MULTIMEDIA UNIVERSITY ®

FACULTY OF COMPUTING AND INFORMATICS

TGD2251 Game Physics

TRIMESTER 2 2016/2017

PROJECT #1

Report

Lecture Section : TC01
Tutorial Section : TT01

for:

Dr. Wong Ya Ping

from:

Student ID	Name	Email Address	Phone No.
1142700937	Arsyad bin Anuar	rsyad126@gmail.com	019-3456645

Date: 1.1.2017

Introduction

This is a physics-based game using SFML library as the rendering engine and MinGW GCC g++ as the C++ compiler. The purpose of creating this program is to enhance our knowledge in game physics and to deepen our learning in C++ programming.

The idea of creating the game is referred from a simple classic game where there is a moving ball and a bar that act as the bat for the ball. In the original game, the player will control the bar to move left or right to prevent the ball from falling at the bottom of the screen. If the ball hits the bar, it will bounce on the opposite direction as well as it hits the side of the screen. Usually, there will be blocks of bar assembled at the top of the screen waiting for the ball to smash them to pieces. The player will win the game if all the blocks have been smashed without the ball falling to the bottom of the screen.

The control for this game is just left and right arrow keys and Esc key. Left arrow is to move the bar to the left and right arrow is vice versa. Esc key is to navigate to main menu screen. Other than using keyboard keys, mouse button is also being used in navigating from one screen to another.

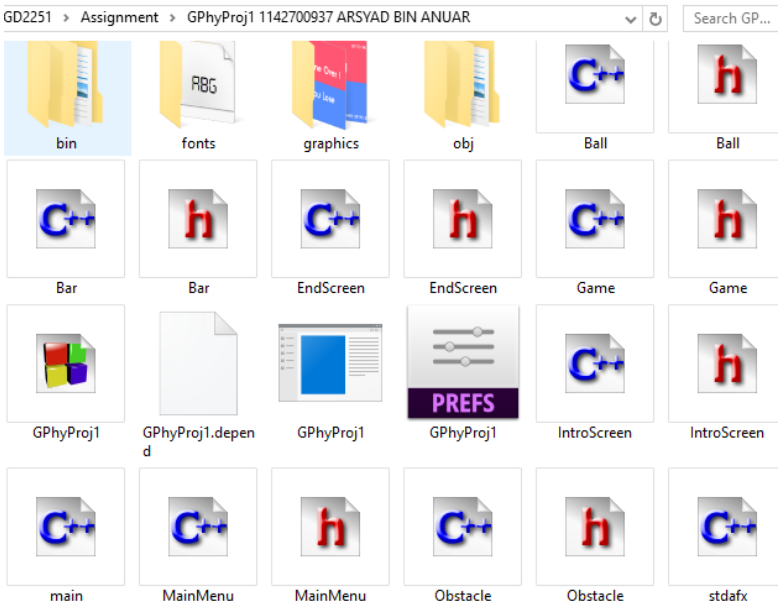
In this version of the game, the winning condition is modified to where the player needs to get the ball to the top of the screen without the ball falling to the bottom of the screen. Of course there will be obstacle in order to win the game. There are three obstacles with different colors. The red colored obstacle needs to be hit by the ball 3 times, the magenta colored obstacle needs to be hit 2 times and the white colored obstacle only 1 time. Once all the obstacles are cleared, the ball now can reach the top easily and the player can win the game.

The losing condition is when the player has run out of his/her lives. In the starting of the game, the player will be given 3 lives. Each life will lost if the ball falls to the bottom of the screen but the ball will be reset and the game will still run. Only if all the lives are lost, then the game will end with the player losing. Also, beware of the speed of the ball as it is getting faster and faster every time it hits the obstacle. The player need to be quick in following the ball to go to where it will fall down.

