Documentation

for

Triangle Cyber

Version <2.5>

Prepared by

Group: Generic Game dev group

|  |  |  |
| --- | --- | --- |
| B2111967 | Lưu Hoài Vũ | Team leader |
| B2105727 | Nguyễn Quang Vinh | Vice of team leader |
| B2111947 | Trịnh Thanh Sang | Member |
| B2111885 | Hà Quốc Huy | Member |

|  |  |
| --- | --- |
| Instructor: | Phan Phương Lan |
| Course: | Introduction to Software Engineering |

Table of Contents:

[Part A. Team Work Plan 5](#_Toc422174853)

[1. Group Organization 5](#_Toc326787164)

[2. Plan 5](#_Toc1589294719)

[3. Rules on the group 6](#_Toc1287661581)

[Part B. Software Requirement Specification 7](#_Toc1582258030)

[1. Introduction 7](#_Toc786091591)

[1.1. Purpose 7](#_Toc896483422)

[1.2. Product Scope 7](#_Toc946042647)

[1.3. Definitions, Acronyms and Abbreviations 8](#_Toc1486237566)

[1.4. References 8](#_Toc616863951)

[2. Overall Description 9](#_Toc506882747)

[2.1. Product Perspective 9](#_Toc163542512)

[2.2. Product Functions 9](#_Toc1277689558)

[2.3. User Classes and Characteristics 10](#_Toc1277520346)

[2.4. Operating Environment 10](#_Toc719810048)

[2.5. Design and Implementation Constraints 11](#_Toc1068583285)

[2.6. Assumptions and Dependencies 11](#_Toc705673001)

[3. Requirements Specification 12](#_Toc753108196)

[3.1. External Interface Requirements 12](#_Toc1643986881)

[3.1.1. User Interfaces 12](#_Toc1280615740)

[3.1.2. Hardware Interfaces 13](#_Toc679944824)

[3.1.3. Software Interfaces 13](#_Toc842877020)

[3.1.4. Communications Interfaces 14](#_Toc1942824085)

[3.2. Functional Requirement 15](#_Toc1575057160)

[3.2.1. Use Case 1 (Starting selection) 15](#_Toc22832043)

[3.2.2. Use Case 2 (Master setting): 17](#_Toc754993692)

[3.2.3. Use Case 3 (Gameplay screen): 19](#_Toc707133690)

[3.2.4. Use Case 4 (Pop-up): 21](#_Toc442840687)

[3.3. Nonfunctional Requirements 22](#_Toc916873070)

[3.3.1. Performance 22](#_Toc647898814)

[3.3.2. Reliability 22](#_Toc1915116117)

[3.3.3. Safety and Security 23](#_Toc1339047923)

[3.3.4. Adaptability and Portability 23](#_Toc974685978)

[3.4. Business Rules 23](#_Toc1356927189)

[3.5. Other Requirements 24](#_Toc479225856)

[Part C. Software Design 25](#_Toc409460360)

[1. Introduction 25](#_Toc2143018780)

[1.1. Purpose 25](#_Toc1375709278)

[1.2. Product Scope 25](#_Toc1355503007)

[1.3. Definitions, Acronyms and Abbreviations    26](#_Toc1481772698)

[1.3.1. References 26](#_Toc1992573229)

[2. Application Architecture  27](#_Toc1862385755)

[2.1. Application Architecture 27](#_Toc1645315210)

[2.2. Description             27](#_Toc1122779139)

[3. Data Design  28](#_Toc992422453)

[3.1. Data Description   28](#_Toc217641610)

[3.1.1. Conceptual Data Model  28](#_Toc43878776)

[3.1.2. Logical Data Model  29](#_Toc1698095454)

[3.1.3. Physical Data Model        29](#_Toc970749807)

[3.2. Data Dictionary 30](#_Toc1687865657)

[4. Detailed Design 33](#_Toc831227546)

[4.1. Starting Selection: 33](#_Toc1650694631)

[4.1.1. Menu screen: 33](#_Toc383259029)

[4.1.2. Level selection and move to the sets of level: 35](#_Toc626567983)

[4.2. Master setting: 37](#_Toc1078268143)

