CMPE 491
Senior Project 1

“VR Project Blue”
(VR Video Game)

Project Supervisor
Dr. Özlem Albayrak

Project Team Members
Uğur Evren Çamalan
Sancaralp Elmas
Eren Çağlaroğlu
Hasan Yasin Yıldız

Project Jury Members
Dr.Ulaş Güleç
Dr. Elif Kurtaran Özbudak

Course Coordinator
Dr. Fırat Akba

Project Website
teduvrsp.github.io

1.Introduction

The purpose of this report is to introduce an educational serious hacker game developed by our team. The game takes place in a VR environment and aims to merge gaming and learning, offering an immersive experience with the goal of imparting knowledge to players. In the realm of serious games, this innovative genre seeks to educate individuals while providing an enjoyable gameplay experience. Our game focuses on the captivating world of white-hat hacking, where players assume the role of a skilled hacker. The objective is to collaborate with corporations, infiltrate their systems, and teach the art of defense in the process. By employing social engineering techniques such as USB bait and mail bait, players gain hands-on experience in the field of cybersecurity. The ultimate aim of the game is to equip individuals with a solid foundation in this domain and inspire them to explore further possibilities within the cybersecurity field.

Throughout this report, we will search into the unique features and design elements of our game, illustrating how it seamlessly blends entertainment and education. We will explore the captivating narrative that drives the gameplay, the interactive learning materials incorporated into the game, and the progressive difficulty curve that challenges players while nurturing their skills. Additionally, we will discuss the significance of feedback and statistics in enhancing the learning experience. It is our belief that this educational serious hacker game will revolutionize the way individuals perceive cybersecurity. By immersing players in an engaging gameplay environment, we aim to foster a generation of knowledgeable and vigilant digital citizens. This report will serve as a comprehensive guide to our game, outlining its key features and highlighting the importance of cybersecurity education in today's rapidly evolving digital landscape.

1.1 Purpose of the system

Our main goal in the project we are working on is to create a project that not only provide to individuals who are interested in or want to acquire knowledge in the field of cybersecurity but also provides them with an enjoyable experience. We aim to achieve this by incorporating VR technology, aiming to become pioneers in this field.

1.2 Design goals

1. Merge gaming and learning: The primary design goal is to seamlessly blend gaming and learning elements, creating an immersive experience for players. The game should not only entertain but also educate, imparting knowledge and skills in the field of cybersecurity.

2. Immersive virtual reality environment: The game should leverage virtual reality technology to provide a highly immersive environment for players. This immersive experience will enhance engagement and enable players to feel like they are truly part of the hacking world.

3. Captivating narrative: The game should feature a compelling storyline that drives the gameplay. A captivating narrative will keep players engaged and motivated to progress through the game, while also providing context for the cybersecurity concepts being taught.

4. Hands-on experience in cybersecurity: The game should provide players with practical, hands-on experience in the field of cybersecurity. By allowing players to collaborate with corporations and engage in hacking activities, they can learn about various hacking techniques and the importance of defense strategies.

5. Social engineering techniques: The game should incorporate social engineering techniques, such as USB bait and mail bait, to educate players about real-world hacking methods. By experiencing these techniques in a controlled virtual environment, players can understand the risks and develop a better understanding of how to defend against them.

1.3 Definitions, acronyms, and abbreviations

VR [1]: Virtual Reality is a simulated environment that allows people to experience a virtual environment by interacting with it like a real environment by making use of their senses.

USB [2]: Universal Serial Bus plug and play interface enables seamless communication between technological devices and peripherals or other devices. It provides a convenient way for devices to connect and interact with each other without complicated setup processes.

CPU: CPU [3], also known as the Central Processing Unit, is a crucial hardware component often referred to as the "brain" of the system. It carries out essential tasks for computation and control.

GPU [4]: Graphics Processing Unit, is a hardware component designed to efficiently process and display graphics. It is specifically optimized for rapid graphics rendering. However, it can also be utilized for general-purpose computations, allowing for high-performance parallel calculations to be executed.

UI [5]: User Interface is an interface that appeals to the human senses that allows the user to interact with a software product or other digital product. The purpose of the UI is to facilitate the user's interaction with the product and to improve the user experience of the product in a positive way.

HUD (Head-Up Display) [6]: UI elements used to provide information needed by players during gameplay. The HUD displays information such as the player's health and score. In addition, elements such as mission objectives, maps or mini maps can be included in the HUD.

1.4 Overview

Our project aims to introduce an innovative educational serious hacker game that combines entertainment and learning. The game takes place in a virtual reality environment and immerses players in a captivating world of white-hat hacking. By assuming the role of a skilled hacker, players collaborate with corporations, infiltrate their systems, and teach the art of defense. The game incorporates social engineering techniques like USB bait and mail bait to provide hands-on experience in cybersecurity. The ultimate goal is to equip individuals with a solid foundation in cybersecurity and inspire exploration in this field. This groundbreaking game revolutionizes cybersecurity education by immersing players in an engaging experience, fostering knowledgeable digital citizens.

