**EEE521 Final Year Project Report**

**School of Computing, Engineering, & Intelligent Systems**

**Zita Koczka B00786645**

**BSc Hons Computer Science**

**A computer game of my own design**

**Supervisor: Dr Darryl Charles**

**Second Marker: Dr Saugat Bhattacharyya**

**18th November 2022**

Logo

Description automatically generated

Plagiarism Statement

I declare that this is all my own work and does not contain unreferenced material copied from any other source. I have read the University's policy on plagiarism and understand the definition of plagiarism. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions in accordance with the policies and procedures of the University. A mark of zero may be awarded and the reason for that mark will be recorded on my file.

I confirm that the Originality Score provided by TurnItIn for this report is 10%.

Zita Koczka

Acknowledgements

I would like to thank my husband, Elliot for his encouragement and for proofreading my work.

I would also like to thank my supervisor, Dr Darryl Charles for his guidance throughout and all the extra resources he recommended.

Contents

[1 Background 1](#_Toc122285632)

[1.1 Problem Statement 1](#_Toc122285633)

[1.2 Aim and objectives 2](#_Toc122285634)

[1.2.1 Aim 2](#_Toc122285635)

[1.2.2 Objectives 2](#_Toc122285636)

[1.3 Summary of the report 2](#_Toc122285637)

[2 Research & Analysis 4](#_Toc122285638)

[2.1 Literature Review 4](#_Toc122285639)

[2.1.1 The difference between linear and branching story readers 4](#_Toc122285640)

[2.1.2 Branching storylines within video games 4](#_Toc122285641)

[2.1.3 Branching storylines in other media 6](#_Toc122285642)

[2.2 Investigatory Process 6](#_Toc122285643)

[2.3 Requirements Specification 8](#_Toc122285644)

[2.3.1 External Interface Requirements 9](#_Toc122285645)

[2.3.2 Functional Requirements 9](#_Toc122285646)

[2.3.3 Non-functional Requirements 11](#_Toc122285647)

[2.4 Project Management 11](#_Toc122285648)

[3 Design 13](#_Toc122285649)

[3.1 Game Objectives 13](#_Toc122285650)

[3.1.1 Genre 13](#_Toc122285651)

[3.1.2 Title Screen and Start Screen 13](#_Toc122285652)

[3.1.3 Character customisation 14](#_Toc122285653)

[3.1.4 Introduction to the game 14](#_Toc122285654)

[3.1.5 Loading screen 15](#_Toc122285655)

[3.1.6 Game Screen 15](#_Toc122285656)

[3.1.7 Core Game Mechanics 16](#_Toc122285657)

[3.1.8 The Player 17](#_Toc122285658)

[3.2 Design Rationale 18](#_Toc122285659)

[3.3 Modelling 19](#_Toc122285660)

[3.3.1 Entity-Relationship Diagram 20](#_Toc122285661)

[3.3.2 Database Schema 21](#_Toc122285662)

[3.4 Testing 21](#_Toc122285663)

[3.4.1 Unit Testing 21](#_Toc122285664)

[3.4.2 Progression Testing 22](#_Toc122285665)

[3.4.3 User Testing 23](#_Toc122285666)

[4 Summary 24](#_Toc122285667)

[5 References 25](#_Toc122285668)

[6 Appendices 26](#_Toc122285669)

[6.1 Appendix A Design Models 26](#_Toc122285670)

[6.2 Appendix B Questionnaire/Survey Results 27](#_Toc122285672)

[6.3 Appendix C Concept Art 33](#_Toc122285673)

**Table of Figures:**

[Figure 2.1.1: Detroit: Become Human flow chart of story progression and in-game choices. 5](#_Toc122285674)

[Figure 2.2.1: Initial sketch of in-game map, list of enemy entities and story-progression items. 8](#_Toc122285675)

[Figure 2.2.2: More detailed version of the map in Figure 2.2.1, created with Articy Draft. 8](#_Toc122285676)

[Figure 2.4.1: Project timeline. 11](#_Toc122285677)

[Figure 2.4.2: Detailed design, development, and testing timeline. 12](#_Toc122285678)

[Figure 3.1.1: Start Screen illustration. 13](#_Toc122285679)

[Figure 3.1.2: Character customisation screen illustration. 14](#_Toc122285680)

[Figure 3.1.3: Game introduction screen illustration. 14](#_Toc122285681)

[Figure 3.1.4: Game screen illustration. 15](#_Toc122285682)

[Figure 3.1.5: Character design illustration. 17](#_Toc122285683)

[Figure 3.1.6: Character attack movement illustration. 17](#_Toc122285684)

[Figure 3.1.7: Character axe interaction illustration. 17](#_Toc122285685)

[Figure 3.3.1: ER diagram. 20](#_Toc122285686)

[Figure 6.2.1: Questionnaire question 1. 27](#_Toc122285687)

[Figure 6.2.2: Questionnaire question 2. 27](#_Toc122285688)

[Figure 6.2.3: Questionnaire question 3. 28](#_Toc122285689)

[Figure 6.2.4: Questionnaire question 4. 28](#_Toc122285690)

[Figure 6.2.5: Questionnaire question 5. 28](#_Toc122285691)

[Figure 6.2.6: Questionnaire question 6 part 1. 29](#_Toc122285692)

[Figure 6.2.7: Questionnaire question 6 part 2. 29](#_Toc122285693)

[Figure 6.2.8: Questionnaire question 7. 29](#_Toc122285694)

[Figure 6.2.9: Questionnaire question 8. 30](#_Toc122285695)

[Figure 6.2.10: Questionnaire question 9. 30](#_Toc122285696)

[Figure 6.2.11: Questionnaire question 10. 31](#_Toc122285697)

[Figure 6.2.12: Questionnaire question 11. 31](#_Toc122285698)

[Figure 6.2.13: Questionnaire question 12. 31](#_Toc122285699)

[Figure 6.2.14: Questionnaire question 13. 32](#_Toc122285700)

[Figure 6.2.15: Questionnaire question 14. 32](#_Toc122285701)

