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)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Nguyễn Quang Vinh | 19/04/2023 | Correct some typos. | 1.0 |
| Hà Quốc Huy | 19/04/2023 | Add a character model to the game play screen. | 1.0 |
| Nguyễn Quang Vinh | 20/04/2023 | Ajusting the positioning of multiple images and tables. | 2.0 |

# Team Work Plan

## Group Organization

|  |  |  |  |
| --- | --- | --- | --- |
| # | Student Code | Full Name | Role |
| 1 | B2111967 | Lưu Hoài Vũ | Team Leader |
| 2 | B2105727 | Nguyễn Quang Vinh | Vice of Team Leader |
| 3 | B2111947 | Trịnh Thanh Sang | Member |
| 4 | B2111885 | Hà Quốc Huy | Member |

## Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Task | Member (Name) | Deadline | Level of completion | Note |
| Software Requirement Specification | | | | | |
| 4 | Finding functional requirement | All member | 7/2/2023 | 100% |  |
| Finding non-function requirement | All member | 7/2/2023 | 100% |  |
| 5 | Meeting: Classifying requirements and adding to SRS | All member | 9/2/2023 | 100% |  |
| Finding external interface requirement and adding into SRS | Sang: User interface  Vinh: Hardware interface  Vũ: Software Interfaces  Huy: Communications Interfaces | 10/2/2023 | Vũ : 100%  Vinh :100%  Sang : 100%  Huy : 100% |  |
| 6 | Meeting:Re-check external interface requirements | All member | 15/2/2023 | 100% |  |
| Design User case | Sang: Starting Selection  Vinh: Master setting  Huy: Playing Screen  Vũ: Pop-up | 17/2/2023 | Vũ : 100%  Vinh :100%  Sang : 100%  Huy : 100% |  |
| 7 | Meeting:Re-check Use case | All member | 22/2/2023 | 100% |  |
|  | | | | | |
| 8 | Design the architecture of the system | All member | 24/2/2023 | 100% |  |
| 9 | Meeting:Check the architecture | All member | 28/2/2023 | 100% |  |
| Design the data model of the system | All member | 3/3/2023 | 100% |  |
| 10 | Meeting:Check the data model and design the interface | All member | 8/3/2023 | 100% |  |
| Design the interface of Use Case | Sang: Starting Selection:  Vinh: Master setting:  Huy: Playing Screen  Vũ: Pop-up | 10/3/2023 | Vũ : 100%  Vinh :100%  Sang : 100%  Huy : 100% |  |
| 11 | Meeting: Re-check the interface | All member | 15/3/2023 | 100% |  |
| Add information to SDS document | All member | 17/3/2023 | 100% |  |
| 12 | Meeting:Re-check the SDS document | All member | 22/3/2023 | 100% |  |
|  | | | | | |
| 13 | Create Test cases for each Use Cases | Sang: Starting Selection  Vinh: Master setting  Huy: Playing Screen  Vũ: Pop-up | 24/3/2023 | Vũ : 100%  Vinh :100%  Sang : 100%  Huy : 100% |  |
| 14 | Meeting: Recheck Test cases | All member | 28/3/2023 | 100% |  |
| Add test cases to Test cases document | All member | 31/3/2023 | 100% |  |
| 15 | Metting: Recheck Test cases document | All member | 5/4/2023 | 100% |  |
|  | | | | | |
| 16 | Recheck all document | All member | 14/4/2023 | 100% |  |
| 17 | Create Powerpoint presentation | All member | 21/4/2023 | 100% |  |
| Prepare for presentation | All member | 23/4/2023 | 100% |  |
| 18 | Presentation | All member | 28/4/2023 | 100% |  |

## Rules on the group

- Contact channel:

+ Discord

+ Zalo

- Group meeting:

+ Discord

+ Zoom (as backup)

+ Real life meeting

- Software to use to develop:

+ Game engine: Unity

+ GIMP as image minipulation program

+ Audacity for sound and music

# Software Requirement Specification

## 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.

### Definitions, Acronyms and Abbreviations

The software requirements specification does not contain any technical terms or acronyms that require definition. However, it does contain some abbreviations that needs clarifying:

|  |  |  |
| --- | --- | --- |
| **No.** | **Term / Acronyms** | **Definition / Explanation** |
| 1. | NPC | Non-playable character. |
| 2. | AI | Artificial intelligence. |
| 3. | OS | Operating system. |
| 4. | GB | Gigabyte. |
| 5. | MB | Megabyte. |
| 6. | RAM | Random access memory. |
| 7. | FTP | File transfer protocol. |
| 8. | API | Application program interface. |

### References

[1] IEEE Computer Society, IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998, 1998.

<http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>

[2] Perforce Software, Inc, Gerhard Krüger and Charles Lane, How to Write a Software Requirements Specification (SRS Document).

<https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>

[3] Relevant Software LLC, Andrew Burak, Your 2023 Guide to Writing a Software Requirements Specification (SRS) Document.

<https://relevant.software/blog/software-requirements-specification-srs-document/#Platform>

[4] Bộ Thông tin và Truyền thông, Hướng dẫn về các yêu cầu phi chức năng chung cho các hệ thống thông tin cung cấp dịch vụ công trực tuyến, 2013. <https://thuvienphapluat.vn/cong-van/Cong-nghe-thong-tin/Cong-van-1276-BTTTT-UDCNTT-2013-huong-dan-yeu-cau-phi-chuc-nang-chung-he-thong-dich-vu-cong-truc-tuyen-248044.aspx>

## 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 scence 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.

### Design and Implementation Constraints

**Platform Limitations:** The game is developed exclusively for Android OS version 12 and beyond. The game is compatible and operates correctly with these specific operating system versions.

**User Interface Design:** The game's user interface is intuitive and easy to use. The game's user interface is ensured to have met the standard design specifications and that it is consistent throughout the game.

**Localization:** The game is localized into several languages to reach a wider audience.

**Security Considerations:** The game is designed to be secure and protect user data. The game's code is ensured to be secure and that user data is stored securely on the device.

**Performance:** The game performs well on a variety of hardware configurations. The game is optimized for performance and is able to run smoothly on Android devices with varying levels of hardware capabilities.

**Programming Standards:** The development team has followed programming standards and conventions to ensure that the game's code is maintainable and easily extensible.

### Assumptions and Dependencies

**Assumptions:**

1. Users have a basic understanding of how to operate an Android device and play games on it.
2. Users have access to an Android device running Android OS version 12 or above.
3. The game will not be subject to any major updates or changes post development.

**Dependencies:**

1. The game is dependent on the Android OS version 12 or above, and will not run on any previous versions of the OS.
2. The game is dependent on the hardware of the Android device it is being run on, and may have performance issues on older or lower-end devices.

## Requirements Specification

### External Interface Requirements

#### User Interfaces

The game is designed to be played horizontally on a mobile device. The interface to start the game will have a "cog" icon, with a radius of 0.5 cm, located in the upper right corner of the screen, which allow player to access the master setting. The game's name is in the middle, rectangular in shape and can be changed depending on the size of the devices, the start button is located under the game’s name (It is a rounded rectangle 2cm long and 1cm wide).

After selecting the "cog" icon, an interface will appear in the middle of the screen with a length of 7cm and a width of 4cm. In the settings section, the word "Setting" will be at the top and centered on the interface. The buttons to change the game's configuration are as follows. A button directly below and slightly to the right of the "Setting" is used for connecting to Google Play, next to it is a button for toggling notification. Both of these buttons are 3 cm in length and 1 cm in width. Directly below them are two "flags" which are the language options, they are organized in a row with the length of 6cm and width of 1cm. Below is a slider for the game's sound volume. Again, below the sound slider is the music slider. Both of which have the same length and width as the language option. Finally, there will be two final buttons, the "Revert to default" button and "Save and exit" button, located on the left and right of the interface with the width and length of 1cm and 3cm respectively.

After pressing the start button, an interface called the level selection screen will appear. The interface will display 3 levels arranged horizontally. This is just a preview list of level and there are two arrows below that the user can use to move the list to the left or right.

After the level selection screen , the game screen will appear (level screen). In the game screen, there will be movement buttons of the player character. The button to move is located in the lower left corner of the screen, including the left-to-right button (the button will be a square corner with the length of the sides is 1 cm). The jump button will be located in the lower right corner of the screen (the width is 0.5 times larger than the button to move). In the upper left corner, there will be a pause button (square shape with a length of 1cm), after clicking the pause button, a table will appear (5cm long, 4cm wide). The panel will appear in the center of the device screen. In the table, there will be 3 buttons in the shape of a circle (1 cm radius, arranged from top to bottom of the table) including the restart, resume and save button in that respective order.

At the end of a game screen (level screen), will appear a board (6cm long, 4.5cm wide) with a congratulating message. Below will be the achievement achieved including completion time, in game currency (coin) used as a point system, number of deaths. The bottom will have three buttons to play again, next level, save and exit (same size as the three buttons restart, resume, save).

#### Hardware Interfaces

Triangle Cyber is created to run on a wide range of mobile devices, including smartphones and tablets Android version 12 and beyond operating system to function properly and for future maintenance. The minimal requirements for the game to run properly are as follow:

**Display:** The game requires a minimum of a color screen of minimum resolution of 720 x 1280 pixels with a refresh rate of at least 30 hz and support OpenGL ES 3.0 or later version.

**Memory and processors:** The game requires a minimum of 2GB of ram and a quad-core CPT for minimal delay or lag between user input and game response as well as to provides a smooth and enjoyable gaming experience for all users.

**Storage:** The game requires a minimum of 500MB of available internal space for installing the game, installing new updates when available and storing local game data.

**Network connectivity:** The game requires a stable wifi or cellular data connection for downloading updates and, optionally, accessing the global leaderboard.

**Audio:** The game requires the mobile device to come equipped with at least stereo speakers for audio output and support audio format such as Ogg Vorbis.

Additionally, users can also have the option of using external third party hardware such as: smartphone adapted controller, earphone, etc.

#### Software Interfaces

The communication between the product and specific software components depends on the specific architecture of the product and its functional requirements. However, to give a concrete example, we can describe some typical connections as follows:

**Database:** The game is connected to the database to store information about users, game history and results achieved. The database can be deployed using a database management system such as MySQL or PostgreSQL.

**Operating system:** The game can run on one operating system which is Android. Therefore, the game is compatible with multiple Android operating system versions (from version 12 and onward) and uses appropriate protocols to interact with the operating system.

**Tools:** The game uses many tools for development, testing, and deployment such as Unity, gimp, Unity remote 5, Audacity and Visual Studio Code. The product uses tools for version control, namely Git and Github.

**Libraries:** The game may use libraries to provide functions such as drawing graphics, input control, or audio processing such as: Unity, cocos2d-x and libGDX.

Some data items or messages that may go out of the system include:

**Storage data:** The game stores data such as player information, game information and configuration for future use.

**Game Information:** The product may send game information, including scores, progress, and playing time, to the product's server and the Google Play server

for the global leaderboard.

The product uses the Google Play API to allow players to log in with their account. The product may also use system APIs to access databases or cloud storage to store and retrieve data.

#### Communications Interfaces

When entering the game, if there is a new update or bug fixes available, there will be a message board notifying the player about the update. The Player then, can tap anywhere off the board to exit it and the option to update is left up to the player themselves.

In the master setting which is the “cog” icon on the top left corner of the menu, the player can tap on the “Connect to Google Play” to be able to connect to their Google Play account which in turn will create a Player account unique to the player themselves. This allows players to be able to save their game’s progress on their google account (synchronize on other devices) and connect to the global leaderboard.