2. Proposed software architecture

Component based approach.

2.1 Overview

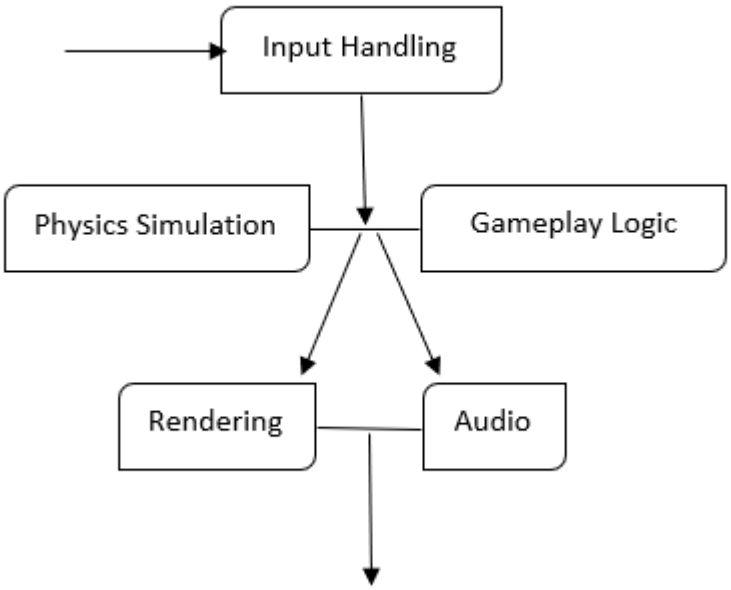
The proposed software architecture adopts a component-based approach that offers modularity, reusability, and maintainability, aligning perfectly with the requirements of our game. By designing game objects as compositions of reusable components, we gain several advantages. Components promote reusability by allowing us to create a library of shared components, reducing redundancy and improving code efficiency. This approach enables us to save development time and fosters a modular design where components can be easily understood, tested, and maintained individually, resulting in a more robust and manageable codebase.

This architecture is highly compatible with Unity's GameObject and MonoBehaviour system, which serves as the foundation of our game development. The GameObject system in Unity provides the structure for organizing and manipulating entities in the game world, while MonoBehaviour scripts attach behavior to these GameObjects. The component-based approach aligns seamlessly with Unity's architecture as components can be implemented as MonoBehaviours, allowing them to be attached to GameObjects and leveraging the benefits of Unity's built-in features and editor tools. This compatibility ensures a smooth integration with existing Unity workflows and maximizes the flexibility and extensibility of our game development process.

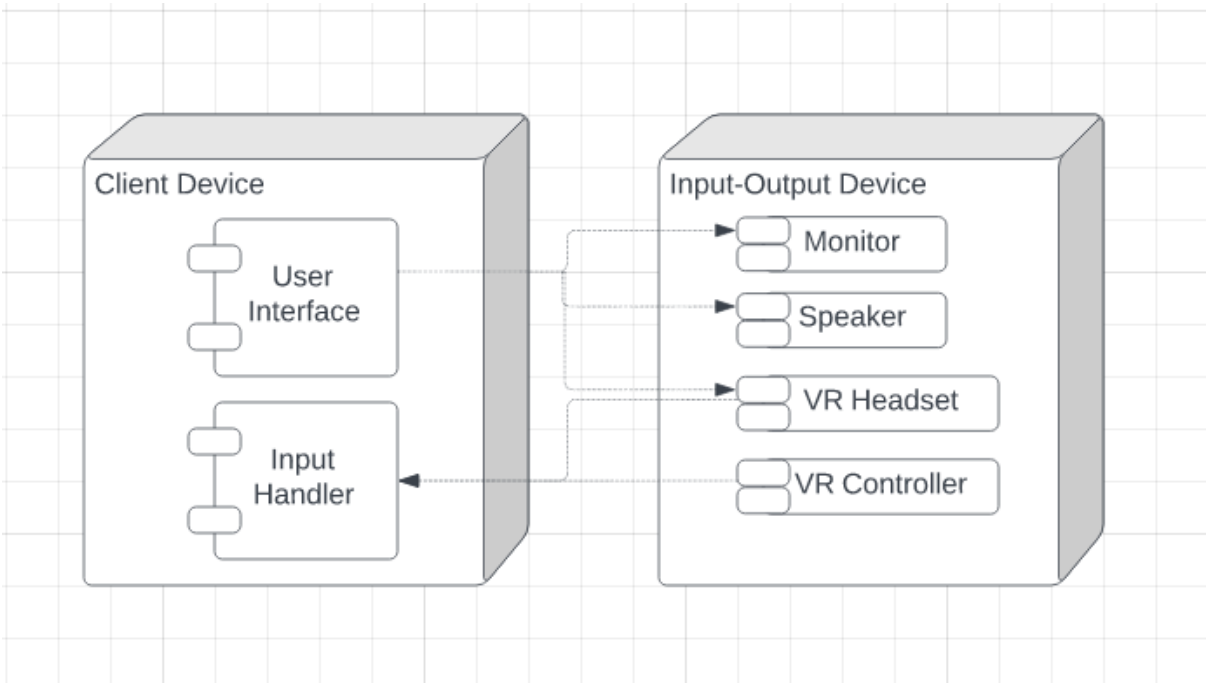
Furthermore, the component-based architecture fosters efficient collaboration within the development team. With each component encapsulating specific functionality, multiple team members can work on different components simultaneously without interference. This clear separation of concerns facilitates faster development cycles and smoother integration of various features. Additionally, the component-based approach allows for the easy addition, removal, or modification of components at runtime, providing dynamic and flexible gameplay behavior. This adaptability allows us to iterate on gameplay mechanics, experiment with different combinations of components, and readily respond to player feedback.