[Figure 6.2.16: Questionnaire question 15. 32](#_Toc122285702)

**Table of tables:**

[Table 3.1.1: Player attack and health loss functionality illustration. 18](#_Toc122285703)

[Table 3.3.1: ER diagram specifications in a table format. 20](#_Toc122285704)

[Table 3.4.1: Unit tests example test cases. 22](#_Toc122285705)

[Table 3.4.2: Example progression test cases. 22](#_Toc122285706)

[Table 3.4.3: Example user test cases. 23](#_Toc122285707)

[Table 6.1.1: Example JSON file. 26](#_Toc122285708)

[Table 6.3.1: Concept art of a neon blue rat. 33](#_Toc122285709)

[Table 6.3.2: Concept art of a deformed raven. 33](#_Toc122285710)

[Table 6.3.3: Concept art of a neon green frog. 34](#_Toc122285711)

[Table 6.3.4: Concept art of a blue fox with glowing eyes. 34](#_Toc122285712)

Abstract

The popularity of the role-playing genre within video games has been soaring for a long time. However, creating an immersive and fun game within the genre is still challenging. The genre is dominated by games with linear storylines, making it hard for players to find a game that stands out within the genre they like.

A game must have unique or innovative functionalities to stand out within the role-playing genre. For example, one unique feature of branching storylines is the different paths players can take, which are influenced by in-game decisions. Unfortunately, creating such a story normally requires great planning and significant additional content. This project aims to solve this issue by looking at ways in which it is possible to create the illusion of agency within the game without the player losing the feel of the weight of decisions.

The game's primary focus will be on creating a mystery that can be solved in multiple ways depending on the player's decisions in-game. For example, they can spend most of their time exploring the world and not bother with the information they can gain from the non-player characters. Alternatively, they can explore a minimal amount and find the required items and answers from dialogues and other information that non-player characters or enemy entities provide

# Background

Within video game development, the narrative has often been treated as a secondary component of the game design, as seen in successful games like *Minecraft* (Mojang Studios, 2009 onwards), *Super Mario Bros.* (Nintendo, 1985 onwards), and the *Crash Bandicoot* series (*Naughty Dog,* 1996 onwards). Previously game development has separated narrative and gameplay components because they provide different modes of engagement for the players (Cameron, 2017). A more valuable driver of engagement is gameplay, an active mode of engagement; players directly interact within the digital spaces. As a result, the narrative is mainly used as a passive medium, interrupting gameplay with cutscenes which can hinder immersion.

However, games with a strong emphasis on story content that changes significantly based on player decisions are increasingly popular. As a result, there is a demand for video games where the player's in-game decisions significantly influence the game's ending. We can see this trend in successful video games such as the *Mass Effect* series and *Detroit: Become Human*.

Video games are not the only type of entertainment where branching or interactive storylines are a driving factor in their success. The popularity of “choose your own adventure books”, for example, *Little Kid, Big City!: New York*, by Beth Beckman (2021), interactive media such as *Black Mirror: Bandersnatch* and *You vs Wild* series(Netflix, 2018, 2019, 2021), as well as tabletop Role-Playing Games (RPG) like *Dungeons & Dragons*, create a need for more video games with immersive and branching storylines.

## Problem Statement

With the increasing amount of story content associated with branching storylines comes an increase in the effort required from the author of those games. Despite the increase in popularity of interactive story-based games, it is unclear if a typical player can appreciate the amount of content presented. It would require multiple playthroughs with different decisions made in-game to experience the full story with all its endings (Fendt, 2012).

The overlapping of the features of linear and branching storylines, where some decisions lead to the same outcome but still create multiple paths in the story with different endings, gives the player the illusion of agency.

## Aim and objectives

This project aims to design a 2D top-down video game which combines the successful and immersive genre of RPG and the highly interactive gameplay of branching storylines. The story's focus is a mystery that the player will be able to solve in multiple ways by both exploring the in-game world and gathering clues from the non-player characters (NPC).

### Aim

To instil stronger feelings of agency within a player and thus enhance the fun of gameplay by implementing features such as choice-based story progression, experience-based levelling system, inventory and craftable items, and a health and stamina system.

### Objectives

1. Research linear and branching storylines within various media
2. Draw a conclusion on the best approach to design the video game with branching storylines
3. Write up requirements specifications based on research and analysis
4. Design game objectives and modelling based on requirements
5. Game development process
6. Testing

## Summary of the report

The project aims to create a game that is immersive as well as enjoyable. The following chapters will elaborate on the findings of previous studies on linear and branching storylines. In addition, the subsequent chapters detail the research and analysis process on the literature on topics such as video games and other media with branching storylines. As well as findings from a survey to find the right path to solve the issues branching storylines bring up, such as the amount of content needed to present the player with multiple story paths and endings.

The research and analysis section follows, which contains the literature reviews, the investigatory process, and the requirements specifications, as well as the project management section, which includes the project plan. This plan puts the project timeline into perspective, allowing the start of the design process. The Design chapter will consist of multiple subchapters detailing each element of the design process, starting with the Game Objectives, which will contain multiple points on the game design. Secondly, the design rationale will detail the reasoning behind the decisions made for the design process. Lastly, the modelling section will provide more information on entity relationships and database schema.

# Research & Analysis

The following chapter contains in-depth research and analysis on various media with linear or branching storylines, as well as data gathered to assess what players find necessary within a video game's story and other features that they find desirable.

## Literature Review

### The difference between linear and branching story readers

The earliest form of storytelling was orally from person to person, which resulted in multiple versions of the same story as each teller changed or added to the narrative, which led to minor or significant differences in plot and motifs, resulting in the diversity of alternative stories. This particular type of diversity can be found in the interactive fiction genre, which contains multiple parallel alternative stories causing an inherent diversity of story plots. In addition, branching storybooks became popular in the 1980s as a type of gamebook with a series of "create your own adventure", meaning that the choices made by the reader affect the plot of the story or the game.

