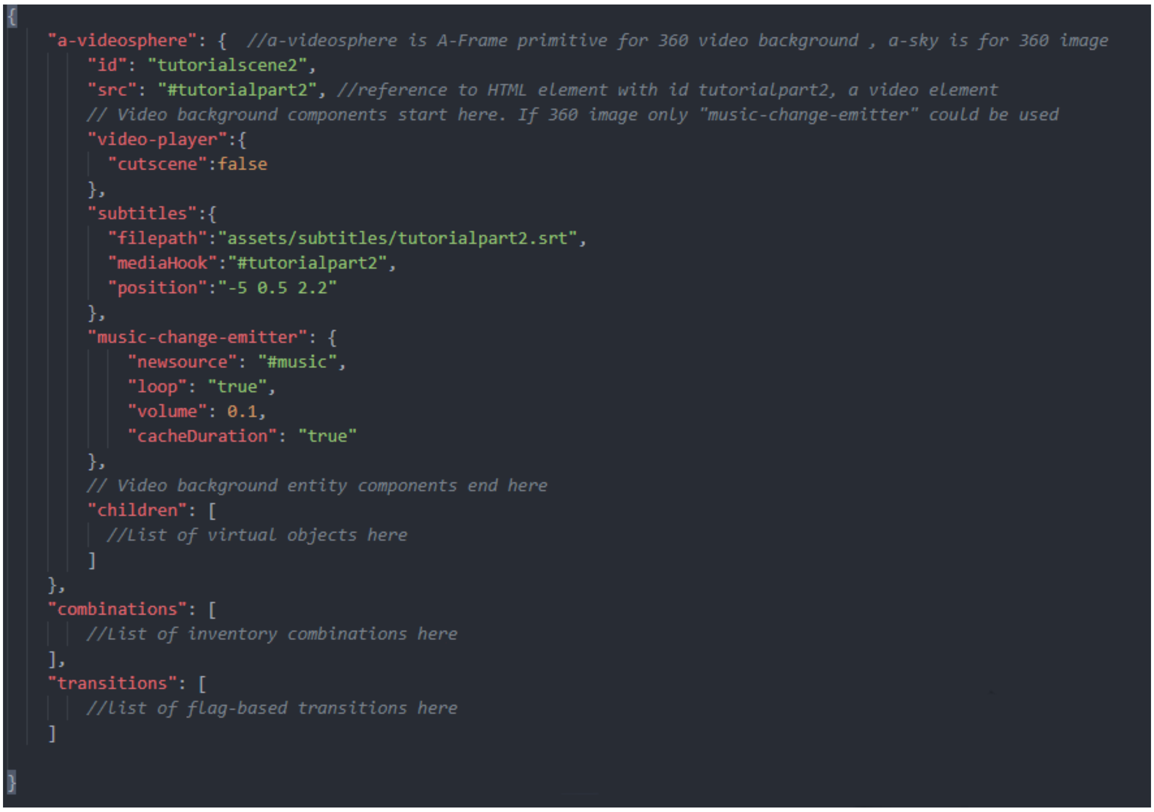
keyframes	• If we want to make a moving target in recorded video an interactive object we need to define the movement keyframes for the invisible geometry that provides the interactivity to that static video element • By passing an array of frames that contain the desired position of the virtual object in a specific timestamp of the 360 video it is possible to accompany the movement of the moving target in recorded video

Miscellaneous

There is only one miscellaneous component, the *guide-widget*. Whenever present as a component of an entity *X* - if the player does not contain *X* within the field-of-view - an arrow will be displayed in the game world pointing to X. This is done to guide the player's attention in the scene.

This is useful in scenarios such as when a video cutscene is playing and the player should be looking at the action (the 360^{o} subtitles text box would be the entity with the *guide-widget* component), when there is a choice to be made and the user is not looking at the player choice/dialogue tree interface or when there is some active unread dialogue.

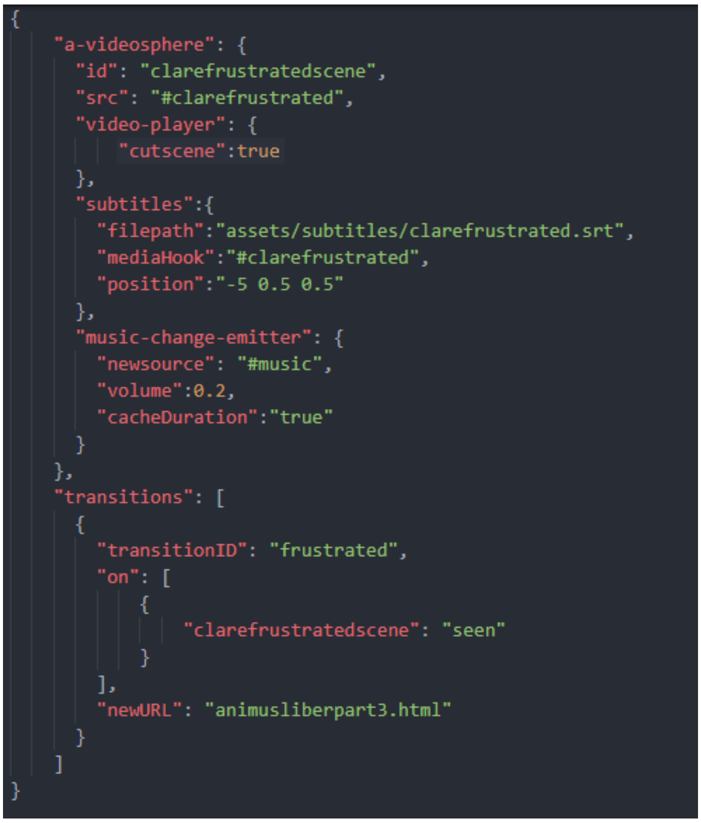
3.2.5 Descriptive Language - Real Adventure JSON



```
{
    "a-videosphere": {  //a-videosphere is A-Frame primitive for 360 video background , a-sky is for 360 image
        "id": "tutorialscene2",
        "src": "#tutorialpart2", //reference to HTML element with id tutorialpart2, a video element
        // Video background components start here. If 360 image only "music-change-emitter" could be used
        "video-player":{
          "cutscene":false
        },
        "subtitles":{
          "filepath":"assets/subtitles/tutorialpart2.srt",
          "mediaHook":"#tutorialpart2",
          "position":"-5 0.5 2.2"
        },
        "music-change-emitter": {
            "newsource": "#music",
            "loop": "true",
            "volume": 0.1,
            "cacheDuration": "true"
        },
        // Video background entity components end here
        "children": [
          //List of virtual objects here
        ]
    },
    "combinations": [
        //List of inventory combinations here
    ],
    "transitions": [
        //list of flag-based transitions here
    ]

}
```

Figure 3.17: General structure of a JSON description of a game scene



```
{
    "a-videosphere": {
      "id": "clarefrustratedscene",
      "src": "#clarefrustrated",
      "video-player": {
          "cutscene":true
      },
      "subtitles":{
        "filepath":"assets/subtitles/clarefrustrated.srt",
        "mediaHook":"#clarefrustrated",
        "position":"-5 0.5 0.5"
      },
      "music-change-emitter": {
        "newsource": "#music",
        "volume":0.2,
        "cacheDuration":"true"
      }
    },
    "transitions": [
      {
        "transitionID": "frustrated",
        "on": [
            {
                "clarefrustratedscene": "seen"
            }
        ],
        "newURL": "animusliberpart3.html"
      }
    ]
}
```

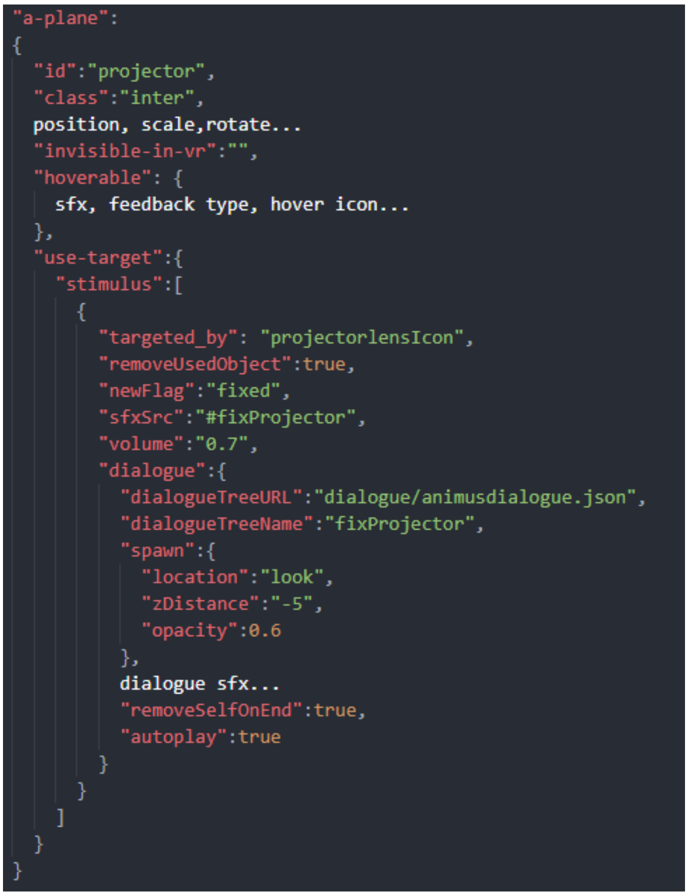
Figure 3.18: JSON definition of a cutscene with automatic transition into the next scene

In the previous sections, the general inner workings of the framework and the components that implement all the possible functionalities were discussed. That was the necessary context to finally explore how content creators can make games with Real Adventure with the JSON descriptive language. To aid in this overview it is necessary to analyze some examples. Firstly in Figure 3.17 we have the general structure of a scene defined with JSON.

The first level of the JSON description should contain the main background which is either an *a-videosphere* or an *a-sky* entity for 360^{o} video and image respectively. These entities are so called A-Frame primitives, basically entities that already have some pre attached components and a well defined purpose. We could create a sphere by having an *a-entity* with the component *geometry* with the data attributes necessary to make it a sphere or it would be possible to simply use an *a-sphere* primitive directly without the need of including the previously mentioned components/data attributes as they are already present by default in the primitive.

Aside from the main background it can also optionally contain *combinations* which define the possible combinations between inventory objects and *transitions* which allow content creators to describe flag-based transitions.

All the virtual objects of the scene would be listed as children of the main background, *a-videosphere* in the case of Figure 3.17. To see how it all comes together, observe Figure 3.18 carefully for a definition of a simple scene that contains a cutscene that plays and upon ending automatically transitions into some other scene through a flag-based transition.



```
"a-plane":
{
  "id":"projector",
  "class":"inter",
  position, scale,rotate...
  "invisible-in-vr":"",
  "hoverable": {
    sfx, feedback type, hover icon...
  },
  "use-target":{
    "stimulus":[
      {
        "targeted_by": "projectorlensIcon",
        "removeUsedObject":true,
        "newFlag":"fixed",
        "sfxSrc":"#fixProjector",
        "volume":"0.7",
        "dialogue":{
          "dialogueTreeURL":"dialogue/animusdialogue.json",
          "dialogueTreeName":"fixProjector",
          "spawn":{
            "location":"look",
            "zDistance":"-5",
            "opacity":0.6
          },
          dialogue sfx...
          "removeSelfOnEnd":true,
          "autoplay":true
        }
      }
    ]
  }
}
```

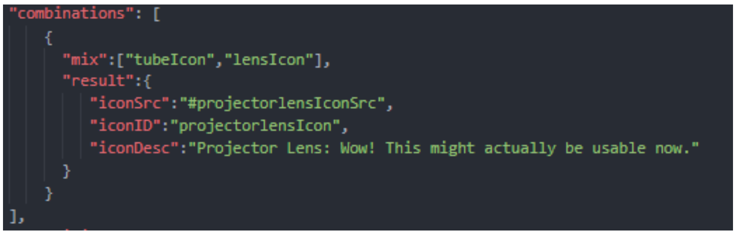
Figure 3.19: JSON definition of a recorded video element made examinable object

In this definition we can see the main video background containing the *video-player* component with the data attribute that sets it as a cutscene and thus when the video finishes playing a flag will be added to the global state in the key-value format entityID-"seen" (the value of the flag for cutscenes is by default "seen").

Since the entity that houses the *video-player* component has *id* "clarefrustratedscene" the flag will be saved in the key-value pair "clarefrustratedscene"-"seen". This flag is then used by the defined transition of *id* "frustrated" to navigate automatically to the next scene.

Additionally we can also see that the background entity has the component *subtitles* that will display the subtitle text at the desired location.

In Figure 3.19 we have a description of a recorded element in 360º video (in this case a projector as seen in Figure 3.6) turned interactive through an invisible plane with proper transforms to position it correctly in the environment. Finally, this entity also contains the *use-target* component that will allow us to drag the inventory object with *id* "projectorlensIcon" into the recorded projector and effectively express the use of the object by removing it from the inventory and triggering a dialogue that will let the player know the projector has been successfully fixed. The dialogue is the consequence of the stimulus in the *use-target* component and it autoplays as soon as the player completes the action. The anatomy of how a dialogue is described in JSON will be explored later



```
"combinations": [
    {
      "mix":["tubeIcon","lensIcon"],
      "result":{
        "iconSrc":"#projectorlensIconSrc",
        "iconID":"projectorlensIcon",
        "iconDesc":"Projector Lens: Wow! This might actually be usable now."
      }
    }
],
```

Figure 3.20: Combinations in JSON - The key *iconID* becomes the DOM *id* of the resulting object so that it can be used in other contexts of the JSON definition.

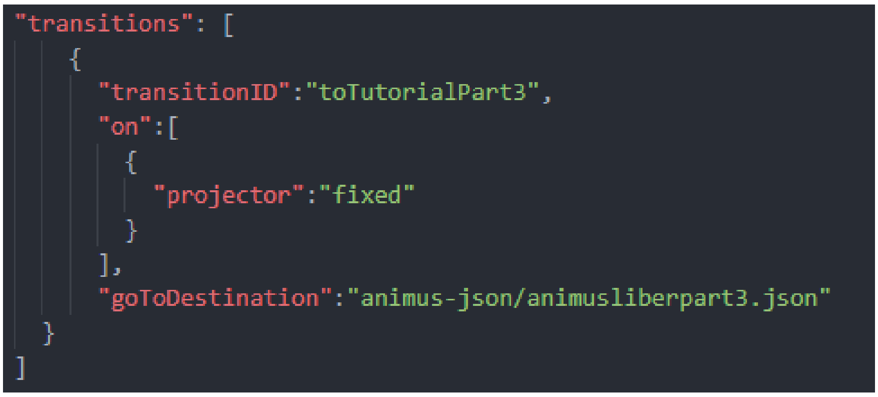
in this section.

To conclude the general overview of Real Adventure JSON it is necessary to explore how combinations between different objects in the environment are described. In Figure 3.20 we can see the needed structure. Generally we can have a list of combinations and each combination is an object with key *mix* which is a simple list of the two IDs of the inventory objects that can combine and lastly key *inventoryData* that as previously explained allows us to describe a single inventory object with its DOM *id*, CSS selector to the image to be used as icon and the inventory description. The specific combination in Figure 3.20 defines the very same object that will used in the interaction featured in Figure 3.6.

