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

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 15+ 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

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.

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.

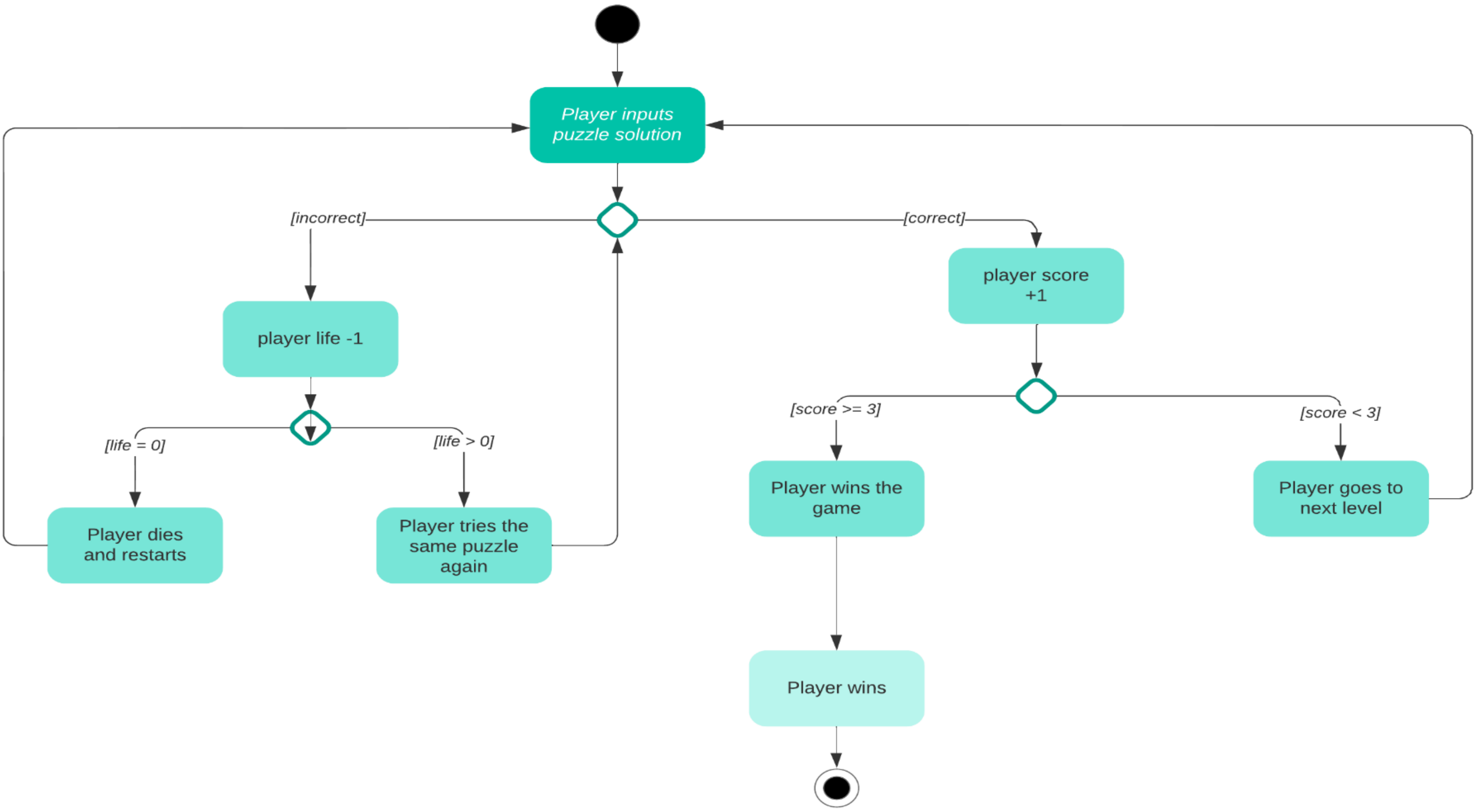
```
Function Interact Action
    Object (I.e. box, door, light switch)

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



Storyboard

You wake up inside of an abandoned house with no escape

You have to look around for the puzzles to solve and find your way out

Once you wake up, you escape into another house, the next level and repeat

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.

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.

