National College of Ireland

Configuration Manual

MSc Research Project
MSc Cybersecurity

Faatih Kajogbola
Student ID: x23312131

School of Computing
National College of Ireland

Supervisor: Mr. Joel Aleburu

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Kajogbola Faatih Adekunle		
Student ID:	23312131		
Programme:	MSc. Cybersecurity	**Year:**	2024/2025
Module:	Research in Computing		
Lecturer:	Mr. Joel Aleburu		
Submission Due Date:	11-08-2025		
Project Title:	Practicum Configuration Manual		
Page Count:	23		
Word Count:	1683		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: 

Date: 11-08-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Table of Contents

1. Introduction and Overview

1.1 Purpose and Scope

This configuration manual provides comprehensive guidance for installing, configuring, and running the Unreal Engine 5–based gamified cybersecurity training module, as used in the MSc research study on gamification and knowledge retention. The manual is designed for:

- **Educational Administrators** implementing cybersecurity training programs.
- **IT Trainers and Instructors** conducting cybersecurity education.
- **Researchers** replicating or extending the experimental methodology.
- **Course Designers** adapting the platform for specific organizational needs.

1.2 Platform Overview

The platform incorporates evidence-based gamification elements including:

- **Progressive Learning Modules**: Cybersecurity concepts presented as problem-solving scenarios (e.g., phishing awareness, password security and network basics), mimicking real-world threats.
- **Points**: Earned for answering questions correctly answers, providing feedback and a sense of accomplishment.
- **Immediate Feedback**: Instant corrective feedback upon wrong answers to help track progress and adjust learning strategies.
- **Narrative Scenarios**: A compelling narrative to immerse users and make learning more engaging.
- **Leaderboards**: Used to keep track of user performance and foster healthy competition without discouraging participants.

The design principles are grounded in Self-Determination Theory (SDT) and the Octalysis Framework, aiming to satisfy basic psychological needs such as autonomy, competence, and relatedness, thereby cultivating genuine learning. Images showing the implementation of each these features are shown below:

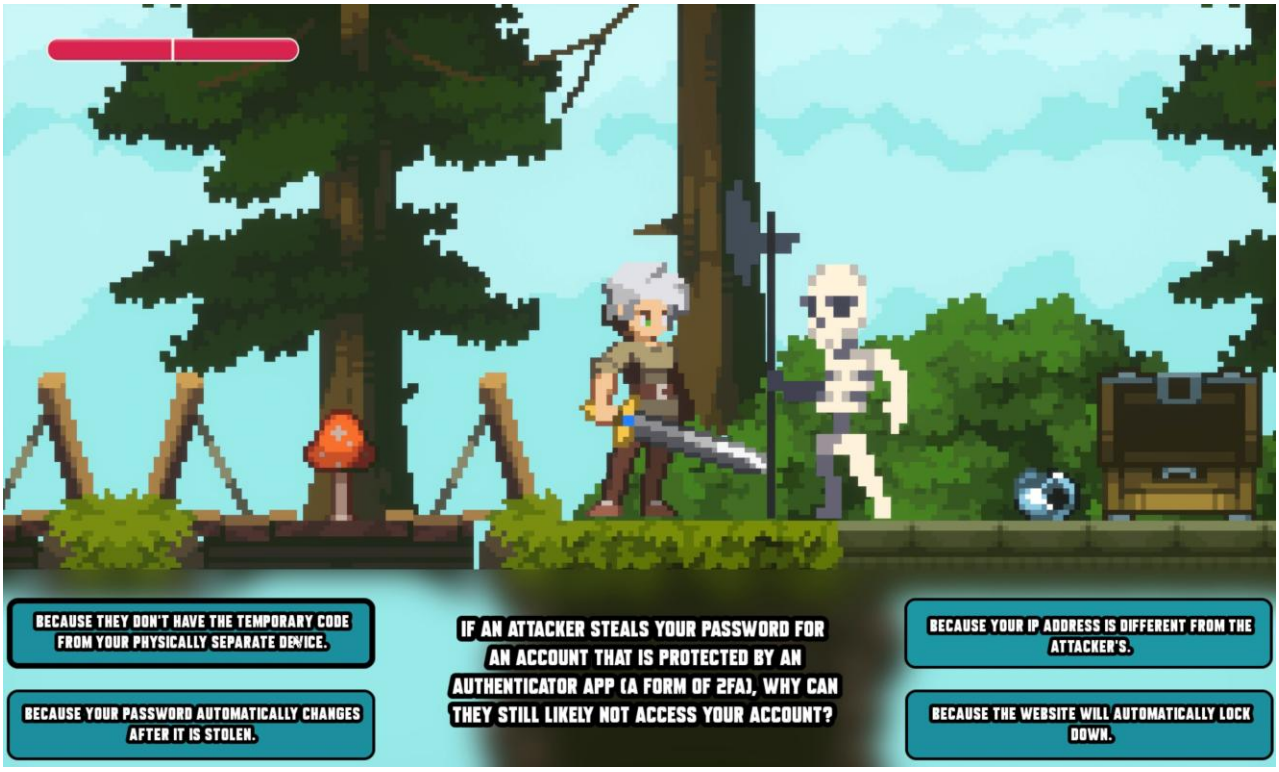


Figure 1: Shows an example of cybersecurity concepts presented as problem-solving scenarios.

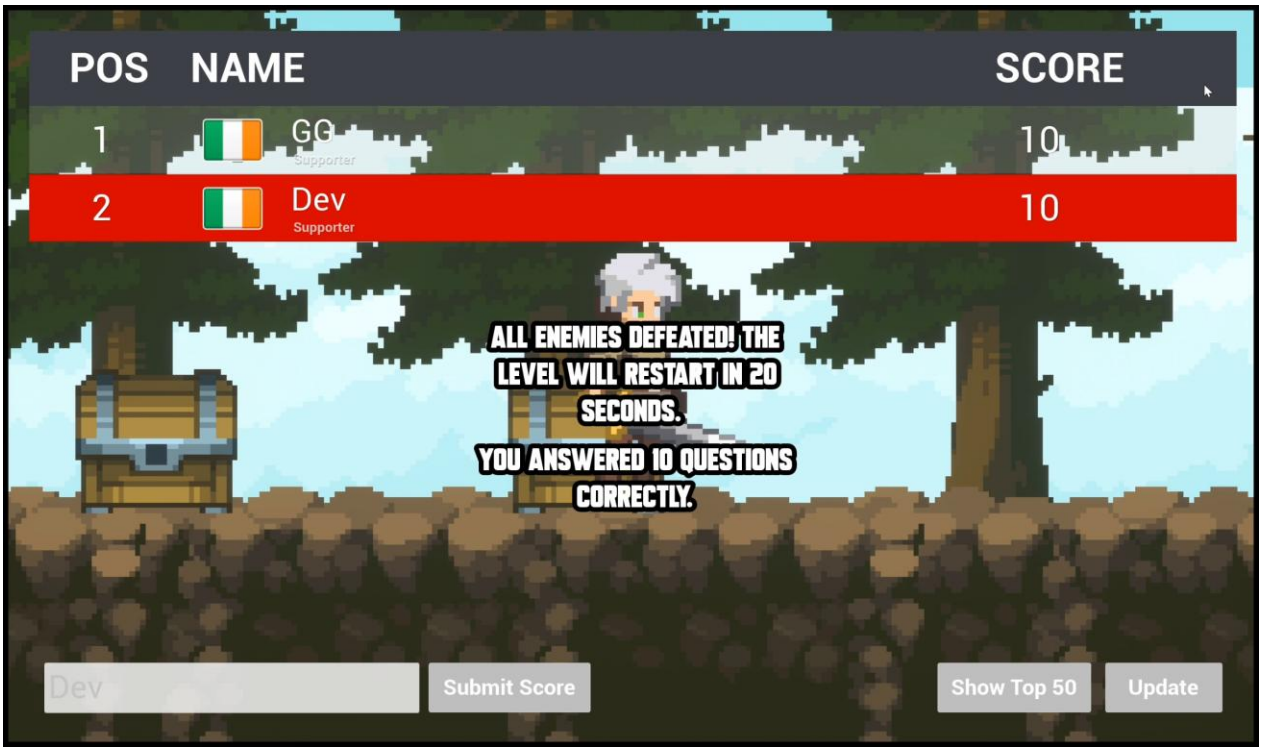


Figure 2: Shows the points obtained from answering questions correctly being entered into the leaderboard.