2.2 Subsystem decomposition

The hardware/software mapping for the game involves the VR headset as the primary hardware component, enabling an immersive virtual reality experience. Input devices, such as VR controllers, keyboard, mouse, or gamepads, capture user interactions. The GPU handles graphics rendering, while the CPU executes game logic and manages system performance. The game is developed using the Unity game engine, which interfaces with the operating system and utilizes a graphics rendering pipeline. Input handling libraries process user input, audio libraries manage sound effects and music, and optional networking libraries handle multiplayer functionality. This mapping ensures the interaction between the game software and hardware components, enabling the delivery of a compelling and immersive gameplay experience.



2.3 Hardware/software mapping



2.4 Persistent data management

When it comes to data management, maintaining online properties is unnecessary for this game as it is designed to be played offline by a single player. To handle save game operations, a local database is employed to store all the essential data. The user can save his progress online using the Steam Cloud feature according to his request. Steam Cloud has its own data privacy policies, and the user must accept them. From our point of view, we only need to make Steam Cloud compatibility. Data management is provided by Steam. The local database functionality is tailored specifically for the personal computer running the game, ensuring seamless operation. Leveraging the capabilities of Unity3D, which utilizes serialization, the game can convert data structures and object states into a format that can be reconstructed later. This convenient feature facilitates effortless management of both saving and loading game progress. Apart from these, there are only the game files as permanent data, and the user agrees to download this software to his computer by accepting the terms of use through Steam. Game files saved locally on the user's computer do not access the user's information in any way.

2.5 Access control and security

There is no authentication and login requirement as multiple different content will not be served for different users. The game is accessible to anyone who buys it on Steam. The player's game progress is stored locally or on the Steam Cloud. However, since only the progress of the game will be recorded and personal data will not be used, no security measures are taken to hide the data.

2.6 Global software control

Our project is serious game, so it equipped with an event-driven system, which means that the flow of the game is controlled by changes in its states. Right from the start of the game, the user will encounter various scenarios that prompt them to take specific actions, ultimately altering the course of the game. Each scenario will present the user with multiple options, and depending on the option chosen, the game will transition to a different state, branching out into diverse pathways.

2.7 Boundary conditions

The game incorporates three distinct boundary conditions: Initialization, Termination, and Failure, each serving a specific purpose within the gameplay experience.

a. Initialization:

The game starts with a menu screen that gives the user two options: the option to start the game by loading their previously saved progress, or to start a completely new game. If the player chooses to continue where he/she left off, player will continue from where player saved the last game. If the player is entering the game for the first time or if player wants to start a new game, the user will first complete the tutorial about the game and then starts to the game.

b. Termination:

During gameplay, the user encounters an in-game stop menu by pausing the game. If the player wishes, player can exit the game using this menu. The game will automatically save the game if player chooses to log out.

c. Failure:

If an unexpected malfunction causes the game to forcefully close, the player's progress is partially preserved due to the presence of the auto-save mechanism. Since the auto-save mechanism will run periodically, it ensures that the user's progress up to the last auto-save is preserved even if the game unexpectedly quits.

3. Subsystem services

Here are the key subsystems:

Input Handling Subsystem: This subsystem is responsible for handling user input, including keyboard, mouse, or VR controller inputs. It consists of components that interpret and process user interactions, such as movement, interaction with objects, and triggering actions within the game. Thanks to this subsystem, the player performs the movements of his character in the game through inputs.