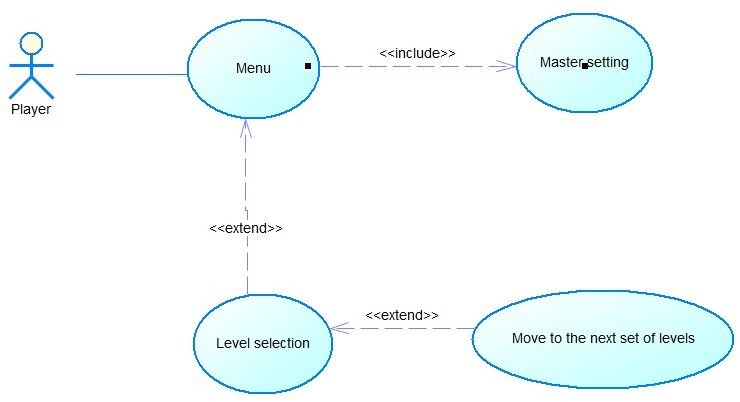
Game has a home page on facebook to post information about the game and the latest updates and it is also where the player can offer feedback to the developer team.

When logging into the game for the first time, there will be a message board that summarizes the safety and security regulations and the game's policies, players need to read carefully before clicking the agree button and start playing the game.

The game uses the FTP protocol to collect and store player information.

### Functional Requirement

#### Use Case 1 (Starting selection)

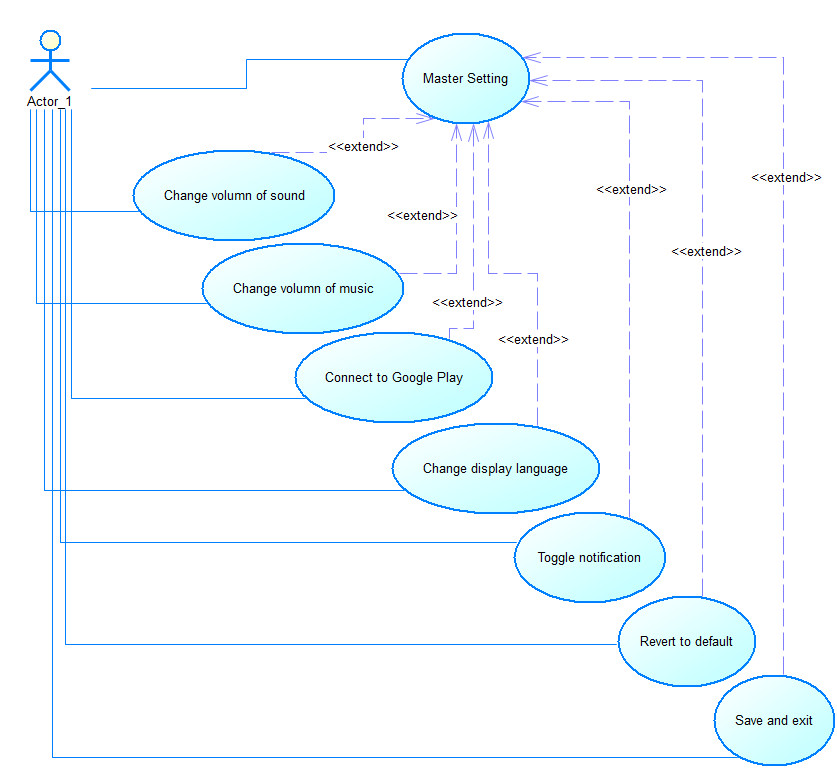


|  |  |
| --- | --- |
| **Use Case:** Menu | ID: UC1.1 |
| **Main actor:** Player | Priority: 1 |
| **Brief description:**  When players presse the game’s icon, the menu is the first thing the greets them and from there, they can access other functions. | |
| **Trigger:** The game finishes downloading 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>> | |

|  |  |
| --- | --- |
| **Use Case:** Level selection | ID: UC1.2 |
| **Main actor:** Player | Priority: 1 |
| **Brief description:** Interface for players to choose a screen to start the game. | |
| **Trigger:** The player taps the “Start” button on the menu screen.  **Type:** external | |
| **Relationship:**  **+Association:** Player - Level selection  **+Include:** None  **+Extend:** Move to the next set of levels  **+Generalization:** None | |
| **Normal flow:**   1. The system displays the game screen for players to choose. 2. If the player chooses the screen switch button. 3. <<Move to the sets of level>> 4. If the player chooses any level, the level starts. | |
| **Alternate/Exceptional flows**:   1. If the player does not select the level after 3 minutes. 2. The system appears with a notification screen reminding the player. 3. If the player does not select the screen after 5 minutes, the system will automatically save and turn off the device. | |

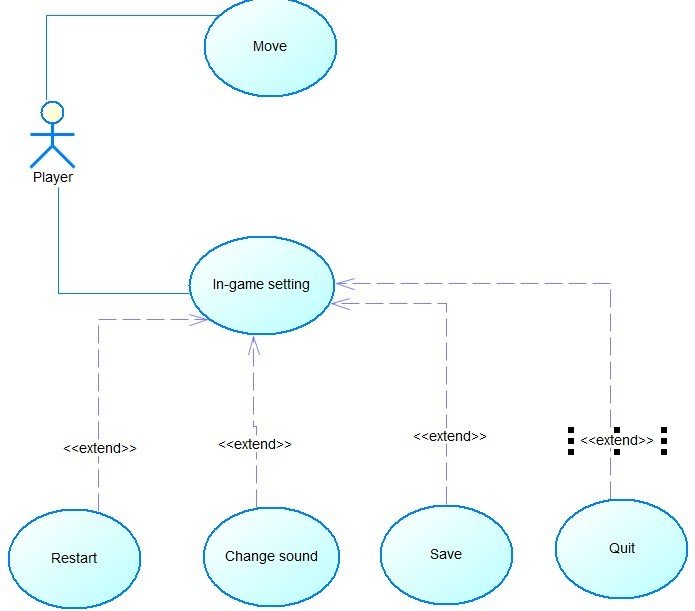
|  |  |
| --- | --- |
| **Use Case:** Move to the set of levels. | ID: UC1.3 |
| **Main actor:** Player | Priority: 1 |
| **Brief description:** The two arrow buttons used to move to change the game screen. | |
| **Trigger:** On the Level selection, the player either taps on the right arrow button or the left arrow button.  **Type:** external | |
| **Relationship:**  **+Association:** Player - Move to the set of levels  **+Include:** None  **+Extend:** None  **+Generalization:** None | |
| **Normal flow:**   1. When the player presses the screen switch button. 2. If the player presses the button to go to. 3. The system will move to the next level. 4. If the player presses the switch back. 5. The system will switch back to the old game screen. | |

#### Use Case 2 (Master setting):



|  |  |
| --- | --- |
| **Use Case:** Master setting | ID: UC2 |
| **Main actor:** Player | Priority: 1 |
| **Brief description:**  Allow player to change the volume of the game’s sound, music, language, revert to default, allow player to connect to the global leaderboard and turn on notification. | |
| **Trigger:** Player taps on the “cog” icon on the top right corner of the menu screen.  **Type:** external | |
| **Relationship:**  **+Association:** Player - Master setting  **+Include:** None  **+Extend:** Change volume of sound, Change volume of music, Save and exit, Connect to Google Play, Change display language, Toggle notification, Revert to default.  **+Generalization:** None | |
| **Normal flow:**   1. The player taps the cog icon on the top right corner of the screen. 2. The player is introduced to the setting screen. 3. If the player taps the “Connect to global leader board” button, the game will connect to the player’s google account. 4. If the player taps the dot on the right side of the “Notification” button to turn on the game’s notification (the dot slides to the right). 5. If the player taps on any of the flag icon (From left to right: English and Vietnamese), the game will ask the player to confirm the language change and reset the game for the change to take effect (Note that nothing will happen if the player tap on the flag icon of the language they are already using). 6. If the player adjusts the sound slider to the left the game’s sound will decrease and vice versa. 7. If the player adjusts the music slider to the left the game’s music will decrease and vice versa. 8. If the player taps the “Revert to default” button, all changes up until that point, if any, will be reset to the default value without any prompt. 9. If the player taps the “Save and exit” button, all changes to the game configuration will be saved to the device’s storage. | |
| **Alternate/Exceptional flows**:   1. If the player is already connected to the global leader, they can tap the “Connect to global leader board” button again to disconnect from it. 2. If the player already has notification turned on, they can tap the dot on the right side of the “Notification” button to turn it off (the dot slides to the left). 3. At any point, the device can suffer from hardware or software issues that can cause the device to reset. Upon which, all settings will be reverted to the previous time that was changed. 4. At any point, the device can run out of battery and will be forced to shut down. Upon which, all settings will be reverted to the previous time that was changed. | |

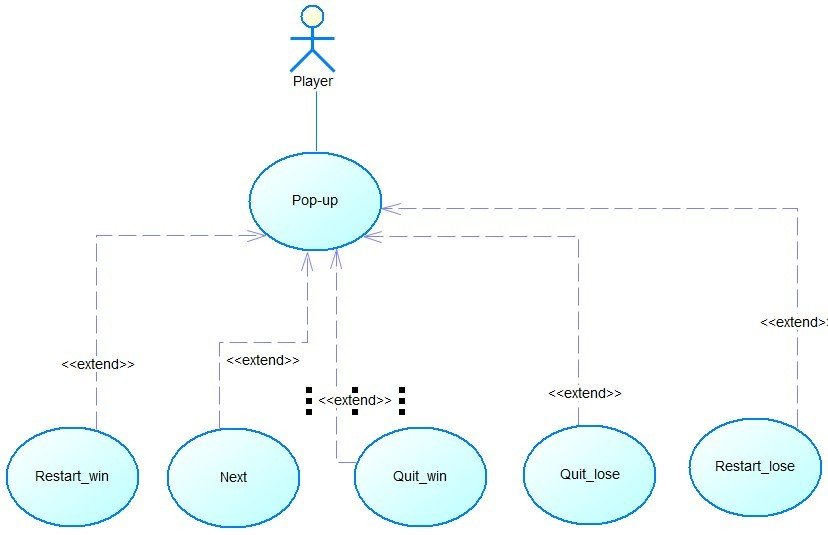
#### Use Case 3 (Gameplay screen):



|  |  |
| --- | --- |
| **Use Case:** Move | **ID:** UC3.1 |
| **Main actor:** Player | **Priority:** 1 |
| **Brief description:** When you press the movement buttons, you control the character's movements. | |
| **Trigger:** When the player taps on the middle Level in the Level selection interface.  **Type:** external | |
| **Relationship:**  **+Association:** Player - Move  **+Include:** None  **+Extend:** None  **+Generalization:** None | |
| **Normal flow:**   1. Player chooses the level to play. 2. Game system downloads screen data. 3. Players press the control buttons to move the character. 4. Press left arrow : character moves left. 5. Press the right arrow : the character moves to the right. 6. Press the side arrow pointing up : the character jumps high. | |

|  |  |
| --- | --- |
| **Use Case:** In-game setting | **ID:** UC3.2 |
| **Main actor:** Player | **Priority:** 2 |
| **Brief description:** The player can perform a number of actions including (quit game, restart, change sound, save and return to menu). | |
| **Trigger:** When the player taps the “cog” icon on the top right conner of any gameplay screen.  **Type:** external | |
| **Relationship:**  **+Association:** Player - In-game setting  **+Include:** None  **+Extend:** Restart, Change sound, Save, Quit  **+Generalization:** None | |
| **Normal flow:**   1. Players when clicking on the settings icon in the game. 2. The in-game settings screen appears. 3. The player manipulates the selections in the settings screen. 4. Press the "quit game" button -> the system displays the "cancel, confirm" menu:  * If one selects "cancel" -> return to the in-game settings screen. * If one selects "confirm" -> go back to the level picker screen.  1. Press the "restart" button -> the system displays the "cancel, confirm" menu:  * If one selects "cancel" -> return to the in-game settings screen. * If one selects "confirm" -> reload current level.  1. Press the "change sound" button -> the system displays the sound control panel. Player adjusts the sound -> the system updates the data 2. Press the "save and return to menu" button -> the system displays the "cancel, confirm" menu:  * If one selects "cancel" -> return to the in-game settings screen. * If one selects "confirm" -> save the current level and return to level selection. | |

#### Use Case 4 (Pop-up):



|  |  |
| --- | --- |
| **Use Case:** Pop-up | **ID:** UC4 |
| **Main actor:** Player | **Priority:** 3 |
| **Brief description:** Pop-up is a function used in video games to display an independent screen interface depending on the player's results. Pop-ups appear after the player completes or fails the game. | |
| **Trigger:** When the player finishes any level, either winning or losing.  **Type:** external | |
| **Relationship:**  **+Association:** Player - Pop-up  **+Include:** None  **+Extend:** Restart-win, Next, Quit\_win, Quit\_lose, Restart\_lose.  **+Generalization:** None | |
| **Normal flow:**   1. If the player loses, a pop-up will display a screen with the score obtained and a restart button to start the game again. 2. If the player wins, a pop-up will display a screen with the score obtained, a restart button to play again and improve the score and a next button to move to the next level. 3. The player can select any button on the pop-up. 4. If the player chooses "restart", the game will start from the beginning. 5. If the player chooses "next", the game will move to the next level. | |
| **Alternative flow:**   1. Player chooses to quit the game. 2. When the pop-up appears, the player selects the "quit" button. 3. The system prompts the player to confirm their choice to quit. 4. If the player confirms, the game ends and the player returns to the main menu or exits the game entirely. 5. If the player cancels, the pop-up remains on the screen and the player can choose another option. | |

### Nonfunctional Requirements

#### Performance

**Timing requirement:**. The game has a response time of no more than 300 milliseconds for any user action or game event. The processing time for each game event or user action is no more than 50 milliseconds. The software is designed to load within 3 seconds from the time the user initiates the app and takes no more than 20 seconds to start a new game session. While playing, the software can handle up to 500 game events (particle effects) and user actions per second. All of these statistics are measured with and collected during the testing phrases using Quicktime user. The measurements are based on the technical capabilities of the software and the devices it will be running on, as well as the needs and expectations of the users.

