Interim Report: Integration of Mixed Reality in Educational Escape Room for Self-Directed Learning in Object-Oriented Programming

Shaoteng KE
20320287
scysk1@nottingham.edu.cn

Supervised by: Dr. Boon Giin LEE

Submitted on 19 December 2024

School of Computer Science University of Nottingham, Ningbo China

Contents

1 Introduction

1.1 Background and Motivation

Self-directed learning has been widely recognized for its essential role in fostering students' critical thinking, problem-solving abilities, and intrinsic motivation, particularly in complex and abstract fields like computer science (CS). As one of the most high-potential disciplines, CS education has significantly benefited from integrating advanced educational technologies, such as large language models (LLMs) and gamification. However, a significant gap remains in promoting self-directed learning for certain challenging concepts in CS, especially in object-oriented programming (OOP), where abstract concepts can be difficult for students to grasp independently.

Given these challenges, educational escape rooms (EERs) have emerged as an promising approach to active learning and breakout traditional classroom, blending problem-solving tasks with narratives to create immersive learning experiences [1]. EERs have been applied in many academic disciplines, including computer science [2], physics [3], mathematics [4–8], biology [9], and chemistry [10]. Furthermore, as EERs are based on a game structure and design, they are linked to the methodology of gamification [1]. Students engage in subject-specific challenges within a structured, game-based environment. By requiring students to solve a series of puzzles to "escape", EERs foster critical thinking and the application of knowledge in a stimulating, time-bound setting. However, traditional physical EERs' are resource-intensive, restricted by classroom availability, budget, and a lack of time for preparation [11].

To address these limitations, several researches turned to digital educational escape rooms (DEERs) to provide greater flexibility and immersion [12]: Baziak *et al.* developed a mathematical DEER integrating Cave Automatic Virtual Environment (CAVE) technology [4]; Tsalidis *et al.* created Earscape, an auditory DEER using immersive virtual reality (IVR) [13]; Guillen-Sanz *et al.* developed an IVR escape room for teaching physiological regulation [14]. However, despite the growing interest in DEERs, few studies have explored the integration of head-mounted display (HMD) mixed reality (MR) technology, which offers unique advantages by seamlessly blending physical and virtual environments, allowing users to interact with digital objects within their physical surroundings [15].

While the fusion of real and virtual elements has been proven to provide a more intuitive and immersive experience, enabling learners to connect abstract concepts with tangible interactions [15], there is limited exploration of MR-enhanced EERs in CS education. Therefore, it's worth investigating the integration of MR to the development of EERs focused on CS education, especially in OOP concepts.

1.2 Aims and Objectives

The project aims to integrate MR technology into the development of a programming EER game, specifically focusing on enhancing self-directed learning in OOP. By combining the immersive aspects of MR with the interactive nature of EERs, the project seeks to provide an innovative and engaging platform for students to grasp complex programming concepts, with the addressing the following research questions as objectives:

- **RQ1:** How does immersive MR EER gameplay influence students' learning motivation, engagement, and learning outcomes within the context of OOP self-directed learning?
- **RQ2:** How does the integration of physical and virtual elements in the MR environment influence students' problem-solving and critical thinking?

2 Related Work

2.1 Education Escape Rooms (EERs)

Escape rooms have considerable use in educational settings due to their potential to enhance learning experiences. Fotatris and Mastoras [11] conducted a systematic review based on 68 peer-reviewed studies exploring the impact of EERs on education. They concluded that EERs are able to provide students with an engaging learning experience, fostering the development of critical cognitive skills such as creativity, decision-making, and critical thinking. Quek et al. [16] reviewed 52 studies focusing on the use of EERs in the healthcare field. Their findings revealed that EERs are enjoyable for students and can contribute to increased motivation for learning. They also identified DEERs as a promising and emerging approach in educational settings. For more specific educational contexts, López-Pernas et al. [2] examined the use of EER in higher education programming courses. They concluded that EERs can significantly enhance student engagement and learning outcomes, with students showing a stronger preference for EERs compared to traditional teaching methods and materials. Morrell and Ball [17] explored the application of two different versions of EERs in undergraduate nursing education, revealing that EERs encouraged students to reflect on their successes and challenges more effectively. The students involved in the experiment exhibited higher levels of engagement and competitiveness, further highlighting the potential of EERs to foster active participation in learning. Their research highlighted the potential of EERs to create interactive and effective learning environments across different fields of study.

