P a g e | **1**



**University of Dhaka**

**Department of Computer Science and Engineering**

**CSE – 1211**

**Lab Project: Survivor**

P a g e | **2**



Submitted to:

**Md. Shiplu Hawlader**

Lecturer

Hasnain Heckel

Lecturer



Submitted by:

Koushik Deb

[JH-41]

Syeed Abrar Zaoad

[SH-23]

Department of Computer Science and Engineering Session (2015-16)

Date-of-Submission: 24th November, 2016

P a g e | **3**



**Table of contents:**

|  |  |  |
| --- | --- | --- |
| Serial | Topic | Page number |
|  |  |  |
| 01 | Introduction | 4 |
|  |  |  |
| 02 | Project introduction | 4 |
|  |  |  |
| 03 | Game outline | 4 |
|  |  |  |
| 04 | Game objective | 4 |
|  |  |  |
| 05 | Challenges for the gamer | 5 |
|  |  |  |
| 06 | Game features | 5 |
|  |  |  |
| 07 | Source code properties | 7 |
|  |  |  |
| 08 | Graphical interface of the game | 8 |
|  |  |  |
| 09 | Game structure | 12 |
|  |  |  |
| 10 | How the levels work | 13 |
|  |  |  |
| 11 | Challenges in coding | 13 |
|  |  |  |
| 12 | Functions used in source code | 14 |
|  |  |  |
| 13 | Overview | 16 |
|  |  |  |
| 14 | The source code | 17 |
|  |  |  |
| 15 | Special thanks to | 49 |
|  |  |  |
| 16 | Reference | 49 |
|  |  |  |

P a g e | **4**



**Introduction:**

The project is intended to make the students familiar with some application of basic Cpp programming using Simple and Fast Multimedia Library {SFML}. Implementation of the students’ theoretical knowledge of C++ language in some real life scenario was the objective of this project.

**Project introduction:**

The project is a car based 2D game, in which the player has to control a car and escape collisions. The game is named “Survivor”, which means “To remain alive”. We chose this name because the meaning goes with the theme of the game and also, the word sounds pretty good to us.

**Game outline:**

This is a simple, elegant, car based game, where the player has to control one car and has to survive as long as he can.

There are three different level in this game.

In the first level, the player has to use the direction keys to avoid collision. The task becomes a little hard in level two, where the “enemy cars” gain a little bit more speed. Level three is the hardest as the opposite cars come up with full speeds.

The velocity of the cars, coming from opposite direction, increases with level.

Every passing car rewards 1points.

Hitting the car or hitting other obstacles, such as “baby, lady, police barricade” will lead you to “game over”.

**Game objective:**

The Game was designed with the aim to keep the player busy always. Before starting on the project I had thought about the conception of “great games” and “boring games” and finally

P a g e | **5**



decided that a game that requires the player to be alert all the time is less likely to be considered as “boring”.

So, the main objective of the game was to provide its user 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.

**Challenges for the gamer:**

In this game, the gamer will have to master the skill of fast thinking and multitasking capabilities to achieve a high score. There are many obstacles created to make the game more interesting such as baby, lady, police barricades and cars. In addition to that, in level 3, the gamer actually have to concentrate hard as “baby and lady” come fast .Along with the increasing velocity factor, this feature makes this level incredibly hard.

**Game features:**

**Main features:**

1. **Almost zero-flicker game play:**

Using SFML has its advantage. It provides you with zero-flicker game play.

Though there were some problems, but we tried our best and here we stand with almost zero-flicker game play.

****

1. **Interactive menus:**

This game has several interactive menus which are very easy to use, and also greatly eye catching. This feature adds to the charm of this game by creating great first

P a g e | **6**



impressions. The menus and the submenus were created with high quality images [3] and appropriate keyboard and mouse functions.

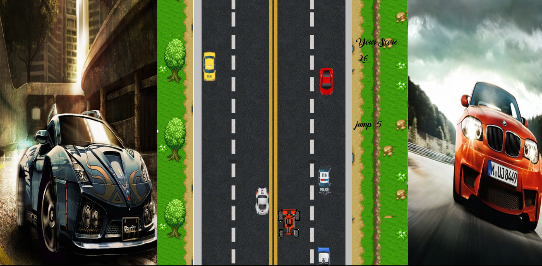
1. **A professional look in the interface:**

With the help of my limited image processing skill and coding skill, I tried to create a good looking graphics interface for my game.

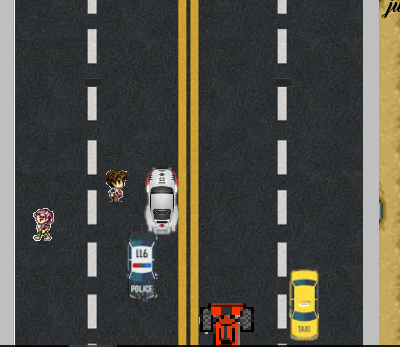
1. **Different levels of game play:**

The game comes with three different levels; each is different in the level of difficulty. This feature adds to the variation of the game.









1. **Sound effects:**

This is another feature that makes this game different. We added sound to make this game more approachable. There will be sound from the very first moment. And sound will vary whenever you execute different sub menus.

**Additional features:**

1. **Main menu:**

The main menu is like a dashboard that gives the user access to any part of the program.

****

1. **Leader board:**

The game also has a leader board option that can be accessed through the main menu. The game stores the top nine scores from each level and shows the highest three of them in the Leader board.

1. **Credit:**

The game has a option which show you the name of people who put effort on making this game



1. **Exit option:**

The game can be terminated at anytime using the EXIT option, which is located at the main menu.

P a g e | **7**



**Bonus features:**

1. **Small animations:**

Small works of animation has been included in the game. In this mode, some simple user defined functions and images were used to simulate move and car collision.

****

1. **Change of appearance:**

The game changes its appearance in every menu and submenu. There is a wide variety of cars as well. A random integer generating function was used to create different game play scenarios in different time.

1. **Image- based number system:**

We used image based number system to show the scores and high scores in the “game over” screen using CPP programming. The system breaks a number into its digits and finds the respective images in a file for each digit.



**Source Code properties:**

1. The source was written by following an efficient modular programming system, dividing the whole code in different separate parts and making new user defined header files. This method was encouraged by our teachers and it made our tasks a lot easier while combining all the features and joining the whole project together.

P a g e | **8**



1. I tried to use as many different types of programming tools as possible, so that it covers the maximum amount of what we have been taught in C language. The coding part of this project contains-
   1. Basic input output
   2. File input output
   3. Conditional logic
   4. For loop and While loop
   5. Single arrays
   6. String handling
   7. Mouse operations
   8. Keyboard operations
   9. Pointers
   10. Structures
   11. Sorting
   12. Reading image and music files
   13. Graphics operations.

And so on.

**Graphical interface of the game:**

**The main menu:**



P a g e |**9**



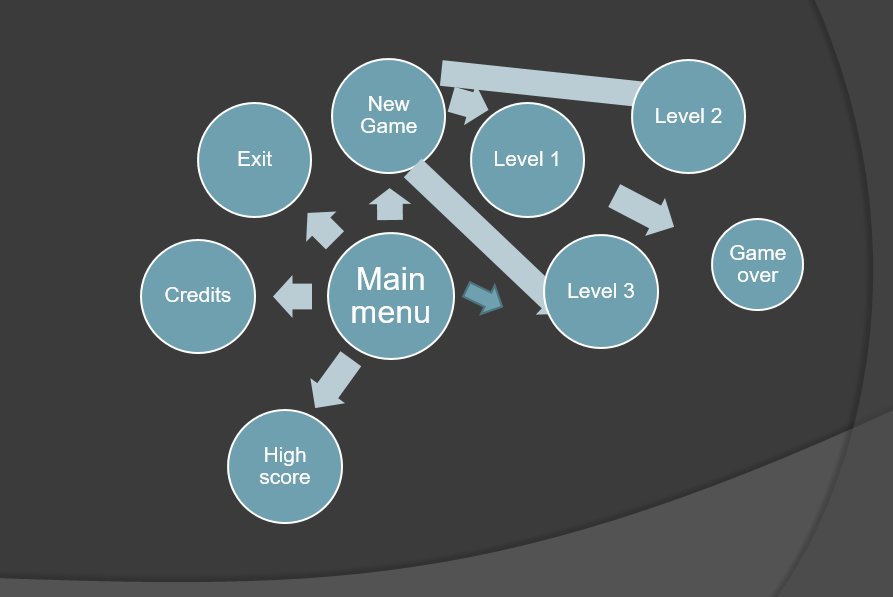
**The level choosing menu:**



P a g e |**10**



**The game structure:**

****

The game flow is shown in the figure drawn above. After opening the program, it takes the user to the main menu. From the main menu, the user can take one of four possible options. These options are the “New Game” submenu, the “High score” screen, the “Credit” screen, and the “Exit” option.

