



UNIT 8 COMPUTER GAMES DEVLEOPMENT

Assignment 2

Learning Aim B and C

Design a computer game to meet client requirements and develop a computer game to meet client requirements

Oliver Collins-Cope
2102775@rutc.ac.uk

Contents

Introduction	2
Purpose of the game.....	2
Target audience	2
Requirements.....	2
Programming languages	2
Intended platform for delivery	2
Game design	2
Data dictionary.....	2
Pseudocode.....	3
Activity diagram	4
Storyboard	5
Visual styles.....	5
Assets	5
Gameplay features.....	8
External feedback and review.....	9
Expansion of feedback	12
Summary of feedback	13
Timeline/Production Schedule	13
Justification	13
Evaluation of design against client requirements	14
Test plan.....	15
Optimisation	16
Evaluation of game against the requirements	16

Introduction

Purpose of the game

The purpose of my puzzle game is to entertain my players and to help train their cognitive skills through different puzzles of varying difficulty. They will incrementally get harder and help to identify the players skill level, therefore showing them where they need to improve on themselves.

Target audience

The target audience for my game is 8+ years old. This is because I do not believe that the game will be interesting for anyone younger than this, and that they also might not have developed the life skills and problem-solving ability to solve the puzzles without outside support, which obviously defeats the purpose of testing the player and encouraging them to improve their skills. Furthermore, it might limit the type of puzzles I can make if the target audience is too young.

Requirements

The hardware requirements to run my game include 8 GB of RAM on the computer, an i5 intel core. Furthermore, the computer game will be based on WASD input, and a mouse to look around. This means that the player will need both a mouse and keyboard for input in order to play. A monitor will be needed to view the game screen and if the player wants, they can use headphones or speakers for audio.

The software for the game will be the actual download of the game, as well as any additional software required, possibly such as .NET and there is also a need to be running the correct OS.

Programming languages

The programming language used to develop will be C#, as that is the programming language of Unity and therefore will be used for development.

Intended platform for delivery

The intended platform for delivery of my game can be adjusted depending on what is needed, however it will initially be developed for windows operating system on PC. It can be built for any OS on PC, however the game will need separate work in the future for it to work on console and other platforms.

Game design

Data dictionary

Data name	Data type	Scope	Size	Example	Validation
Score	Integer	Class	4 bytes	private int score;	score >=0
Lives	Float	Class	4 bytes	private float lives;	lives >= 0
playerName	String	Class	Dynamic	private string playerName;	playerName != PLAYERNAME

Timer	Integer	Class	4 bytes	private int timer;	while timer != 0{ timer = timer - 1 second}
Level	Integer	Class	4 bytes	private int level;	if levelfinish = true{ level = 1; else level = 0 }
Collision	Boolean	Game wide	2 bytes	public Boolean collision;	if collision == true { player move through collision }

Pseudocode

For every game there is a need to intricately design as much as is needed, and to an extent that means writing out the code for the different scripts that will be used inside of the game. As there are many different features within a game, from a character that is able to move around, to a light switch turning off and on, it allows for lots of room to develop pseudocode in order to smoothly integrate your design into development.

The most important feature of my game will be the ability to interact with the environment, and therefore it is important to ensure that the character has a way to detect what they are looking at and whether they can interact with that item, while the item they are interacting with will decide what the interact action does.

This is the pseudocode once the player launches the game:

```
if startGame == TRUE Then
```

```
    Load Scene "level0"
```

```
    Load gameobjects
```

```
else
```

```
    Exit.Application
```

From here, the player is then able to play the game and perform more options/functions such as being able to look at something and interact with it. This is shown below.

The pseudocode below displays how the character detects if they are looking at something:

```
make Ray
```

```
if Ray Hit == TRUE Then
```

```
    if Ray Hit Interact == TRUE Then
```

```
        if Interact Button == TRUE Then
```

```
            Interact Action
```

This is a very basic outline, aka pseudocode, of how the game will check if the player is looking at something. This is the interaction pseudocode and determines how the interacted with object will act.

```
Function Interact Action
```

```
    Object (I.e. box, door, light switch)
```