**Maximum capacity requirement:** the game is able to handle at least 100 active game sessions at any given time, to ensure that players are able to start and complete any session quickly and easily without any impact on performance or user experience. This will help ensure that players are able to enjoy the game without any issues. When connected to the Internet, the game requires no more than 500 Kbps of bandwidth to ensure that it can upload and pull data from app store or google play, while keeping the network costs low. The database size should be able to accommodate at least 5,000 active user profiles and game records, and should be able to handle at least 50 write operations per second and 500 read operations per second. These specific measurements will help ensure that the mobile game is able to handle the necessary volume of users and database operations, while maintaining a reasonable level of performance and user experience.

#### Reliability

Reliability of the software interface for the game is critical to ensuring a smooth and uninterrupted gaming experience. To achieve this, measures are taken to ensure system availability and resiliency.

**Availability:** In Triangle Cyber, availability is ensured by using cloud storage technologies through Google Play, to store player data, including player progress and player information. When a user logs in to another mobile device, their data is automatically synced from the cloud, allowing them to resume the game from where they left off on the old device.

**Resilience:** In Triangle Cyber, resilience is ensured by using libraries and error checking tools to detect and report bugs that are occurring in the game. When an error is detected, the system automatically restores the state of the game or provides options for the player to fix the error and continue playing. If data is lost or corrupted, the system needs to be able to restore data from backups or previous states of the game to ensure that players do not lose their progress.

#### Safety and Security

The game has ensured the confidentiality, integrity, and availability of data used or created by the product.

The product has prevented unauthorized access, modification, or destruction of data by only allow those who have the correct access clearance to access, modify or destroy data.

The product must provide user authentication and authorization mechanisms to ensure that data access is allowed only for authorized objects.

The product must include error handling mechanisms to prevent system crashes and data loss.

The product must have backup and recovery mechanisms to ensure the integrity and availability of data in case of system failures.

#### Adaptability and Portability

As this is a standalone project developed specifically for the Android OS version 12 and beyond, adaptability and portability are not significant concerns.

The system is not intended to be adapted to a variety of hardware platforms, software, operating systems, or usage environments beyond those specified in the requirements. The game is developed with the intention of running on any Android device with the specified operating system, without requiring any additional costs or changes to the system.

Therefore, the focus of the system is on providing a stable, efficient, and optimized user experience on the intended platform.

### Business Rules

As this is a low budget, non-profit, standalone game project, there are only a handful of general principles that is applied for Triangle Cyber:

1. Only registered users can save their game progress (after the game has been deleted and reinstalled) and access the global leaderboard.
2. Users must agree to the terms and conditions before playing the game.
3. Cheating or exploiting glitches in the game is not allowed and will result in disqualification from the global leaderboard.
4. The game is for entertainment purposes only and should not be used for any other commercial purposes.
5. Users should not share their login information with others.
6. The game should not contain any offensive or inappropriate content.

### Other Requirements

Additional requirements for Triangle Cyber:

**Database Requirements:** the game has a local database to store user progress, such as completed levels, high scores, and configuration settings.

**Legal Requirements:** the game complies with all applicable laws and regulations, including but not limited to copyright laws, privacy laws, and data protection laws.

**Reuse Objectives:** The game aims to maximize code reuse and maintainability through the use of software design standards and documentation.

**User Documentation:** The game includes a comprehensive user documentation in the form of a tutorial that aims to help users understand the game mechanics and controls.

**Maintenance and Support:** The game isn’t planned for any major updates post release. However, it will still receive bug fixes as needed.

# Software Design

## Introduction

### Purpose

The purpose of the “simple platformer” game software is to provide an enjoyable and engaging experience for players of all ages. The software allows players to control a character on a platform jumping game, collect coins, and complete levels while recording their scores based on performance. The sound effects and music add to the overall immersive experience. The software is designed to be intuitive, easy to use but only accessible on android devices. The software's primary aim is to provide entertainment and fun for players while keeping track of their scores and progress.

### Product Scope

The “simple platformer” game is a software system that allows players to control a character on a platform jumping game. The system is designed to be intuitive, easy to use, and enjoyable for players of all ages. The primary features of the system include:

**Character control:** Players can move the character left or right and jump up.

**Level design:** The system comes with pre-built levels.

**Scoring:** The system records players' scores based on how many coins they collect and how quickly they complete each level.

**Sound effects and music:** The system includes sound effects for character movements, level completion, and background music.

**Inputs:**

1. Player controls, such as left/right movement and jumping.
2. Level editor inputs for creating custom levels.

**Data Stores:**

1. The system will use a single database to store player scores and game progress.
2. The system will store data on the player's device if they are not connected to the internet.

**Outputs:**

1. The system will display the game screen, including the character, levels, and score.
2. The system will provide sound effects and background music.
3. The system will display the final score and game over screen when the game is completed or the player loses.

**Constraints:**

1. The system will use only one server and one database for storing data.
2. The game will be able to connect to Google Play through Wi-Fi or mobile cell networks to send data. If not connected, data will be stored on the player's device.
3. The system is designed only for android mobile devices.

### Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| No. | Term / Acronyms | Definition / Explanation |
| 1. | Bname | Boss name. |
| 2. | DEF | Defence. |
| 3. | DMG | Damage. |
| 4. | HP | Hit points. |
| 5. | Crit\_rate | Critical rate. |
| 6. | Crit\_DMG | Critical damage. |
| 7. | LID | Level Identification code. |
| 8. | LName | Level name. |
| 9. | MP | Magic points. |
| 10. | Mob | A computer-controlled non-player character in a video game. |
| 11. | NPC | Non-player character. |
| 12. | PID | Player indentification code. |
| 13. | IP | Internet protocal. |

#### References

1016-1998 - IEEE Recommended Practice for Software Design Descriptions

<https://ieeexplore.ieee.org/document/741934/figures#figures>

Nuclino, How to Write a Software Design Document (SDD),

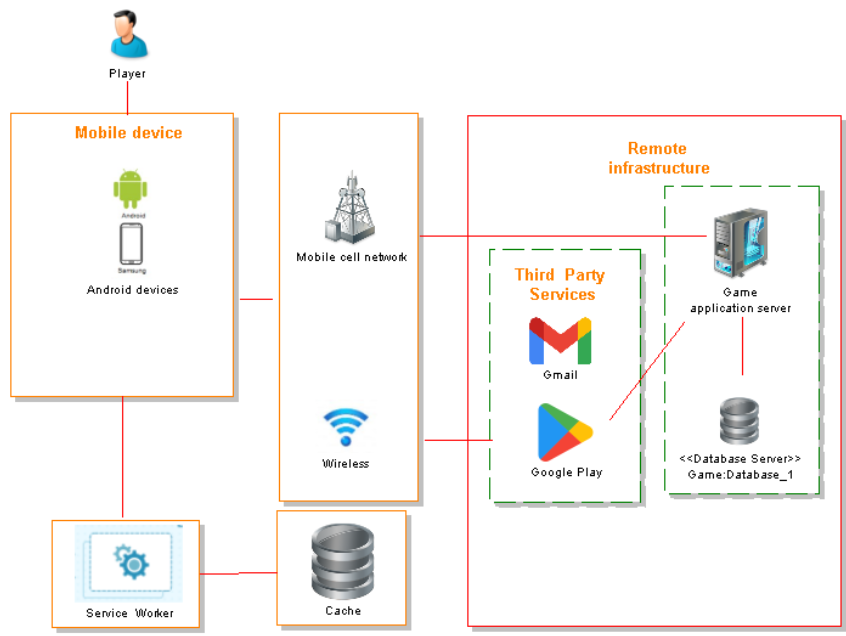
<https://www.nuclino.com/articles/software-design-document>

Angela Zhang, freeCodeCamp, How to write a good software design doc

<https://www.freecodecamp.org/news/how-to-write-a-good-software-design-document-66fcf019569c/>

## 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

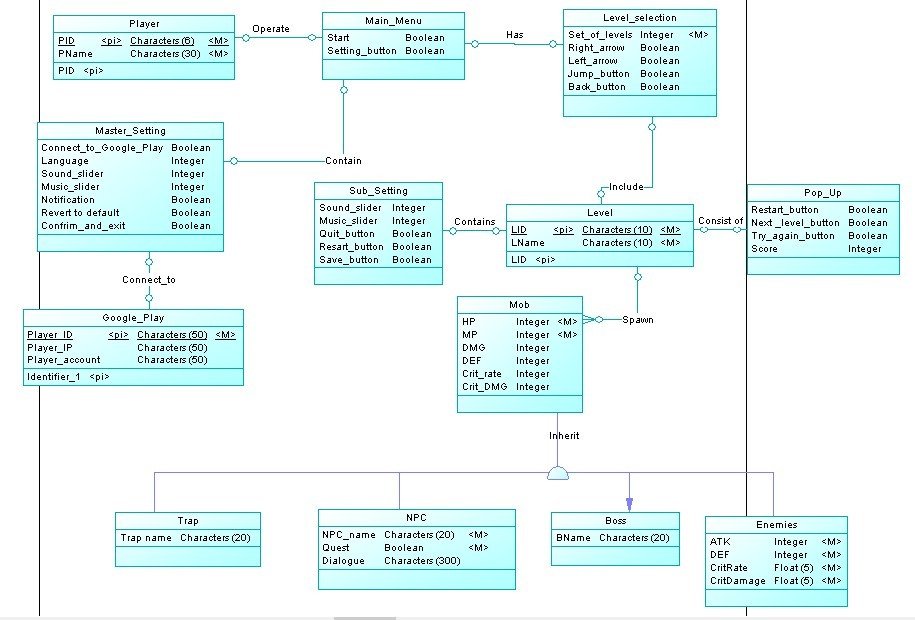
The game's data is primarily related to the player’s progress, scores, levels, and game settings. The following are the major entities that will be stored, processed, and organized within the system:

**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.

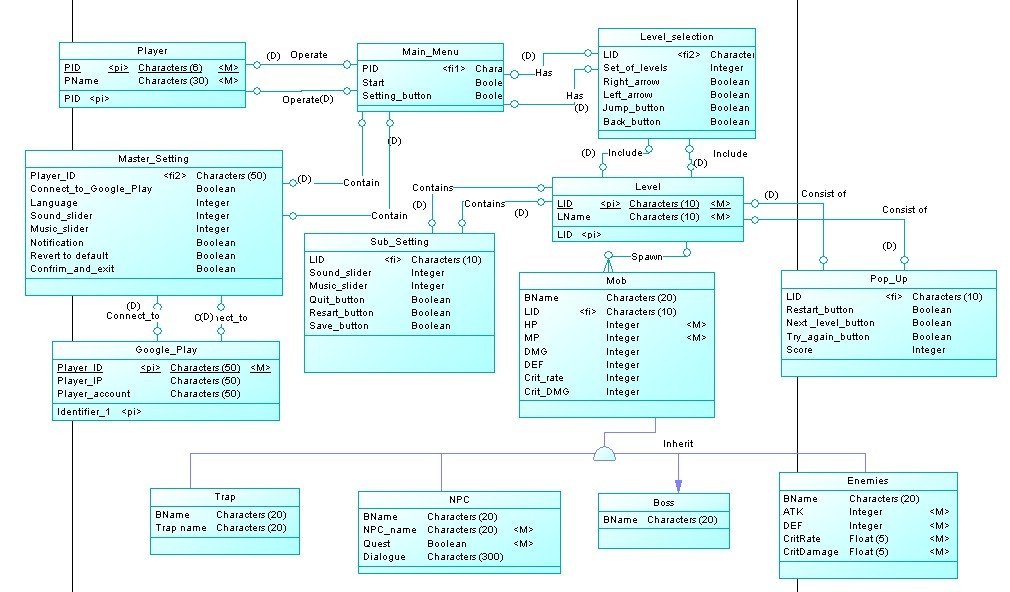
**Scores:** The system will store players' scores based on how many coins they collect and how quickly they complete each level. These scores will be processed and organized within the game's database and displayed to the player at the end of each level. The score system’s data then can be upload to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

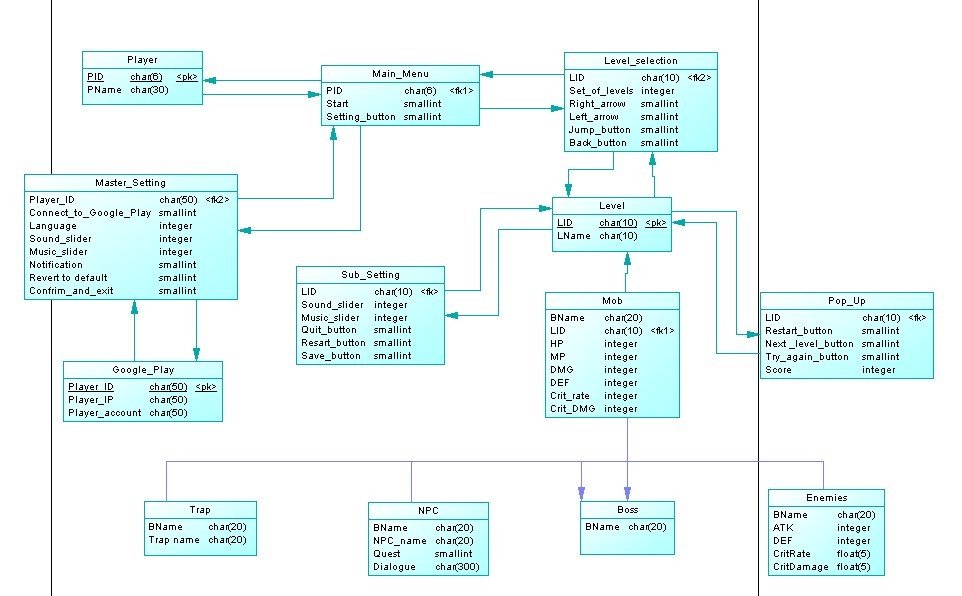
#### Conceptual Data Model



#### 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:**

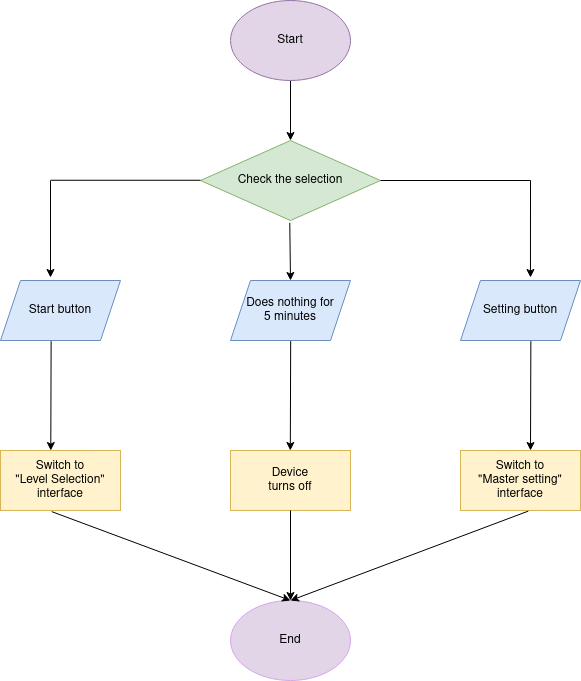


**- The components of interface:**

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**

****

#### Level selection and move to the sets of level:

**- Purpose:** Helps the player in selecting and navigating between each playable level.

****

41

5

6

3

2

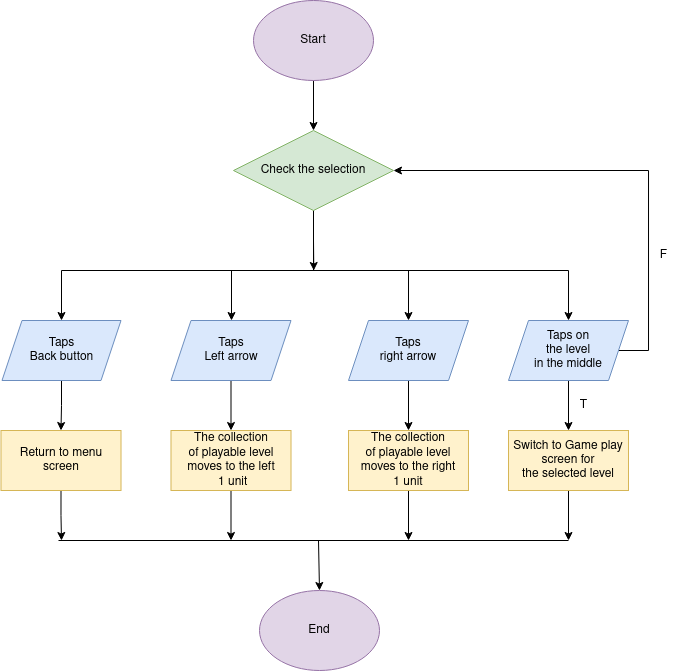
1

**-** **Interface:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Controller type** | **Default value** | **Note** |
| 1. | Button | Idle | An untappable button, a preview for the previous level. |
| 2. | Button | Idle | Tap on the button to proceed to the game play screen of that level. |
| 3. | Button | Idle | An untappable button, a preview for the next level. |
| 4. | Button | Idle | Player taps will switch to the level selection interface on the left. If there is no interface, the button will not work. |
| 5. | Button | Idle | Player taps will switch to the level selection interface on the right. If there is no interface, the button will not work. |
| 6. | Button | Idle | Player taps this button to return to the Menu screen. |

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

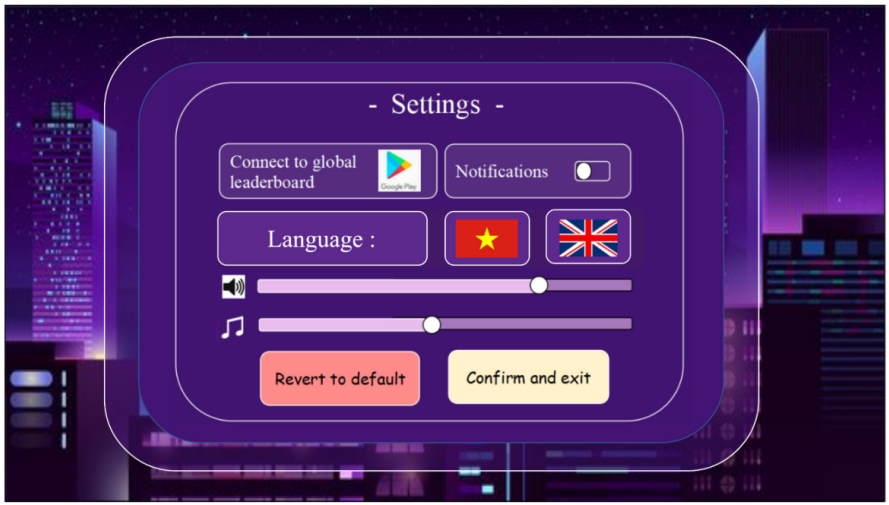
**- Progress:**

****

### Master setting:

**- Purpose:** Allow players the ability to change the volume of the game’s sound and music. As well as changing the display language, connecting to the global leaderboard via google play and turning on the notification.

**- Interface:**



5

4

9

8

7

6

2

3

1

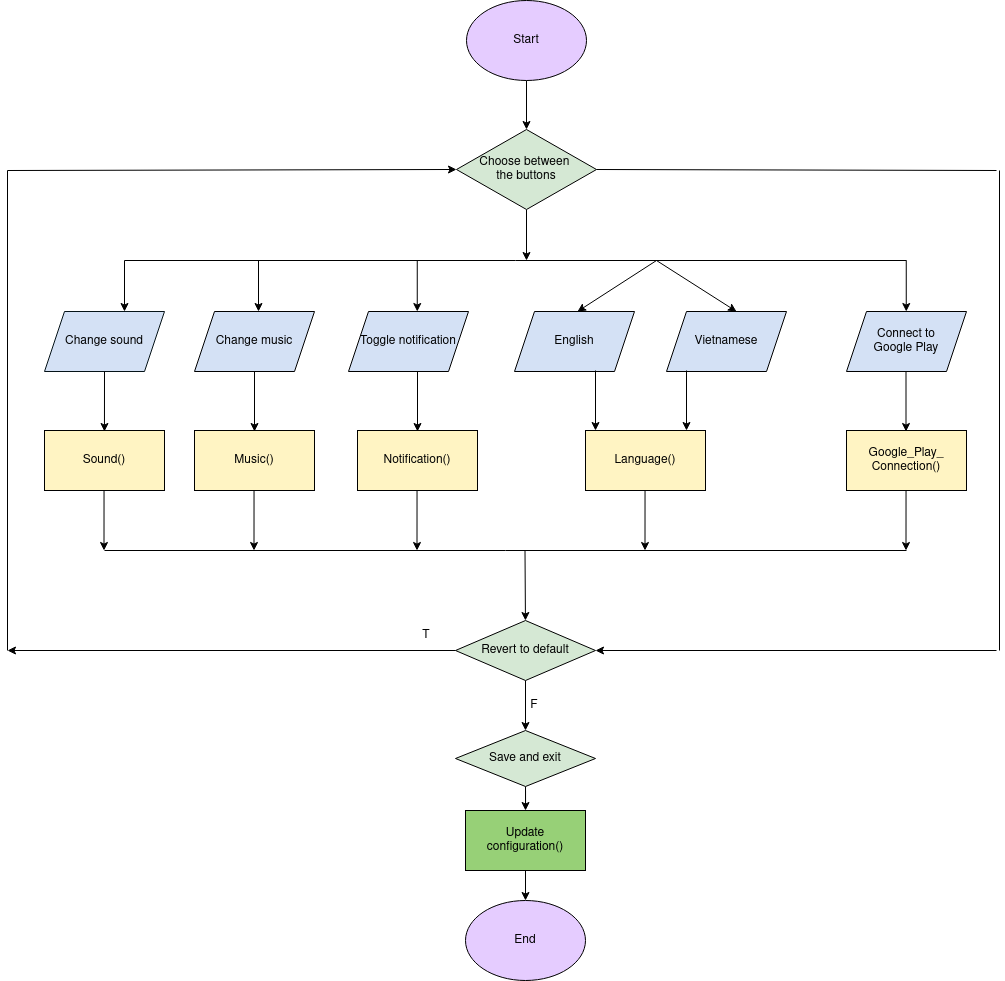
**- The components of interface:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Controller type** | **Default value** | **Note** |
| 1. | Form |  |  |
| 2. | Button | Unconnected | Upon successful connection, give the player a notification saying “Connection successful”, otherwise say “Connection unsuccessful”. |
| 3. | Button | At non-active state (node in the left) | Upon activation, send the user a notification confirming the success of the action. Send a reminder notification every 24 hours. |
| 4. | Button | Chosen by default | Require a game restart if the language was changed to take effect. |
| 5. | Button | Unactivated | Require a game restart if the language was changed to take effect. |
| 6. | Slider | At full capacity (100%) |  |
| 7. | Slider | At full capacity (100%) |  |
| 8. | Button |  | Reset all buttons and sliders to default value except for the 6th button. |
| 9. | Button |  |  |