[4.3. Gameplay Screen: 39](#_Toc406091072)

[4.3.1. Play Screen: 39](#_Toc1381561676)

[4.3.2. in-game setting: 41](#_Toc1785401834)

[4.4. Pop-up: 43](#_Toc848931760)

[4.4.1. Success pop-up 43](#_Toc150951098)

[4.4.2. Failure pop-up 45](#_Toc285817000)

[5. References to Requirements  46](#_Toc616564229)

[Part D. Unit Test 47](#_Toc1489999021)

[1. Introduction 47](#_Toc1260502978)

[1.1. Purpose 47](#_Toc1973491418)

[1.2. Definitions, Acronyms and Abbreviations      47](#_Toc1969224877)

[1.3. References 47](#_Toc1669963338)

[2. Test Cases  48](#_Toc1969026550)

[2.1. Test 1 - Level selection 48](#_Toc1197450507)

[2.1.1. Menu screen: 48](#_Toc877982698)

[2.1.1.1. Drawing the flow graph 48](#_Toc1303315601)

[2.1.1.2. Determining a basis set of independent paths 49](#_Toc1042540088)

[2.1.1.3. Preparing test cases 49](#_Toc592884805)

[2.1.2. Level selection and move to the sets of level: 50](#_Toc801147163)

[2.1.2.1. Drawing the flow graph 50](#_Toc17835579)

[2.1.2.2. Determining a basis set of independent paths 51](#_Toc1585307258)

[2.1.2.3. Preparing test cases 51](#_Toc1018788774)

[2.2. Test 2 - Master setting    52](#_Toc61714355)

[2.2.1. Drawing the flow graph 52](#_Toc1135919064)

[2.2.2. Determining a basis set of independent paths 53](#_Toc1989538581)

[2.2.3. Preparing test cases 54](#_Toc1749580012)

[2.3. Test 3 - Gameplay screen and in-game setting 55](#_Toc1967146610)

[2.3.1. Gameplay screen 55](#_Toc1492749564)

[2.3.1.1. Drawing the flow graph 55](#_Toc2132839041)

[2.3.1.2. Determining a basis set of independent paths 56](#_Toc446230946)

[2.3.1.3. Preparing test cases 56](#_Toc423534059)

[2.3.2. in-game setting 57](#_Toc391446465)

[2.3.2.1. Drawing the flow graph 57](#_Toc1827792622)

[2.3.2.2. Determining a basis set of independent paths 58](#_Toc61452245)

[2.3.2.3. Preparing test cases 59](#_Toc1240378225)

[2.4. Test 4 - Pop-up   60](#_Toc1978743720)

[2.4.1. Success pop-up 60](#_Toc347269245)

[2.4.1.1. Drawing the flow graph 60](#_Toc1856942455)

[2.4.1.2. Determining a basis set of independent paths 61](#_Toc1321259093)

[2.4.1.3. Preparing test cases 61](#_Toc1607772223)

[2.4.2. Failure pop-up 62](#_Toc1682950225)

[2.4.2.1. Drawing the flow graph 62](#_Toc1143000322)

[2.4.2.2. Determining a basis set of independent paths 63](#_Toc1130251914)

[2.4.2.3. Preparing test cases 63](#_Toc1504493128)

## Introduction

### Purpose

This software requirement documentation is for the video game product "Triangle Cyber" made by our group the “Generic game dev group”. This game is a 2D side scroller adventure platformer which aims to provide an enjoyable, hard but fair challenge for players of all ages to relax after intensive work or study.

### Product Scope

This software requirement documentation describes the functional and non-functional requirements of the game's four main components: the master setting, starting selection, gameplay screen, and pop-up.

The scope of this product includes the design and development of a complete, playable 2D side-scrolling platformer game, with intuitive user interfaces and engaging gameplay. The game should be compatible with Android OS from version 12 and beyond, and should be built using Unity game engine.

These are the following subsystems of the game:

**Master setting:** This includes the configuration and setting options of the game, such as audio (sound and music), display language, connect to Google Play and toggle notification.

