

**University of Dhaka**

**Department of Computer Science and Engineering**

**CSE-1211, Fundamentals of Programming Lab**

**1st Year 2nd Semester**

**Session: 2015-2016**

**Lab Project: NINJA OF CSEDU**

**Submitted by:**

**1. Md. Musfiqur Rahman**

**Roll: 05**

**2. Fairuz Nawer Meem**

**Roll: 17**

**Submitted to:**

**1. Md. Shiplu Hawlader,Lecturer,Department of Computer Science and Enginneering**

**2. Mr. Hasnain Heickal,Lecturer,Department of Computer Science and Enginneering**

**3. Mrs. Ismat Rahman,Assistant Professor,Department of Computer Science and Enginneering**

**4. Iffat Anjum,Lecturer,Department of CSEDU**

**5. Tamal Adhikary,Lecturer,Department of CSEDU**

**Submission Date: 24.11.2016**

Table of contents:

|  |  |  |
| --- | --- | --- |
| Serial | Topic | Page Number |
| 1 | Introduction | 4 |
| 2 | Project name | 4 |
| 3 | Project introduction | 4 |
| 4 | Game objectives | 4 |
| 5 | Game outline | 5 |
| 6 | Challenges for the gamer | 6 |
| 7 | Game features | 6 |
| 8 | Coding challenges | 11 |
| 9 | Graphical interface | 12 |
| 10 | Flowchart | 16 |
| 11 | Project overview | 16 |
| 12 | Conclusion | 16 |
| 13 | Source code | 17 |
| 14 | References | 98 |

**Introduction:** This project was assigned so that we can try out our own logical and implementation skills of basic knowledge of C/C++ programming language in a practical manner. The project is intended to make the students familiar with basic level graphics and to make the students interested about future game development.

**Project Name:** Ninja of CSEDU

**Project Introduction:** This project is a ninja running game. It is a single player game consisting of two unique levels. This project reflects our CSEDU life a little. The problems we face here while studying have been used as obstacles in this game. The player has to overcome these obstacles with bravery just like a ninja. So we choose the name “Ninja of CSEDU” for this game.

**Game Objectives:** The game was designed with the aim to keep the player busy always. As the player will continuously be moving in this game, the player has to be alert all the time and so the player will get little time to get bored. The main objective of this game was to provide the player an entertaining experience and also to make the game as addictive and eye-pleasing as possible. High quality interactive menus, submenus as well as clean and flicker-free game play environments were created with great care and efforts to achieve that certain goal.

**Game Outline:** This game is mainly one kind of running game where the player has to move forward and face different obstacles. It has some additional features which add more spontaneity to the game play. The player is given the freedom of choosing any of the two levels for playing from the level option. This option allows the player to have two different experiences. In the first level the player has to face obstacles related to the difficulties we face in the very first year. He/ She has to jump over or slide down the obstacles. In the second level the difficulty level increases. The continuous movement in the game gives the player a taste of interest while different unique obstacles give the player a taste a challenge. The player has to collect the coins while running. Again the player has to throw weapons along with superpowers which he/she can achieve throughout the levels. In the end of the second level the player has to face an enemy which is compared to ACM ICPC contest. At the beginning of the game the player will have five lives which will be decreasing if the player cannot overcome the obstacles or the weapons thrown by the enemy touch the player. The game is over if the number of lives becomes zero or if the player completes the level.

**Challenges for the gamer:** This game will provide the player with some challenges. The player has to be aware of the fact of time management. The obstacles will be coming at a random time with a random speed. So the player has to be alert always to avoid these obstacles. He/ She has to make decisions quickly whether to jump over or slide down. The player has to throw weapons towards the enemies. Again the number of the knives the player can use is limited. Moreover, he/ she can achieve some superpowers throughout the levels. This superpower is also limited. So the player has to be very careful about using this weapons or superpowers. In the second level the player can also move forward or backward using the keyboards. So controlling the player movement as well as destroying the enemies and avoiding the collision with the obstacles will make the game harder for the player.

**Game Features:**

Main Features:

1. Welcome screen and main menu: The player will have a welcome screen along with the main menu. The main menu section will be divided into these parts-

a. New Game: By clicking on this tab the player can start playing the game from level 1.

b. Levels: By clicking on this tab, a page with the following two tabs will appear-

(i) Level-1: 1st year - This will lead the player to start the game from level 1.

(ii) Level-2: 2nd year - This will lead the player to start the game from level 2.

c. Help: By clicking on this tab, a page will appear with the necessary instructions of playing the game.

d. Scoreboard: By clicking on this tab, a player can see the top five scorer’s name along with their scores.

e. Quit: The player can easily exit the game by clicking on this tab.

2. Movement of the player: In level 1, the player will continuously be running but in level 2 the player can control the player’s movement with the help of keyboard. That means the player can move forward or backward using the keyboard while running in the meantime of running. This will help the player to achieve his goal within shorter time.

3. Extra lives: In the beginning of the game the player will have five lives which will be decreasing by time if the player has a collision with obstacles or if any weapon thrown by the enemies hit the player. So he/ she has to complete the full levels with these five lives. But at the end of the second level when the player has to face the strongest enemy (which is shown as a representative of ACM ICPC in this game), his number of lives will be back in five again even after he has less than that.

4. Existence of levels: The game allows the player to advance to two different levels which ensure the gradual change of difficulty and environment. By avoiding all the obstacles of the first level successfully the next level can be accessed which adds variance and spontaneity to the game play.

5. Variance of difficulties: The gradual increase of difficulties is introduced as the gradual development of levels. The number of obstacles which are to be avoided or destroyed increase with the advancement of levels and newer enemies are to be faced with time allowing a gradual change of difficulties.

6. Zero-flicker game play: A game should be eye soothing so that it can hold the player’s attraction and interest. While working on the project using BGI, we faced some problems among which flickering was the most common one. So we shifted to SFML (Simple and Fast Multimedia Library) to avoid flickering.

Special Features:

1. Superpowers: In this game the player can achieve some superpowers while running towards his destination. This superpowers will help him/ her to destroy his enemies easily than before.

2. Teleportation: In level 2 there is a teleportation system which will help the player move a very long distance within little time.

Additional Features:

1. Use of keyboard: The player has to play this game with the help of some basic keys of keyboard. As an example the player has to use the up arrow key to jump over the obstacles.

2. Use of mouse: The player has to use mouse movements to access to the game through the main menu along with the level menu. Different menus in this game work in an animated manner. The player can see a button getting highlighted if the mouse cursor goes to that area. This makes the visuals more attractive and adds an interactive and animated feeling.

3. Option to pause: The game play can be paused easily by pressing Esc key in the middle of the game and the user can resume easily by pressing the same key.

4. Scoreboard: The scoreboard is a part of this game which can be viewed from the Scoreboard tab. In this section the achievement of the best five players are stored and can be viewed by simply choosing the option.

5. Exit option: The player can exit the game by simply clicking on the quit option displayed in the Main Menu tab.

**Coding Challenges:**

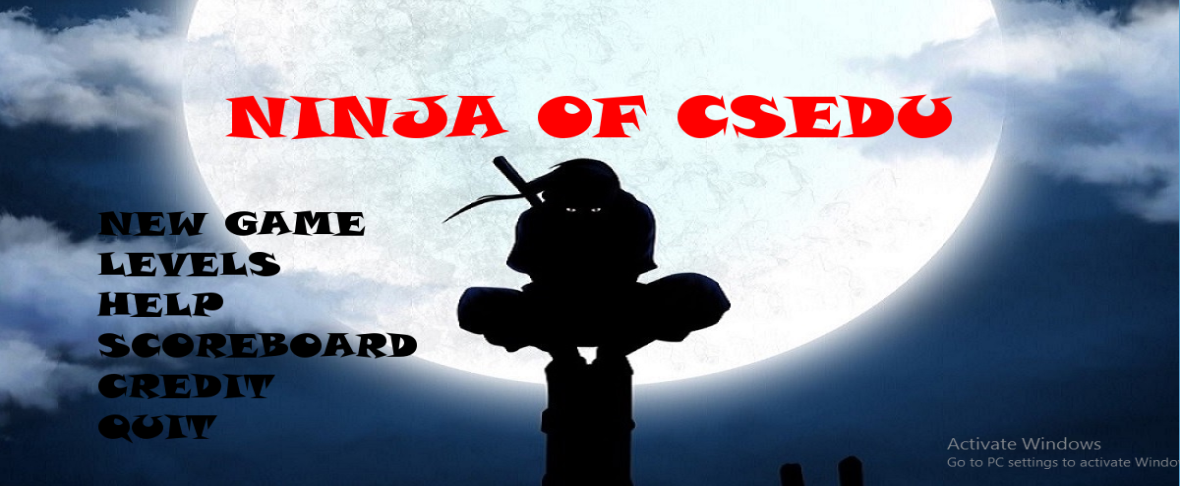
1. Controlling the movements: As it is a ninja running game which means the player will continuously be running. So to make this realistic we had to change the position of the background continuously to make the player feel that the background is actually moving. Not only the background but we had to move all the obstacles along with the weapons also. So we kept a variable and decreased its value and connected this variable with every other features to change their position along with the decrease of the value of this variable.

2. Vanishing the obstacles: The player has to hit the enemies with his/ her weapons to destroy them. So if this weapon touches the enemy, the enemy along with that weapon will be vanished. So we had to check the collision and if it took place we had to set the position of the enemy and the weapon out of the visible window to make the player feel that they vanished.

3. Maintaining the position of superpowers: As superpowers will be used to destroy a number of enemies at a time so we had to place these powers’ position calculating.

