Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет телекоммуникаций и информатики»

Факультет ИВТ

Кафедра прикладной математики и кибернетики

**Курсовая** **работа**

по дисциплине «Объектно-ориентированное программирование»

Тема: «Мультимедийная библиотека SFML»

Выполнил: студент гр. ИВ-921 Гилев М.А.

Проверил:

к.т.н., доцент Кафедры ПМиК

Ситняковская Е.И.

Новосибирск, 2021

**Содержание**

1. Постановка задачи
2. Введение
3. Выполнение

* Классы
* Функции

1. Заключение
2. Список использованной литературы
3. Листинг

**ЗАДАНИЕ**

Изучение модулей библиотеки SFML: System, Window, Graphics, Audio. Создание 2D игры с демонстрацией полученных навыков и знаний. Использование методов ООП при разработке приложения.

**ВВЕДЕНИЕ**

**SFML** (англ. *Simple and Fast Multimedia Library* — простая и быстрая мультимедийная библиотека) — свободная кроссплатформенная мультимедийная библиотека. Написана на С++, но доступна и для других языков языков программирования (С, C#, .Net, Java и пр.). SFML содержит ряд модулей для простого программирования игр и мультимедиа приложений.

В настоящее время доступны следующие модули:

System - управление временем и потоками, он является обязательным, так как все модули зависят от него.

Window - управление окнами и взаимодействием с пользователем.

Graphics - делает простым отображение графических примитивов и изображений.

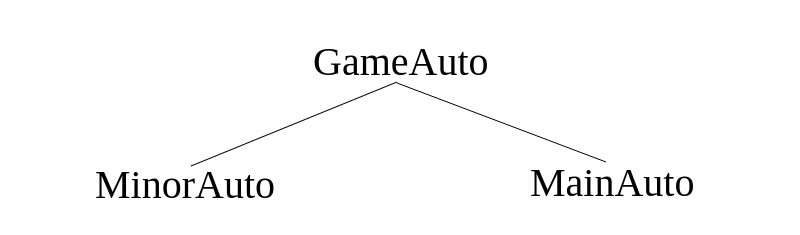
Audio - предоставляет интерфейс для управления звуком.

Network - для сетевых приложений.

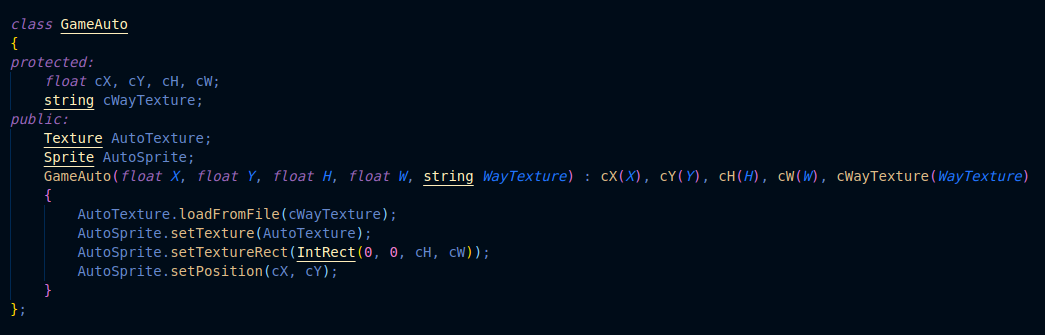
Игра была создана в жанре сайд-скроллер, для реализации которой нам понадобятся модули System, Window, Graphics и Audio.