The experience may change when the reader reads a non-interactive story multiple times, as the reader already knows the story. This means that even if the reader re-reads a story, their feelings may differ from the first reading, despite them reading the same story. However, interactive-story readers do not feel like they are re-reading a story until they understand the consequences that their choices resulted in, even if the story is unchanged between the readings. There are multiple reasons for re-reading a non-interactive story. These reasons included back-tracking, recalling details or understanding information that might have been missed during the first reading, recapturing the first reading experience, or looking at the story analytically (Sellis, 2019). While readers of interactive stories re-read in an attempt to reach some form of closure, when they reach that closure, they focus on their understanding of the story and switch to either a simple re-reading or a more analytic re-reading (Sellis, 2019).

### Branching storylines within video games

With the rise of the popularity of video games, one of the genres that have constantly been popular is role-playing video games. One of the most important aspects of video games is their story and the narrative used to tell those stories. However, with RPG, the attractive feature is the player's ability to control that story to sculpt their experiences with their decisions during the game.

For each path of the branching storyline, more content, dialogue, voice acting, and testing must be done and written, thus creating customisable story experiences. Therefore, by creating the illusion of control, players can have a false sense of agency within the game without making all the content highly desirable (Fendt et al., 2012). Not showing the consequences of the player’s in-game decisions directly but still creating scenarios where the player must make decisions, even if that decision has minimal or no implications for the game’s progression, will give the player a false sense of agency.

This solution, however, is not applicable in games where each decision and its consequences are shown to the player (see Figure 2.1.1). Games like *Detroit: Become Human,* is a game that shows the player's in-game decisions in a flow chart. It also shows the percentages of the worldwide decision behaviour of players who contributed their data through the PlayStation network. For players to access all statistics of each decision path, they must play every outcome at least once, meaning the game must be played multiple times for full completion (Holl, 2019).

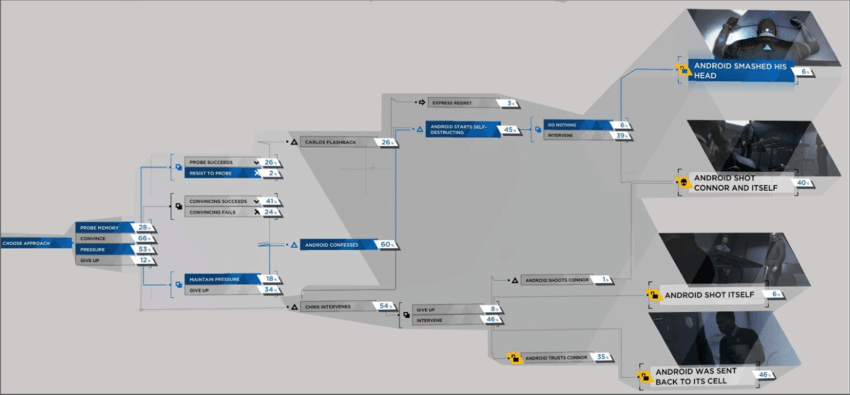


Figure 2.1.1: Detroit: Become Human flow chart of story progression and in-game choices.

Another excellent example of a story-driven video game is Naughty Dog's big hit, *The Last of Us* (2013). The game's narrative is treated as a crucial component alongside gameplay instead of as a subordinate feature. To make the narrative crucial, various design methods where the narrative and gameplay overlap and interact are implemented, which means that the narrative and the gameplay support one another instead of competing for the player's attention (Cameron, 2017).

### Branching storylines in other media

However, the player's control over the story in games like *Detroit: Become Human* and *The Last of Us* dwarfs in comparison to the Tabletop RPG game *Dungeons & Dragons.* In this highly interactive board game, the story is influenced by multiple factors, the player's previously designed character and their backstory, the rolling of a dice, the battle maps, and other players' decision-making. All these factors create a highly immersive and interactive game, where not only the Game Master but also the players are influencing the storytelling (Haslett, 2020).

To introduce the players to one another and invest themselves in the storyline, Dungeons & Dragons campaigns often start with a similar exposition. However, these stories very quickly differ as players insert their characters and decisions into the narrative. These stories are not passive ones like those in a book, a play or a movie; instead, they use the dialogue between the Dungeon Master and the players to shape the narrative (Haslett, 2020).

Dungeons and Dragons sessions can last from a couple of hours to a day, with adventures that are either a one-shot or a part of a campaign that can last multiple sessions. Furthermore, these adventures are part of a campaign that can last years.

## Investigatory Process

To identify specific issues and how to solve those, a survey was undertaken with a group of 22 people, which probed their video game experiences (see Figure 6.2.1 – Figure 6.2.16 in Appendix 6). The results from the survey have shown that the most popular gaming genres are RPG, strategy, and puzzle games. 81.8% of people submitted "Yes" or "Maybe" as their response to the question asking whether their in-game decisions should influence the story. Two-thirds of respondents, on a scale from very weak to very strong, agreed that the influence should be on the stronger side. Most people said yes when asked if they would like to see how their decisions influenced the story in a visual format. Their main reasonings are that they want to see all possible paths or how much their decisions have influenced the story. The majority agreed that to experience each story path of the game, they will likely replay the game.

As this project aims to create an RPG game, the survey asked further questions about people’s experiences with the genre. Over 70% of the people have said that their main focus is the story, emphasising world-building and well-written dialogues. When asked what the most important aspects of an RPG game are, they answered: "exploration" and "story and setting", with "combat", "character customisation", and “graphics” close behind.

To further understand their experiences with branching storylines, the survey asked if they had read a book with that feature. Only three people answered yes, and when asked about their experiences, they answered that books with branching storylines are simpler than linear storylines. Often, the story itself is weaker but still enjoyable. People with no experience with branching storylines in books said they would consider reading one.

The solution to the issue of a weaker story that comes with a branching storyline is to look at other important factors that can create an immersive and fun experience for the player. Considering the survey's feedback, emphasising other game features, such as character customisation, graphics and exploration of the map, eases the heavy burden branching storylines carry.

