**Entity.h**

#ifndef \_\_ENTITY\_H\_\_

#define \_\_ENTITY\_H\_\_

#include "stdafx.h"

#include "map.h"

class Entity { ///Класс сущности//

protected:

float dx, dy, x, y, speed; //dx,dy изменение направления, x,y координаты, speed скорость изменения

int w, h; //w-ширина, h-длина нашей сущности

Texture texture;//текстура сущности

Map mp; //объект класса карта

float CurrentFrame;//текущий кадр

bool life; //переменная жизни

bool game;

public:

Sprite sprite;//спрайт сущности

Entity(Image &image, float X, float Y, int W, int H);//конструктор

FloatRect getRect() //получение прямуоголника (координаты и размеры)

{

FloatRect FR(x, y, w, h);

return FR;

}

bool getLife();

void setLife(bool l);

bool getGame();

void setGame(bool l);

virtual void update(float time) = 0; //метод обновления

};

#endif

**Player.h**

#ifndef \_\_PLAYER\_H\_\_

#define \_\_PLAYER\_H\_\_

#include "stdafx.h"

#include "Entity.h"

using namespace sf;

class Player :public Entity {

private:

int playerScore; //Очки игрока

int dir; //Направление

public:

Player(Image &image, float X, float Y, int W, int H); //конструктор класса image - изображение игрока, X Y координаты игрока, W H высота и ширина модели игрока

int getScore(); //геттер для playerScore

void setScore(int); //сеттер для playerScore

void control(); // метод управления игроком

void checkCollisionWithMap(float Dx, float Dy); //Проверка столкновение с препядствиями на карте

void update(float time); //Обновление состояния игрока

};

#endif

**Enemy.h**

#ifndef \_\_ENEMY\_H\_\_

#define \_\_ENEMY\_H\_\_

#include "stdafx.h"

#include "Entity.h"

class Enemy :public Entity{ //дочерний класс врага, призрака от класса сущности

private:

float yy, xx;

int direction;//Направление врага

public:

Enemy(Image &image, float X, float Y, int W, int H);

void checkCollisionWithMap(float Dx, float Dy); //метод проверки на столкновение

void update(float time);//метод обновления поведения врага

};

#endif;

**Interface.h**

#ifndef INTERFACE\_H

#define INTERFACE\_H

#include "Enemy.h"

#include "Player.h"

#include "Entity.h"

#include "Player.h"

#include "stdafx.h"

using namespace sf;

class Interface

{

private:

Player\* p;

Font font;//Шрифт для строк

Texture map;//текстура карты

Sprite s\_map;//спрайт карты

Clock clock, gameTimeClock; //таймеры

int gameTime; //игровое время

Map inter;

Event event;

Image map\_image, heroImage, enemy1, enemy2, enemy3;

std::list<Entity\*> enemy; //список врагов

std::list<Entity\*>::iterator it; //итератор для списка врагов

public:

Interface();

~Interface();

void interact(); //интерфейс программы

};

#endif

**Map.h**

#ifndef \_\_MAP\_H\_\_

#define \_\_MAP\_H\_\_

#pragma once

#include "stdafx.h"

//Класс карты

class Map{

public:

static std::string TileMap[];

};

#endif