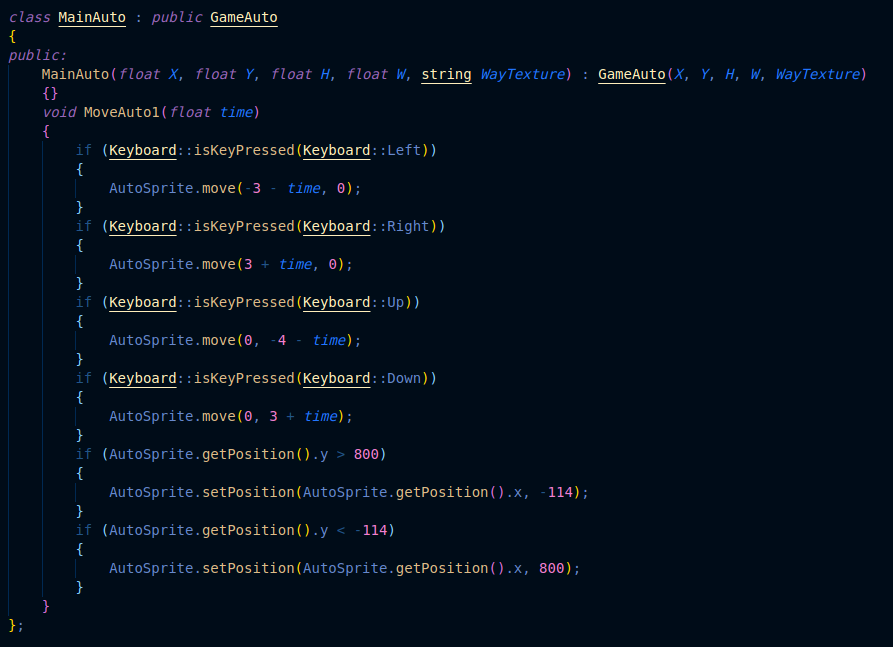
**Выполнение**

**1. Классы**

В игре были реализованы три класса со следующей иерархией:

С помощью методов **GameAuto** задается спрайт и расположение «машинки».



В **MainAuto** мы создаем управление и физику движения для «главной машинки». Именно

этой «машинкой» и предстоит управлять во время игры пользователю.

**MinorAuto** отвечает за остальные «машинки» на дороге, которые и будут создавать транспортный поток и, вместе с тем, препятствия игроку. Здесь описывается физика их движения по проезжей части.

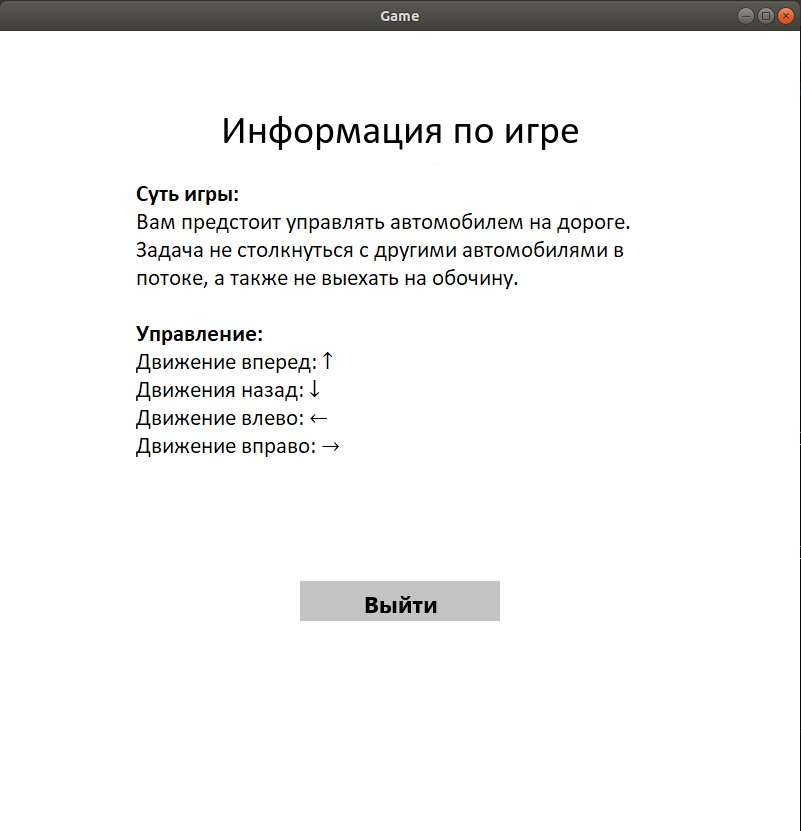
**2.Функции**

Для реализации проекта были написаны функции **road**, **menu**, **info** и **crash**.