3.2.5.1 Flag-based transitions

When players execute certain actions in the game, the internal logic of the framework will automatically add a flag into the global state that can be used by content creators to script automatic transitions. Most flags are automatically added with default values but in a few contexts game creators might need to define the flag that needs to be added. Flags are added in the following contexts when:

- An object is examined
- An object is collected
- The player makes a choice
- A new environment is visited
- A code puzzle is solved
- A new object is created through combining of objects
- An item is presented
- A cutscene has been seen to completion
- A scripted audio track (not music) has been heard to completion
- Objects are used (if game creators defined a flag to add in use-target component)



```
"transitions": [
    {
      "transitionID":"toTutorialPart3",
      "on":[
        {
          "projector":"fixed"
        }
      ],
      "goToDestination":"animus-json/animusliberpart3.json"
  }
]
```

Figure 3.21: JSON definition of a flag-based transition

Transitions are added to the global state at the parsing of the the JSON file before the scene is actually injected into the game's scene graph. Each transition should have its own *id* and the component *transitions-manager* will periodically be running through the current list of transitions in the global state and execute them if the necessary conditions are met. Not only are the flag conditions needed but the inventory or any code puzzle interface must be closed, additionally any ongoing cutscenes or dialogue must finish before a transition is actually executed.

In a single transition it is possible to have multiple non-exclusive effects and a single exclusive effect, these effects are detailed in table 3.8. It is also possible to have transitions that happen without any flags, in essence instantaneous transitions, by simply leaving the value of the *on* key present in each transition object (Figure 3.21) to be an empty array.

If we want to have a sequence of exclusive effects based on the same set of flags it is possible to sequence them by simply ordering them in the JSON in the same order we want them to execute. For example, on an hypothetical flag "puzzleX"-"solved" we can define multiple transitions such as one that transitions into a cutscene and another that triggers some dialogue and the cutscene will always play out before the dialogue triggers unless we change the order in which they are defined in the JSON.

Lastly it is also possible to only do transitions if some flags are not present in the global state by defining an *ifnot* key-value pair in a transition object. The value of the aforementioned key-value pair would be in itself another object with several key-value pairs of all the flags that must not be present in case we want to execute the given transition. This is useful for dialogue trees with recorded video vs text (meaning the game responds with videos instead of text to player choices). It is useful in case a content creator wants to present the same choices at the end of a response in video format to some chosen dialogue tree option by the player. Until all topics of conversation are explored it is necessary that the dialogue choices are not displayed again at the end of the video that was the response to the last unexplored topic by the player. To put it simply, the dialogue tree will only display again at the end of a response if it is not the very last option the player had to explore.

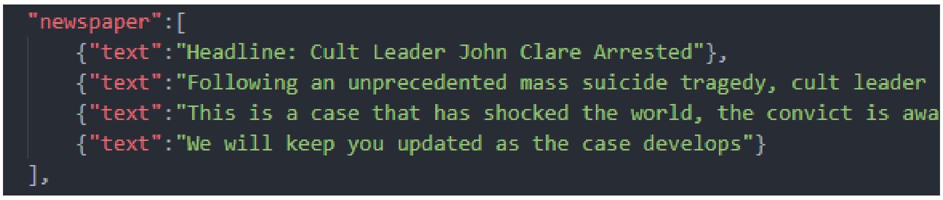
Table 3.8: Transition Effects - Exclusive and Non-exclusive effects

Transition Effect	Description	Type
Delay Transition	The transition executes after a delay in seconds	Non-exclusive
Add to inventory	The transition adds some object to the inventory, needs defining of inventoryData for new object	Non-exclusive
Clear inventory	Erases all current inventory objects	Non-exclusive
Play audio	Injects a *scripted-audio-player* component into the root scene element that will instantly play some audio (autoplay should be set to true for these transitions otherwise just use the component in some virtual object). It is possible to customize the attributes of the *scripted-audio-player* as described before	Non-exclusive
Make visible	Sends force-visibility event to some virtual object of a given *id*. If that object has the *scripted-invisibility* component it will handle this event accordingly and make the object visible	Non-exclusive
Make invisible	Sends force-invisibility event to some virtual object of a given *id*. If that object has the *scripted-invisibility* component it will handle this event accordingly and make the object invisible	Non-exclusive
Reset transitions	Clears the global state of all current transitions. Might be useful to cleanup the global state if there are leftover transitions that did not trigger (there may be different transitions for different player actions)	Non-exclusive
Inject flat video	Injects a 2D video into game environment, needs to be passed data attributes such as the transforms and others to customize the behaviour of this injected 2D video. Triggers a flat cutscene.	Exclusive but concurrent with change background source
Change background source	Changes the background 360^{o} video or image by defining a new media source without transitioning into a different scene with a different JSON file	Exclusive but concurrent with inject flat video

Prompt present item	Triggers an interface for the player to choose a specific item from the inventory. Injects a *present-item* component in the scene to achieve that and needs to be given the *id* of the correct item as a data attribute, sound effects for correct and wrong answers and other customizations as discussed in table 3.5	Exclusive
Trigger Dialogue	Triggers a dialogue. This is achieved by injecting a *dialogue* component and it is customizable as discussed in table 3.5	Exclusive
Change scene	Send event to navigate into a different scene. This event is handled by the navigation-manager of the scene as discussed in table 3.4	Exclusive
New URL	Transitions the game into a different game part/HTML page	Exclusive
Pause background video	Can only be done if the background 360° video is not defined as a cutscene as transitions cannot occur during cutscenes. Sends event that will pause the video at a given timestamp	Exclusive
Resume background video	Can only be done if the background 360° video is not defined as a cutscene as transitions cannot occur during cutscenes. Sends event that will resume the video if paused	Exclusive

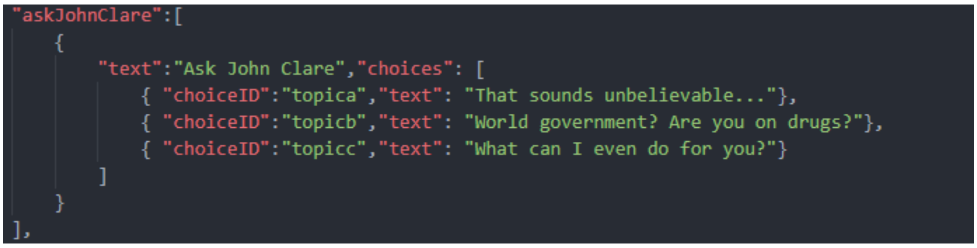
3.2.5.2 Dialogue

Dialogues are defined in their JSON files and referenced in scenes by the file path and the name of the wanted dialogue in that specific file. The way a simple dialogue composed of a sequence of text boxes is defined is very simple. A dialogue is an array of JSON objects that may have multiple



```
"newspaper":[
    {"text":"Headline: Cult Leader John Clare Arrested"},
    {"text":"Following an unprecedented mass suicide tragedy, cult leader
    {"text":"This is a case that has shocked the world, the convict is awa
    {"text":"We will keep you updated as the case develops"}
],
```

Figure 3.22: JSON definition of a simple dialogue



```
"askJohnClare":[
    {
        "text":"Ask John Clare","choices": [
            { "choiceID":"topica","text": "That sounds unbelievable..."},
            { "choiceID":"topicb","text": "World government? Are you on drugs?"},
            { "choiceID":"topicc","text": "What can I even do for you?"}
        ]
    }
],
```

Figure 3.23: JSON definition of a dialogue tree with options for player choice

key-value pairs. The core of dialogues is the key *text* whose value will be the contents of a text box, see Figure 3.22.

Player choice/conversational dialogue trees can be in turn defined with the pair of key *choices* (as seen in Figure 3.23) that contains an array of possible choices to present. A choice is composed of an *id*, the corresponding choice text and optionally a key *next* (as seen in Figure 3.24) with a value that references a *label* for the next piece of text in case a player chooses that option for use in fully text-based dialogue trees. In opposition to a mixture of presented options in text and in-game responses to player choice with video which should be handled through flag-based transitions as discussed previously.

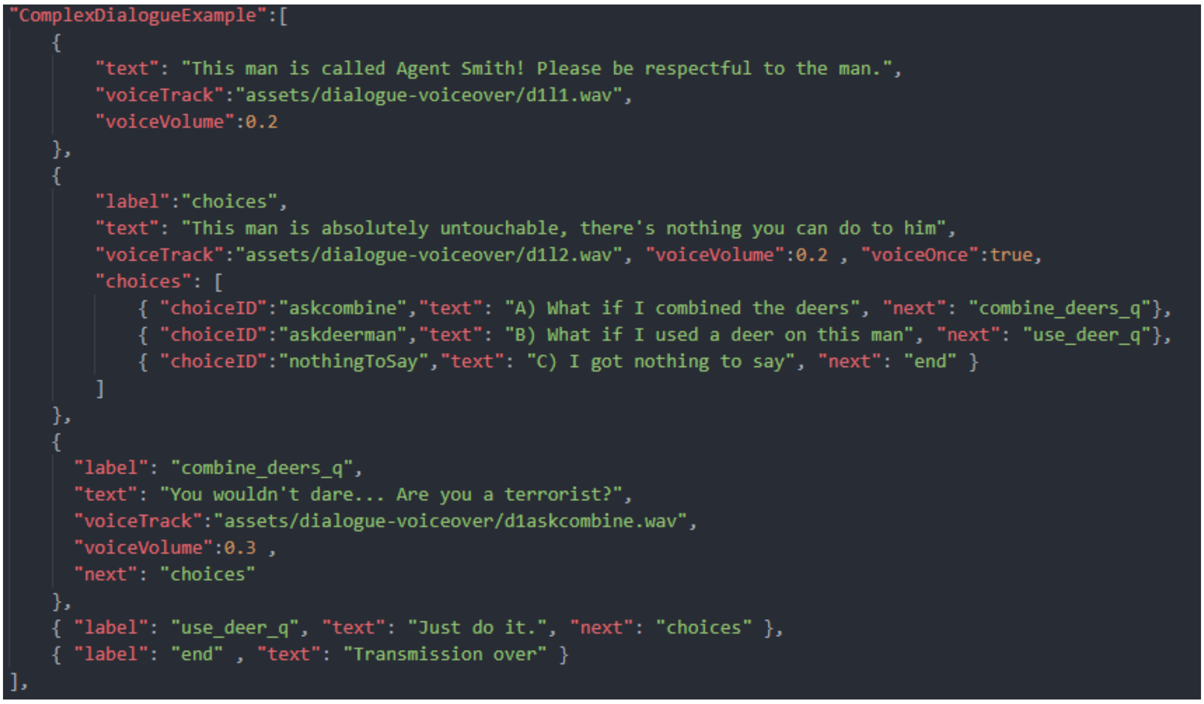
More advanced uses of the dialogue include having the aforementioned *label* for a specific text box and transition into different pieces of the dialogue by using this *label* on the *next* attribute in the choices. This mechanism makes it possible to create circular dialogue trees. An example of this is present in Figure 3.24. In this example when the player chooses the first option of *choiceID* "askcombine" he will advance to the text with *label* "combine_deers_q" and in turn the next step after reading the text in "combine_deers_q" is taking the player back to the original choice point with *label* "choices". The second choice features similar behaviour. Only by picking the third choice will the dialogue enter a linear sequence that will eventually end.

If this mechanism is not used, upon advancing the dialogue, the index of the overall dialogue array will be used instead to determine which piece of dialogue is next.

A piece of dialogue may also have a voice track associated with it. It is only necessary to include the file path of the desired voice track and it will play out in-game whenever that dialogue is presented.

3.2.6 Development Methodology

The development of the framework followed an iterative methodology. Whenever new components were implemented into the framework they would be tested by using them in test scenes and they would be refined not only in terms of stability but also performance and simplicity when translating them over to Real Adventure JSON. It was very important to make sure it was easy to use the component in the JSON. Interactions would be tested for usability with a select few volunteers and iterated on until a satisfactory state of usability.



```
"ComplexDialogueExample":[
    {
        "text": "This man is called Agent Smith! Please be respectful to the man.",
        "voiceTrack":"assets/dialogue-voiceover/d1l1.wav",
        "voiceVolume":0.2
    },
    {
        "label":"choices",
        "text": "This man is absolutely untouchable, there's nothing you can do to him",
        "voiceTrack":"assets/dialogue-voiceover/d1l2.wav", "voiceVolume":0.2 , "voiceOnce":true,
        "choices": [
            { "choiceID":"askcombine","text": "A) What if I combined the deers", "next": "combine_deers_q"},
            { "choiceID":"askdeerman","text": "B) What if I used a deer on this man", "next": "use_deer_q"},
            { "choiceID":"nothingToSay","text": "C) I got nothing to say", "next": "end" }
        ]
    },
    {
      "label": "combine_deers_q",
      "text": "You wouldn't dare... Are you a terrorist?",
      "voiceTrack":"assets/dialogue-voiceover/d1askcombine.wav",
      "voiceVolume":0.3 ,
      "next": "choices"
    },
    { "label": "use_deer_q", "text": "Just do it.", "next": "choices" },
    { "label": "end" , "text": "Transmission over" }
],
```

Figure 3.24: JSON definition of a dialogue with all the possible functionalities

When a component/interaction or feature had all the necessary functionality implemented and was properly tested and iterated on, it would be time to move on into the implementation of a new component/interaction or feature.

Generally speaking, it was also a concern to keep components as modular and detached from each other as possible. There were two primary techniques that were very useful to achieve a modular component design.

- By having access to a shared global state it is possible to save and access information about the state of other components and implement restrictions on certain interactions such as not being able to examine or pickup an object whenever the inventory was open.

- The use of custom and existing DOM events were also employed where appropriate to ensure the proper functioning of more global scoped components such as the navigation or music manager. The use of events allows for unrelated components to essentially "communicate" and react to those "communications". It is possible to trigger an event in different entities in the scene from any component (regardless of the entity the component is attached to) by simply executing a DOM query to get that entity and emitting an event that can originate from and be handled in by the very same queried entity.

3.3 Summary

In the conception phase of the framework, by outlining system requirements and the expected game creation workflow and output of the projected framework, it was possible to look for development tools and technologies with specific characteristics such as an easily translatable format into a descriptive language, a working and fully featured, cross browser and cross VR device framework among others. As a technology that fit the required characteristics, A-Frame was chosen for the development of the framework.

The general interactions and the types of virtual objects that needed to be supported based on the chosen game mechanics were delineated and used as a guide for the entirety of the development phase.

Following the conception phase we move into the development of the framework and with it an overview of the general system and architecture of the framework was presented. By explaining in detail the structure of a game produced in the framework it was possible to get a deeper understanding into the overall inner workings of the Real Adventure framework. This understanding comes from knowing how games are divided into parts and various scenes based on asset size limitations and the logical divisions of the game itself respectively. With general purpose scene components such as the navigation manager it is possible to quickly remove an existing scene and switch into another newly parsed scene from a JSON description through DOM operations.

Moreover the classification, detailing of the purpose and customizable data attributes of the various implemented components was shown followed by a comprehensive overview of the Real Adventure JSON specification and along with it a deeper dive into how to setup flag-based transitions and describe dialogue to be used in game. To wrap the chapter up the general iterative development methodology was discussed alongside techniques that were used to ensure a software design focused on modularity.

Chapter 4

Evaluation

In order to evaluate the general capabilities of the framework in producing content, as well as to investigate the proposed research questions, usability tests were conducted in order to ascertain if the chosen game mechanics for 360 video based adventure games were well suited. Additionally it was also important to get user feedback on the adequacy and efficiency of the general control scheme in this type of experience. Ultimately these usability tests allowed for an exploration of the entertainment potential both in terms of engaging gameplay and as a means of storytelling of 360 video based adventure games.

4.1 Proof of Concept Creation - Animus Liber

A proof of concept created with the framework was needed to explore all the intended research questions. This proof of concept needed to contain very succinctly an example of all the possible interactions and storytelling mechanisms that the framework would afford to potential creators to use in their creations.

The proof of concept needed to be small enough to be tested in under half an hour and also needed to include a complete narrative with branching paths and multiple endings to display the potential of the content that could be created with the framework.

The storyline was the first element that was created and developed. It was used as a base for the structure of the game. It can be read in its entirety in appendix B. Filming took place in a single morning with one actor followed by two days of video editing. 3D assets needed for the framework were created in Microsoft's Paint3D software[1] in 10-15 minutes per object with basic geometries, transforms and texturing. Finally the actual game was described/created with the framework in 2 days with half of the last day being used for testing and debugging.