Multiple tools have been considered during the research process to help with development. One of these tools, called *Articy Draft*, is an interactive game content and storytelling management system (Articy, 2022). This tool is a great aid that allows the user to start with a rough sketch draft (see Figure 2.2.1) and turn it into a more refined map with markings for locations and items (see Figure 2.2.2). This tool also allows the content to be imported directly into *Unity*, the chosen game engine for this project. *Unity* is a platform for creating and operating interactive, real-time 3D content, with *Unity* being their royalty-free version. For further information on why this tool was chosen, refer to the “3.2 Design Rationale” section.

Diagram

Description automatically generated

Figure 2.2.1: Initial sketch of in-game map, list of enemy entities and story-progression items.

Diagram

Description automatically generated

Figure 2.2.2: More detailed version of the map in Figure 2.2.1, created with Articy Draft.

## Requirements Specification

This chapter contains a list of external interface requirements, which describes heads-up display (HUD), the way information is visually relayed to the player. Next, the functional requirements describe the game’s features and functions, and lastly, the non-functional requirements give information on the general properties of the system.

### External Interface Requirements

**EIR-1.** MainWindow

**EIR-1.1.** Main Menu

**EIR-1.1.1.** New Game

**EIR-1.1.2.** Load Game

**EIR-1.1.4.** Exit

**EIR-2.** New Game Window

**EIR-2.1.** Character customisation

**EIR-2.1.1.** Enter name

**EIR-2.1.2.** Choose hair

**EIR-2.1.3.** Choose body

**EIR-2.1.4.** Choose top clothing

**EIR-2.1.5.** Choose bottom clothing

**EIR-2.1.6.** Start Game

**EIR-3.** Game Window

**EIR-3.1.** Inventory

**EIR-3.1.1.** Crafting

**EIR-3.2.** In-game time

**EIR-3.3.** Health status

**EIR-3.4.** Stamina status

**EIR-3.5.** Map

**EIR-3.5.1.** Discovered locations

### Functional Requirements

**FR-1.** After starting the application, the user can choose between the Main Menu options

**FR-1.1.** The user can start a new game

**FR-1.2.** The user can load a game

**FR-1.3.** The user can exit the application

**FR-2.** When the user starts a new game

**FR-2.1.** The user can customise their characters; customisation options are according to **EIR-2.1.**

**FR-2.2.** After the character customisation, information on the beginning of the story will appear in text format

**FR-3.** Character movement

**FR-3.1.** The user can control the character with the 'W',' A', 'S', and 'D' buttons on the keyboard

**FR-3.2.** The character can move faster when the 'LShift' button on the keyboard is held down

**FR-3.2.1** The character can move slower when the 'LCtrl' button on the keyboard is held down

**FR-3.2.3** The character's stamina will be drained faster or slower depending on the character's movement speed

**FR-4.** The in-game inventory can be opened with the 'I' button on the keyboard

**FR-4.1** When the inventory is open, the user can craft items

**FR-5.** The user can open the map with the 'M' button on the keyboard

**FR-5.1** The map will show the entirety of the game world

**FR-5.2** The map will show the location of already discovered items and locations that are uncovered by the user

**FR-6.** The user will be able to see the progression of in-game time

**FR-7.** The user will be able to see the character's health status

**FR-8.** The user will be able to see the character's stamina status

**FR-9.** The user will be able to open a secondary menu by pressing the 'ESC' button on the keyboard

**FR-9.1** The user will be able to save the game

**FR-9.2** The user will be able to exit the game

**FR-10.** The user will be able to interact with items with the 'E' button or by clicking on the item when their selected inventory spot is empty

**FR-11.** The user will be able to perform an action when they press the left mouse button depending on the item in the selected inventory spot

**FR-11.1.** The user will be able to gather resources

**FR-11.2.** The user will be able to perform combat actions

**FR-12.** NPCs

**FR-12.1.** Multiple NPCs will be located around the map

**FR-12.2.** When interacting with an NPC, their respective dialogue, based on game progression, will appear

**FR-13.** Combat

**FR-13.1.** Multiple types of enemy entities

**FR-13.2.** When hit by an enemy entity, the player's character will lose health

**FR-13.3.** When the player performs a combat action, the character will lose stamina

**FR-13.4.** When the character loses all their stamina, they will walk slower and will not be able to perform actions that require stamina points

**FR-13.5.** When the character loses all their health, they will respawn in a certain location and will lose some of their non-story progression items from the inventory

**FR-14.** Story progression will be dependent on collectable items, discovered locations and certain NPC dialogues

### Non-functional Requirements

**NFR-1.** The game will be developed with C# using *Unity* as the game engine

**NFR-2.** Three different game progressions can be saved and loaded one at a time

**NFR-3.** The game will run smoothly

**NFR-4.** The game will not have a long loading time

## Project Management

A project plan has been drawn up at the start to help plan and visualise specific timeframes for the project (see Figure 2.4.1). This plan includes Semesters 1 and 2 from weeks 1 to 13, with some weeks omitted to make the timeline easier to see. With the red dots representing deadlines and the weeks highlighted for each step of the project, it is easy to see how much time is allocated for each task. During the game design process, a more detailed plan for the development and testing phases was also created (see Figure 2.4.2).

Graphical user interface, application, Word

Description automatically generated

Figure 2.4.1: Project timeline.