**Graphical Interface:**

1. Welcome screen and main menu:



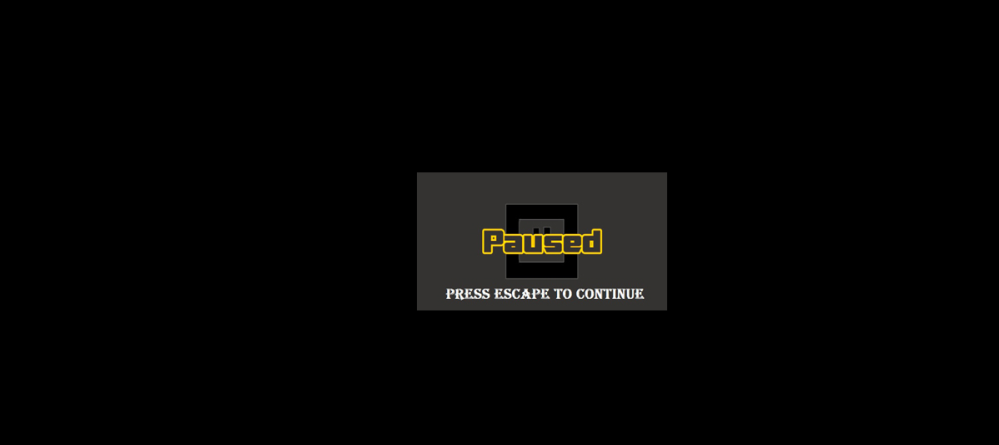
2. When mouse is moved to an option:



3. Level menu:



4. Pause screen:



5. Level-1 (1st year):



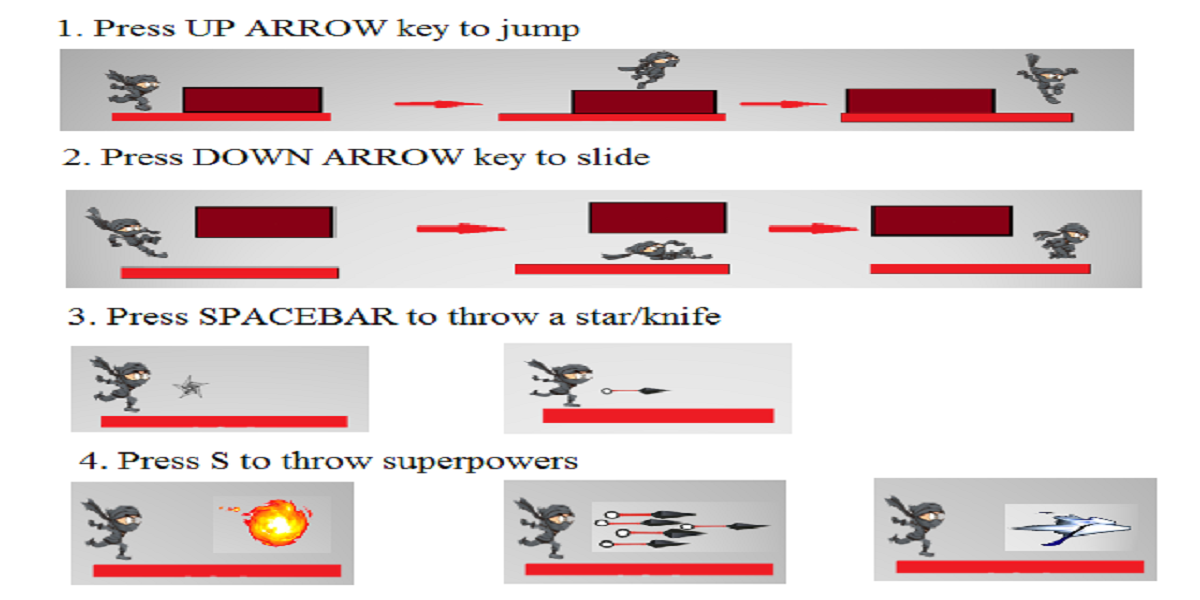
6. Level-2 (2nd year):



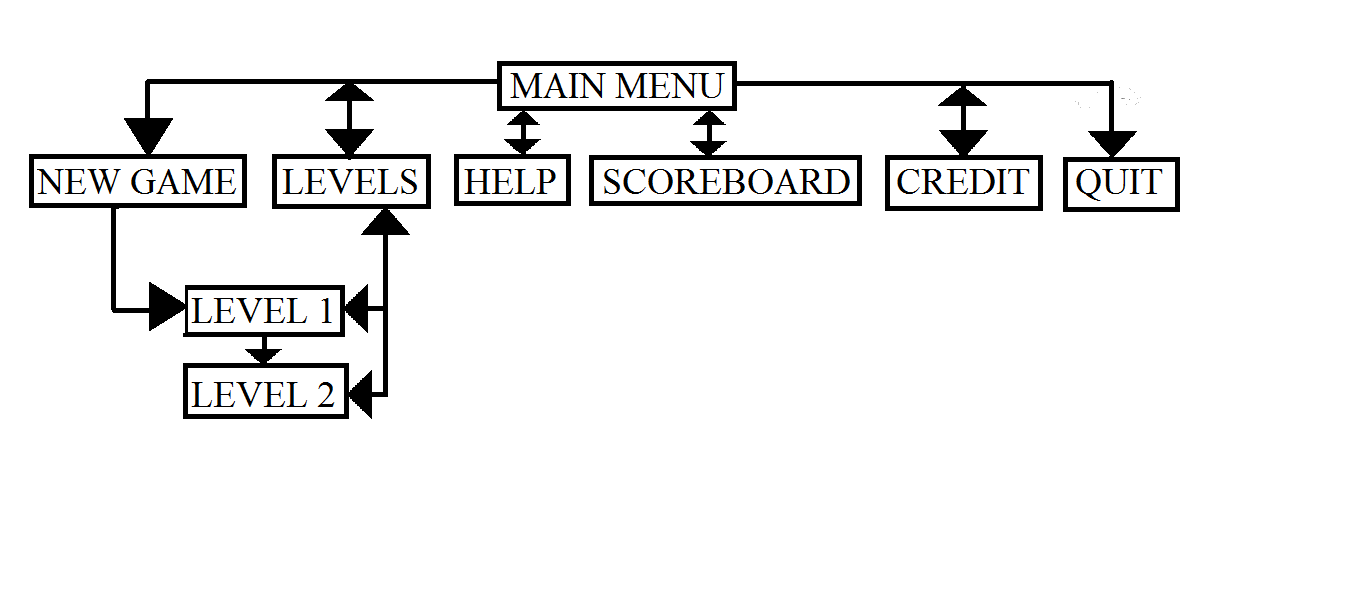
7. Scoreboard:



8. Instruction page:



**Flowchart:**



**Project Overview:**

We tried to make the game as realistic and smooth as possible. When the player keeps scoring high, it will surely make him/ her play the game again and again. The game play is swift. The changes between the two levels in the game make it more interesting and challenging.

**Conclusion:** Though we had to give this project a lot of our time and effort, we had a good time working together to make our very first game and learnt a lot by doing so. The aim of this project was to sharpen our implementation skill and implementing our C/C++ language skill to make something user friendly and interactive was truly a new and exciting experience for us. We hope the players will enjoy this as much as we had in the process. We also hope we can improve in the future and progress in this field.

**Source code:**

The “main.cpp” file:

#include <SFML/Graphics.hpp>

#include<time.h>

#include<iostream>

#include<stdio.h>

#include<string>

using namespace std;

using namespace sf;

RenderWindow window(VideoMode(1200, 600), "The Game!");

#include "level\_menu.h"

#include "mainmenu.h"

#include "1st\_year.h"

#include "2nd\_year.h"

#include "credit.h"

int main()

{

int y=0,l1=0,l2=0;

window.setFramerateLimit(50);

window.setKeyRepeatEnabled(false);

if (main\_menu(window) == 1)

l1=level1(window);

if (main\_menu(window) == 2)

y=level\_menu(window);

if (main\_menu(window) == 3)

creditx(window);

if (y==1)

l1=level1(window);

if (y==2)

level2(window);

if (l1 == 2)

level2(window);

return 0;

}

The “mainmenu.h” header file:

#include <SFML/Graphics.hpp>

#include<time.h>

#include<iostream>

#include<stdio.h>

#include<string>

using namespace std;

using namespace sf;

Text newgame, levels, help, scoreboard, credit, quit,name;

Font a, b, c, d, e, f,za;

//#include "credit.h"

int main\_menu(RenderWindow &window)

{

Texture fairuztex;

Sprite fairuz;

fairuztex.loadFromFile("menu.jpg");

fairuz.setTexture(fairuztex);

fairuz.setPosition(0, 0);

a.loadFromFile("font.TTF");

newgame.setColor(Color::Black);

newgame.setPosition(100, 250);

newgame.setString("NEW GAME");

newgame.setCharacterSize(40);

newgame.setFont(a);

b.loadFromFile("font.TTF");

levels.setColor(Color::Black);

levels.setPosition(100, 300);

levels.setString("LEVELS");

levels.setCharacterSize(40);

levels.setFont(b);

c.loadFromFile("font.TTF");

help.setColor(Color::Black);

help.setPosition(100, 350);

help.setString("HELP");

help.setCharacterSize(40);

help.setFont(c);

d.loadFromFile("font.TTF");

scoreboard.setColor(Color::Black);

scoreboard.setPosition(100, 400);

scoreboard.setString("SCOREBOARD");

scoreboard.setCharacterSize(40);

scoreboard.setFont(d);

e.loadFromFile("font.TTF");

credit.setColor(Color::Black);

credit.setPosition(100, 450);

credit.setString("CREDIT");

credit.setCharacterSize(40);

credit.setFont(e);

f.loadFromFile("font.TTF");

quit.setFont(f);

quit.setColor(Color::Black);

quit.setPosition(100, 500);

quit.setString("QUIT");

quit.setCharacterSize(40);

za.loadFromFile("font.TTF");

name.setFont(za);

name.setColor(Color::Red);

name.setPosition(230, 100);

name.setString("NINJA OF CSEDU");

name.setCharacterSize(70);

int i = 0;

while (window.isOpen())

{

while (window.pollEvent(main\_event\_1))

{

if (main\_event\_1.type == sf::Event::Closed)window.close();

}

//i++;

if (newgame.getGlobalBounds().contains(Mouse::getPosition(window).x, Mouse::getPosition(window).y))

{

//cout << newgame.getGlobalBounds().contains(Mouse::getPosition().x, Mouse::getPosition().y);

newgame.setColor(sf::Color::Cyan);

if (sf::Mouse::isButtonPressed(sf::Mouse::Left))

return 1;