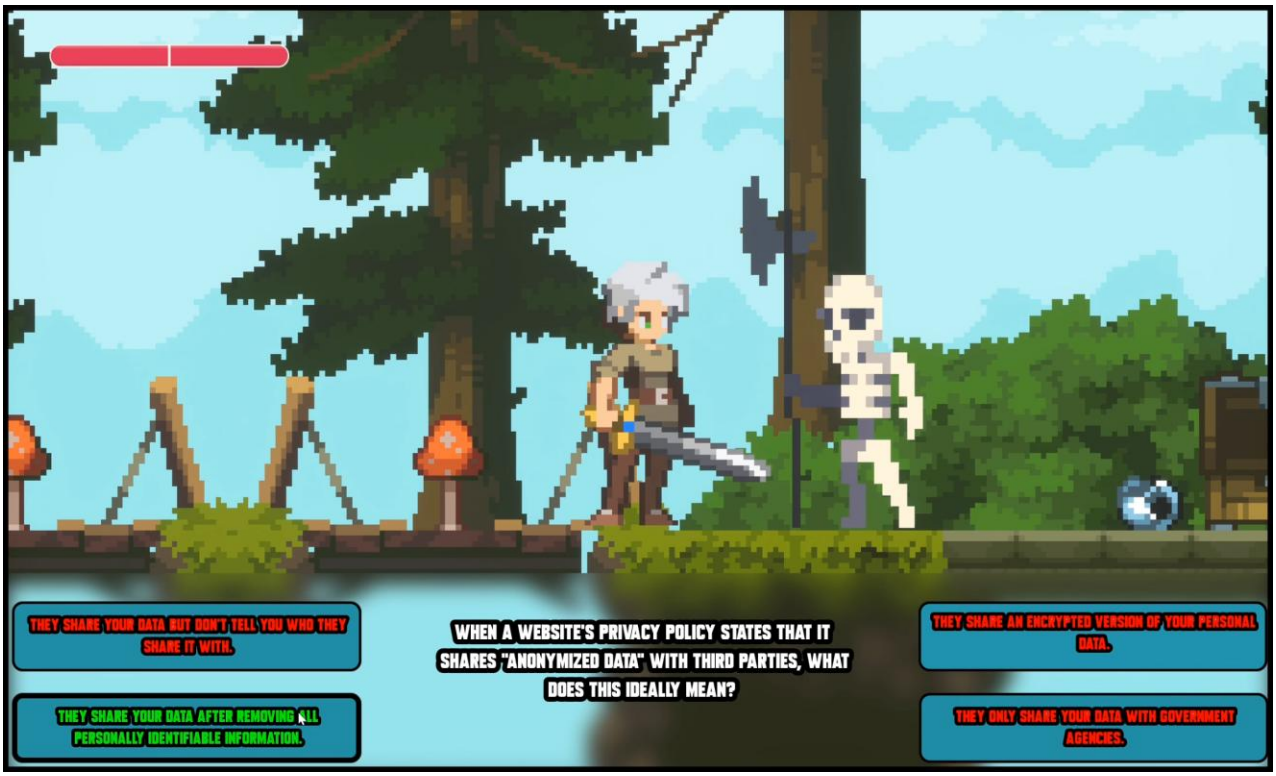
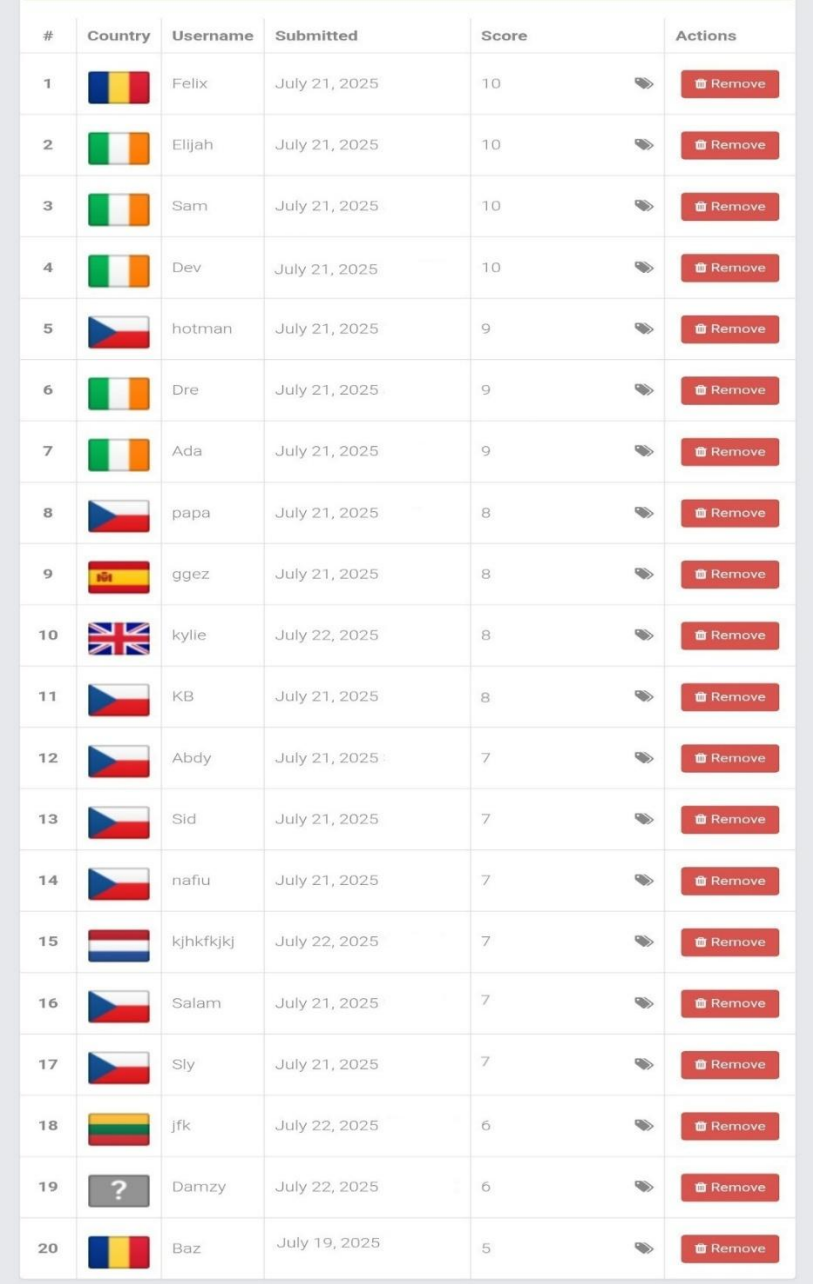


Figure 3: Shows the corrective feedback given for wrong answers.



#	Country	Username	Submitted	Score	Actions
1		Felix	July 21, 2025	10	Remove
2		Elijah	July 21, 2025	10	Remove
3		Sam	July 21, 2025	10	Remove
4		Dev	July 21, 2025	10	Remove
5		hotman	July 21, 2025	9	Remove
6		Dre	July 21, 2025	9	Remove
7		Ada	July 21, 2025	9	Remove
8		papa	July 21, 2025	8	Remove
9		ggez	July 21, 2025	8	Remove
10		kylie	July 22, 2025	8	Remove
11		KB	July 21, 2025	8	Remove
12		Abdy	July 21, 2025	7	Remove
13		Sid	July 21, 2025	7	Remove
14		nafiu	July 21, 2025	7	Remove
15		kjhkfkjkj	July 22, 2025	7	Remove
16		Salam	July 21, 2025	7	Remove
17		Sly	July 21, 2025	7	Remove
18		jfk	July 22, 2025	6	Remove
19	?	Damzy	July 22, 2025	6	Remove
20		Baz	July 19, 2025	5	Remove

Figure 4: Shows the backend of the leaderboard in its entirety.

2. System Requirements and Dependencies

2.1 Minimum System Requirements

Hardware Specifications:

- **Processor**: Intel i5-8400 / AMD Ryzen 5 2600 or equivalent.
- **Memory**: 8 GB RAM minimum (16 GB recommended).
- **Graphics**: DirectX 11 compatible GPU (GTX 1060 / RX 580 or better).
- **Storage**: 15 GB available space.
- **Network**: Broadband internet connection for initial setup and analytics.

Operating System Support:

- Windows 10/11 (64-bit).
- macOS 10.15+ (Intel/Apple Silicon).
- Linux Ubuntu 18.04+ (experimental support).

2.2 Software Dependencies

Required Software:

- Unreal Engine 5.0+ (for developers and researchers who wish to access the source files).
- Microsoft Visual C++ Redistributable 2015 – 2022.
- Microsoft DirectX Runtime.
- Microsoft Visual Studio 2019+ or VS Code 2019+ (for developers and researchers who wish to access the source files).

Optional Tools:

- Git (for version control).
- Blender 3.0+ (for custom asset creation).

2.3 Network and Security Requirements

- **Firewall Configuration**: Allow UE5 through Windows Firewall.
- **Port Requirements**: TCP 7777 for multiplayer features (optional).
- **Data Storage**: Local file system or cloud storage integration.

3. Installation and Initial Setup

Method A: Pre-Built Executable (Recommended for End Users)

1. Open the **ICT Solution Artefacts.zip** file provided and extract the contents of **Final Executable.zip** to the desired location.
2. Navigate to the **Final Executable/Windows** directory and run **CyberKnights.exe .**

Method B: Source Code Deployment (Recommended for Developers and Researchers)

Epic Games Launcher and Unreal Engine Installation

Step 1: Download Epic Games Launcher

1. Visit epicgames.com
2. Download and install the Epic Games Launcher
3. Create Epic Games account or sign in

Step 2: Install and Configure Unreal Engine 5

1. Open Epic Games Launcher
2. Navigate to the "Unreal Engine" tab
3. Click "Install Engine" and select version 5.0 or later
4. Choose the installation directory (minimum 15 GB free space)
5. Complete the installation (installation time may vary depending on bandwidth)
6. Click here to download and install the “PaperZD” Unreal Engine plugin.
7. Click here to download and install the “Epic Leaderboard” Unreal Engine plugin.

Platform Deployment

1. Open the **ICT Solution Artefacts.zip** file provided and extract the contents of **Source Files.zip** to the desired location.
2. Navigate to the **Source Files/Thesis Deliverable** directory and open **Thesis.uproject .**
3. Since this is the very first build, it will take a few minutes to compile the program files. Once the process is complete, open **Thesis.uproject** from the directory **main/Thesis** in Unreal Engine 5.
4. After this, you may choose to run the project in either of 2 ways:
    - Click on the green play icon located near the top left corner to launch the project within the editor.

- Click on the “Platforms” section near the top left corner to package the project into a standalone executable file.

Figures 5 & 6 below show both ways to run the project:

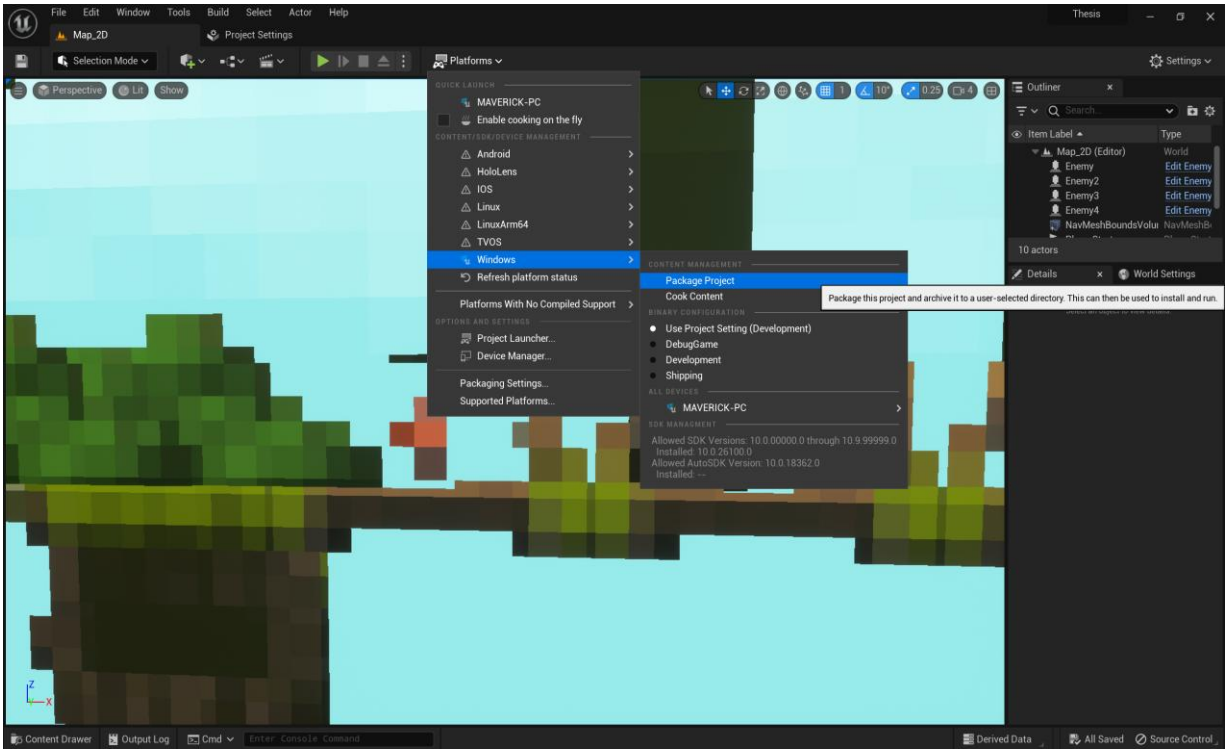


Figure 5: Shows how to package the project for the selected operating system.