In the new game submenu, the user can choose between the three level of the game. Once a choice has been made, the program will take the user to his desired destination.

The High score feature stores the topmost score from each level. However, the feature only prints the top result on the screen.

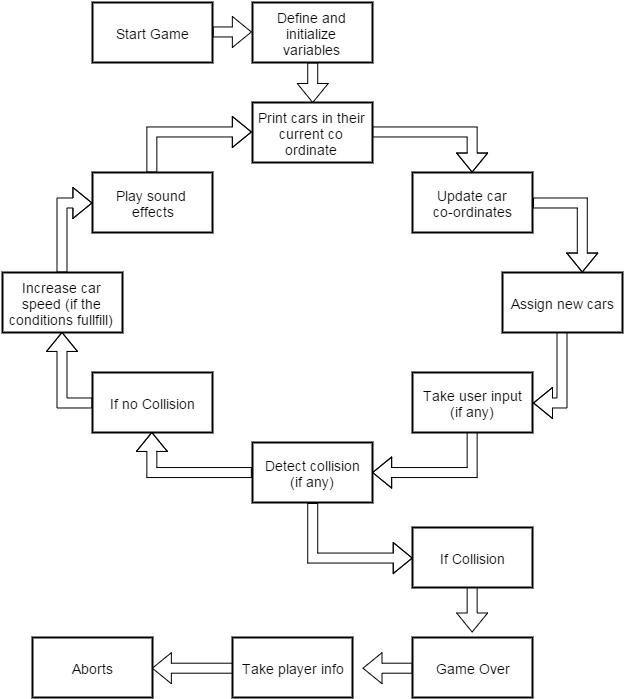
All menus, submenus and options are completely reversible, that is the user can roam freely from any sector of the interface to the other.

P a g e | **13**



**How the levels work:**

The logic steps of the game play operation are not very complex. The basic idea is shown below with the help of a flow diagram.



**Challenges in coding:**

**Avoiding impossible case:**

As the game is an infinite game (at least until the player kills himself), sometimes an impossible case might rise where there is no escape and the game must end. There is jump function to avoid this kind of possibilities. Again, if jump is not available, then this possible scenario was avoided by calculating the L.C.M of the car co-ordinates and updating the game accordingly.

**Handling bugs:**

The interactive menu is done by calling blocks. A certain amount of blocks are made to make this game easier to code, such as main\_menu,scoring,credit,game1,exit\_game etc. We used one header file, too.

**The functions used in the source code:**

**User defined functions:**

* Didn’t use any.

**Created Block:**

* Main\_menu:
* Scoring:
* Credit\_game:
* Game1:
* Exit\_game:

P a g e | **15**



**Functions from “graphics.hpp”:**

* asMilliseconds() : [sf::Time](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Time.php#a85e6deb41fa71896508ce0f64059a6ae)
* draw() : sf::RenderTarget , sf::Drawable
* Font() : sf::Font
* getBody() : sf::Http::Response
* getCharacterSize() : [sf::Text](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Text.php#a9abb85c6966c9879f6ba4d6e47be1dd5)
* getColor() : [sf::Sprite](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Sprite.php#ab05fafd4e9999608a5cae4985e7b52df) , [sf::Text](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Text.php#ae42818342a74a9d04644e2fbbd4ca29a)
* getFillColor() : [sf::Shape](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Shape.php#ad7f7fe601a8bb24efe9aa77809a35c12)
* getFont() : [sf::Text](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Text.php#ab831de193307ab591b34221440613aa1)
* getGlobalBounds() : [sf::Shape](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Shape.php#a5257341fe832884dbba6b9dc855e33cc) , [sf::Sprite](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Sprite.php#a203d2d8087bfdca2ebc3c0485cdb7409) , [sf::Text](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Text.php#a95d732f58bd12bf7ec388b106f3729ba)
* getOrigin() : [sf::Transformable](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Transformable.php#a6bddc485d22bb64449d9d2d3a99a778f)
* getString() : [sf::Text](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Text.php#a14d580e8afdd43c210429505310ecc95)
* getSize() : [sf::RenderTexture](http://www.sfml-dev.org/documentation/2.0/classsf_1_1RenderTexture.php#a757ba45ec7a7deefcaef717049b00b8c) , [sf::Image](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Image.php#a5c3e9bebdc001c3ebf85ca97039fc86b) , [sf::RenderWindow](http://www.sfml-dev.org/documentation/2.0/classsf_1_1RenderWindow.php#a2c7ff414be32621a453745cf2a0f8a3e) , [sf::Texture](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Texture.php#a0f370acd8f41c8b97a6959389c521c2c) , [sf::View](http://www.sfml-dev.org/documentation/2.0/classsf_1_1View.php#aa130cf34676d715242bee661537a6257) , [sf::String](http://www.sfml-dev.org/documentation/2.0/classsf_1_1String.php#a635d75c4cd830d5f639a41815dd0ce23) , [sf::RectangleShape](http://www.sfml-dev.org/documentation/2.0/classsf_1_1RectangleShape.php#acaacbaee87c38a526a9d895742faab54) , [sf::Window](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Window.php#ad2b55a731ba1680fe67292991ef1610e) , [sf::RenderTarget](http://www.sfml-dev.org/documentation/2.0/classsf_1_1RenderTarget.php#a2e5ade2457d9fb4c4907ae5b3d9e94a5) , [sf::InputStream](http://www.sfml-dev.org/documentation/2.0/classsf_1_1InputStream.php#a311eaaaa65d636728e5153b574b72d5d)
* isButtonPressed() : [sf::Mouse](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Mouse.php#ab647159eb88e369a0332a9c5a7ba6687) , [sf::Joystick](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Joystick.php#ae0d97a4b84268cbe6a7078e1b2717835)
* isKeyPressed() : [sf::Keyboard](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Keyboard.php#a80a04b2f53005886957f49eee3531599)
* isOpen() : [sf::Window](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Window.php#a5aa9c2b2b0e51d3423c2b66c80253337)
* loadFromFile() : [sf::SoundBuffer](http://www.sfml-dev.org/documentation/2.0/classsf_1_1SoundBuffer.php#a2be6a8025c97eb622a7dff6cf2594394) , [sf::Shader](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Shader.php#ac9d7289966fcef562eeb92271c03e3dc) , [sf::Texture](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Texture.php#a8e1b56eabfe33e2e0e1cb03712c7fcc7) , [sf::Font](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Font.php#ab020052ef4e01f6c749a85571c0f3fd1) , [sf::Image](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Image.php#a9e4f2aa8e36d0cabde5ed5a4ef80290b) , [sf::Shader](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Shader.php#a053a5632848ebaca2fcd8ba29abe9e6e)
* move() : [sf::View](http://www.sfml-dev.org/documentation/2.0/classsf_1_1View.php#a0c82144b837caf812f7cb25a43d80c41) , [sf::Transformable](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Transformable.php#ab9ca691522f6ddc1a40406849b87c469) , [sf::View](http://www.sfml-dev.org/documentation/2.0/classsf_1_1View.php#a4c98a6e04fed756dfaff8f629de50862) , [sf::Transformable](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Transformable.php#a86b461d6a941ad390c2ad8b6a4a20391)
* Music() : [sf::Music](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Music.php#a0bc787d8e022b3a9b89cf2c28befd42e)
* pause() : [sf::SoundStream](http://www.sfml-dev.org/documentation/2.0/classsf_1_1SoundStream.php#a932ff181e661503cad288b4bb6fe45ca) , [sf::Sound](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Sound.php#a5eeb25815bfa8cdc4a6cc000b7b19ad5)
* play() : [sf::Sound](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Sound.php#a2953ffe632536e72e696fd880ced2532) , [sf::SoundStream](http://www.sfml-dev.org/documentation/2.0/classsf_1_1SoundStream.php#afdc08b69cab5f243d9324940a85a1144)
* pollEvent() : [sf::Window](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Window.php#a338e996585faf82e93069858e3b531b7)
* read() : [sf::InputStream](http://www.sfml-dev.org/documentation/2.0/classsf_1_1InputStream.php#a8dd89c74c1acb693203f50e750c6ae53)
* Rect() : [sf::Rect< T >](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Rect.php#a27fdf85caa6d12caeeff78913cc59936)
* RectangleShape() : [sf::RectangleShape](http://www.sfml-dev.org/documentation/2.0/classsf_1_1RectangleShape.php#a83a2be157ebee85c95ed491c3e78dd7c)
* RenderWindow() : [sf::RenderWindow](http://www.sfml-dev.org/documentation/2.0/classsf_1_1RenderWindow.php#a25c0af7d515e710b6eebc9c6be952aa5)
* reset() : [sf::View](http://www.sfml-dev.org/documentation/2.0/classsf_1_1View.php#ac95b636eafab3922b7e8304fb6c00d7d)
* resetBuffer() : [sf::Sound](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Sound.php#acb7289d45e06fb76b8292ac84beb82a7)
* setCharacterSize() : [sf::Text](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Text.php#ae96f835fc1bff858f8a23c5b01eaaf7e)
* setColor() : [sf::Sprite](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Sprite.php#a14def44da6437bfea20c4df5e71aba4c) , [sf::Text](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Text.php#afd1742fca1adb6b0ea98357250ffb634)
* setBuffer() : [sf::Sound](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Sound.php#a8b395e9713d0efa48a18628c8ec1972e)
* setFillColor() : [sf::Shape](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Shape.php#a3506f9b5d916fec14d583d16f23c2485)
* setFont() : [sf::Text](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Text.php#a2927805d1ae92d57f15034ea34756b81)
* setFramerateLimit() : [sf::Window](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Window.php#af4322d315baf93405bf0d5087ad5e784)
* setLoop() : [sf::Sound](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Sound.php#af23ab4f78f975bbabac031102321612b) , [sf::SoundStream](http://www.sfml-dev.org/documentation/2.0/classsf_1_1SoundStream.php#a43fade018ffba7e4f847a9f00b353f3d)
* setPosition() : [sf::Transformable](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Transformable.php#af1a42209ce2b5d3f07b00f917bcd8015) , [sf::Listener](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Listener.php#a5bc2d8d18ea2d8f339d23cbf17678564) , [sf::SoundSource](http://www.sfml-dev.org/documentation/2.0/classsf_1_1SoundSource.php#a0480257ea25d986eba6cc3c1a6f8d7c2) , [sf::Transformable](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Transformable.php#a4dbfb1a7c80688b0b4c477d706550208) , [sf::Mouse](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Mouse.php#a1222e16c583be9e3d176d86e0b7817d7) , [sf::Window](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Window.php#a6c4078bfbf61c29bfc4b4732ce764f17)
* setString() : [sf::Text](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Text.php#a7d3b3359f286fd9503d1ced25b7b6c33)
* setStyle() : [sf::Text](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Text.php#ad791702bc2d1b6590a1719aa60635edf)
* setTexture() : [sf::Sprite](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Sprite.php#a3729c88d88ac38c19317c18e87242560) , [sf::Shape](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Shape.php#af8fb22bab1956325be5d62282711e3b6)
* setTextureRect() : [sf::Sprite](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Sprite.php#a3fefec419a4e6a90c0fd54c793d82ec2) , [sf::Shape](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Shape.php#a2029cc820d1740d14ac794b82525e157)
* setTitle() : [sf::Window](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Window.php#a3b3f3513bb6be90f5cd456c20b5fd5fa)
* SoundBuffer() : [sf::SoundBuffer](http://www.sfml-dev.org/documentation/2.0/classsf_1_1SoundBuffer.php#a0cabfbfe19b831bf7d5c9592d92ef233)
* Vector2() : [sf::Vector2< T >](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Vector2.php#a58c32383b5291380db4b43a289f75988)
* Window() : [sf::Window](http://www.sfml-dev.org/documentation/2.0/classsf_1_1Window.php#a5359122166b4dc492c3d25caf08ccfc4)

