MULTIMEDIA UNIVERSITY®

TGD 2251

GAME PHYSICS

TRIMESTER 2, Year 2019/2020

Lecturer: Ts. Dr. Wong Ya Ping

10 Seconds Diver

Assignment #1 Report

Done by:

Name:	**Oo Jin Heng**
Student ID:	**1161303917**
Email:	**oojinheng@gmail.com**
Contact Number:	**012-3482077**

TABLE OF CONTENT

INTRODUCTION

This assignment project is to develop a physics based game in C++ which make use of game physics library Box2D and a multimedia library SFML. The delivered source code is capable to compile without error and the game will run without crashing in the middle.

The goal of the game design of this game project is to create an urgent, panic and fast paced experience to players. The main way of achieving this is to create a platformer game in an underwater environment with very brief limited time for players to achieve a certain goal. The title of the game is 10 seconds diver. An underwater diving game where requires players to reach the exit within 10 seconds. Players can extend their limited time by consuming air bubbles during the level to travel longer distances. A level will also contain various environmental hazards to stop players from progressing such as spikes, moving enemies, blocked path to create a fast paced puzzle solving gameplay experience to player.

This documentation will include the complete game design documentation for the game, several screenshots of complete product, a user manual which consist of compilation instruction and game tutorials, all origin source of game assets.

GAME DESIGN DOCUMENTATION

Design Definition

The goal of the game design is to give player the experience of time constraint. Players will have a sense of urgency and panic feeling during a level when they are trying to achieve the goal within the limited time frame. In addition, the controls in the game must feel easy and smooth for players to make them believe that the level is possible with numerous attempts and training. Hence, the game is meant for the players who enjoy achieving certain goals or completing difficult challenges.

To achieve the intended game experience for the players, this game is decided to be a fast paced underwater platformer game with 10 seconds time limitation for the players to reach the exit of the level. All player movement in this underwater diving game is based on a specific environment physics to simulate the feeling of swimming underwater. Thus, players are meant to master the controls and understand the physics to be able to swiftly move from point A to B within the 10 seconds time frame.

Core Mechanism

The core mechanism of the game is to swim from start point to the end point of a level within 10 seconds. Players can control to swim left or right but will slowly sink to the bottom over time. Player can also move downward to sink significantly faster than normal. Lastly, players can swim upward with a moment of thrusting force pushing players slightly higher. Players need to move through the maze of walls to get to the exit.

Player will always fail the level after 10 seconds ran out without getting to the exit. However, players can consume an air bubble to reset the 10 seconds timing for longer distance. Air bubbles will usually appear at longer level. Other than time constraints, there are also traps and enemies throughout the level. These obstacles are very deadly as they will kill the player immediately upon contact.

Extended Mechanism

Other than achieving the common goal of getting from start to exit within 10 seconds, coins are added into the game to provide advanced players more challenges to collect them for getting a higher score in the level.

There will be movable walls which allow the players to push them over and also switch and door mechanism which allow the players to toggle which door to appear or disappear. The detail of switch and door will explain in the game object section.

Traps and enemies can also be moving in a certain pattern to provide more challenge to players.

Game Objects

Player

Player is the main entity that *players* controls throughout the game. The main entity need to consume air bubble to replenish the air before the 10 seconds timing run out. In addition, the main entity also vulnerable to traps and enemies object which will kill it upon collision occurs.

Wall

Wall is a static object that block player entity from moving across. The wall is mainly used for making the shape or outline of each level.

Movable Wall

Movable wall is a dynamic wall that allow player entity to push over. Player entity will need extra force and time to push them. Movable wall follows the world physics.

Air Bubble

Air bubble is a consumable item in the game which reset the 10 seconds timing by replenishing the air for player entity. Air bubble acts as a bridge across sections in the core mechanism for longer level.

Coin

Coin is a collectable item to award players extra score upon completion of each level. Coins are usually place in odd position in a level to give players who are willing to collect them extra challenges.

Trap and Enemy

Trap and enemy are the deadly obstacles in the level which will kill the player entity immediately upon contact. They can be static or moving in a certain pattern.

Switch and Door

Switch and door mechanism consists of a switch and two types of doors. A level can consist of multiple switches and doors but there can only be one type of door active at a time. The functionality of the switch is to toggle the activation between these two types of door.