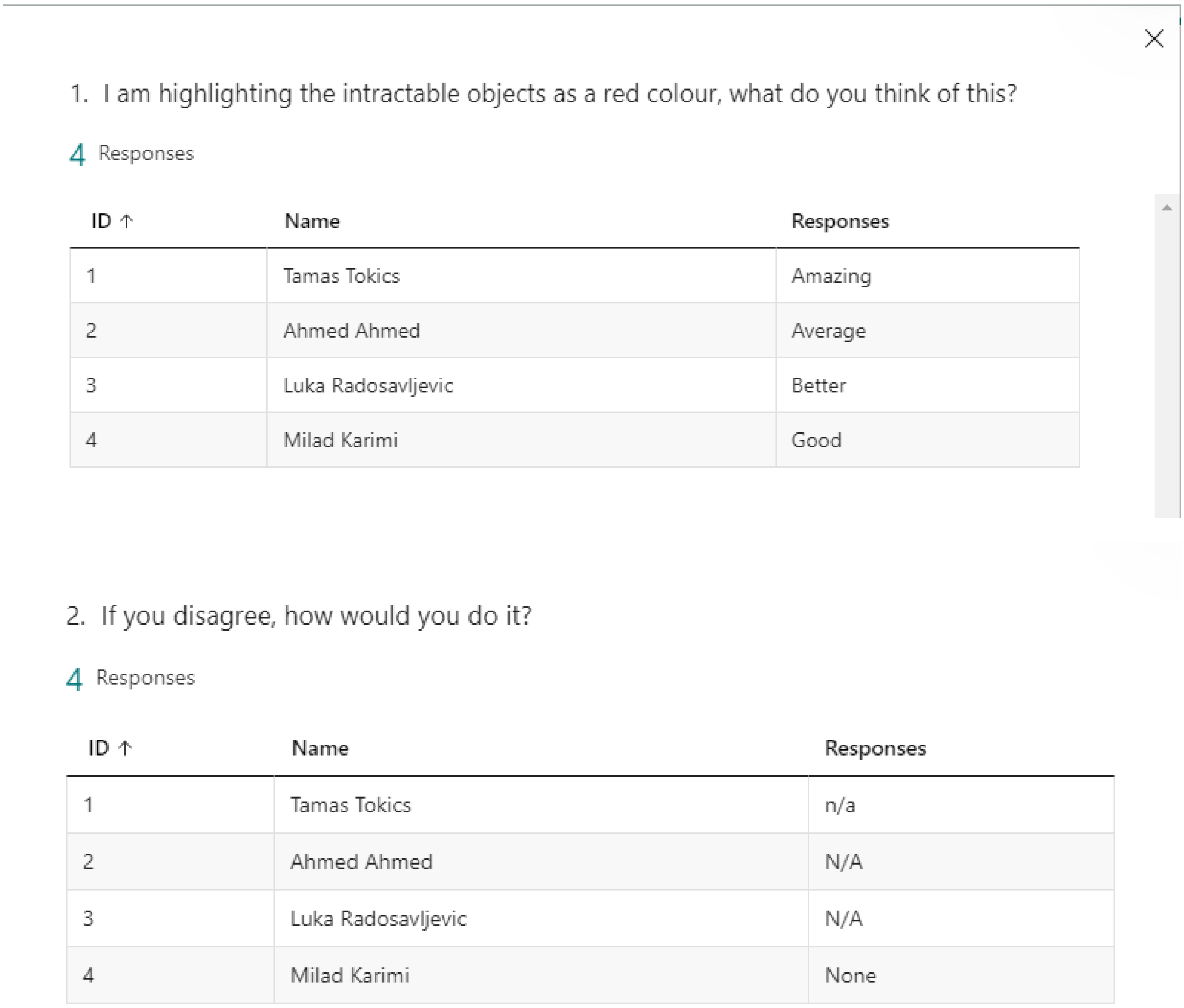
Keypad

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.

Rotational puzzle

There will be a puzzle in my game that the player will be able to solve by interacting with the object and turning it until it is completed and validating.