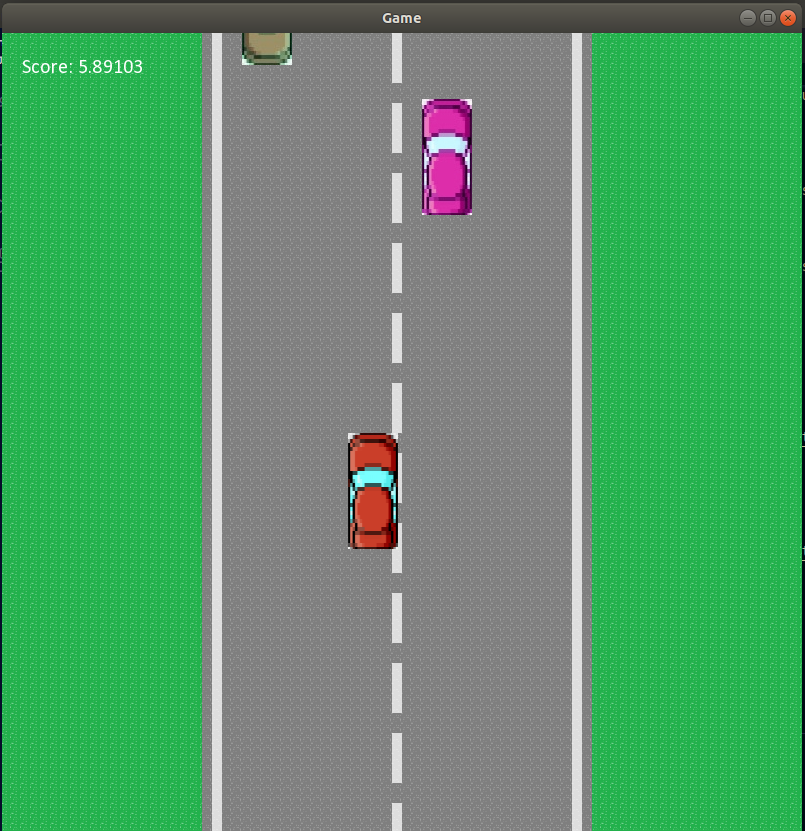
Функция **menu** отвечает за главное меню игры, отображение объектов в меню и взаимодействие с ними.



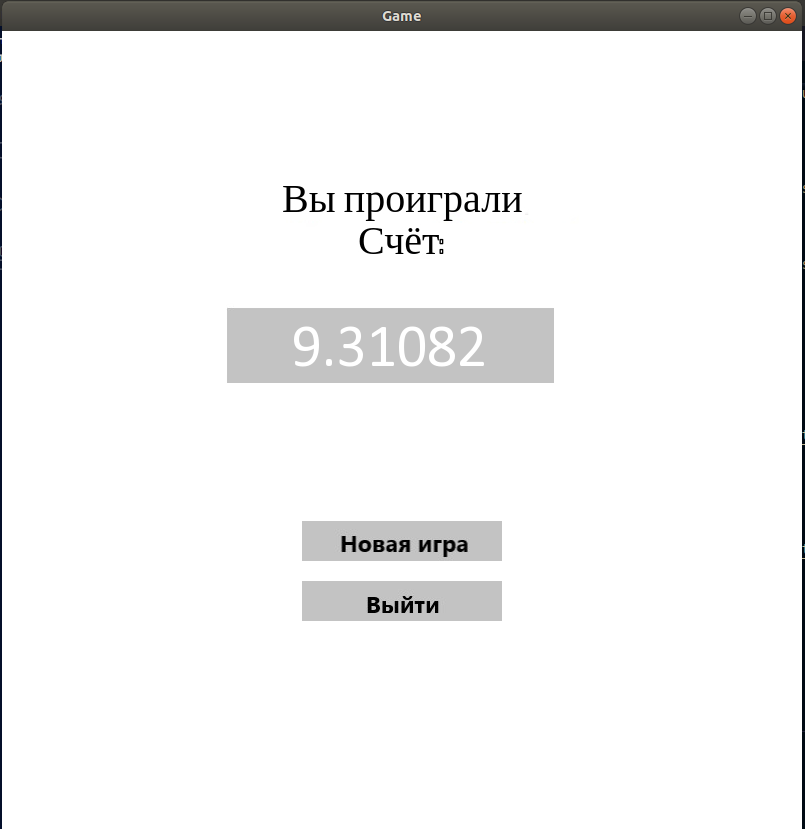
Функция **info** отвечает за пункт главного меню «Информация». Там содержится информация об игровом процессе и управлении. Сродни функции **menu**, содержит в себе информацию об отображении объектов и возможных взаимодействиях с ними.



Функция **road**, грубо говоря, является тем, что мы бы назвали игрой. Именно она отвечает за создание игрового поля, «машинок», так же за расположение этих машинок на поле, воспроизведение фоновой музыки и прочее. Если говорить короче, то данная функция отвечает за сам игровой процесс.