Score System

The score of each level is decided by the maximum time and clear time of the players. The maximum time of each level depends on the number of air bubbles presented. Getting each coin in a level can increase the max score value by 25000. Below is the basic calculation formula for the score, assuming the unit for time is seconds:

$$\begin{aligned}
&\mathit{maxTime} = \mathit{airBubbleCount} * 10 \\
&\mathit{maxScore} = 1000000 + (\mathit{coinCount} * 25000) \\
&\mathit{score} = 1000000 * (\mathit{clearTime} / \mathit{maxTime})
\end{aligned}$$

USER MANUAL

This section is also written on the project's GitHub repository:
https://github.com/Tajam/3d-game-programming
Please visit the GitHub repository if you prefer to read markdown version of the documentation.
Moreover, the GitHub version also contains various useful links.

Compilation

System Requirements

Operating System: Windows 10

Compiler: MinGW Build 7.3.0 (32-bit)

The G++ compiler version must match the requirement as SFML depends on it.

Disclaimer: The listed requirements above are based on the only tested environment during the development. Machine that meet the requirement or not, might or might not be able to build the project or run the program successfully. There is no guarantee.

Building Executable

In the command line terminal, move to the root folder of the project and execute "*build*". The built executable can be found in *"/bin"* folder. The executable must remain in the folder to run. The game can also start by executing *"run"* at the root folder.

The executable file building mechanism is simple. The build batch file iterate through all the hardcoded path and execute another batch file that contains command for compilation to create object files. Some sub-folder also contain the same type of batch file to create object files for its child folder's source code. Lastly, all the object files are linked together to make the final executable.

Game Tutorial

Controls

Scenes	**Controls**
Main menu	- Use up and down key to navigate through items - Press enter to select the item
Level selection menu	- Use left and right key to select levels - Use up and down key to toggle between level selection and back to menu option - Press enter to start the level or select the item
Playing the level	- Use left and right key to move left or right - Press space bar key to swim upward - Press down key to sink faster
Game over menu	- Use up and down key to navigate through items - Press enter to select the item
Game clear menu	- Press enter to confirm - If player start the level in level selection, the game will send player to credit scene - If player start the level in start, the game will move players to the next level

Game Rules

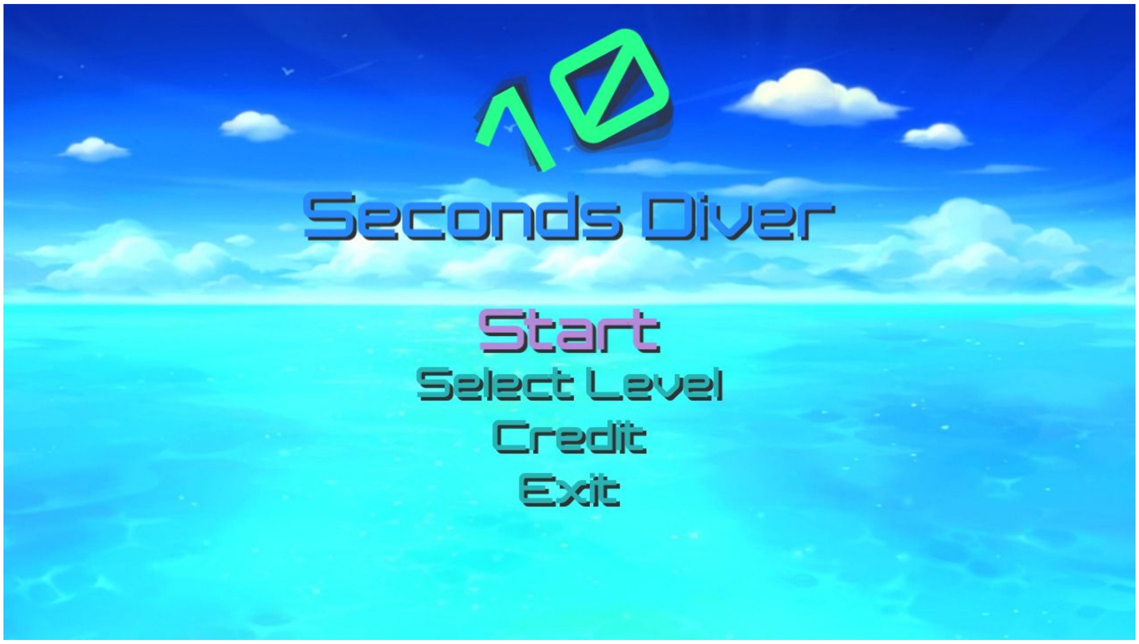
Each level will start within a 2 seconds margin time. Once the level starts, the players have only 10 seconds of time to swim to the exit of the level. If the players failed to do so, the whole level needs to restart. Otherwise, players will move on to the next level.