**- Data to be used:** list of tables in a database or the data structure needed by this function.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Table name/ Data structure** | **Method** | | | |
| **Add** | **Modify** | **Delete** | **Query** |
| 1. | Google Play |  |  |  | x |
| 2. | Notification |  | x |  |  |
| 3. | English |  | x |  |  |
| 4. | Vietnamese |  | x |  |  |
| 5. | Sound |  | x |  |  |
| 6. | Music |  | x |  |  |

**- Progress:**

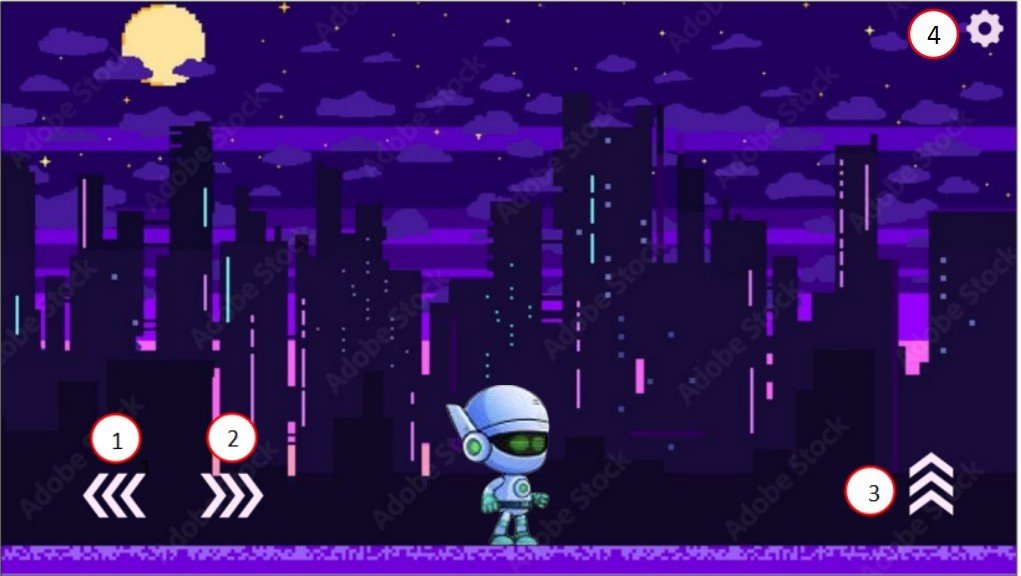


### Gameplay Screen:

#### Play Screen:

**- Purpose:** allows the player to use the movement buttons to control the character interactively at the game screen.

**- Interface :**



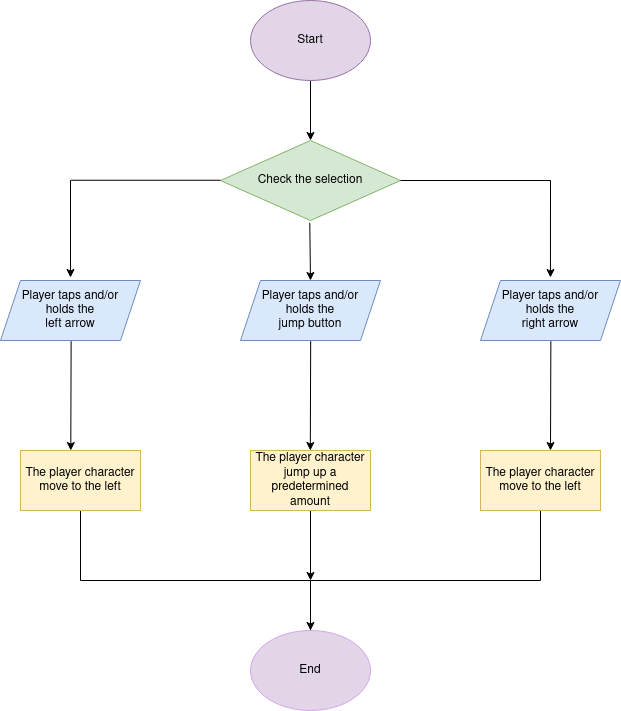
5

**- The components of interface:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Controller type** | **Default value** | **Note** |
| 1. | Button | Idle | Moves the character to the left. |
| 2. | Button | Idle | Moves the character to the right. |
| 3. | Button | Idle | Makes the character to jump up. |
| 4. | Button | Idle | Access the in-game setting. |
| 5. | Character | Idle | This represent the player character. |

**- 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:**



#### in-game setting:

**- Purpose:** Allows the player to perform several actions including (quit the game, restart, change the sound, save and return to the menu).

**- Interface :**



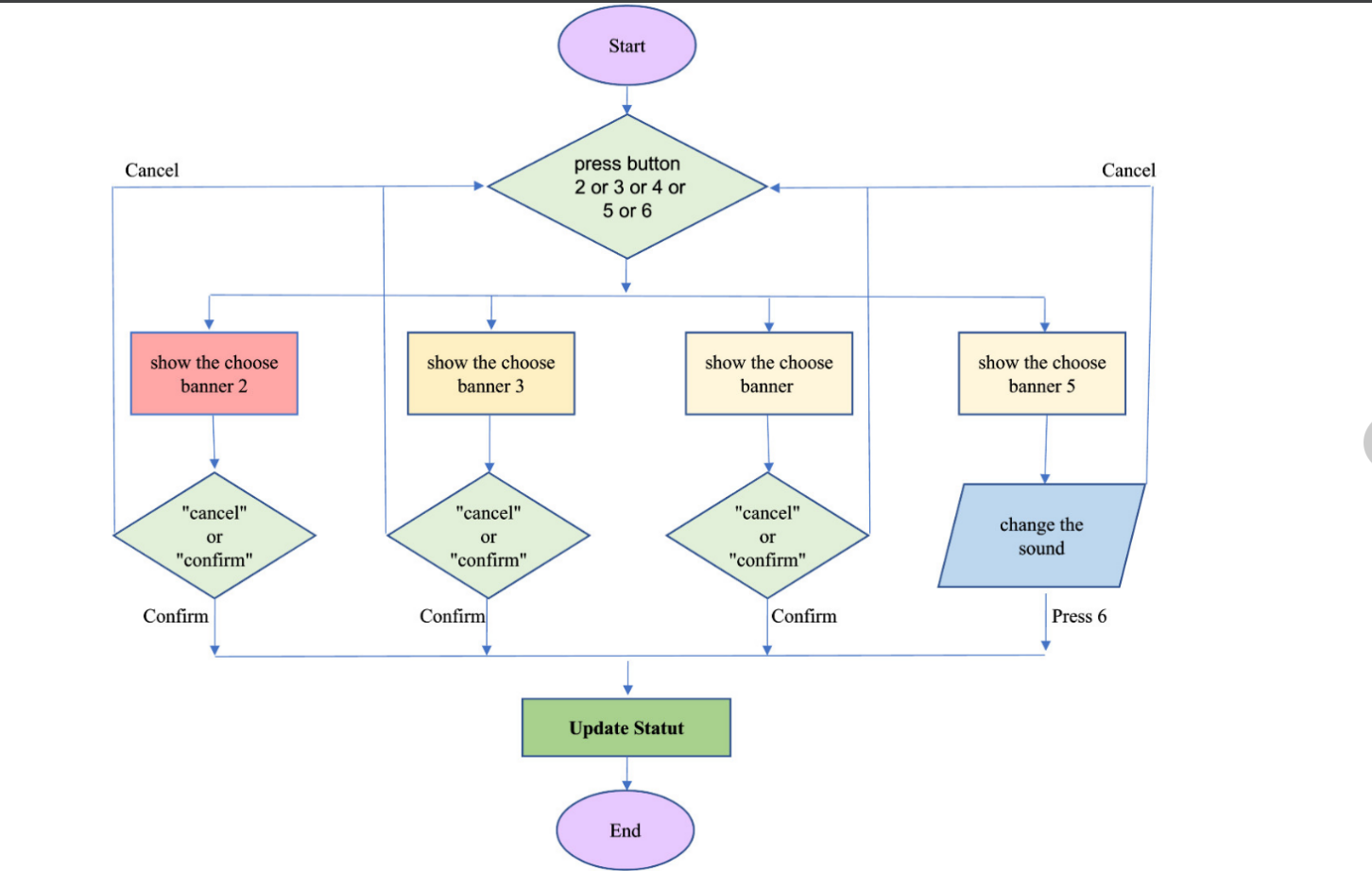
**- The components of interface:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Controller type** | **Default value** | **Note** |
| 1. | Button | Idle | Player press on it will show the in-game settings. |
| 2. | Button | Idle | Player press this button when they want to exit the level. |
| 3. | Button | Idle | Player press this button when they want to restart the level. |
| 4. | Button | Idle | Player press this button when they want to save their current level and return to the level selection. |
| 5. | Button | Idle | Player press this button when they want to change the game sound. |
| 6. | Button | Idle | Player press here when they want to turn off the in-game settings. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Table name/ Data structure | Method | | | |
| Add | Modify | Delete | Query |
| 1 | Master Setting |  | x |  |  |
| 2 | Level selection |  |  |  | x |

**- Data to be used:** list of tables in a database or the data structure needed by this function.

**- Progress:**



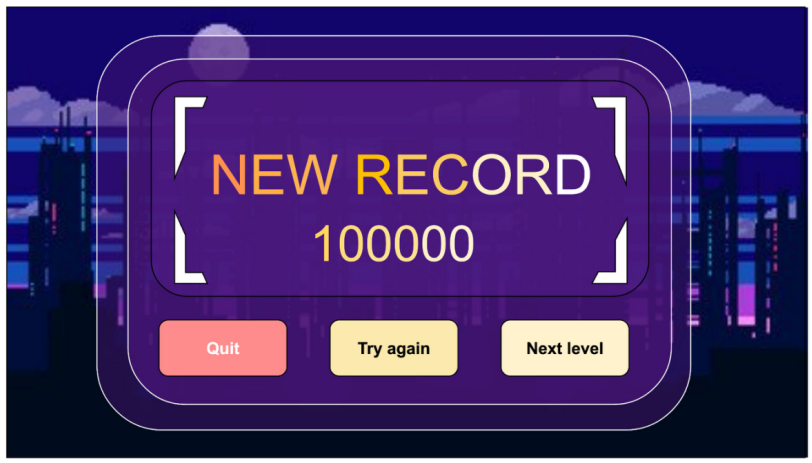
.

### Pop-up:

#### Success pop-up

**- Purpose:** Notify players when they have succeeded in passing a level as well as show the points accumulated throughout that particular level.