P a g e | **16**



**The list of header files:**

1. highscore.h

**Overview and conclusion:**

The project was a very good way to encourage young students to develop their skills and make something useful with their knowledge. This project is a perfect example of all the exceptional things that can be done with some basic CPP programming.

This project will also encourage young minds who want to be a game developer in the future. The project promotes originality and creativity and the skill of working in a team.

The project also summarizes the basic tools of C language as they had to be implemented while making the game.

P a g e | **17**



**The source code:**

**The “main.cpp” file:**

#include <SFML/Window.hpp>

#include <SFML/Graphics.hpp>

#include <SFML/Audio.hpp>

#include <iostream>

#include <SFML/System.hpp>

#include <Windows.h>

#include "time.h"

#include <sstream>

#include <string.h>

#include <fstream>

#include "HighScore.h"

#include <random>

using namespace std;

using namespace sf;

int ppp;

int c = 0;

int zz = 1;

int cr = 1;

int j = 1;

int lev1 = 1;

int lev2= 2;

int lev3 = 3;

bool game = true;

bool release = true;

int main()

{

j = 1;

cr = 1;

srand(time(0));

sf::RenderWindow window(sf::VideoMode(1370, 680), "Moving a shape");

window.setVerticalSyncEnabled(true);

int count = 0;

static int flag=0;

//////sound/////

/\* sf::SoundBuffer buffer0,buffer1,buffer2;

sf::Sound gari,horn,crash ;

if (!buffer0.loadFromFile("car1.wav"))

{

std::cout << "Error" << std::endl;

}

P a g e | **18**



gari.setBuffer(buffer0);

gari.setVolume(60);

if (!buffer1.loadFromFile("horn.wav"))

{

std::cout << "Error" << std::endl;

}

horn.setBuffer(buffer1);

horn.setVolume(80);

if (!buffer2.loadFromFile("crash.wav"))

{

std::cout << "Error" << std::endl;

}

crash.setBuffer(buffer2);

crash.setVolume(80);

\*/

goto main\_menu;

main\_menu:{

///////////main menu image//////////////////////

count = 0;

zz = 1;

j = 1;

cr = 1;

sf::Texture main\_menu;

main\_menu.loadFromFile("front.jpg");

sf::Sprite menu;

menu.setTexture(main\_menu);

////////////////////////////////////////////////

sf::Music music, music1;

music.openFromFile("menusound.wav");

music.setVolume(40);

music1.openFromFile("highscoremusic.wav");

music1.setVolume(50);

////////////////////////////////////////////////////////////

music.play();

//////////////////////main menu//////////////////////////////

sf::Font title;

title.loadFromFile("text.ttf");

sf::Text ttl("Surviving Car", title, 90);

ttl.setStyle(sf::Text::Bold);

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

ttl.setPosition(900 / 2 - ttl.getGlobalBounds().width / 2, 150);

sf::Mouse::setPosition(sf::Vector2i(10, 100), window);

sf::Text menu\_text[4];

window.draw(menu);

window.draw(ttl);

string str[] = { "New Game", "High Score", "Credit", "Exit" };

int k;

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

{

menu\_text[k].setFont(title);

menu\_text[k].setCharacterSize(45);

menu\_text[k].setString(str[k]);

menu\_text[k].setColor(sf::Color::Red);

menu\_text[k].setPosition(550, 300 + k \* 100);

window.draw(menu\_text[k]);

}

while (window.isOpen())

{

sf::Event event;

sf::Vector2f mouse(sf::Mouse::getPosition(window));

while (window.pollEvent(event))

{

switch (event.type)

{

case sf::Event::Closed:

window.close();

break;

case sf::Event::MouseButtonPressed:

if (release && (menu\_text[0].getGlobalBounds().contains(mouse)))

goto level;

if (release && (menu\_text[1].getGlobalBounds().contains(mouse)))

{

music.stop();

music1.play();

HighScore h;

HighScore HighScore();

h.draw(window);

goto main\_menu;

}

if (release && (menu\_text[2].getGlobalBounds().contains(mouse)))

goto credit\_game;

if (release && (menu\_text[3].getGlobalBounds().contains(mouse)))

goto exit\_game;

release = false;

break;

case sf::Event::MouseButtonReleased:

release = true;

break;