External feedback and review



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

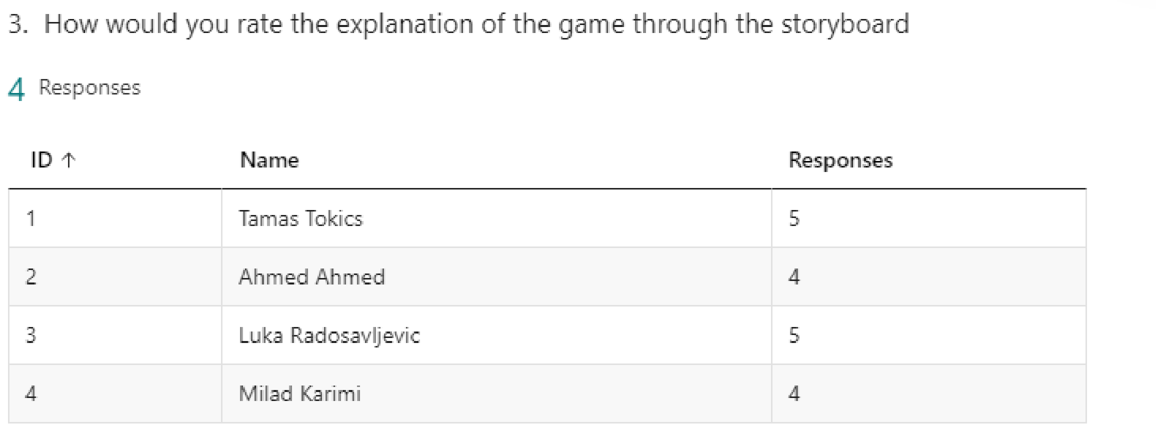
4 Responses

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

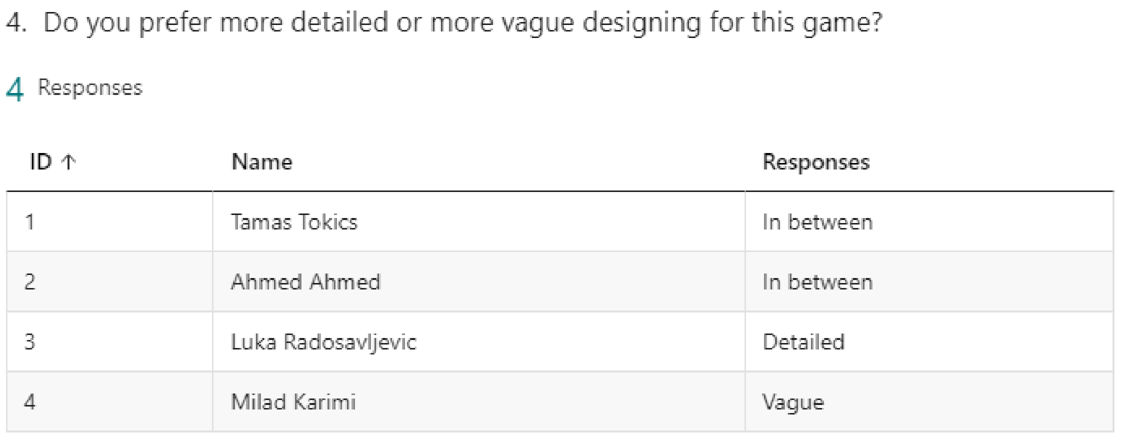
ID ↑	Name	Responses
1	Tamas Tokics	n/a
2	Ahmed Ahmed	N/A
3	Luka Radosavljevic	N/A
4	Milad Karimi	None



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

4 Responses

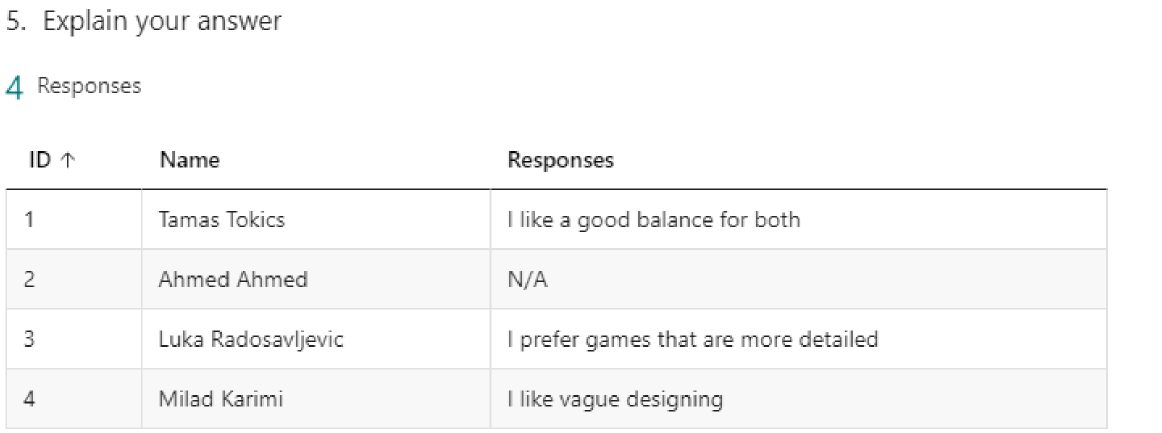
ID ↑	Name	Responses
1	Tamas Tokics	5
2	Ahmed Ahmed	4
3	Luka Radosavljevic	5
4	Milad Karimi	4



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

4 Responses