}

else

newgame.setColor(sf::Color::Black);

if (levels.getGlobalBounds().contains(Mouse::getPosition(window).x, Mouse::getPosition(window).y))

{

//cout << newgame.getGlobalBounds().contains(Mouse::getPosition().x, Mouse::getPosition().y);

levels.setColor(sf::Color::Cyan);

if (sf::Mouse::isButtonPressed(sf::Mouse::Left))

return 2;

}

else

levels.setColor(sf::Color::Black);

if (help.getGlobalBounds().contains(Mouse::getPosition(window).x, Mouse::getPosition(window).y))

{

//cout << newgame.getGlobalBounds().contains(Mouse::getPosition().x, Mouse::getPosition().y);

help.setColor(sf::Color::Cyan);

//if (sf::Mouse::isButtonPressed(sf::Mouse::Left))

//return 0;

}

else

help.setColor(sf::Color::Black);

if (scoreboard.getGlobalBounds().contains(Mouse::getPosition(window).x, Mouse::getPosition(window).y))

{

//cout << newgame.getGlobalBounds().contains(Mouse::getPosition().x, Mouse::getPosition().y);

scoreboard.setColor(sf::Color::Cyan);

//if (sf::Mouse::isButtonPressed(sf::Mouse::Left))

//return 0;

}

else

scoreboard.setColor(sf::Color::Black);

if (credit.getGlobalBounds().contains(Mouse::getPosition(window).x, Mouse::getPosition(window).y))

{

//cout << newgame.getGlobalBounds().contains(Mouse::getPosition().x, Mouse::getPosition().y);

credit.setColor(sf::Color::Cyan);

//if (sf::Mouse::isButtonPressed(sf::Mouse::Left))

//return 3;

}

else

credit.setColor(sf::Color::Black);

if (quit.getGlobalBounds().contains(Mouse::getPosition(window).x, Mouse::getPosition(window).y))

{

//cout << newgame.getGlobalBounds().contains(Mouse::getPosition().x, Mouse::getPosition().y);

quit.setColor(sf::Color::Cyan);

if (sf::Mouse::isButtonPressed(sf::Mouse::Left))

window.close();

}

else

quit.setColor(sf::Color::Black);

window.clear();

window.draw(fairuz);

window.draw(newgame);

window.draw(levels);

window.draw(help);

window.draw(scoreboard);

window.draw(credit);

window.draw(quit);

window.draw(name);

window.display();

}

}

The “level\_menu.h” header file:

#include <SFML/Graphics.hpp>

#include<time.h>

#include<iostream>

#include<stdio.h>

#include<string>

using namespace std;

using namespace sf;

Event main\_event\_1;

Text levelt1, year1, levelt2, year2;

Font g, h, i, j, k, l;

int level\_menu(RenderWindow &window)

{

Texture shohantex;

Sprite shohan;

shohantex.loadFromFile("levels.jpg");

shohan.setTexture(shohantex);

shohan.setPosition(0, 0);

g.loadFromFile("font.TTF");

levelt1.setColor(Color::Black);

levelt1.setPosition(400, 100);

levelt1.setString("LEVEL-1:");

levelt1.setCharacterSize(40);

levelt1.setFont(g);

h.loadFromFile("font.TTF");

year1.setColor(Color::Red);

year1.setPosition(450, 175);

year1.setString("1st Year");

year1.setCharacterSize(40);

year1.setFont(h);

i.loadFromFile("font.TTF");

levelt2.setColor(Color::Black);

levelt2.setPosition(400, 250);

levelt2.setString("LEVEL-2:");

levelt2.setCharacterSize(40);

levelt2.setFont(i);

j.loadFromFile("font.TTF");

year2.setColor(Color::Red);

year2.setPosition(450, 325);

year2.setString("2nd Year");

year2.setCharacterSize(40);

year2.setFont(j);

window.setFramerateLimit(24 + 6);

int i = 0;

Clock faifai;

faifai.restart();

while (faifai.getElapsedTime().asMilliseconds() < 250);

while (window.isOpen())

{

while (window.pollEvent(main\_event\_1))

{

if (main\_event\_1.type == sf::Event::Closed)window.close();

}

//i++;

if (year1.getGlobalBounds().contains(Mouse::getPosition(window).x, Mouse::getPosition(window).y))

{

//cout << levelt1.getGlobalBounds().contains(Mouse::getPosition().x, Mouse::getPosition().y);

year1.setColor(sf::Color::Cyan);

if (sf::Mouse::isButtonPressed(sf::Mouse::Left))

return 1;

}

else

year1.setColor(sf::Color::Red);

if (year2.getGlobalBounds().contains(Mouse::getPosition(window).x, Mouse::getPosition(window).y))

{

//cout << levelt1.getGlobalBounds().contains(Mouse::getPosition().x, Mouse::getPosition().y);

year2.setColor(sf::Color::Cyan);

if (sf::Mouse::isButtonPressed(sf::Mouse::Left))

return 2;

}

else

year2.setColor(sf::Color::Red);

//i %= 30;

window.clear();

window.draw(shohan);

window.draw(levelt1);

window.draw(year1);

window.draw(levelt2);

window.draw(year2);

window.display();

}

}

The “credit.h” header file:

The “highscore.h” header file:

The “1st\_year.h” header file:

#include <SFML/Graphics.hpp>

#include<time.h>

#include<iostream>

#include<stdio.h>

#include<string>

////level1 1////

using namespace std;

using namespace sf;

int lives = 5, flag = 0, flag5[5], flag6[5], upj[5], crash[5], jombie\_flag[5], jcount[5], ccount[5], cxcount[5], pcount[5], mulcount[5], gm\_over = 0;

int cdistance[10], pdistance[14], p1distance[14], muldistance[10], mul1distance[10], pyac[20], pyth[20], mx[20], jdistance[10], mdistance[10];

Sprite ninja\_star[40], sobstacle[20], python[8], playerImage, python1[8], cxImage[5], cImage[5], sgame\_over, hit\_image, multi[10], jombie[10], paused, tied\_man;

String sentence, count\_begin;

Font font;

Text text(sentence, font, 40);

Text text\_begin(count\_begin, font, 40);

int coin\_sect[50];

int tx[20];

bool isCollide(Sprite s1, Sprite s2)

{

return s1.getGlobalBounds().intersects(s2.getGlobalBounds());

}

int attack(Sprite star)

{

int cj;

for (cj = 0; cj <= 2; cj++)

{

if (isCollide(cImage[cj], star))

{

ccount[cj]++;

if (ccount[cj] >= 2)

cdistance[cj] = -2000;

return 1;

}

if (isCollide(cxImage[cj], star))

{

cxcount[cj]++;

if (cxcount[cj] >= 2)

cdistance[cj] = -3000;

return 1;

}

}

for (cj = 0; cj <= 2; cj++)

{

if (isCollide(python[cj], star))

{

pcount[cj]++;

if (pcount[cj] >= 2)

pdistance[cj] = -1000;

return 1;

}

}

for (cj = 0; cj < 5; cj++)

{

if (isCollide(star, jombie[cj]))

{

jcount[cj]++;

if (jcount[cj]>2)

jdistance[cj] = -2500;

return 1;

}

}

for (cj = 0; cj < 16; cj++)

{

if (isCollide(sobstacle[cj], star))

{

return 1;

}

}

for (cj = 0; cj < 4; cj++)

{

if (isCollide(multi[cj], star))

{

mulcount[cj]++;

if (mulcount[cj] >= 2)

muldistance[cj] = -1000;

return 1;

}

}

return 0;

}

void GAME\_OVER(int count)

{

int count1 = count;

string str;

while (count1 > 0)

{

int a;

a = count1 % 10;

count1 = count1 / 10;

a = a + 48;

char b = a;

str = str + b;

}

reverse(str.begin(), str.end());

if (gm\_over == 0){

sentence += str;

if (count == 0) sentence += '0';

gm\_over = 1;

}

text.setString(sentence);

text.setPosition(450, 50);

window.draw(text);

if (playerImage.getPosition().x < 60){

text.setString("Re Admission :(");

text.setPosition(450, 150);

window.draw(text);

}

}

int level1(RenderWindow &window)

{

window.setKeyRepeatEnabled(false);

int i, xod = 2000, upward = 200, j, s, posx = 0, posy = 0, flag2 = 1, up = 0, fl = 0, ninja\_starflag = 0, total\_trans = 11, pys = 0, life\_flag = 0,

nsf = 0, ns = 0, star\_num = 40, js = 0, ts = 0, flagjump = 0, player\_height = 300, jump\_flag = 0, firo = 0, fire\_flag = 0, Anife\_flag = 0, throw\_flag = 0,

sobstacle\_index = 0, hole\_flag = 0, holedex = 6, ms = 0, flag3 = 1, flag4 = 0, up1 = 0, flagf = 1, upf = 0;

enum Task { Run, Jump, Slide };

int distance[16];

distance[0] = 0;

int count = 0;

//////Positioning everything on the co ordinates///////

j = 2;

for (i = 1; i < 16; i++)

{

if (i <= 5)

distance[i] = distance[i - 1] + 1200 + rand() % 100;

else

distance[i] = distance[i - 1] + 1800 + rand() % 100;

if (i == 1)

cdistance[0] = distance[i] - 200;

if (i == 2)

{

cdistance[1] = distance[i] - 400;

cdistance[2] = distance[i] - 200;

}

else if (i == 3)

{

pdistance[0] = distance[i] - 200;

p1distance[0] = distance[i] - 200;

}

else if (i == 4)

{

pdistance[1] = distance[i] - 300;

p1distance[1] = distance[i] - 300;

}

if (i == 5)

{

pdistance[2] = distance[i] - 100;

p1distance[2] = distance[i] - 100;

}

if (i == 6)

{

muldistance[0] = distance[i] - 500;

muldistance[1] = distance[i] - 100;

mul1distance[0] = distance[i] - 500;

mul1distance[1] = distance[i] - 100;

}

if (i == 7)