}

}

window.draw(menu);

window.draw(ttl);

if (menu\_text[0].getGlobalBounds().contains(mouse))

{

menu\_text[0].setColor(sf::Color::Blue);

menu\_text[0].setCharacterSize(55);

}

else

{

menu\_text[0].setColor(sf::Color::Red);

menu\_text[0].setCharacterSize(45);

}

if (menu\_text[1].getGlobalBounds().contains(mouse))

{

menu\_text[1].setColor(sf::Color::Green);

menu\_text[1].setCharacterSize(55);

}

else

{

menu\_text[1].setColor(sf::Color::Red);

menu\_text[1].setCharacterSize(45);

}

if (menu\_text[2].getGlobalBounds().contains(mouse))

{

menu\_text[2].setColor(sf::Color::Cyan);

menu\_text[2].setCharacterSize(55);

}

else

{

menu\_text[2].setColor(sf::Color::Red);

menu\_text[2].setCharacterSize(45);

}

if (menu\_text[3].getGlobalBounds().contains(mouse))

{

menu\_text[3].setColor(sf::Color::Magenta);

menu\_text[3].setCharacterSize(55);

}

else

{

menu\_text[3].setColor(sf::Color::Red);

menu\_text[3].setCharacterSize(45);

}

window.draw(menu\_text[0]);

window.draw(menu\_text[1]);

window.draw(menu\_text[2]);

window.draw(menu\_text[3]);

window.display();

}

}

level:{

///////////main menu image//////////////////////

count = 0;

zz = 1;

sf::Texture level;

level.loadFromFile("front.jpg");

sf::Sprite level0;

level0.setTexture(level);

////////////////////////////////////////////////

sf::Music music, music1;

music.openFromFile("menusound.wav");

music.setVolume(40);

music1.openFromFile("highscoremusic.wav");

music1.setVolume(50);

////////////////////////////////////////////////////////////

music.play();

//////////////////////main menu//////////////////////////////

sf::Font title;

title.loadFromFile("text.ttf");

sf::Text ttl("LEVEL", title, 90);

ttl.setStyle(sf::Text::Bold);

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

ttl.setPosition(900 / 2 - ttl.getGlobalBounds().width / 2, 150);

sf::Mouse::setPosition(sf::Vector2i(10, 100), window);

sf::Text level\_text[3];

window.draw(level0);

window.draw(ttl);

string str[] = { "Level 1", "Level 2", "Level 3" };

int k;

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

{

level\_text[k].setFont(title);

level\_text[k].setCharacterSize(45);

level\_text[k].setString(str[k]);

level\_text[k].setColor(sf::Color::Red);

level\_text[k].setPosition(550, 300 + k \* 100);

window.draw(level\_text[k]);

}

while (window.isOpen())

{

sf::Event event;

sf::Vector2f mouse(sf::Mouse::getPosition(window));

while (window.pollEvent(event))

{

switch (event.type)

{

case sf::Event::Closed:

window.close();

break;

case sf::Event::MouseButtonPressed:

if (release && (level\_text[0].getGlobalBounds().contains(mouse)))

{

count = 0;

goto game1;

lev1 = 2;

}

if (release && (level\_text[1].getGlobalBounds().contains(mouse)))

{

count = 51;

goto game1;

lev2 = 2;

}

if (release && (level\_text[2].getGlobalBounds().contains(mouse)))

{

count = 95;

goto game1;

lev3 = 2;

}

release = false;

break;

case sf::Event::MouseButtonReleased:

release = true;

break;

}

}

window.draw(level0);

window.draw(ttl);

if (level\_text[0].getGlobalBounds().contains(mouse))

{

level\_text[0].setColor(sf::Color::White);

level\_text[0].setCharacterSize(55);

}

else

{

level\_text[0].setColor(sf::Color::Red);

level\_text[0].setCharacterSize(45);

}

if (level\_text[1].getGlobalBounds().contains(mouse))

{

level\_text[1].setColor(sf::Color::White);

level\_text[1].setCharacterSize(55);

}

else

{

level\_text[1].setColor(sf::Color::Red);

level\_text[1].setCharacterSize(45);

}

if (level\_text[2].getGlobalBounds().contains(mouse))

{

level\_text[2].setColor(sf::Color::White);

level\_text[2].setCharacterSize(55);

}

else

{

level\_text[2].setColor(sf::Color::Red);

level\_text[2].setCharacterSize(45);

}

window.draw(level\_text[0]);

window.draw(level\_text[1]);

window.draw(level\_text[2]);

window.display();

}

}

credit\_game:

{

sf::RenderWindow window(sf::VideoMode(1070, 690), "Moving a shape");

Texture t50;

if (!t50.loadFromFile("Capture.png"))

cout << "error" << endl;

Sprite Capture(t50);

Capture.setPosition(0, 0);

while (window.isOpen()){

sf::Event event;

while (window.pollEvent(event))

{

switch (event.type)

{

case sf::Event::Closed:

window.close();

break;

}

}

if (sf::Keyboard::isKeyPressed)

{

if (event.key.code == sf::Keyboard::Escape)

goto main\_menu;

}

window.clear();

window.draw(Capture);

window.display();

}

}

game1:

{

Texture t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13, t14, t30, t31, t32, t33;

Texture t15, t16, t17, t18, t19, t20, t21, t22, t23, t24, t25, t26, t27, t28, t34,t35;

t15.loadFromFile("hgrass.png");

t30.loadFromFile("go.png");

t16.loadFromFile("carcanfly.png");

t1.loadFromFile("mcar1.png");

t2.loadFromFile("road1.png");

t3.loadFromFile("roadrun.png");

t4.loadFromFile("roadrun.png");

t5.loadFromFile("police.png");

t6.loadFromFile("truck.png");

t7.loadFromFile("road1.png");

t8.loadFromFile("truck1.png");

t9.loadFromFile("ambulance.png");

t10.loadFromFile("road1.png");

t11.loadFromFile("lalcar.png");

t12.loadFromFile("race2.png");

t13.loadFromFile("race3.png");

t14.loadFromFile("race4.png");

t17.loadFromFile("sidewalk.png");

t18.loadFromFile("sidewalk.png");

t31.loadFromFile("firet.png");

t19.loadFromFile("mcar2.png");

t20.loadFromFile("mcar3.png");

t21.loadFromFile("mcar4.png");

t22.loadFromFile("bons.png");

t23.loadFromFile("b.png");

t24.loadFromFile("sideimage.png");

t25.loadFromFile("grass.png");

t26.loadFromFile("rasta.png");

t27.loadFromFile("hgrass.png");

t32.loadFromFile("sideimage1.png");

t33.loadFromFile("ladeywlk.png");

t34.loadFromFile("baby.png");

t35.loadFromFile("countline.png");

Sprite car1(t1), ima(t2), border(t3), border1(t4), fish2(t5), fish3(t6), ima1(t7), truck1(t8), ambulance(t9);

Sprite race1(t11), race2(t12), race3(t13), race4(t14), side1(t17), side2(t18), grass(t25);

Sprite h1(t15), carfly(t16), car2(t19), car3(t20), car4(t21), bonus(t22), policeb(t23), sidebackground(t24), h0(t26), hf(t27), fire(t31), sidebackground1(t32);

Sprite lady(t33), baby(t34),line(t35);

ima.setTexture(t2);

ima1.setTexture(t7);

//////////music/////

sf::Music musicc;

musicc.openFromFile("garirawaj");

musicc.setVolume(60);

musicc.setLoop(true);

musicc.play();

sf::SoundBuffer buffer1,buffer2;

sf::Sound horn,crash;

if (!buffer1.loadFromFile("horn.wav"))

{

std::cout << "Error" << std::endl;

}

horn.setBuffer(buffer1);

horn.setVolume(80);

if (!buffer2.loadFromFile("crash.wav"))

{

std::cout << "Error" << std::endl;

}

crash.setBuffer(buffer2);

crash.setVolume(80);

if (zz == 1){

game = true;

cout << "zz 1" << endl;

}

else {

game = false;