2.2 MR in Education

Speicher et al. concluded [18] that MR is an interactive technology that merges virtual and physical elements in real time, enabling learners to engage with both environments and creating immersive and hands-on learning experiences. MR has been increasingly utilized in education to enhance learning outcomes through its hands-on, experiential approach. Pellas et al. [19] reviewed 42 studies on MR applications in K-12 education, finding that MR can significantly improve student engagement, participation, skill acquisition, and knowledge transfer within well-structured instructional designs. Similarly, Banjar et al. [20] conducted a systematic review of 12 studies in higher education, concluding that MR is beneficial, feasible, and effective, enhancing learning environments, increasing student motivation, and improving performance. However, a gap remains in understanding how the integration of physical and virtual elements in MR environments influences students' problem-solving and critical thinking, particularly in complex subjects like OOP.

Despite the valuable insights provided by existing studies, there is a notable gap in research exploring the integration of MR and EERs in programming education. To address this gap, this project focuses on designing strategies that combine MR with an EER game for teaching OOP. It aims to reduce the difficulty of learning complex programming concepts and explore how immersive EER gameplay and MR visualization influence problem-solving and critical thinking during the learning process. The goal is to use MR interaction and animation to visualize OOP concepts, thereby enhancing the learning motivation and outcomes for beginner programmers.

3 Methodologies

3.1 Game Design Framework

The Game Design Document (GDD) serves as a comprehensive blueprint for game development projects, outlining the game's structure and design framework [21]. It plays a vital role in defining the scope, establishing the game's overall direction, and ensuring consistency across the development process. For this AVP MR project, implemented using the Unity3D engine, the Unity Project Design Document (PDD) serves as the primary GDD. The document is organized with six key components: player control, basic gameplay, sound and effects, gameplay mechanics, user interface, and other features. The PDD template can be found in Appendix A.1.

In this section, the design framework includes a clear overview of the game, followed by detailed descriptions of three levels within the game. These level descriptions are systematically supported by the six components outlined in the PDD, which ensures clarity, consistency, and effective guidance throughout the project's implementation.

3.1.1 Game Design Overview

In this MR Escape Room game, players are required to learn different OOP concepts and practice hands-on programming skills to solve puzzles and escape the room. In level 1, the player's task is to collect the body components of a small robot companion scattered throughout the room. Using a drag-and-drop graphical interface for OOP programming at a workbench, the player assembles the robot successfully.

After assembling the robot, level 2's objective is to activate it by finding the robot's TF card hidden within the room. The player must program a Detector and use the programmed Detector to search for the TF card in the physical room.

Once the robot is activated, a small maze rises from the tabletop. At level 3, the player must apply more advanced OOP concepts to program the robot's automated movements. The goal is to guide the robot through the maze, overcoming various obstacles, and ultimately reaching the maze's end to obtain the key needed for both the robot and the player to escape the room.

The game design overview is summarized in Table 1.

Table 1: Overview of game design for learning OOP concepts

Type	Description	OOP Concepts
Level 1	The player will gather the robot's body components and assemble them on the workbench by solving coding puzzles	Class; Object; Methods; Attributes
Level 2	The player will program a detector and utilize it to search the room for the TF chip, which is needed to activate the robot	Constructors; Parameters
Level 3	The player will develop the robot's movement logic to navigate through the maze, overcome obstacles, and retrieve the key required for their escape	Inheritance; Abstraction; Encapsulation; Polymorphism

3.1.2 Level 1: Assemble Your Partner

The PDD for the Level 1: "Assemble Your Partner" is outlined as follows:

1. *Player Control*: The player takes the role of themselves in this "Assemble Your Partner" level, where the player is free to explore in the MR environment. Using **intuitive drag-and-drop gestures**, players **solve coding tasks to assemble the robot's body components**.

2. *Basic Gameplay*: During the game, **robot body parts** appear **scattered throughout the room**, and the goal of the game is to **collect these components and assem-**

ble the robot on the workbench by solving coding puzzles on a drag-and-drop interface.

3. *Sound and Effects*: There will be sound effects **when the player picks up, drops, or successfully assembles a robot part**, and animation of the robot assembling with particle effects **when code blocks are placed correctly into position**. There will also be an animation of unsuccessful assembling **when the player puts the code blocks in the wrong way**.

4. *Gameplay Mechanism*: As the game progresses, the player must **learn and use beginner-level OOP concepts, including Class, Object, Methods, and Attributes**, to solve the coding task.