{

muldistance[2] = distance[i] - 700;

muldistance[3] = distance[i] - 200;

mul1distance[2] = distance[i] - 700;

mul1distance[3] = distance[i] - 200;

}

if (i == 8)

{

jdistance[0] = distance[i] - 700;

jdistance[1] = distance[i] - 300;

}

if (i == 9)

{

jdistance[2] = distance[i] - 800;

jdistance[3] = distance[i] - 600;

jdistance[4] = distance[i] - 400;

jdistance[5] = distance[i] - 200;

}

}

Clock clock, clock2, begin\_clock;

float timer = 0, delay = 0.1, timer2 = 0;

float frameCounter = 0, switchFrame = 100, frameSpeed = 500;

window.setFramerateLimit(10);

Vector2i source(0, Run);

Texture pTexture, bart, tground, tobstacle, game\_over, pcoins, b1Texture, paus, cTexture, elect, cxTexture, knifetx, pstar, firef, c\_attack, fire, build, heart, jombiex, pythontexture, python1texture, mulTexture, hit;

Sprite bars, sground, sgame\_over, b1Image, b2Image, player\_heart, building1, building2, knife\_effect, fire\_flag\_flag, cattack, fire\_ball;

Sprite coins[100], electricity[12];

if (!b1Texture.loadFromFile("MrShadow3.png"))

cout << "Error could not load player image" << endl;

b1Image.setTexture(b1Texture);

b2Image.setTexture(b1Texture);

build.loadFromFile("building.jpg");

building1.setTexture(build);

building2.setTexture(build);

building1.setPosition(0, 450);

building2.setPosition(1180, 450);

paus.loadFromFile("Paused.jpg");

paused.setTexture(paus);

knifetx.loadFromFile("knife\_effct.png");

knife\_effect.setTexture(knifetx);

c\_attack.loadFromFile("cattack.png");

cattack.setTexture(c\_attack);

firef.loadFromFile("fire\_flag.png");

fire\_flag\_flag.setTexture(firef);

fire.loadFromFile("fireball.png");

fire\_ball.setTexture(fire);

hit.loadFromFile("got\_hit.png");

hit\_image.setTexture(hit);

heart.loadFromFile("player\_heart.png");

player\_heart.setTexture(heart);

jombiex.loadFromFile("jombie.png");

for (i = 0; i < 10; i++)

{

jombie[i].setTexture(jombiex);

jombie[i].setPosition(-1000, -1000);

}

cTexture.loadFromFile("c\_language.png");

for (i = 0; i < 3; i++)

{

cImage[i].setTexture(cTexture);

cImage[i].setPosition(600 + cdistance[i], 292);

}

cxTexture.loadFromFile("c\_language\_attack.png");

for (i = 0; i < 3; i++)

{

cxImage[i].setTexture(cxTexture);

cxImage[i].setPosition(1200, 1200);

}

pythontexture.loadFromFile("c++.png");

for (i = 0; i < 3; i++)

{

python[i].setTexture(pythontexture);

python[i].setPosition(600 + pdistance[i], 292);

}

python1texture.loadFromFile("c1++.png");

for (i = 0; i < 4; i++)

{

python1[i].setTexture(python1texture);

python1[i].setPosition(600 + pdistance[i], 292);

}

mulTexture.loadFromFile("multimeter.png");

for (i = 0; i < 4; i++)

multi[i].setTexture(mulTexture);

elect.loadFromFile("electricity.png");

for (i = 0; i < 11; i++)

{

electricity[i].setTexture(elect);

//electricity[i].setPosition(1000, 1000);

}

pstar.loadFromFile("ninja\_star222.png");

for (i = 0; i < 40; i++)

{

ninja\_star[i].setTexture(pstar);

ninja\_star[i].setPosition(550, -50);

}

b2Image.setPosition(800, 0);

if (!pTexture.loadFromFile("xyz.png"))

cout << "Error could not load player image" << endl;

playerImage.setTexture(pTexture);

playerImage.setPosition(50, 300);

game\_over.loadFromFile("GAME\_OVER.png");

sgame\_over.setTexture(game\_over);

sgame\_over.setPosition(0, 0);

tground.loadFromFile("ground.png");

sground.setTexture(tground);

tobstacle.loadFromFile("obstacle1.png");

int t = 0;

while (t < 16){

for (i = 0; i < 4; i++)

{

for (j = 0; j < 4; j++)

{

sobstacle[t].setTexture(tobstacle);

sobstacle[t].setTextureRect(IntRect(4 + j \* 194, 4 + i \* 109, 195, 109));

t++;

}

}

}

pcoins.loadFromFile("coin.png");

for (i = 0; i < 50; i++)

coins[i].setTexture(pcoins);

font.loadFromFile("naruto.ttf");

//text.setColor(Color(44, 127, 255));

text.setStyle(Text::Bold);

sentence = "Your CGPA is: ";

////THE REAL GAME START FROM HERE/////

long long wait = 1;

while (window.isOpen())

{

Event e;

while (window.pollEvent(e))

{

switch (e.type)

{

case Event::Closed:

window.close();

break;

case Event::KeyPressed:

if (e.key.code == Keyboard::Up)

{

source.y = Jump; j = 0;

}

else if (e.key.code == Keyboard::Down)

{

source.y = Slide; s = 0;

}

break;

}

}

if (Keyboard::isKeyPressed(Keyboard::Escape))

wait++;

if (wait % 2 == 0)

{

paused.setPosition(500, 200);

window.draw(paused);

window.display();

window.clear();

}

if (wait % 2 == 1){

if (flag == 0){

window.draw(b1Image);

b1Image.move(-10, 0);

window.draw(b2Image);

b2Image.move(-10, 0);

window.draw(sground);

window.draw(building1);

building1.move(-10, 0);

window.draw(building2);

building2.move(-10, 0);

//////features///////

if (b1Image.getPosition().x + 800 < 0) b1Image.setPosition(800, 0);

if (b2Image.getPosition().x + 800 < 0) b2Image.setPosition(800, 0);

if (building1.getPosition().x < 0) building2.setPosition(building1.getPosition().x + 1100, 450);

if (building2.getPosition().x < 0)building1.setPosition(building2.getPosition().x + 1100, 450);

float time = clock.getElapsedTime().asMilliseconds();

clock.restart();

timer += time;

timer2 = clock2.getElapsedTime().asMilliseconds();

if (life\_flag == 1 && timer2 > 500)

{

timer2 = 0;

life\_flag = 0;

distance[sobstacle\_index] = -1000;

lives--;

if (lives <= 0)

life\_flag = 1;

}

if (life\_flag == 1)

{

if (lives <= 0)

{

if (flag3 == 1)

up1 = up1 - 30;

if (up1 <= -60) {

flag4 = 1; flag3 = 0;

}

if (flag4 == 1)

up1 = up1 + 50;

hit\_image.setPosition(playerImage.getPosition().x, 220 + up1);

window.draw(hit\_image);

GAME\_OVER(count);

}

else

{

hit\_image.setPosition(playerImage.getPosition().x - 20, playerImage.getPosition().y - 10);

window.draw(hit\_image);

}

}

player\_heart.setTextureRect(IntRect(2, 2, lives \* 65, 60));

window.draw(player\_heart);

///JUMP////

if (life\_flag == 0)

{

if (source.y == Jump)

{

int dif = 50;

playerImage.setTextureRect(IntRect(j \* 135, 136, 135, 115));

if (j == 0) playerImage.setPosition(50, player\_height - dif \* 1);

else if (j == 1) playerImage.setPosition(50, player\_height - dif \* 2);

else if (j == 2) playerImage.setPosition(50, player\_height - dif \* 3);

else if (j == 3) playerImage.setPosition(50, player\_height - dif \* 2);

else if (j == 4) playerImage.setPosition(50, player\_height - dif \* 1);

else if (j == 5)

playerImage.setPosition(50, player\_height - dif \* 0);

xod -= 20;

window.draw(playerImage);

j++;

if (j >= 6)

{

source.x = 0; source.y = Run;

}

}

/////SLIDE//////

else if (source.y == Slide)

{

playerImage.move(0, 2);

int x1;

if (s == 0)

x1 = 50;

if (s == 0)

{

playerImage.setTextureRect(IntRect(s \* 145, 2 \* 136, 145, 115));

playerImage.setPosition(x1, playerImage.getPosition().y - 20);

}

else if (s == 4 || s == 5)

{

playerImage.setTextureRect(IntRect(s \* 145, (2 \* 136 + 30), 145, 80));

playerImage.setPosition(x1, playerImage.getPosition().y + 20);

}

else{

playerImage.setTextureRect(IntRect(s \* 145, (2 \* 136 + 55), 145, 60));

playerImage.setPosition(x1, playerImage.getPosition().y + 35);

}

xod -= 30;

window.draw(playerImage);

s++;

if (s >= 6)

{

source.x = 0; source.y = Run;

}

playerImage.move(0, -2);

}

/////RUN////

else if (source.y == Run){

source.x++;

if (source.x >= 6)

source.x = 0;

playerImage.setTextureRect(IntRect(source.x \* 95, source.y, 95, 90));

window.draw(playerImage);

}

///RUN END////

}

////Obstacles moving ////

if (timer > 50 && !life\_flag)

{

xod -= 25;

timer = 0;

}

int c = 0;

for (t = 0; t < 16; t++)

{

if (t % 2 == 0)

sobstacle[t].setPosition(xod + distance[t], 292);

else

sobstacle[t].setPosition(xod + distance[t], 220);

window.draw(sobstacle[t]);

for (i = 0; i < 3; i++)

{

if (t != 7){

if (t % 2 == 0)

coins[c].setPosition(xod + distance[t] + i \* 65, 220 + coin\_sect[c]);

if (t % 2 == 1)

coins[c].setPosition(xod + distance[t] + i \* 65, 339 + coin\_sect[c]);

window.draw(coins[c]);

c++;

}

}

}

sground.setPosition(0, 400);

if (sobstacle[15].getPosition().x + 196 <= 0) xod = 600;

//// Throwing Ninja star/////

if (Keyboard::isKeyPressed(Keyboard::Space) && life\_flag == 0)