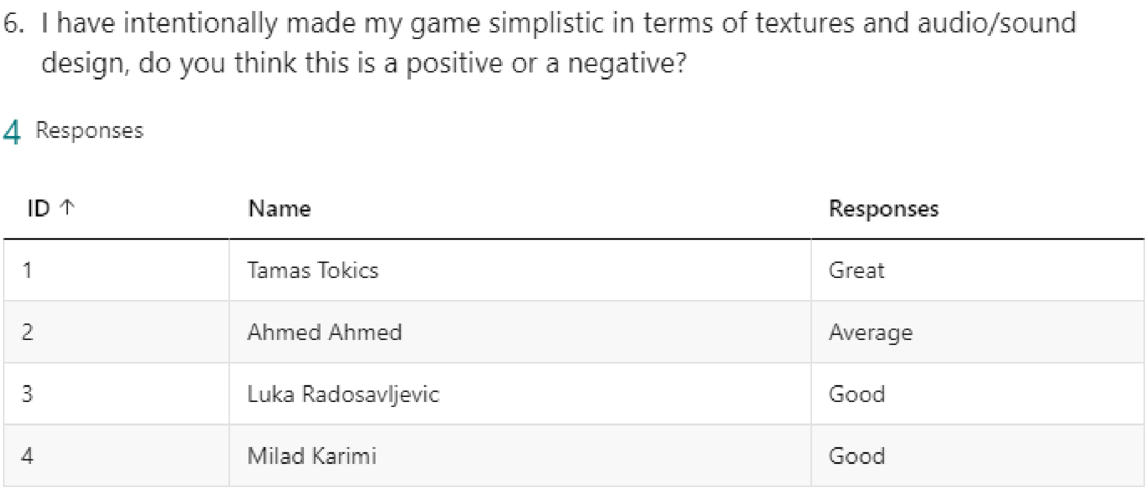
ID ↑	Name	Responses
1	Tamas Tokics	In between
2	Ahmed Ahmed	In between
3	Luka Radosavljevic	Detailed
4	Milad Karimi	Vague



5. Explain your answer

4 Responses

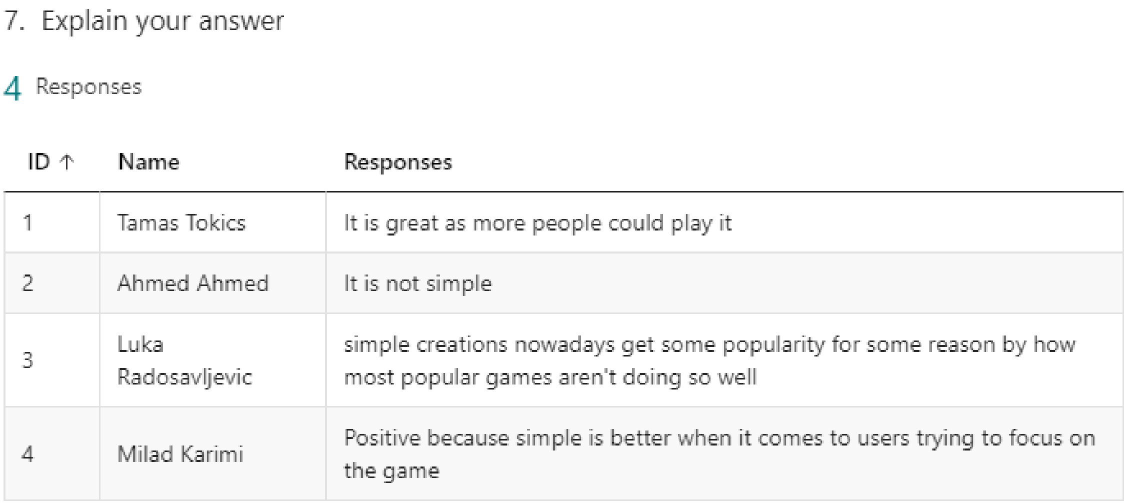
ID ↑	Name	Responses
1	Tamas Tokics	I like a good balance for both
2	Ahmed Ahmed	N/A
3	Luka Radosavljevic	I prefer games that are more detailed
4	Milad Karimi	I like vague designing



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

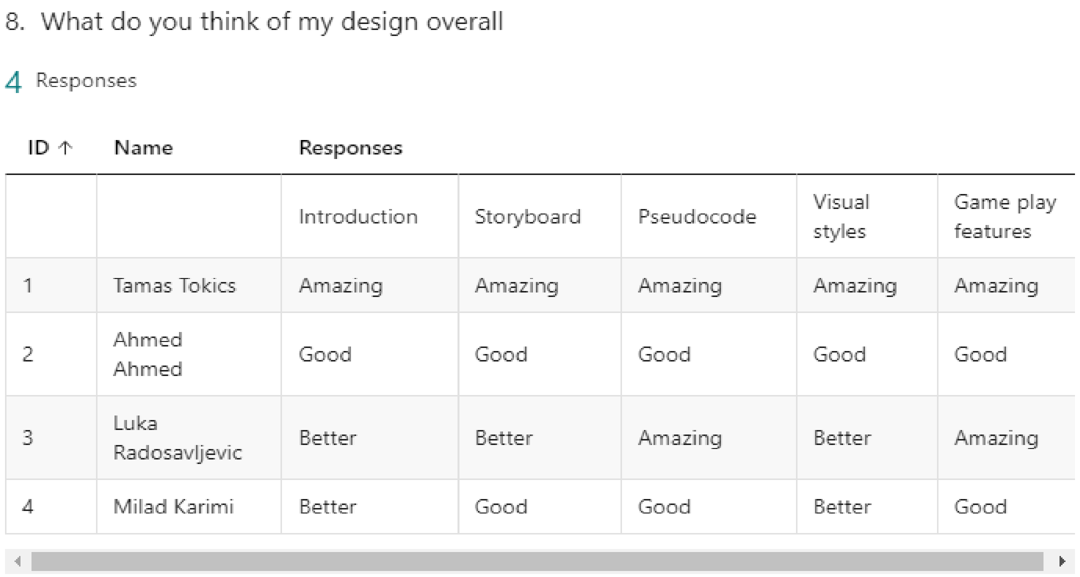
ID ↑	Name	Responses
1	Tamas Tokics	Great
2	Ahmed Ahmed	Average
3	Luka Radosavljevic	Good
4	Milad Karimi	Good