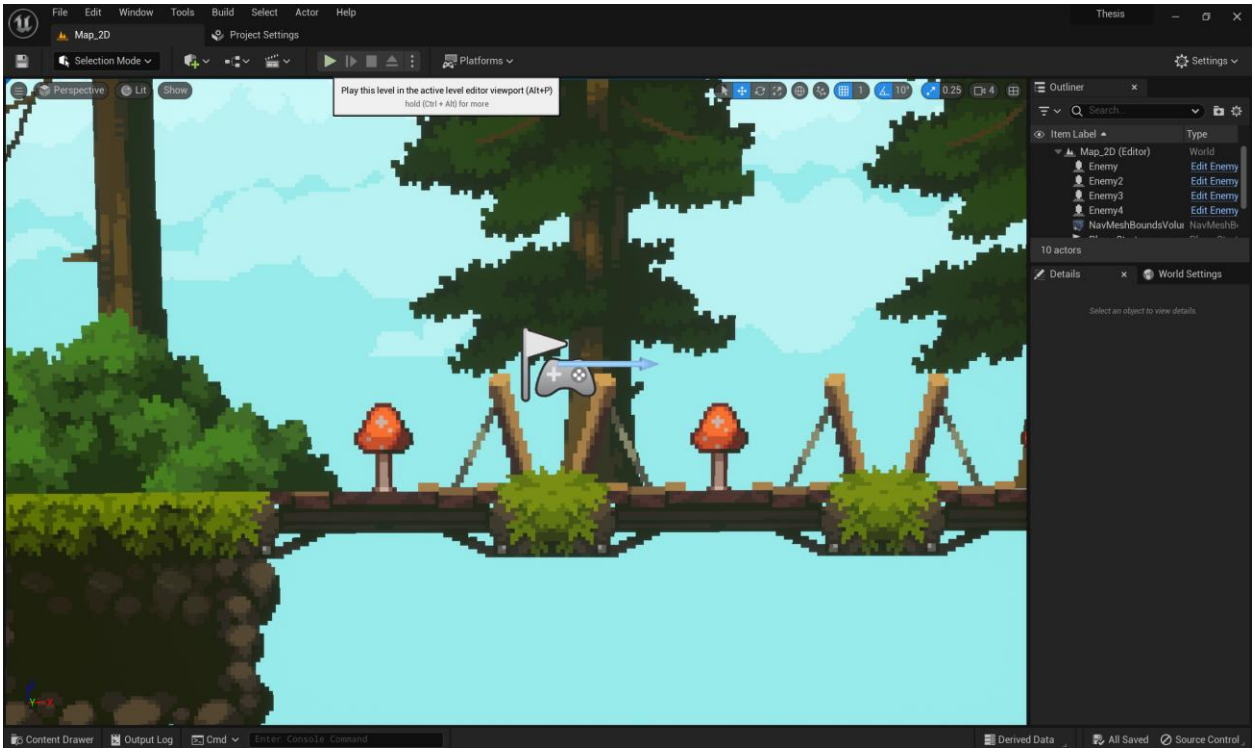


Figure 6: Shows how to run the project within the editor.

4. Platform Configuration

4.1 Configuring Core Gamification Elements

4.1.1 Game Mode Logic

The Game Mode serves as the backbone of the entire gamification process. It is essentially the central hub for high-level logic, which integrates all the variables, functions and events that control the way an Unreal Engine project operates. The images below show all the configurable variables, functions and events created for this class.

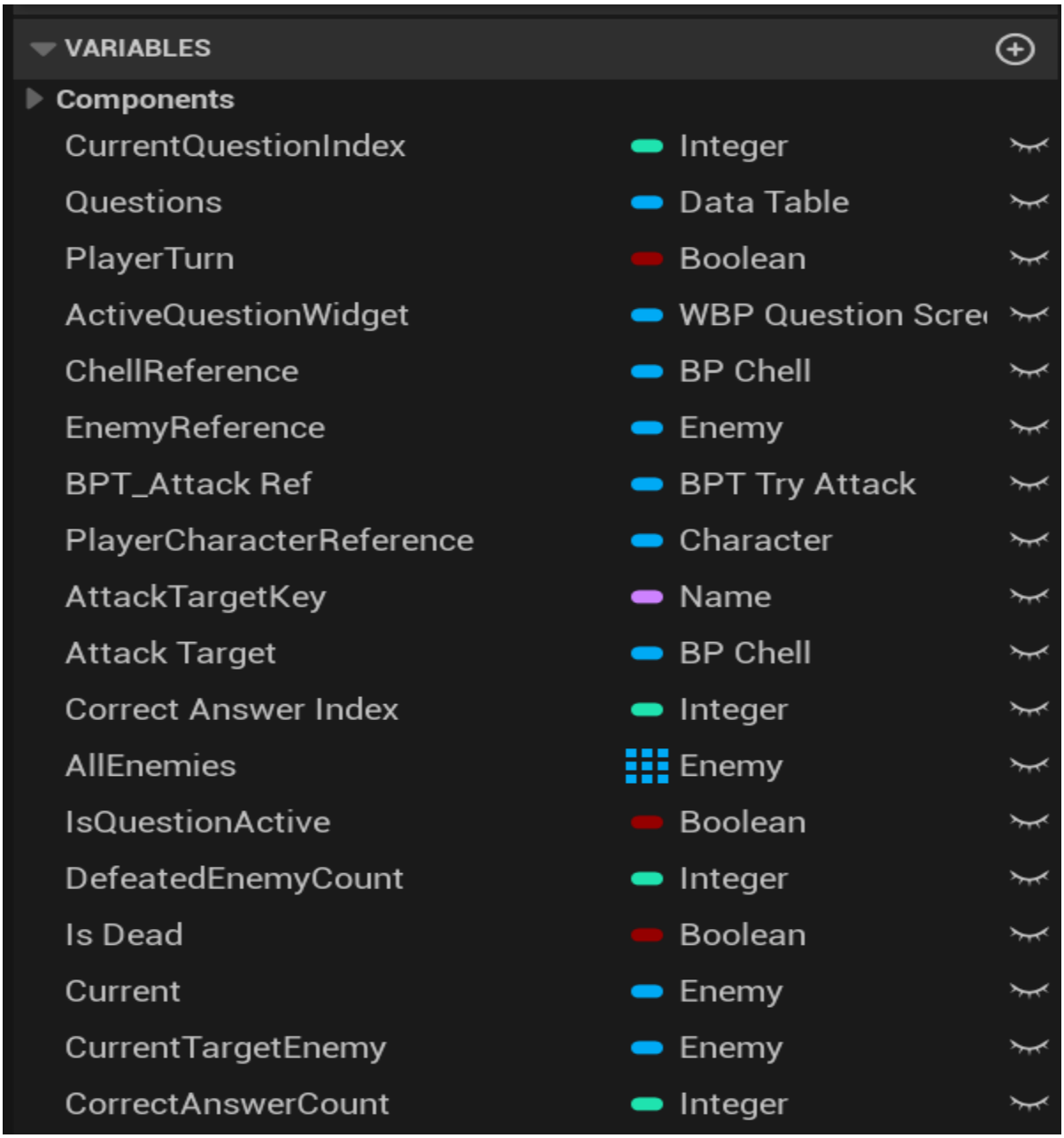


Figure 7: List of variables used in Game Mode logic and their data types.

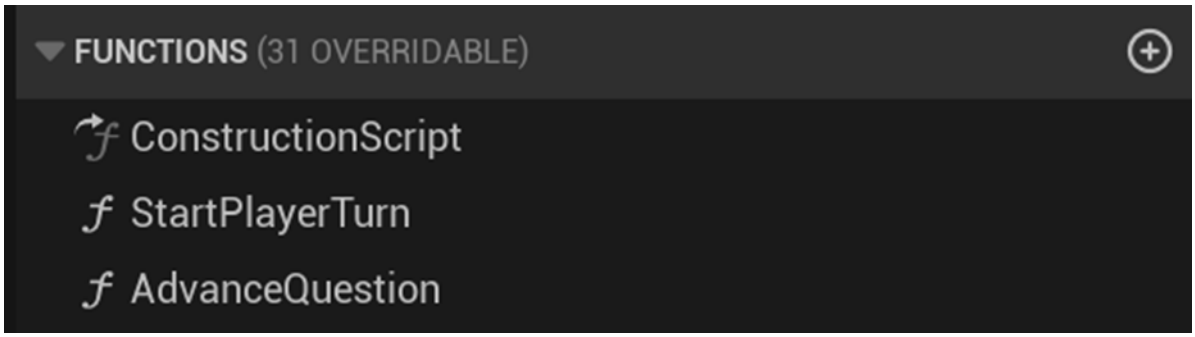


Figure 8: List of functions created to control logic flow in Game Mode.