{

ninja\_starflag = 1;

nsf = 0;

ninja\_star[ns].setPosition(playerImage.getPosition().x, playerImage.getPosition().y);

ns++;

if (ns > star\_num) ninja\_starflag = 0;

}

for (i = 0; i < ns && ninja\_starflag == 1; i++)

{

if (ninja\_star[i].getPosition().x - playerImage.getPosition().x < 20 && ninja\_star[i].getPosition().x>0)

{

knife\_effect.setPosition(playerImage.getPosition().x + 15, playerImage.getPosition().y - 10);

window.draw(knife\_effect);

}

if (ninja\_star[ns - 1].getPosition().x <= 1000)

{

ninja\_star[i].setTextureRect(IntRect(nsf \* 60, 0, 60, 50));

ninja\_star[i].move(40, 0);

nsf++;

nsf = nsf % 4;

window.draw(ninja\_star[i]);

}

}

//////throwing fire ball////

fire\_flag\_flag.setPosition(xod + distance[7] + 80 + firo, 350);

window.draw(fire\_flag\_flag);

if (isCollide(fire\_flag\_flag, playerImage))

{

fire\_flag = 1;

firo = -9000;

}

if (fire\_flag == 1){

if (Keyboard::isKeyPressed(Keyboard::S) && throw\_flag == 0 && life\_flag == 0)

{

throw\_flag = 1;

fire\_flag = 0;

fire\_ball.setPosition(playerImage.getPosition().x, playerImage.getPosition().y);

}

else if (throw\_flag == 0)

{

if (flagf == 1)

upf = upf - 20;

if (upf <= -80) flagf = 0;

if (flagf == 0)

upf = upf + 20;

if (upf >= 0) flagf = 1;

fire\_ball.setPosition(playerImage.getPosition().x, playerImage.getPosition().y - 180 + upf);

window.draw(fire\_ball);

}

}

if (throw\_flag == 1)

{

fire\_ball.move(40, 0);

window.draw(fire\_ball);

for (i = 0; i < 16; i++)

{

if (isCollide(fire\_ball, sobstacle[i]))

{

throw\_flag = 0;

fire\_ball.setPosition(-999, -888);

}

}

}

for (i = 0; i < 5; i++)

{

if (isCollide(jombie[i], fire\_ball))

{

flag5[i] = 1;

crash[i] = 1;

}

if (crash[i] == 1){

if (flag5[i] == 1)

upj[i] = upj[i] - 30;

if (upj[i] <= -60) {

flag6[i] = 1; flag5[i] = 0;

}

if (flag6[i] == 1)

upj[i] = upj[i] + 50;

jombie[i].setPosition(jombie[i].getPosition().x, 220 + upj[i]);

window.draw(jombie[i]);

jombie\_flag[i] = 1;

}

}

// C , python, multi, jomibe Moving and attacking////

if (flag2 == 1)

up = up - 30;

if (up <= -60) flag2 = 0;

if (flag2 == 0)

up = up + 30;

if (up >= 0) flag2 = 1;

for (i = 0; i < 3; i++)

{

if ((cImage[i].getPosition().x - playerImage.getPosition().x) < 500)

{

cImage[i].setPosition(-100, -100);

cxImage[i].setPosition(xod - 30 + cdistance[i], 295 + up);

cxImage[i].move(-80 - i \* 20, 0);

window.draw(cxImage[i]);

}

else{

cImage[i].setPosition(xod + cdistance[i], 295 + up);

cImage[i].move(-80 - i \* 20, 0);

window.draw(cImage[i]);

}

python[i].setPosition(xod + pdistance[i], 320 + up);

python[i].move(-80 - i \* 20, 0);

window.draw(python[i]);

if ((python[i].getPosition().x - playerImage.getPosition().x) < 850)

{

python1[i].setTextureRect(IntRect(2 + 55 \* pys, 2, 55, 54));

python1[i].setPosition(xod + p1distance[i] - pyac[i] \* 30 + 80, 280);

window.draw(python1[i]);

if (pyth[i] == 0)

{

python1[i].setPosition(xod + p1distance[i] - pyac[i] \* 30, 280);

window.draw(python1[i]);

}

pyac[i]++;

}

else

{

python1[i].setTextureRect(IntRect(2, 2, 55, 54));

python1[i].setPosition(xod + p1distance[i] + 40, 350 + up);

python1[i].move(-80 - i \* 20, 0);

window.draw(python1[i]);

python1[i].setTextureRect(IntRect(2, 2, 55, 54));

python1[i].setPosition(xod + p1distance[i] + 130, 350 + up);

python1[i].move(-80 - i \* 20, 0);

window.draw(python1[i]);

}

}

for (i = 0; i < 5; i++)

{

if (jombie\_flag[i] == 0){

jombie[i].setTextureRect(IntRect(2 + 120 \* (pys % 3), 2, 120, 105));

jombie[i].setPosition(xod + jdistance[i], 292);

window.draw(jombie[i]);

}

//-pyac[i] \* 10 + 80

}

pys++;

pys = pys % 6;

for (i = 0; i < 4; i++)

{

multi[i].setTextureRect(IntRect(140 \* ms + 4, 4, 140, 120));

multi[i].setPosition(xod + muldistance[i], 292 + up);

multi[i].move(-80 - i \* 20, 0);

window.draw(multi[i]);

ms++;

ms = ms % 3;

if (multi[i].getPosition().x > 0 && multi[i].getPosition().x < 1200 || electricity[i].getPosition().x>0)

{

int mup = -20, mdown = 30;

if (i % 2 == 0) {

mup = 70; mdown = 30;

}

electricity[i].setPosition(xod + mul1distance[i] - mx[i] \* mdown, 292 + mup);

window.draw(electricity[i]);

mx[i]++;

}

}

///// c , python, obstacles are getting attacked ////

int stari, cj;

for (i = 0; i < 40; i++)

{

cj = attack(ninja\_star[i]);

if (cj == 1)

ninja\_star[i].setPosition(-400, -100);

}

///Ground and train checking and game over////

int k, x1, y1, r1, r2, m1;

for (k = 0; k < 16; k++)

{

x1 = playerImage.getPosition().x + 95;

y1 = playerImage.getPosition().y;

r1 = sobstacle[k].getPosition().x;

r2 = sobstacle[k].getPosition().x + 130;

m1 = sobstacle[k].getPosition().y;

if (x1 >= r1 - 10 && playerImage.getPosition().x <= r2 && m1 - y1 >= 88)// here i have debugged from game over

{

if (jump\_flag == 0){

playerImage.setPosition(50, m1 - 95);

source.y = Run;

jump\_flag = 1;

}

player\_height = 300 - 100;

break;

}

if (x1 >= r1 + 15 && playerImage.getPosition().x <= r2 && (m1 + 100 > y1) && life\_flag == 0)

{

life\_flag = 1;

clock2.restart();

sobstacle\_index = k;

}

}

for (i = 0; i < 5; i++)

{

if (isCollide(playerImage, jombie[i]))

{

jombie[i].setPosition(-1000, 0);

jdistance[i] = -2500;

life\_flag = 1;

clock2.restart();

}

}

for (i = 0; i < 3; i++)

{

if (isCollide(playerImage, cxImage[i]))

{

cxImage[i].setPosition(-1000, 0);

cdistance[i] = -1000;

life\_flag = 1;

clock2.restart();

}

if (isCollide(python[i], playerImage))

{

python[i].setPosition(-1000, 0);

pdistance[i] = -1000;

life\_flag = 1;

clock2.restart();

}

if (isCollide(python1[i], playerImage))

{

p1distance[i] = -1000;

life\_flag = 1;

clock2.restart();

}

}

for (i = 0; i < 4; i++)

{

if (isCollide(playerImage, multi[i]))

{

muldistance[i] = -1000;

life\_flag = 1;

clock2.restart();

}

if (isCollide(playerImage, electricity[i]))

{

cout << playerImage.getPosition().x + 95 - electricity[i].getPosition().x << endl;

if (playerImage.getPosition().x + 95 - electricity[i].getPosition().x > 30 && playerImage.getPosition().x + 95 - electricity[i].getPosition().x < 95)

{

mul1distance[i] = -1000;

life\_flag = 1;

clock2.restart();

}

}

}

if (k == 16) {

jump\_flag = 0;

playerImage.setPosition(playerImage.getPosition().x, sground.getPosition().y - 95);

player\_height = 300;

}

if (playerImage.getPosition().x > 50) playerImage.move(-15, 0);

//// Intersection checking with player and good things happen////

for (i = 0; i < 50; i++)

{

if (isCollide(playerImage, coins[i]) == 1)

{

coin\_sect[i] = -1000;

count++;

int count1 = count;

}

}

if (sobstacle[9].getPosition().x < 0)

{

xod += 25;

playerImage.move(55, 0);

if (playerImage.getPosition().x > 1300)

return 2;

}

float starting\_pause = begin\_clock.getElapsedTime().asSeconds();

text\_begin.setPosition(400, 200);

if (starting\_pause <= 2)

text\_begin.setString("First year in CSEDU");

else if (starting\_pause <= 3){

text\_begin.setString("1");

text\_begin.setPosition(500, 200);

}

else if (starting\_pause <= 4)

{

text\_begin.setString("2");

text\_begin.setPosition(500, 200);

}

else if (starting\_pause <= 5)

{

text\_begin.setString("3");

text\_begin.setPosition(500, 200);

}

else if (starting\_pause <= 6 && starting\_pause<7)

text\_begin.setString("Go!! Go!! Go!!");

else text\_begin.setPosition(-100, -100);

if (playerImage.getPosition().x > 60)

{

text\_begin.setPosition(500, 200);

text\_begin.setString("1st!!year!!Passed");

window.draw(text\_begin);

GAME\_OVER(count);

}

window.draw(text\_begin);

window.display();

window.clear();

} ////flag ending bracket

} /// wait ending bracket

}

}

The “2nd\_year.h” header file:

#include <SFML/Graphics.hpp>

#include<time.h>

#include<iostream>

#include<stdio.h>

#include<string>

#include<iomanip>

#include <Windows.h>

using namespace std;

using namespace sf;