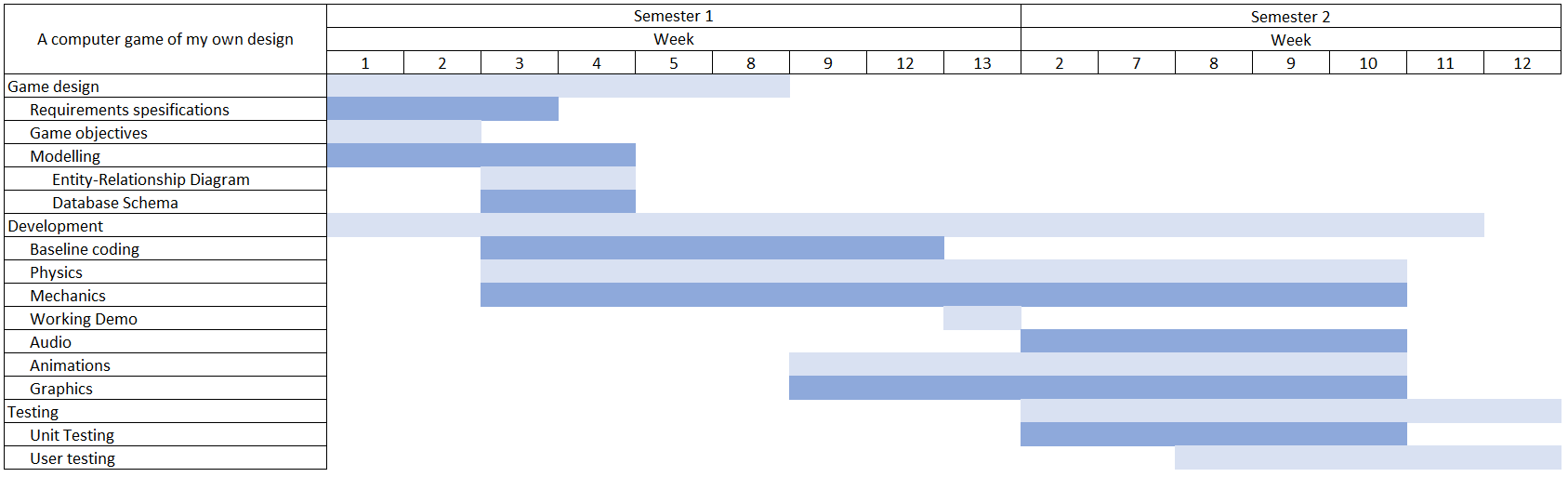


Figure 2.4.2: Detailed design, development, and testing timeline.

# Design

## Game Objectives

The game will start with an explanation in text format, which tells the player that they are currently working as a biologist. The government tasked the player to discover why strange animals appear on a specific island. The player arrives at this island, and to solve the mystery, they must spend multiple days capturing the weird-looking creatures. To solve the mystery, the player can spend time exploring the island, which may lead to them finding locations or items required to advance the story. Alternatively, they can spend their time interacting with the NPCs, who may give the player information on where these locations or items are. To obtain some items or to follow specific paths in the branching storyline, the player must interact with specific NPCs multiple times, but it is not required to solve the mystery.

### Genre

The game is a 2D, top-down RPG mystery with features such as combat, health, stamina, and world exploration.

### Title Screen and Start Screen

As the game loads up, the player will be greeted with a title splash screen, which will turn into the main menu, which will show the title and three options; new game, load game and exit (see Figure 3.1.1).

Graphical user interface

Description automatically generated with low confidence

Figure 3.1.1: Start Screen illustration.

### Character customisation

Once a new game starts, the character customisation page opens (see Figure 3.1.2), allowing the player to enter their character's name and change how it looks. Once the player is done with the character customisation, they can press a button to start a new game.

Diagram

Description automatically generated

Figure 3.1.2: Character customisation screen illustration.

### Introduction to the game

When character customisation is done, and a new game starts, the player will see a window that will introduce the game’s story in a text format (see Figure 3.1.3). This introduction is the story's setting and will allow the player to learn about the character and the game's objective.

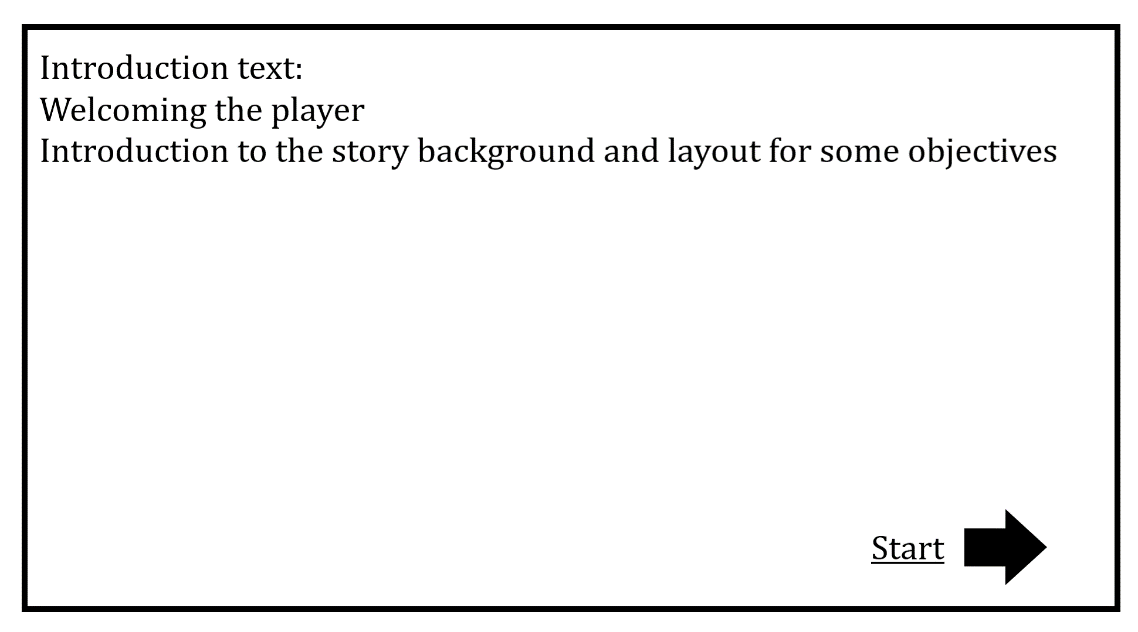


Figure 3.1.3: Game introduction screen illustration.

### Loading screen

The loading screen will appear between windows as a transition, allowing the game to load while also providing tips or hints to the player regarding the game, such as how to move faster or slower or providing smaller objectives to progress the story.