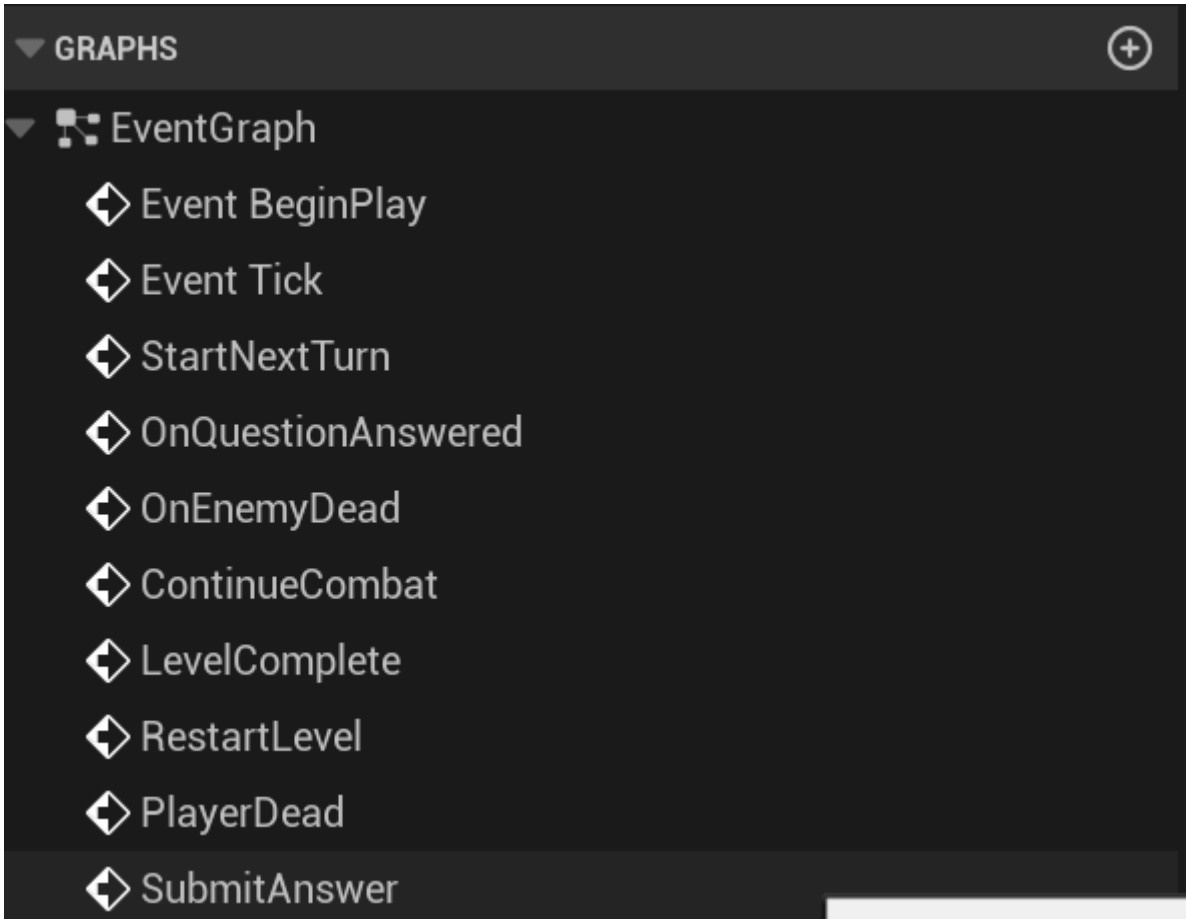


Figure 9: List of events created to control logic flow in Game Mode.

For a better understanding of how all the variables, functions and events come together to create the complex logic that drives the project, the Game Mode file can be found in the directory: **Content/Blueprint/GM_Beta.**

4.1.2 Player Character Logic

The player character logic is controlled by a character blueprint component which integrates a movement component and an animation component. It serves as the physical representation of

the player, allowing them to interact with the module. All configurable components, variables, functions and events created for this class are shown in the images below.

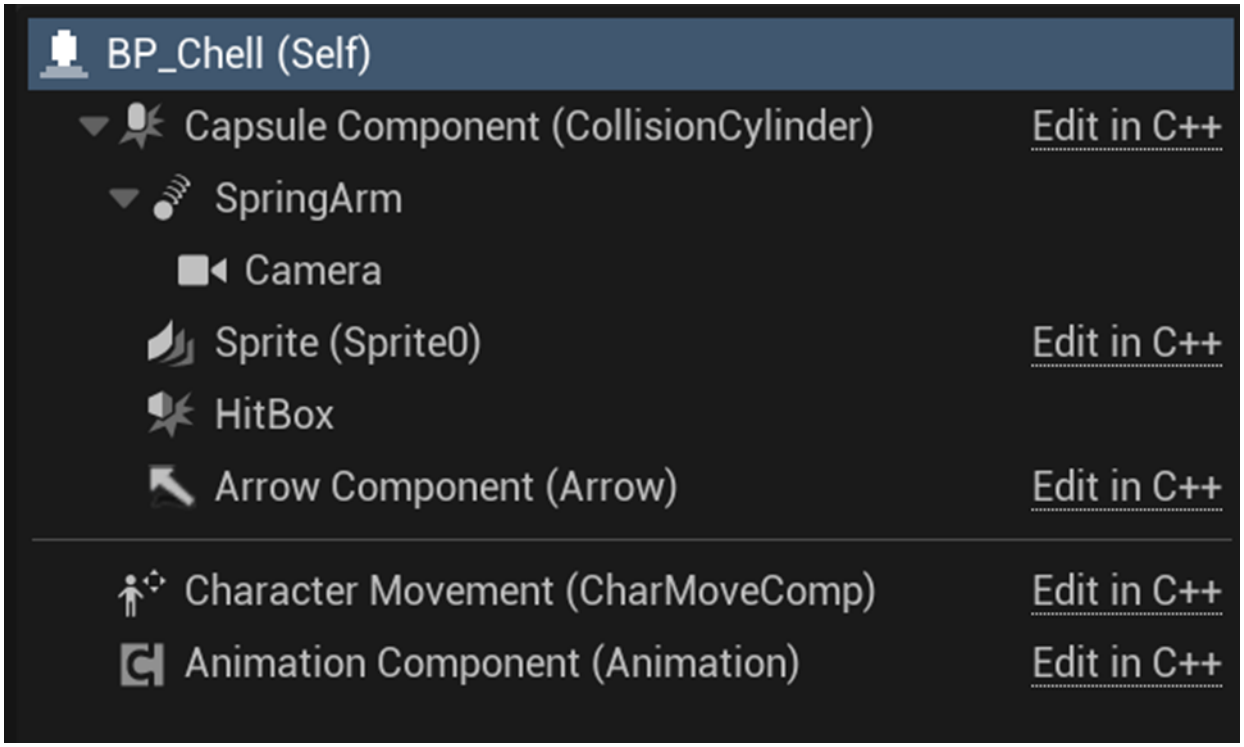


Figure 10: List of components attached to the player character.

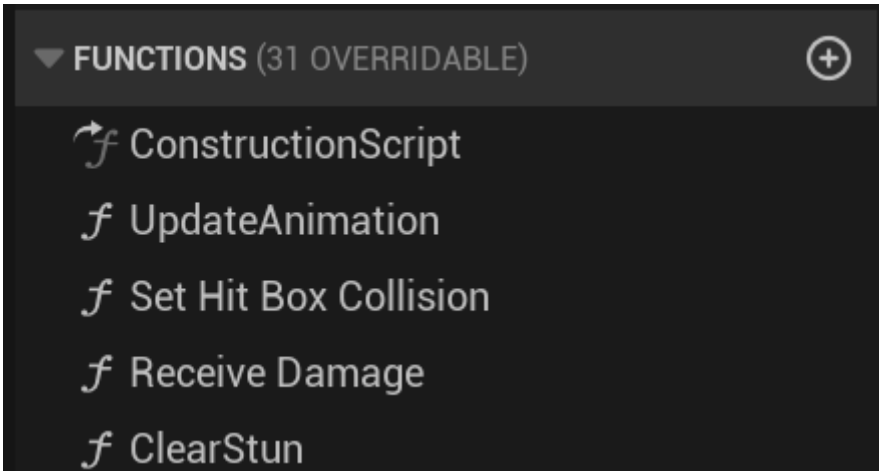


Figure 11: List of functions created to control the player character's logic.

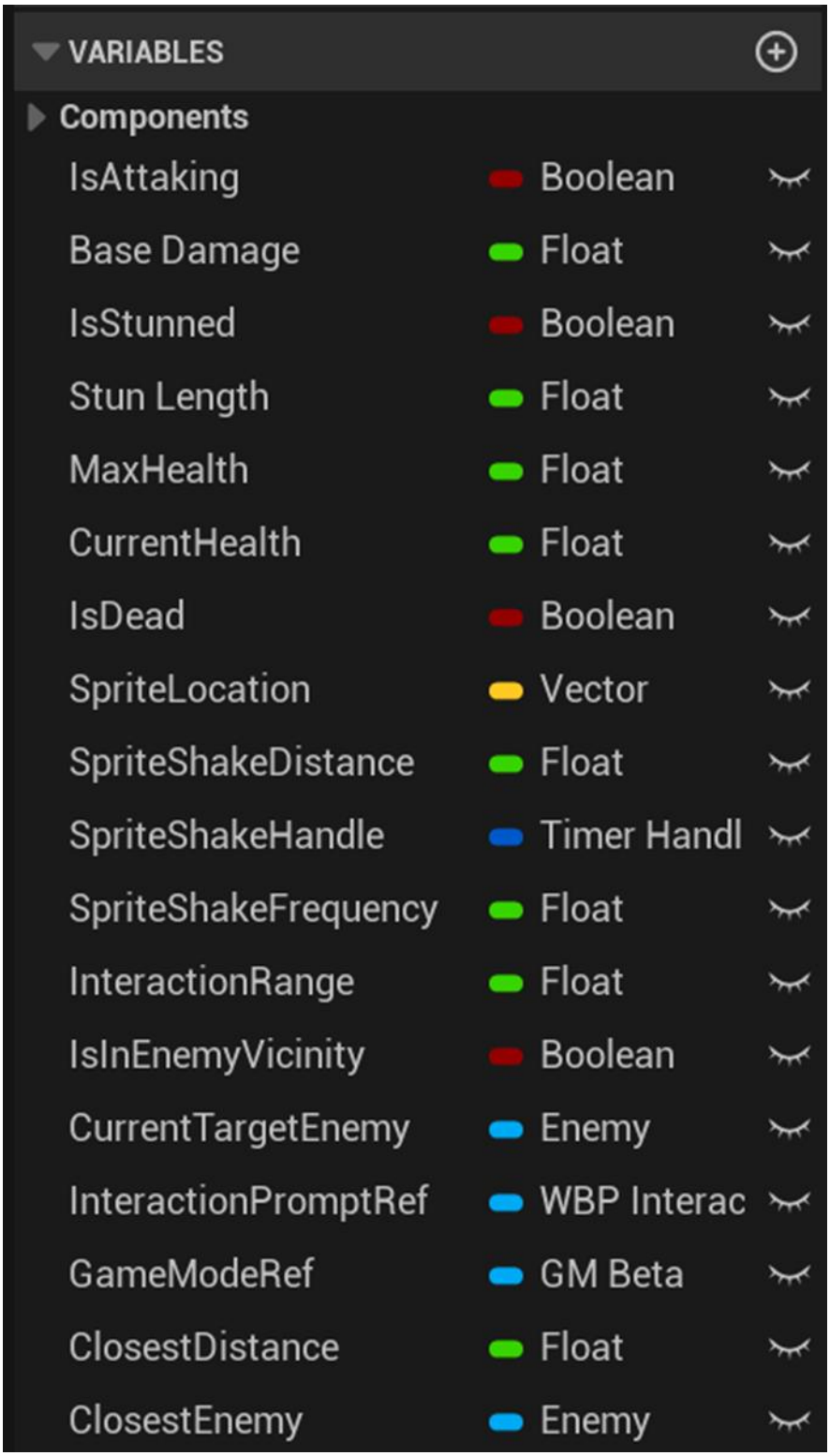


Figure 12: List of variables used to implement player character logic and their data types.