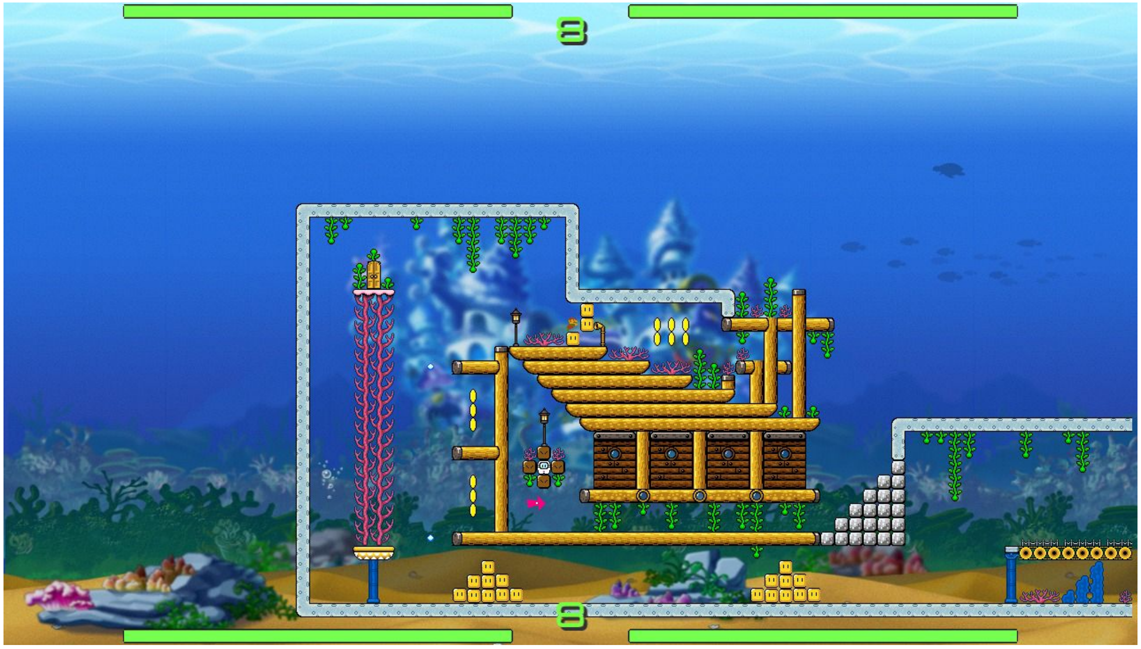
Consuming an air bubble will reset the 10 seconds timing which allow the players to travel longer distances in the level. Moreover, collecting a coin will grant players more scores upon completion of the level.