Функция **road** вызывает функцию **crash**, когда «машинка», находящаяся под управлением игрока, сталкивается с другой «машинкой» на дороге, или же когда она съезжает на обочину. Обобщая, функция **crash** отвечает за поражение в игре. Данная функция отображает меню проигрыша, воспроизводит мелодию окончания игры.



Также для создания игры потребовалось несколько изображений, для использования их в качестве текстур и спрайтов, а также несколько мелодий.

**ЗАКЛЮЧЕНИЕ**

В результате была разработана программа на языке С++, реализующая игру в жанре сайд-скроллер для одного игрока. При разработке программы были получены навыки работы с графической библиотекой SFML, а также получены и закреплены знания по объектно-ориентированному программированию.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

Искусство программирования на С++. Фленов М. Санкт - Петербург «БВХ-Петербург» 2006 г

[Электронный ресурс] https://www.sfml-dev.org/tutorials/2.5/

Дата обращения: 28.04.21

[Электронный ресурс] https://salikov.net/virtual-life/opisanie-zhanrov-2d-igr/

Дата обращения: 28.04.21

**ПРИЛОЖЕНИЯ**

main.cpp

#include <iostream>

#include <string>

#include <sstream>

#include <ctime>

#include <SFML/Graphics.hpp>

#include <SFML/Audio.hpp>

using namespace std;

using namespace sf;

const int roadh = 80;

const int roadw = 80;

void road(RenderWindow& window);

void menu(RenderWindow& window);

void info(RenderWindow& window);

void crash(RenderWindow& window, float time);

String roaddistr[roadh] =

{

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

"ggggggggggggggggggggawaaaaaaaaaaaaaaaaawaaaaaaaaaaaaaaaaawagggggggggggggggggggg",

};

class GameAuto

{

protected:

float cX, cY, cH, cW;

string cWayTexture;

public:

Texture AutoTexture;

Sprite AutoSprite;

GameAuto(float X, float Y, float H, float W, string WayTexture) : cX(X), cY(Y), cH(H), cW(W), cWayTexture(WayTexture)

{

AutoTexture.loadFromFile(cWayTexture);

AutoSprite.setTexture(AutoTexture);

AutoSprite.setTextureRect(IntRect(0, 0, cH, cW));

AutoSprite.setPosition(cX, cY);

}

};

class MainAuto : public GameAuto

{

public:

MainAuto(float X, float Y, float H, float W, string WayTexture) : GameAuto(X, Y, H, W, WayTexture)

{}

void MoveAuto1(float time)

{

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

{

AutoSprite.move(-3 - time, 0);

}

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

{

AutoSprite.move(3 + time, 0);

}

if (Keyboard::isKeyPressed(Keyboard::Up))

{

AutoSprite.move(0, -4 - time);

}

if (Keyboard::isKeyPressed(Keyboard::Down))

{

AutoSprite.move(0, 3 + time);

}

if (AutoSprite.getPosition().y > 800)

{

AutoSprite.setPosition(AutoSprite.getPosition().x, -114);

}

if (AutoSprite.getPosition().y < -114)

{

AutoSprite.setPosition(AutoSprite.getPosition().x, 800);

}

}

};

class MinorAuto : public GameAuto

{

public:

MinorAuto(float X, float Y, float H, float W, string WayTexture) : GameAuto(X, Y, H, W, WayTexture)

{}

void MoveAuto1(float time)

{

try

{

if (AutoSprite.getPosition().y < -100)

{

throw 0;

}

AutoSprite.move(0, -2 - time);

}

catch (int)

{

AutoSprite.setPosition(AutoSprite.getPosition().x, 800 + rand() % 2000);

}

}

void MoveAuto2(float time)

{

try

{

if (AutoSprite.getPosition().y > 800)

{

throw 0;

}

AutoSprite.move(0, 2 + time);

}

catch (int)

{

AutoSprite.setPosition(AutoSprite.getPosition().x, -114 - rand() % 2000);

}

}

};

void road(RenderWindow& window)

{

bool roadbool = 1;

while (roadbool)

{

Clock clock;

Music music;

music.openFromFile("./textures/The Weeknd - In Your Eyes.wav");

music.play();

Font font;

font.loadFromFile("./textures/calibri.ttf");

Text text("", font, 20);

text.setPosition(20, 20);

Texture roadtexture;

roadtexture.loadFromFile("./textures/road.png");

Sprite road;

road.setTexture(roadtexture);

MainAuto mainauto1(400, 400, 50, 116, "./textures/auto.png");