5. *User Interface*: The **drag-and-drop coding interface** will appear above the workbench **whenever the player places robot components floating on it**. At the start of the game, the title "Assemble Your Partner" will appear, and the game will end **when all robot parts are correctly assembled**.

6. *Other Features*: There will be a **narrative introduction from the broken small robot** in a weak voice explaining its importance in progressing through the escape room. **Hints** will be available for players who struggle with either finding components or solving coding puzzles.

3.1.3 Level 2: Search the TF Card

The PDD for the Level 2: "Search the TF Card" is outlined as follows:

1. *Player Control*: The player takes the role of themselves in this "Search the TF Card" level, where the player is free to explore in the MR environment, **program a detector to search for and locate the TF card within the room**.

2. *Basic Gameplay*: During the game, **the TF card of the robot is hidden somewhere in the room**, and the goal of the game is to **program and use a functional Detector to find the TF card and activate the robot**.

3. *Sound and Effects*: There will be sound effects **when the player successfully programs the Detector and when the TF card is found**, and particle effects and beeps **indicate hotspots or the card's location when the Detector is active**.

4. *Gameplay Mechanism*: As the game progresses, the player **must adjust and refine the detector's programming logic by learning and using middle-level OOP concepts, including Constructors and their parameters**.

5. *User Interface*: The **drag-and-drop coding interface** will appear above the workbench **whenever the player places the detector or the robot floating on it**. At the start of the level, the title "Search the TF Card" will appear, and the game will end **when the robot is activated**.

6. *Other Features*: The robot will provide **guidance or hints** if the player struggles to program the detector or locate the TF card.

3.1.4 Level 3: The Maze

The PDD for the Level 3: "The Maze" is outlined as follows:

1. *Player Control*: The player takes the role of themselves in this "The Maze" level, where they can freely explore the MR environment. The player **programs the robot's movement logic, integrating new functional components**, to guide it through a maze to progress.

2. *Basic Gameplay*: During the game, **a small maze will rise from the tabletop**, and the goal of the game is to **program the robot to overcome obstacles and reach the end of the maze** to obtain the escape key.

3. *Sound and Effects*: There will be animation **when the robot is armed with new functional components**. There will also be sound effects **when the robot successfully navigates obstacles or reaches checkpoints in the maze**, and particle effects **highlight the robot's path or obstacles encountered**.

4. *Gameplay Mechanism*: As the game progresses, **the player must apply advanced OOP concepts to program** the robot's movement logic and arm the robot, **including Abstraction, Encapsulation, inheritance, and Polymorphism**, making it capable of **navigating the maze and avoiding obstacles**.

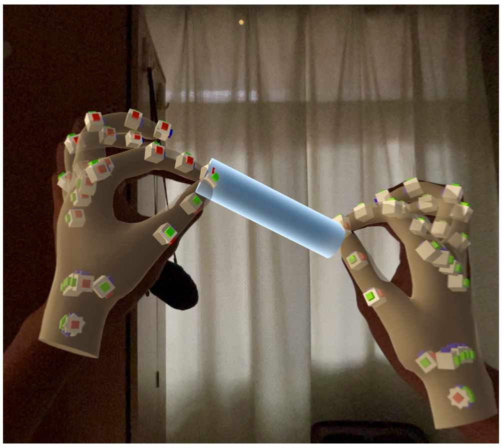
5. *User Interface*: **The robot programming interface will appear** above the workbench, allowing players to code the robot's movement step-by-step. At the start of the level, the title "The Maze" will appear, and the game will end **when the robot reaches the maze's end and retrieves the key to escape**.

6. *Other Features*: The maze **will dynamically interact with the robot's movements**, featuring obstacles, enemies, and pathways that test the player's programming skills.

3.2 Implementation

3.2.1 MR Interaction Setup

Through the gameplay, the player can freely interact with virtual objects within the real escape room. There are two types of interactions: direct interaction and indirect interaction, as depicted in Fig. 1. The former involves physically manipulating virtual objects using the hands, while the latter utilizes eye-tracking to select objects, followed by hand gestures to move, rotate, or scale them. Basic interactions include Picking/Transforming, Scaling, and Rotating. MR interaction has been successfully implemented in this project using Polyspatial and the Unity XR Interaction Toolkit.



