Adapting Educational Content to Maximise Reuse Across Knowledge Groups via Interactive Experiences

Progress Report

Teodora Militaru
2055695

1 Introduction

In recent years, computer games have become a popular means of delivering educational content, due to them facilitating a more interactive learning environment, which promotes blending learning with entertainment [1]. However, this type of learning tool is usually only targeted to one knowledge group (mostly beginners and younger students), being unusable for any other category of players and authoring content to suit such groups is an expensive and time-consuming process. This project aims to explore ways to adapt educational content such that it is suitable for all knowledge groups through a proof of concept escape room-style game.

2 Background

While one could argue that games are not an appropiate learning environment for higher education or for complex topics, such as programming, there have been many studies that disprove this statement [2]. In particular, the escape room-style games are regarded as an interactive experience targetted at adults and have been proven to be successful in teaching certain skills, specifically in STEM subjects.

Since the concept of adapting content for diffferent knowledge groups is quite new when it comes to educational games, the focus of this research had to be shifted towards any genre of games. There were two types of difficulty levels identified: static (where the player chooses the level they want to play at) and dynamic (where the level is adjusted based on the player's experience in the game) [3]. The most suitable for educational games would be a combination of the two, where the player is able to choose a level at the begining, but the game also adapts based on experience. In this project, this will be reflected by having the chance of appearance of easier levels decrease as the player engages more with the game.

Another mechanism of adapting the level to fit different difficulties is changing the environment [4]. In the context of escape room-style games, this could be achieved by either changing the placement of certain objects needed to complete the puzzles or the placement (or existence) of hints. In this project, this will be reflected by randomising the location of all objects needed for puzzles every time the player encounters a certain room. Therefore, the game becomes more challenging even though the content that is used does not change.

Because the concept of the game revolves around randomising different aspects, whether those are objects, rooms or puzzles, an essential part of the research step was reading literature regarding randomisation algorithms. Thus, two key resources were found, each catering to a different type of randomisation.

Firstly, Johnson [18] explores alternatives for randomly placing objects around the rooms, without making the puzzles unsolvable. All of the approaches described view the world generated by the game as a graph, where the nodes are the different locations and the edges represent different rules which need to be respected to reach them. There are three algorithms that can be used to distribute the objects around the rooms: random fill, forward fill and assumed fill. While random fill simply generates distributions until they do not cause conflicts, forward fill considers the rules on the edges before deciding the location of an object. The algorithm starts by identifying all the reachable locations from the beginning of the game, then choses a random object and places it in one of those locations, after which the list expands. On the other hand, the assumed fill algorithm assumes all the locations are reachable, therefore the player owns all objects. A random object is then chosen to be removed from the inventory, the reachable locations are recalculated and the object is randomly placed in one of them. For this project, the most suitable algorithm would be the forward fill algorithm as it perfectly balances time complexity with ease of implementation.

Secondly, Grasas [19] describes different randomisation algorithms that make use of some type of bias and their benefits and applications. The most common issue with a lot of randomisation algorithms is that, when selecting a candidate with a uniform probability distribution, the ranking resulting from the bias function is lost. However, if we were to just consider the higher ranking candidates, there would be no randomness involved. The biased randomised procedures (BRP) presented in this resource combine exploitation (through favouring the higher ranking options) and exploration (through introducing a weighted randomness factor). Therefore, a few priority functions could be assigned to the candidates and a subset could be created with the higher ranking ones. In order to introduce randomness, a few other candidates would be picked randomly from the entire set then added to the subset. In the context of this project, the priority functions would include: how suitable the difficulty of the puzzle is for the player, how often the puzzle has been encountered, how much experience the player has gained.

3 Technical Progress

Considering the issues this project is aiming to solve, designing the game revolves around two very important aspects: an interesting storyline to keep the player interested and puzzles promoting reuse of content and allowing the player to learn through an interactive experience.

3.1 Storyline

In the context of educational games, it has been proven that storytelling is an important aspect, as it provides a mechanism to motivate the player as well as make the information more memorable [5]. While it may seem that making the story and the gaming experience the focus of the development process may hurt the educational aspect, it has been proven not to be the case, as the benefits of using an interactive environment for the learning process are decreased otherwise [6].

The story follows a scientist who lost his daughter and created a system to help him relive their moments together. However, a few things go wrong and it ends up erasing all of his memories and generating a glitched world from where he must escape. By fixing the environment and completing different quests given by some robots, the player unlocks "memory fragments" which help reveal the story step by step. This fragments can be viewed by using the TV in the player's room, where he respawns after every escape attempt.

By using this memory fragment system, the player is provided with a goal (unlock all the memory fragments to uncover the story) as well as a checkpoint to test their knowledge (since the puzzles from the robots will be more difficult and have no hints). However, in order to not put pressure on the learning process, these puzzles will be optional and a game run can be completed without them.

3.2 Puzzles

There have been various studies focusing on the compatibility of escape room-style games and educational content, proving that it is one of the most successful integration of learning objectives within interactive experiences due to the ability to design puzzles around the targeted topics [7]. For this project, there are two types of puzzles the player will encounter: classic escape room-style puzzles and educational puzzles. The latter revolve around four topics in Computer Science: object oriented programming, logic and verification, database systems and web development.

The proccess of designing the puzzles begins with creating a few variations for each targeted topic.

Initially, the plan was to have the same number of puzzles for each of the categories, however, during development I realised their potential to generate interesting puzzles that could seamlessly blend with the content of the game is different. Therefore, there are going to be more variations for topics such as object oriented programming than topics such as web development.