**Starting selection:** This includes the game menu, game options, and the starting selection of the game.

**Gameplay screen:** This includes the main gameplay screen, character movements, player controls, collision detection, game physics, and other gameplay-related functions.

**Pop-up:** This includes all the pop-up messages, such as in-game messages, game-over screens, level complete, and other notifications.

## Overall Description

### Product Perspective

This is a standalone product and there are no plans for any subsequent releases or follow-on members.

### Product Functions

**Menu:**

1. Access Master setting.
2. Change scene Level selection.

**Master setting :**

1. Changing volume for sound.
2. Changing volume for music.
3. Changing display language.
4. Connecting to google play.
5. Revert back to the default setting.
6. Confirm and exit. (save the configuration changes)
7. Toggle notification.

**Level selection:**

1. Level selection.
2. Move to the next sets of level.
3. Back button.

**Game play screen:**

1. Moving player character ( jump, left-right movement).
2. Moving enemy (left-right movement by the AI).
3. Moving obstacles such as saw, spike, etc. (horizontal and verticle movement).
4. Interact with NPC (lore and accept quest).
5. Restart level.
6. Counting score during a level.
7. Level setting.
8. Restart level.
9. Changing sound.
10. Save game state.
11. Quit level.

**Pop-up:**

Showing a board pop-up with user time and score at the end of the level.

### User Classes and Characteristics

Triangle Cyber is designed to be played by a wide range of users, with varying levels of technical expertise, educational levels, and experience. The primary user classes for this product are:

**Casual Players:** These users are looking for a fun and enjoyable gaming experience. They may not have a lot of experience playing video games or may only play games occasionally. They will likely use all the product functions, but may not be as concerned with completing the game or achieving high scores.

**Experienced Players:** These users are looking for a more challenging gaming experience. They may have a lot of experience playing video games and enjoy games that offer a significant challenge. They will likely use all the product functions, including features like restarting a level or saving game state, to try and complete the game and achieve high scores.

**Competitive Players:** These users are interested in competing with other players and achieving high scores. They may have a lot of experience playing video games and enjoy games that require skill. They will likely use all the product functions, including features like global leaderboards, to compete with other players and achieve high scores.

The most important user class for this product is likely to be Casual Players, as they will make up the majority of the user base. However, we also understand that it is important to ensure that the game offers enough challenge and engagement for Experienced and Competitive Players.

### Operating Environment

Triangle Cyber is designed to operate exclusively on Android devices running OS version 12 and beyond. The game can run on any Android device that meets the minimum hardware and software requirements.

**Hardware Platform:** The game is designed to operate on Android devices with a minimum of 2GB of RAM, 500MB of available internal storage, and a screen resolution of 720p or higher. The game requires a touch screen interface for user input.

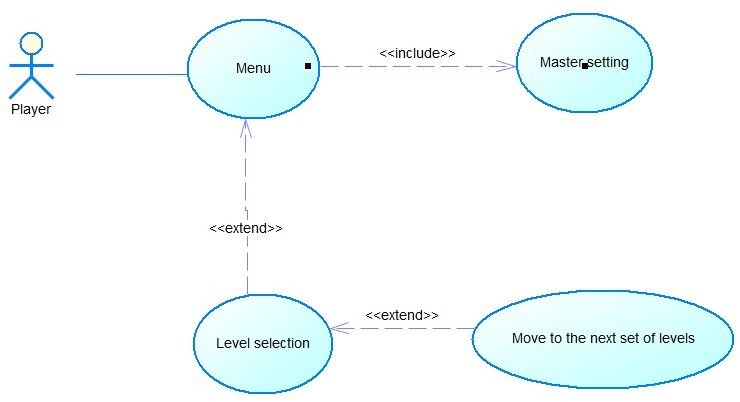
**Operating System and Versions:** The game is designed to operate on Android OS version 12 and beyond. It may not be compatible with earlier versions of the operating system.

**Software Components or Applications:** The game is designed as a standalone product and does not require any additional software components or applications to operate. There are no known conflicts with other software components or applications on Android devices.