//RenderWindow window(VideoMode(1200, 600), "The Game!");

int livesxx = 5, flagxx = 0, mxdistance, mj, mmove = 0, sup1ff = 0, sup1fff = 0, flags1 = 0, ups1 = 0, sup0ff = 0, sup0fff = 0, mummy\_attack\_count = 0,

devil\_livesxx = 3, superpowercount = 0, swordcount = 0, javacount[5], transcount[13], gm\_overx;

int java\_distance[10], tdistance[14], t1distance[14], train\_distance[5];

Sprite AnifeImgae[5], playerImage2, ninja\_starx[60], sobstaclexx[20], transistor[12], java\_Image[8], fire\_ball2[4], pausedx, sgame\_overx, icpc, mummy, hit\_imagex, strike\_image, sup1f, sup0f, sword;

String sentencexx, count\_beginxx;

Font Font2x;

Text textxx(sentencexx, Font2x, 40);

Text textxx\_begin(count\_beginxx, Font2x, 40);

bool isCollide2(Sprite s1, Sprite s2)

{

return s1.getGlobalBounds().intersects(s2.getGlobalBounds());

}

int coin\_sectx[50];

int txx[20];

void knife2()

{

AnifeImgae[0].setPosition(50, 300);

AnifeImgae[1].setPosition(55, 320);

AnifeImgae[2].setPosition(40, 290);

AnifeImgae[3].setPosition(45, 330);

AnifeImgae[4].setPosition(60, 340);

printf("doom\n");

}

int attack2(Sprite star)

{

int cj;

for (cj = 0; cj <= 4; cj++)

{

if (isCollide2(java\_Image[cj], star))

{

javacount[cj]++;

if (javacount[cj] >= 2)

java\_distance[cj] = -1000;

return 1;

}

}

for (cj = 0; cj <= 11; cj++)

{

if (isCollide2(transistor[cj], star))

{

transcount[cj]++;

if (transcount[cj] >= 2)

tdistance[cj] = -1000;

return 1;

}

}

for (cj = 0; cj < 16; cj++)

{

if (isCollide2(sobstaclexx[cj], star))

return 1;

}

if (isCollide2(mummy, star)){

strike\_image.setPosition(mummy.getPosition().x, mummy.getPosition().y);

window.draw(strike\_image);

mummy\_attack\_count++;

if (mummy\_attack\_count>15 && devil\_livesxx == 1) devil\_livesxx--;

//cout << mummy\_attack\_count << endl;

return 1;

}

return 0;

}

void GAME\_OVER2(int count)

{

int count1 = count;

string str;

while (count1 > 0)

{

int a;

a = count1 % 10;

count1 = count1 / 10;

a = a + 48;

char b = a;

str = str + b;

}

reverse(str.begin(), str.end());

if (gm\_overx == 0){

sentencexx += str;

if (count == 0) sentencexx += '0';

gm\_overx = 1;

}

textxx.setString(sentencexx);

textxx.setPosition(450, 50);

window.draw(textxx);

if (mummy.getPosition().y<600){

textxx.setString("Re Admission :(");

text.setPosition(450, 150);

window.draw(text);

}

}

void level2(RenderWindow &window)

{

window.setKeyRepeatEnabled(false);

int i, xod = 1800, upward = 200, j, s, posx = 0, posy = 0, flag2 = 1, up = 0, fl = 0, ninja\_starxflag = 0, total\_trans = 11, life\_flag = 0, sobstaclexx\_index = 0,

nsf = 0, ns = 0, star\_num = 60, js = 0, ts = 0, flagjump = 0, player\_height = 300, jump\_flag = 0, Anife\_flag = 0, hole\_flag = 0, holedex = 6

, flag3 = 1, flag4 = 0, up1 = 0;

enum Task { Run, Jump, Slide };

int distance[16];

distance[0] = 0;

int count = 0;

//////Positioning everything on the co ordinates///////

j = 2;

for (i = 1; i < 16; i++)

{

if (i <= 10)

{

distance[i] = distance[i - 1] + 1100 + rand() % 100;

train\_distance[i] = distance[i - 1] + 1100 + rand() % 100;

}

else

{

distance[i] = distance[i - 1] + 1600 + rand() % 100;

train\_distance[i] = distance[i - 1] + 1600 + rand() % 100;

}

if (i == 1)

java\_distance[0] = distance[i] - 300;

if (i == 2)

{

java\_distance[1] = distance[i] - 400;

java\_distance[2] = distance[i] - 200;

}

else if (i == 3)

{

tdistance[0] = distance[i] - 200;

t1distance[0] = distance[i] - 200;

}

else if (i == 4)

{

tdistance[1] = distance[i] - 200;

t1distance[1] = distance[i] - 200;

}

if (i == 7 || i == 8)

{

tdistance[j] = distance[i] - 300;

t1distance[j] = distance[i] - 300;

j++;

tdistance[j] = distance[i] - 100;

t1distance[j] = distance[i] - 100;

j++;

}

if (i == 9)

{

java\_distance[3] = distance[i] + 700;

java\_distance[4] = distance[i] + 900;

}

if (i == 11 || i == 12)

{

tdistance[j] = distance[i] - 400;

t1distance[j] = distance[i] - 400;

j++;

tdistance[j] = distance[i] - 100;

t1distance[j] = distance[i] - 100;

j++;

}

if (i == 13)

{

tdistance[j] = distance[i] - 400;

t1distance[j] = distance[i] - 400;

j++;

tdistance[j] = distance[i] - 100;

t1distance[j] = distance[i] - 100;

j++;

}

if (i == 14)

{

mxdistance = distance[i] - 100;

}

}

Clock clock, clock2, mummy\_clock, begin\_clock;

float timer = 0, delay = 0.1, timer2 = 0, mummy\_timer = 0;

float frameCounter = 0, switchFrame = 100, frameSpeed = 500;

window.setFramerateLimit(10);

Vector2i source(0, Run);

Texture pTexture, bart, tground, tobstacle, game\_over, pcoins, b1Texture, java\_Texture, ic, bm, paus, pstar, fire2, swrd, dheart, knifetx, strike, sp1f, s1, sp0f, s2, trantexture, mball, heart, mumy, tball, trainTexture, Anife, Aknife\_flag, hole, hit;

Sprite bars, sground, b1Image, b2Image, Aknife\_flagImage, hole\_image, boom, player\_heart, knife\_effect, mummy\_ball, superpower1, devil\_heart;

Sprite coins[100], trainImage[2], trans\_ball[12];

if (!b1Texture.loadFromFile("bg.png"))

cout << "Error could not load player image" << endl;

b1Image.setTexture(b1Texture);

b2Image.setTexture(b1Texture);

knifetx.loadFromFile("knife\_effct.png");

knife\_effect.setTexture(knifetx);

paus.loadFromFile("Paused.jpg");

pausedx.setTexture(paus);

bm.loadFromFile("boom.png");

boom.setTexture(bm);

Aknife\_flag.loadFromFile("knife\_flag.png");

Aknife\_flagImage.setTexture(Aknife\_flag);

Aknife\_flagImage.setPosition(distance[10], 252);

hole.loadFromFile("black\_hole.png");

hole\_image.setTexture(hole);

dheart.loadFromFile("devil\_heart.png");

devil\_heart.setTexture(dheart);

strike.loadFromFile("strike.png");

strike\_image.setTexture(strike);

sp0f.loadFromFile("sp0.png");

sup0f.setTexture(sp0f);

ic.loadFromFile("icpc.png");

icpc.setTexture(ic);

swrd.loadFromFile("shieldsword.png");

sword.setTexture(swrd);

sp1f.loadFromFile("sp1.png");

sup1f.setTexture(sp1f);

s1.loadFromFile("superpower1.png");

superpower1.setTexture(s1);

mumy.loadFromFile("mummy.png");

mummy.setTexture(mumy);

mball.loadFromFile("mummy\_ball.png");

mummy\_ball.setTexture(mball);

hit.loadFromFile("got\_hit.png");

hit\_imagex.setTexture(hit);

heart.loadFromFile("player\_heart.png");

player\_heart.setTexture(heart);

trainTexture.loadFromFile("train.png");

for (i = 0; i < 2; i++)

trainImage[i].setTexture(trainTexture);

tball.loadFromFile("transistor ball.png");

for (i = 0; i <= 11; i++)

trans\_ball[i].setTexture(tball);

Anife.loadFromFile("angled knives.png");

for (i = 0; i < 5; i++)

AnifeImgae[i].setTexture(Anife);

fire2.loadFromFile("fire\_ball2.png");

for (i = 0; i < 3; i++) fire\_ball2[i].setTexture(fire2);

java\_Texture.loadFromFile("javax.png");

for (i = 0; i < 8; i++)

java\_Image[i].setTexture(java\_Texture);

trantexture.loadFromFile("trransistors.png");

for (i = 0; i <= 11; i++)

transistor[i].setTexture(trantexture);

pstar.loadFromFile("knife.png");

for (i = 0; i < 60; i++)

{

ninja\_starx[i].setTexture(pstar);

ninja\_starx[i].setPosition(550, -50);

}

b2Image.setPosition(800, 0);

if (!pTexture.loadFromFile("xyz.png"))

cout << "Error could not load player image" << endl;

playerImage2.setTexture(pTexture);

playerImage2.setPosition(50 + mmove, 300);

game\_over.loadFromFile("GAME\_OVER.png");

sgame\_overx.setTexture(game\_over);

sgame\_overx.setPosition(0, 0);

tground.loadFromFile("ground.png");

sground.setTexture(tground);

tobstacle.loadFromFile("obstacle1.png");

int t = 0;

while (t < 16){

for (i = 0; i < 4; i++)

{

for (j = 0; j < 4; j++)

{

sobstaclexx[t].setTexture(tobstacle);

sobstaclexx[t].setTextureRect(IntRect(4 + j \* 194, 4 + i \* 109, 195, 109));

t++;

}

}

}

pcoins.loadFromFile("coin.png");

for (i = 0; i < 50; i++)

coins[i].setTexture(pcoins);