Screenshots of the proof of concept have been used in most discussions related to the previous chapter.

[1] Paint3D, https://www.microsoft.com/en-us/p/paint-3d/9nblggh5fv99?activetab=pivot:overviewtab, last access 2019

4.2 Proof of Concept Description

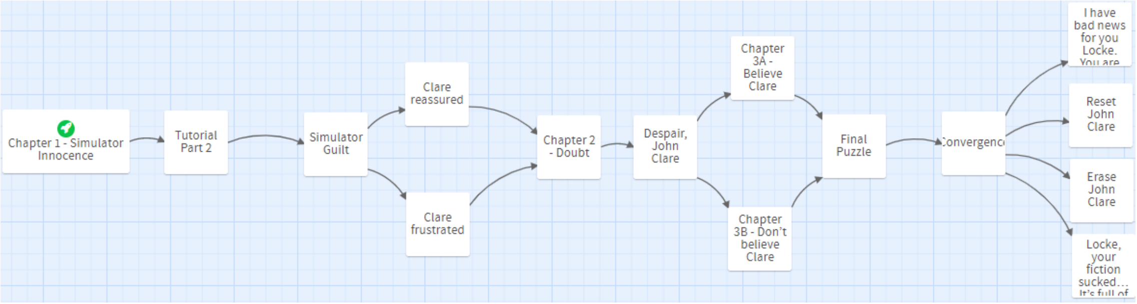

Chapter 1 - Simulator Innocence
Tutorial Part 2
Simulator Guilt
Clare reassured
Clare frustrated
Chapter 2 - Doubt
Despair, John Clare
Chapter 3A - Believe Clare
Chapter 3B - Don't believe Clare
Final Puzzle
Convergence
I have bad news for you Locke. You are
Reset John Clare
Erase John Clare
Locke, your fiction sucked… It's full of

Figure 4.1: Story structure of Animus Liber

The game is composed of a series of narrative moments with two puzzles in between. The first puzzle serves as a tutorial and introduction to the base game mechanics and the second puzzle features a more advanced challenge that will put the player's understanding of the mechanics to the test (Chapter 1 - Simulator Innocence and Tutorial Part 2).

The first puzzle has the player examining a recorded video object, a projector in a classroom and picking up a tube from a table nearby (Chapter 1 - Simulator Innocence). Afterwards the player is prompted to open their inventory and within it they have a tube and a lens. The inventory descriptions of these objects hint that they should be put together and by combining them it is possible to drag the newly made projector lens object into the projector in order to fix it (Tutorial Part 2).

After the first puzzle and by introducing the player to the navigation mechanic (allows them to switch environments by interfacing with a sphere in the environment as seen in Figure 3.1) a series of narrative events will happen where the player will have to make conversation with a character, an actual person acting as the character in 360^{o} video (Simulator Guilt). This bit will allow the players to interact with dialogue trees where they will have to select all the conversation topics before moving on to the next scene automatically. In another narrative sequence a character that appears in a flat 2D video in the game world will prompt the player to examine a newspaper that falls from above into the ground (Chapter 2 - Doubt).

Following these events the player will be prompted to make the first big narrative choice by selecting whether they believe a character or not and this will lead into different cutscenes depending on their choice (Despair, John Clare). These branching paths (Chapter 3A and 3B) will eventually converge after a few scenes into the final puzzle (Final Puzzle). From the final puzzle onwards the game is the same until the player has to make a choice at the end for one of four different endings (Convergence).

The final puzzle has two different locations. Players can use the navigation sphere present in each location at any time to switch between them. One location has a collectable notebook object and a safe with a code puzzle, the main goal of the players is to find the correct code to open the safe. The second location has an examinable object, a periodic table that upon examination will automatically add a note with the contents to the player's inventory. This note contains the number of protons of certain chemical elements, notably sodium (11 protons) and chloride (17 protons). A second collectable object is present in this location, a piece of sandpaper.

After completing these actions in any order by opening the inventory and by hovering on the icon of the notebook it says the cover of the notebook is peeling and that there seems to be something underneath. As such it is possible to combine the notebook with the sandpaper in order to uncover what is beneath. The resulting object is a scraped notebook that contains some text that hints at the combination of the safe being related to tears.

Players have to Figure out that since tears are salty they have to look at the number of protons for the elements that make up the chemical composition of salt (Sodium Chloride). By inputting the correct code, 1117, the safe will open, adding an object to the inventory and transitioning into the final scenes of the game.

In the choices for different endings if the player chooses the first option they will be prompted to present the correct item as evidence while other endings do not require further interaction (I have bad news for you Locke...).

4.3 Evaluation Protocol and Session Structure

The structure of the usability tests was conducted as follows. A user would come in, sign the informed consent (appendix C) and fill out a pre-experience questionnaire (appendix C) to get an understanding of their background on relevant topics such as previous experience with VR and adventure games. After a brief explanation of the general controls they would play the game (on HTC Vive or Oculus Rift) from start to finish with the possibility of asking for help or giving up altogether. To wrap up the usability session they would fill out a post-experience questionnaire (appendix C) that contains a mixture of open answer questions and Likert-Scale styled questions, but mostly questions of the latter.

4.3.1 Usability Questionnaires

The data from the pre-experience questionnaire contain information about the users such as their age, education, previous experiences with virtual reality and similar media to that of 360 video based adventure games (different types of adventure games, interactive movies and role-playing games).

Related to the user experience of the game are the questions present in the post-experience questionnaire. These questions focus on user experience of the game mechanics, engagement in terms of gameplay and storytelling and lastly on the potential of the format of 360 video based adventure games.

On the topic of user experience the questionnaire was designed to evaluate the general control scheme with some degree of granularity and the usability of the more complex interactions such as those related to the inventory system, interacting with objects in the environment and the code puzzle interface. These include questions inspired in the Likert-Scale with five levels that range from **very confusing** to **very intuitive**, a few open questions and a yes/no inquiry.

Regarding engagement the Likert-Scale questions are still in five levels but now they range from **strongly disagree** to **strongly agree**, the engagement questions try to ascertain the reasons why users were engaged or not engaged by the gameplay and the storytelling whilst also inquiring them about the desire to see alternative branches of the narrative by replaying the game.

Finally, the last few questions are more focused on the generality of the concept of 360 video based video games such as if users would be willing to pay for a commercial game developed in this style or if there are any mechanics they would like to see or remove.

4.3.2 Observations and Additional Data

Some observations were made during the usability tests such as when the users would struggle with the puzzles. Additional data includes the time each user took to complete the game and the number of times users needed hints to advance in the puzzles. Users were also asked if they felt any sort of sickness after the experience.

4.4 User Experiments

In this section the sample population for the usability tests will be described and characterized and the main results of the tests will be presented in preparation for the final analysis and discussion in the next section.

4.4.1 User Sample Overview

Overall, 28 users participated in the usability tests. The age of the users (table 4.1) ranged from 18 to 43 and the median age value was 23. Around 80% of the users had an education (table 4.2) level higher than high school and those who did not were at least enrolled in higher education at the time of the usability test.

All of the users had at least some previous experience with VR (table 4.3) with over 80% having enjoyed those previous VR experiences and on a scale of 1-5 with an average rating of 4. Within the sample 53% of users have experienced simulator sickness at least once.

Regarding previous experience with adventure games or media that involves branching narratives and user choice (table 4.2), 50% of the users had a previous positive experience with an adventure game, 21% were curious to try one of the adventure games in the questionnaire, 14% were completely unfamiliar with the genre and the remainder 15% either had no interest in or did not like adventure games.

When it comes to related media (RPG games with narrative choices or interactive movies/TV) 50% had a past positive experience, 17% were curious to try at least one example of this media and the remainder 33% either were unfamiliar with or had no interest in trying these experiences.

Table 4.1: Age demographics

Age	Participants
18-22	5
23-27	17
28-32	4
40-44	2
Total	28

Table 4.2: Education demographics

Education	Participants
PhD	2
Master's	12
Bachelor's	8
High School	6
Total	28

Table 4.3: Number of participants per experience type

Experience	Participants
Mobile VR	23
Desktop/Console VR	23
360 Video	24
VR Video Games	25
Total	28

Table 4.4: Related Media demographics

Media Type	Do not know	No interest	Did not like	Want to try	Liked
Adventure Games	4	3	1	6	14
RPGs with Player Choice	10	1	0	8	9
Interactive Movies/TV	4	4	1	11	8

4.4.2 Results

Before delving into the specific results of each question it is important to create an index to guide the reader of this dissertation. It is advisable to the reader for the optimal experience to read the analysis and discussion in tandem with each set of related results. Whenever it is intended to switch to the analysis and discussion of particular result there will be a reference to it.

- **Question 1 (Q1)** How would you rate the general game control scheme? - [Likert-inspired question](Ranges from 1-Very Confusing to 5-Very Intuitive) 28 users answered for each item:
  - Laser Pointing
  - Trigger button for general actions
  - Menu button to bring up inventory
  - Overall experience with control scheme

- **Question 2 (Q2)** If you had any problems with the general control scheme, briefly describe them. - [Open-question] 2 users answered
- **Question 3 (Q3)** How would you rate the ease of use of the inventory system? - [Likert-inspired question](Ranges from 1-Very Confusing to 5-Very Intuitive) 28 users answered for each item:
  - Combining objects
  - Using inventory objects in environment
  - Overall experience with inventory
- **Question 4 (Q4)** If you had any problems with the inventory system, briefly describe them. - [Open-question] 5 users answered
- **Question 5 (Q5)** How easy was it to distinguish between the different interactions with objects in the environment? - [Likert-inspired question](Ranges from 1-Very Confusing to 5-Very Intuitive) 28 users answered for each item:
  - Distinguish pickable objects
  - Distinguish examinable objects
  - Distinguish code puzzle objects
  - Distinguish objects overall
- **Question 6 (Q6)** If you had any problems distinguishing between different actions on interactive objects, briefly describe them. - [Open-question] 3 users answered
- **Question 7 (Q7)** How easy was it to use the code input interface used for locked safe puzzles? - [Likert-inspired question](Ranges from 1-Very Confusing to 5-Very Intuitive) 28 users answered
- **Question 8 (Q8)** If you had any problems with the code input interface, briefly describe them. - [Open-question] 1 user answered
- **Question 9 (Q9) / Observation 1 (O1)** Did you ever feel lost during the experience? - [Yes/No question] 28 users answered / Number of hints needed to finish the game - [Observation] 28 users observed
- **Question 10 (Q10)** If your answer to the previous question was yes, briefly explain when and why. - [Open-question] 14 users answered
- **Question 11 (Q11)** Regarding the gameplay, how do you feel about these items? - [Likert-scale question](Ranges from 1-Strongly disagree to 5-Strongly Agree) 28 users answered for each item:
  - Gameplay was simple and clear

  - Puzzles were fun
  - Interface was unobtrusive
  - Controls were responsive
  - Always knew what to do
  - Wish there was more of it
  - Gameplay was engaging overall

- **Question 12 (Q12)** If you thought the gameplay was not engaging, briefly detail why. - [Open-question] 5 users answered

- **Question 13 (Q13)** Regarding the story, how do you feel about these items? - [Likert-scale question](Ranges from 1-Strongly disagree to 5-Strongly Agree) 28 users answered for each item:

  - Story premise was intriguing
  - Enjoyed the twists
  - Dialogue captivated interest
  - I was a part of the story
  - Engaged because choices mattered
  - Acting was good
  - Sense of presence enhanced story
  - Story was engaging overall

- **Question 14 (Q14)** If you thought the story was not engaging, briefly detail why. - [Open-question] 4 users answered

- **Question 15 (Q15)** Overall did you feel engaged throughout the experience and motivated to see it through till the end? - [Likert-scale question](Ranges from 1-Strongly disagree to 5-Strongly Agree) 28 users answered

- **Question 16 (Q16)** Would you play it again to experience the other endings? - [Yes/No question] 28 users answered

- **Question 17 (Q17)** Why do you want to play/not play again? - [Open-question] 28 users answered

- **Question 18 (Q18)** If a game of this kind was available for purchase and it had a positive reception, would you buy it? - [Yes/No question] 28 users answered

- **Question 19 (Q19)** Are there other mechanics you would like to see in this type of game? - [Open-question] 9 users answered

- **Question 20 (Q20)** Do you have any suggestions for overall improvements? - [Open-question] 15 users answered

The proof of concept took users an average of 21 minutes to complete with standard deviation of 3 minutes. Only one of twenty-eight users felt a bit of simulator sickness during the experience.

The first results to be presented concern questions regarding the usability of the general control scheme (**Q1** and **Q2**). These results for **Q1** can be seen in table 4.5.

Table 4.5: Results for **Q1**: "How would you rate the general game control scheme?"

Question	1st Quartile	Median	3rd Quartile	Mean
Laser pointing	5	5	5	4.68±0.72
Trigger button for general actions	4	5	5	4.57±0.57
Menu button to bring up inventory	4	4	5	4.14±1.04
Overall experience with control scheme	4	4	5	4.25±0.8

There were two users with feedback in **Q2**. Even if these users still found the general control scheme to be intuitive there were still some minor issues that were highlighted. One user thought it was bothersome that the inventory did not close upon clicking with the trigger button outside the inventory window or close automatically when looking away from the inventory. The other user found some inconsistencies in some actions. While generally speaking all actions are done with the trigger button, some only require the button to be pressed while others require pressing and releasing of the trigger until the action is performed. For the analysis and discussion on this particular set of results please read section 4.5.1.

The next set of questions (**Q3** and **Q4**) relate to the inventory system and the interactions related to it. These include combining objects within the inventory and using objects from the inventory in the environment. The results for **Q3** are available in table 4.6.

Table 4.6: Results for **Q3**: "How would you rate the ease of use of the inventory system?"

Question	1st Quartile	Median	3rd Quartile	Mean
Combining objects	4	4	5	4.11±0.99
Using inventory objects in environment	4	4	4.25	4.04±0.74
Overall experience with inventory	3.75	4	5	4.04±0.92

Direct feedback from **Q4** was given by five users. The user who had suggested improvements to the inventory controls reiterated that it was bothersome to have to click the dedicated menu button again to close the inventory. Other users mentioned that they needed to try a few times to use the inventory object on the recorded video element made interactive and one user suggested the use of a snapping mechanism for when dragging and dropping inventory objects into either the environment or for combining with other inventory objects. For the analysis and discussion on this particular set of results please read section 4.5.2.

The following set of results concern questions **Q5** and **Q6**. These questions intend on evaluating the ease of distinguishing between different types of interactive objects through the laser hover feedback and generally interactive vs non interactive objects. The results for **Q5** can be seen in table 4.7

Table 4.7: Results for **Q5**: "How easy was it to distinguish between the different interactions with objects in the environment?"

Question	1st Quartile	Median	3rd Quartile	Mean
Distinguish pickable objects	3	4	5	3.86±1.08
Distinguish examinable objects	3	4	5	3.93±1.05
Distinguish code puzzle objects	3.75	4	5	3.93±1.05
Distinguishing objects overall	4	4	5	4.07±0.94

Q6 contains three pieces of direct feedback. One user mentions that it was confusing to tell apart pickable objects from examinable objects but also said that since the actions were automatic upon clicking this confusion did not matter or have much impact in the experience. Two other users said that it was difficult to tell the fake virtual objects apart from the actual recorded environment in the 360^{o} video and that there should be some kind of highlight to make them stand out. For the analysis and discussion on this particular set of results please read section 4.5.3.

Next up are questions **Q7** and **Q8** that relate to the code puzzle interface. These are still usability focused questions. The results for **Q7** are present in table 4.8.

Table 4.8: Results for **Q7**: "How easy was it to use the code input interface used for locked safe puzzles?"