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

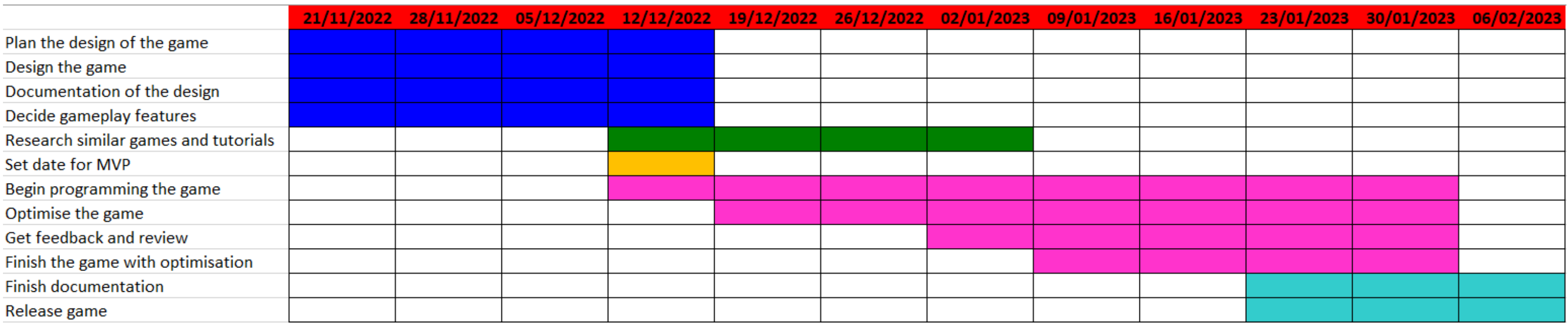


8. What do you think of my design overall

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

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	🟦	🟦	🟦	🟦								
Design the game	🟦	🟦	🟦	🟦								
Documentation of the design	🟦	🟦	🟦	🟦								
Decide gameplay features	🟦	🟦	🟦	🟦								
Research similar games and tutorials				🟩	🟩	🟩	🟩					
Set 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
- Lives
- Collision detection
- Score
- Adding Score
- Deducting Lives

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.