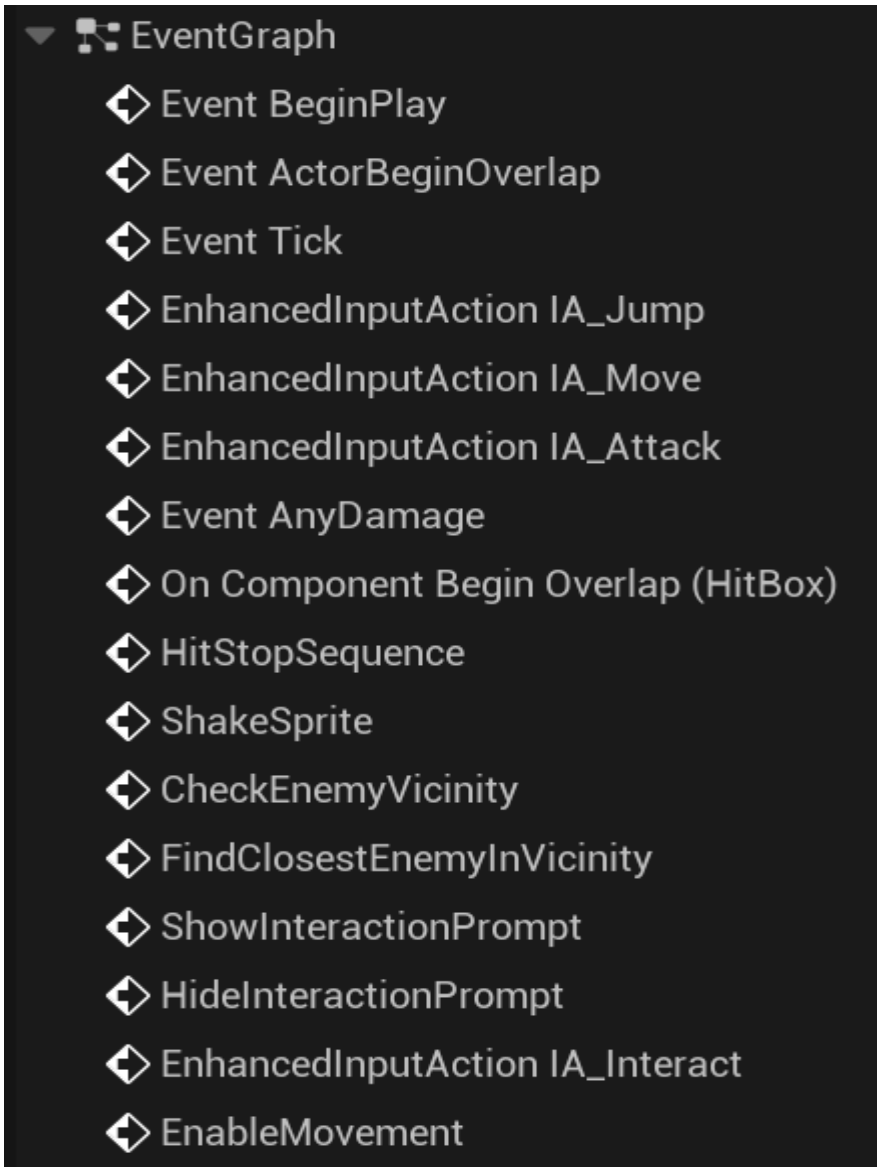


Figure 13: List of functions created to control the player character's logic.

For a more in-depth look at the player character logic, the player character class can be found in the directory: **Content/Blueprint/Chell/BP_Chell**.

4.1.3 Enemy Logic

The enemy class was created using the same base components as the player character, but AI controller and blackboard components were added to add some autonomy to the class. All configurable components, variables, functions and events created for this class are shown in the images below.

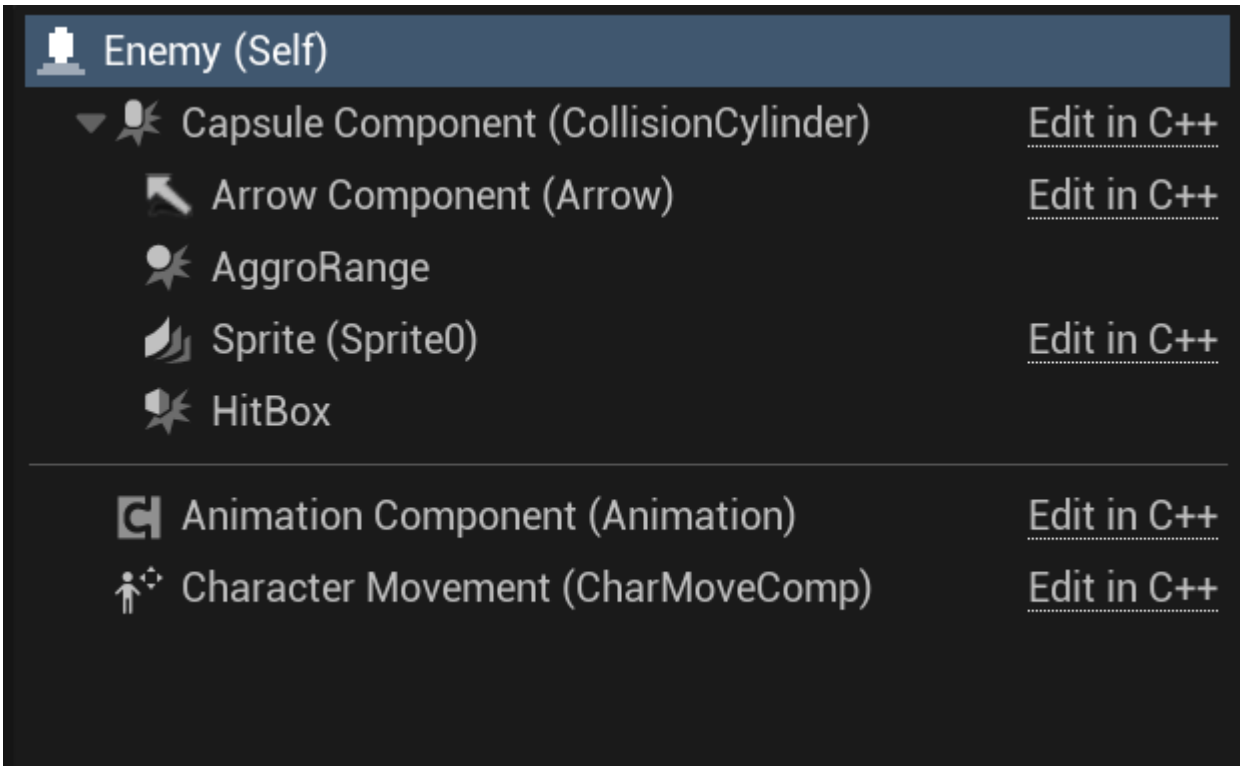


Figure 14: List of components attached to the enemy character.

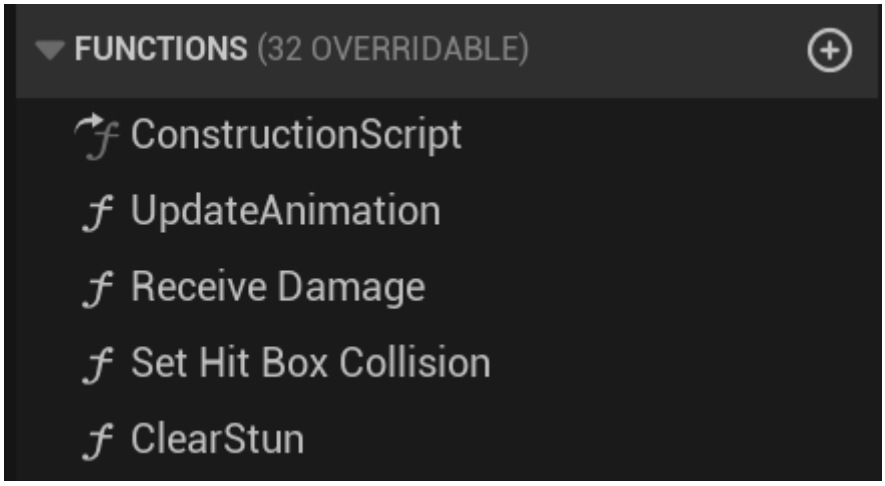


Figure 15: List of functions created to control the enemy's logic.