It is important to note that the game may operate differently on different hardware platforms or devices with different specifications, and performance may vary depending on the device's hardware capabilities. It is recommended that users run the game on devices that meet the minimum requirements to ensure optimal performance

### Functional Requirement

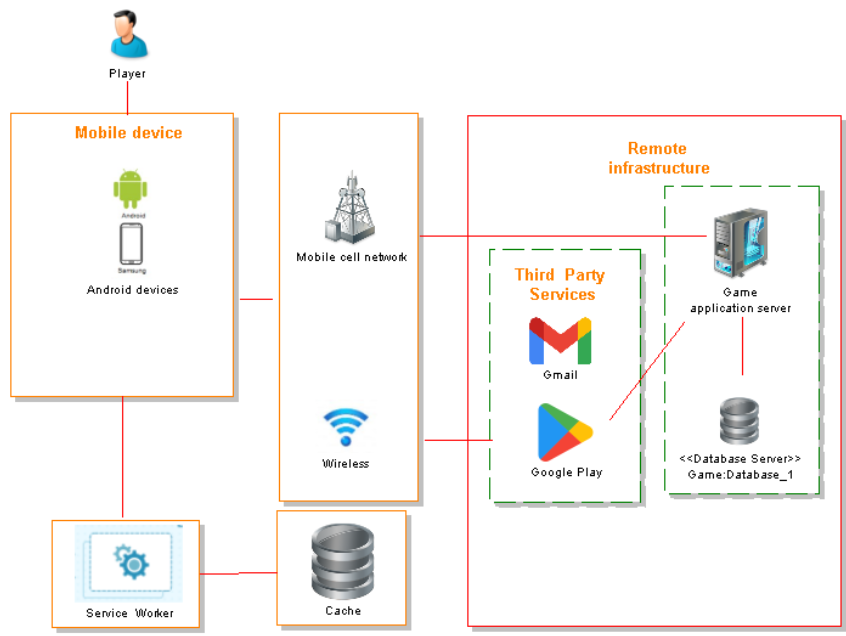
#### Use Case 1 (Starting selection)



|  |  |
| --- | --- |
| **Use Case:** Menu | ID: UC1.1 |
| **Main actor:** Player | Priority: 1 |
| **Brief description:**  When players presse the game’s icon, the menu is the first thing the greets them and from there, they can access other functions. | |
| **Trigger:** The game finishes downloading the game data, the screen shows the game name and the start button.  **Type:** external | |
| **Relationship:**  **+Association:** Player - Menu  **+Include:** Master setting  **+Extend:** Level selection  **+Generalization:** None | |
| **Normal flow:**   1. The Player taps on the game icon. 2. The game display the menu screen. 3. The Player taps on the ‘cog’ icon on the top right corner of the screen <<Master setting>> (Use case 2). 4. <<Level selection>> | |

## Application Architecture

### Application Architecture



### Description

The application architecture of the simple platformer game is designed for Android devices and involves the following components:

**Client devices:** The game will only run on Android devices, which will serve as the primary point of interaction for players.

**Game server:** The game server will handle requests from clients and store data related to the game, such as player progress, scores, and levels. The server will also handle communication with third-party services such as Google Play and Gmail.

**Game database:** The game database will store information related to the game, such as player profiles, scores, and level designs.

**Third-party services:** The game will integrate with third-party services such as Google Play and Gmail for software updates and notes as well as connecting the player to the globle leaderboard.

**Internet connectivity:** Players can connect to the game server and database via either Wi-Fi or mobile cell networks. If there is no internet connectivity, the game will use a service worker to store data as cache until the connection is restored.

The overall architecture is designed to provide a seamless experience for players, allowing them to interact with the game on their Android devices while communicating with the game server and database to store and retrieve game data. The integration with third-party services such as Google Play and Gmail ensures that the game remains up-to-date with the latest features and updates, while the use of a service worker ensures that players can continue to play the game even if they lose internet connectivity.