### Game Screen

After the introduction to the game, the player will see the game screen (see Figure 3.1.4), where they can move the character around the world to explore. The player will also see the inventory, which will have nine spots, the in-game time and the health and stamina status. From this screen, the player can also open the inventory with the 'I' button on the keyboard, popping up an inventory screen with multiple tabs, which will show the items in the inventory recipes to craft and information on the story progression. If the player presses the 'M’ key, a map view will be displayed, aiding the player in navigating the world. Finally, by pressing the ’ESC’ key or clicking on the menu button, the player will see the options to save the game progression or exit the game, which will take the player back to the home screen.

Diagram

Description automatically generated

Figure 3.1.4: Game screen illustration.

### Core Game Mechanics

#### Interactive Items

To progress the story, the player must find certain interactive items. The player has multiple ways to gather information about those items. For example, they can talk to the NPCs or explore the world.

#### Interactive Locations

There will be locations that are only accessible by progressing the story and gathering interactive items. These locations can be found by either exploring the world or gathering information from the NPCs.

#### Enemy entities

Enemy entities, wild mutant animals (see list on the left in Figure 2.2.1) that the player can catch with certain items, will appear at specific locations depending on the in-game time. These entities will attack the player, and to destroy these entities, the player must perform combat actions requiring certain items. The enemy entities will have a health status and an attack damage point that will take off a certain amount from the player’s health status when hit. In addition, enemy entities will drop experience points when eliminated. Enemy entities' health and damage base points will scale with the player’s level.

#### NPCs

The player will find NPCs they can interact with at specific locations. The more they interact with the NPCs, the more information the player can learn from them. These NPCs have a name, backstory, and information for story progression that the player can learn from their dialogues. In addition, with each interaction, the player’s and the NPC’s relationship status will rise, making the NPC more likely to reveal specific information.

#### Items and crafting

The player can collect story progression items and non-story-related items. Non-story progression items the player can collect worldwide will be needed to craft other un-collectable items. When the player opens the inventory, they can navigate to the crafting tab and see craftable items, and the items needed to craft them.

#### Soundtrack and sound effects

The game will include a soundtrack that will play during the game and sound effects that will occur when the player performs actions.

#### Screen

The game’s screen size will be 1920pixels-by-1080pixels and will consist of 16pixel-by-16pixel tiles

### The Player

#### Character measurements and animation specification

The player’s character size will be one tile wide and two in height, with each tile being 32 pixels by 32 pixels. The character will have walk animations (see Figure 3.1.8.1), combat, and other action animations (see examples Figure 3.1.8.2, Figure 3.1.8.3), each action accompanied by a sound effect.



Figure 3.1.5: Character design illustration.



Figure 3.1.6: Character attack movement illustration.



Figure 3.1.7: Character axe interaction illustration.

#### Character Mechanics

The character will have a health status which will deplete depending on the enemy entity's attack damage points (see Figure 3.1.1). The character will have a stamina status which will deplete depending on the actions performed by the player. The base status for health and stamina will increase depending on the player’s level. The player will be able to increase their level by collecting experience points earned by eliminating enemy entities and finding story progression necessary items or locations. The character's movement speed will depend on the keyboard buttons the player is pressing. Depending on the movement speed, the character’s stamina will deplete proportionately. The faster the character moves, the quicker the stamina depletes, and the slower the character moves, the slower it will deplete. The player will have a base attack point which will determine the damage the player will cause to enemy entities; this will increase depending on the player’s level. If the player’s health completely depletes, the player will respawn at a specific location and lose some non-story progression items. Suppose the player’s stamina status depletes completely. In that case, the player will walk slower and be unable to perform actions requiring stamina points, and the stamina status will not fill up completely for the next in-game day. To refill the health and stamina status, the player can use certain items or progress the in-game day.

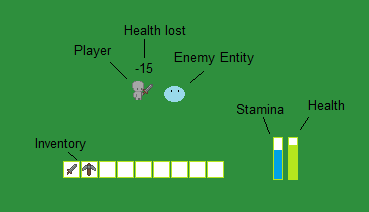


Table 3.1.1: Player attack and health loss functionality illustration.

## Design Rationale

The project aims to create a game that will encourage the player to try and solve the mystery and keep them entertained. Multiple design decisions were set during the initial planning process, such as the game genre, the art style and the story. The game design started with a simple idea of a story of a biologist trying to solve the mystery of the bizarre-looking creatures on an island (for concept art see Figure 6.3.1 – Figure 6.3.4).

The graphics of the game is 2D, which allows for simpler animations and art style, which is popular within the RPG genre. Furthermore, *Krita*, a free tool, will be used to create the graphics for the game, allowing creative freedom for the art (Krita Foundation, 2005 onwards).

A JavaScript Object Notation (JSON) file will store items and objects information, which uses human-readable text to store data. *Unity Enterprise* uses C# as the coding language, which provides the functionality to quickly parse JSON files creating a lightweight database for the project.

To create gameplay that is more engaging and entertaining, some simple artificial intelligence (AI) will be added to the NPCs. This will include functionalities such as moving towards the player when the NPCs are within a certain distance or appearing at different places on the map at different in-game times.

The decision to use the *Unity* game engine was made early in the design process. It is a great tool which uses C# as the primary coding language. Visual Studio is an integrated development environment (IDE) with great *Unity* support, making the development process more manageable. The *Unity* Engine has many built-in tools, which makes assets and the code more modular. The solutions for this are *Unity’s* Asset Store, which provides a platform for developers to share their libraries and assets. The assets can be copied into the project and modified for personal needs. The other platform aiding reusability is the package manager, which allows the user to plug open-source packages into the project (Buckley, 2020).

Other game engines, such as *Unreal,* were considered. However, *Unity* was chosen based on its toolsets for 2D game development and the easily accessible *Unity Learn* platform, which provides ability-based learning pathways (Unity Learn, 2020).

## Modelling

This chapter showcases and describes the diagrams used to aid the development process.