In order to promote reuse of content, each of the variations has a basic layout where there are a number of variables that can be customised depending on the selected difficulty. Some proposed puzzle variations for the chosen topics are as follows:

Object Oriented Programming

- Creating a logic diagram based on some given code
- Changing some values in a given code to modify the environment (for example, turning on the lights in a room)
- Find errors in a given code
- Perform vector operations to get the password to a lock
- Calculate the output of some code
- Find the input for given output
- Create an object to use in the game
- Customize an object's properties (they will be reflected in the game)
- Import certain libraries which will spawn as books where the player can find information about different functions

Logic and Verification

- Choose an equivalent logic expression to the given one
- Complete a truth table with missing information (the missing values can then be used as a password to unlock a door)
- Convert from natural language to logic expression
- Select or create the correct parse tree for a given expression
- Use laws of boolean algebra to simplify formulas (the simplified formula can be a password)
- Compute DNF and CNF of a formula

Database Systems

- Retrieve information from a database to solve a puzzle (for example, the player has to put pictures in an yearbook and they have to retrieve the student names from a database based on the year)
- Insert an entry into a database (for example, they could add an item on a shopping list and the item would spawn in the game)
- Correct errors in a database schema (for example the primary key being NULL)
- Compute result for a given SQL statement
- Perform Join Operations

- Grant SQL priviledges to get access to data

Web Development

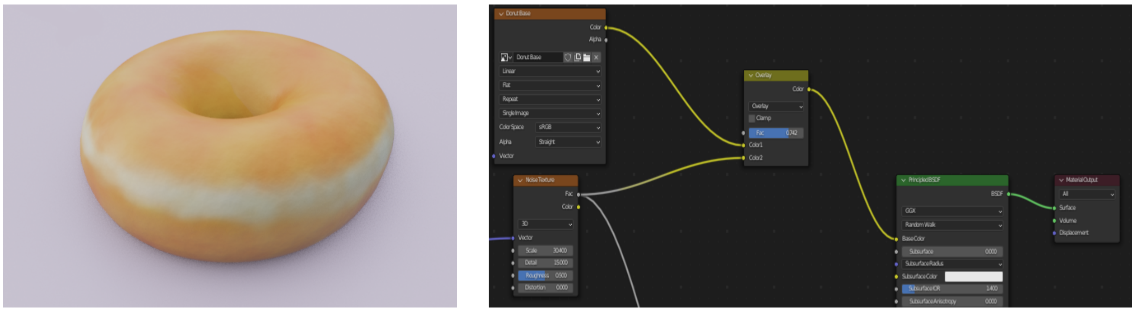
- Correct errors in a given code
- Create a form to get data that can be used in another puzzle
- Change lines of code that will affect the environment (for example, reveal hidden text that was the same colour as the background)

4 Use of Tools

4.1 Blender

The original plan was to use Blender to create all 3D assets needed in the game, however, due to the time limit and the complexity of the system, this proved to not be feasible. Since assets are not the focus of this project, most of them were imported from websites such as Kenney [8] or Unity Store [9], making sure they are free to use for educational purposes. Even though most assets are imported, Blender was still used for the more specific ones needed in the game such as the robots giving memory fragments. While following various tutorials, a few notable features of Blender were discovered, which allow for creation of realistic models that do not require much rendering complexity.

For instance, in order to add detail to a texture such that it is not a single colour, noise texture nodes can be used in the shadder [10]. These have various properties that customise the look of the model, such as the level of distortion and the roughness, which can be combined with a base colour, by using an overlay node. An example of this feature can be seen in figure 1, where they were used to make the donut base more realistic.



Donut Base
Linear
Flat
Repeat
Single Image
Color Space sRGB
Alpha Straight
Overlay
Clamp
Fac 0.742
3D
Scale 30.400
Detail 15.000
Roughness 0.500
Distortion 0.000
GGX
Random Walk
Subsurface 0.000
Subsurface Radius
Subsurface Color
Subsurface IOR 1.400
Subsurface Anisotropy 0.000
All

Figure 1: Donut base created in Blender and the corresponding nodes used in the shader

Another important feature are geometry nodes [11] which allow to procedurally alter an object's properties such as colour, geometry and material. They are a very useful tool as they can be used to modify the structure of an object, without affecting the rest of the model. An application of this feature can be seen in figures 2 and 3, where geometry nodes have been used to generate the sprinkles on the donut. There are three different variations of these sprinkles with different randomly generated colours, which is achieved by using a ColorRamp node, as seen in figure 4. If we wanted to add another type of sprinkle, that would be as easy as just creating the model for it and adding it to the group of sprinkle models.