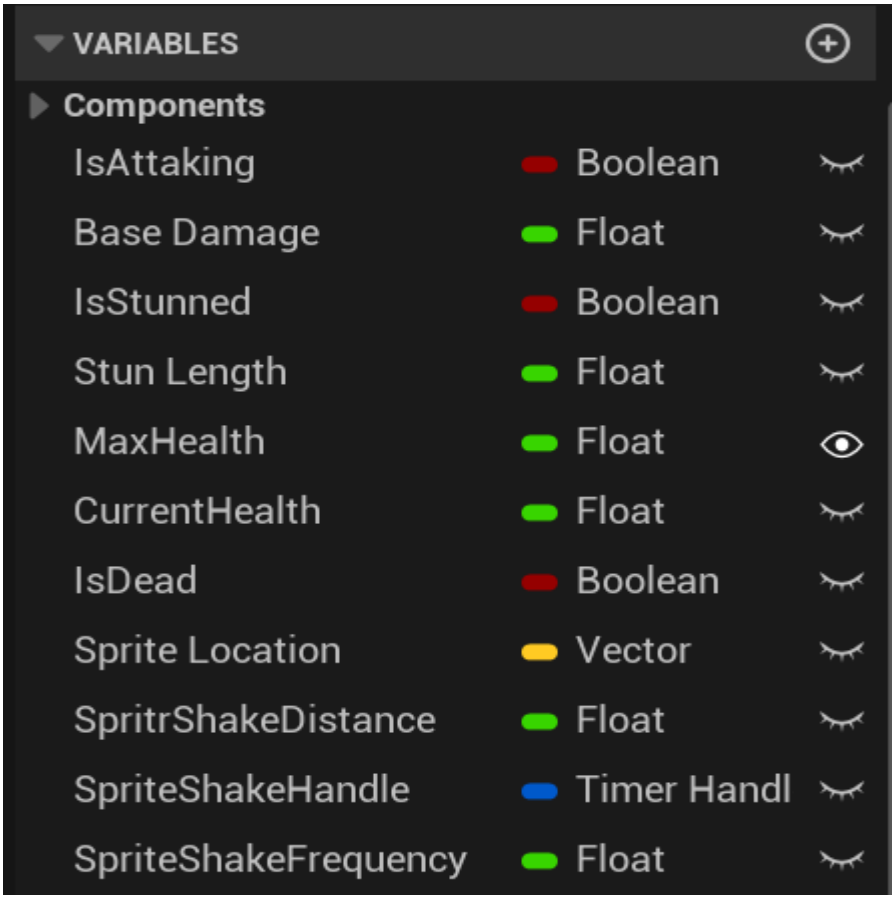


Figure 16: List of variables used to implement enemy logic and their data types.

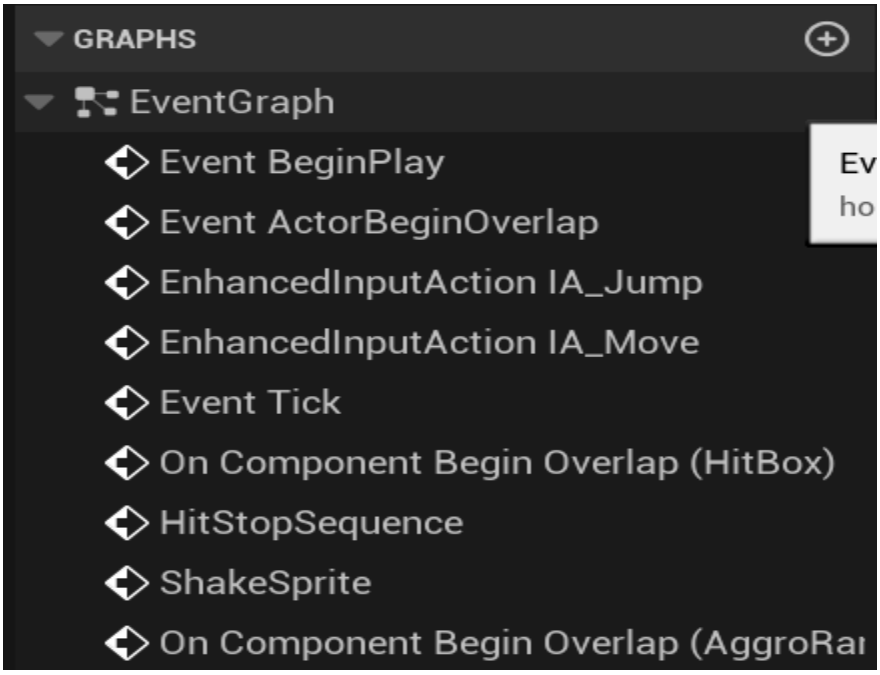


Figure 17: List of events created to control the enemy's logic.

For a more in-depth look at the enemy's logic, the enemy class can be found in the directory: **Content/Blueprint/Enemy/Enemy**.

4.2 Configuring Quiz Questions

The quiz questions were implemented by using a configurable data table tied to a blueprint struct class. The data table essentially functions as a spreadsheet containing the questions, while the struct functions as a template for what a single question's data should look like, defining the columns and their data types. The blueprint struct featured three unique elements:

- The body of the questions, which was of the data type text.
- The answer choices from A to D, which were an array of the data type text.
- The array index representing the correct answer choice, which was of type integer.

New questions can be added to the data table at any time by simply adding a new row and populating the needed columns. Each of these unique elements and how they were populated in the data table for all the different questions are shown below:

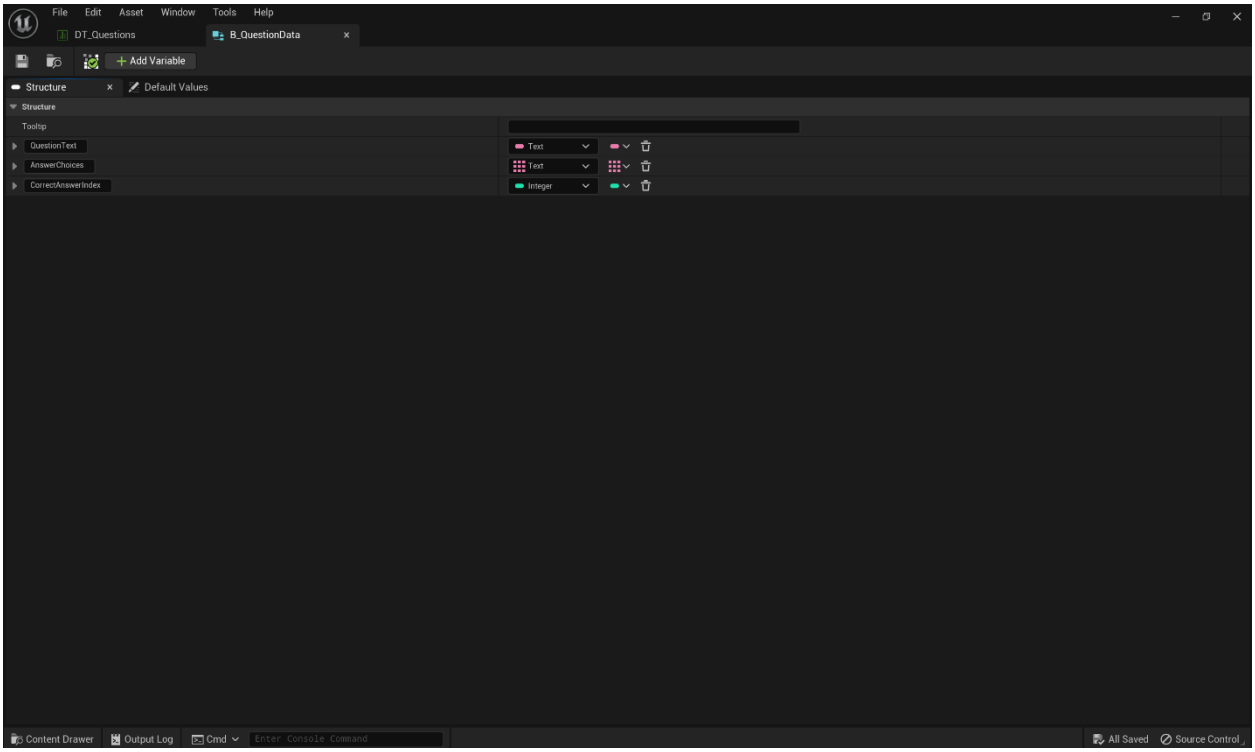


Figure 18: Blueprint struct "B_QuestionData" containing unique elements.

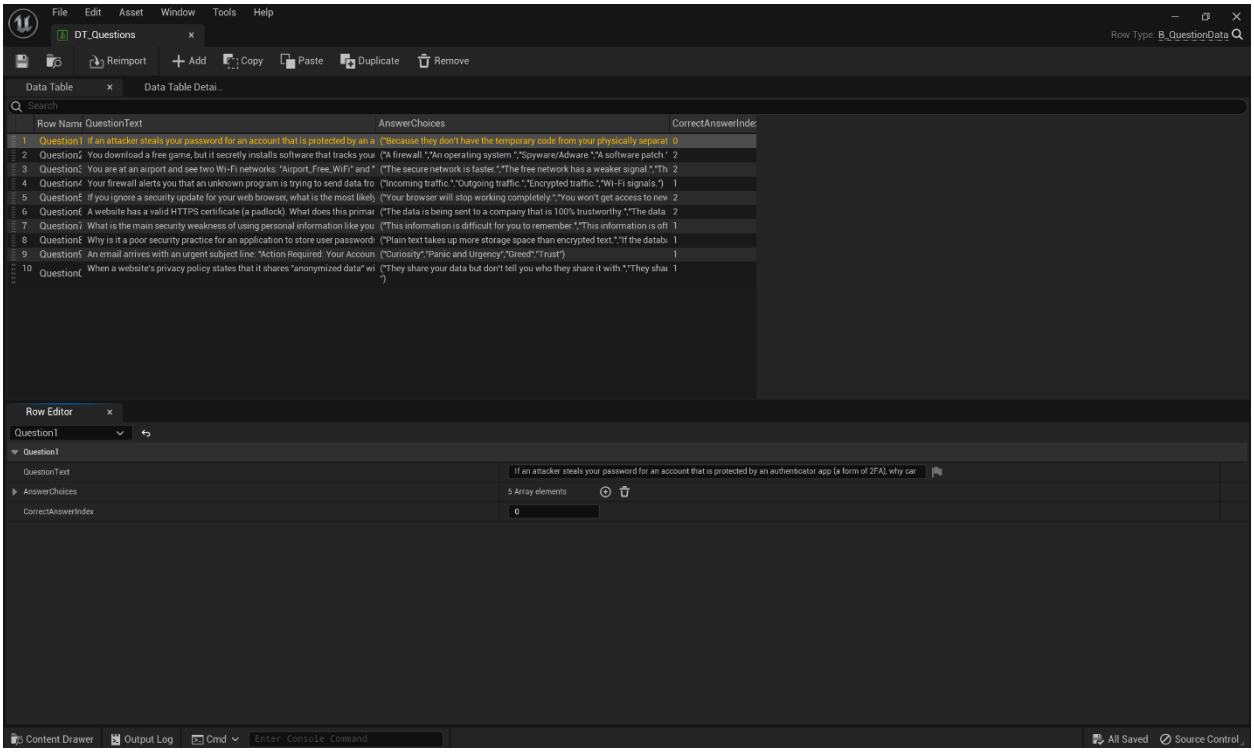


Figure 19: Data table “DT_Questions” showing the 10 configured questions used in the project.