(a)

(b)

Figure 1: Two types of interactions provided by AVP, including (a) direct interaction and (b) indirect interaction

3.2.2 Spatial Anchoring

In this project, the Spatial Anchor serves as a key feature for persisting and stabilizing virtual content within the real-world environment. For example, before the game starts, the researcher/teacher can customize the location and rotation of the workbench, ensuring it remains permanently fixed on the real escape room's tabletop throughout the game, as depicted in Fig. 2. The functionality for automatic loading, manual creation, and deletion of spatial anchors to fix the position of virtual objects has been implemented in this project using AR Foundation.

3.2.3 Hands Gesture Tracking

Hand Gesture Tracking has been implemented in this project using XR Hands. It enables the game to continuously track the player's hand gestures and dynamically trigger events based

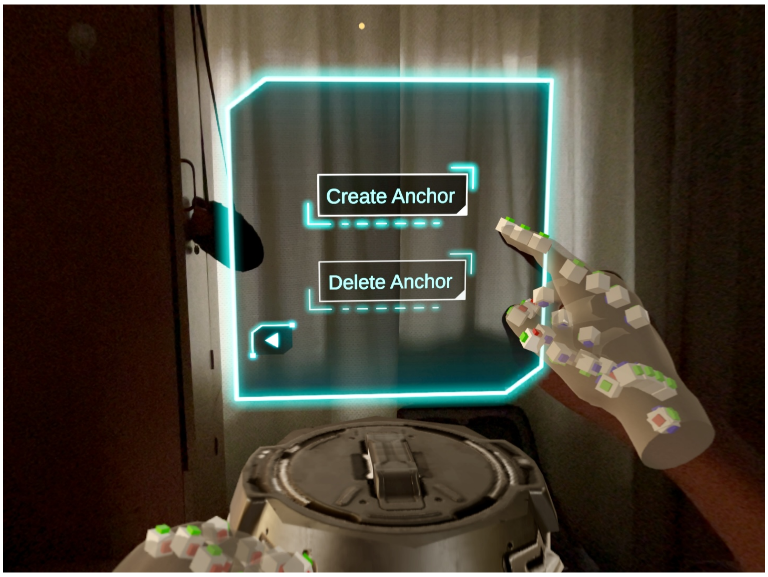


Figure 2: The screenshot of the implemented spatial anchoring interface

on the current gesture. As shown in Fig. 3, this enables the game to simulate a real grabbing action instead of the default pinch gesture to interact with the detector.

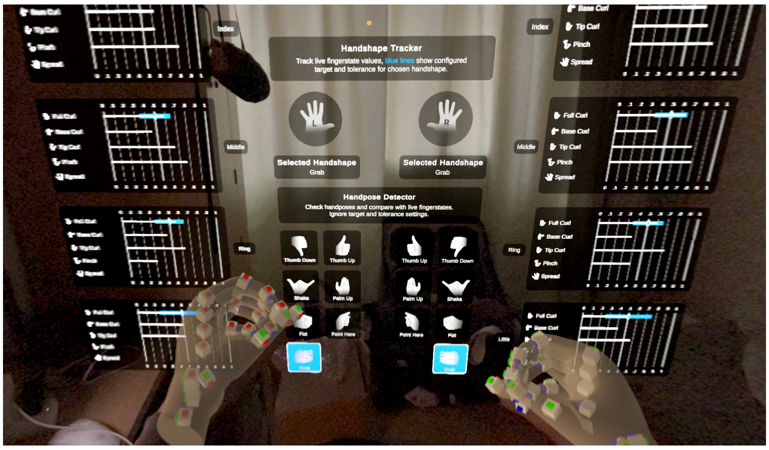


Figure 3: Example of hand gesture tracking for grabbing

4 Project Management

Fig. 4 shows the updated project timeline, with completed milestones marked in purple. All progress has remained consistent with my previously proposed schedule. The project started with a comprehensive literature review, covering relevant fields as summarized in Sections 1 and 2, and established clear research questions to guide the work. This foundation has ensured a structured approach to the methodologies outlined in Section 3, including game design, experimental planning, and implementation. Moving forward, the primary focus will be completing the game implementation to ensure the system is ready for the following controlled experiment. The results are planned to be submitted to an education- or HCI-related conference or journal.