Function Door

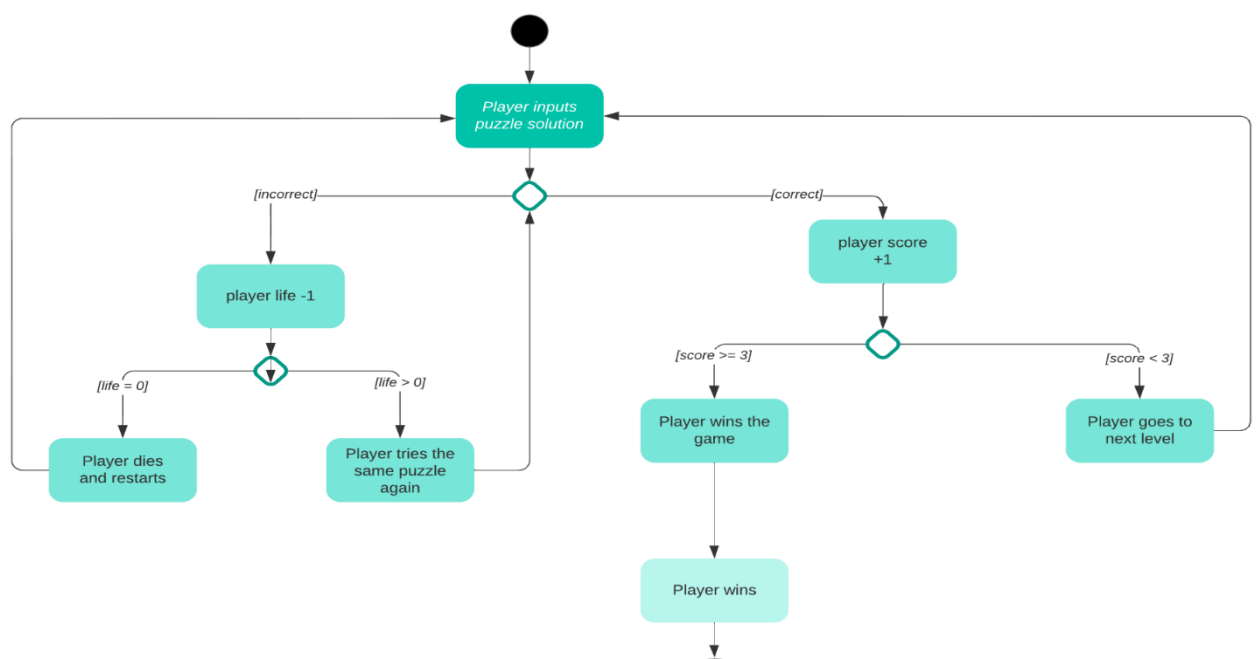
If Key Active == TRUE Then
Door Unlocked == True

If Door Open == False AND Door Unlocked == TRUE Then
Play Animation Open Door

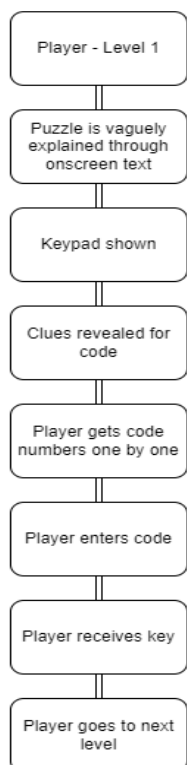
Else if Door Open == TRUE AND Door Unlocked == TRUE Then
Play Animation Close Door

This is another very basic version of the code that will be used for the doors in the game and will be expanded on in development.