}

int walk = 0;

int lal = 1;

int c = 1;

int bby = 1;

int ldy = 1;

int po = 1;

int fly = 30;

int car = 0;

int r3 = 1;

int r2 = 1;

int hol = 1;

int r1 = 1;

int blu = 1;

int am = 4;

int tr = 3;

int f3 = 2;

int f2 = 1;

int r4 = 1;

int beg = 1;

int num = 3;

int jump = 3, delay = 50;

int kk = 1;

int pol = 1;

int trk = 1; int fis2 = 1; int fis3 = 1; int ambu = 1; int r1st ; int r2nd = 1; int r3rd = 1; int r4th = 1;

while (window.isOpen()){

sf::Event event;

while (window.pollEvent(event))

{

switch (event.type)

{

case sf::Event::Closed:

window.close();

break;

}

}

window.clear();

if (game){

cout << "zz 1" << endl;

c++;

fly++;

car++;

delay--;

r1++;

po++;

if (fly > 24){

if (car1.getGlobalBounds().intersects(truck1.getGlobalBounds()) || car1.getGlobalBounds().intersects(race1.getGlobalBounds()) || car1.getGlobalBounds().intersects(race2.getGlobalBounds()) || car1.getGlobalBounds().intersects(race3.getGlobalBounds()) || car1.getGlobalBounds().intersects(race4.getGlobalBounds()) || car1.getGlobalBounds().intersects(fish2.getGlobalBounds()) || car1.getGlobalBounds().intersects(fish3.getGlobalBounds()) || car1.getGlobalBounds().intersects(ambulance.getGlobalBounds()) || car1.getGlobalBounds().intersects(policeb.getGlobalBounds()) || car1.getGlobalBounds().intersects(lady.getGlobalBounds()) || car1.getGlobalBounds().intersects(baby.getGlobalBounds()))

{

j = j+1;

cout << "j=2" << endl;

//goto gameovr;

}

}

if (j == 2){//carmove

if (event.type == sf::Event::KeyPressed){

if (sf::Keyboard::isKeyPressed(sf::Keyboard::Up)){

if (car1.getPosition().y > -600){

car1.move(0, -5);

car2.move(0, -5);

car3.move(0, -5);

car4.move(0, -5);

fire.move(0, -5);

}

Sleep(1);

}

}

if (event.type == sf::Event::KeyPressed){

if (sf::Keyboard::isKeyPressed(sf::Keyboard::Down)){

if (car1.getPosition().y < 0){

car1.move(0, 5);

car2.move(0, 5);

car3.move(0, 5);

car4.move(0, 5);

fire.move(0, 5);

}

Sleep(1);

}

}

if (event.type == sf::Event::KeyPressed){

if (sf::Keyboard::isKeyPressed(sf::Keyboard::Left)){

if (car1.getPosition().x > -47){

car1.move(-5, 0);

car2.move(-5, 0);

car3.move(-5, 0);

car4.move(-5, 0);

fire.move(-5, 0);

}

Sleep(1);

}

}

if (event.type == sf::Event::KeyPressed){

if (sf::Keyboard::isKeyPressed(sf::Keyboard::Right)){

if (car1.getPosition().x < 260){

car1.move(5, 0);

car2.move(5, 0);

car3.move(5, 0);

car4.move(5, 0);

fire.move(5, 0);

}

Sleep(1);

}

}

//fly car

if (delay < 70){

if (event.type == sf::Event::KeyPressed){

if (sf::Keyboard::isKeyPressed(sf::Keyboard::LShift)){

if (jump > 0){

carfly.setPosition(car1.getPosition().x, car1.getPosition().y);

//car1.setPosition(100,100 );

car1.move(0, -4);

car2.move(0, -4);

car3.move(0, -4);

car4.move(0, -4);

fire.move(0, -4);

fly = 0;

jump = jump - 1;

delay = 100;

}

}

}

}

}

//vehiles stand still when people walk over

if (fly < 25)

{

carfly.move(0, -4);

}

if (fly == 24){

car1.move(0, -96);

car2.move(0, -96);

car3.move(0, -96);

car4.move(0, -96);

fire.move(0, -96);

}

//jump bonus

if (border.getPosition().y == 50){

border.setPosition(0, 0);

}

if (border1.getPosition().y == 50){

border1.setPosition(0, 0);

}

//car return

//////////////////////////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*leval1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///////////

if (count <= 50)

{

if (c % 500 == 499&&fish2.getPosition().y>750){

fis2 = 1;

int xx = rand() % 4;

if (xx == 0)

{

fish2.setPosition(0, 0);

}

if (xx == 1)

{

fish2.setPosition(100, 0);

}

if (xx == 2)

{

fish2.setPosition(190, 0);

}

if (xx == 3)

{

fish2.setPosition(270, 0);

}

}

if (c % 450 == 449 && fish3.getPosition().y>750)

{

fis3 = 1;

int xy = rand() % 4;

if (xy == 0)

{

fish3.setPosition(0, 0);

}

if (xy == 1)

{

fish3.setPosition(-90, 0);

}

if (xy == 2)

{

fish3.setPosition(90, 0);

}

if (xy == 3)

{

fish3.setPosition(180, 0);

}

}

if (c % 480 == 479 && truck1.getPosition().y>750)

{

trk = 1;

int xy = rand() % 4;

if (xy == 0)

{

truck1.setPosition(0, 0);

}

if (xy == 1)

{

truck1.setPosition(-90, 0);

}

if (xy == 2)

{

truck1.setPosition(-180, 0);

}

if (xy == 3)

{

truck1.setPosition(95, 0);

}

}

//bonusmove

if (c % 1000 == 900)

{

int xy = rand() % 4;

if (xy == 0)

bonus.setPosition(0, 0);

if (xy == 1)

bonus.setPosition(-90, 0);

if (xy == 2)

bonus.setPosition(-180, 0);

if (xy == 3)

bonus.setPosition(95, 0);

}

if (c % 600 == 599 && ambulance.getPosition().y>750)

{

ambu = 1;

int xy = rand() % 4;

ambulance.setPosition(0, 0);

if (xy == 0)

{

ambulance.setPosition(-270, 0);

}

if (xy == 1)

{

ambulance.setPosition(-190, 0);

}

if (xy == 2)

{

ambulance.setPosition(-90, 0);

}

if (xy == 3)

{

ambulance.setPosition(0, 0);

}

}

if (r1 % 600 == 590 && race1.getPosition().y>810){

//r1st = 1;

int xy = rand() % 4;

if (xy == 0){

race1.setPosition(0, 0);

}

if (xy == 1){

race1.setPosition(100, 0);

}

if (xy == 2){

race1.setPosition(190, 0);

}

if (xy == 3)

{

race1.setPosition(270, 0);

}

}

if (c % 700 == 669 && race2.getPosition().y>810){

int xy = rand() % 4;

r2nd = 1;

if (xy == 0){

race2.setPosition(0, 0);

}

if (xy == 1)

{

race2.setPosition(-90, 0);

}

if (xy == 2)

{

race2.setPosition(180, 0);

}

if (xy == 3)

{

race2.setPosition(95, 0);

}

}

if (c % 680 == 620 && race3.getPosition().y>810){

r3rd = 1;

int xy = rand() % 4;

if (xy == 0)

{

race3.setPosition(0, 0);

}

if (xy == 1)

{

race3.setPosition(-90, 0);

}

if (xy == 2)

{

race3.setPosition(-180, 0);

}

if (xy == 3)

{

race3.setPosition(95, 0);

}

}

if (c % 760 == 700 && race4.getPosition().y>810){

r4th = 1;

int xy = rand() % 4;

if (xy == 0)

{

race4.setPosition(0, 0);

}

if (xy == 1)

{

race4.setPosition(-90, 0);

}

if (xy == 2)

{

race4.setPosition(-175, 0);

}

if (xy == 3)

{

race4.setPosition(-275, 0);

}

}

}

////////////////////////////////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*lavel2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?/////////////////////

if (count > 50 && count <= 100)

{

if (c % 430 == 420){

fis2 = 1;

int xx = rand() % 4;

if (xx == 0)

{

fish2.setPosition(0, 0);

}

if (xx == 1)

{

fish2.setPosition(100, 0);

}

if (xx == 2)

{

fish2.setPosition(190, 0);

}

if (xx == 3)

{

fish2.setPosition(270, 0);

}

}

if (c % 400 == 399)

{

fis3 = 1;

int xy = rand() % 4;

if (xy == 0)

{

fish3.setPosition(0, 0);

}

if (xy == 1)

{

fish3.setPosition(-90, 0);

}

if (xy == 2)