**- Interface :**



4

5

31

2

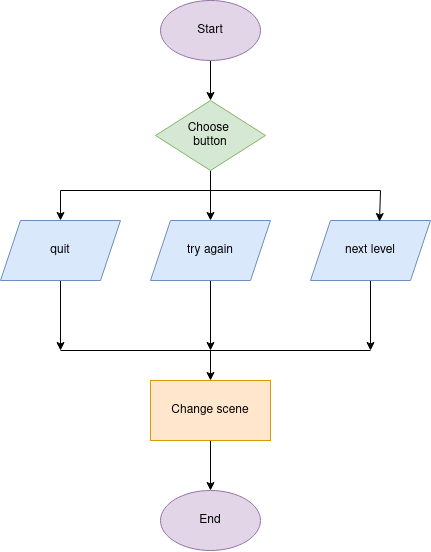
1

**- The components of interface:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Controller type** | **Default value** | **Note** |
| 1. | Form |  |  |
| 2. | Form |  |  |
| 3. | Button | Idle | Players press this button when they want to quit and return to level selection screen. |
| 4. | Button | Idle | Players press this button when they want to play the level again with the current result erased. |
| 5. | Button | Idle | Players press this button when they want to proceed to the next level. |

**- 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:**

****

#### Failure pop-up

**- Purpose:** Notify players when they have failed in passing a level as well as show the points accumulated throughout that particular level.

**- Interface :**



4

3

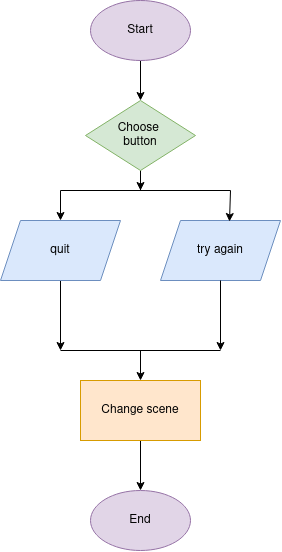
2

1

**- The components of interface:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Controller type** | **Default value** | **Note** |
| 1. | Form |  |  |
| 2. | Form |  |  |
| 3. | Button | Idle | Players press this button when they want to quit and return to the level selection screen. |
| 4. | Button | Idle | Players press this button when they want to play the level again with the current result erased. |

**- Progress:**

****

## References to Requirements

|  |  |  |
| --- | --- | --- |
| **No.** | **References to Requirements (SRS)** | |
| 1. | **4.1** Starting Selection:  **4.1.1** Menu Screen  **4.1.2** Level selection and move to the sets of level | **3.2.1** Use Case 1 (Starting selection). |
| 2. | **4.2** Master setting | **3.2.2** Use Case 2 (Master setting). |
| 3. | **4.3** Gameplay Screen:  **4.3.1** Play Screen  **4.3.2** in-game setting | **3.2.3** Use Case 3 (Gameplay screen). |
| 4. | **4.4** Pop-up | **3.2.4** Use Case 4 (Pop-up). |

# Unit Test

## Introduction

### Purpose

The purpose of this document is to provide a comprehensive overview of the testing approach and test cases for Triangle Cyber. It is intended for the development team and testers who will be involved in the testing and quality assurance process.

The document will detail the various test scenarios and expected results to ensure that the game functions as intended and meets the project requirements. Additionally, the document will serve as a reference guide for future maintenance of the game.

### Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| No. | Term / Acronyms | Definition / Explanation |
| 1. | TC | Test case. |
| 2. | ID | Identification. |

### References

Thomas Hamilton, GURU99, How to Write Test Cases in Software Testing with Examples

<https://www.guru99.com/test-case.html>

Dennis Gurock, testmo, Writing Test Cases: Examples, Best Practices & Test Case Templates

<https://www.testmo.com/guides/writing-test-cases-examples>

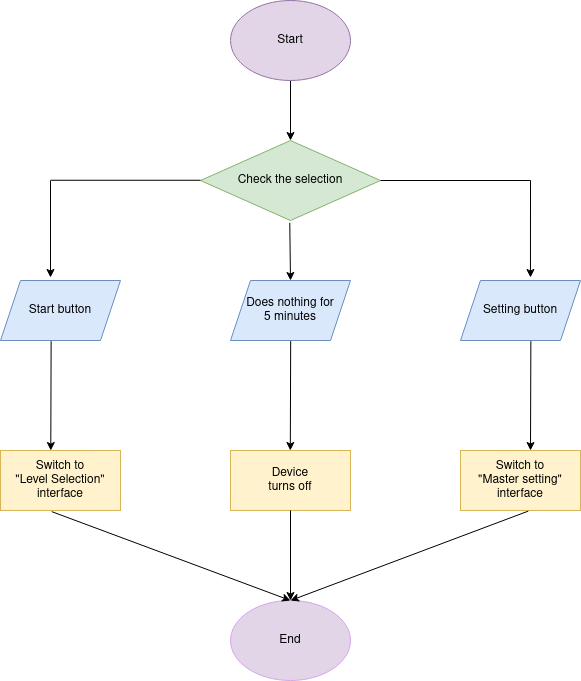
## Test Cases

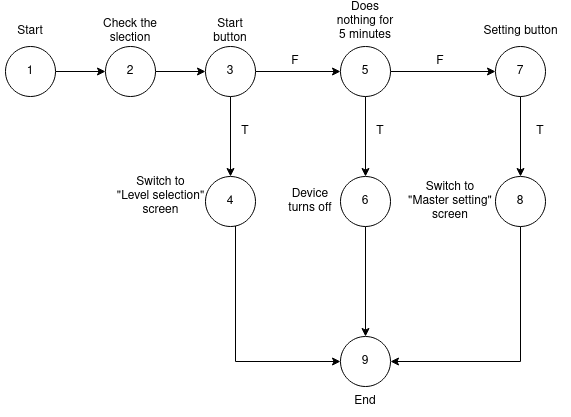
### Test 1 - Level selection

#### Menu screen:

|  |  |
| --- | --- |
| Test ID: | T001.1 |
| Test description | Test the function of the main menu. |
| Created by: | Trịnh Thanh Sang |
| Date of creation: | 31 – 3 – 2023 |
| Date of review: | 1 – 4 -2023 |
| Test priority: | Intermediate |
| Pre-requisite: | The game is downloaded and turned on. |

##### **Drawing the flow graph**

****



##### **Determining a basis set of independent paths**

**TC1:** 1 - 2 - 3 - 4 - 5 ( Select the start button ).

**TC2:** 1 - 2 - 3 - 5 - 6 - 9 ( Does nothing ).

**TC3:** 1 - 2 - 3 - 5 - 7 - 8 - 9 ( Select the Master setting button).

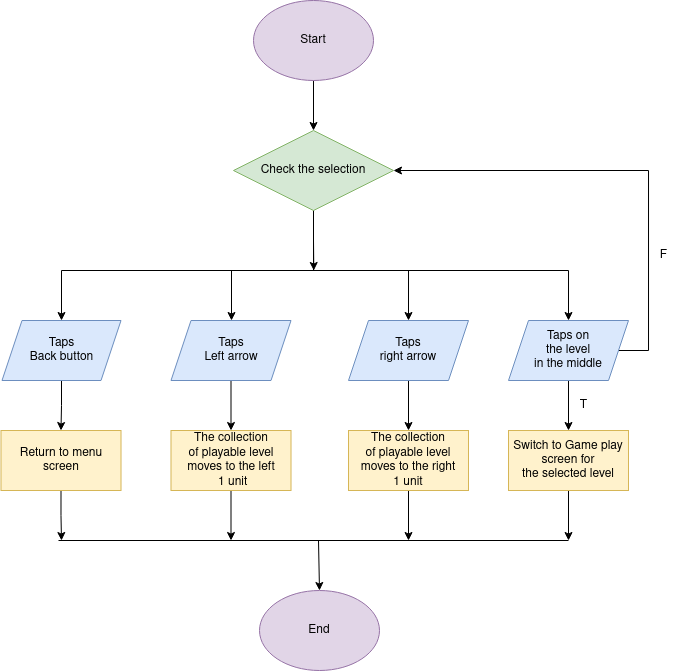
##### **Preparing test cases**

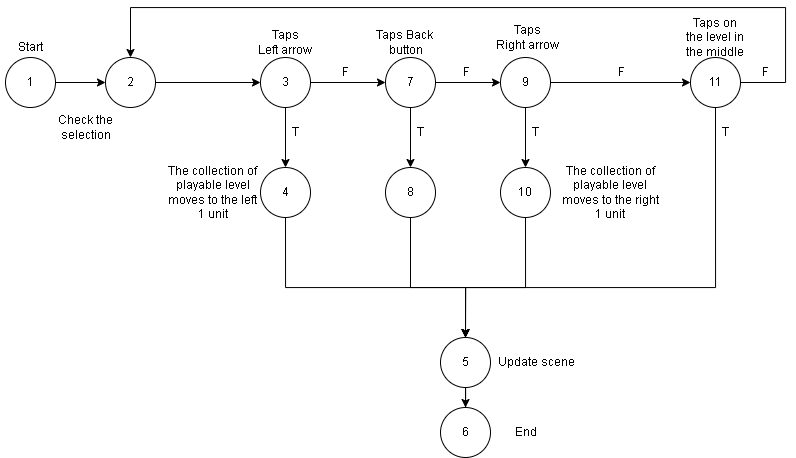
|  |  |  |
| --- | --- | --- |
| Test Case | Input Data | Expected Result |
| **User Input** |
| T001.1-TC1 | The user taps the start button. | Switch to “Level Selection” screen. |
| T001.1-TC2 | The user does nothing for 5 minutes. | Device turns off. |
| T001.1-TC3 | The user taps the setting button. | Switch to “Master setting” screen. |

#### Level selection and move to the sets of level:

|  |  |
| --- | --- |
| Test ID: | T001.2 |
| Test description | Test the operation of left arrow, right arrow and entering a playable level. |
| Created by: | Trịnh Thanh Sang |
| Date of creation: | 31 – 3 – 2023 |
| Date of review: | 1 – 4 -2023 |
| Test priority: | High |
| Pre-requisite: | The player presses the “Start” button. |