## Data Design

### Data Description

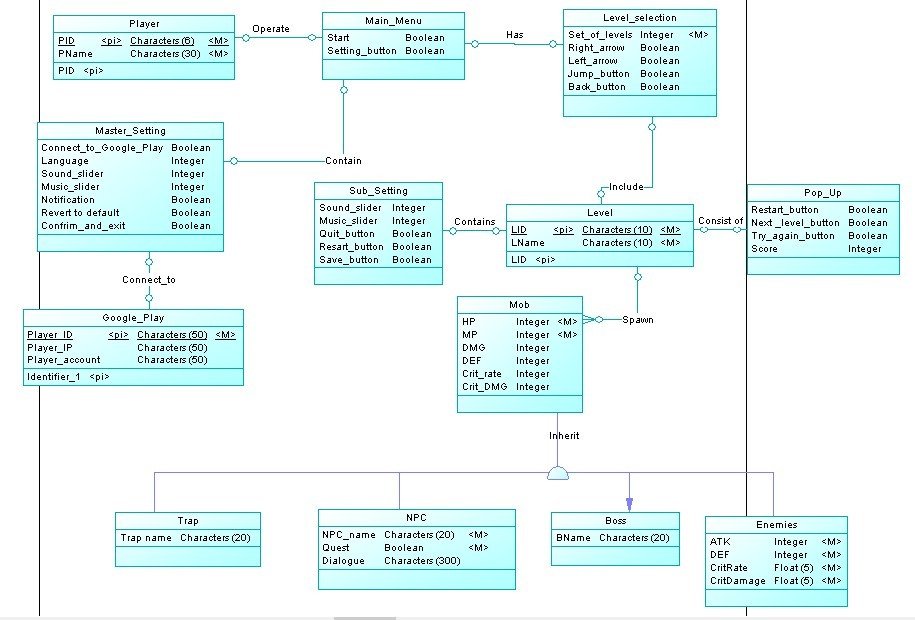
The game's data is primarily related to the player’s progress, scores, levels, and game settings. The following are the major entities that will be stored, processed, and organized within the system:

**Player profiles**: Each player's progress scores, and game settings will be stored in the game's database as a player profile. This profile will include information such as the player's username, current level, collected coins, and current configuration.

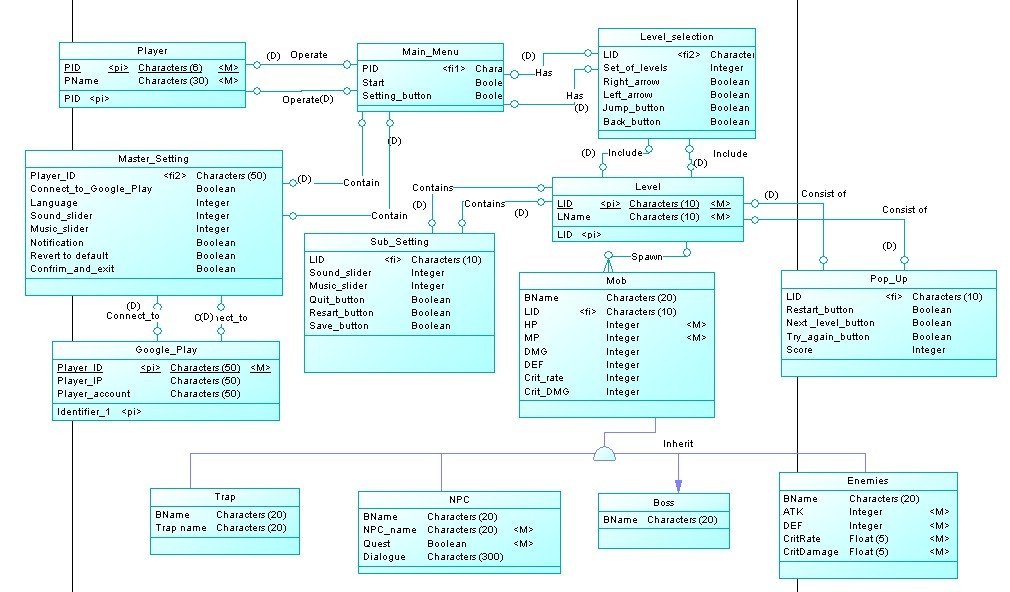
**Scores:** The system will store players' scores based on how many coins they collect and how quickly they complete each level. These scores will be processed and organized within the game's database and displayed to the player at the end of each level. The score system’s data then can be upload to Google Play to compare the player ability with others around the world on a global leaderboard.