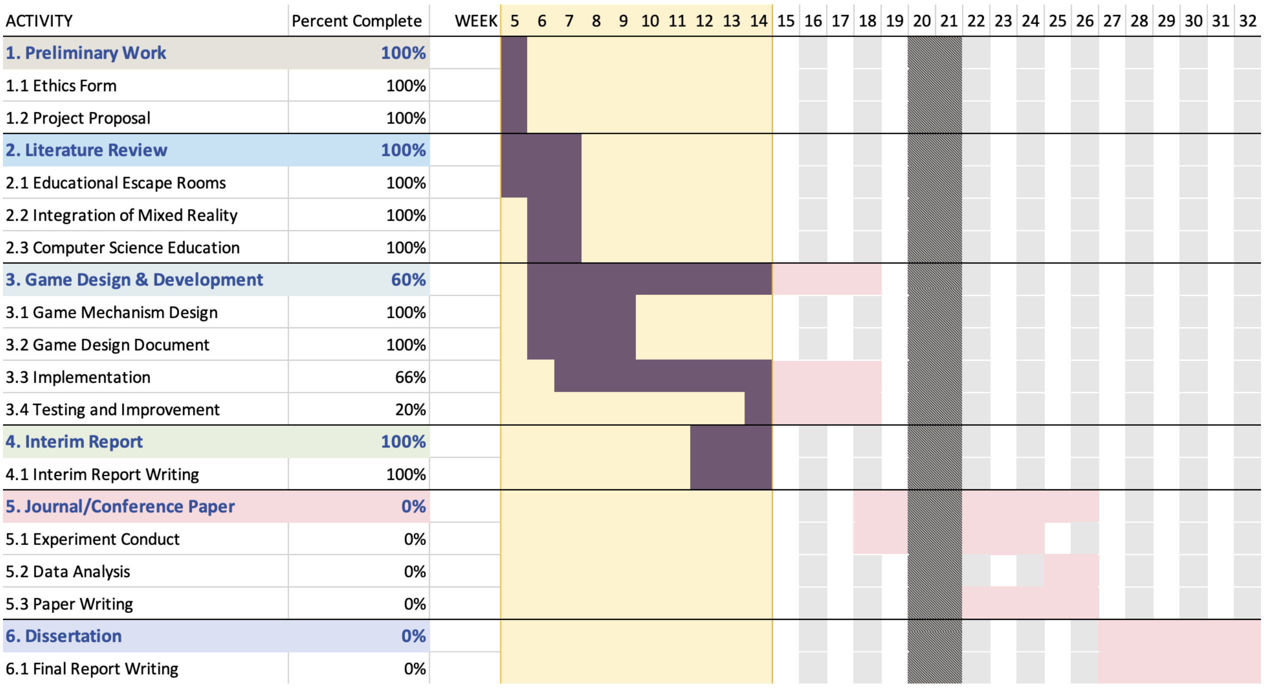


Figure 4: Gantt chart of the project timeline

5 Reflection

Since the beginning of this semester, the majority of the early milestones were achieved as planned, such as background research, formulation of the research questions, game design, and design documentation.

However, despite these positive outcomes, I encountered numerous challenges related to the implementation, particularly with the use of AVP development tools. The greatest problem is the instability of developing the AVP project in Unity. Given that many of the AVP development tools and packages are still in beta, some key features, such as gesture recognition, have been unreliable. This has resulted in prolonged testing cycles, as real-device debugging in Unity is hindered by issues with direct deployment to the AVP, making the bug-fixing process both time-consuming and cumbersome. Additionally, the need to use Reality Kit for AVP rendering has created compatibility issues, as many commonly used Unity art assets are no longer supported in the new device. As a result, additional efforts have been required to create or source alternative visual resources for the project.

Therefore, in response to these challenges, the timeline for future work will be adjusted to allocate more time for the resolution of unforeseen technical issues, as well as for the preemptive acquisition of necessary art assets.

References

[1] A. Veldkamp, L. van de Grint, M.-C. P. Knippels, and W. R. van Joolingen, "Escape education: A systematic review on escape rooms in education," *Educational Research Review*, vol. 31, p. 100364, 2020.

[2] S. López-Pernas, A. Gordillo, E. Barra, and J. Quemada, "Examining the use of an educational escape room for teaching programming in a higher education setting," *IEEE Access*, vol. 7, pp. 31723–31737, 2019.

[3] S. Mystakidis, G. Papantzikos, and C. Stylios, "Virtual reality escape rooms for stem education in industry 4.0: Greek teachers perspectives," in *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pp. 1–5, IEEE, 2021.