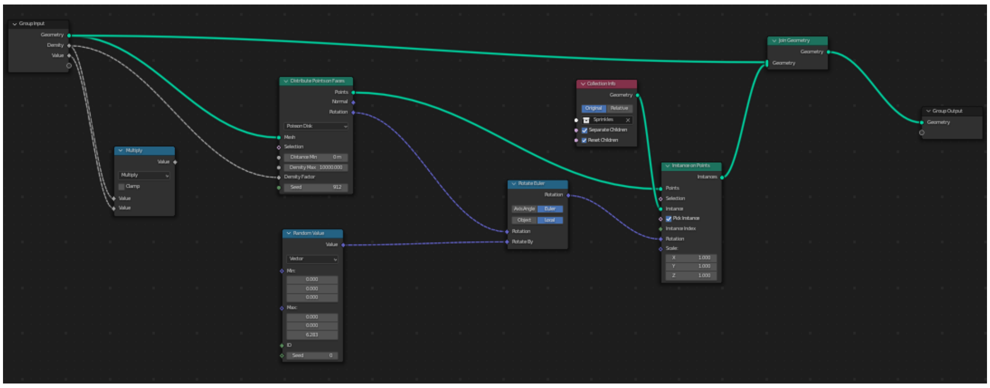


Figure 2: Nodes used to generate the sprinkles

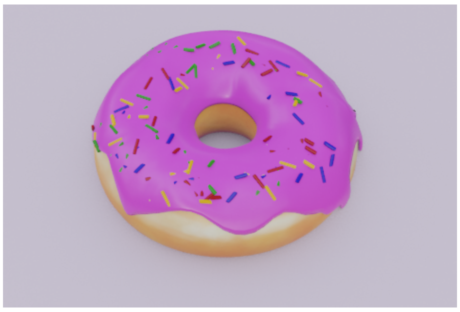


Figure 3: Sprinkles created using geometry nodes

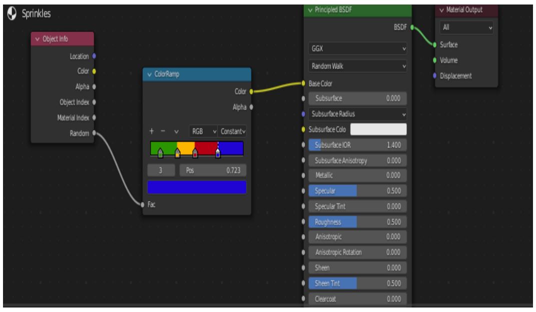


Figure 4: ColorRamp node example

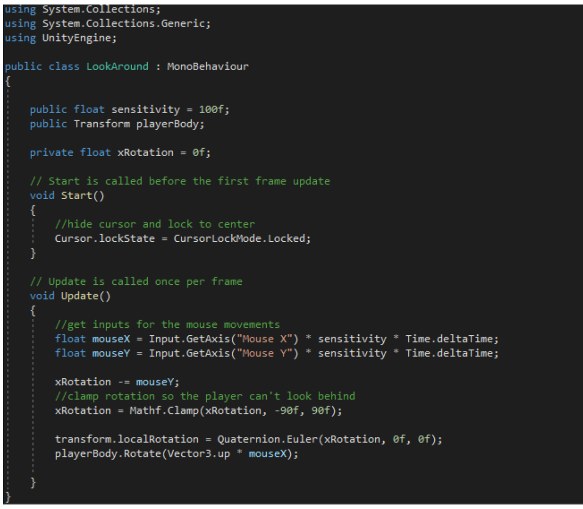
4.2 Unity

The proof of concept game is developped using Unity, a game engine I was slightly unfamiliar with at the beginning of this project. This did not prove to be an issue as there are various tutorials available online, most notably the Create With Code tutorials made by Unity [12]. These tutorials involve making a few small games in order to learn how to implement basic features such as: character controls, physics, text in the game, score functions and so on. The most important aspects of the game implementation stage were the player controls and camera movement and building the rooms.

4.2.1 Player Controls and Camera Movement

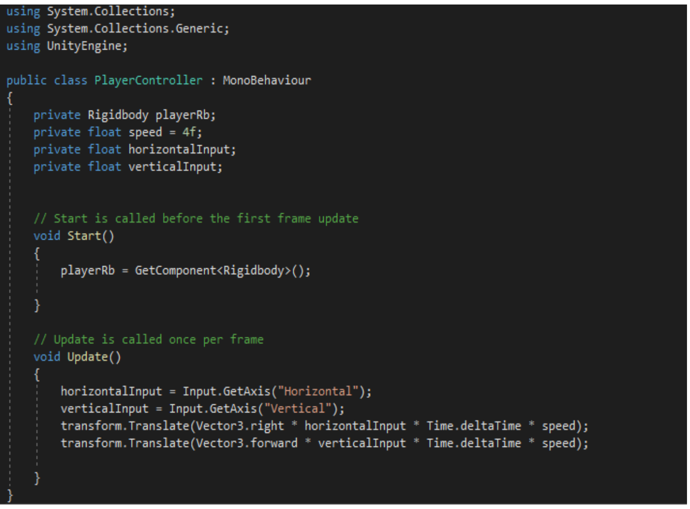
Because this is an escape room type of game, the player should be able to view the world from a first person point of view. This can be achieved in two ways: by using the Rigidbody component or by using a CharacterController component [13]. Both of these implementations have advantages and disatvantages depending on the type of game that is created. Because the focus of this project are puzzles revolving around educational topics, the game does not need complex physics and the character's body interacting with the environment. Therefore, the approach I chose is the second one, using the CharacterController component [14], which allows for movement contstrained by collisions, but not affected by forces.

In order to look around the world, the player controlls the camera with the mouse, which moves on two axis X and Y. An example of the code used to implement this movement in shown in figure 5. In addition, the start function contains code to lock the cursor in place such that the player will not misclick outside the game screen.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class LookAround : MonoBehaviour
{

    public float sensitivity = 100f;
    public Transform playerBody;

    private float xRotation = 0f;

    // Start is called before the first frame update
    void Start()
    {
        //hide cursor and lock to center
        Cursor.lockState = CursorLockMode.Locked;
    }

    // Update is called once per frame
    void Update()
    {
        //get inputs for the mouse movements
        float mouseX = Input.GetAxis("Mouse X") * sensitivity * Time.deltaTime;
        float mouseY = Input.GetAxis("Mouse Y") * sensitivity * Time.deltaTime;

        xRotation -= mouseY;
        //clamp rotation so the player can't look behind
        xRotation = Mathf.Clamp(xRotation, -90f, 90f);

        transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
        playerBody.Rotate(Vector3.up * mouseX);

    }
}
```

Figure 5: Script for camera movement



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    private Rigidbody playerRb;
    private float speed = 4f;
    private float horizontalInput;
    private float verticalInput;


    // Start is called before the first frame update
    void Start()
    {
        playerRb = GetComponent<Rigidbody>();

    }

    // Update is called once per frame
    void Update()
    {
        horizontalInput = Input.GetAxis("Horizontal");
        verticalInput = Input.GetAxis("Vertical");
        transform.Translate(Vector3.right * horizontalInput * Time.deltaTime * speed);
        transform.Translate(Vector3.forward * verticalInput * Time.deltaTime * speed);

    }
}
```

Figure 6: Script for player movement

The player object can move in four directions: forward, backward, left and right. In order to implement this movement, the code shown in figure 6 accesses input from the keyboard keys and multiplies it with a speed variable and Time.deltaTime, which is used to make sure the game functions the same regardless of the frame-rate of the computer [15].