{

fish3.setPosition(90, 0);

}

if (xy == 3)

{

fish3.setPosition(180, 0);

}

}

if (c % 430 == 429)

{

trk = 1;

int xy = rand() % 4;

if (xy == 0)

{

truck1.setPosition(0, 0);

}

if (xy == 1)

{

truck1.setPosition(-90, 0);

}

if (xy == 2)

{

truck1.setPosition(-180, 0);

}

if (xy == 3)

{

truck1.setPosition(95, 0);

}

}

//bonusmove

if (c % 1200 == 1000)

{

int xy = rand() % 4;

if (xy == 0)

bonus.setPosition(0, 0);

if (xy == 1)

bonus.setPosition(-90, 0);

if (xy == 2)

bonus.setPosition(-180, 0);

if (xy == 3)

bonus.setPosition(95, 0);

}

if (c % 510 == 500)

{

ambu = 1;

int xy = rand() % 4;

ambulance.setPosition(0, 0);

if (xy == 0)

{

ambulance.setPosition(-270, 0);

}

if (xy == 1)

{

ambulance.setPosition(-190, 0);

}

if (xy == 2)

{

ambulance.setPosition(-90, 0);

}

if (xy == 3)

{

ambulance.setPosition(0, 0);

}

}

if (r1 % 510 == 500){

//r1st = 1;

int xy = rand() % 4;

if (xy == 0){

race1.setPosition(0, 0);

}

if (xy == 1){

race1.setPosition(100, 0);

}

if (xy == 2){

race1.setPosition(190, 0);

}

if (xy == 3)

{

race1.setPosition(270, 0);

}

}

if (c % 500 == 490){

int xy = rand() % 4;

r2nd = 1;

if (xy == 0){

race2.setPosition(0, 0);

}

if (xy == 1)

{

race2.setPosition(-90, 0);

}

if (xy == 2)

{

race2.setPosition(180, 0);

}

if (xy == 3)

{

race2.setPosition(95, 0);

}

}

if (c % 460 == 459){

r3rd = 1;

int xy = rand() % 4;

if (xy == 0)

{

race3.setPosition(0, 0);

}

if (xy == 1)

{

race3.setPosition(-90, 0);

}

if (xy == 2)

{

race3.setPosition(-180, 0);

}

if (xy == 3)

{

race3.setPosition(95, 0);

}

}

if (c % 530 == 500){

r4th = 1;

int xy = rand() % 4;

if (xy == 0)

{

race4.setPosition(0, 0);

}

if (xy == 1)

{

race4.setPosition(-90, 0);

}

if (xy == 2)

{

race4.setPosition(-175, 0);

}

if (xy == 3)

{

race4.setPosition(-275, 0);

}

}

}

//////////////////////////88888888\*\*\*\*\*level3\*\*\*\*\*\*/////////////////////////

if (count > 100)

{

if (c % 400 == 399){

fis2 = 1;

int xx = rand() % 4;

if (xx == 0)

{

fish2.setPosition(0, 0);

}

if (xx == 1)

{

fish2.setPosition(100, 0);

}

if (xx == 2)

{

fish2.setPosition(190, 0);

}

if (xx == 3)

{

fish2.setPosition(270, 0);

}

}

if (c % 370 == 350)

{

fis3 = 1;

int xy = rand() % 4;

if (xy == 0)

{

fish3.setPosition(0, 0);

}

if (xy == 1)

{

fish3.setPosition(-90, 0);

}

if (xy == 2)

{

fish3.setPosition(90, 0);

}

if (xy == 3)

{

fish3.setPosition(180, 0);

}

}

if (c % 350 == 339)

{

trk = 1;

int xy = rand() % 4;

if (xy == 0)

{

truck1.setPosition(0, 0);

}

if (xy == 1)

{

truck1.setPosition(-90, 0);

}

if (xy == 2)

{

truck1.setPosition(-180, 0);

}

if (xy == 3)

{

truck1.setPosition(95, 0);

}

}

//bonusmove

if (c % 1500 == 1400)

{

int xy = rand() % 4;

if (xy == 0)

bonus.setPosition(0, 0);

if (xy == 1)

bonus.setPosition(-90, 0);

if (xy == 2)

bonus.setPosition(-180, 0);

if (xy == 3)

bonus.setPosition(95, 0);

}

if (c % 480 == 479)

{

ambu = 1;

int xy = rand() % 4;

ambulance.setPosition(0, 0);

if (xy == 0)

{

ambulance.setPosition(-270, 0);

}

if (xy == 1)

{

ambulance.setPosition(-190, 0);

}

if (xy == 2)

{

ambulance.setPosition(-90, 0);

}

if (xy == 3)

{

ambulance.setPosition(0, 0);

}

}

if (r1 % 400 == 390){

//r1st = 1;

int xy = rand() % 4;

if (xy == 0){

race1.setPosition(0, 0);

}

if (xy == 1){

race1.setPosition(100, 0);

}

if (xy == 2){

race1.setPosition(190, 0);

}

if (xy == 3)

{

race1.setPosition(270, 0);

}

}

if (c % 380 == 370){

int xy = rand() % 4;

r2nd = 1;

if (xy == 0){

race2.setPosition(0, 0);

}

if (xy == 1)

{

race2.setPosition(-90, 0);

}

if (xy == 2)

{

race2.setPosition(180, 0);

}

if (xy == 3)

{

race2.setPosition(95, 0);

}

}

if (c % 300 == 290){

r3rd = 1;

int xy = rand() % 4;

if (xy == 0)

{

race3.setPosition(0, 0);

}

if (xy == 1)

{

race3.setPosition(-90, 0);

}

if (xy == 2)

{

race3.setPosition(-180, 0);

}

if (xy == 3)

{

race3.setPosition(95, 0);

}

}

if (c % 330 == 320){

r4th = 1;

int xy = rand() % 4;

if (xy == 0)

{

race4.setPosition(0, 0);

}

if (xy == 1)

{

race4.setPosition(-90, 0);

}

if (xy == 2)

{

race4.setPosition(-175, 0);

}

if (xy == 3)

{

race4.setPosition(-275, 0);

}

}

}

//policebarricades move

if (policeb.getPosition().y>950)

{

//count++;

int xy = rand() % 2;

if (xy == 0)

policeb.setPosition(0, 0);

if (xy == 1)

policeb.setPosition(170, 0);

}

//people walker

lady.move(-2, .08);

walk++;

if (walk % 32 == 0){

walk++;

}

lady.setTextureRect(sf::IntRect(((walk % 32) / 8) \* 27, 0, 25, 35));

if (lady.getPosition().x <= -630)

{

int xy = rand() % 3;

if (xy == 0)

{

lady.setPosition(0, 0);

}

if (xy == 1)

{

lady.setPosition(0, 100);

}

if (xy == 2)

{

lady.setPosition(0, 200);

}

if (xy == 3)

{

lady.setPosition(0, 300);

}

}

if (lady.getGlobalBounds().intersects(race1.getGlobalBounds()))

race1.move(0, -2);

if (lady.getGlobalBounds().intersects(race2.getGlobalBounds()))

race2.move(0, -3);

if (lady.getGlobalBounds().intersects(race3.getGlobalBounds()))

race3.move(0, -4);

if (lady.getGlobalBounds().intersects(race4.getGlobalBounds()))

race4.move(0, -3);

if (lady.getGlobalBounds().intersects(fish2.getGlobalBounds()))

fish2.move(0, -3);

if (lady.getGlobalBounds().intersects(fish3.getGlobalBounds()))

fish3.move(0, -4);

if (lady.getGlobalBounds().intersects(truck1.getGlobalBounds()))

truck1.move(0, -2);

if (lady.getGlobalBounds().intersects(ambulance.getGlobalBounds()))

ambulance.move(0, -5);

baby.move(1.7, -.05);

walk++;

if (walk % 21 == 0){

walk++;

}

baby.setTextureRect(sf::IntRect(((walk % 21) / 7) \* 30, 0, 30, 35));

if (baby.getPosition().x >= 699)

{

int xy = rand() % 3;

if (xy == 0)

{

baby.setPosition(0, 0);

}

if (xy == 1)

{

baby.setPosition(0,-200 );

}

if (xy == 2)

{

baby.setPosition(0, -110);