Activity diagram



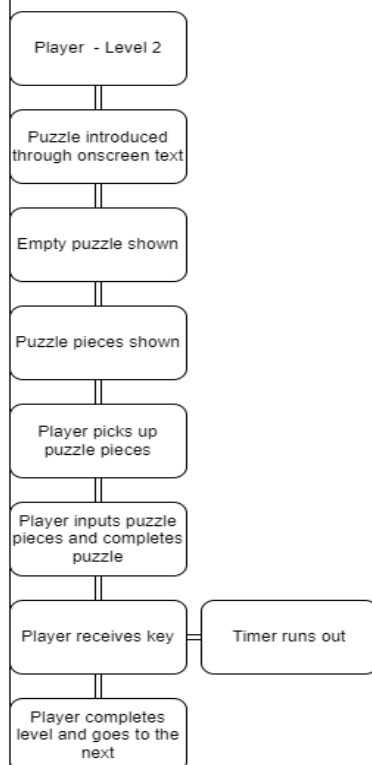
Level 1



Player dies

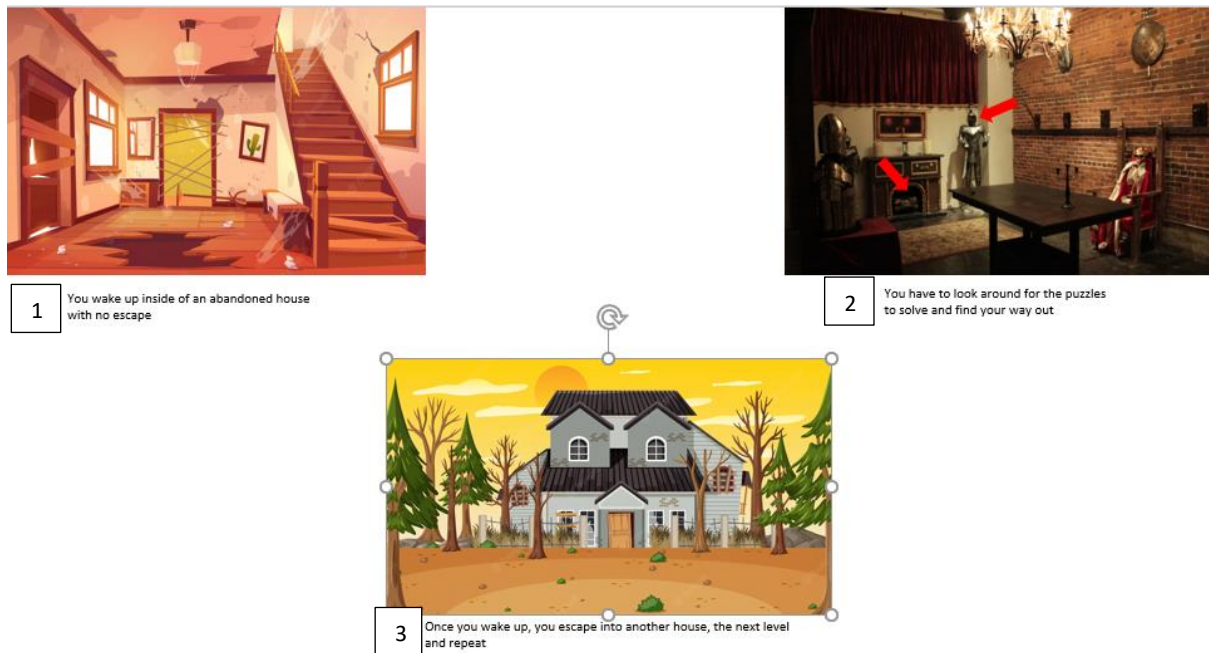
Timer runs out

Level 2



Timer runs out

Storyboard



While extremely vague, the storyboard gives a basic insight into the game and how it will be play out and the general style. The game will have a few different levels and offer a score and life system based on completing and failing the puzzle respectively. Finally, the story board also helps to understand a little about how the game takes place, however as story is not a focus of my game it will not be covered deeply both inside and outside of development.

Visual styles

The visual styles in the game will help to bring it to life and create depth for the player to explore and enjoy it. The world will be in 3D, with most of the textures being plain white or other solid colours, while interactable objects will be a different colour in order to help differentiate the differences between what is part of a puzzle and what is not. Eventually, the plan would be to implement textures and hide the interactable objects in the environment to make the game harder.

There will be only the player character and likely no non-playable characters, though this is subject to change in the future. The perspective of the game will be based in first person with WASD movement and mouse view.

Assets

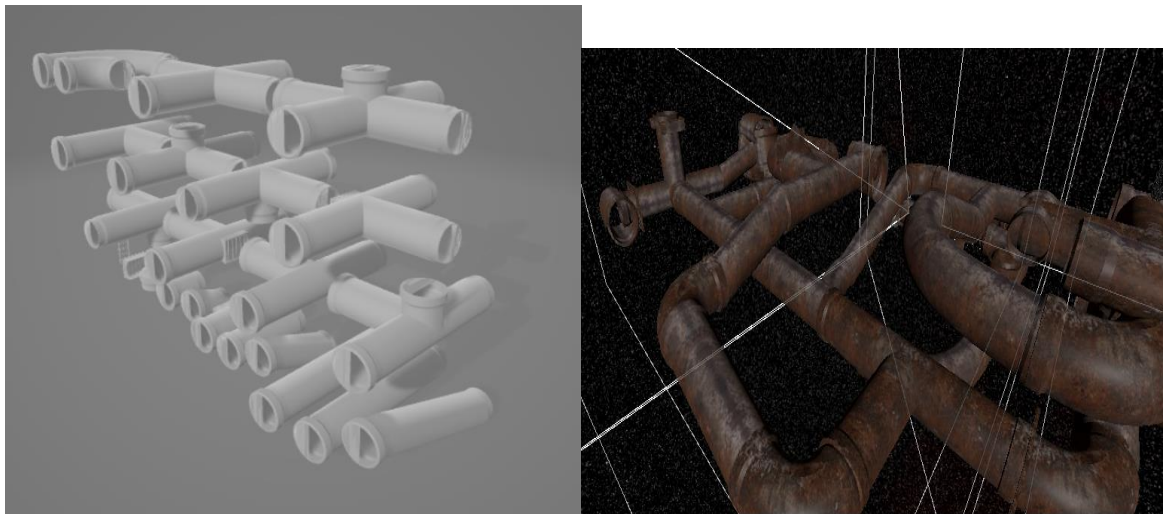
I will be using mostly my own assets, with the exception of a player controller and potentially some buildings or puzzles. This is because I will not be using any textures so I will not be using any other sources for this, therefore greatly limiting the need for external assets. I will also be using audio assets for different features, such as button pressing and losing a life, as I am not confident that I can create high quality assets for audio. Other than this, I do not plan to use any assets. There are some assets like for map design and some features in the game that I will use and these are displayed below:

Keypad



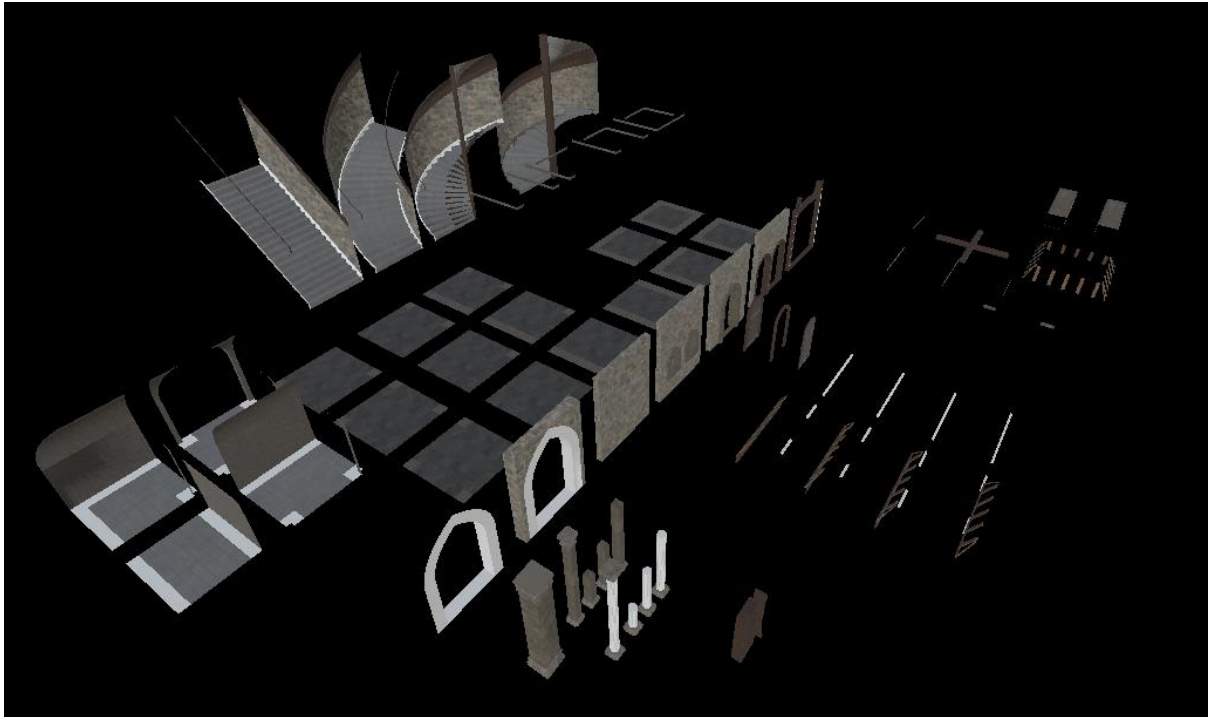
The keypad is an important asset in my game as it plays an important role in ensuring that the player completes the puzzles correctly in order to progress forward, meaning that they are fully experiencing the game.