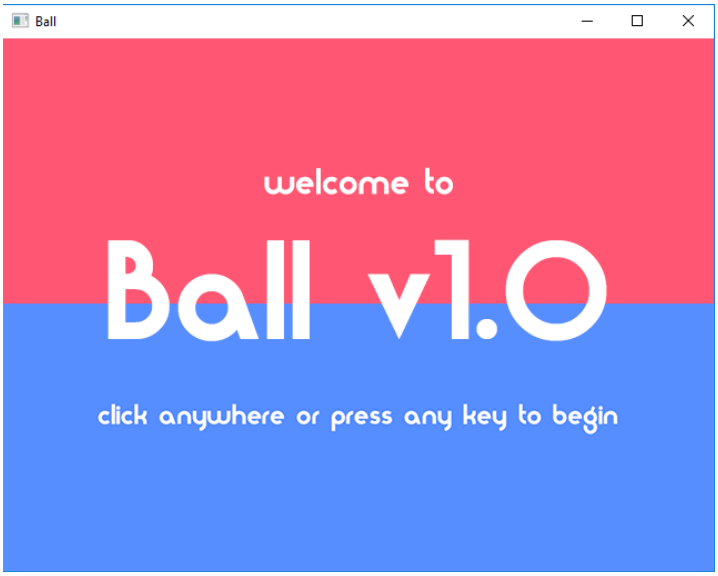
User Manual / Instructions

GPhyProj1
1142700937
ARSYAD BIN
ANUAR

Firstly, unzip all the necessary files inside any folder.

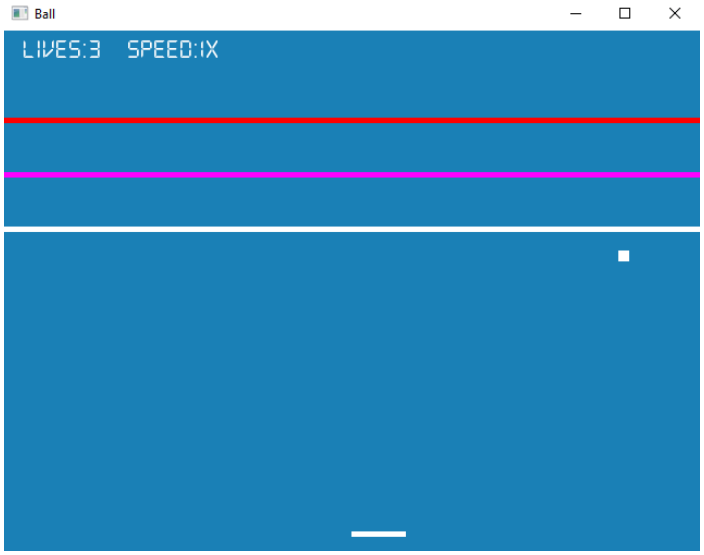


After unzipping the files, open the **GPhyProj1.cbp** to compile the project using CodeBlocks. Otherwise, run the **GPhyProj1.exe** to run directly without compiling the project.

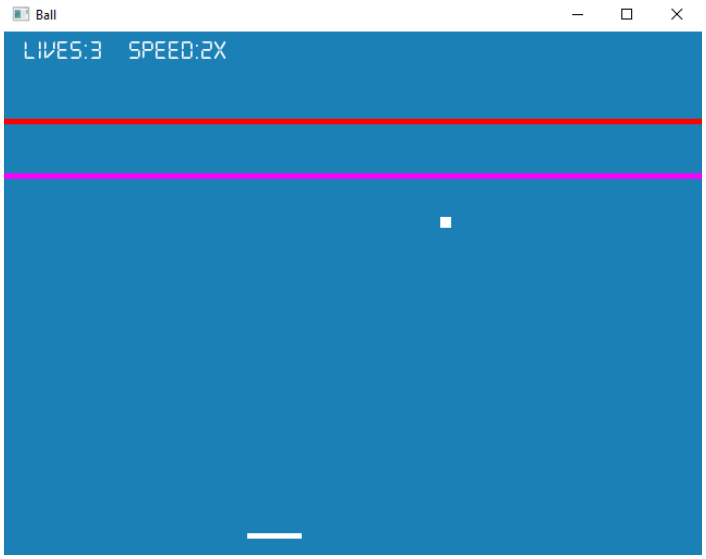


The intro screen of the game will be opened as the first game window. Then click anywhere or press any key to navigate to the next screen which is the menu screen.

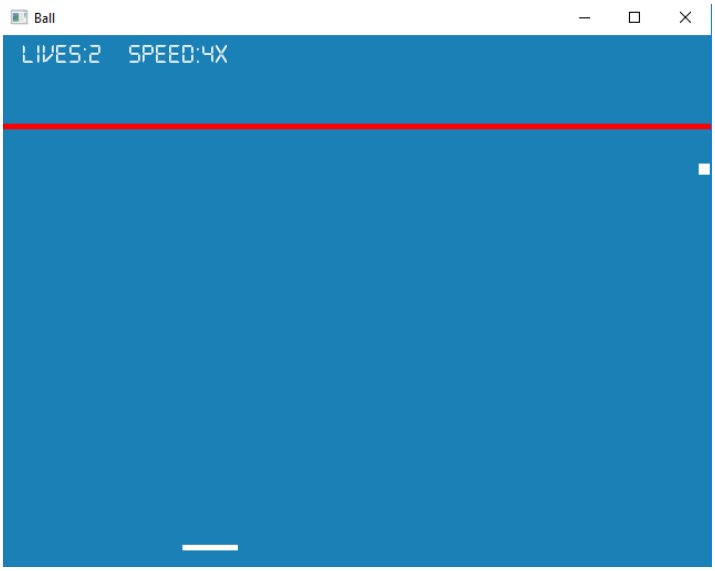
Next is the menu screen. The menu screen has only 2 clickable buttons which are Play button and Exit button. If player clicks on Play button, the program will navigate to the next screen which is the play screen. If player clicks on Exit button, the program will be closed immediately. There are also a quick instructions on the menu screen as a guidance for the player to learn the flow of the game.