4.2.2 Building Rooms with ProBuilder

ProBuilder is a package in Unity which allows for quick building, editing and texturing of custom geometry and is generally used for level design [16]. The first notable feature of this package is the New Shape button which introduces 12 primitive shapes. By adding a shape, a new game object is created and can be viewed and edited in the Inspector. There are four selection modes for objects: object selection, vertex selection, edge selection and face selection, each having different options available to customise the model.

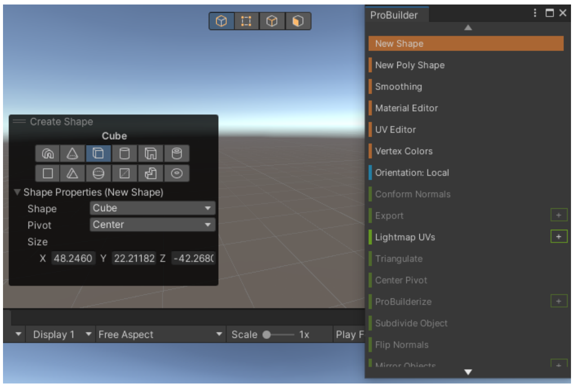
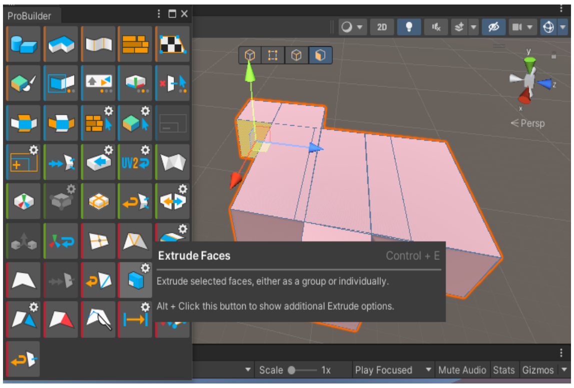


Figure 7: Example of the ProBuilder interface



Extrude selected faces, either as a group or individually.

Alt + Click this button to show additional Extrude options.

Figure 8: Using the extrude tool in face selection

One of the most useful tools for an escape room type of game is flip normals which converts an exterior-modeled shape into an interior [17]. By using this together with the extrude option, I was able to quickly create room shapes that were not just simple cubes (figure 8).

5 Project Management

Overall, the plan set in the project specification was followed with slight modifications being made during development:

- The time allowed for research was extended in order to acomodate for the difficulty of finding any papers on adapting content based on varying difficulty levels
- The create puzzles and storyline step was completed in week 5 instead of week 6 since the puzzles were just researched and designed, their implementation being moved to week 9, after the rooms and randomisation algorithms are fully developped. The decision to implement the puzzles so late was taken in order to be able to add the randomisation algorithms before the progress report deadline as they are the most interesting and complex technical aspect of the game.

Even though these adjustments were made, the project is still on track and the first version of the game will be available by the end of week 10. Figure 9 shows how the development progressed up until this point, with more detailed steps than those mentioned in the project specification.

5.1 Objectives

Throughout the research step, some of the initially set objectives were modified and new ones were added. The list bellow shows all the objectives of the project, with the newly added or modified ones being in bold and the ones that have been completed being checkmarked.

5.1.1 Main Objectives:

1. The player will be able to select a difficulty level from the provided list: beginner, intermediate and advanced ✓
2. The player will be able to move around the world from the perspective of a controllable character ✓
3. It will be possible to change the difficulty at any time ✓
4. After playing for some time, the player will be prompted to switch to a more advanced difficulty ✓
5. Beginner and intermediate level players will be able to find hints around the rooms to aid them in solving the puzzles
6. Intermediate and advanced level players will have a time limit to solve some of the rooms and puzzles
7. The players will be able to see a countdown of how much time they have left
8. Minimalistic sound effects will be used to make the experience more immersive ✓
9. The player will be able to complete a run of the game by escaping from 3 to 6 rooms
10. **The player will be able to obtain memory fragments from completing tasks for some robots**

11. **The robots will pe spawned at random moments in time**
12. **The player will be able to view their memory fragments by using the TV in the main room**
13. Each room will contain 3 to 5 puzzles that require computer science knowledge
14. Each of the puzzles will have elements that can be randomly generated to ensure the chances of encountering the same puzzle twice are as low as possible ✓
15. The rooms will also contain a few basic escape-room style puzzles
16. **The objects containing puzzles which the player can interact with will be highlighted**
17. **The objects which the player is able to pick up will be highlighted** ✓
18. **The player will be able to access an inventory consisting of objects found in the room** ✓
19. **The objects in the inventory can be used to solve some of the puzzles**
20. After a player encounters a certain room, the chance of it reappearing will be lowered ✓
21. **The player will be able to modify the volume of the music and sound effects playing in the background** ✓
22. **The game will have a title screen offering a few options: load game, new game, set difficulty, game options** ✓
23. **The player will be able to load their game from the title screen** ✓
24. **The player will be able to reset their progress from the tilte screen** ✓

5.1.2 Non-functional Objectives:

1. The game will be entertaining
2. People with no previous computer science knowledge will be able to play the game
3. The game will be challenging for players with previous computer science knowledge
4. The storyline will be engaging

5.1.3 Extensions

1. More intricate sound effects could be used to improve the atmosphere of the game
2. New computer science topics could be added
3. The 3D assets could be improved to make the experience more immersive
4. **A glitch effect could be applied to all rooms and, as the story progresses, it could slowly be removed**
5. **The TV could display the last memory fragment earned when the player is in the main room**

6. **The player could have a notebook in which to add screenshots of the different hints in the game**
7. **A challenge mode where the player would have to escape as many rooms as possible in a set time**