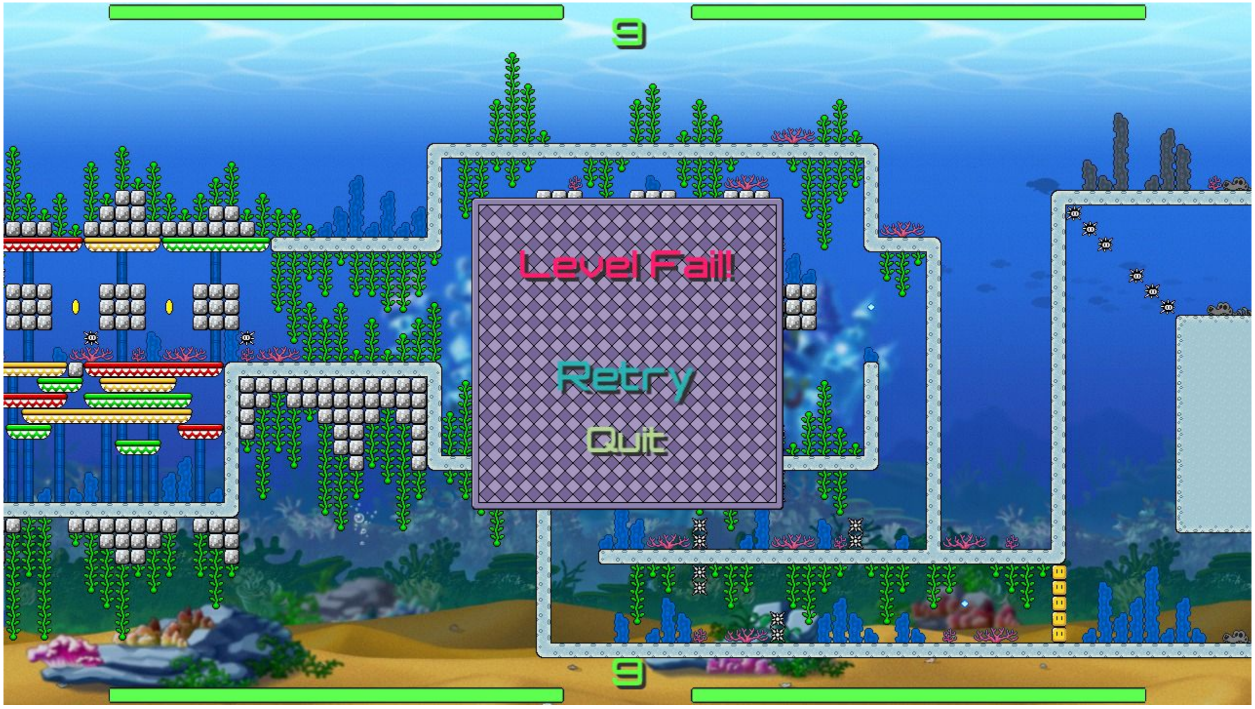
All the traps and enemies in game will kill players immediately upon contact. So beware.

