**Отчет по командной работе Лебедевой Светланы**

**Графика**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата/Время** | **commit** | **комментарий** | **Цель создания** |
| 2018-12-17  01:04:11 | e99d6a5da7f6336a0cc5b7dae6a26eb293478354 | классы | Описание классов монстров, выделение общего класса и подклассов |
| 2018-12-25  01:15:49 | 3363acc3834724435ea673951d56d712081c21a9 | V5.0 | Изменение загрузки, спрайтов, изображений, текстур для главного персонажа, монстров, пуль |
| 2018-12-25  01:17:03 | 371b86f5e0136f4f3ca75b041a870dfae6f9cedc | V5.1 | Изменение загрузки, спрайтов, изображений, текстур для главного экрана, изменение шрифта, отрисовка фона, отрисовка кнопок |
| 2018-12-25  01:50:21 | 30c21c75164c6ffc873fd4d9b77be2847688fa2b | НОВЫЙ ПЕРСОНАЖ!!!! | Добавление файла нового персонажа |

**Фрагменты исходного кода**

**Файл allclass.hpp**

|  |
| --- |
| #ifndef allclass\_h |
|  | #define allclass\_h |
|  |  |
|  | using namespace sf; |
|  |  |
|  |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | class enemy { |
|  | public: |
|  | class enemy(); |
|  | bool enemyMove; |
|  | bool enemyLive; |
|  | int lives; |
|  | Image image; |
|  | Texture texture; |
|  | Sprite enemysprite; } |
|  |  |
|  |  |
|  |  |
|  | class enemy1 : public enemy { |
|  |  |
|  | enemy1 (float X, float Y){ |
|  | image.loadFromFile("bones.png"); |
|  | texture.loadFromImage(image); |
|  | enemysprite.setTexture(texture); |
|  | enemysprite.setTextureRect(IntRect(48, 128, 48, 64)); |
|  | enemysprite.setPosition(X, Y); |
|  | } |
|  | }; |
|  |  |
|  | class enemy2 : public enemy { |
|  |  |
|  |  |
|  | enemy2(float X, float Y){ |
|  | image.loadFromFile("ghost.png"); |
|  | texture.loadFromImage(image); |
|  | enemysprite.setTexture(texture); |
|  | enemysprite.setTextureRect(IntRect(64, 48, 64, 64)); |
|  | enemysprite.setPosition(X, Y); |
|  | } |
|  | }; |
|  |  |
|  | class enemy3 : public enemy { |
|  |  |
|  |  |
|  | enemy3(float X, float Y){ |
|  | image.loadFromFile("ghost.png"); |
|  | texture.loadFromImage(image); |
|  | enemysprite.setTexture(texture); |
|  | enemysprite.setTextureRect(IntRect(64\*16, 16, 128, 128)); |
|  | enemysprite.setPosition(X, Y); |
|  | } |
|  | }; |

**файл allclass.hpp**

|  |
| --- |
| #ifndef allclass\_h |
|  | #define allclass\_h |
|  |  |
|  | using namespace sf; //потом |
|  |  |
|  | class platform { |
|  | public: |
|  | Image image;//cфмл изображение |
|  | Texture texture;//сфмл текстура |
|  | Sprite platformsprite;//сфмл спрайт |
|  |  |
|  | platform(float X, float Y){ //Конструктор с параметрами(формальными) для класса Player. При создании объекта класса мы будем задавать имя файла, координату Х и У, ширину и высоту |
|  | image.loadFromFile("platform.jpg");//загружаем изображение из файла |
|  | texture.loadFromImage(image);//закидываем наше изображение в текстуру |
|  | platformsprite.setTexture(texture);//заливаем спрайт текстурой |
|  | platformsprite.setColor(sf::Color(128, 128, 255, 255));//создаем спрайт |
|  | platformsprite.setTextureRect(IntRect(0, 0, 256, 64));//выбор прямоугольной области ,которую берем от текстуры |
|  | platformsprite.setPosition(X,Y); |
|  | } |