Sewers



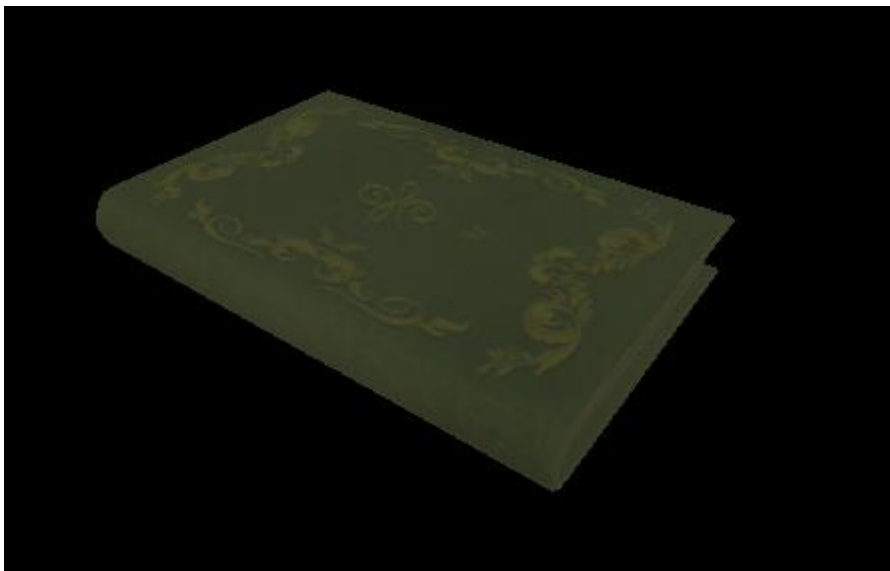
The sewers are another vital gameplay asset that I have used as they are the base world in one of my levels. Without them, I would have been unable to complete my game and the player would be unable to fully experience the world that I eventually developed in one of my levels.

Modular Dungeon



This dungeon world building asset was essential to making my second level, and is where most of the game takes place. Without this, I would have struggled to make my second level and the game would be at a lower standard/quality than it is currently. I consider this, alongside the sewer models, some of the most valuable assets in this game.

Puzzle Book



This green book plays a vital role in the second level, being the puzzle book that the players have to collect in order to obtain the key, which enables them to complete the level. Without this specific book model, I believe that it would make the game much harder and make development significantly longer due to how long it could take to make a book like this.

Key and Padlock



The key and padlock assets that I used are very important in my game. They are used repeatedly throughout and become very easily identifiable. It helps to teach the player that a key is required for this part of the game and teaches them to find said key through exploration. This is an essential asset in my game and therefore plays an important role through the functionality of locking and unlocking doors.

Gameplay features

There are many gameplay features within my game which will be covered below.

Jump

The character will be able to jump around which will help to solve the puzzles in certain scenarios. This will be executed with a spacebar.

Movement

The character will be able to move around with the WASD keys on the keyboard and this will allow them to play the game.

Interact

The player will be able to interact with the interactable game objects by looking at the with the mouse key and pressing the interact key, which is currently E.

Score

The player is able to gain score by completing the puzzles in the game and progressing levels.

Lives

The player will have a limited amount of lives meaning that they have to complete the puzzle correctly within limited time and lives, giving them a chance to lose.

Collision detection

My game will also feature collision detection to prevent the player from walking through walls and enabling them to interact with game objects within the match.

Keypad

There will be a puzzle featuring a keypad which the player will be able interact with. It will function by validating input that the player inputs and checking if it is right or wrong.

Book Puzzle

There will be 4 books that the player has to collect and interact with a bookshelf in order to retrieve a key for a door.

Random Passcode Puzzle

A random passcode is generated and then the numbers are individually associated with a symbol. The player will have to find these numbers and the symbol and then find where the symbols are placed in the correct order and match the numbers to the symbol, providing them with the passcode for the keypad.

Timer

A timer will run and if the player does not complete the level in time, they die and restart.

External feedback and review

×

1. I am highlighting the intractable objects as a red colour, what do you think of this?

4 Responses

ID ↑	Name	Responses
1	Tamas Tokics	Amazing
2	Ahmed Ahmed	Average
3	Luka Radosavljevic	Better
4	Milad Karimi	Good

▲

2. If you disagree, how would you do it?

4 Responses

ID ↑	Name	Responses
1	Tamas Tokics	n/a
2	Ahmed Ahmed	N/A
3	Luka Radosavljevic	N/A
4	Milad Karimi	None

3. How would you rate the explanation of the game through the storyboard

4 Responses

ID ↑	Name	Responses
1	Tamas Tokics	5
2	Ahmed Ahmed	4
3	Luka Radosavljevic	5
4	Milad Karimi	4

4. Do you prefer more detailed or more vague designing for this game?

4 Responses

ID ↑	Name	Responses
1	Tamas Tokics	In between
2	Ahmed Ahmed	In between
3	Luka Radosavljevic	Detailed
4	Milad Karimi	Vague

5. Explain your answer

4 Responses

ID ↑	Name	Responses
1	Tamas Tokics	I like a good balance for both
2	Ahmed Ahmed	N/A
3	Luka Radosavljevic	I prefer games that are more detailed
4	Milad Karimi	I like vague designing

6. I have intentionally made my game simplistic in terms of textures and audio/sound design, do you think this is a positive or a negative?