Font2x.loadFromFile("naruto.ttf");

text.setColor(Color(44, 127, 255));

text.setStyle(Text::Bold);

sentencexx = "Your CGPA is: ";

long long int wait = 1;

////THE REAL GAME START FROM HERE/////

while (window.isOpen())

{

if (Keyboard::isKeyPressed(Keyboard::Escape))

wait++;

if (wait % 2 == 0)

{

pausedx.setPosition(500, 200);

window.draw(pausedx);

window.display();

window.clear();

}

if (wait % 2 == 1){

if (flagxx == 0){

window.draw(b1Image);

b1Image.move(-10, 0);

window.draw(b2Image);

b2Image.move(-10, 0);

//////features///////

///// blackhole////

if (hole\_flag == 0){

hole\_image.setPosition(xod + distance[holedex + 1] - 800, 210);

window.draw(hole\_image);

}

if (b1Image.getPosition().x + 800 < 0) b1Image.setPosition(800, 0);

if (b2Image.getPosition().x + 800 < 0) b2Image.setPosition(800, 0);

float time = clock.getElapsedTime().asMilliseconds();

clock.restart();

timer += time;

timer2 = clock2.getElapsedTime().asMilliseconds();

if (life\_flag == 1 && timer2 > 500)

{

timer2 = 0;

life\_flag = 0;

distance[sobstaclexx\_index] = -1000;

livesxx--;

if (livesxx <= 0)

life\_flag = 1;

}

if (life\_flag == 1)

{

if (livesxx <= 0)

{

if (flag3 == 1)

up1 = up1 - 30;

if (up1 <= -60) {

flag4 = 1; flag3 = 0;

}

if (flag4 == 1)

up1 = up1 + 50;

hit\_imagex.setPosition(playerImage2.getPosition().x, 220 + up1);

window.draw(hit\_imagex);

GAME\_OVER2(count);

}

else

{

hit\_imagex.setPosition(playerImage2.getPosition().x - 20, playerImage2.getPosition().y - 10);

window.draw(hit\_imagex);

}

}

player\_heart.setTextureRect(IntRect(2, 2, livesxx \* 65, 60));

window.draw(player\_heart);

Event e;

while (window.pollEvent(e))

{

switch (e.type)

{

case Event::Closed:

window.close();

break;

case Event::KeyPressed:

if (e.key.code == Keyboard::Up)

{

source.y = Jump; j = 0;

}

else if (e.key.code == Keyboard::Down)

{

source.y = Slide; s = 0;

}

break;

}

}

///JUMP////

if (life\_flag == 0)

{

if (source.y == Jump)

{

int dif = 50;

playerImage2.setTextureRect(IntRect(j \* 135, 136, 135, 115));

if (j == 0) playerImage2.setPosition(50 + mmove, player\_height - dif \* 1);

else if (j == 1) playerImage2.setPosition(50 + mmove, player\_height - dif \* 2);

else if (j == 2) playerImage2.setPosition(50 + mmove, player\_height - dif \* 3);

else if (j == 3) playerImage2.setPosition(50 + mmove, player\_height - dif \* 2);

else if (j == 4) playerImage2.setPosition(50 + mmove, player\_height - dif \* 1);

else if (j == 5)

playerImage2.setPosition(50 + mmove, player\_height - dif \* 0);

xod -= 10;

window.draw(playerImage2);

j++;

if (j >= 6)

{

source.x = 0; source.y = Run;

}

}

/////SLIDE//////

else if (source.y == Slide)

{

playerImage2.move(0, 2);

int x1;

if (s == 0)

x1 = 50;

if (s == 0)

{

playerImage2.setTextureRect(IntRect(s \* 145, 2 \* 136, 145, 115));

playerImage2.setPosition(x1 + mmove, playerImage2.getPosition().y - 20);

}

else if (s == 4 || s == 5)

{

playerImage2.setTextureRect(IntRect(s \* 145, (2 \* 136 + 30), 145, 80));

playerImage2.setPosition(x1 + mmove, playerImage2.getPosition().y + 20);

}

else{

playerImage2.setTextureRect(IntRect(s \* 145, (2 \* 136 + 55), 145, 60));

playerImage2.setPosition(x1 + mmove, playerImage2.getPosition().y + 35);

}

xod -= 30;

window.draw(playerImage2);

s++;

if (s >= 6)

{

source.x = 0; source.y = Run;

}

playerImage2.move(0, -2);

}

/////RUN////

else if (source.y == Run){

source.x++;

if (source.x >= 6)

source.x = 0;

playerImage2.setTextureRect(IntRect(source.x \* 95, source.y, 95, 90));

window.draw(playerImage2);

}

///RUN END////

}

////Obstacles moving ////

if (timer > 50 && !life\_flag)

{

xod -= 25;

timer = 0;

}

int c = 0;

for (t = 0; t < 16; t++)

{

if (t % 2 == 0)

sobstaclexx[t].setPosition(xod + distance[t], 292);

else

sobstaclexx[t].setPosition(xod + distance[t], 220);

window.draw(sobstaclexx[t]);

for (i = 0; i < 3; i++)

{

if (t % 2 == 0)

coins[c].setPosition(xod + distance[t] + i \* 65, 220 + coin\_sectx[c]);

if (t % 2 == 1)

coins[c].setPosition(xod + distance[t] + i \* 65, 339 + coin\_sectx[c]);

window.draw(coins[c]);

c++;

}

}

sground.setPosition(0, 400);

//if (sobstaclexx[15].getPosition().x + 196 <= 0) xod = 600;

//// Throwing Ninja star/////

if (Keyboard::isKeyPressed(Keyboard::Space))

{

ninja\_starxflag = 1;

nsf = 0;

ninja\_starx[ns].setPosition(playerImage2.getPosition().x, playerImage2.getPosition().y);

ns++;

if (ns > star\_num) ninja\_starxflag = 0;

}

for (i = 0; i < ns && ninja\_starxflag == 1; i++)

{

if (ninja\_starx[i].getPosition().x - playerImage2.getPosition().x < 20 && ninja\_starx[i].getPosition().x>0 && ninja\_starx[i].getPosition().y == playerImage2.getPosition().y)

{

knife\_effect.setPosition(playerImage2.getPosition().x + 15, playerImage2.getPosition().y - 10);

window.draw(knife\_effect);

}

if (ninja\_starx[ns - 1].getPosition().x <= 1000)

{

//ninja\_starx[i].setTextureRect(IntRect(nsf \* 60, 0, 60, 50));

ninja\_starx[i].move(40, 0);

window.draw(ninja\_starx[i]);

}

}

///throwing super Anives////

int knife1;

Aknife\_flagImage.setPosition(xod + distance[10] + 80, 150);

if (isCollide2(playerImage2, Aknife\_flagImage) == 1)

{

Anife\_flag = 1;

knife1 = 0;

}

if (Anife\_flag == 1)

{

if (Keyboard::isKeyPressed(Keyboard::S))

{

knife1 = 1;

knife2();

}

for (i = 0; i < 5 && knife1 == 1; i++)

{

AnifeImgae[i].move(50 - i \* 5, 0);

window.draw(AnifeImgae[i]);

if (AnifeImgae[4].getPosition().x > 900)

knife1 = 0;

}

for (i = 0; i < 5; i++)

{

if (AnifeImgae[i].getPosition().x > 900)

AnifeImgae[i].setPosition(-20, -20);

}

}

// java is Moving and transistor also////

for (i = 0; i <= 4; i++)

{

if (flag2 == 1)

up = up - 30;

if (up <= -60) flag2 = 0;

if (flag2 == 0)

up = up + 30;

if (up >= 0) flag2 = 1;

java\_Image[i].setTextureRect(IntRect(js \* 112 + 4, 4, 112, 162));

java\_Image[i].setPosition(xod + java\_distance[i], 240);

//java\_Image[i].move(-80 \* 20, 0);

window.draw(java\_Image[i]);

}

for (i = 0; i <= total\_trans; i++)

{

transistor[i].setTextureRect(IntRect(ts \* 112 + 4, 4, 112, 108));

transistor[i].setPosition(xod + tdistance[i], 292);

//transistor[i].move(-80 - i \* 20, 0);

window.draw(transistor[i]);

if (transistor[i].getPosition().x > 0 && transistor[i].getPosition().x < 1200 + 50 || trans\_ball[i].getPosition().x>0)

{

trans\_ball[i].setPosition(xod + t1distance[i] - tx[i] \* 20, 285);

window.draw(trans\_ball[i]);

tx[i]++;

}

}

js++;

js = js % 4;

ts++;

ts = ts % 3;

//if (mummy.getPosition().x - playerImage2.getPosition().x < 600)

//mj = 1;

////mummy from here////

if (sobstaclexx[13].getPosition().x + 200 < 0 && sobstaclexx[13].getPosition().x + 200 > -50)

{

mummy\_clock.restart();

livesxx = 3;

}

if (sobstaclexx[13].getPosition().x + 200 < 0)

{

Anife\_flag = 0;

mummy\_timer = mummy\_clock.getElapsedTime().asMilliseconds();

if (Keyboard::isKeyPressed(Keyboard::Right))

{

if (playerImage2.getPosition().x<400)

mmove += 20;

}

if (Keyboard::isKeyPressed(Keyboard::Left))

{

if (playerImage2.getPosition().x>40)

mmove += -20;

}

if (mummy\_timer < 5000)

{

if (sobstaclexx[13].getPosition().x + 500 < 0) mj = 1;

mummy.setTextureRect(IntRect(109 \* mj, 2, 109, 162));

mummy.setPosition(1000, 230);

mummy.move(-20, 0);

mj++;

mj = mj % 6;

}

// 1st rotation ans left right

if (mummy\_timer > 2000)

{

mummy\_ball.move(-50, 0);

window.draw(mummy\_ball);

}

if (isCollide2(playerImage2, mummy\_ball))

{

life\_flag = 1;

clock2.restart();

mummy\_ball.setPosition(-100, -80);