Question	1st Quartile	Median	3rd Quartile	Mean
Ease of use of code puzzle interface	4	4	5	4.18±0.98

The paired question **Q8** had only one user response mentioning the lack of a backspace to fix an error during the input of the code. For the analysis and discussion on this particular set of results please read section 4.5.4.

Up until now the questions were more focused on the usability of the controls, interfaces and specific mechanics. It is now time to transition into more general game design questions and engagement metrics.

First up are **Q9 / O1** a combination of a question that asks the user if they ever felt lost and observations made during the tests regarding the number of times users needed hints to advance in the game and also **Q10**, a question that requests direct feedback from users as a followup to **Q9**. The results for **Q9 / O1** are displayed in Figure 4.2 and in table 4.9 respectively.

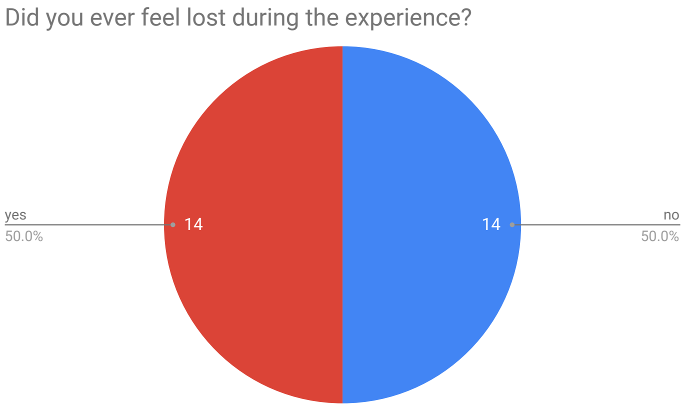


Figure 4.2: Results for **Q9**: "Did you ever feel lost during the experience?"

Table 4.9: Observations for number of needed external hints to finish the game

Observation	1st Quartile	Median	3rd Quartile	Mean
Number of hints needed	0	1	2	1.14±1.24

As for the reasons why users felt lost or needed hints throughout the experience, the data in table 4.10 was achieved by aggregating the observations as to why users needed hints with the responses to **Q10** by the users themselves.

Table 4.10: Reasons for feeling lost or needing external hints by number of participants

Reasons	Participants
Puzzle logic	7 (25% total users)
Could not find an object in the environment	7 (25% total users)
Did not know/forgot objects could be combined	5 (18% total users)
Felt disorientated when transitioning between environments	2 (7% total users)

Within the puzzle logic reason most of them relate to users not knowing the chemical composition of salt or for not making the connection between tears and salt. The full analysis and discussion can be read in section 4.5.5.

On engagement metrics, questions **Q11** and **Q12** were made with a focus on gameplay and trying to dissect which components of it the users appreciated. On the other hand questions **Q13** and **Q14** have a focus on the storytelling of the proof of concept as to give an idea of the potential of 360 video based adventure games in this regard.

Questions **Q15** to **Q17** are more general in nature. **Q15** seeks to understand if the user was generally engaged throughout the experience and motivated to finish it (their continuation desire).

Lastly, questions **Q16** and **Q17** inquire about the replayability of the experience and if the branching narrative gathered enough interest for users wanting to see alternative story paths. All of these results will be presented in tables 4.11, 4.12, 4.13, Figure 4.3 and finally table 4.14 consecutively.

As a reminder the all questions going forward are Likert-Scale, scaled from **Strongly Disagree** to **Strongly Agree**.

Table 4.11: Results for **Q11**: "Regarding the gameplay, how do you feel about these items?"

Assertion	1st Quartile	Median	3rd Quartile	Mean
Gameplay was simple and clear	4	4	5	4.11±0.79
Puzzles were fun	4	4	5	4.11±0.88
Interface was unobtrusive	3.75	5	5	4.25±0.93
Controls were responsive	4	4.5	5	4.36±0.78
Always knew what to do	3	3	4	3.29±1.01
Wish there was more of it	4	4	4	3.93±0.86
Gameplay was engaging overall	4	4	5	3.93±1.05

As for direct feedback as to why the gameplay was not as engaging as it could be, five users expressed their opinion in **Q12**. Four users expressed the sentiment that the ratio of gameplay/story was too unbalanced as the dialogues were too long and there were not enough puzzles. Another user expressed discontentment towards the initial tutorial as it was effectively one of the two puzzles in the game.

Table 4.12: Results for **Q13**: "Regarding the story, how do you feel about these items?"

Assertion	1st Quartile	Median	3rd Quartile	Mean
Story premise was intriguing	4	4	5	4.14±0.65
Enjoyed the twists	3	4	5	3.96±0.84
Dialogue captivated interest	3	4	4.25	3.86±0.89
I was a part of the story	3	4	4	3.71±0.85
Engaged because choices mattered	3	4	4	3.71±0.85
Acting was good	4	4	4	3.93±0.86
Sense of presence enhanced story	3.75	4	4.25	3.89±0.96
Story was engaging overall	3.75	4	4.25	4.00±0.72

The related question **Q14** had four responses, a few users said that they felt more like observers during certain scenes in the story as opposed to active parts of it because the actor was not looking directly at the camera. A couple of users also re-iterated here that the dialogues should have been shorter.

Table 4.13: Results for **Q15**: "Overall did you feel engaged throughout the experience and motivated to see it through till the end?"

Question	1st Quartile	Median	3rd Quartile	Mean
Motivated and engaged to finish game	4	4	5	4.25±0.80

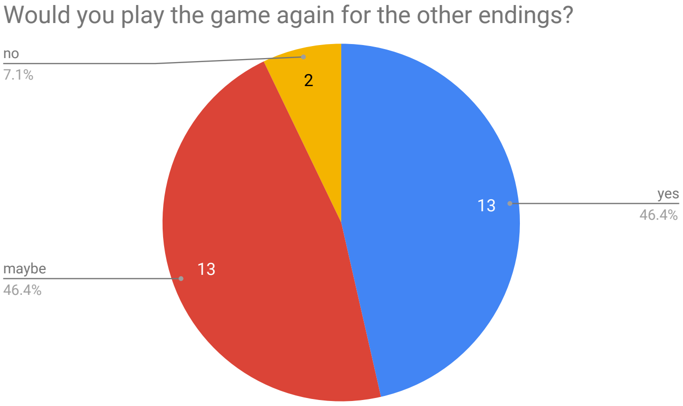


Figure 4.3: Results for **Q16**: "Would you play it again to experience the other endings"

Table 4.14: Results of **Q17**: "Why do you want to play/not play again?"

Reasons	Participants
Curious to see other choices/endings	17 (61% total users)
Would replay if there are alternative puzzles too	5 (18% total users)
Would replay if seen content is skippable	3 (11% total users)
Would not replay because it is too slow paced	2 (7% total users)
Would not replay because I prefer a 2D game over this	1 (3% total users)

The discussion for the topics related to engagement and replayability can be found in section 4.5.6.

The last few questions are related to the concept of 360º video based adventure games. They inquire if users would be willing to pay for a commercial game with positive critical reception in this style **Q18**, if there are any other mechanics users would like to see in this type of game **Q19** and finally suggestions for overall improvements **Q20**.

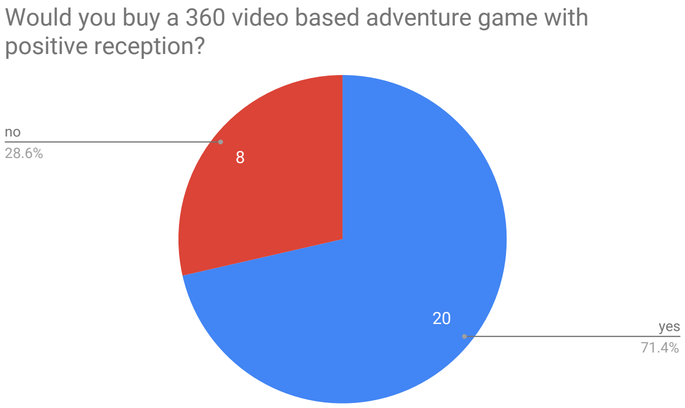


Figure 4.4: Results for **Q18**: "If a game of this kind was available for purchase and it had a positive reception, would you buy it?"

For Q19 suggested additional mechanics that would be possible under the limitations of a game in this format include a movement system within the same location akin to the Google street views, walking and voice commands/recognition.

For the final question **Q20** overall improvements suggested by users that have not been addressed in the discussion yet include a way to re-center interface elements that appear where the user is looking (dialogues, inventory and player choice), replacing the navigation sphere with a dedicated map interface to change locations (by using a second VR controller to bring up the menu interface, at present Real Adventure only makes use of a single controller) and more things to do in general. The final analysis and discussion can be read in section 4.5.7.

4.5 Analysis and Discussion

The discussion will ponder on the results of the usability tests and address general user feedback. It is important to keep in mind that any proposed changes to improve the experience within this discussion would need to be validated through another round of usability tests.

4.5.1 General Control Scheme

While users generally displayed a high level of satisfaction towards the usability of the general control scheme there were definitely two pieces of invaluable feedback from those who found minor issues. It seems like a better arrangement to have the inventory automatically close when a user looks away instead of displaying the guide widget to remind the user that the inventory is still open and that he should press the menu button again to close it. It is also a general usability practice that clicking away from an interface element will either minimize that interface or close it altogether and as such both of these options could be implemented to improve the interactions related to closing and opening the inventory.

Only one of the users offered this kind of feedback directly but the data also shows a higher standard deviation and lower overall mean in the scoring of the controls related to opening and closing the inventory. This points to the fact that overall opinions were more mixed on this particular item than the rest of the control scheme. That shows that there is certainly room for improvement.

The other piece of feedback highlighted an inconsistency in the way some actions only require the trigger button to be pressed while others require pressing and releasing that very same button. The solution here would be to choose only having to press the trigger button to trigger the action as it enhances the feeling of responsiveness of the game. This inconsistency happens because in code there were two different events that were wrongfully thought of as interchangeable (the generic DOM 'click' event and the controller specific 'triggerdown' event). Unfortunately this issue was not noticed during development and as such was not fixed in time for the usability tests.

4.5.2 Inventory System

On the matter of the inventory system, overall metrics were lower than those that concern the general control scheme but users still found the system to be generally intuitive. There were some users that did not pay much attention to the spoken dialogue in the beginning of the game that was intended to function as a tutorial and they were not aware that it was possible to combine objects without external intervention. While the majority of the users managed to pick up on this fact through either the actual tutorial or the clues in the inventory description of the tutorial items, others were left confused and that points to a usability problem as the game systems should be organically discernible through a more helpful interface.

A possible solution to alleviate or erase this matter altogether would be to create a dedicated interface for combining objects instead of dragging and dropping the inventory icons into other inventory icons. Having for example two empty slots below the inventory with some visual indication that it was an interface for combining objects would inform the user that it was a possible mechanic without having to explicitly tell them through a tutorial.

Regarding the use of inventory objects in external entities in the environment there was definitely a problem with recorded entities made interactive as they need the presence of an invisible interactive geometry and during the user experiments it was apparent that if the box was correctly adjusted on the Oculus Rift headset, it would be slightly misaligned in the HTC Vive. Not to the extent that it was impossible to interact with the object but to an extent that the invisible geometry no longer covered the entirety of the static video element intended to be interactive and while most users were fortunate to drag the object into a valid position, some were not.

There are three possible solutions to this problem. One of them would be to make the invisible geometry bigger to the point that the misalignment with different VR HMDs would not make a difference but this could be harmful in situations where there are multiple nearby video entities that a creator wants to make interactive. Not the ideal solution but one that would not require further modifications to the framework, a simple adjustment of the scaling of the invisible geometry in the JSON description would suffice.

A better and more precise solution would be to create generic calibrations and scaling profiles for each of the main headsets, thus relieving content creators of this problem.

The third solution is one suggested by a user, the snapping functionality. This means that if a player drags an inventory object in some location, the framework will have to scan the vicinity of where the player dragged the object for other interactive entities and effectively snap into the nearest one if within an acceptable distance. Ideally both the second and third solutions would be implemented for a much improved user experience.

4.5.3 Distinguishing Objects

Again, the general results regarding this topic display an acceptable level of usability. Addressing the first piece of feedback it would be simple to differentiate examinable objects from those that are collectable by simply designing a better icon for hover feedback of pickable objects. The current icons are a magnifying glass for the examinable objects and a hand with the fingers suggesting a gesture of picking something up. Perhaps it would be clearer if the icon for collectable objects was an inventory bag icon instead but it is certainly not a structural issue of the framework as the assets can be easily replaced by creators without changing the internal code.

The second piece of feedback - difficulties in distinguishing virtual objects from the background - introduces a more intriguing dilemma. While only two users referenced this issue explicitly, observations made during tests executed by other users suggested behaviour that point to the same issues (pointing the laser at everything in the game environments). It is arguable whether this is a problem at all, since some users might have thought it to be a positive aspect of the game, having to carefully explore the environment. Others on the other hand might find it frustrating. Delving into this would require further testing to discover the ideal solution.

The 3D assets produced for the proof of concept game were designed specifically to merge with the real world environment captured in the 360 videos by using realistic looking textures. This is also not a structural framework problem and is related with the artistic choice of having more realistic assets or less realistic ones that would definitely stand out and be immediately more noticeable by players.

On the one hand it can be said that the virtual objects should stand out and that player attention should be directed to thinking about how to solve the puzzles and not scouring the environment for interactive objects. However it is also arguable that looking for the interactive objects in the environment constitutes a form of exploration, a game mechanic that may have some entertainment value of its own. Another consideration to take into account is the impact that having less realistic looking objects would have on the immersion of the player. It is possible that the disparity between the real world recorded environments and the less realistic objects would lead to a jarring feeling of disconnect in the player and thus lessen the immersion. Having less realistic objects might also clash with the afforded possibility of the framework of making recorded video content interactive.

It is also of merit to mention that in general most users felt engaged by the gameplay, as it will be possible to see in the metrics related to engagement, with one user in particular writing on the additional comments section of the questionnaire how the interaction between virtual objects and the 360^{o} environment was seamless and a great experience.

If having objects that stand out more would lessen frustration and allow the player to focus more on the puzzles, would the potential decrease in immersion be worth the trade-off? It is certainly an interesting question that would merit further research. For now a reasonable solution that could potentially satisfy both types of users would be to include a toggle in the inventory menu interface for an option to highlight interactive objects.

4.5.4 Code Puzzle Interface

When it comes to the code puzzle interface the results could have been higher if it allowed the user to fix an unintended mistake in the current inputted solution without having to click the 'enter' button in orfder to clear everything up or close and reopen the whole interface again. As a reminder, the interface can be seen in Figure 3.4. Despite this, users still found the system to be generally intuitive.

This is an interesting component of the framework as it shows the potential to extend the gameplay possibilities allowed by Real Adventure with the creation of other plugin type components, essentially interfaces with mini-games that open upon interaction with an entity that contains that component. One user even remarked they would like to see rhythm based mini-games. Other possibilities include small chess puzzles such as trying to avoid a checkmate by grabbing a piece and placing it somewhere else. There are certainly a lot of fun mini-games that could be potentially integrated in the framework and easily used by creators.

4.5.5 Lost During the Experience

The main two reasons for players having felt lost or needing external help during the experience are faults in the game design of the proof of concept and not so much structural problems inherent to 360 video based adventure games or the framework. It was certainly a mistake to include knowledge that the user might not know as part of the puzzle solution and one of the objects that needed to be collectable was a bit small and as such it was missed by some of the users.

To fix the latter issue it would have been a wiser decision to create an entire roll of sandpaper as the virtual object instead of a small slip of sandpaper. An entire roll could be a textured cylinder and it would attract more attention and prevent the issue altogether. A general guideline regarding the design of 360 video based games can be learned from these experiments. Objects should be big enough to be noticeable by users, placed carefully in high contrast areas and the information needed to solve puzzles should be entirely within the game. If there is an intent in fostering exploration it may be desirable to hide objects more carefully, however it should be done in such a way that it is still fair to users in opposition to making them feel like they are "pixel hunting", a common issue in older point-and-click games.