**Game settings:** The game settings, both default the subsequences changes made by the players, will be stored locally on their device.

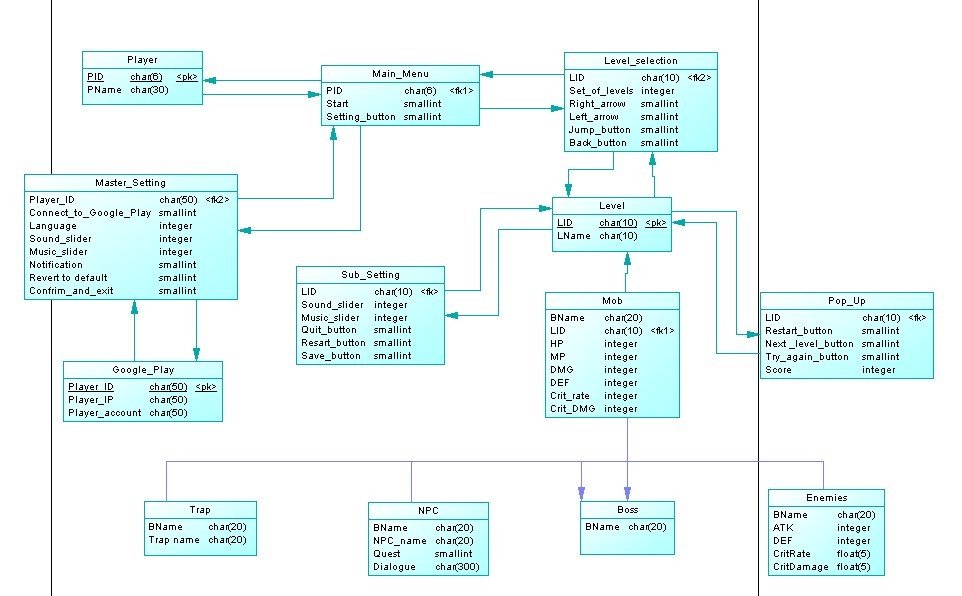
#### Conceptual Data Model



#### Logical Data Model



#### Physical Data Model



### Data Dictionary

|  |  |  |
| --- | --- | --- |
| **Entity/Data** | **Type** | **Description** |
| Back\_button | smallint | Used in the Level selection screen to return to Main menu. |
| BName | Char(20) | A string contains the name of a Boss. |
| Confirm\_and\_exit | smallint | Confirm and save setting changes in master setting. |
| Connect\_to\_ Google\_play | smallint | Confirm the game is connect to Google Play or not. |
| Crit\_DMG | float | The amount of critical damage an NPC can inflict on the player. |
| Crit\_rate | float | The chances that an NPC can inflict a critical attack on the player. |
| DEF | integer | The amount of damage an NPC can withstand before losing hit points. |
| Dialouge | char(300) | The text an NPC can say to the player. |
| DMG | interger | The amount of HP damage an NPC can deal to the player. |
| Ename | char(20) | A string represents an enemy’s name. |
| Enemies | Object | An entity that represents a hostile NPC. |
| GooglePlay | Object | An entity that represents the Google Play API used by the system. |
| HP | integer | The amount of hit points an NPC can withstand before losing hit points. |
| Language | integer | An integer value for each language (1 for English and 0 for Vietnamese). |
| Left\_arrow | smallint | A button that allows the player character to move to the left. |
| Level | Object | An entity that represents the playable level. |
| Level\_selection | Object | An entity that represents the level selection screen. |
| LID | char(10) | The string that represents the level ID. |
| LName | char(10) | A string represents the level’s name. |
| Main\_menu | Object | An object that represents the main menu. |
| Master\_setting | Object | An object that represents the master setting. |
| Mob | Object | An object that represents the game’s NPC both hostile and friendly. |
| MP | integer | The amount of magic points used to cast magic by an NPC. |
| Music slider | integer | A number that represents the music volume ranges from 0 to 100 and can be changed. |
| Next\_level\_  button | smallint | A value of either 1 (left) or 0 (right), in the success pop-up that transport the player to the next level. |
| Notification | smallint | A value of either 1 or 0, used to notify either the device has turned the game’s notification on or not. |
| NPC | Object | An entity that represents a non-playable character. |
| NameNPC | char(10) | A string that represents an NPC’s name. |
| PID | char(6) | A string that represents the player ID. |
| Player | Object | An entity that represents the player. |