|  |
| --- |
| class enemy { |
|  | public: |
|  | bool enemyMove; |
|  | bool enemyLive; |
|  | int lives; |
|  | Image image; |
|  | Texture texture; |
|  | Sprite enemysprite; |
|  |  |
|  | enemy(float X, float Y){ |
|  | image.loadFromFile("bones.png"); |
|  | texture.loadFromImage(image); |
|  | enemysprite.setTexture(texture); |
|  | enemysprite.setTextureRect(IntRect(48, 128, 48, 64)); |
|  | enemysprite.setPosition(X, Y); |
|  | } |
|  | }; |
|  |  |
|  | class enemy2 { |
|  | public: |
|  | bool enemyMove; |
|  | bool enemyLive; |
|  | int lives; |
|  | Image image; |
|  | Texture texture; |
|  | Sprite enemysprite; |
|  |  |
|  | enemy2(float X, float Y){ |
|  | image.loadFromFile("ghost.png"); |
|  | texture.loadFromImage(image); |
|  | enemysprite.setTexture(texture); |
|  | enemysprite.setTextureRect(IntRect(64, 48, 64, 64)); |
|  | enemysprite.setPosition(X, Y); |
|  | } |
|  | }; |
|  |  |
|  | class enemy3 { |
|  | public: |
|  | bool enemyLive; |
|  | int lives; |
|  | Image image; |
|  | Texture texture; |
|  | Sprite enemysprite; |
|  |  |
|  | enemy3(float X, float Y){ |
|  | image.loadFromFile("ghost.png"); |
|  | texture.loadFromImage(image); |
|  | enemysprite.setTexture(texture); |
|  | enemysprite.setTextureRect(IntRect(64\*16, 16, 128, 128)); |
|  | enemysprite.setPosition(X, Y); |
|  | } |
|  | }; |
|  |  |
|  | class bullet { |
|  | public: |
|  | int dirBullet; |
|  | bool bulletLive; |
|  | Image image; |
|  | Texture texture; |
|  | Sprite bulletsprite; |
|  |  |
|  | bullet(float X, float Y){ |
|  | image.loadFromFile("bullet.png"); |
|  | texture.loadFromImage(image); |
|  | bulletsprite.setTexture(texture); |
|  | bulletsprite.setTextureRect(IntRect(0, 0, 32, 14)); |
|  | bulletsprite.setPosition(X, Y); |
|  | } |
|  | }; |
|  |  |
|  |  |
|  | #endif /\* allclass\_h \*/ |

**Файл gameV5.cpp**

#include <SFML/Graphics.hpp>

#include <iostream>

#include "core.hpp"

#include <SFML/Audio.hpp>

#include "allclass.hpp"

void finishGame (RenderWindow & window){

Font font;//шрифт

font.loadFromFile("12838.ttf");//передаем нашему шрифту файл шрифта

Text menuButton("", font, 70);

menuButton.setString("YOU WON");

Texture menuBackground;//загрузка текстуры главного меню

menuBackground.loadFromFile("background.png");//выгрузка в текстуру из файла

Sprite menuBg(menuBackground);//спрайт

menuButton.setPosition(512, 260);

menuBg.setPosition(0, 0);

menuNum = 0;

menuButton.setColor(Color::White);

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

void EndGame (RenderWindow & window){

Font font;//шрифт

font.loadFromFile("12838.ttf");//передаем нашему шрифту файл шрифта

Text menuButton("", font, 70);

menuButton.setString("GAME OVER");

Texture menuBackground;

menuBackground.loadFromFile("background.png");

Sprite menuBg(menuBackground);

menuButton.setPosition(512, 160);

menuBg.setPosition(0, 0);

menuNum = 0;

menuButton.setColor(Color::White);

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

window.draw(menuBg);

window.draw(menuButton);

window.display();

void menu (RenderWindow & window) {

Font font;//шрифт

font.loadFromFile("12838.ttf");//передаем нашему шрифту файл шрифта

Text menu1("", font, 70);//создаем объект текст. закидываем в объект текст строку, шрифт, размер шрифта(в пикселях);//сам объект текст (не строка)

Text menu2("", font, 70);

Text menu3("", font, 70);

menu1.setString("START GAME");

menu2.setString("VIEW SCORE");

menu3.setString("EXIT GAME");

Texture menuBackground;

menuBackground.loadFromFile("background.png");

Sprite menuBg(menuBackground);

bool isMenu = 1; //Отвечает за главный цикл окна //Мы сейчас находимся в меню

int menuNum = 0; //Оператор отвечающий за кнопки цикла

menu1.setPosition(512, 260); // Задаем параматры расположения

menu2.setPosition(512, 360);

menu3.setPosition(512, 460);

menuBg.setPosition(0, 0);

////////////////////////////МЕНЮ///////////////////

while (isMenu)

menu1.setColor(Color::White); //Задаем цвет надписи

menu2.setColor(Color::White);

menu3.setColor(Color::White);

window.clear(Color(129, 181, 221)); //Обновление экрана

if (IntRect(512, 260, 300, 50).contains(Mouse::getPosition(window))) { menu1.setColor(Color::Red); menuNum = 1; } // Задаем прямоугольник текста и берем параметры указателя мыши

if (IntRect(512, 360, 300, 50).contains(Mouse::getPosition(window))) { menu2.setColor(Color::Red); menuNum = 2; }

if (IntRect(512, 460, 300, 50).contains(Mouse::getPosition(window))) { menu3.setColor(Color::Red); menuNum = 3; }

window.draw(menuBg); //Отрисовка фона

window.draw(menu1); //Отрисовка кнопок

window.draw(menu2);

window.draw(menu3);

window.display();