A few users found the instantaneous transitions to different environments by using the navigation spheres to be a bit disorienting but it is certainly not a major issue as users will quickly find their footing by looking around and recognizing the location.

Finally, the matter of not knowing or forgetting that objects can be combined has already been discussed and analyzed in section 4.5.2.

4.5.6 Engagement and Replayability

The gameplay/puzzle sections of the proof of concept make up around 20% of the length of the experience so it is understandable that some users thought the ratio was too skewed towards narrative. Given that the proof of concept had to both contain a fully fledged storyline and be kept under the half an hour of playtime it was not possible to include more puzzles under those limitations but potential creators of 360 video based adventure games could certainly design their games with a better balance of gameplay and story.

User consensus points to an agreement that the gameplay was simple and clear, puzzles were fun, that the interface was unobtrusive and that the controls were responsive. Many users also wanted more puzzles and overall the gameplay seemed to engage the users. The only metric that fell short in comparison was "Always knew what to do" and it is related to the discussions regarding why some players felt lost or needed hints in section 4.5.5.

The feedback on the story was also overall positive with users in agreement that they were a part of the story. This metric could have scored even higher since the actor was not looking at users during many of the scenes and many commented on that fact. The actor had no time to practice the lines or memorize the dialogue and as such there was a need for a teleprompter with the lines at all times during filming. The positioning of the camera had been previously set to optimize the 360-degree view over the chosen locations.

Even so, it was an oversight during the video production not to align the camera between the actor and the teleprompter or, alternatively, to film different angles of the same location for the gameplay and for the story scenes. This problem was only noticed after everything had already been filmed and there was no opportunity or time to redo it.

Story is a very subjective component of an interactive experience and even with an amateur effort it was possible to delight and engage many users. 61% of users cited their curiosity to see what would happen with different choices or in different endings as a reason to definitely or maybe replay the game.

18% would replay if there were alternative puzzles in different story paths and that is certainly something that creators of this type of game should keep in mind. As for the scope of this thesis it was necessary to evaluate the same gameplay scenarios for all users and as such different puzzles for the two alternate story paths were not made.

It is possible to jump to different points of a game in Real Adventure by simply accessing different URLs. If game creators make the game in such a way that there are no dependencies between different game parts it is possible to skip to parts where choices are made to replay those specific segments. It is possible to even create buttons on the main menu of the game through the

JSON descriptions that would send the user immediately to some of these parts without changing the internal code of the framework.

Being able to skip already seen content is very important for users even to those who would still replay the game if that was not possible. Additionally, by designing different puzzles for different story paths and having an engaging branching story overall, it is possible to create a highly re-playable game even when that is not a trait that most adventure games are known for.

4.5.7 Final Remarks

With over 70% of users being willing to spend money on a 360º video based adventure game (Figure 4.4) it is plausible to say that there is some potential for this type of game as both an engaging gameplay experience and as a means of storytelling. The rest of the results also corroborate this possibility but the user sample is small so there would be a need for further research to make more conclusive remarks.

Regarding the suggested game mechanics, a movement system similar to Google street view where one can click arrows to move within the 360º environment in small bursts is a mechanic that could be easily integrated within Real Adventure. This would be possible by creating a component that changes the background image or video on clicking an object. The code for changing the background image or video without changing to an entirely different scene is already implemented as a flag-based transition. As such, it would be trivial to allow that functionality. This would however require much more effort in the filming/capture process of the 360 environments, as game designers would have to capture various shots and positions of the same location.

Walking as a game mechanic would be logistically difficult as there would be a need for special equipment such as a robot with a gimbal[2]. Alternatively, having a person moving the camera could be justified by certain types of narratives or scenarios. There would be no need for further code changes to the framework to do walking because it is all done in the filming process.

There are browser-based frameworks for voice recognition/commands and it could be an interesting way to interact with adventure games, by being able to speak to in-game characters for example. It would be worthwhile to explore this idea in future developments.

Finally, regarding the overall improvements suggested by users, they are all useful and positive suggestions that would require additional development time to implement.

[2] Gimbal is defined as a pivoted support that allows rotation of any object in a single axis

Chapter 5

Conclusions

Game development costs are rising rapidly as scope and expectations grow bigger with each passing year and while game engine technology is more accessible than ever, the skill barrier and labour required to make fully featured products is still quite high. It is with this reality in mind that the idea for interactive 360^o video based adventure games came to fruition. Real Adventure, a framework that allows the making of such content, was developed to offer an inexpensive way to create adventure games using immersive media by significantly streamlining the creation of game environments.

Through the state of the art review it was possible to determine which game mechanics were most prevalent in the adventure game genre while simultaneously being appropriate to the format of using 360^o visual content as the game world. These mechanics include gameplay-focused mechanics, such as contextual look-and-click interaction, point-and-click navigation, an object inventory system, use and examination of objects, combining objects to enable the creation of more diverse puzzles and hint systems. In addition to object-based puzzles it was decided that the framework would have to support number or text input puzzles. Narrative-focused mechanics include dialogue trees for more depth in NPC interaction and player choice that will allow the creation of branching narratives.

Moreover, from the literature review the primary factors that influence immersion and presence in virtual reality were understood, and important design guidelines were taken from previous research on interactive 360^o video. These guidelines call for special attention in focusing the user on important elements of the 360^o view, proper handling of transitions between different 360^o sources, natural integration of virtual 3D objects in the environments and use of camera techniques for more engaging navigation and interaction. Through the use of non-intrusive interfaces and exploitation of the factors that enhance the user's sense of presence in virtual reality it is possible to design for immersion.

By exploiting the cross-platform capabilities of A-Frame and with an entity-component-system architecture it was possible to design a low-code framework that allows the creation of fully featured 360^o video based adventure games through JSON descriptions. The modularity of the code allows for extensibility opportunities to Real Adventure, the developed framework, as game mechanics are implemented as isolated software components that can be plugged-in to different entities in the scene to add behaviour and functionality.

Through the creation of a fully featured proof of concept game, Animus Liber, it was possible to demonstrate that the framework allows for rapid development of content and, through the usability tests conducted on Animus Liber, it was possible to validate that the core mechanics studied and highlighted in section 2.1.3 were indeed adequate for this format of game. The results also point to a high level of satisfaction with the chosen control scheme of a laser pointer mixed with button interactions, even if there is room for improvement. The fact that the chosen game mechanics were indeed adequate and that users were satisfied with the general control scheme answers the first two research questions outlined in section 1.2.

Regarding the final research question in section 1.2 - "Do 360 video based adventure games have the potential to satisfy users both as a game and as a means of storytelling?" - users were engaged both in terms of gameplay and narrative even with an amateur production. Additionally, there is also the fact that a significant amount of users would be willing to buy commercial products of this style. Thus it is possible to say that 360^{o} video-based adventure games at the very least have the potential to provide a quality entertainment experience, both as a game and as a means of storytelling.

Moreover, the usability tests pointed to some improvements that can be made for the general usability of games developed with the framework. These include tweaks to the general inventory system and related mechanics such as combining objects among others, detailed in section 4.5.

Finally, the usability tests also point to game design principles that should be applied when developing a 360^{o} video based adventure game. Of most importance is creating different puzzles for different story paths, to create a highly replayable game not only for the narrative possibilities afforded by branching narratives but also because the puzzles are fun and engaging to users, providing an incentive to explore the alternative narrative paths. It is also important to balance out the ratio of story and gameplay, essentially fine-tuning the pacing of the game for an optimal experience and to create virtual objects with proper dimensions and placement to avoid player frustration during exploration.

The work that was presented throughout this thesis demonstrates that the research goals defined in the beginning of this document in section 1.3 were achieved in full. Likewise the research questions posed in section 1.2 were also fully answered earlier. With it comes the opportunity for further research and refinement of the concept of interactive 360^{o} video based adventure games, as well as of the framework that allows their creation. This future work will be discussed in the next section 5.1. The time is not yet ripe for mainstream adoption and creation of these games. 360^{o} content is still not commonplace and widely adopted by the masses. As virtual reality equipment evolves and the software rapidly changes, it will be possible to create more impressive looking content with higher resolutions and crisper image quality through the concurrent advancement of 360^{o} video capturing technology. The framework has potential to be a tool for adventure game/VR enthusiasts who wish to create their own games and for people in the academic world to explore other research avenues related to interactive narratives, game design, filmmaking and possibly psychology as discussed in section 2.1.4.

While other types of VR games can offer richer interactive experiences and more immersion, there is still a place for interesting independent games created in this format that do not require as many technical skills or as much time, staff and resources to create.

5.1 Future Work

Firstly, it would be necessary to apply the improvements that were discussed in section 4.5. With the intent of making the framework more developer-friendly it would be required to create a visual graphical interface for game creation. This interface could abstract the game creation even further by allowing the entire game to be created through drag-and-drop interactions. The structure of the JSON descriptions can certainly be translated into such an arrangement. This would essentially turn the framework from low-code to no-code.

In-depth documentation would aid potential developers immensely. It would be possible to even go as far as developing a sort of semantic parser to detect invalid combinations of components in the game descriptions, for an example.

It would also be interesting to integrate new functionalities and game mechanics such as new mini-games, the aforementioned Google street view movement system and voice recognition for deeper interaction with in-game characters.

Ultimately, further usability tests would need to be conducted to test the improvements and new game mechanics that could be implemented, and it would also be important to test the framework from the developer's perspective in order to optimize the creation process even further.

References

[AEBD16] L. Argyriou, D. Economou, V. Bouki, and I. Doumanis. Engaging immersive video consumers: Challenges regarding 360-degree gamified video applications. In *Proceedings of the 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*, 2016.

[AL04] Sandy Aylett and Ruth Louchart. Narrative theory and emergent interactive narrative. *International Journal of Continuing Engineering Education and Lifelong Learning*, 14(6):506–518, 2004.

[And15] Kosta Andreadis. The Past, Present and Future of Adventure Games. [Online]. Available at https://www.ign.com/articles/2015/03/02/the-past-present-and-future-of-adventure-games?page=1, 2015.

[ANVA99] Alan Amory, Kevin Naicker, Jacky Vincent, and Claudia Adams. The use of computer games as an educational tool: Identification of appropriate game types and game elements. *British Journal of Educational Technology*, 1999.

[Ayl99] Ruth Aylett. Narrative in Virtual Environments - Towards Emergent Narrative. *Proceedings of the AAAI fall symposium on narrative*, pages 83–86, 1999.

[Bro15] Mark Brown. Point and Click Puzzle Design | Game Maker's Toolkit. [Online]. Available at https://www.youtube.com/watch?v=EDt6XXsRXag, 2015.

[BSEW16] A. Brown, A. Sheikh, M. Evans, and Z. Watson. Directing attention in 360-degree video. In *IBC 2016 Conference*, pages 29 (9 .)–29 (9 .). Institution of Engineering and Technology, 2016.

[Cha78] S. Chatman. *Story and Discourse: Narrative structure in fiction and film*. Cornell University Press, 1978.

[Dic05] Michee D. Dickey. Engaging by design: How engagement strategies in popular computer and video games can inform instructional design. *Educational Technology Research and Development*, 53(2):67–83, 2005.

[Dic11] Michele D. Dickey. Murder on Grimm Isle: The impact of game narrative design in an educational game-based learning environment. *British Journal of Educational Technology*, 2011.

[DN12] Isaac Dart and Mark J. Nelson. Smart terrain causality chains for adventure-game puzzle generation. In *2012 IEEE Conference on Computational Intelligence and Games, CIG 2012*, pages 328–334, 2012.

[Dou18] Nick Douglas. Tour Art Museums and Street Art on Google Street View. [Online]. Available at https://lifehacker.com/1829448470, 2018.

[EEW18] Ahmed Elmenzeny, Ninda Edenhofer, and Jeffrey Wimmer. Immersive storytelling in 360-degree videos: An analysis of interplay between narrative and technical immersion. *Journal of Virtual Worlds Research*, 11(1):1–13, 2018.

[Fer09] Clara Fernández Vara. *Integrating Story Into Simulation Through Performance*. PhD thesis, Georgia Institute of Technology, Georgia, 2009.

[FO15] Laura Freina and Michela Ott. A literature review on immersive virtual reality in education: state of the art and perspectives. In *Conference proceedings of »eLearning and Software for Education« (eLSE)*, volume 1, page 133, 2015.

[Fre14] Alexander Freed. Branching Conversation Systems and the Working Writer, Part 1: Introduction. [Online]. Available at http://www.gamasutra.com/blogs/AlexanderFreed/20140902/224609/Branching_Conversation_Systems_and_the_Working_Writer_Part_1_Introduction.php, 2014.

[FV08] Clara Fernández-Vara. Shaping Player Experience in Adventure Games: History of the Adventure Game Interface. In Olli Leino, Hanna Wirman, and Amyris Fernandez, editors, *Extending Experiences: Structure, Analysis and Design of Computer Game Player Experience*, pages 210–227. Lapland University Press, illustrate edition, 2008.

[FV11] Clara Fernández-Vara. From "open mailbox" to context mechanics. *Proceedings of the 6th International Conference on Foundations of Digital Games - FDG '11*, pages 131–138, 2011.

[FVT12] Clara Fernández-Vara and Alec Thomson. Procedural Generation of Narrative Puzzles in Adventure Games. In *Proceedings of the The third workshop on Procedural Content Generation in Games - PCG'12*, pages 1–6, 2012.

[Gal06] Alexander R. Galloway. *Gaming: Essays on Algorithmic Culture*. Choice Current Reviews for Academic Libraries, 2006.

[Har08] Tilo Hartmann. Parasocial interactions and paracommunication with new media characters. In *Mediated Interpersonal Communication*. 2008.

[HCCJ17] Jingwei Huang, Zhili Chen, Duygu Ceylan, and Hailin Jin. 6-DOF VR videos with a single 360-camera. In *Proceedings - IEEE Virtual Reality*, pages 37–44, 2017.

[HD92] Richard Held and Nathaniel Durlach. Telepresence. *Presence: Teleoperators and Virtual Environments*, 1(1):109–112, 1992.

[HW56] Donald Horton and R Richard Wohl. Mass communication and para-social interaction; observations on intimacy at a distance. *Psychiatry: Interpersonal and Biological Processes*, 1956.

[ICK$^+$15] Ioanna Iacovides, Anna Cox, Richard Kennedy, Paul Cairns, and Charlene Jennett. Removing the HUD: The Impact of Non-Diegetic Game Elements and Expertise on Player Involvement. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play - CHI PLAY '15*, 2015.

[JP09] Seung-A Annie Jin and Namkee Park. Parasocial Interaction with My Avatar: Effects of Interdependent Self-Construal and the Mediating Role of Self-Presence in an Avatar-Based Console Game, Wii. *CyberPsychology & Behavior*, 12(6):723–727, 2009.

[JW97] Edward Ju and Christian Wagner. Personal Computer Adventure Games: Their Structure, Principle and Applicability for Training. *The Data Base for Advances in Information Systems*, 28(2):78–92, 1997.

[Kav12] Katrine Kavli. The player's parasocial interaction with digital entities. *Proceeding of the 16th International Academic MindTrek Conference on - MindTrek '12*, page 83, 2012.

[Kim06] Scott Kim. What is a Puzzle ? Is it fun ? Does it have a right answer ? [Online]. Available at http://www.scottkim.com.previewc40.carrierzone.com/thinkinggames/whatisapuzzle/index.html, 2006.

[Kwi12] Karol Kwiatek. How to preserve inspirational environments that once surrounded a poet? Immersive 360-degree video and the cultural memory of Charles Causley's poetry. In *Proceedings of the 2012 18th International Conference on Virtual Systems and Multimedia, VSMM 2012: Virtual Systems in the Information Society*, 2012.