MinorAuto minorauto1(420, 800 + rand() % 2000, 50, 116, "./textures/auto11.jpg");

MinorAuto minorauto2(420, 800 + rand() % 2000, 50, 116, "./textures/auto12.jpg");

MinorAuto minorauto3(500, 800 + rand() % 2000, 50, 116, "./textures/auto13.jpg");

MinorAuto minorauto4(500, 800 + rand() % 2000, 50, 116, "./textures/auto11.jpg");

MinorAuto minorauto5(240, -114 - rand() % 2000, 50, 116, "./textures/auto21.jpg");

MinorAuto minorauto6(240, -114 - rand() % 2000, 50, 116, "./textures/auto22.jpg");

MinorAuto minorauto7(320, -114 - rand() % 2000, 50, 116, "./textures/auto23.jpg");

MinorAuto minorauto8(320, -114 - rand() % 2000, 50, 116, "./textures/auto21.jpg");

while (window.isOpen())

{

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

Event event;

while (window.pollEvent(event))

{

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

{

roadbool = 0;

clock.restart();

window.close();

}

}

ostringstream timeString;

timeString << time;

text.setString("Score: " + timeString.str());

mainauto1.MoveAuto1(time/100);

minorauto1.MoveAuto1(time/100);

minorauto2.MoveAuto1(time/100);

minorauto3.MoveAuto1(time/100);

minorauto4.MoveAuto1(time/100);

minorauto5.MoveAuto2(time/100);

minorauto6.MoveAuto2(time/100);

minorauto7.MoveAuto2(time/100);

minorauto8.MoveAuto2(time/100);

for (int i = 0; i < roadh; i++)

for (int j = 0; j < roadw; j++)

{

if (roaddistr[i][j] == 'a') road.setTextureRect(IntRect(0, 0, 10, 10));

if (roaddistr[i][j] == 'g') road.setTextureRect(IntRect(0, 10, 10, 10));

if (roaddistr[i][j] == 'w') road.setTextureRect(IntRect(0, 20, 10, 10));

road.setPosition(j \* 10, i \* 10);

window.draw(road);

}

window.draw(minorauto1.AutoSprite);

window.draw(minorauto2.AutoSprite);

window.draw(minorauto3.AutoSprite);

window.draw(minorauto4.AutoSprite);

window.draw(minorauto5.AutoSprite);

window.draw(minorauto6.AutoSprite);

window.draw(minorauto7.AutoSprite);

window.draw(minorauto8.AutoSprite);

window.draw(mainauto1.AutoSprite);

window.draw(text);

window.display();

if (mainauto1.AutoSprite.getGlobalBounds().intersects(minorauto1.AutoSprite.getGlobalBounds()) ||

mainauto1.AutoSprite.getGlobalBounds().intersects(minorauto2.AutoSprite.getGlobalBounds()) ||

mainauto1.AutoSprite.getGlobalBounds().intersects(minorauto3.AutoSprite.getGlobalBounds()) ||

mainauto1.AutoSprite.getGlobalBounds().intersects(minorauto4.AutoSprite.getGlobalBounds()) ||

mainauto1.AutoSprite.getGlobalBounds().intersects(minorauto5.AutoSprite.getGlobalBounds()) ||

mainauto1.AutoSprite.getGlobalBounds().intersects(minorauto6.AutoSprite.getGlobalBounds()) ||

mainauto1.AutoSprite.getGlobalBounds().intersects(minorauto7.AutoSprite.getGlobalBounds()) ||

mainauto1.AutoSprite.getGlobalBounds().intersects(minorauto8.AutoSprite.getGlobalBounds()) ||

mainauto1.AutoSprite.getPosition().x > 580 ||

mainauto1.AutoSprite.getPosition().x < 170)

{

music.stop();

crash(window, time);

roadbool = 0;

}

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

{

music.stop();

roadbool = 0;

menu(window);

}

}

}

}

void menu(RenderWindow& window)

{

bool menubool = 1;

while (menubool)

{

while (window.isOpen())

{

Event event;

while (window.pollEvent(event))

{

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

{

menubool = 0;

window.close();

}

}

Texture menuTexture1, menuTexture2, menuTexture3, menubackground;

menuTexture1.loadFromFile("./textures/NewGame.jpg");

menuTexture2.loadFromFile("./textures/Info.jpg");

menuTexture3.loadFromFile("./textures/Exit.jpg");

menubackground.loadFromFile("./textures/MenuBG.jpg");