##### **Drawing the flow graph**





##### **Determining a basis set of independent paths**

**TC1:** 1 - 2 - 3 - 4 - 5 - 6 (Taps the left arrow).

**TC2:** 1 - 2 - 3 - 7 - 8 - 5 - 6 (Taps back button).

**TC3:** 1 - 2 - 3 - 7 - 9 - 10 - 5 - 6 (Taps the right arrow).

**TC4:** 1 - 2 - 3 - 7 - 9 - 11 - 5 - 6 (Enter a new level immediately).

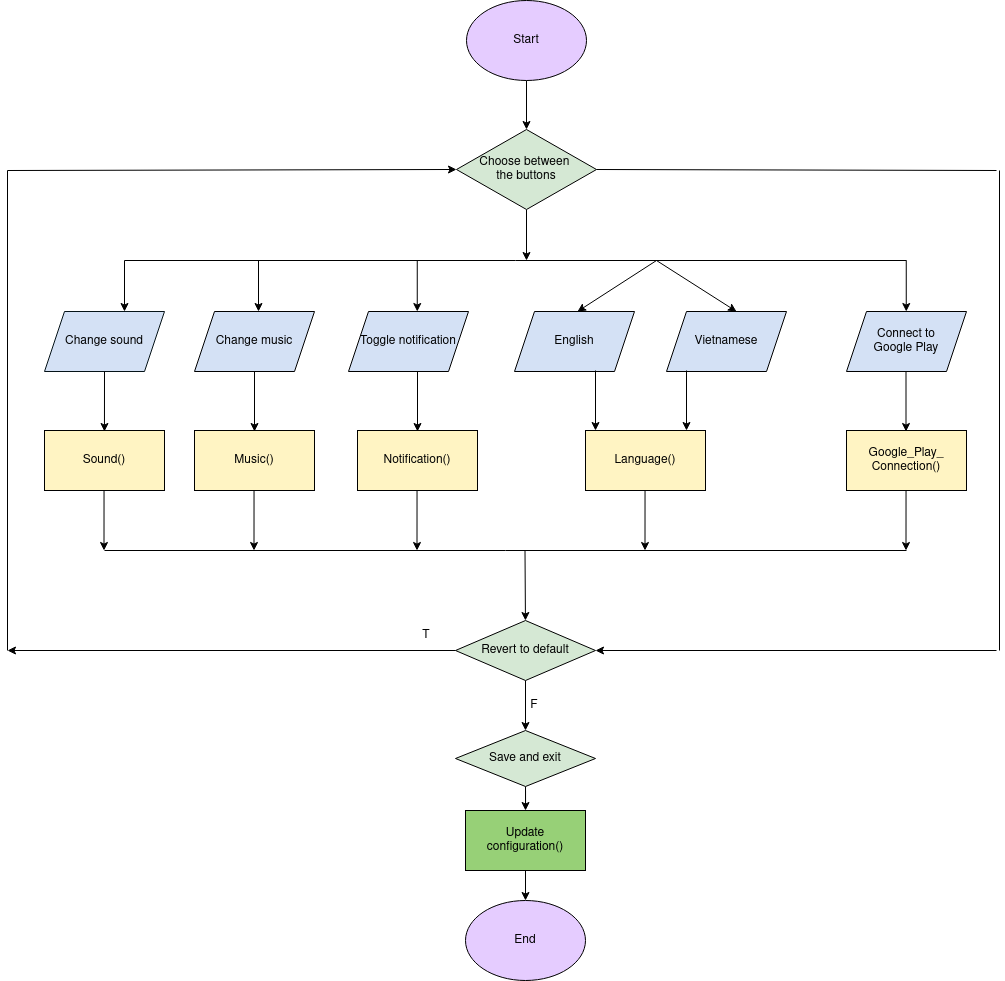
##### **Preparing test cases**

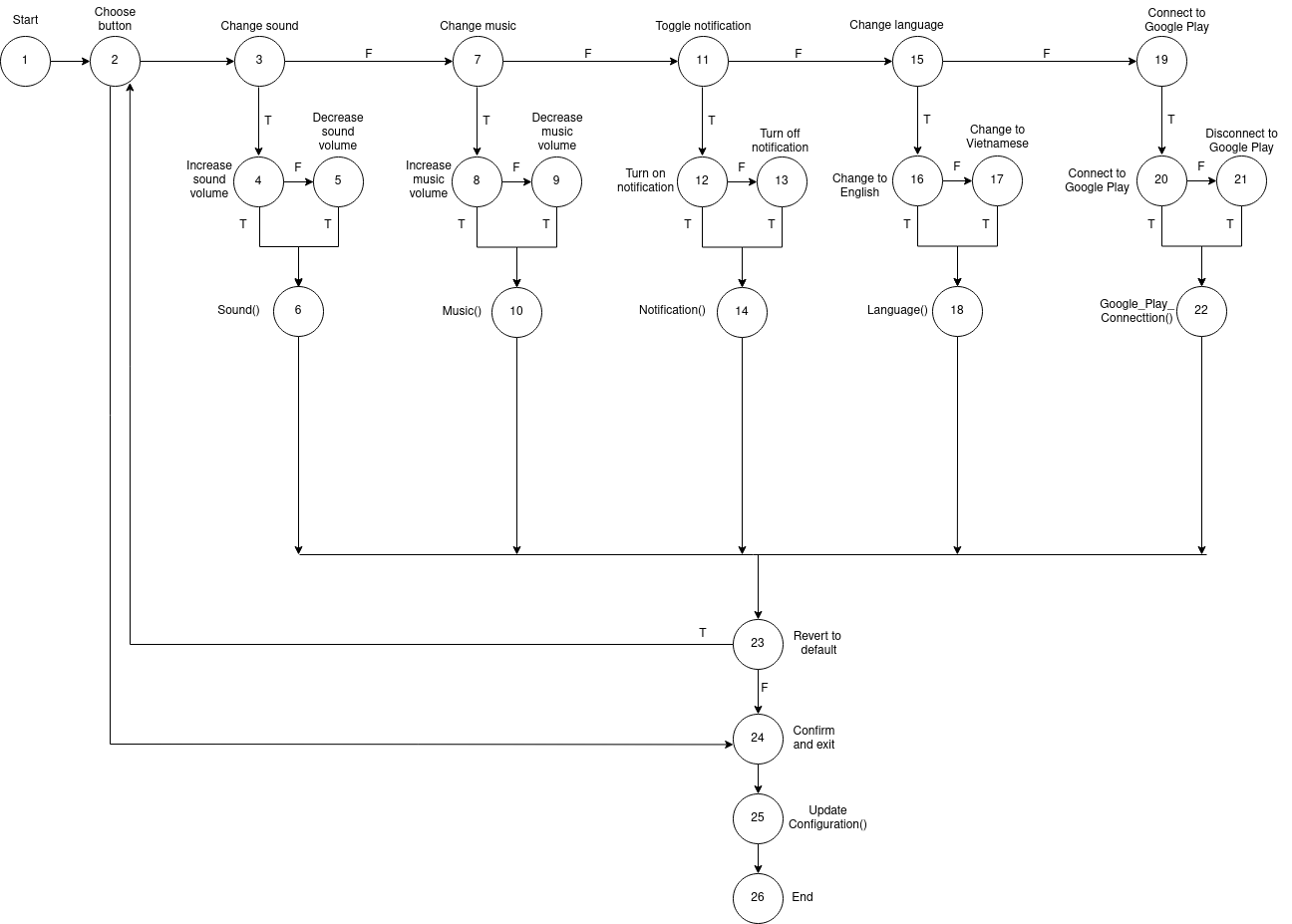
|  |  |  |
| --- | --- | --- |
| Test Case | Input Data | Expected Result |
| **User Input** |
| T001.2-TC1 | Users tap the left arrow. | The set of levels moves to the left 1 unit. |
| T001.2-TC2 | Users tap the back button. | Return to the main menu. |
| T001.2-TC3 | Users tap the right arrow. | The set of levels moves to the right 1 unit. |
| T001.2-TC4 | Users tap the level in the middle immediately. | Switch to the Game play screen for that particular level. |

### Test 2 - Master setting

|  |  |
| --- | --- |
| Test ID: | T002 |
| Test description: | Test the function of each setting options in the master setting interface. |
| Created by: | Nguyễn Quang Vinh |
| Date of creation: | 31 - 3 - 2023 |
| Date of review: | 1 - 4 - 2023 |
| Test priority: | High |
| Pre-requisite: | The Setting button on the Menu screen must be working and  the Display language is set to Vietnamese. |

#### Drawing the flow graph





#### Determining a basis set of independent paths

**TC1:** 1 - 2 - 24 - 25 - 26 (Choose the confirm and exit button immediately).

**TC2:** 1 - 2 - 3 - 4 - 6 - 23 - 24 - 25 - 26 (Increase sound volume).

**TC3:** 1 - 2 - 3 - 4 - 5 - 6 - 23 - 24 - 25 - 26 (Decrease sound volume).

**TC4:** 1 - 2 - 3 - 7 - 8 - 10 - 23 - 24 - 25 - 26 (Increase music volume).

**TC5:** 1 - 2 - 3 - 7 - 8 - 9 - 10 - 23 - 24 - 25 - 26 (Decrease music volume).

**TC6:** 1 - 2 - 3 - 7 - 11 - 12 - 14 - 23 - 24 - 25 - 26 (Turn on notification).

**TC7:** 1 - 2 - 3 - 7 - 11 - 12 - 13 - 14 - 23 - 24 - 25 - 26 (Turn off notification).

**TC8:** 1 - 2 - 3 - 7 - 11 - 15 - 16 - 18 - 23 - 24 - 25 - 26 (Change display language to English).

**TC9:** 1 - 2 - 3 - 7 - 11 - 15 - 16 - 17 - 18 - 23 - 24 - 25 - 26 (Change display language to Vietnamese).

**TC10:** 1 - 2 - 3 - 7 - 11 - 15 - 19 - 20 - 22 - 23 - 24 - 25 - 26 (Connect to Google Play).

**TC11:** 1 - 2 - 3 - 7 - 11 - 15 - 19 - 20 - 21 - 22 - 23 - 24 - 25 - 26 (Disconnect to Google Play).

#### Preparing test cases

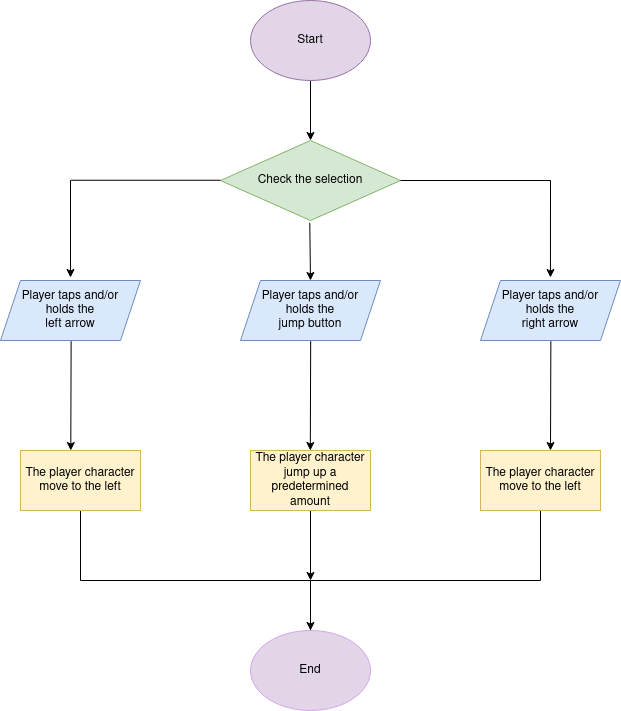
|  |  |  |
| --- | --- | --- |
| Test Case | Input Data | Expected Result |
| **User input** |
| T002-TC1 | Users tap on the  “Save and Exit” button. | The game saves all changes and switch from the master setting screen to the menu screen. |
| T002-TC2 | Users increase the sound volume by sliding the slider to the right. | Sound volume increases. |
| T002-TC3 | Users decrease the sound volume by sliding the slider to the left. | Sound volume decreases. |
| T002-TC4 | Users increase the music volume by sliding the slider to the right. | Music volume increases. |
| T002-TC5 | Users decrease the music volume by sliding the slider to the left. | Music volume decreases. |
| T002-TC6 | User taps on the right side (tilted left) of the notification button. | The game’s notification is turned on. |
| T002-TC7 | User taps on the right side (tilted left) of the notification button . | The game’s notification is turned off. |
| T002-TC8 | User taps on the “English flag’ icon. | The game’s display language is switched to English. |
| T002-TC9 | User taps on the “Vietnamese flag’ icon. | The game’s display language is switched to Vietnamese. |
| T002-TC10 | User taps on the  “Connect to Google Play” button. | Connection to Google Play is established . |
| T002-TC11 | User taps on the  “Connect to Google Play” button. | Connection to Google Play is terminated . |

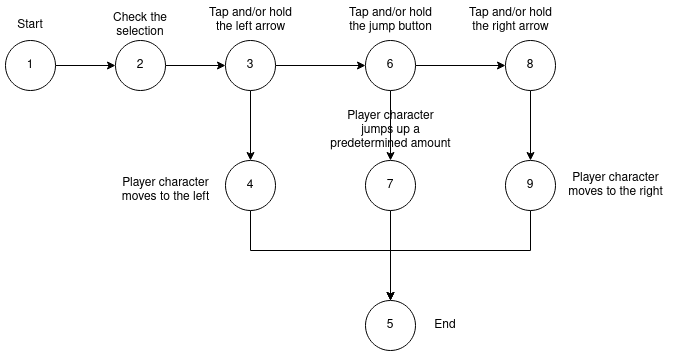
### Test 3 - Gameplay screen and in-game setting

#### Gameplay screen

|  |  |
| --- | --- |
| Test ID: | T003.1 |
| Test description: | Test the movement function of the Gameplay screen interface. |
| Created by: | Hà Quốc Huy |
| Date of creation: | 31 - 3 - 2023 |
| Date of review: | 1 - 4 - 2023 |
| Test priority: | High |
| Pre-requisite: | Play must enter any of the playable level. |

