Name: Len Sreyorn  
ID: 12190175  
Department: Computer Engineering

**Project2: Breakout Game with SFML**

1. Introduction

The game was assigned to build with SFML (**Simple and Fast Multimedia Library)** which is an   
open source library supporting GUI and easy to use. The game was built using object oriented programming methods and interaction with SFML functions.

1. The Game

To play the game, player has to use mouse to move the paddle and the paddle make to ball move up to break the bricks.

1. Code

#include <SFML/Graphics.hpp>

#include<Windows.h>

#include<string>

#include<iostream>

using namespace std;

using namespace sf;

const float BALL\_SPEED = 5.0;

Font font;

Text text;

string status;

//Class Paddle

class Paddle :public RectangleShape {

float init\_x, init\_y;

public:

Paddle(float x, float y) :init\_x(x), init\_y(y) {

setSize({ 80,20 });

setPosition(x, y);

setFillColor(Color(0, 255, 64));

setOrigin(0, 0);

}

void update(int x) { setPosition(x, init\_y); }

};

//class Ball

class Ball : public CircleShape {

public:

float speedx = BALL\_SPEED, speedy = BALL\_SPEED;

Ball(float x, float y) :CircleShape(12.0) {

setPosition(x, y);

setFillColor(Color(255, 128, 0));

setOrigin(0, 0);

}

void update();

bool isIntersecting(Paddle &paddle);

void handleCollision(Paddle &paddle);

};

//Class Brick

class Brick :public RectangleShape {

public:

bool deleted = false;

Brick(float x, float y) {

setSize({ 60,20 });

setPosition(x, y);

setFillColor(Color::Yellow);

setOrigin(0, 0);

}

bool isIntersecting(Ball &ball) {

return getGlobalBounds().intersects(ball.getGlobalBounds());

}

void handleCollision(Ball &ball) {

if (!isIntersecting(ball)) return;

deleted = true;

}

};

//Update Function

void Ball::update() {

move(speedx, speedy);

if ((getPosition().x) < 0)

speedx = BALL\_SPEED;

else if ((getPosition().x + 2 \* 20) > 800)

speedx = -BALL\_SPEED;

if (getPosition().y < 0)

speedy = BALL\_SPEED;

else if ((getPosition().y + 2 \* 20) > 600)

speedy = -BALL\_SPEED;

}

bool Ball::isIntersecting(Paddle &paddle) {

return getGlobalBounds().intersects(paddle.getGlobalBounds());

}

void Ball::handleCollision(Paddle &paddle) {

if (!isIntersecting(paddle)) return;

speedy = -BALL\_SPEED;

if (getPosition().x < paddle.getPosition().x)

speedx = -BALL\_SPEED;

else

speedx = BALL\_SPEED;

}

int main() {

//for displaying text

/\*if (!font.loadFromFile("../OpenSans\_Bold.ttf")) {

cout << "Cannot open file";

}

status = "Use mouse to move!";

\*/

Ball ball = { 800.0 / 2,600.0 / 2 };

Paddle paddle = { 800 / 2,550 };

vector<Brick>bricks;

for (int x = 0; x < 10; x++)

for (int y = 0; y < 5; y++)

bricks.push\_back(Brick(x\*(60 + 3) + 20, y\*(20 + 3) + 40));

RenderWindow window(VideoMode(800, 600), "My window");

window.setFramerateLimit(60);

while (window.isOpen()) {

window.clear(sf::Color::Blue);

Event event;

while (window.pollEvent(event)) {

if (event.type == sf::Event::Closed)

window.close();

}

sf::Vector2i position = sf::Mouse::getPosition(window);

paddle.update(position.x);

ball.handleCollision(paddle);

for (auto &brick : bricks)

brick.handleCollision(ball);

bricks.erase(remove\_if(begin(bricks), end(bricks), [](Brick &b) {

return b.deleted; }), end(bricks));

ball.update();

window.draw(ball);

window.draw(paddle);

for (auto&brick : bricks)

window.draw(brick);

window.display();

// display text

/\*text.setFont(font);

text.setString(status);

text.setCharacterSize(20);

text.setPosition(10, 100);\*/

}

}

(P.S. I was trying to display text and update score after the game ends, but I could not get what I wanted, so I ended up using the code that professor provided)

4. Image of result output