[4] R. Baziak, T. Daruk, K. Żyra, D. Żarek, and J. Lebiedź, "Virtual escape room in mathematics," 2024.

[5] A. Glavaš and A. Staščik, "Enhancing positive attitude towards mathematics through introducing escape room games," *Mathematics education as a science and a profession*, vol. 281, p. 293, 2017.

[6] A. Fuentes-Cabrera, M. E. Parra-González, J. López-Belmonte, and A. Segura-Robles, "Learning mathematics with emerging methodologies—the escape room as a case study," *Mathematics*, vol. 8, no. 9, p. 1586, 2020.

[7] H. Kaufmann, "Virtual environments for mathematics and geometry education.," *Themes in science and technology education*, vol. 2, pp. 131–152, 2009.

[8] Á. A. Magreñán, C. Jiménez, L. Orcos, and S. Roca, "Teaching calculus in the first year of an engineering degree using a digital escape room in an online scenario," *Computer Applications in Engineering Education*, vol. 31, no. 3, pp. 676–695, 2023.

[9] G. Alonso and K. T. Schroeder, "Applying active learning in a virtual classroom such as a molecular biology escape room," *Biochemistry and Molecular Biology Education*, vol. 48, no. 5, pp. 514–515, 2020.

[10] A. Janonis, E. Kiudys, M. Girdžiūna, T. Blažauskas, L. Paulauskas, and A. Andrejevas, "Escape the lab: Chemical experiments in virtual reality," in *Information and Software Technologies: 26th International Conference, ICIST 2020, Kaunas, Lithuania, October 15–17, 2020, Proceedings 26*, pp. 273–282, Springer, 2020.

[11] P. Fotaris and T. Mastoras, "Escape rooms for learning: A systematic review," in *Proceedings of the European Conference on Games Based Learning*, pp. 235–243, 2019.

[12] A. Makri, D. Vlachopoulos, and R. A. Martina, "Digital escape rooms as innovative pedagogical tools in education: A systematic literature review," *Sustainability*, vol. 13, no. 8, p. 4587, 2021.

[13] C. Tsalidis, A. Adjorlu, L. M. Percy-Smith, and S. Serafin, "Earscape: A vr auditory educational escape room," in *Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–2, 2024.

[14] H. Guillen-Sanz, I. Q. Bayona, and G. L. Pérez, "Design and development of an immersive virtual reality serious game with biofeedback for physiological regulation: Alice, beyond reality," in *European Conference on Games Based Learning*, vol. 18, pp. 312–319, 2024.

[15] S.-C. Chen and H. Duh, "Mixed reality in education: Recent developments and future trends," in *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, pp. 367–371, IEEE, 2018.

[16] L. H. Quek, A. J. Tan, M. J. Sim, J. Ignacio, N. Harder, A. Lamb, W. L. Chua, S. T. Lau, and S. Y. Liaw, "Educational escape rooms for healthcare students: A systematic review," *Nurse education today*, vol. 132, p. 106004, 2024.

[17] B. L. Morrell and H. M. Ball, "Can you escape nursing school? educational escape room in nursing education," *Nursing Education Perspectives*, vol. 41, no. 3, pp. 197–198, 2020.

[18] M. Speicher, B. D. Hall, and M. Nebeling, "What is mixed reality?," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–15, 2019.

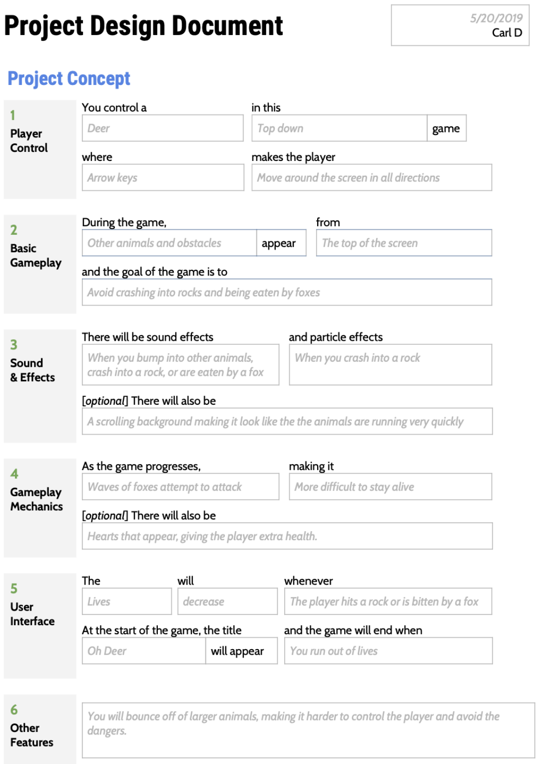
[19] N. Pellas, I. Kazanidis, and G. Palaigeorgiou, "A systematic literature review of mixed reality environments in k-12 education," *Education and Information Technologies*, vol. 25, no. 4, pp. 2481–2520, 2020.