[LA05] Sandy Louchart and Ruth Aylett. Managing a non-linear scenario - A narrative evolution. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3805 LNCS, pages 148–157, 2005.

[MK94] Paul Milgram and Fumio Kishino. Taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, 1994.

[MNRP18] Teresa Matos, Rui Nóbrega, Rui Rodrigues, and Marisa Pinheiro. Dynamic annotations on an interactive web-based 360^o video player. In *Proceedings of the 23rd International ACM Conference on 3D Web Technology*, page 22. ACM, 2018.

[MS17] Andrew MaCquarrie and Anthony Steed. Cinematic virtual reality: Evaluating the effect of display type on the viewing experience for panoramic video. In *Proceedings - IEEE Virtual Reality*, 2017.

[NMH$^+$16] Lasse T. Nielsen, Matias B. Møller, Sune D. Hartmeyer, Troels C. M. Ljung, Niels C. Nilsson, Rolf Nordahl, and Stefania Serafin. Missing the point. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology - VRST '16*, pages 229–232, 2016.

[NN16] Max North and Sarah North. A comparative study of sense of presence of traditional virtual reality and immersive environments. *Australasian Journal of Information Systems*, 20(0), 2016.

[RKM$^+$16] Michael A. Rupp, James Kozachuk, Jessica R. Michaelis, Katy L. Odette, Janan A. Smither, and Daniel S. McConnell. The effects of immersiveness and future VR expectations on subjective-experiences during an educational 360-degree video. *Proceedings of the Human Factors and Ergonomics Society*, 2016.

[RTGG17] Jose Rubio-Tamayo, Manuel Gertrudix Barrio, and Francisco García García. Immersive Environments and Virtual Reality: Systematic Review and Advances in Communication, Interaction and Simulation. *Multimodal Technologies and Interaction*, 1(4):21, 2017.

[SF16] Wen Song and Jesse Fox. Playing for Love in a Romantic Video Game: Avatar Identification, Parasocial Relationships, and Chinese Women's Romantic Beliefs. *Mass Communication and Society*, 19(2):197–215, 2016.

[Shn83] Shneiderman. Direct Manipulation: A Step Beyond Programming Languages. *Computer*, 16(8):57–69, 8 1983.

[SS02] Wallace Sadowski and Kay Stanney. Presence in virtual environments. In *Handbook of virtual environments: Design, implementation, and applications.*, Human factors and ergonomics., pages 791–806. Lawrence Erlbaum Associates Publishers, Mahwah, NJ, US, 2002.

[Ste92] Jonathan Steuer. Defining Virtual Reality: Dimensions Determining Telepresence. *Journal of Communication*, 1992.

[SU93] Mel Slater and Martin Usoh. Representations Systems, Perceptual Position, and Presence in Immersive Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 1993.

[SW97] Mel Slater and Sylvia Wilbur. A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and Virtual Environments*, 1997.

[VCSF16] Luis Valente, Esteban Clua, Alexandre Ribeiro Silva, and Bruno Feijó. Live-action Virtual Reality Games. *arXiv preprint arXiv:1601.01645*, 2016.

[VKP10] Rick Van Krevelen and Ronald Poelman. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality (ISSN 1081-1451)*, 9:1, 06 2010.

[WAEG06] Laurie M. Wilcox, Robert S. Allison, Samuel Elfassy, and Cynthia Grelik. Personal space in virtual reality. *ACM Transactions on Applied Perception*, 3(4):412–428, oct 2006.

[WS98] Bob G. Witmer and Michael J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and Virtual Environments*, 1998.

[Yee18] Mary (SIE) Yee. Celebrating 3 Million PS VR Systems Sold. [Online]. Available at https://blog.us.playstation.com/2018/08/16/celebrating-3-million-ps-vr-systems-sold/, 2018.

[Zyd05] Michael Zyda. From visual simulation to virtual reality to games. *Computer*, 38(9):25–32, 2005.

Appendix A

Cited Games

"Colossal Cave Adventure", Crowder and Woods, 1976
"Zork", Infocom, 1977
"Mystery House", Sierra Online, 1980
"The Hobbit", Melbourne House, 1982
"King's Quest", Sierra Online, 1984
"The Secret of Monkey Island", LucasArts, 1991
"Myst", Cyan Worlds, 1993
"Full Throttle", LucasArts, 1995
"Grim Fandango", LucasArts, 1998
"Phoenix Wright: Ace Attorney", Capcom, 2001
"Hotel Dusk: Room 215", Cing, 2007
"The Secret of Monkey Island Special Edition", LucasArts, 2009
"9 Hours 9 Persons 9 Doors", Spike Chunsoft, 2009
"Machinarium", Amanita Design, 2009
"The Walking Dead", Telltale Games, 2012
"The Wolf Among Us", Telltale Games, 2013
"I Expect You to Die", Schell Games, 2016
"Obduction", Cyan Worlds, 2016
"Danganronpa V3", Spike Chunsoft, 2017
"Detroit: Become Human", Quantic Dream, 2018
"Return of the Obra Dinn", Lucas Pope, 2018

Cited Games

Appendix B

Animus Liber Storyline

Download

Download the browser interactive fiction made with twine[1] at:

https://drive.google.com/open?id=17G0Q26pVenalu4GkVnQj0qJ9bMZhYZbj

Full Story Text

Chapter 1 - Simulator Innocence

Location 1 - Some empty classroom where basic introductory puzzles can be created, the main hub where teleport spheres to other locations will be displayed:

- Some robotic voice(voiceover): Welcome to the Armis Corporation VR test. Thank you for volunteering. Our virtual friend John Clare will be your assistant throughout the demonstration.

- John Clare(monotone voice): My name is John Clare. Your aim is to successfully complete a series of challenges. I will be showing you very quickly how to interact and you'll be on your way. Firstly let's go over the basics. Point your laser to different objects in the environment and use the trigger button to examine or pick up said objects. There are two objects that you can interact with in this room, give it a try.

(**Playable Scene:** You pick up a tube and examine a projector in the room. Upon examination of the projector a dialogue reads "This projector is broken. It's missing a part.")

I'm done playing

[1] https://twinery.org/

Tutorial Part 2

- John Clare(monotone voice): Great work... Now use the menu button to bring up your inventory. You can grab objects from your inventory by holding the trigger button and combine them with other objects in the inventory. Alternatively you can also use inventory objects in the environment by releasing the trigger button in the correct location. Try assembling a projector lens to fix the broken projector.

(Playable Segment : You successfully complete the task)

- John Clare("tired" voice): Well now you know the basics, the last thing I'll tell you is how you can navigate to different environments. See this sphere right here? try clicking on it with the trigger button. Generally speaking the trigger button will do everything you need.

Proceed

Simulator Guilt

- John Clare(exasperated, rushing): Finally we're out of that hellhole.This is going to sound absurd but you have to believe me. I am actually a real person, not just a dumbass computer program. I've been stuck in this mind numbing static virtual reality for years. I was kidnapped by these world government bastards and used as a guinea pig for their research, I am probably already dead in real life

(Playable Segment: You can point and click to John Clare and a dialogue tree will pop up with three topics of conversation. You explore all conversation options.)

Topic A) That sounds unbelievable. Wouldn't anyone have looked for you or notice you're gone?

- John Clare(sad and defeated): What they've done to me is monstrous... My only connection to the outside world is this computer. I can look up information about the real world but there's no way to reach out to anyone. I tried looking up stories about my disappearance but got nothing. (Sounding more desperate) It's impossible that my family hasn't gone to the cops... I fear for their lives.

Topic B) World government? Are you on drugs?

- John Clare(sorta scared, conspiratory tone): Governments, countries, borders, it's all a sham, our world is controlled by an invisible elite. The ones who did this to me are a top secret research group backed by that very same world government, led by the genius Dr Locke.

- John Clare(angrier): They wanted to successfully convert the human mind into data. Congratulations to them fuckers, they've succeeded. I don't know what they want to do with you... you're probably not safe either, watch your back. I don't have time to explain everything, they'll get suspicious.

Topic C) What can I even do for you?

- John Clare(desperate): Please just get me out of here somehow, put me out of my misery for all I care. I just want to die... This is no way to live.

You: "I'll try my best."
You:"..."

Clare reassured

- John Clare: Thank you... You're a kind soul.

- John Clare(exasperated): We have to go back, we're taking too long, please try to help me, release me or erase me somehow. Let's get out of here -> automatic transition back to Location 1.

Proceed

Clare frustrated

- John Clare: Shit... You don't actually believe me do you... Can't say I blame you.

- John Clare(exasperated): We have to go back, we're taking too long, please try to help me, release me or erase me somehow. Let's get out of here -> automatic transition back to Location 1.

Proceed

Chapter 2 - Doubt

A big hologram screen appears in the simulation playing a live feed of a masked man

- Masked man in 2D screen(distorted voice): Clare oh Clare... You just had to mess with our little experiment. Did you really think we wouldn't have eyes and ears on the entire simulation, I thought you were brighter than that.

- Masked man: Look friend (referring to the player), I cannot disclose particulars about this demonstration or my identity because it's classified, but I can assure you it's all perfectly legal and safe. John Clare here is actually a death row convict, yes in truth he was an organic being turned digital but all for the benefit of humanity.

- Masked man: This man was destined to die, we offered him a second chance at life if he played along and followed correct procedures during our experiments. I suppose it was our mistake to trust a man of his nature. Don't believe me? (bring a newspaper into frame and scan it(maybe with a phone with torch light on) - A 3d newspaper object falls into the scene following said scan). Have a look for yourself.

(**Playable Scene:** You inspect the newspaper. Headline reads: Cult leader John Clare sentenced to death following tragic mass suicide incident.

Body Reads: Following an unprecedented mass suicide tragedy, cult leader John Clare was finally apprehended after a violent siege against police forces. This is a case that has shocked the world, the convict is awaiting trial but the public is already calling for the death penalty. We will keep you updated as the case develops.)

- Masked man: As you can see, John Clare is a despicable man who ruined the lives of hundreds. We're so sorry you've had to deal with this. It wasn't supposed to be like this. Finish the simulation and we will greatly reward you for your troubles. And Mr Clare, if you don't want your situation to worsen any further(threatening voice), help our little friend here... quietly. Just do your job.

Big hologram screen disappears

Proceed

Despair, John Clare

- John Clare (Sounding defeated and pleading): Please come this way, this navigation sphere will take you to your next challenge. But before we go, since the jig is up, let me tell you... I'm an innocent man, that newspaper isn't proof at all, he can just will anything he wants into existence in this world.

- John Clare(angrier): The bastard, he must be Dr Locke. Why would he conceal his identity if he were a man of good intent. There is no point in trying to fake obedience anymore... I know it all sounds insane but please believe me...

You:"I believe you"

You:"Bullshit"

Chapter 3A - Believe Clare

- John Clare reaction (relief): Thank you from the bottom of my heart. I might be beyond saving but your kindness has given me some much needed peace. Let's keep going so you can get out of here.

(**Playable Scene:** You use navigation sphere to move to Location 3)

- John Clare: We've arrived at the final challenge but before we move on to that I'd like to just get a few things off my chest.

- John Clare: You know just a bit ago when I took you to that room where I first asked for your help? I was actually a part of Locke's research team and I had snuck that secret room in the simulation myself but it seems Locke found out about it. I was naive to think they would never find it. All this time I tried to be secretive and pretended to cooperate. I wanted to keep my family safe... in case they are still even alive but I fucked it all up.

- John Clare (melancholic): You must be wondering why I ended up here. I was a top ranking researcher, only second to Dr Locke. I thought I was making a difference, you know... really important work. Doing cutting edge research for all of mankind.

- John Clare: We were tasked by the elite to develop a solution for transferring the human consciousness into the virtual world as a means of escaping the next grand extinction event on earth. The public is completely unaware but twenty years from now a giant meteor will fall on earth. The impact, subsequent natural catastrophes and atmosphere changes will annihilate us all...

- John Clare: Some remote islands off the pacific were calculated to be left mostly untouched. I was a fool and thought they were going to try to save everyone but eventually I discovered that they had handpicked a couple thousand to be humanity's last stand while leaving everyone else to die.

- John Clare: Of the privileged chosen those who would prefer to do so would continue their biological existence, living peacefully in the small pacific islands with an artificial atmosphere isolated from the rest of the planet.

- John Clare: I was horrified and disgusted by these self aggrandizing parasites trying to play the hand of god. Even though my family was guaranteed survival at the end of the world I still couldn't let it stand. I was preparing an exposé but as you can surmise... my cover was blown. Those bastards killed two birds with one stone, they forcefully used me as the very first test subject for human virtualization.

Sudden interruption by hologram screen appearing in the simulation once again

- Masked man: Wow, no wonder you made hundreds of people commit suicide. It's hard to listen to all that drivel without wanting to off yourself. My friend you've seen the irrefutable evidence I've given you before. Since Mr.Clare is being a nuisance and tainting the experience instead of doing his job we will have to get him out of your hair. Don't worry there is only one challenge left and then you'll be on your merry way with a lot of money in your pocket. (John clare is paralyzed).

- Masked man is seen in the video talking into the background: Send the Special Ops team to deal with guest of ours. Oh and locate his family too just in case.

- Masked man(Look at the front of the screen again): Oops, forgot to turn this off. Anyway good luck with the final challenge friend (Sarcastic).

Proceed

Chapter 3B - Don't believe Clare

- John Clare reaction (arrogance): Eh.. it was worth a shot. Maybe you're not as much of an idiot as I thought. I can respect that at least. Now let's just get this over with, I'm already sick of you.

(**Playable Scene:** You use navigation sphere to move to Location 3)

- John Clare: You seem smart enough to understand, you saw right through my ruse. You're not like those fools at my old cult. They were weaklings, mentally frail sheep. They were mine, body and soul. I gave them hope and excitement, it was so easy. They all died believing they were going to paradise. Some bullshit I made up because I was sick of dealing with their neediness.

- John Clare: For a while it was nice to feel like a god. These people did everything for me. You can't even imagine the things I've made them do in my name. Unfortunately for them the excitement wore off and only boredom remained. I had an escape plan after making them take the paradise pills but things just didn't work out. So here am I now in this freak show. I don't even know what the point of all this is. Why did they even bring you here?

Sudden interruption by hologram screen appearing in the simulation once again

- Masked man: Enough drivel. My esteemed guest, I will get this nuisance out of your way so that you may complete the final challenge. We appreciate your cooperation. (John clare is paralyzed)

Proceed

Final Puzzle

(**Playable Scene:** You solve a series of puzzles (involving combining objects, using objects and examining to gather clues to ultimately input a code and open a safe) - Inside the safe is the picture of the actor that plays both John Clare and Dr Locke)

I'm done playing

Convergence

Hologram screen appears again

- Masked man in 2D screen (distorted voice): Congratulations on finishing the final challenge. Let's just unfreeze our little John Clare here.

- Masked man in 2D screen (non distorted voice as he takes off mask): Now you two listen closely. Everything you were told is a lie. My real name is Dr Locke but in a way I am also John Clare. There is no cult leader or world government elite. It's all hogwash. This was a demonstration of the ULTIMATE REAL FICTION.

- Dr Locke: Clare is indeed a virtualized version of myself. I altered Clare so that he would react to your choices. He was given different sets of fake memories that would trigger depending on stimuli from you, my esteemed guest. All of this to create a true to life character, one that feels and thinks like a real human, one that can adapt to the circumstances of his given backstory. All to create the most visceral and intense fiction.

(At this point Clare goes catatonic)

- Dr Locke: Look at that thing going crazy. Don't feel sorry for it. I can just reset his memories and he'll be ready for the next person to experience the ultimate real fiction. He'll forget all of his pain and existential dread. Since you've made it this far I'll give you the privilege to choose. If you have some ethical objection I'll even allow you to choose to erase him completely.