}

}

Texture backgroundtexture;

backgroundtexture.loadFromFile("background.png");

Sprite backgroundsprite;

backgroundsprite.setTexture(backgroundtexture);

backgroundsprite.setTextureRect(IntRect(0, 0 , 1280, 720));

backgroundsprite.setPosition(0, 0);

backgroundsprite.setColor(sf::Color(196, 0, 0, 255));

Texture hearttexture;

hearttexture.loadFromFile("heart.png");

Sprite heartsprite;

heartsprite.setTexture(hearttexture);

heartsprite.setTextureRect(IntRect(0, 0 , 1280, 720));

heartsprite.setPosition(1280-64, 0);

Texture poisontexture;

poisontexture.loadFromFile("poison.png");

Sprite poisonsprite;

poisonsprite.setTexture(poisontexture);

poisonsprite.setTextureRect(IntRect(0,0,1279,32));

poisonsprite.setPosition(0, 720-32);

poisonsprite.setColor(sf::Color(0, 196, 0, 196));

enemystrikes.bulletsprite.setColor(sf::Color(0,255,255,255)); //Обьект пули.Спрайт обьекта.Задан цвет

float CurrentFrame = 0; //хранит текущий кадр для анимации

**файл hero.cpp**

#include <SFML/Graphics.hpp>

#include <iostream>

#include "core.hpp"

#include <SFML/Audio.hpp>

#include "allclass.hpp"

using namespace sf;

using namespace std;

Image heroimage;

heroimage.loadFromFile("mainhero.png");

Texture herotexture;

herotexture.loadFromImage(heroimage);

Sprite herosprite;

herosprite.setTexture(herotexture);

herosprite.setTextureRect(IntRect(0, 0, 64, 128));

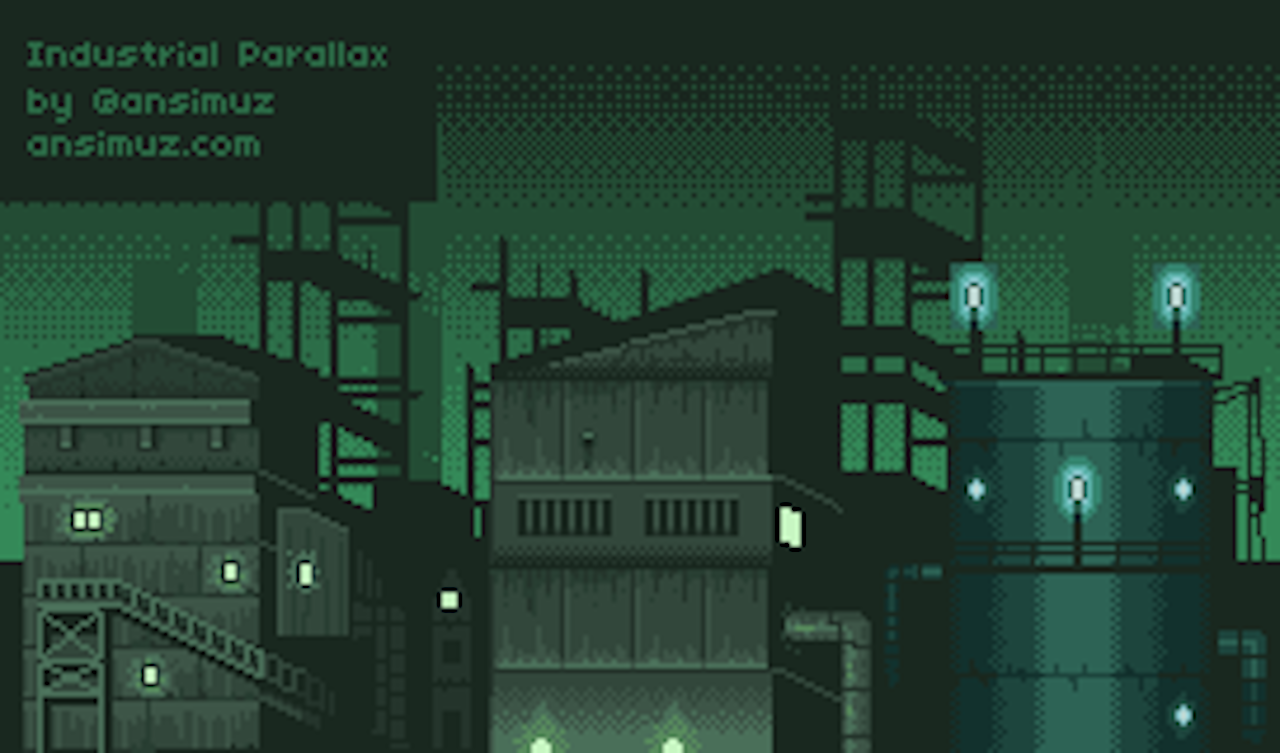
herosprite.setPosition(592, 720-128-64);

startGame(herosprite.getPosition().x, herosprite.getPosition().y); // Функция которые передают начальные координаты персонажа