### Entity-Relationship Diagram

The Entity-Relationship (ER) Diagram (see Figure 3.3.1) helps visualise the basic design used as the basis of the database. ER diagrams specify the data that will be stored, the entities and their attributes. They also help to visualise the relationship between the entities.

Diagram

Description automatically generated

Figure 3.3.1: ER diagram.

Table 3.3.1: ER diagram specifications in a table format.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Player | Inventory | Items | Map | NPC | Enemy Entity |
| * Name * Speed * Health Points * Stamina Points * Experience * Attack Points * Level * Sprite * Character Portrait * Animations | * Quick Equip Slots (9) * Standard Slots (16) | * ID * Name * Damage Points * Speed Modifier * Craftable (Boolean) * Interactive (Boolean) * Story Progress Item (Boolean) | * Interactive Locations * Interactive Items * NPCs | * Name * Location * Dialogues * Relationship with player * Sprite * Animations * Portrait | * ID * Name * Level * Location * Attack Points * Health Points * Speed * Experience Dropped * Animations |

### Database Schema

JSON files are written in a human-readable format, which allows for quick editing or additions when necessary, and are easily parsed, providing a lightweight database. *Articy Draft* is a great tool with functionalities to add and edit items and entities, which can then be exported as JSON files.

This data will be embedded within the application, meaning all relevant information can be retrieved in a single query. Embedding too much data in one query can cause performance issues. However, it is easy to prevent that by limiting the size of the document and the data stored within the JSON file (see example Figure 6.1.1).

To meet the necessary

functional requirements, the database is designed based on the ER diagram (see Figure 3.3.1). A JSON file will be used as a database, which enables storing multiple collections of arrays for “items”, “NPCs”, and “enemy entities”, each section holding the information detailed in Table 3.3.1.

The player has an inventory with 25 slots, of which 9 are quick-equip slots that can be accessed via the numpad. The inventory can hold multiple items, which can be moved around inside the inventory. The player also has an interactive map which will show discovered locations and items. Certain items can be crafted, while others can be foraged or found at specific locations on the map.

## Testing

Testing methodologies such as unit testing, progression testing and user testing will ensure that the application will perform all functionalities defined in the requirements.

### Unit Testing

Unit testing is performed throughout the development process to validate that individual units of code perform as expected. This enables the testing of individual components rather than the whole application. Performing unit testing from the early stages of development saves time and effort as it highlights development errors that require fixing as soon as they appear. Unit testing is reusable, thus providing reassurance that future modifications will not break existing functionalities within the software. Example unit test cases are shown below (see Table 3.4.1).

Table 3.4.1: Unit tests example test cases.

|  |  |
| --- | --- |
| Test case 1:  **When** the inventory is empty  **Given** the player picks up an item | Expected result:  **Then** inventory contains the picked-up item |
| Test case 2:  **When** the player moves around the map  **Given** the player is not at the edge of the map | Expected result:  **Then** the camera will follow the player as its centre |

### Progression Testing

This testing method will be used to ensure that the progression will follow the correct path of the branching storyline based on the player’s decisions. Progression testing will be performed at least as many times as there are paths that can be taken within the branching storyline. It will also ensure that the player cannot jump from one path to another without the requirements for that jump. Example progression test cases are shown below (see Table 3.4.2).

Table 3.4.2: Example progression test cases.

|  |  |
| --- | --- |
| Test case 1:  **When** the player follows path 1 of the branching storyline  **Given** the player performed all requirements for the path | Expected result:  **Then** the player will reach ending 1 |
| Test case 2:  **When** the player follows path 1 of the branching storyline  **Given** the player has performed a requirement to change to path 2  **And** the player then performs all requirements for the new path | Expected result:   * **Then** the player will reach ending 2 |

### User Testing

User testing will be one of the last steps in the testing phase. This will allow the application to be tested by a fresh set of eyes, making it easier to discover bugs and errors within the software. Third-party feedback will be invaluable to improving the overall user experience. Example user test cases are shown below (see Table 3.4.3).

Table 3.4.3: Example user test cases.

|  |  |
| --- | --- |
| Test case 1:  **When** the player performs an attack action  **Given** the player has the required items equipped in hand | Expected result:  **Then** the player character will perform the attack function accompanied by the animation and sound effect |
| Test case 2:  **When** the player walks to the edge of the map | Expected result:  **Then** the player will not be able to walk off the map |

# Summary

The first chapter of the report, “Background”, gives all the necessary information on the aims and objectives of the project. It further elaborates that game development has previously separated the narrative from the gameplay, providing different kinds of engagement for the players. However, there is a rise in the popularity of games with the story as the focus. The project originated from a simple idea of a two-dimensional, role-playing mystery game where the player’s decisions impact the storyline.

The next chapter, “Research & Analysis”, gives further information on literature that has explored linear and branching storylines in different kinds of media. In that chapter, the result of the survey that was conducted for the project can be found. Following the survey results, the functional and non-functional requirements specifications were defined. The final section elaborates on the project planning.

The “Design” chapter follows. During the design process, diagrams were drawn to visualise the game's look and specify the player’s character sprite and mechanics, measurements, looks, and core gameplay mechanics like enemy entities, items, crafting, locations, navigation, and NPCs. This chapter includes the design rationale for the decisions made during the design process, such as what tools to use and what game mechanics are implemented. In this chapter, there is a section that contains the ER diagram and database schema and further information on the testing that will be used.

# References

Articy (2022) *articy draft*. [online] Available at: <https://www.articy.com/en/>.

Beckman, B. (2021). *Little Kid, Big City!: New York*. Quirk Books.

Buckley, D. (2020). *Unity vs. Unreal – Choosing a Game Engine*. [online] GameDev Academy. Available at: <https://gamedevacademy.org/unity-vs-unreal/>. [Accessed 18 November 2022]

Cameron, K. (2017). *NARRATIVE AND GAMEPLAY DESIGN IN THE STORY-DRIVEN VIDEOGAME: A CASE STUDY ON THE LAST OF US*. [online] Available at: <https://core.ac.uk/download/pdf/132419502.pdf>.