Physics Simulation Subsystem: This subsystem is responsible for simulating the physical behavior of game objects, including collisions, gravity, and object dynamics. It utilizes components that interact with Unity's physics engine or custom physics implementations to provide realistic movement and interaction within the game world. The physics simulation subsystem is important for providing realistic movement and interaction in the game world. For example, whether a game's objects fall under gravity, react to collisions, or interact with other objects makes the game experience more believable.

Rendering Subsystem: This subsystem handles the rendering and visualization of the game's graphics. It includes components that control rendering settings, handle shaders, lighting, and special effects. Shaders determine how the surfaces of objects are displayed. Lighting simulates how light behaves in-game. Special effects are used to simulate the atmosphere, weather events or other events in the game world. This subsystem works closely with Unity's rendering pipeline to produce visually appealing and immersive visuals for the player.

Audio Subsystem: The audio subsystem is responsible for managing and playing audio within the game. It includes components that handle sound effects, background music, voice-overs, and spatial audio to enhance the overall audio experience for the player. Also, this subsystem allows players to be fully immersed in the game world and connect more deeply to the story or events of the game, thanks to its sound effects.

Gameplay Logic Subsystem: This subsystem contains components that implement the game's specific rules, mechanics, and logic. It encompasses components for player progression, game events, scoring systems, and game state management. The gameplay logic subsystem is an important component that forms the core of the game and shapes the player's experience. It is important that the game is designed correctly to ensure its fluency and fun. Thanks to this subsystem, the developer manages the content and rules of the game, ensuring the integrity and playability of the game.

User Interface Subsystem: This subsystem handles the user interface elements and interactions. It includes components for menus, HUDs, dialogues, and other UI elements. These components allow players to navigate through the game, customize settings, and interact with in-game menus. Game development platforms such as Unity provide developers with graphical tools and components to create and manage user interfaces.

4. Glossary

White-Hat Hacker [7]: White hat hackers are hackers who protect the system or software from malicious hacker attacks. Usually, their purpose is to prevent data leakage or system downtime. They test the system for security vulnerabilities with various techniques.

Cybersecurity [8]: Cybersecurity is protecting networks, servers and technological devices from malicious attacks or information leaks. Cybersecurity ensures that users are secure in a virtual environment and protects users and data from attacks as much as possible.

GameObject [9]: In Unity, the GameObject class acts as the foundation for all entities within a game scene. It serves as a versatile container for representing and organizing various objects or elements present in the game world, including characters, items, obstacles, lights, and cameras. GameObjects hold and manage multiple components that collectively determine the entity's behavior, appearance, and functionality. Essentially, GameObjects provide a framework to encapsulate and manipulate the fundamental building blocks of a Unity scene.

MonoBehaviour [10]: In Unity, every script is derived from the base class called MonoBehaviour. When using C# in Unity, it is mandatory for developers to explicitly inherit from MonoBehaviour.

5.Appendix

a) Hack and Cybersecurity Games Review

b) VR games

5. References

[1] Sheldon, R. (2022, August 3). What is virtual reality? WhatIs.com. https://www.techtarget.com/whatis/definition/virtual-reality#:~:text=Virtual%20reality%20is%20a%20simulated,perceived%20through%20the%20users'%20senses.

[2] *What is USB (Universal serial bus)?* Computer Hope. (2022, October 18). https://www.computerhope.com/jargon/u/usb.htm

[3] Gartner_Inc. (n.d.). *Definition of CPU (central processing unit) - Gartner Information Technology Glossary*. Gartner. https://www.gartner.com/en/information-technology/glossary/cpu-central-processing-unit

[4] Intel. (n.d.). *What is a GPU?* Intel. https://www.intel.com/content/www/us/en/products/docs/processors/what-is-a-gpu.html#:~:text=What%20does%20GPU%20stand%20for,designed%20to%20accelerate%20graphics%20rendering.

[5] *What is User Interface (UI)? | indeed.com*. Indeed. (2022, July 22). https://www.indeed.com/career-advice/career-development/user-interface

[6] *What is HUD (heads-up display)?*. Computer Hope. (2017, April 26). https://www.computerhope.com/jargon/h/hud.htm

[7] Kaspersky. (2023, April 19). *White Hat hackers: The good, the bad, or the ugly?* https://www.kaspersky.com/resource-center/definitions/white-hat-hackers

[8] Kaspersky. (2023, May 18). *What is cyber security?*. www.kaspersky.com. https://www.kaspersky.com/resource-center/definitions/what-is-cyber-security

[9] Unity. (n.d.). *GameObject*. Unity. https://docs.unity3d.com/ScriptReference/GameObject.html

[10] Unity. (n.d.). *Monobehaviour*. Unity. https://docs.unity3d.com/ScriptReference/MonoBehaviour.html