**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Петрозаводский государственный университет»

Физико-технический институт

КУРСОВОЙ ПРОЕКТ

по дисциплине

”Технология программирования”

Выполнила:

Студентка группы 21318

А.А. Мелентьева

« » 2017г.

Научный руководитель:

канд. физ.-мат. наук, доцент

А.В.Бульба

« » 2017г.

# Оглавление

# Перечисление (дата, время) и цель сделанных коммитов 3

1. Фрагменты исходного кода, над которым я работала 3
2. Приложение1 5

**Перечисление (дата, время) и цель сделанных коммитов**

# Фрагменты исходного кода, над которыми я работала.

## Enemy.h

#include "\_CLASS\_H\_.h"

////////////////////////////КЛАСС ВРАГА////////////////////////

class Enemy :public Entity {

public:

int direction;//направление движения врага

Enemy(Image &image, float X, float Y, int W, int H, std::string Name);//:Entity(image, X, Y, W, H, Name);

void checkCollisionWithMap(float Dx, float Dy);//ф-ция проверки столкновений с картой

void update(float time);

};//класс Enemy закрыт

## Enemy.cpp

#include "\_CLASS\_H\_.h"

#include "map.h"

#include "Enemy.h"

Enemy::Enemy(Image &image, float X, float Y, int W, int H, std::string Name):Entity(image, X, Y, W, H, Name)

{

if (name == "EasyEnemy"){

//Задаем спрайту один прямоугольник для

//вывода одного игрока. IntRect – для приведения типов

sprite.setTextureRect(IntRect(0, 0, w, h));

direction = rand() % (5); //Направление движения врага задаём случайным образом

//через генератор случайных чисел

speed = 0.05;//даем скорость.этот объект всегда двигается

dx = speed;

}

}

void Enemy::checkCollisionWithMap(float Dx, float Dy)//ф-ция проверки столкновений с картой

{

for (int i = y / 50; i < (y + h) / 50; i++)//проходимся по элементам карты

for (int j = x / 50; j<(x + w) / 50; j++)

{

if (TileMap[i][j] == '0')//если элемент - тайлик земли

{

if (Dy > 0) {

y = i \* 50 - h; dy = -0.1;

direction = rand() % (3); //Направление движения врага

}//по Y

if (Dy < 0) {

y = i \* 50 + 50; dy = 0.1;

direction = rand() % (3);//Направление движения врага

}//столкновение с верхними краями

if (Dx > 0) {

x = j \* 50 - w; dx = -0.1;

direction = rand() % (3);//Направление движения врага

}//с правым краем карты

if (Dx < 0) {

x = j \* 50 + 50; dx = 0.1;

direction = rand() % (3); //Направление движения врага

}// с левым краем карты

}

}

}

void Enemy::update(float time)

{

if (name == "EasyEnemy"){//для персонажа с таким именем логика будет такой

if (life) {//проверяем, жив ли герой

switch (direction)//делаются различные действия в зависимости от состояния

{

case 0:{//состояние идти вправо

dx = speed;

CurrentFrame += 0.000001\*time;

if (CurrentFrame > 3) CurrentFrame -= 3;

sprite.setTextureRect(IntRect(58 \* int(CurrentFrame), 54, 58, 54));

break;

}

case 1:{//состояние идти влево

dx = -speed;

CurrentFrame += 0.000001\*time;

if (CurrentFrame > 3) CurrentFrame -= 3;

sprite.setTextureRect(IntRect(58 \* int(CurrentFrame), 0, 58, 54));

break;

}

case 2:{//идти вверх

dy = -speed;

CurrentFrame += 0.000001\*time;

if (CurrentFrame > 3) CurrentFrame -= 3;

sprite.setTextureRect(IntRect(0 \* int(CurrentFrame), 0, 58, 54));

break;

}

case 3:{//идти вниз

dy = speed;

CurrentFrame += 0.000001\*time;

if (CurrentFrame > 3) CurrentFrame -= 3;

sprite.setTextureRect(IntRect(0 \* int(CurrentFrame),54, 58, 54));

break;

}

}

x += dx\*time; //движение по “X”

checkCollisionWithMap(dx, 0);//обрабатываем столкновение по Х

y += dy\*time; //движение по “Y”

checkCollisionWithMap(0, dy);//обрабатываем столкновение по Y

sprite.setPosition(x, y); //спрайт в позиции (x, y).

if (health <= 0){ life = false; }//если жизней меньше 0, либо равно 0, то умираем

}

//if (playerScoreSting = 50)

}

};

# Приложение 1

Ссылка на репозиторий: https://github.com/LISSANEN/Frog.git