}

}

if (baby.getGlobalBounds().intersects(race1.getGlobalBounds()))

race1.move(0, -2);

if (baby.getGlobalBounds().intersects(race2.getGlobalBounds()))

race2.move(0, -3);

if (baby.getGlobalBounds().intersects(race3.getGlobalBounds()))

race3.move(0, -4);

if (baby.getGlobalBounds().intersects(race4.getGlobalBounds()))

race4.move(0, -3);

if (baby.getGlobalBounds().intersects(fish2.getGlobalBounds()))

fish2.move(0, -3);

if (baby.getGlobalBounds().intersects(fish3.getGlobalBounds()))

fish3.move(0, -4);

if (baby.getGlobalBounds().intersects(truck1.getGlobalBounds()))

truck1.move(0, -2);

if (baby.getGlobalBounds().intersects(ambulance.getGlobalBounds()))

ambulance.move(0, -2);

if (car1.getGlobalBounds().intersects(bonus.getGlobalBounds()))

{

jump = jump + 2;

bonus.setPosition(-1000, 0);

}

}

if (game)

{

fish2.move(0, 3.5);

fish3.move(0, 4.5);

truck1.move(0, 3);

ambulance.move(0, 5.5);

if (c > 300)

race1.move(0, 2.7);

if (c > 250)

race2.move(0, 3.8);

if (c > 350)

race3.move(0, 4.5);

if (c > 400)

race4.move(0, 3.5);

bonus.move(0, 2);

policeb.move(0, 2);

border.move(0, 5);

border1.move(0, 5);

hf.move(0, 4);

h0.move(0, 4);

h1.move(0, 4);

//bonus not intersect with vehicles

if (bonus.getGlobalBounds().intersects(race1.getGlobalBounds()) || bonus.getGlobalBounds().intersects(truck1.getGlobalBounds()))

{

bonus.move(0, -5);

}

if (bonus.getGlobalBounds().intersects(fish2.getGlobalBounds()) || bonus.getGlobalBounds().intersects(race2.getGlobalBounds()) || bonus.getGlobalBounds().intersects(race4.getGlobalBounds()))

{

bonus.move(0, -3);

}

//vehicles not crush with each other

if (truck1.getGlobalBounds().intersects(race1.getGlobalBounds()))

{

truck1.move(0, -5);

}

if (truck1.getGlobalBounds().intersects(fish2.getGlobalBounds()) || truck1.getGlobalBounds().intersects(race2.getGlobalBounds()) || truck1.getGlobalBounds().intersects(race4.getGlobalBounds()))

{

truck1.move(0, -3);

}

if (race1.getGlobalBounds().intersects(fish2.getGlobalBounds()) || race1.getGlobalBounds().intersects(race2.getGlobalBounds()) || race1.getGlobalBounds().intersects(race4.getGlobalBounds()))

{

race1.move(0, -3);

}

//3 move vehicles not crush

if (fish2.getGlobalBounds().intersects(race2.getGlobalBounds()) || fish2.getGlobalBounds().intersects(race4.getGlobalBounds()))

{

fish2.move(0, -5);

}

if (fish2.getGlobalBounds().intersects(race3.getGlobalBounds()) || fish2.getGlobalBounds().intersects(fish3.getGlobalBounds()))

{

fish2.move(0, -2);

}

if (race2.getGlobalBounds().intersects(race4.getGlobalBounds()))

{

race2.move(0, -5);

}

if (race2.getGlobalBounds().intersects(race3.getGlobalBounds()) || race2.getGlobalBounds().intersects(fish3.getGlobalBounds()))

{

race2.move(0, -2);

}

if (race4.getGlobalBounds().intersects(race3.getGlobalBounds()) || race4.getGlobalBounds().intersects(fish3.getGlobalBounds()))

{

race4.move(0, -2);

}

//4 move vehicles not crush

if (fish3.getGlobalBounds().intersects(race3.getGlobalBounds()))

{

fish3.move(0, -5);

}

if (fish3.getGlobalBounds().intersects(ambulance.getGlobalBounds()))

{

fish3.move(0, -2);

}

if (race3.getGlobalBounds().intersects(ambulance.getGlobalBounds()))

{

race3.move(0, -2);

}

if (race1.getGlobalBounds().intersects(race2.getGlobalBounds()) || race1.getGlobalBounds().intersects(race3.getGlobalBounds()) || race1.getGlobalBounds().intersects(race4.getGlobalBounds()) || race1.getGlobalBounds().intersects(fish2.getGlobalBounds()) || race1.getGlobalBounds().intersects(fish3.getGlobalBounds()) || race1.getGlobalBounds().intersects(ambulance.getGlobalBounds()))

{

race1.move(0, -15);

}

if (j == 3)

{

game = false;

flag = 1;

zz = 2;

}

if (h1.getPosition().y == 1900)

h1.setPosition(0, 0);

if (bonus.getPosition().y == 1900)

bonus.setPosition(0, 0);

if (h0.getPosition().y == 1900)

h0.setPosition(0, 0);

if (hf.getPosition().y == 1900)

hf.setPosition(0, 0);

line.setPosition(400, 750);

side1.setPosition(495, 0);

side2.setPosition(864, 0);

h0.setOrigin(-408, 1180);

h1.setOrigin(-408, 70);

bonus.setOrigin(-690, 500);

hf.setOrigin(-408, 1830);

car1.setOrigin(-560, -600);

car2.setOrigin(-560, -600);

car3.setOrigin(-560, -600);

car4.setOrigin(-560, -600);

carfly.setOrigin(-560, -600);

ima.setPosition(510, 0);

grass.setPosition(410, 0);

ima1.setPosition(510, 320);

border.setOrigin(-586, 0);

border1.setOrigin(-776, 0);

fish2.setOrigin(-510, 70);

fish3.setOrigin(-600, 70);

truck1.setOrigin(-690, 70);

ambulance.setOrigin(-790, 70);

if (c > 200)

{

race1.setOrigin(-530, 100);

if (lal == 1)

{

r1 = 1;

race1.setPosition(0, 0);

lal = 2;

}

}

if (c > 250)

{

race2.setOrigin(-610, 100);

if (blu == 1)

{

r2 = 1;

race1.setPosition(0, 0);

blu = 2;

}

}

if (c > 150)

{

race3.setOrigin(-700, 100);

if (hol == 1)

{

r3 = 1;

race3.setPosition(0, 0);

hol = 2;

}

}

if (c > 100)

{

race4.setOrigin(-790, 100);

if (beg == 1)

{

r4 = 1;

race1.setPosition(0, 0);

beg = 2;

}

}

fire.setOrigin(-530, -575);

sidebackground.setPosition(0, 0);

sidebackground.setPosition(890, 0);

///////////9483\*\*\*\*\*\*\*\*\*counttttttttttttttttttttttttt/////////////////////////////////////

if (race1.getGlobalBounds().intersects(line.getGlobalBounds()))

{

count++;

race1.move(0,200);

}

if (race2.getGlobalBounds().intersects(line.getGlobalBounds()))

{

count++;

race2.move(0, 200);

}

if (race3.getGlobalBounds().intersects(line.getGlobalBounds()))

{

count++;

race3.move(0, 200);

}

if (race4.getGlobalBounds().intersects(line.getGlobalBounds()))

{

count++;

race4.move(0,200);

}

if (fish2.getGlobalBounds().intersects(line.getGlobalBounds()))

{

count++;

fish2.move(0,200);

}

if (fish3.getGlobalBounds().intersects(line.getGlobalBounds()))

{

count++;

fish3.move(0,200);

}

if (truck1.getGlobalBounds().intersects(line.getGlobalBounds()))

{

count++;

truck1.move(0,200);

}

if (ambulance.getGlobalBounds().intersects(line.getGlobalBounds()))

{

count++;

ambulance.move(0,200);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*level2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/////////////////////////

if (count > 50 && j == 2){

policeb.setOrigin(-530, 200);

if (pol == 1)

{

policeb.setPosition(0, 0);

pol = 2;

}

fish2.move(0, 1.5);

fish3.move(0, 1.5);

truck1.move(0, 1.5);

ambulance.move(0, 1.5);

race1.move(0, 1.5);

race2.move(0, 1.5);

race3.move(0, 1.5);

race4.move(0, 1.5);

bonus.move(0, 1.5);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*level3\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//////////////////////////////

if (count > 100 && j == 2){

baby.setOrigin(-450, -400);

if (bby == 1)