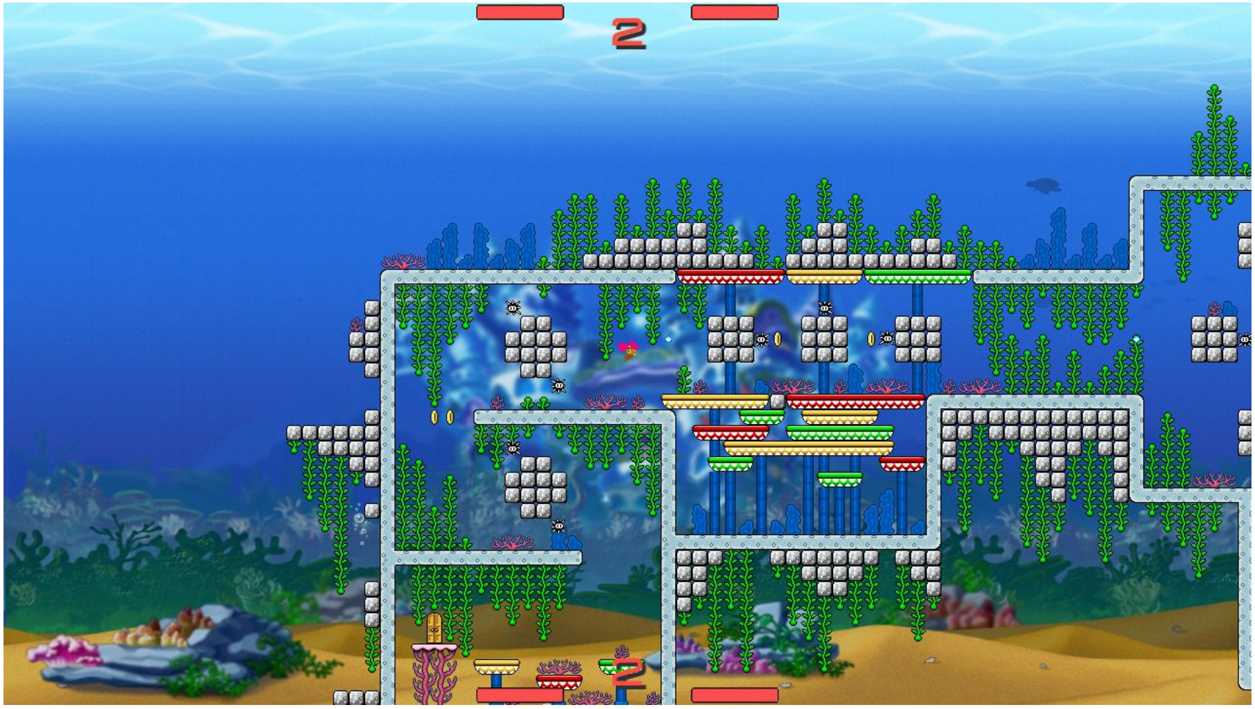
SHOWCASE

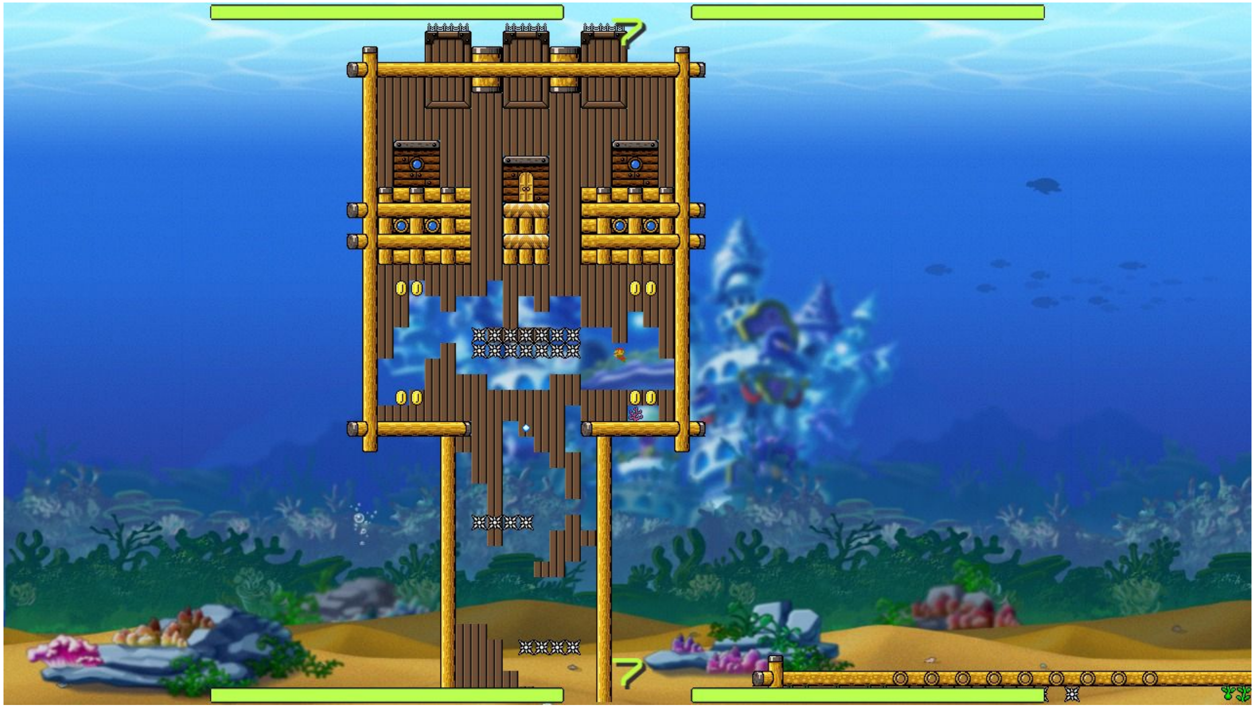
Screenshots

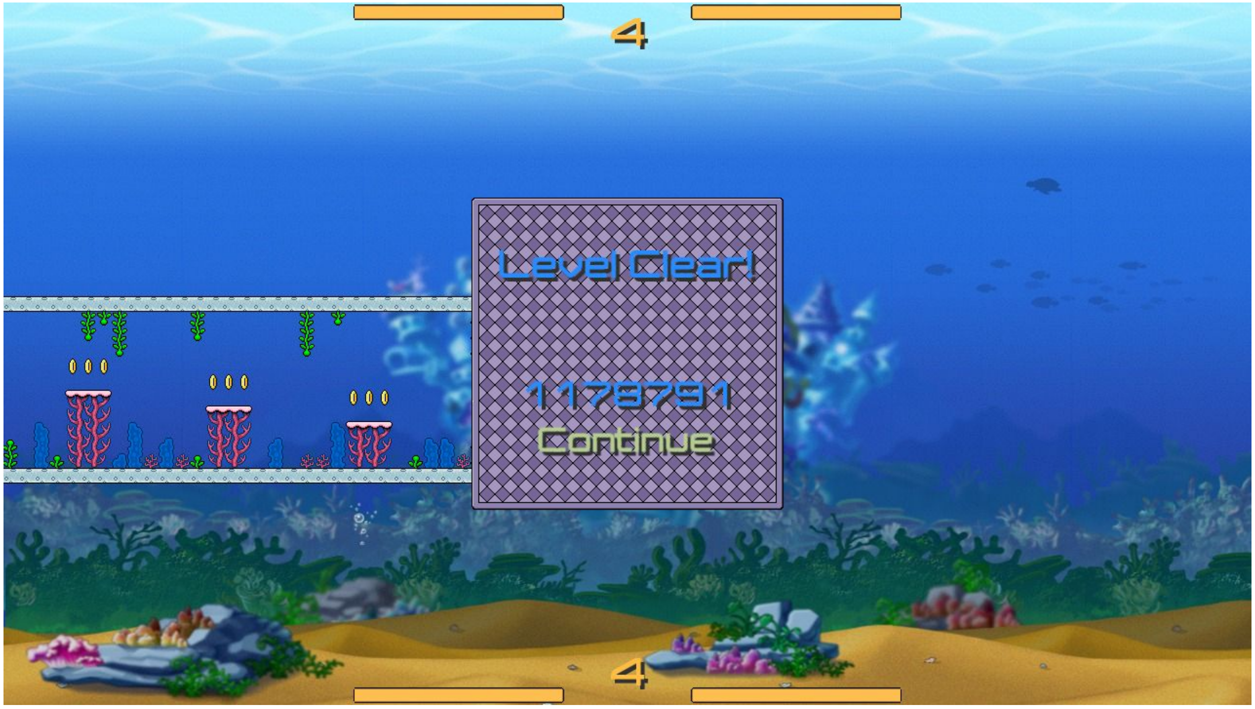





9
Level Fail!
Retry
Quit
9


2
2


7
7


4
Level Clear!
1178791
Continue
4

Taking Screenshot

The game program have the ability to take screenshot internally. Press *"F12"* anywhere during the game to capture a screenshot. The screenshot will be saved to *"./ss"* folder.

Demo Video

The link below is a full playthrough of each level video of the game:

https://drive.google.com/open?id=1JgWyCDVektfZw3ukFVKKOuuWQTNN6uOq

Please do not hesitate to contact me if the video could not play or the link is broken.

REFERENCES

This section is also written on the project's GitHub repository:
https://github.com/Tajam/3d-game-programming/tree/master/ref
Please visit the GitHub version of the documents for all the clickable links.

The file paths are relative to the project root directory.

Used for background textures. *"./assets/textures/main-bg.jpg"*. Taken from website, http://forums.maplestory2.nexon.net/discussion/27656/beautiful-backgrounds-to-choose-for-your-home

Used for background textures. *"./assets/textures/level-bg.jpg"*. Belongs to Nexon, Wizet.

Used for game sprite. *"./assets/textures/objects.png"*. Belongs to Nintendo, Super Mario World.