5. Advanced Configuration and Customization

5.1 Adjusting Gameplay Settings and Parameter

The gamified module allows modifying core operational parameters, such as overall difficulty levels, scoring thresholds, frequency of feedback, or other configurable game rules. These adjustments allow developers and researchers to better tailor the learning experience to specific audiences or educational objectives.

Configuring these settings requires editing the variables, functions and events shown in the previous section from the directories provided. However, it is recommended to make a backup of these files before attempting to change too many parameters.

5.2 Modifying User Interface Elements

The gamified module also allows customizing the multiple UI elements used in the project such as changing icons or updating specific instructional text strings displayed within the UI. Modifying these files requires navigating to: `Content/Blueprint/HUD` which contains all the elements related to the UI. The images below show the different configurable UI elements:

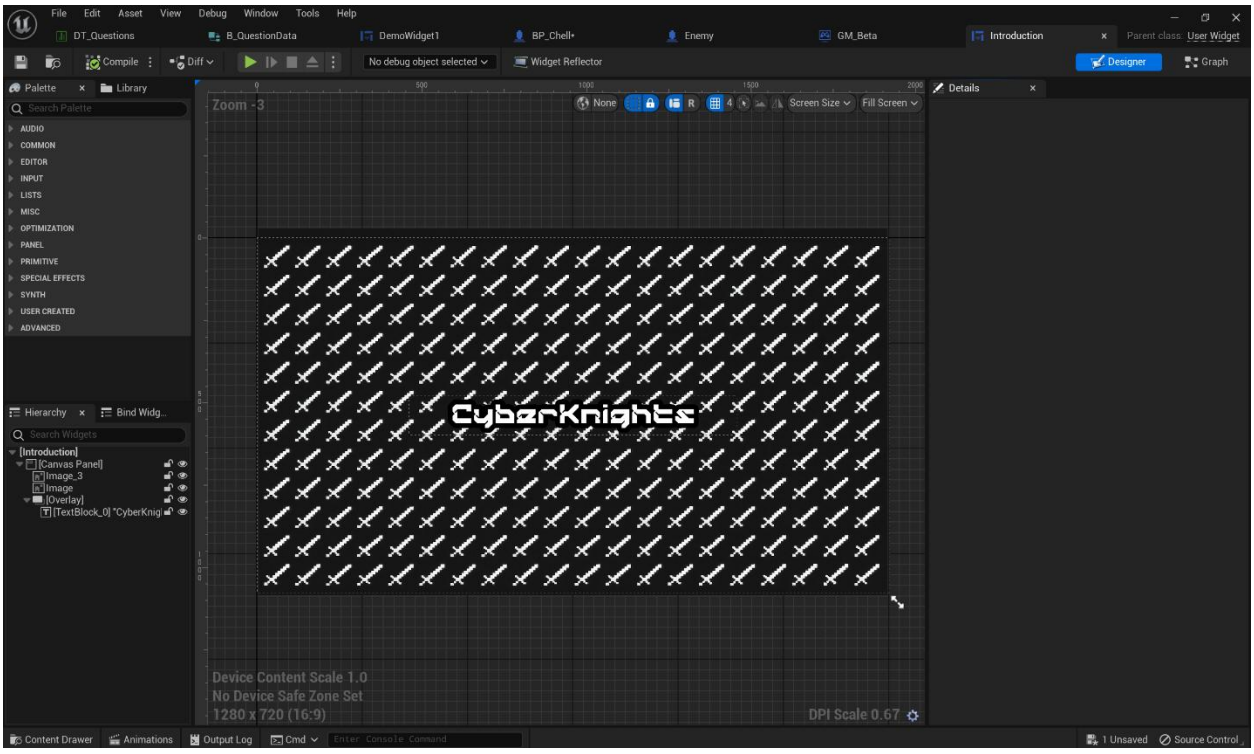


Figure 20: The introductory splash screen shown on startup.

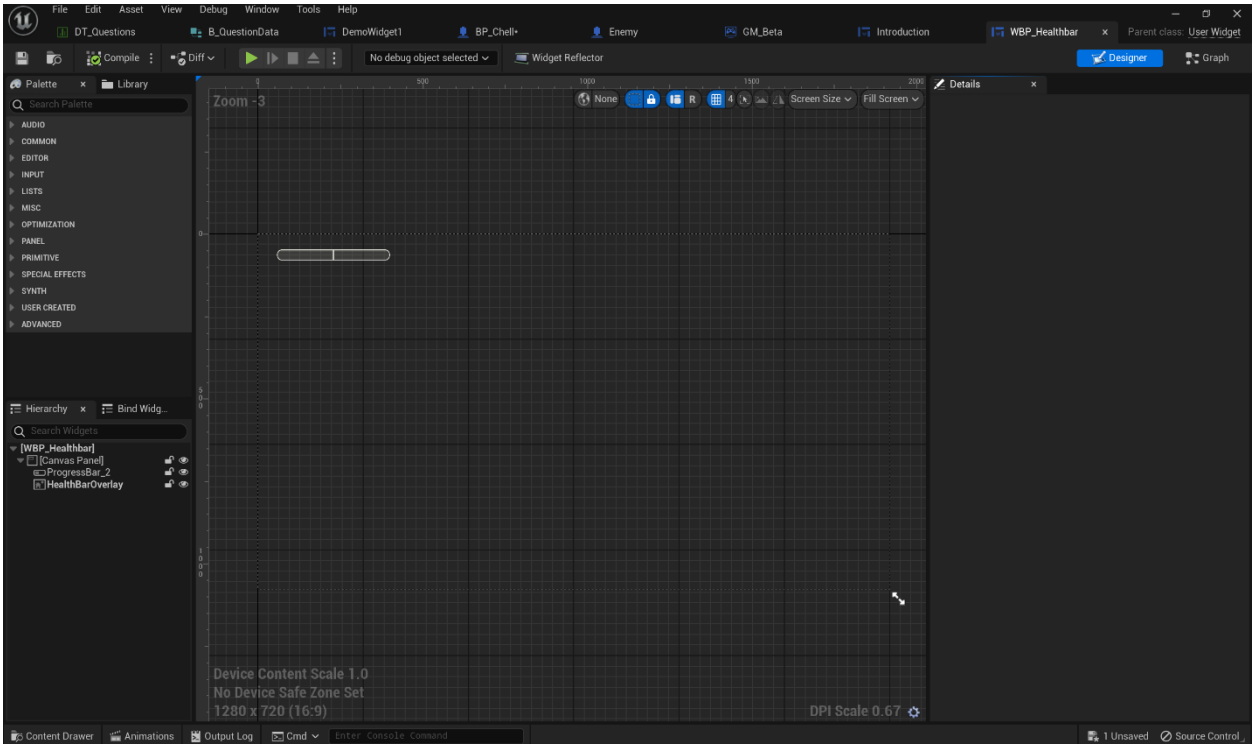


Figure 21: The player character's health bar.

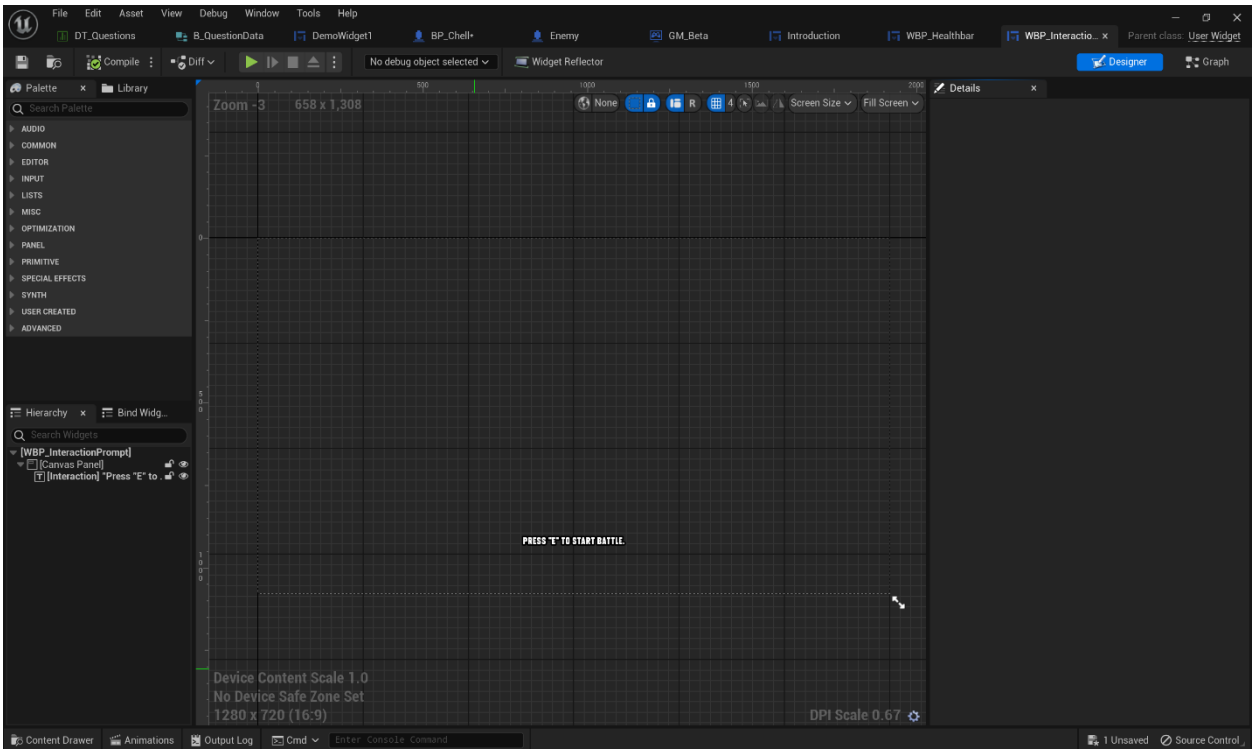


Figure 22: The interaction prompt shown when an enemy is within range of the player character.