You:"I have bad news for you Locke. You are fiction too"
You:"Let's reset John Clare, that was a great story!"
You:"John deserves dignity, erase him from his torment"
You:"Locke, your fiction sucked... It's full of plot holes"

Ending 1 - Who cares about Free Will END

- Dr Locke: What are you talking about, that's nonsense...

(**Playable Scene:** Prompt Player to Present Item -Choose picture of real life actor)

- Locke(shocked): Whaat? I don't remember taking that picture.

- Locke(clearly shocked but still arrogant): Ironic isn't it? I guess I'm not so different from that thing... NOT. Your attempt at downplaying my achievements is pathetic. Who cares about free will? From my perspective I have achieved brilliance all the same.

- Locke(sound confident again): Even if my life is ephemeral and predestined, I don't know what the future holds so goodbye my friend, nice try. Have you ever considered that you too might be a puppet to even higher beings than yourself? In a way I'm also Ultimate real fiction. Locke OUT

Ending 2 - The fiction will go on END

- Locke: I see that you too are a person of culture. Congratulations in finishing the ultimate real fiction. John Clare will be ready for the next lucky user.

Ending 3 - Do you really think I would stop END

Locke: As promised I will delete John Clare forever. Bid your friend farewell. I'll just let you on a little secret before we end this. While you've been testing out the simulation I have managed to virtualize your consciousness to be the next character in the ultimate real fiction. Splendid, thank you for your invaluable contribution (sarcastic). Locke OUT.

Ending 4 - Ultimate real failure END

Locke: Damn… So my ultimate real fiction was more like an ultimate real failure. Fear not. I will give Clare better memories next time. You haven't seen the last of me. I'll be back.

Appendix C

Usability Questionnaires

DECLARAÇÃO DE CONSENTIMENTO

(Baseada na declaração de Helsínquia)

No âmbito da realização da tese de Mestrado do Mestrado Integrado de Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto, intitulada **"A framework for 360º video adventure games"**, realizada pelo estudante Francisco José Rodrigues de Pinho, orientada pelo Prof. Rui Rodrigues e sob a co-orientação do Prof. Rui Nóbrega, eu abaixo assinado declaro que compreendi a explicação que me foi fornecida acerca do estudo no qual irei participar, nomeadamente o carácter voluntário dessa participação, tendo-me sido dada a oportunidade de fazer as perguntas que julguei necessárias.

Tomei conhecimento de que a informação ou explicação que me foi prestada versou os objetivos, os métodos, o eventual desconforto e a ausência de riscos para a minha saúde, e que será assegurada a máxima confidencialidade dos dados.

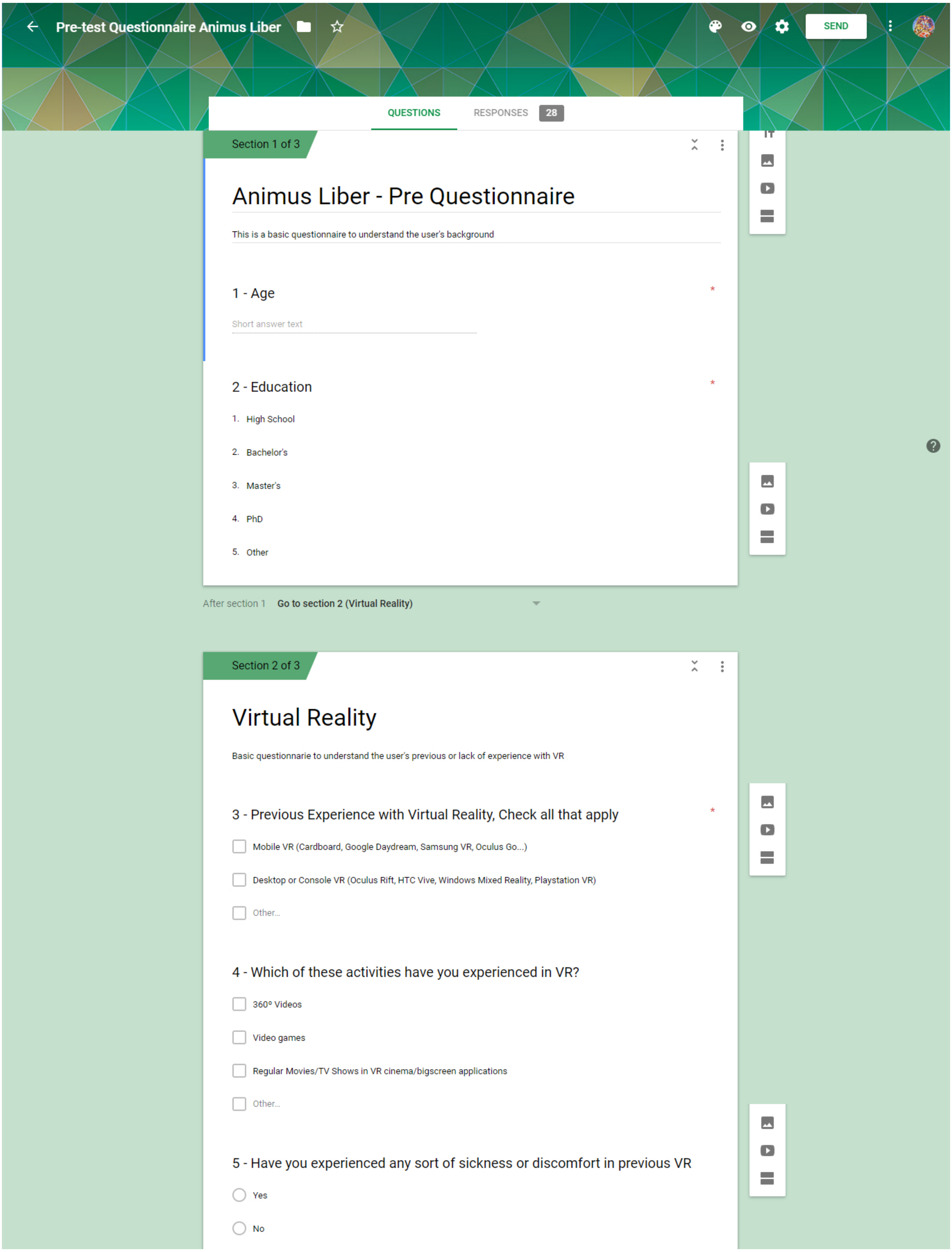
Explicaram-me, ainda, que poderei abandonar o estudo em qualquer momento, sem que daí advenham quaisquer desvantagens.

Por isso, consinto participar no estudo e na recolha de imagens necessárias, respondendo a todas as questões propostas.

Porto, __ de ___________ de 201_

____

(Participante ou seu representante)



Animus Liber - Pre Questionnaire

This is a basic questionnaire to understand the user's background

1 - Age *

Short answer text

2 - Education *

1. High School
2. Bachelor's
3. Master's
4. PhD
5. Other

After section 1 Go to section 2 (Virtual Reality)

Virtual Reality

Basic questionnarie to understand the user's previous or lack of experience with VR

3 - Previous Experience with Virtual Reality, Check all that apply *

□ Mobile VR (Cardboard, Google Daydream, Samsung VR, Oculus Go...)

□ Desktop or Console VR (Oculus Rift, HTC Vive, Windows Mixed Reality, Playstation VR)

□ Other...

4 - Which of these activities have you experienced in VR?

□ 360° Videos

□ Video games

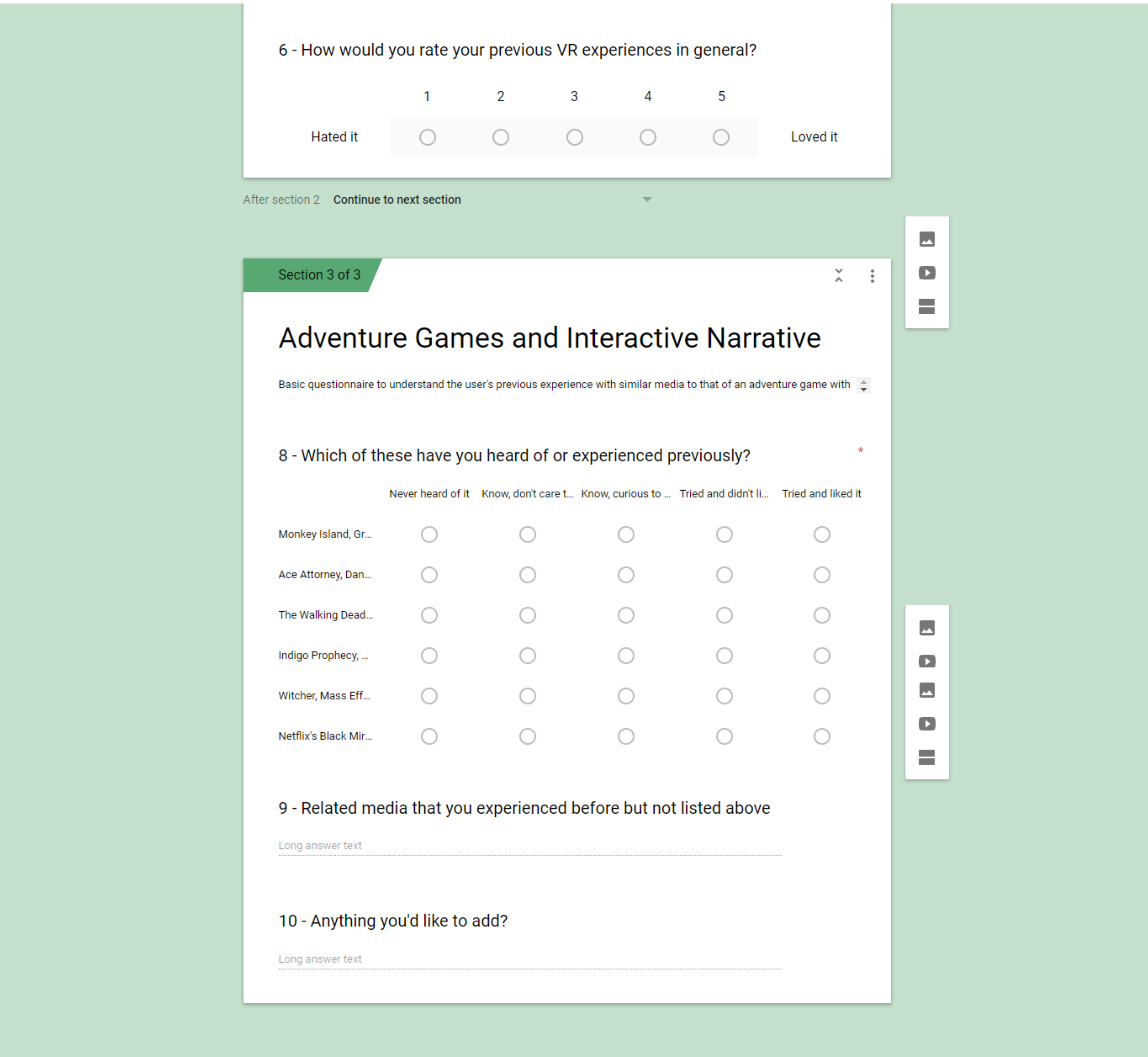
□ Regular Movies/TV Shows in VR cinema/bigscreen applications

□ Other...

5 - Have you experienced any sort of sickness or discomfort in previous VR

○ Yes

○ No



6 - How would you rate your previous VR experiences in general?

	1	2	3	4	5	
Hated it	○	○	○	○	○	Loved it

After section 2 Continue to next section

Section 3 of 3

Adventure Games and Interactive Narrative

Basic questionnaire to understand the user's previous experience with similar media to that of an adventure game with

8 - Which of these have you heard of or experienced previously? *

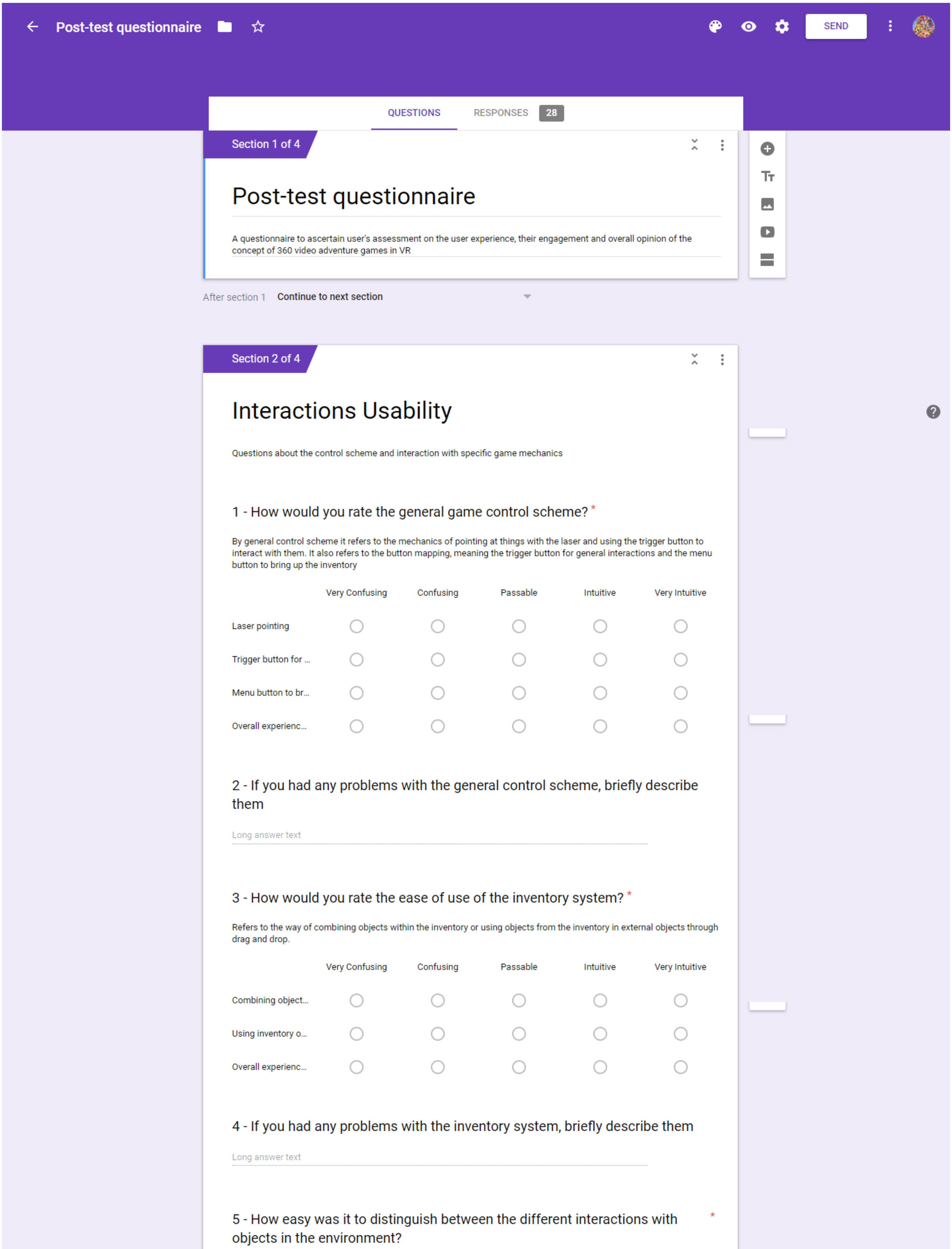
	Never heard of it	Know, don't care t...	Know, curious to ...	Tried and didn't li...	Tried and liked it
Monkey Island, Gr...	○	○	○	○	○
Ace Attorney, Dan...	○	○	○	○	○
The Walking Dead...	○	○	○	○	○
Indigo Prophecy, ...	○	○	○	○	○
Witcher, Mass Eff...	○	○	○	○	○
Netflix's Black Mir...	○	○	○	○	○

9 - Related media that you experienced before but not listed above

Long answer text

10 - Anything you'd like to add?