4 Responses

ID ↑	Name	Responses
1	Tamas Tokics	Great
2	Ahmed Ahmed	Average
3	Luka Radosavljevic	Good
4	Milad Karimi	Good

7. Explain your answer

4 Responses

ID ↑	Name	Responses
1	Tamas Tokics	It is great as more people could play it
2	Ahmed Ahmed	It is not simple
3	Luka Radosavljevic	simple creations nowadays get some popularity for some reason by how most popular games aren't doing so well
4	Milad Karimi	Positive because simple is better when it comes to users trying to focus on the game

8. What do you think of my design overall

4 Responses

ID ↑	Name	Responses				
		Introduction	Storyboard	Pseudocode	Visual styles	Game play features
1	Tamas Tokics	Amazing	Amazing	Amazing	Amazing	Amazing
2	Ahmed Ahmed	Good	Good	Good	Good	Good
3	Luka Radosavljevic	Better	Better	Amazing	Better	Amazing
4	Milad Karimi	Better	Good	Good	Better	Good



Expansion of feedback

1. I was told that highlighting my interactable objects is a good game design choice as it helps to make the game more enjoyable and easier to play. It provides an easier way to play the game where the interactable objects are highlighted and helps to make it easier.
2. No one disagreed with my design choice and therefore there this means that there were no issues with this choice. Furthermore, this reinforces the idea I had with a red highlight for my interactable objects.
3. The rating of my storyboard was high for all members that reviewed my storyboard. This means that the quality of the storyboard was good enough to be understood, and it effectively displayed the way that the game will be showcased at least partially.
4. I asked here for the users to express how they found the story board in terms of whether or not they preferred a vague or detailed design and the answer was not clear. I asked them to elaborate on their opinions in the future.
5. The answers are unclear as they are not that detailed in their responses however I will elaborate. The balance for both is a good option as it ensures that both aspects of the possible options are chosen between detailed and vague. It allows for the design to be modified in the future and elaborated on. Preferring a more detailed design also works well, as it ensures that the game is more closely and intricately designed, which could lead to a more successful completion. Finally, a more vague design might be appreciated as it allows more creativity to be brought into the game and therefore less details restraining the games possibility. Additionally, it will also leave less that is known about for the members who reviewed it, and therefore increases their enjoyment later.
6. The responses regarding the simplicity of the game in terms of audio and visuals show that this is a more positive change than a negative as this allows the game to

be developed while focusing on the components that matter. Furthermore, it ensures that the game will be enjoyable through the simplicity of it.

7. As elaborated on in the previous section, the simplicity of the game helps to increase the quality of it in a number of ways. For example, simplicity is more popular these days and therefore this aspect of the game helps to promote this. Additionally, it is stated that the simplicity will help to allow people to focus on the game and enjoy the experience, as it will be easier to understand and straighter forward. Finally, while someone said that it is not simple, I do not think it is possible to make the game any simpler in terms of textures and audio due to the fact that it would be using just red to highlight the interactable objects and a blank colour for the rest of it.
8. Finally, I asked for an overall opinion on my game and what the members thought of it, and they all responded positively. I believe this is because of the maintained quality and efficiency in my design that helps to get the general idea of my design through to the different members, which therefore increases the overall design standards, leading to the positive responses.

Summary of feedback

To summarise the feedback that I had received, I was told to keep a balance of the different sections I had mentioned, such as the simplicity of the game and the complexity of the textures. This means that the primary focus of my game will be to maintain a solid balance between all of the features and design elements. Furthermore, I will continually refer back to the feedback such as the overall opinions in my design to ensure that I get through all of the features and parts of the game and the final game will be a successful continuation of what I have done.

Timeline/Production Schedule

	21/11/2022	28/11/2022	05/12/2022	12/12/2022	19/12/2022	26/12/2022	02/01/2023	09/01/2023	16/01/2023	23/01/2023	30/01/2023	06/02/2023
Plan the design of the game										KEY :	Completed	
Design the game											In progress	
Documentation of the design											Not started	
Decide the gameplay features												
Research similar games and tutorials												
Set a date for MVP												
Begin programming the game												
Optimise the game												
Get feedback and review												
Finish the game with optimisation												
Finish documentation												
Release game												

Justification

I believe that my activity diagram and storyboard are both justified in their design as they work together to help present the idea of the game will run and the system behind it. Alone they might not make the most sense, however when they are combined it is undeniable that you can understand the system behind gaining score and losing lives, alongside progressing levels and advancing through the game.