##### **Drawing the flow graph**





##### **Determining a basis set of independent paths**

**TC1:** 1 - 2 - 3 - 4 - 5 ( Character moves left ).

**TC2:** 1 - 2 - 3 - 6 - 7 - 5 ( Character jumps up ).

**TC3:** 1 - 2 - 3 - 6 - 8 - 9 - 5 ( Character moves right ) .

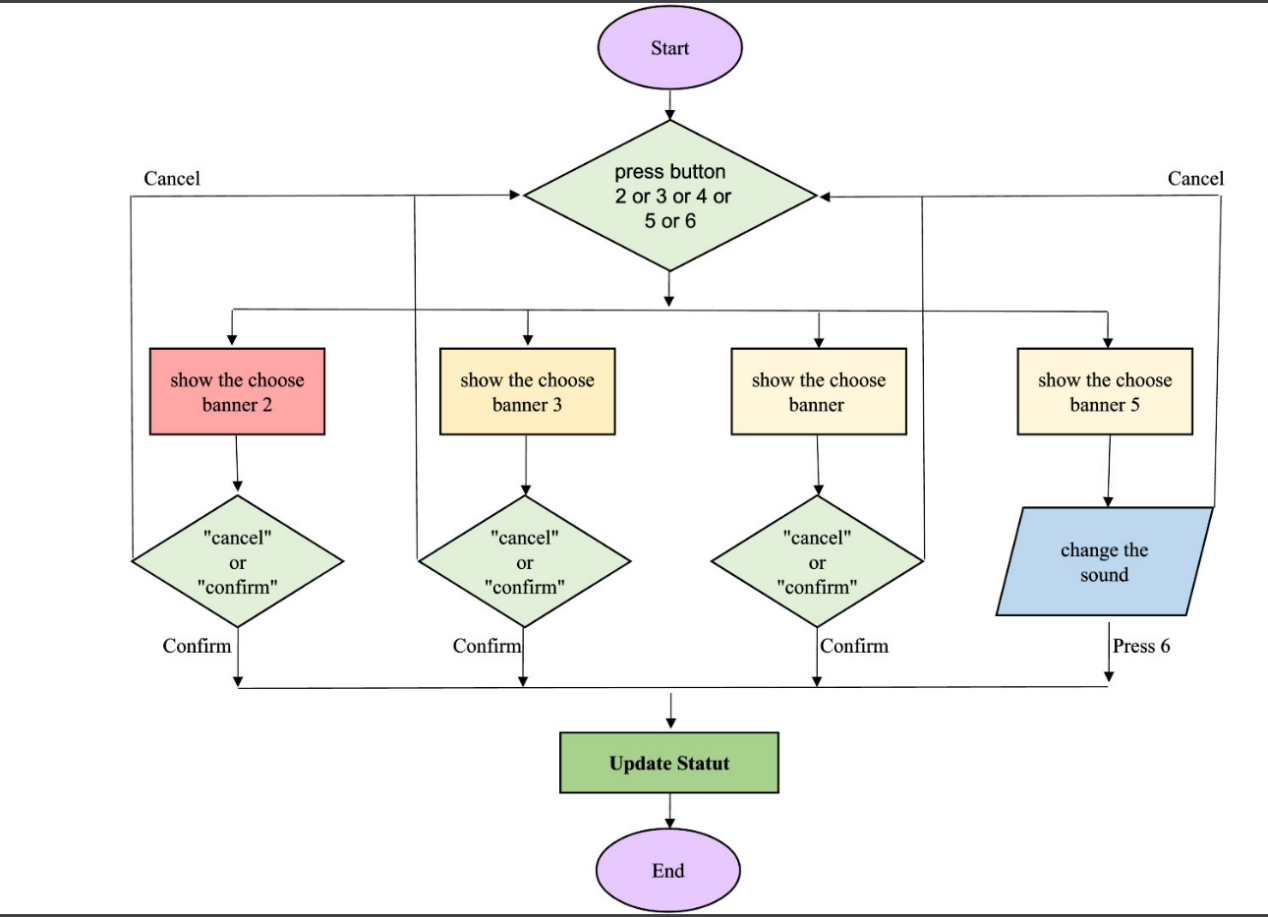
##### **Preparing test cases**

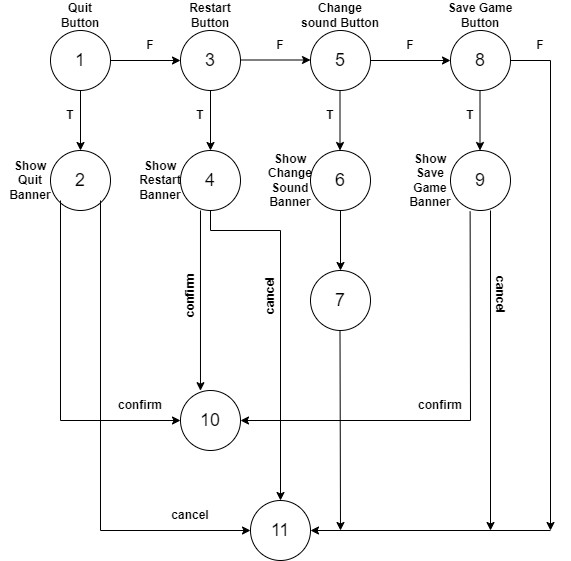
|  |  |  |
| --- | --- | --- |
| Test Case | Input Data | Expected Result |
| **User input** |
| T003.1-  TC1 | User tap the move icon left. | Character moves left. |
| T003.1-  TC2 | User tap the move. | Character jumps up. |
| T003.1-  TC3 | User tap the move. icon right. | Character moves right. |

#### in-game setting

|  |  |
| --- | --- |
| Test ID: | T003 |
| Test description: | Test the function of each setting options in the Play Screen interface. |
| Created by: | Hà Quốc Huy |
| Date of creation: | 31 - 3 - 2023 |
| Date of review: | 1 - 4 - 2023 |
| Test priority: | High |
| Pre-requisite: | The movement buttons must control the character in the right direction.  The buttons in the in-game settings must be able to perform the functions mentioned in the button name. |

##### **Drawing the flow graph**





##### **Determining a basis set of independent paths**

**TC1:** 1-2-10 ( Confirm to Quit the game ).

**TC2:** 1-2-11 ( Cancel to Quit the game ).

**TC3:** 1-3-4-10 ( Confirm to Restart the game ).

**TC4:** 1-3-4-11 ( Cancel to Restart the game ).

**TC5:** 1-3-5-6-7-11 ( Change the sound volumn ).

**TC6:** 1-3-5-8-9-10 ( Confirm to Save the game and return to menu ).

**TC7:** 1-3-5-8-9-11 ( Cancel to Save the game and return to menu ).

**TC8:** 1-3-5-8-11 ( Close the Setting in-game menu ).

##### **Preparing test cases**

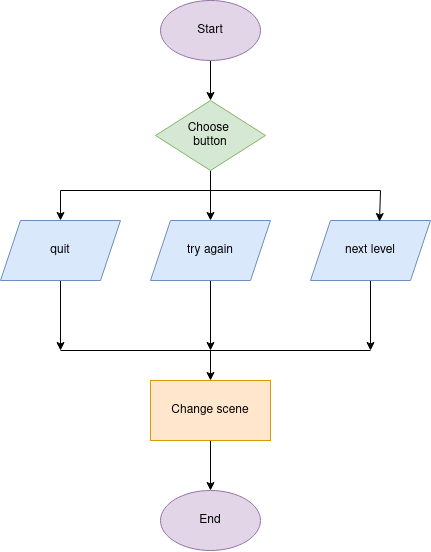
|  |  |  |
| --- | --- | --- |
| Test Case | Input Data | Expected Result |
| **User input** |
| T003-TC1 | User taps on the  “Confirm” button in banner Quit game. | Close the current level and return to the level selection menu. |
| T003-TC2 | User taps on the  “Cancel” button in banner Quit game. | Close the current banner and go back to the in-game settings. |
| T003-TC3 | User taps on the  “Confirm” button in banner Restart game. | Restart the current level. |
| T003-TC4 | User taps on the  “Cancel” button in banner Restart game. | Close the current banner and go back to the in-game settings. |
| T003-TC5 | User slide the buttons to change the sound. | Music volume decreases or increases. |
| T003-TC6 | User taps on the  “Confirm” button in banner Save game. | Save the current level and return to the level selection menu. |
| T003-TC7 | User taps on the  “Cancel” button in banner Save game. | Close the current banner and go back to the in-game settings. |
| T003-TC8 | User taps on the “X’ icon. | Close the Setting in-game and return to current level. |

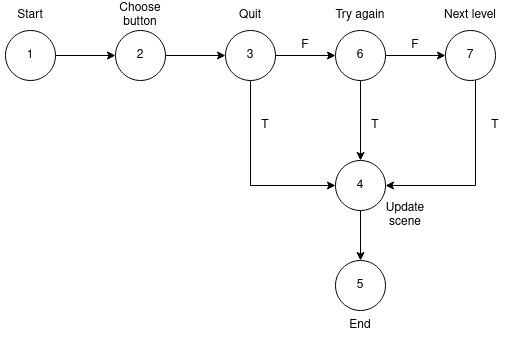
### Test 4 - Pop-up

#### Success pop-up

|  |  |
| --- | --- |
| Test ID: | T004.1 |
| Test description: | Test the function of the success pop-up. |
| Created by: | Lưu Hoài Vũ |
| Date of creation: | 31 - 3 - 2023 |
| Date of review: | 1 - 4 - 2023 |
| Test priority: | Low |
| Pre-requisite: | Player must successfully finish a game. |

##### **Drawing the flow graph**

****



##### **Determining a basis set of independent paths**

**TC1:** 1 - 2 - 3 - 4 - 5 (Choose the Quit button).

**TC2:** 1 - 2 - 3 - 6 - 4 - 5 (Choose the Try again button).

**TC3:** 1 - 2 - 3 - 6 - 7 - 4 - 5 (Choose the Next level button).

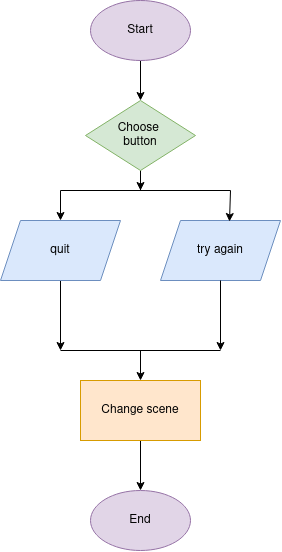
##### **Preparing test cases**

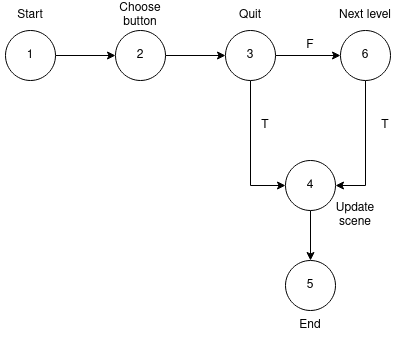
|  |  |  |
| --- | --- | --- |
| Test Case | Input Data | Expected Result |
| **User input** |
| T004.1 - TC1 | Player taps the Quit button. | The game switches to the level selection screen. |
| T004.1 - TC2 | Player taps the Try again button. | The game restarts the level and all the points the player accumulates so far in that level. |
| T004.1 - TC2 | Player taps the Next level button. | The game switches to the start of the next level. |

#### Failure pop-up

|  |  |
| --- | --- |
| Test ID: | T004.2 |
| Test description: | Test the function of the failure pop-up. |
| Created by: | Lưu Hoài Vũ |
| Date of creation: | 31 - 3 - 2023 |
| Date of review: | 1 - 4 - 2023 |
| Test priority: | Low |
| Pre-requisite: | Player must unsuccessfully finish a game. |

##### **Drawing the flow graph**

****



##### **Determining a basis set of independent paths**

**TC1:** 1 - 2 - 3 - 4 - 5 (Choose the Quit button).

**TC2:** 1 - 2 - 3 - 6 - 4 - 5 (Choose the Try again button).

##### **Preparing test cases**

|  |  |  |
| --- | --- | --- |
| Test Case | Input Data | Expected Result |
| **User input** |
| T004.2 - TC1 | Player taps the Quit button. | The game switches to the level selection screen. |
| T004.2 - TC2 | Player taps the Try again button. | The game restarts the level and all the points the player accumulates so far in that level. |