Fendt, M.W., Harrison, B., Ware, S.G., Cardona-Rivera, R.E. and Roberts, D.L. (2012). Achieving the Illusion of Agency. *Interactive storytelling*, pp.114–125. doi:10.1007/978-3-642-34851-8\_11.

Haslett, O. (2020). It crits different: analysis of Dungeons & Dragons and tabletop role-playing games as an oral, collaborative, and immersive genre of literacy.

Holl, Elisabeth. (2019). Rise of the machines - Moral decisions in Detroit: Become Human.

Krita Foundation (2019). *Krita | Digital Painting. Creative Freedom.* [online] Krita.org. Available at: <https://krita.org/en/>. [Accessed 18 November 2022]

M. O. Riedl and R. M. Young, “From linear story generation to branching story graphs,” in IEEE Computer Graphics and Applications, vol. 26, no. 3, pp. 23-31, May-June 2006, DOI: 10.1109/MCG.2006.56.

Naughty Dog (2013) *The Last of Us* [Video game]. PS4. Sony Computer Entertainment.

Quantic Dream (2018) *Detroit: Become Human* [Video game]. PS4. Sony Interactive Entertainment America LLC

Sellis, D. (2019), Exploring the story structure of branching books”, OSF Preprints, 3 April, available at: https://doi.org/10.31219/osf.io/xhsyd.

Tactical Studies Rules, Inc (1974) *Dungeons & Dragons* [Boardgame].

Technologies, U. (2022). *Wondering what Unity is? Find out who we are, where we’ve been and where we’re going | Unity*. [online] unity.com. Available at: <https://unity.com/our-company>. [Accessed 18 November 2022]

‌‌Unity Learn. (2020). *Unity Learn*. [online] Available at: <https://learn.unity.com/>.

# Appendices

## Appendix A Design Models

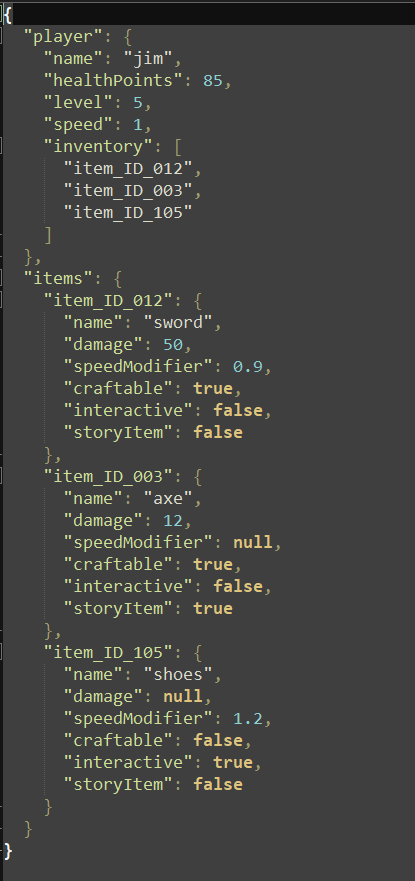


Table 6.1.1: Example JSON file.

## Appendix B Questionnaire/Survey Results

Results of questionnaires/surveys used to evaluate the software and identify requirements

Chart, pie chart

Description automatically generated

Figure 6.2.1: Questionnaire question 1.

Chart, bar chart

Description automatically generated

Figure 6.2.2: Questionnaire question 2.

Chart, pie chart

Description automatically generated

Figure 6.2.3: Questionnaire question 3.

Chart, pie chart

Description automatically generated

Figure 6.2.4: Questionnaire question 4.

Chart, pie chart

Description automatically generated

Figure 6.2.5: Questionnaire question 5.

Graphical user interface, text, application, email

Description automatically generated

Figure 6.2.6: Questionnaire question 6 part 1.

Graphical user interface, text, application

Description automatically generated

Figure 6.2.7: Questionnaire question 6 part 2.

Chart, pie chart

Description automatically generated

Figure 6.2.8: Questionnaire question 7.

Chart, pie chart

Description automatically generated

Figure 6.2.9: Questionnaire question 8.

Chart, pie chart

Description automatically generated

Figure 6.2.10: Questionnaire question 9.

Chart, bar chart

Description automatically generated

Figure 6.2.11: Questionnaire question 10.

Chart, bar chart

Description automatically generated

Figure 6.2.12: Questionnaire question 11.

Chart, pie chart

Description automatically generated

Figure 6.2.13: Questionnaire question 12.

Chart, pie chart

Description automatically generated

Figure 6.2.14: Questionnaire question 13.

Graphical user interface, text, application

Description automatically generated

Figure 6.2.15: Questionnaire question 14.

Chart, pie chart

Description automatically generated

Figure 6.2.16: Questionnaire question 15.

## Appendix C Concept Art

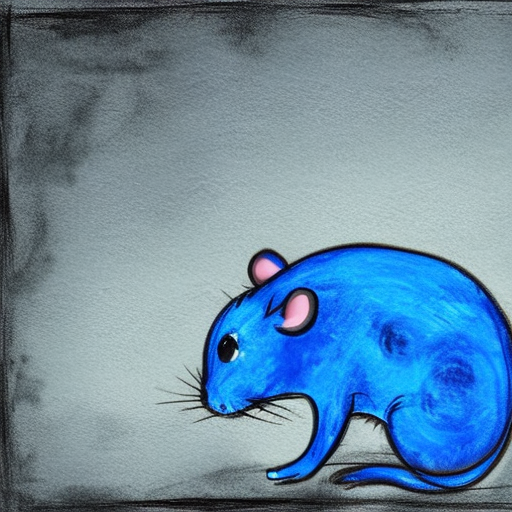


Table 6.3.1: Concept art of a neon blue rat.

A crow on a tree branch

Description automatically generated with low confidence

Table 6.3.2: Concept art of a deformed raven.

A picture containing text, vector graphics

Description automatically generated

Table 6.3.3: Concept art of a neon green frog.



Table 6.3.4: Concept art of a blue fox with glowing eyes.