Furthermore, the activity diagram goes into depth about the conditions needed to win the game and lose respectively, being a score requirement to win and a loss of all the lives in order to lose, and then displays what should happen in the scenario that these conditions are fulfilled. Similar to most of my design, the details are vague in order to leave room for the changes that might occur during development later down the line in order for my game to be malleable and modified without too much worry on a rigid design that does not allow for any flexibility later on. This focus on flexibility is a key idea in my design and will continue to remain prevalent even later on.

My pseudocode is also limited currently as it is not currently clear how the puzzles and programming behind my game will evolve and will be elaborated on in the future. Currently the pseudocode for how a door might work inside of the game and the way that the character registers what they are looking at is developed. They are both important features that require development in order to functionally test and play the game during development, something that is needed for development.

My gameplay features are each important as they elaborate on the different features that will be included, such as movement, and how different puzzle features, that have been planned out right now, will be working in the future. It also remains vague to allow for development in the future while programming is taking place and the vision for the game becomes clearer. Currently the mention of the two different puzzle types are important as they allow for puzzles to be planned around these types, which can help in speeding up programming and planning later on in the future.

Evaluation of design against client requirements

In order to effectively evaluate the design, the requirements must be listed. These are;

- Levels, such as a first and second level
- Lives, introducing an element of risk for the character so they do not continually survive
- Collision detection, to prevent the player falling through the floors
- Score, to keep track of how the player is doing
- Adding Score, enables the player to have a better score
- Deducting Lives, to enhance the lives feature of the game
- 8+ Age, to make sure that the game is suitable for most ages

Based on these requirements, I believe that I have made an effective and efficient design that successfully meets all of the client requirements and does not make these features seem unnatural or artificially added.

I included multiple levels by creating multiple puzzle levels, something that is common in games. I included lives that decrease based on time-based puzzle solving that can be correct or incorrect which creates a punishment system for the player, encouraging them to get the correct answer, something that is also common in games. The score feature of my game rewards players for successfully completing the puzzles within the time limit and also helps


to progress them to the next level in the game, while also allowing them to complete the game if they get all the score needed to win.

Adding score and deducting lives comes through the implementation of the life and score system respectively and is an important feature of these systems, while the collision is always present in the game as it stops the player from walking through walls or clipping through the floor. Furthermore, it enables the player to interact with the game.

As a result of all of this, I believe that I have made an effective design that has met all the client requirements and exceeded them, successfully completing the design aspect of this game.

Test plan

Feature	Test results	Test evidence
WASD Movement	It works, moves the player around.	See submitted video
Pick up objects	It works, the player is able to pick up objects like books and flashlights.	See submitted video
Timer counts down	It works, the timer ticks down every second.	See submitted video & Picture
Computer is able to run the game without a graphics card.	It works, the video submitted is completed using integrated graphics, ones that come with a CPU.	See submitted video
Game can run on more than one desktop OS	It works, see relevant photo of game running on Mac, and Windows.	See submitted video on Windows, pictures of Mac. 
Jumping	It works, the player is able to jump	See submitted video
The flashlight turns on and off	It works, the flashlight enables and disables	See submitted video
It is able to transition level	It works, the player finishes level one and progresses to the next level	See submitted video
The game can pause	It works, the level pauses and the player cannot move	See submitted video
The score increases	It works, when the player progresses level, the score	See submitted video

	goes up and resets when the player dies/restarts	
The player dies/loses life	It works, the player loses a life when the timer runs out	

Optimisation

Within my game there were a few things that had to be optimised in order to be an enjoyable and satisfactory experience. Initially the first issue that I ran into was the fact that there were performance issues within the second level. I believe this was due to the large amount of polygons in the opening of the level that came from the grass on the ground. Loading and rendering all of these would be very taxing for the computer, and if it was not equipped with an adequate graphics card, then it would be largely unplayable for most people. In order to resolve this issue, I made a change to the second level, where rather than having multiple sets of the grass, which loaded in many polygons for the computer loaded, I simply stretched two pieces of grass across the whole introduction to the level in order to resolve this. Although this would result in the grass looking lower quality compared to before, this is an important choice in order to resolve the issue of quality and makes the game much more playable.