Used for game sprite. *"./assets/textures/tiles.png"*. Belongs to Nintendo, Super Mario World.

Used for game sprite. *"./assets/textures/panel.png"*. Belongs to Nintendo, Super Mario World.

Used for background music. *"./assets/sounds/main-bgm.ogg"*. Belongs to Nexon, Wizet.

Used for background music. *"./assets/textures/level-bgm.ogg"*. Belongs to Nexon, Wizet.

Used for sound effects. *"./assets/sounds/*.wav" (Every .wav files)*. Taken from websites, https://assetstore.unity.com/packages/audio/sound-fx/free-casual-game-sfx-pack-54116, and, http://www.gameburp.com/free-game-sound-fx/

Used for user interface font. *"./assets/fonts/Jupiter.ttf"*. Taken from websites, http://www.fontriver.com/font/isl_jupiter/

Copyright Disclaimer

Copyright Disclaimer Under Section 107 of the Copyright Act 1976, allowance is made for "fair use" for purposes such as criticism, comment, news reporting, teaching, scholarship, and research. Fair use is a use permitted by copyright statute that might otherwise be infringing. Non-profit, educational or personal use tips the balance in favour of fair use.

This game is developed purely for educational purposes, and have no intention of making profit out of the game. All credits are given to the assets owner and otherwise, the source of the assets are provided.