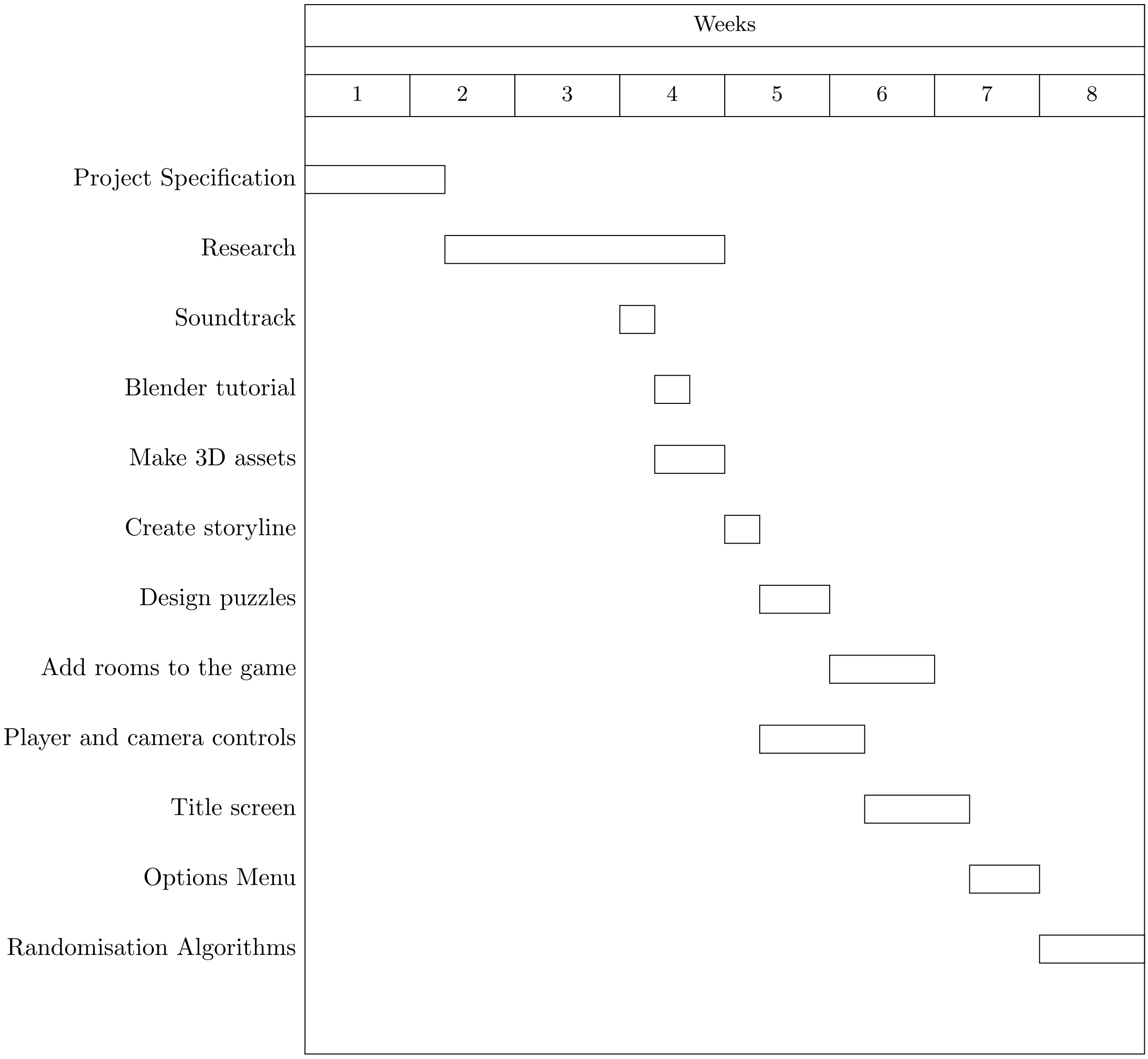


Figure 9: Gantt chart showing progress made up until week 8

5.2 Week 9 and week 10 Schedule

The following two weeks will be dedicated to merging all the aspects previously created, in order to complete the game and adding any missing features. Therefore, the proposed schedule is as follows:

T1 W9	Implement puzzles and their corresponding hints in the game
T1 W9	Add the countdown function for advanced players
T1 W10	Implement the memory fragments system
T1 W10	Create the questionnaire for the testing phase

5.3 Term 2 Schedule

The schedule for the second term heavily relies on the results from the winter break testing phase. Therefore, if the game only requires a few modifications, the schedule will be the following:

T2 W1	Analyse results from the testing phase and write up new objectives or modifications
T2 W1-W2	Implement the new objectives and modifications
T2 W2-W3	Implement notebook for screenshots
T2 W3-W4	Improve 3D assets and add extensions: 4, 5
T2 W4	Start new testing phase and interpret results
T2 W3-W5	Write first half of final report
T2 W5	Implement any new objectives or extensions
T2 W6	Final testing phase
T2 W7	Game is complete
T2 W8-W9	**Project presentation**
T2 W8-W9	Finish writing final report
T3 W1	**Final report**

On the other hand, if after the first testing phase there are a lot of modifications to be made, those will be prioritised instead of the extensions and improving 3D assets.

6 Legal, Social, Ethical and Professional Issues

As explained in the specification, since all the data that will be collected from human participants is limited to questionnaires related strictly to the contents of the project, no ethical issues need to be considered.

References

[1] Stephen Baer. *The Rise Of Educational Games.* Feb. 2018. URL: https://elearningindustry.com/rise-of-educational-games.

[2] Sonsoles López-Pernas et al. “Examining the Use of an Educational Escape Room for Teaching Programming in a Higher Education Setting”. In: 7 (2019), pp. 31723–31737.