Once making these changes, upon playtesting on a device that uses integrated graphics, that is graphics that come along with a CPU and not a discrete graphics card, and I found that the second level was ran much smoother compared to previously. Additionally, I changed the timer, score, and level layout in order to ensure that it scaled well with the screen size and display. This all helped to optimise the game and make it much more playable and enjoyable, all while fulfilling the criteria that was assigned when making the game.

Evaluation of game against the requirements

User requirements	Met/not met?	How have I met them?	How well I have met them?
Levels	This has been met.	I introduced levels into my game by making more than one puzzle level for the player to play. If the first menu is included, it makes a total of 3 levels, or scenes, that the player gets to experience. The first scene is the menu, and then the sewers level, and	I believe that I have adequately met this user requirement. I have introduced two levels that the player has to play through in order to complete the game. Furthermore, they have to be played in sequential order, starting from the main menu, then to

		finally the dungeon/field level that the player plays through.	the sewer level, and then finally the dungeon level which allows the player to finish the game. Introducing these two levels that have many variables in them ensures that I have met the user requirements for levels.
Lives	This has been met.	I have introduced a life system into the game, where if the player runs out of time, they lose a life and then they die. This introduces the idea that the player has to complete the level under time pressure and helps to add complexity and variation to the game, rather than allowing the player to complete it at their own pace.	I believe that I have adequately met this requirement as I have introduced a life system. If the player loses their life they have to restart the level. Furthermore, with the randomness of the game, being that the code is randomised every time for the first and second level, alongside the random placement of the books on the second level, it makes the lives system much more interesting and punishing if the player dies, given that everything will change in the level. I believe that due to this, I have effectively met this user requirement and gone above and beyond for this.
Collision detection	This has been met.	Collision detection is all throughout both level as it prevents the player from falling through the map and being unable to play the game. Without collision detection, the game would be literally unplayable and therefore I have definitely met this.	As previously mentioned, collision detection take place all throughout my level. It is through this that the player is able to walk around the world and interact with different environments. Furthermore, the way the player interacts with the environment in my world, which works through raycasting and detecting what the player is looking at, which is completed through checking the collider of the object, and therefore also features collision detections. Due to this, I believe that I have effectively completed and met this user requirement, and due to using collision detection with raycasting I would evaluate my usage of collision detection as very effective and above the normal standard.
Score	This has been met.	I have introduced a score system that introduces score every time that the player increases in level. This means that when they start the game, they will have 0 score, and then when they progress to the next level, they gain	I believe that I have effectively met the requirement of having score be apart of the game. I introduced a score system that rewards the player for progressing into the next level, and reverts back to 0 when playing the first level to ensure

		another score, meaning that they have 1 score total. This introduces a score system that rewards the player for progressing to the next level.	that they cannot raise their score to higher than is intended. This has led to an efficient score system being introduced into my game. Due to this, I believe that I have met this user requirement very well.
Adding Score	This has been met.	As mentioned above when discussing the initial score system, adding score occurs when the player progresses onto the next level and resets if the player restarts or dies, and therefore keeps track of the score effectively.	Similarly to above, I believe that I have effectively met the user requirements of adding score by progressing to the next level. Furthermore, it also resets the score when restarting, and therefore due to this, alongside the aforementioned information about this, I believe that I have met the user requirements for this efficiently and effectively, especially when considering my game being a 3D puzzle game.
Deducting Lives	This has been met.	As discussed when talking about the lives earlier, I have added a deducting life system into the game that works through reducing the players lives when they run out of time, which is initially set to be 15 minutes to complete a single level. Once the time runs out, the player loses their life, and they have to restart.	I believe that I have met the user requirement for this effectively. This is through the deduction of lives that I have mentioned. The introduction of the loss of lives, combined with the timer aspect of the game, makes the game more challenging to play and more rewarding to complete, achieving the outcome that I was looking for with the game and therefore I believe completing this user requirement at a high level.
8+ Minimum Age	This has been met.	I have met this requirement because I have not added anything violent or remotely not age appropriate for an 8+ minimum game. Although the game is dark and makes use of a flashlight, there are no horror aspects, such as jump scares or eerie audios that might scare the player, and therefore, although a little dark, this meets the requirement allowing an 8-year-old to play the game.	I believe that I have met this user requirement effectively and efficiently within the limits of an 8+ game. This is because I have included all kinds of puzzles and different game features that make the game enjoyable and playable, such as a flashlight, and although I kept the game on the darker side, in terms of brightness, this is the only aspect of the game that could be considered creepy or eerie, and therefore allows an 8-year-old child to be able to play the game without worrying about it. This means that I have made the game complete, alongside meeting the user requirements very well.