This is the play screen as mentioned earlier. The player will be able to control the white bar at the bottom by using keyboard keys which are left arrow key and right arrow key to move the bar left and right. The player can navigate back to the main menu by pressing Esc key on the keyboard. Also, there is a constantly moving ball which the player needs to pay attention to. If the ball falls down, lives will be lost and the player will lose the game. In order to win the game, the player needs to bring the ball to the top of the screen while removing the obstacle ahead. There are 3 obstacles with 3 different colors. Each obstacle has different number of lives.

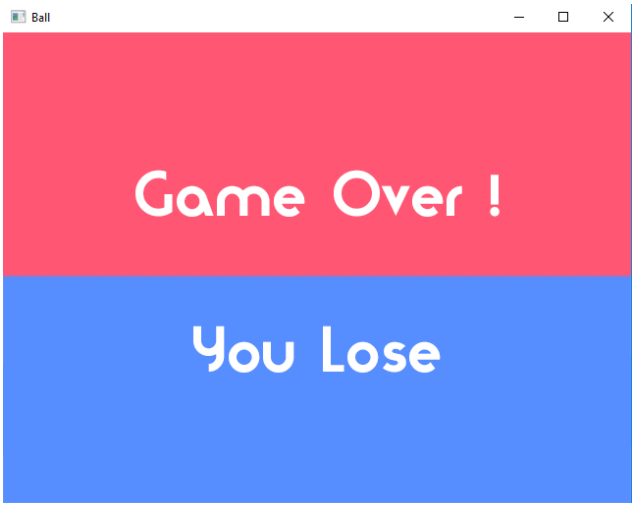


Like in the figure above, the player managed to destroy the first obstacle which is the white obstacle that requires only 1 hit from the ball to be destroyed. Bear in mind, every time the ball hits the obstacle, the speed of the ball will increase by 1.



Now we only have one remaining obstacle which is the hardest. This red obstacle needs to be hit by the ball 3 times in order to be destroyed.

At last, if all the obstacles are cleared and the ball has reached the top, the player wins the game and this end screen will come out saying “Congratulations !, You win”. If the player wants to play again, just click anywhere at the program screen or press any key on the keyboard. It will navigate back to the main menu.



It will be a different story if the player keeps getting the ball falls down until all lives are lost. The game will end with the player losing the game and this end screen will come out saying “Game Over !, You Lose”. If the player wants to try again, just click anywhere at the program screen or press any key on the keyboard. It will navigate back to the main menu.

Acknowledgement

I would like to express my special thanks of gratitude to my lecturer, Dr. Wong Ya Ping who gave me the golden opportunity to do this wonderful project, which also helped me in doing a lot of research and I came to know about so many new things I am really thankful to him.

Secondly, I would also like to thank my friends and my coursemates, especially to Akmal Safwan and Amirul Fahmy, who helped me a lot in finalizing this project within the limited time frame.

Next, I want to thank to the Stackoverflow community for being really helpful in answering my question that has been troubling me in making this project. Also, thanks to the SFML tutorial in making games has helped me a lot in understanding how SFML libraries works.

References

Game From Scratch C++ and SFML Edition! (n.d.). Retrieved from GameFromScratch: http://www.gamefromscratch.com/page/Game-From-Scratch-CPP-Edition.aspx

Horton, J. (n.d.). *Coding a simple Pong game with SFML*. Retrieved from GameCodeSchool: http://gamecodeschool.com/sfml/coding-a-simple-pong-game-with-sfml/

Parrish, E. (n.d.). *How to Install SFML in Code::Blocks for Windows*. Retrieved from Ed Parrish: http://www.edparrish.net/common/sfmlcb.html

SFML and Code::Blocks (MinGW). (n.d.). Retrieved from SFML-Dev: http://www.sfml-dev.org/tutorials/2.4/start-cb.php