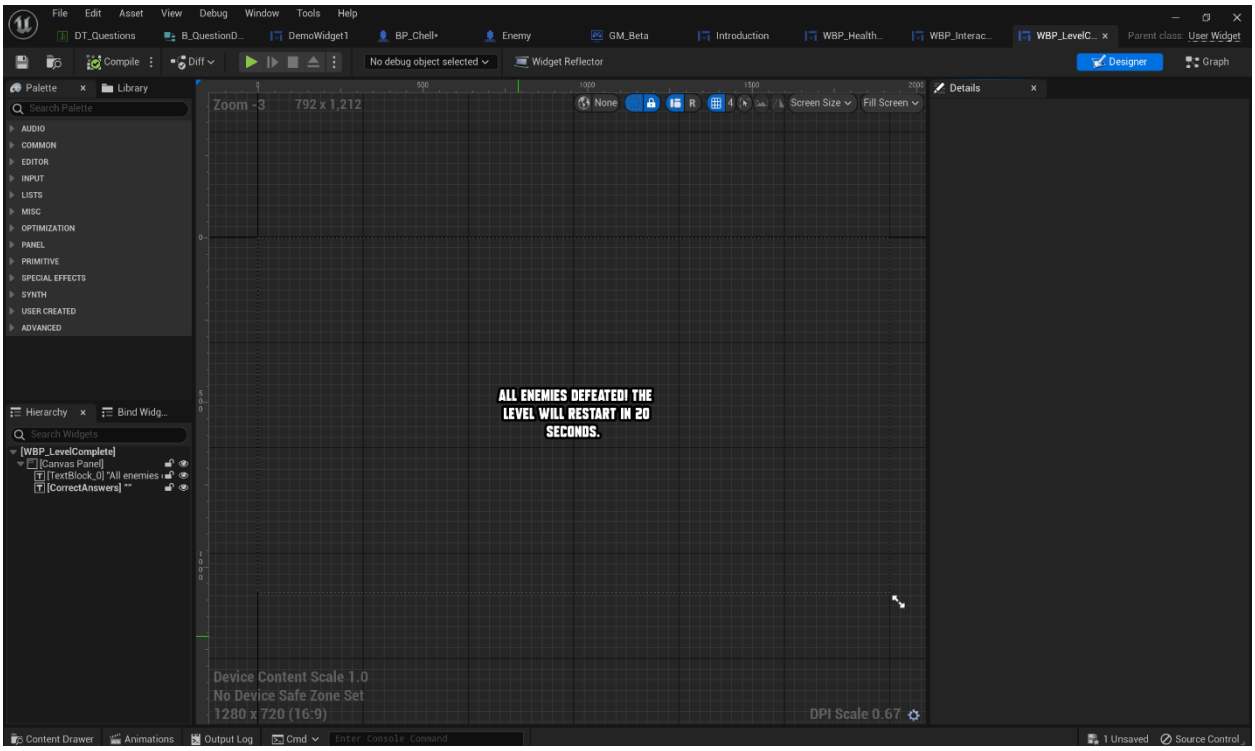


Figure 23: The message shown upon successfully completing the level.

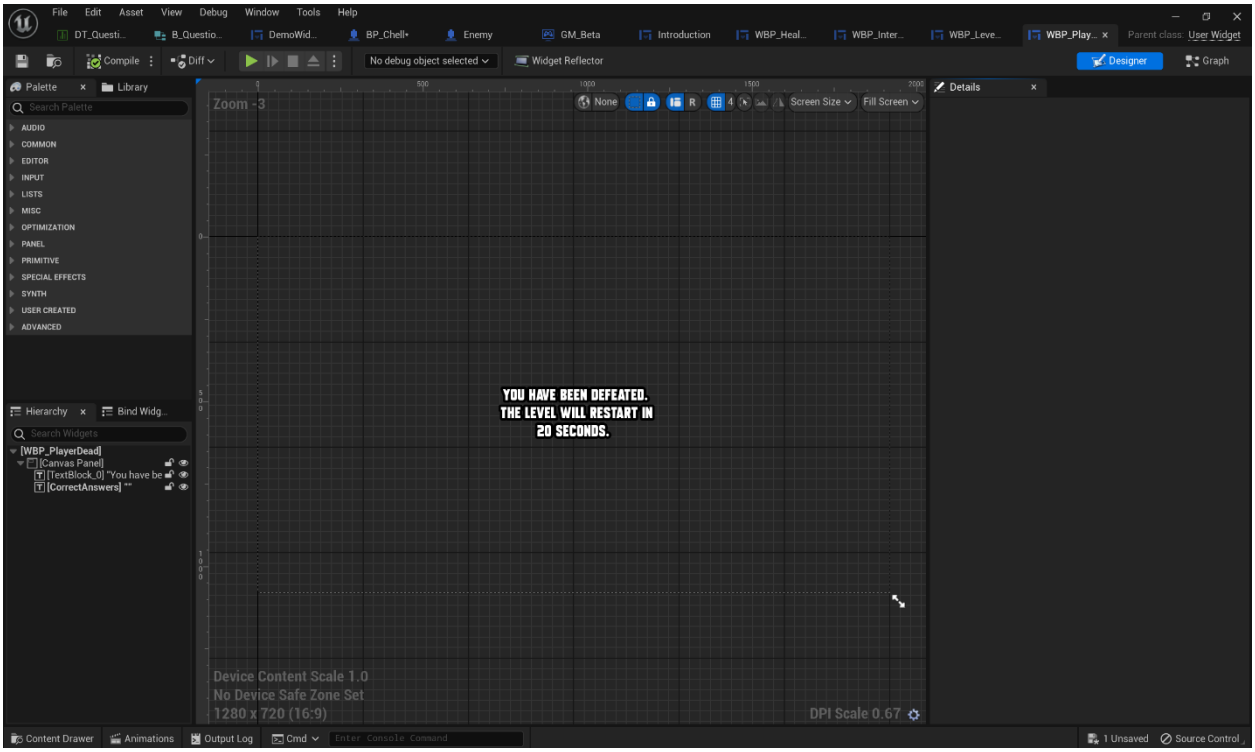


Figure 24: The message shown upon failing the level.

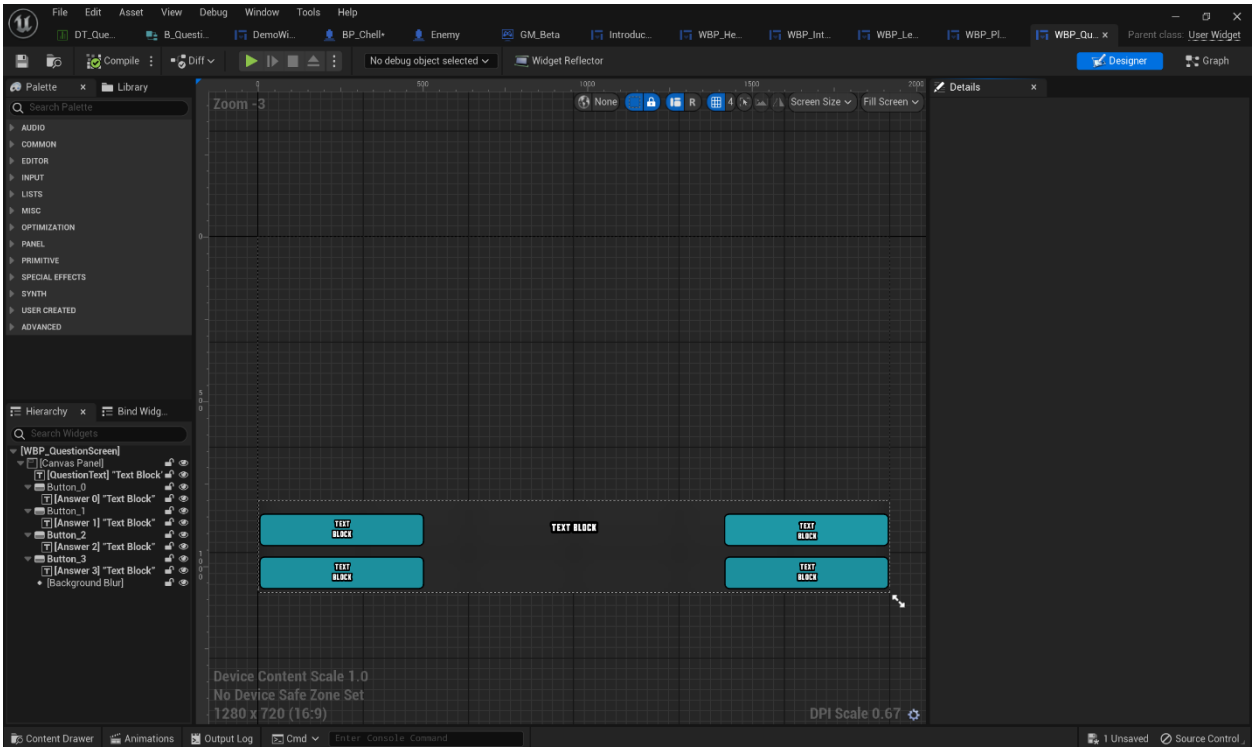


Figure 25: The UI overlay which displays the current questions and available answers.

6. Maintenance and Troubleshooting

Common Issues and Solutions

Effective troubleshooting is essential for maintaining the operational integrity and user satisfaction of the gamified learning module. This section outlines common issues that may occur and provides possible solutions.

Issue	Possible Cause	Solution
Application fails to launch	Missing or outdated Microsoft Visual C++ Redistributable runtime libraries	Verify that the Microsoft Visual C++ Redistributable for Visual Studio 2015-2022 (x64 version) is installed on the system. If not, download and install it from Microsoft's official website.
Application crashes on launch or exhibits graphical glitches, particularly on older systems.	DirectX compatibility issues or outdated graphics drivers.	Ensure graphics drivers are up-to-date.
Quiz questions or learning scenarios appear blank, incomplete, or display incorrect information.	Incorrect format or parsing errors in external CSV/JSON content files.	Ensure no missing commas or empty cells.
External content (e.g., custom logos, specific text strings) is not displayed by the packaged build.	External content files are not found by the packaged build.	Confirm that external files are placed in the correct relative paths within the installation directory.
Leaderboard not updating	Internet connection issue	Check network access and firewall settings.
The application cannot read or write external data files.	File permission issues.	Try running the application with as an administrator. Alternatively, relocate the external files to a less restricted directory, such as a user's Documents folder.