{

baby.setPosition(0, 0);

bby = 2;

}

lady.setOrigin(-900, -200);

if (ldy == 1)

{

lady.setPosition(0, 0);

ldy = 2;

}

fish2.move(0, 1.5);

fish3.move(0, 1.5);

truck1.move(0, 1.5);

ambulance.move(0, 1.5);

race1.move(0, 1.5);

race2.move(0, 1.5);

race3.move(0, 1.5);

race4.move(0, 1.5);

bonus.move(0, 1.5);

policeb.move(0, 1.5);

}

}

window.draw(ima);

window.draw(ima1);

window.draw(border);

window.draw(border1);

window.draw(policeb);

window.draw(fish2);

window.draw(fish3);

window.draw(truck1);

window.draw(ambulance);

window.draw(race1);

window.draw(race2);

window.draw(race3);

window.draw(race4);

window.draw(side1);

window.draw(side2);

window.draw(lady);

if (count>90)

window.draw(baby);

window.draw(sidebackground);

window.draw(sidebackground1);

window.draw(grass);

window.draw(hf);

window.draw(h1);

window.draw(h0);

window.draw(bonus);

window.draw(ambulance);

window.draw(line);

if (fly >= 24 && car % 40 <= 10)

window.draw(car1);

if (fly >= 24 && car % 40 >= 10 && car % 40 <= 20)

window.draw(car2);

if (fly >= 24 && car % 40 >= 20 && car % 40 <= 30)

window.draw(car3);

if (fly >= 24 && car % 40 >= 30 && car % 40 < 40)

window.draw(car4);

if (fly < 25)

window.draw(carfly);

sf::Text text, text1;

sf::Font font, font1;

font.loadFromFile("text.ttf");

font1.loadFromFile("text.ttf");

text.setFont(font);

text1.setFont(font1);

char ch[20], ch1[20];

//nt count=10;

sprintf(ch, "Your Score: %d\n", count);

sprintf(ch1, "jump: %d", jump);

text.setString(ch);

text.setCharacterSize(25);

text.setPosition(900, 100);

window.draw(text);

text1.setString(ch1);

text1.setCharacterSize(25);

text1.setPosition(900, 300);

// gari.play();

window.draw(text1);

if (!game&&c>70)

{

/\*if (event.type == sf::Event::KeyPressed){

if (sf::Keyboard::isKeyPressed(sf::Keyboard::RShift))

{

cout << "key is pressed" << endl;

goto high\_score;

}

}\*/

// crash.play();

// gari.stop();

musicc.stop();

while(cr == 1){

crash.play();

cout << "k=1" << endl;

cr=2;

}

c--;

fish2.move(0, 0);

fish3.move(0, 0);

truck1.move(0, 0);

ambulance.move(0, 0);

race1.move(0, 0);

race2.move(0, 0);

race3.move(0, 0);

race4.move(0, 0);

bonus.move(0, 0);

policeb.move(0, 0);

border.move(0, 0);

border1.move(0, 0);

hf.move(0, 0);

h0.move(0, 0);

h1.move(0, 0);

lady.move(2, -0.08);

baby.move(-1.7, 0.05);

window.draw(car2);

window.draw(fire);

Texture t30;

t30.loadFromFile("go.png");

Sprite gameover(t30);

gameover.setPosition(500, 240);

//window.draw(gameover);

if (sf::Keyboard::isKeyPressed(sf::Keyboard::Return))

{

musicc.stop();

HighScore h;

ppp = h.score();

if (count > ppp)

{

goto scoring;

}

else goto main\_menu;

}

else

{

musicc.stop();

sf::Font scre;

scre.loadFromFile("Lobster\_1.3.otf");

std::stringstream sss;

sss << "Your Score: " << count;

sf::Text pp;

pp.setCharacterSize(70);

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

pp.setFont(scre);

pp.setString(sss.str());

pp.setPosition(500,300);

/////////////////////////////////////////////

sf::Font fontt;

fontt.loadFromFile("Pacifico.ttf");

sf::Text textt;

textt.setFont(fontt);

textt.setString("Press ENTER to continue");

textt.setColor(sf::Color::Blue);

textt.setCharacterSize(40);

textt.setPosition(450, 550);

sf::Font fonttt;

fonttt.loadFromFile("Lobster\_1.3.otf");

sf::Text texttt;

texttt.setFont(fonttt);

texttt.setString("gameover!");

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

texttt.setCharacterSize(70);

texttt.setPosition(530, 200);

// window.clear();

// crash.stop();

// gari.stop();

//window.draw(gameover);

window.draw(pp);

window.draw(textt);

window.draw(texttt);

//window.display();

//count = 0;

//goto main\_menu;

}

}

window.display();

}

printf("%d", count);

}

scoring:

{

int xx = 1;

string name;

int aa;

bool flagg = false;

int ccc = count;

count = 0;

while (window.isOpen())

{

window.setFramerateLimit(300);

sf::Texture g\_over;

g\_over.loadFromFile("score.jpg");

sf::Sprite gover;

gover.setTexture(g\_over);

gover.setPosition(0, 0);

sf::Font fnt;

fnt.loadFromFile("Pacifico.ttf");

sf::Text txt;

txt.setFont(fnt);

txt.setString("Enter your name & Press Enter key to save");

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

txt.setCharacterSize(45);

txt.setPosition(900 / 2 - txt.getGlobalBounds().width / 2, 200);

sf::RectangleShape rct(sf::Vector2f(3, 40));

rct.setFillColor(sf::Color::Black);

rct.setPosition(900 / 2 - txt.getGlobalBounds().width / 2, 400);

rct.setScale(xx, xx);

if (xx == 0)

xx = 1;

else

xx = 0;

bool br = 1;

sf::Event event;

while (window.pollEvent(event))

{

switch (event.type)

{

case sf::Event::Closed:

window.close();

break;

case sf::Event::TextEntered:

if (event.text.unicode != 13 && event.text.unicode != 8)

{

name.push\_back((char)event.text.unicode);

}

else if (event.text.unicode == 8)

{

if (!name.empty())

name.pop\_back();

}

else

{

br = 0;

}

break;

break;

}

}

if (!br)

{

std::ofstream ofs("example.txt", std::ofstream::out);

ofs << name;

ofs << "\n";

ofs << ccc;

ofs.close();

}

sf::Font font;

font.loadFromFile("high.ttf");

sf::Text text;

text.setFont(font);

text.setString(name);

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

text.setCharacterSize(45);

text.setPosition(900 / 2 - text.getGlobalBounds().width / 2, 300);

sf::Font xt;

xt.loadFromFile("high.ttf");

sf::Text txtt;

txtt.setFont(xt);

txtt.setString("Press ESCAPE key for the Main Menu");

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

txtt.setStyle(sf::Text::Bold);

txtt.setCharacterSize(50);

txtt.setPosition(900 / 2 - txtt.getGlobalBounds().width / 2, 650);

if (sf::Keyboard::isKeyPressed)

{

if (event.key.code == sf::Keyboard::Escape)

goto main\_menu;

}

window.clear();

window.draw(gover);

window.draw(txt);

window.draw(txtt);

window.draw(text);

window.display();

}

}

exit\_game:

{

return EXIT\_SUCCESS;

}

}

P a g e | **21**



P a g e | **49**



**Special thanks to:**

1. Dr. Muhammad Asif Hossain Khan Associate Professor, CSEDU
2. Ms. Anna Fariha Lecturer, CSEDU

03. Hasnain Heickal Jami

Lecturer, CSEDU

1. Mesbah Tanvir

Student, CSEDU 20th batch

**References:**

1. <https://en.wikipedia.org/wiki/Borland_Graphics_Interface>
2. <http://www.stereosue.com/>
3. [http://www.thewallpapers.org](http://www.thewallpapers.org/)
4. <https://en.wikipedia.org/wiki/Borland_Graphics_Interface>
5. http://www.stackoverflow.com
6. Borland Graphics Interface for windows: <https://www.cs.colorado.edu/~main/bgi/doc/>
7. [http://www.programmingsimplified.com](http://www.programmingsimplified.com/)
8. [www.youtube.com](http://www.youtube.com/)
9. [http://maps.google.com](http://maps.google.com/)
10. C The Complete Reference (4th Edition) By Herbert Schildt
11. Programming in ANSI C By E.Balaguruswami

**The End**