Sprite menu1(menuTexture1), menu2(menuTexture2), menu3(menuTexture3), menubg(menubackground);

menubg.setPosition(0, 0);

menu1.setPosition(300, 300);

menu2.setPosition(300, 360);

menu3.setPosition(300, 420);

menu1.setColor(Color::White);

menu2.setColor(Color::White);

menu3.setColor(Color::White);

if (IntRect(300, 300, 200, 40).contains(Mouse::getPosition(window)))

{

menu1.setColor(Color::Blue);

}

if (IntRect(300, 360, 200, 40).contains(Mouse::getPosition(window)))

{

menu2.setColor(Color::Blue);

}

if (IntRect(300, 420, 200, 40).contains(Mouse::getPosition(window)))

{

menu3.setColor(Color::Blue);

}

window.draw(menubg);

window.draw(menu1);

window.draw(menu2);

window.draw(menu3);

window.display();

if ((Mouse::isButtonPressed(Mouse::Left)) && (IntRect(300, 300, 200, 40).contains(Mouse::getPosition(window))))

{

menubool = 0;

road(window);

}

if ((Mouse::isButtonPressed(Mouse::Left)) && (IntRect(300, 360, 200, 40).contains(Mouse::getPosition(window))))

{

menubool = 0;

info(window);

}

if ((Mouse::isButtonPressed(Mouse::Left)) && (IntRect(300, 420, 200, 40).contains(Mouse::getPosition(window))))

{

menubool = 0;

window.close();

}

}

}

}

void info(RenderWindow& window)

{

bool infobool = 1;

while (infobool)

{

while (window.isOpen())

{

Event event;

while (window.pollEvent(event))

{

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

{

infobool = 0;

window.close();

}

}

Texture menuTexture3, aboutTexture;

menuTexture3.loadFromFile("./textures/Exit.jpg");

aboutTexture.loadFromFile("./textures/Information.jpg");

Sprite menu3(menuTexture3), about(aboutTexture);

about.setPosition(0, 0);

menu3.setPosition(300, 550);

menu3.setColor(Color::White);

window.clear(Color(129, 181, 221));

if (IntRect(300, 550, 200, 40).contains(Mouse::getPosition(window)))

{

menu3.setColor(Color::Blue);

}

window.draw(about);

window.draw(menu3);

window.display();

if ((Mouse::isButtonPressed(Mouse::Left)) && (IntRect(300, 550, 200, 40).contains(Mouse::getPosition(window))))

{

infobool = 0;

menu(window);

}

}

}

}

void crash(RenderWindow& window, float time)

{

bool crashbool = 1;

while (crashbool)

{

Music music;

music.openFromFile("./textures/End1.wav");

music.play();

while (window.isOpen())

{

Event event;

while (window.pollEvent(event))

{

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

{

crashbool = 0;

window.close();

}

}

Font font;

font.loadFromFile("./textures/calibri.ttf");

Text text("", font, 60);

text.setPosition(290, 275);

ostringstream timeString;

timeString << time;

text.setString(timeString.str());

Texture menuTexture1, menuTexture3, aboutTexture;

menuTexture1.loadFromFile("./textures/NewGame.jpg");

menuTexture3.loadFromFile("./textures/Exit.jpg");

aboutTexture.loadFromFile("./textures/Crash.jpg");

Sprite menu1(menuTexture1), menu3(menuTexture3), about(aboutTexture);

about.setPosition(0, 0);

menu3.setPosition(300, 550);

menu1.setPosition(300, 490);

window.clear();

if (IntRect(300, 550, 200, 40).contains(Mouse::getPosition(window)))

{

menu3.setColor(Color::Blue);

}

if (IntRect(300, 490, 200, 40).contains(Mouse::getPosition(window)))

{

menu1.setColor(Color::Blue);

}

window.draw(about);

window.draw(menu3);

window.draw(menu1);

window.draw(text);

window.display();

if ((Mouse::isButtonPressed(Mouse::Left)) && (IntRect(300, 490, 200, 40).contains(Mouse::getPosition(window))))

{

music.stop();

crashbool = 0;

road(window);

}

if ((Mouse::isButtonPressed(Mouse::Left)) && (IntRect(300, 550, 200, 40).contains(Mouse::getPosition(window))))

{

music.stop();

crashbool = 0;

menu(window);

}

}

}

}

int main()

{

srand(time(0));

setlocale(0, "");

RenderWindow window(VideoMode(800, 800), "Game");

menu(window);

return 0;

}