}

if (mummy\_timer > 1500 && mummy\_timer < 2000) mummy\_ball.setPosition(mummy.getPosition().x - 40, mummy.getPosition().y);

if (mummy\_timer>6500 && mummy\_timer < 6600) mummy\_ball.setPosition(mummy.getPosition().x - 40, mummy.getPosition().y); //ball moving

///mummy moving towards me

if ((mummy\_timer > 11000 && mummy\_timer < 17000) || (mummy\_timer>28000 && mummy\_timer < 31000))

{

if (fabs(playerImage2.getPosition().x - mummy.getPosition().x) > 20)

mummy.move(-20, 0);

}

///sword shield sign is coming and attacking the mummy//

if (mummy\_timer > 8000 && mummy\_timer < 8500) sup0f.setPosition(1200, sground.getPosition().y - 80);

if (mummy\_timer > 8500)

{

sup0f.move(-20, 0);

window.draw(sup0f);

if (isCollide2(sup0f, playerImage2))

{

sup0f.setPosition(-100, -100);

sup0ff = 1;

sword.setPosition(playerImage2.getPosition().x + 50, playerImage2.getPosition().y - 100);

}

}

if (sup0ff == 1)

{

if (Keyboard::isKeyPressed(Keyboard::S))

{

sup0fff = 1;

sup0ff = 0;

}

}

if (sup0fff == 1)

{

sword.move(50, 0);

window.draw(sword);

}

if (isCollide2(sword, mummy) && mummy.getPosition().x < 1000)

mummy.setPosition(sword.getPosition().x + 100, mummy.getPosition().y);

if (sword.getPosition().x>1000) sword.setPosition(-888, 9999);

////mummy is going back

if (mummy\_timer > 17000 && mummy\_timer < 19000)

{

if (mummy.getPosition().x < 1000) mummy.move(30, 0);

}

///1st part done in 19 sec

if (mummy\_timer>18500 && mummy\_timer < 19000)

{

for (i = 1; i < 3; i++)

{

fire\_ball2[i - 1].setPosition(200 \* i - 150, -300 \* i);

}

fire\_ball2[2].setPosition(150, -300 \* i);

}

if (mummy\_timer > 19000)

{

for (i = 0; i < 3; i++)

{

fire\_ball2[i].move(0, 20);

window.draw(fire\_ball2[i]);

}

}

// ///super power 1 is coming////

if (mummy\_timer>24000 && mummy\_timer < 24500) sup1f.setPosition(1200, sground.getPosition().y - 80);

if (mummy\_timer > 24500)

{

sup1f.move(-20, 0);

window.draw(sup1f);

if (isCollide2(sup1f, playerImage2))

{

sup1f.setPosition(-100, -100);

sup1ff = 1;

}

}

if (sup1ff == 1)

{

if (Keyboard::isKeyPressed(Keyboard::S))

{

sup1fff = 1;

sup1ff = 0;

superpower1.setPosition(playerImage2.getPosition().x, playerImage2.getPosition().y - 50);

}

else

{

if (flags1 == 1)

ups1 = ups1 - 20;

if (ups1 <= -80) flags1 = 0;

if (flags1 == 0)

ups1 = ups1 + 20;

if (ups1 >= 0) flags1 = 1;

superpower1.setPosition(playerImage2.getPosition().x, playerImage2.getPosition().y - 180 + ups1);

window.draw(superpower1);

}

}

if (sup1fff == 1)

{

superpower1.move(40, 0);

window.draw(superpower1);

}

if (isCollide2(superpower1, mummy) && mummy.getPosition().x < 1000)

{

mummy.setPosition(superpower1.getPosition().x + 200, mummy.getPosition().y);

if (superpower1.getPosition().x>1000) superpower1.setPosition(-6666, -6666);

}

for (i = 0; i < 3; i++)

{

if (isCollide2(fire\_ball2[i], playerImage2))

{

life\_flag = 1;

clock2.restart();

fire\_ball2[i].setPosition(-100, -80);

}

}

///mummy is getting attack///

if (isCollide2(superpower1, mummy) && superpowercount == 0)

{

devil\_livesxx--;

superpowercount = 1;

}

if (isCollide2(sword, mummy) && swordcount == 0)

{

devil\_livesxx--;

swordcount = 1;

}

devil\_heart.setPosition(1000, 0);

devil\_heart.setTextureRect(IntRect(2, 2, 51 \* devil\_livesxx, 104));

if (devil\_livesxx == 0)

{

boom.setPosition(mummy.getPosition().x, mummy.getPosition().y);

mummy.move(-10, 20);

window.draw(boom);

}

//cout << xod << endl;

if (mummy.getPosition().y > 600)

{

//cout << mummy.getPosition().y << endl;

xod += 25;

playerImage2.move(55, 0);

if (playerImage2.getPosition().x > 1300)

window.close();

}

window.draw(devil\_heart);

icpc.setPosition(mummy.getPosition().x - 20, mummy.getPosition().y - 100);

window.draw(icpc);

window.draw(mummy);

}

///// java , transistor , obstacles are getting attacked ////

int stari, cj;

for (i = 0; i < 60; i++)

{

cj = attack2(ninja\_starx[i]);

if (cj == 1)

ninja\_starx[i].setPosition(-400, -100);

}

for (i = 0; i < 5; i++)

{

int aj = attack2(AnifeImgae[i]);

if (aj == 1)

AnifeImgae[i].setPosition(-400, -100);

}

///Ground and train checking and game over////

int k, x1, y1, r1, r2, m1;

for (k = 0; k < 16; k++)

{

x1 = playerImage2.getPosition().x + 95;

y1 = playerImage2.getPosition().y;

r1 = sobstaclexx[k].getPosition().x;

r2 = sobstaclexx[k].getPosition().x + 130;

m1 = sobstaclexx[k].getPosition().y;

if (x1 >= r1 - 10 && playerImage2.getPosition().x <= r2 && m1 - y1 >= 88)// here i have debugged from game over

{

if (jump\_flag == 0){

playerImage2.setPosition(50 + mmove, m1 - 95);

source.y = Run;

jump\_flag = 1;

}

player\_height = 300 - 100;

break;

}

if (x1 >= r1 + 15 && playerImage2.getPosition().x <= r2 && (m1 + 100 > y1) && life\_flag == 0)

{

life\_flag = 1;

clock2.restart();

sobstaclexx\_index = k;

}

}

for (i = 0; i <= 11; i++)

{

if (isCollide2(playerImage2, trans\_ball[i]))

{

trans\_ball[i].setPosition(-1000, 0);

t1distance[i] = -5000;

life\_flag = 1;

clock2.restart();

}

if (isCollide2(transistor[i], playerImage2))

{

tdistance[i] = -2500;

life\_flag = 1;

clock2.restart();

}

}

for (i = 0; i < 5; i++)

{

if (isCollide2(playerImage2, java\_Image[i]))

{

java\_distance[i] = -5000;

life\_flag = 1;

clock2.restart();

}

}

for (i = 0; i < 2; i++)

{

trainImage[i].setPosition(xod + train\_distance[4 + i] + 400, 310);

window.draw(trainImage[i]);

}

for (i = 0; i < 2; i++){

m1 = trainImage[i].getPosition().y;

if (x1 >= trainImage[i].getPosition().x && x1 <= trainImage[i].getPosition().x + 444)// here i have debugged from game over

{

if (m1 - y1 >= 50)

{

if (jump\_flag == 0){

playerImage2.setPosition(50 + mmove, player\_height - 70);

source.y = Run;

jump\_flag = 1;

}

k = 17;

player\_height = 300 - 70;

break;

}

else

{

train\_distance[4 + i] = -1000;

life\_flag = 1;

clock2.restart();

}

}

}

if (k == 16) {

jump\_flag = 0;

playerImage2.setPosition(playerImage2.getPosition().x, sground.getPosition().y - 90);

player\_height = 300;

}

if (playerImage2.getPosition().x > 50) playerImage2.move(-15, 0);

//// Intersection checking with player and good things happen////

for (i = 0; i < 50; i++)

{

if (isCollide2(playerImage2, coins[i]) == 1)

{

coin\_sectx[i] = -1000;

count++;

int count1 = count;

}

}

if (hole\_image.getPosition().x - playerImage2.getPosition().x <= 20)

{

playerImage2.setPosition(-25000, -15000);

hole\_image.setPosition(-2050, 1500);

hole\_flag = 1;

}

if (xod + distance[8] + 290 < 0)

{

playerImage2.setPosition(50 + mmove, player\_height);

hole\_image.setPosition(50, 210);

hole\_flag = 0;

holedex = 8;

}

if (hole\_flag == 1)

xod -= 100;

/\*for (i = 0; i < 5; i++)

{

window.draw(AnifeImgae[i]);

}

printf("%d\n", count);\*/

float starting\_pause = begin\_clock.getElapsedTime().asSeconds();

textxx\_begin.setPosition(400, 200);

if (starting\_pause <= 2)

textxx\_begin.setString("Second year in CSEDU");

else if (starting\_pause <= 3){

textxx\_begin.setString("1");

text\_begin.setPosition(500, 200);

}

else if (starting\_pause <= 4)

{

textxx\_begin.setString("2");

textxx\_begin.setPosition(500, 200);

}

else if (starting\_pause <= 5)

{

textxx\_begin.setString("3");

textxx\_begin.setPosition(500, 200);

}

else if (starting\_pause <= 6 && starting\_pause<7)

textxx\_begin.setString("Go!! Go!! Go!!");

else textxx\_begin.setPosition(-100, -100);

if (mummy.getPosition().y > 600)

{

textxx\_begin.setPosition(500, 200);

textxx\_begin.setString("2nd!!year!!Passed");

window.draw(textxx\_begin);

GAME\_OVER2(count);

}

//if (playerImage2.getPosition().x > 1300)

//return 2;

cout << Aknife\_flagImage.getPosition().x << endl;

window.draw(textxx\_begin);

window.draw(Aknife\_flagImage);

window.draw(sground);

window.display();

window.clear();

}

}

}

}

**References:**

1. https://www.youtube.com/watch?v=kAbkFY6lwAY&list=PLHJE4y54mpC5j\_x90UkuoMZOdmmL9-\_rg&index=1

2.

http://www.sfml-dev.org/