[20] A. Banjar, X. Xu, M. Z. Iqbal, and A. Campbell, "A systematic review of the experimental studies on the effectiveness of mixed reality in higher education between 2017 and 2021," *Computers & Education: X Reality*, vol. 3, p. 100034, 2023.

[21] Nuclio, "Game design document template and examples." `https://www.nuclino.com/articles/game-design-document-template`, 2023. Accessed: 06 Dec 2023.

A Appendices

A.1 Project Design Document

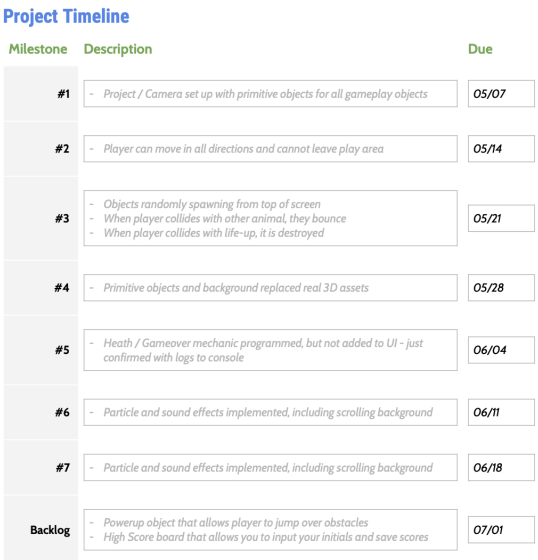


Project Design Document — 5/20/2019, Carl D

Project Concept

1 Player Control	You control a *Deer* in this *Top down* game where *Arrow keys* makes the player *Move around the screen in all directions*
2 Basic Gameplay	During the game, *Other animals and obstacles* appear from *The top of the screen* and the goal of the game is to *Avoid crashing into rocks and being eaten by foxes*
3 Sound & Effects	There will be sound effects *When you bump into other animals, crash into a rock, or are eaten by a fox* and particle effects *When you crash into a rock* [optional] There will also be *A scrolling background making it look like the the animals are running very quickly*
4 Gameplay Mechanics	As the game progresses, *Waves of foxes attempt to attack* making it *More difficult to stay alive* [optional] There will also be *Hearts that appear, giving the player extra health.*
5 User Interface	The *Lives* will *decrease* whenever *The player hits a rock or is bitten by a fox* At the start of the game, the title *Oh Deer* will appear and the game will end when *You run out of lives*
6 Other Features	*You will bounce off of larger animals, making it harder to control the player and avoid the dangers.*

(a)

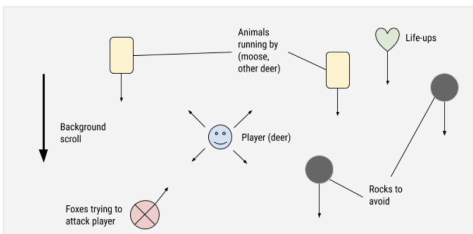


Project Timeline

Milestone	Description	Due
#1	- Project / Camera set up with primitive objects for all gameplay objects	05/07
#2	- Player can move in all directions and cannot leave play area	05/14
#3	- Objects randomly spawning from top of screen - When player collides with other animal, they bounce - When player collides with life-up, it is destroyed	05/21
#4	- Primitive objects and background replaced real 3D assets	05/28
#5	- Heath / Gameover mechanic programmed, but not added to UI - just confirmed with logs to console	06/04
#6	- Particle and sound effects implemented, including scrolling background	06/11
#7	- Particle and sound effects implemented, including scrolling background	06/18
Backlog	- Powerup object that allows player to jump over obstacles - High Score board that allows you to input your initials and save scores	07/01

(b)

Project Sketch



(c)

Figure 5: The PDD template, including (a) project concept (b) project timeline, (c) and project sketch.