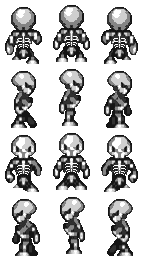
platform p1(512, 720 - 64);

platform p2(-256, 720-128);

**Используемые изображения:**

****

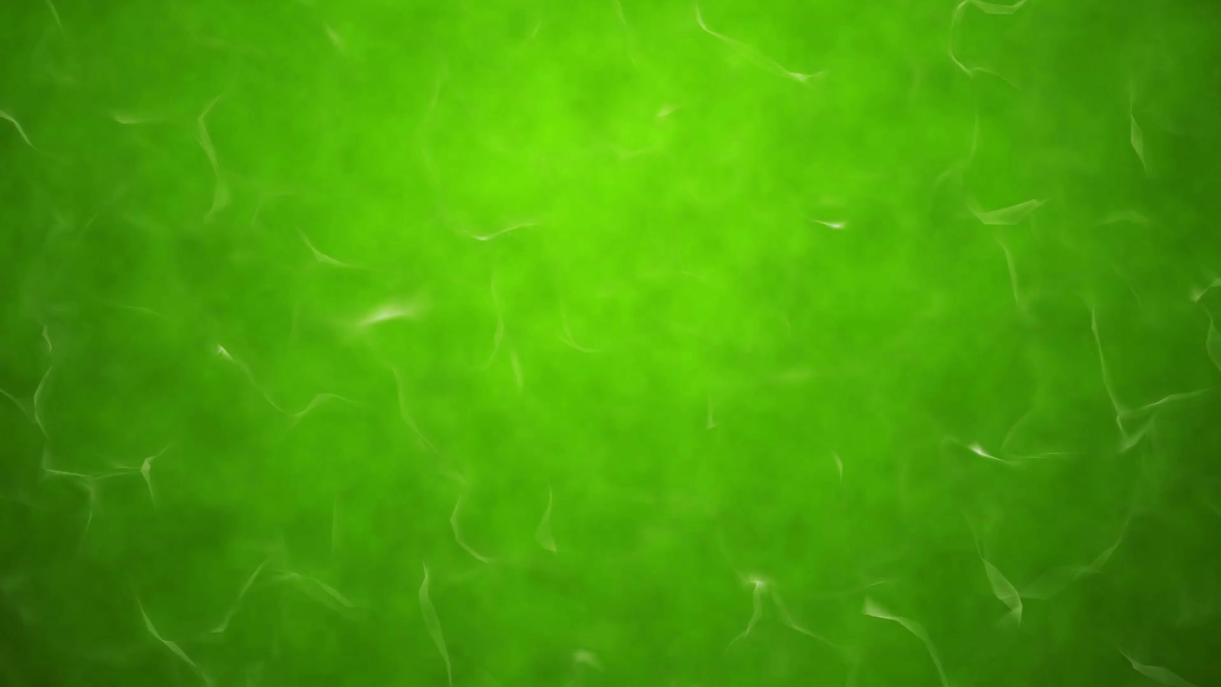
background.png



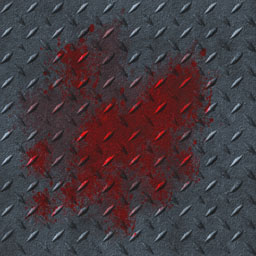
bones.png



mainhero.png



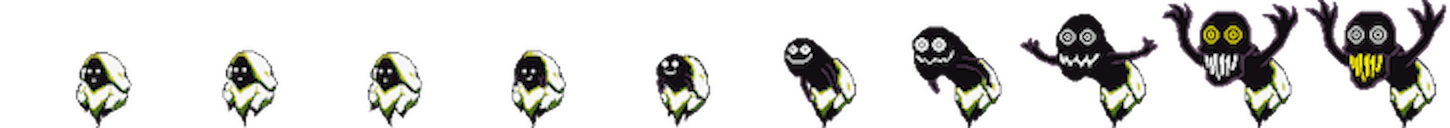
Position.png



Platform.png



Heart.png



Ghost.png