[3] Sertaç Ögüt and Barbaros Bostan. “Game challenges and difficulty levels: lessons learned From RPGs”. In: *International Simulation and Gaming ...* (Jan. 2009). URL: https://www.academia.edu/795475/Game_challenges_and_difficulty_levels_lessons_learned_From_RPGs (visited on 11/23/2022).

[4] Glen Berseth et al. “Characterizing and Optimizing Game Level Difficulty”. In: *Proceedings of the 7th International Conference on Motion in Games.* Association for Computing Machinery, 2014, pp. 153–160. URL: https://doi.org/10.1145/2668064.2668100.

[5] Olaf H. Graven and Lachlan MacKinnon. *Using Games Technology to Develop Reusable Virtual Learning Environments-an Embedded Approach.* June 2012. URL: http://0-search-proquest-com.pugwash.lib.warwick.ac.uk/conference-papers-proceedings/using-games-technology-develop-reusable-virtual/docview/1326324789/se-2.

[6] Neil Peirce, Owen Conlan, and Vincent Wade. “Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences”. In: *2008 Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning.* 2008, pp. 28–35.

[7] Sarah Leon Rojas et al. “Natural Europe Educational Games Suite: Using Structured Museum-Data for Creating Mobile Educational Games”. In: *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology.* Association for Computing Machinery, 2014. URL: https://0-doi-org.pugwash.lib.warwick.ac.uk/10.1145/2663806.2663841.

[8] *Kenney • Home.* URL: https://www.kenney.nl (visited on 11/20/2022).

[9] *Unity Asset Store - The Best Assets for Game Making.* URL: https://assetstore.unity.com/?category=3d&free=true&orderBy=1 (visited on 11/20/2022).

[10] *Noise Texture Node — Blender Manual.* URL: https://docs.blender.org/manual/en/latest/render/shader_nodes/textures/noise.html (visited on 11/20/2022).

[11] *Geometry Nodes — Blender Manual.* URL: https://docs.blender.org/manual/en/latest/modeling/geometry_nodes/index.html (visited on 11/21/2022).

[12] *Create with Code.* URL: https://learn.unity.com/course/create-with-code (visited on 11/20/2022).

[13] *FIRST PERSON MOVEMENT in Unity - FPS Controller.* URL: https://www.youtube.com/watch?v=_QajrabyTJc (visited on 11/21/2022).

[14] Unity Technologies. *Unity - Scripting API: CharacterController.* URL: https://docs.unity3d.com/ScriptReference/CharacterController.html (visited on 11/21/2022).

[15] *TIME.DELTATIME EXPLAINED — Unity For Beginners — Unity Tutorial.* URL: https://www.youtube.com/watch?v=8pYq15Lh0x4 (visited on 11/21/2022).

[16] Unity Technologies. *Build Virtual Worlds From Scratch — Unity ProBuilder.* URL: https://unity.com/features/probuilder (visited on 11/21/2022).

[17] *Flip Normals — ProBuilder — 4.1.2.* URL: https://docs.unity3d.com/Packages/com.unity.probuilder@4.1/manual/Object_FlipNormals.html#:~:text=The%20Flip%20Normals%20tool%20flips,shape%20into%20an%20interior%20space. (visited on 11/21/2022).

[18] Caleb H. Johnson et al. “Evaluation of Algorithms for Randomizing Key Item Locations in Game Worlds”. In: *IEEE Access* 9 (2021), pp. 48286–48302.

[19] Alex Grasas et al. “Biased randomization of heuristics using skewed probability distributions: A survey and some applications”. In: *Computers and Industrial Engineering* 110 (2017), pp. 216–228. URL: `https://www.sciencedirect.com/science/article/pii/S036083521730270X`.

Adapting Educational Content to Maximise Reuse Across Knowledge Groups via Interactive Experiences

Specification

Teodora Militaru
2055695

1 Introduction

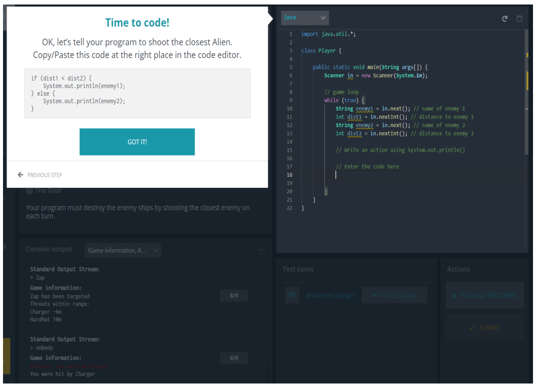
In recent years, the use of technology in education has gained popularity, providing teachers and students with a new means of presenting and absorbing content. One of these resources are educational games, which, through the entertaining and competitive aspects of computer gaming, provide an interactive learning space that promotes analytical thinking.

2 Problem

The main issue with many educational games is that they cater to only one knowledge group, usually beginners. One could argue that players who are already familiar with the content cannot benefit from educational games since there is no more knowledge to be gained. However, these types of games offer an interactive space to not only learn, but practice a certain set of skills in a way which is challenging and entertaining. Therefore, it is important that these types of resources are also available and suitable for more experienced users.

On the other hand, authoring content to be suitable for other categories is difficult, expensive and time consuming, especially in the educational games field. As a result, while games in general might have different difficulty settings, educational games are either made for complete beginners or people who are already confident in their knowledge. For example, the screenshots bellow are taken from two different games (Codin Game [1] and SQL Murder Mystery [2]), each of them targetted to a different knowledge group.

Codin Game



Time to code!

OK, let's tell your program to shoot the closest Alien.
Copy/Paste this code at the right place in the code editor.