| Player\_account | char(50) | A string that stores the player’s Google Play account. |
| Player\_ID | char(50) | A string that stores the player’s ID. |
| Player\_IP | char(50) | A string that stores the player’s IP. |
| Pname | char(30) | A string that stores the player’s name. |
| Point | integer | A number system used to gauge how well the player did in one level as well as ranking the player in the global leaderboard. |
| Pop\_up | Object | An entity that represents the in game pop-up. |
| Quest | smallint | A value of either 1 or 0, used to notify either an NPC has a quest or not. |
| Quit\_button | smallint | A value of either 1 or 0, used to notify either the player has pressed the “quit” button or not. |
| Restart\_button | smallint | A value of either 1 or 0, used to notify either the player has pressed the “restart” button or not. |
| Revert\_to\_ default | smallint | A value of either 1 or 0, used to notify either the player has pressed the “revert to default” button or not. |
| Right\_arrow | smallint | A button that allows the player character to move to the right. |
| Save\_button | smallint | A value of either 1 or 0, used to notify whether the player has pressed the “Save game” button or not. |
| Set\_of\_level | Interger | Technically an array that is used to contain each individual level. |
| Setting\_button | Smallint | A value of either 1 or 0, used in the main menu to access the master setting screen. |
| Sound\_slider | integer | A number that represents the sound volume range from 0 to 100 and can be changed. |
| Start | smallint | A value of either 1 or 0, used in the main menu transport the player to the level selection screen. |
| Sub\_setting | Object | An entity that represents the sub-setting (In level setting). |
| Trap | Object | An entity that represents an obstacle that the player has to overcome. |
| Trap name | char(20) | The string that stores the obstacle’s name. |
| Try\_again\_  buttono | smallint | A value of either 1 or 0, used in the failure pop-up to restart a level. |

## Detailed Design

### Starting Selection:

#### Menu screen:

**- Purpose:** Allows players to access "Select Level" and "Master Setting" through start button, setting button.

**- Interface:**

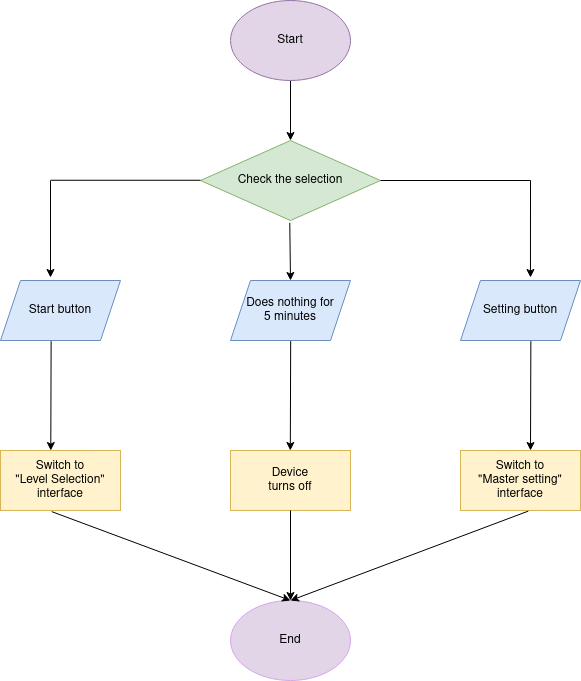


**- The components of interface:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Controller Type** | **Default Value** | **Note** |
| 1. | Button | Idle | Once activated by the player, the screen changes to the "Select Level" interface. |
| 2. | Button | Idle | Once enabled by the player, switch the interface to "Master Setting". |

**- Data to be used:** All actions/data of this interface are taken from the player and won’t add, modify, delete or query from any table or database.

**- Progress:**

****