Long answer text



Section 1 of 4

Post-test questionnaire

A questionnaire to ascertain user's assessment on the user experience, their engagement and overall opinion of the concept of 360 video adventure games in VR

After section 1 Continue to next section

Section 2 of 4

Interactions Usability

Questions about the control scheme and interaction with specific game mechanics

1 - How would you rate the general game control scheme? *

By general control scheme it refers to the mechanics of pointing at things with the laser and using the trigger button to interact with them. It also refers to the button mapping, meaning the trigger button for general interactions and the menu button to bring up the inventory

	Very Confusing	Confusing	Passable	Intuitive	Very Intuitive
Laser pointing	○	○	○	○	○
Trigger button for ...	○	○	○	○	○
Menu button to br...	○	○	○	○	○
Overall experienc...	○	○	○	○	○

2 - If you had any problems with the general control scheme, briefly describe them

Long answer text

3 - How would you rate the ease of use of the inventory system? *

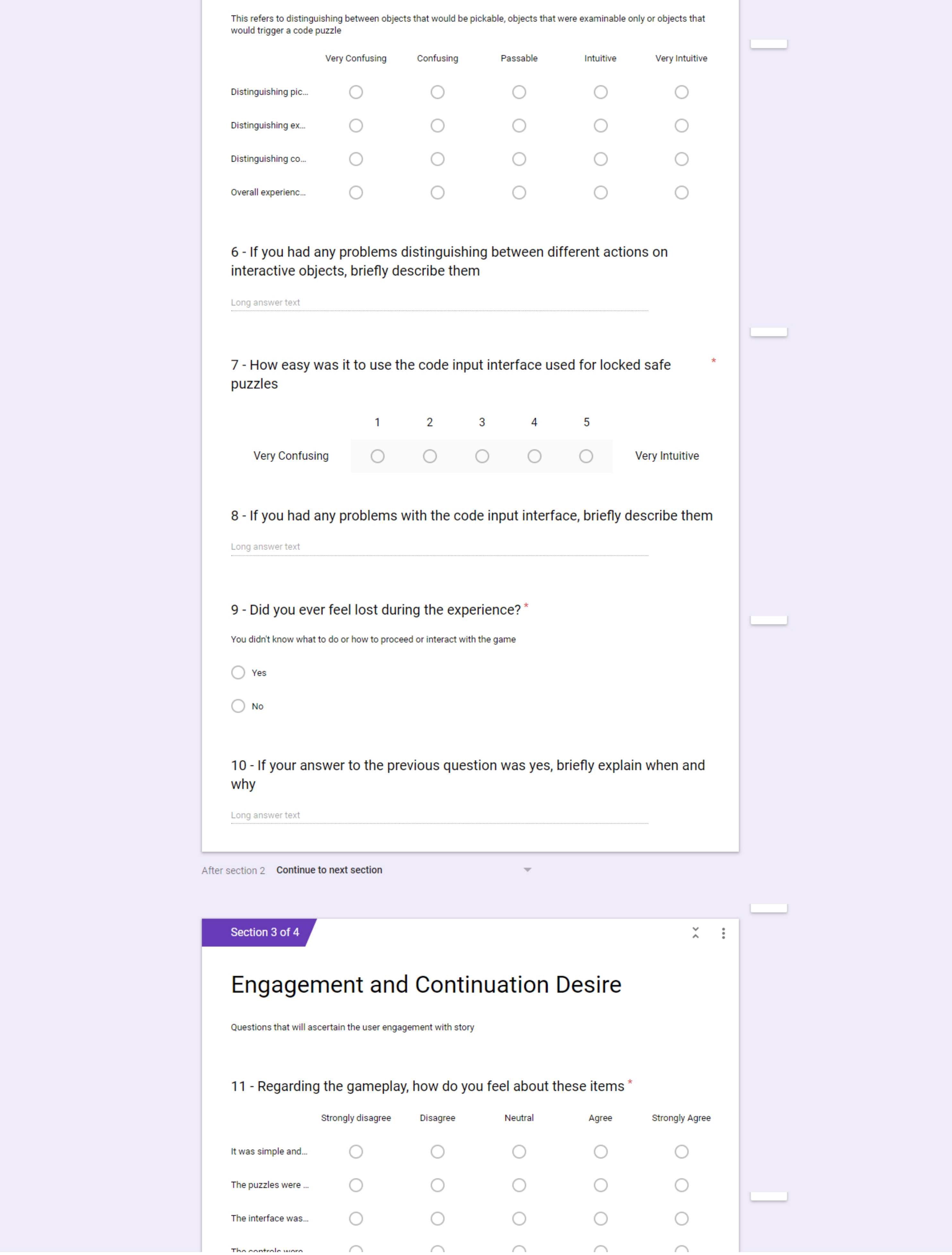
Refers to the way of combining objects within the inventory or using objects from the inventory in external objects through drag and drop.

	Very Confusing	Confusing	Passable	Intuitive	Very Intuitive
Combining object...	○	○	○	○	○
Using inventory o...	○	○	○	○	○
Overall experienc...	○	○	○	○	○

4 - If you had any problems with the inventory system, briefly describe them

Long answer text

5 - How easy was it to distinguish between the different interactions with objects in the environment? *



This refers to distinguishing between objects that would be pickable, objects that were examinable only or objects that would trigger a code puzzle

	Very Confusing	Confusing	Passable	Intuitive	Very Intuitive
Distinguishing pic...	○	○	○	○	○
Distinguishing ex...	○	○	○	○	○
Distinguishing co...	○	○	○	○	○
Overall experienc...	○	○	○	○	○

6 - If you had any problems distinguishing between different actions on interactive objects, briefly describe them

Long answer text

7 - How easy was it to use the code input interface used for locked safe puzzles *

	1	2	3	4	5	
Very Confusing	○	○	○	○	○	Very Intuitive

8 - If you had any problems with the code input interface, briefly describe them

Long answer text

9 - Did you ever feel lost during the experience? *

You didn't know what to do or how to proceed or interact with the game

○ Yes

○ No

10 - If your answer to the previous question was yes, briefly explain when and why

Long answer text

After section 2 **Continue to next section**

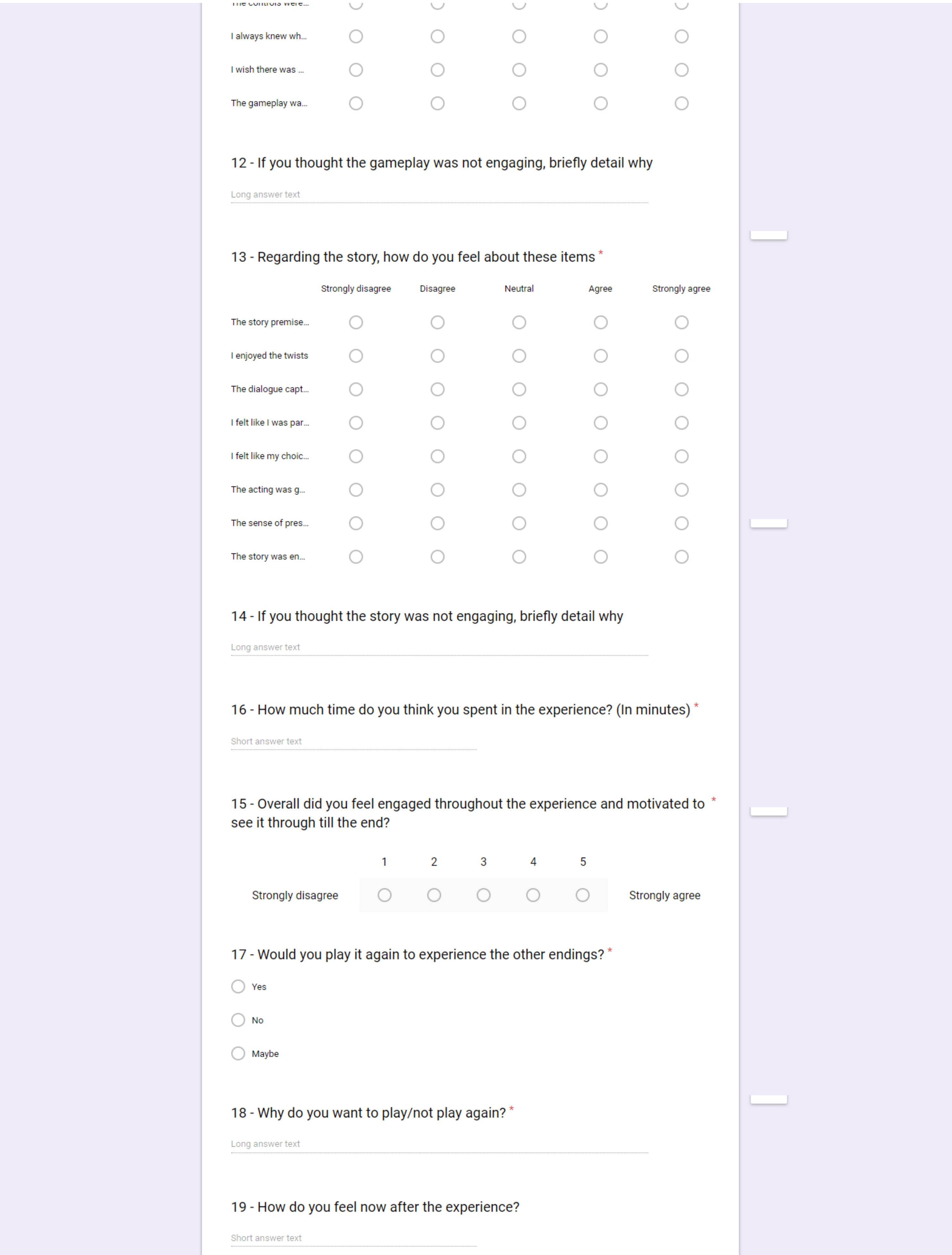
Section 3 of 4

Engagement and Continuation Desire

Questions that will ascertain the user engagement with story

11 - Regarding the gameplay, how do you feel about these items *

	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree
It was simple and...	○	○	○	○	○
The puzzles were ...	○	○	○	○	○
The interface was...	○	○	○	○	○
The controls were	○	○	○	○	○



The controls were...	○	○	○	○	○
I always knew wh...	○	○	○	○	○
I wish there was ...	○	○	○	○	○
The gameplay wa...	○	○	○	○	○

12 - If you thought the gameplay was not engaging, briefly detail why

Long answer text

13 - Regarding the story, how do you feel about these items *

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The story premise...	○	○	○	○	○
I enjoyed the twists	○	○	○	○	○
The dialogue capt...	○	○	○	○	○
I felt like I was par...	○	○	○	○	○
I felt like my choic...	○	○	○	○	○
The acting was g...	○	○	○	○	○
The sense of pres...	○	○	○	○	○
The story was en...	○	○	○	○	○

14 - If you thought the story was not engaging, briefly detail why

Long answer text

16 - How much time do you think you spent in the experience? (In minutes) *

Short answer text

15 - Overall did you feel engaged throughout the experience and motivated to see it through till the end? *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

17 - Would you play it again to experience the other endings? *

○ Yes

○ No

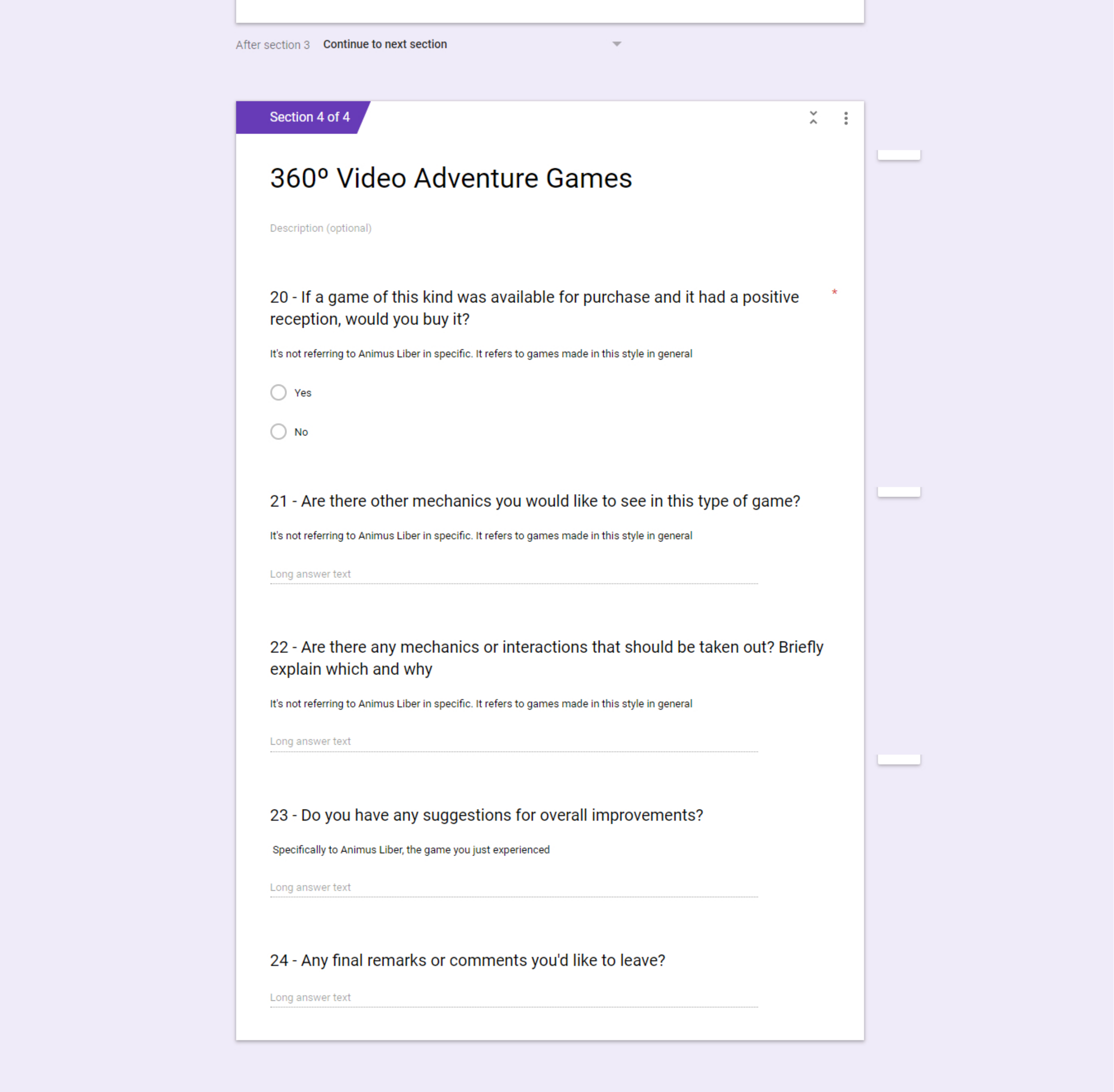
○ Maybe

18 - Why do you want to play/not play again? *

Long answer text

19 - How do you feel now after the experience?

Short answer text



After section 3 Continue to next section

Section 4 of 4

360º Video Adventure Games

Description (optional)

20 - If a game of this kind was available for purchase and it had a positive reception, would you buy it? *

It's not referring to Animus Liber in specific. It refers to games made in this style in general

○ Yes

○ No

21 - Are there other mechanics you would like to see in this type of game?

It's not referring to Animus Liber in specific. It refers to games made in this style in general

Long answer text

22 - Are there any mechanics or interactions that should be taken out? Briefly explain which and why

It's not referring to Animus Liber in specific. It refers to games made in this style in general

Long answer text

23 - Do you have any suggestions for overall improvements?

Specifically to Animus Liber, the game you just experienced

Long answer text

24 - Any final remarks or comments you'd like to leave?

Long answer text