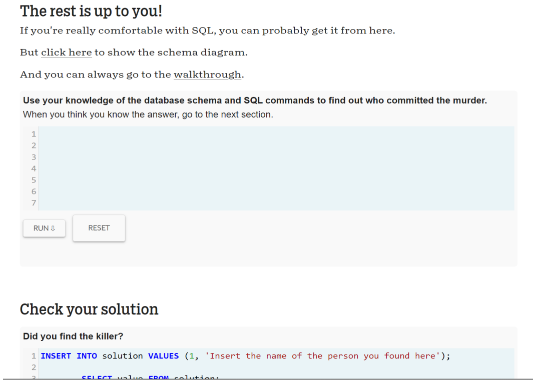
```
if (dist1 < dist2) {
    System.out.println(enemy1);
} else {
    System.out.println(enemy2);
}
```

Your program must destroy the enemy ships by shooting the closest enemy on each turn.

```
import java.util.*;

class Player {

    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);

        // game loop
        while (true) {
            String enemy1 = in.next(); // name of enemy 1
            int dist1 = in.nextInt(); // distance to enemy 1
            String enemy2 = in.next(); // name of enemy 2
            int dist2 = in.nextInt(); // distance to enemy 2

            // Write an action using System.out.println()

            // Enter the code here


        }
    }
}
```

```
Standard Output Stream:
> Zap
Game information:
Zap has been targeted
Threats within range:
Charger -6m
HardHat 10m
                                8/9

Standard Output Stream:
> nobody
Game information:
Failure: You were destroyed
You were hit by Charger
                                9/9
```

SQL Murder Mystery



The rest is up to you!

If you're really comfortable with SQL, you can probably get it from here.

But click here to show the schema diagram.

And you can always go to the walkthrough.

Use your knowledge of the database schema and SQL commands to find out who committed the murder.
When you think you know the answer, go to the next section.

Check your solution

Did you find the killer?

```
INSERT INTO solution VALUES (1, 'Insert the name of the person you found here');

        SELECT value FROM solution;
```

Another common issue with these types of games is that they tend to be targetted at children or younger students, making them unappealing for older students who could also benefit from such content. This is the most prevalent in educational games which tackle scientific content as this type of information is usually expected to be presented in an academic setting to older students.
In addition, most of these educational games rely on a competition based approach and not many options are available for people who do not enjoy this competitive side and prefer to learn at their own pace.

3 Proposed Solution

The game this project is aimng to create will be single player to avoid the overused competitive format, but will contain a few achievements in order to motivate users to complete it. Since the targetted group are older students, the storyline will be engaging, and more parts of the story will be revealed the more they play. The concept is very similar to that of the game Hades [3], which, even though is not an educational game, applies some techniques that promote reusability of content. An example of this would be that as the players are more advanced, they are required to chose some obstacles to make the game more challenging, even though the content itself stays the same.

4 Objectives

The aim of this project is to investigate methods of adapting educational content to maximise reuse across knowledge groups. The result will be an escape-room style game which will make use of the methods found and will focus on four topics in computer science: object oriented programming, logic and verification, database systems and web development. I chose these particular topics as they are suitable for beginners and allow for a variety of puzzles to be implemented. Some examples of these will be given in the methodology section.

The game will consist of a couple of runs, their number being dependent of the difficulty, in which the player will have to escape from a few rooms, each room requiring the completion of a variety of puzzles. There will be three difficulty levels: beginner, intermediate and advanced and the player will be able to switch to any of them during the game. In order to help beginner and intermediate players solve the puzzles, hints will be placed around the room. Such hints could include posters on the walls, books, computers displaying already written code and any object that could contain some text refering to computer science knowledge.
Main objectives:

- Functional:

  1. The player will be able to select a difficulty level from the provided list: beginner, intermediate and advanced
  2. The player will be able to move around the world from the perspective of a controllable character
  3. It will be possible to change the difficulty at any time
  4. After playing for some time, the player will be prompted to switch to a more advanced difficulty
  5. Beginner and intermediate level players will be able to find hint around the rooms to aid them in solving the puzzles
  6. Intermediate and advanced level players will have a time limit to solve some of the rooms and puzzles
  7. The players will be able to see a countdown of how much time they have left
  8. Minimalistic sound effects will be used to make the experience more immersive
  9. The player will be able to complete a run of the game by escaping from 3 to 6 rooms
  10. After the player completes a run of the game, the story will advance
  11. Each room will contain 3 to 5 puzzles that require computer science knowledge
  12. Each of the puzzles will have elements that can be randomly generated to ensure the chances of encountering the same puzzle twice are as low as possible.
  13. The rooms will also contain a few basic escape-room style puzzles
  14. After a player encounters a certain room, the chance of it reapearring will be lowered

- Non-functional:
  1. The game will be entertaining
  2. People with no previous computer science knowledge will be able to play the game
  3. The game will be challenging for players with previous computer science knowledge
  4. The storyline will be engaging

Extensions:
These are objectives which may not be necessary for the goal of the project to be achieved, but which could further improve the quality of the game. Since they are not essential, they will only be implemented at the very end of the development process, if time allows.

1. More intricate sound effects could be used to improve the atmosphere of the game
2. New computer science topics could be added
3. The 3D assets could be improved to make the experience more emerssive

5 Methodology

An agile approach will be the most suitable for this project as I may need to quickly make changes to the initial plan, based on the feedback received during the testing phase or the information acquired during the research step. This agile approach will be combined with plan-based elements to ensure the progress and changes made to the initial objectives are well documented.
Since this is a large project that involves more aspects than just creating a game, its development could be divided in four main steps:

5.1 Research

This step will constitute the foundation of the project as it will define some of the objectives of the implementation stage. With the research done up to this point some of the methods of adapting content that were found are:

- using probabilities to ensure beginners have a lower chance of encountering more difficult puzzles;
- providing hints for the less advanced categories of players to aid them in solving the puzzles;
- using a time limit to make the game more challenging for the more advanced players.

Further research will be done by reading related literature and exploring existing games that implement some of the principles targetted by this project. Depending on the results of the research step, the list of objectives presented in this document might expand.

5.2 Resources Creation

This step involves creating or acquiring 3D assets and sound effects, designing the rooms and puzzles involved in the game and creating the storyline.
The initial 3D models will not be complex to allow more time to be spent for research and implementation since that is the main focus of the project. As development progresses, if time allows, more complex assets might be created in order to make the game more appealing to the players.
There will be two types of puzzles involved in the game, puzzles requiring computer science knowledge and normal puzzles usually found in escape-room style games. The latter will involve finding

various objects and combining them to achieve tasks, solving riddles and decrypting codes. The computer science puzzles will require players to investigate already written code, then apply the knowledge to write their own, using SQL commands to obtain information, modifying already existing code to affect the environment and solving a variety of problems in order to gain access to certain items.

Even though creating an interesting storyline to be used in the game is not the main objective of this project, it is an essential element in assuring that the content is interesting for the target players, which are people who are interested in computer science.

5.3 Game Implementation

This stage will be intercalated with the testing phase because the game may need further improvement upon receiving feedback. Therefore, after a version of the game is completed, it will be playtested by a variety of players, each of them providing feedback which will be taken into account for the next version. Ideally major changes will only occur from the first to the second version, yet the schedule will include time for a major change between the second and third version to account for the worst case scenario. If the second version does not need major improvements, the remaining time will be used to implement some of the extensions mentioned in objectives, in order to make the game more polished.

5.4 Testing phase

Since the aim of the project is to maximise reuse of content across knowledge groups, people with various computer science knowledge will be needed for the testing phase. The targetted groups that I will reach out to are the Game Design Society, collegues from the computer science course and friends with no computer science experience. Along with playtesting the game, they will be asked to provide feedback, evaluating their experience with it and suggesting some improvements. Depending on the results, some of the suggestions may become main objectives or extensions. In order to check that the non-functional objectives have been met, the evaluation will be done from a scale of 1 to 10, aiming for an average of at least 7.

6 Timetable

Week	Activity/Deadline
T1 W1	Write project specification
T1 W2	**Project specification deadline**
T1 W2-W3	Read relevant literature
T1 W4	Create 3D assets and soundtrack
T1 W5-W6	Create puzzles + storyline
T1 W5-W7	Implement simple aspects of the game such as a title screen, player controls, options menu
T1 W8-W9	Introduce the rooms and puzzles in the game, giving them a chance to appear based on the difficulty chosen by the player and other factors
T1 W9	**Progress report deadline**
T1 W10	Implement any missing features and write a questionnaire for the testing phase
Christmas Holiday	Begin the testing phase
T2 W1	Analyse results from the testing phase and write up new objectives
T2 W1-W3	Implement the new objectives
T2 W4	Start new testing phase and interpret results
T2 W3-W4	Write first half of final report
T2 W4-W5	Implement any new objectives or extensions
T2 W6	Final testing phase
T2 W7	Game is complete
T2 W8-W9	**Project presentation**
T2 W8-W9	Finish writing final report
T3 W1	**Final report**

Aside from the tasks pictured above, meetings with the project supervisor will also occur every two weeks to discuss progress and request advice if needed.

7 Resources

The game engine I will use in the development of this project is Unity, which uses C# as the programming language. Unity is the most advantageous game engine as it has a large community, therefore there are many guides and articles I could use in case I encounter any difficulties during the development of the game. Some other resources I will be using include Blender (for 3D assets) and Bfxr (for sound effects). All of these resources are free to use and easily accessible online, with official tutorials available, therefore, acquiring and learning how to use them should not take time away from the actual project.
All the work will be done on my personal computer as I already installed some of these resources and using DCS computers while they are not essential for the development of my project might entail the unnecessary risk of them being unavailable at certain times.

8 Risks

While considering the risks that could affect the proposed timetable, I identified the following list, along with some possible solutions:

- The computer I am working on malfunctions. A backup of the project will be kept on GitHub so I can continue working on it from a different device.

- I am unable to work due to unforseen circumstances. The timetable accounts for a few holidays and some free time which can be used to recover the days I am unable to work.

- I might run into some Unity related issues since I do not have that much experience with it. As mentioned previously, there are various online resources to help with solving that.

- I have no experience in creating 3D assets, so I migth not be able to create quality ones. If I don't manage to create my own, I can always look online for already existing ones and their quality is not one of the main objectives of this project.

- People might not be available to test the game during term time. Since testing is the most important aspect of the game, the first version of the game will be completed before Christmas holiday so that people can have time to test it then.

9 Legal, Social, Ethical and Professional Issues

Since all the data that will be collected from human participants is limited to questionnaires related strictly to the contents of the project, no ethical issues need to be considered.

References

[1] *Coding Games and Programming Challenges to Code Better.* URL: `https://www.codingame.com/start` (visited on 10/12/2022).

[2] *SQL Murder Mystery.* URL: `http://mystery.knightlab.com` (visited on 10/12/2022).

[3] *Hades (video game).* 2022. URL: `https://en.wikipedia.org/w/index.php?title=Hades_(video_game)&oldid=1115058824` (visited on 10/12/2022).

[4] George Kalmpourtzis. *Educational Game Design Fundamentals: A journey to creating intrinsically motivating learning experiences.* 2019.

[5] Nicolas Dietrich. "Escape Classroom: The Leblanc Process—An Educational "Escape Game"". In: *Journal of Chemical Education* 95.6 (2018), pp. 996–999. DOI: `10.1021/acs.jchemed.7b00690`. eprint: `https://doi.org/10.1021/acs.jchemed.7b00690`. URL: `https://doi.org/10.1021/acs.jchemed.7b00690`.

[6] Anonymous. *Game-based learning across the disciplines.* Ed. by Carmela Aprea and Dirk Ifenthaler. 2021.

[7] *20 Programming Games to Level Up Your Programming Skills.* 2021. URL: `https://x-team.com/blog/coding-games/` (visited on 10/12/2022).