#include <SFML/Graphics.hpp>

#include <iostream>

#include <sstream>

#include "mission.h"

#include "map.h"

#include "view.h"

#pragma warning(suppress : 4996)

using namespace sf;

class Player { // класс Игрока

/\*создадим приватные переменные координат персонажа\*/

private: float x, y;

public:

float w, h, dx, dy, speed = 0; //координаты игрока х и у, высота ширина, ускорение (по х и по у), сама скорость

int dir, playerScore,health; //направление (direction) движения игрока

bool life,onGround;//добавили переменную состояния нахождения на земле

String File; //файл с расширением

Image image;//сфмл изображение

Texture texture;//сфмл текстура

Sprite sprite;//сфмл спрайт

enum { left, right, up, down, jump, stay } state;//добавляем тип перечисления - состояние объекта//переменная перечисляемого типа

Player(String F, float X, float Y, float W, float H) {

dx = 0; dy = 0; speed = 0; dir = 0; playerScore = 0; health = 100;

life = true; onGround = false;

File = F;//имя файла+расширение

w = W; h = H;//высота и ширина

image.loadFromFile("images/" + File);//запихиваем в image наше изображение вместо File мы передадим то, что пропишем при создании объекта. В нашем случае "hero.png" и получится запись идентичная image.loadFromFile("images/hero/png");

//image.createMaskFromColor(Color(41, 33, 59));//убираем ненужный темно-синий цвет, эта тень мне показалась не красивой.

texture.loadFromImage(image);//закидываем наше изображение в текстуру

sprite.setTexture(texture);//заливаем спрайт текстурой

x = X; y = Y;//координата появления спрайта

sprite.setTextureRect(IntRect(40, 244, w, h)); //Задаем спрайту один прямоугольник для вывода одного льва, а не кучи львов сразу. IntRect - приведение типов

sprite.setOrigin(w / 2, h / 2);//задает середину спрайта. Когда он коснется земли, он будет приподнят из этой середины

}

void update(float time) //???????? "?????????" ??????? ??????. update - ??????????. ????????? ? ???? ????? SFML , ?????????? ???? ???????? ??????????, ????? ????????? ????????

{

control();//функция управления персонажем

switch (state) // тут делаются различные действия в зависимости от состояния

{

case right: dx = speed; break;//состояние идти вправо

case left: dx = -speed; break;//состояние идти влево

case up: break;//будет состояние поднятия наверх (например по лестнице)

case down: dx = 0; break;//будет состояние во время спуска персонажа (например по лестнице)

case jump: break;//здесь может быть вызов анимации

case stay: break;//и здесь тоже

}

x += dx \* time;//?? ???????? ?? ???????? ?????. ???? ????????? ?? ????? ???????? ???????? ????????? ? ??? ????????? ????????

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

y += dy \* time;//?????????? ?? ??????

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

speed = 0;//???????? ????????, ????? ???????? ???????????.

sprite.setPosition(x+w/2, y+h/2); //??????? ?????? ? ??????? x y , ??????????. ?????????? ??????? ? ???? ???????, ????? ?? ??? ?????? ????? ?? ?????.

if (health <= 0) { life = false; speed = 0; }//если жизней меньше либо равно 0, то умираем и исключаем движение игрока после смерти

dy = dy + 0.0015 \* time; //делаем притяжение к земле

if (life) { setPlayerCoordinateForView(x, y); }

}

float getplayercoordinateX() {// этим методом будем забирать координату Х

return x;

}

float getplayercoordinateY() {

return y;

}

void control()

{

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

state = left;

speed = 0.2;//dir =1 - ??????????? ?????, speed =0.1 - ???????? ????????. ???????? - ????? ?? ??? ????? ?? ?? ??? ?? ???????? ? ????? ?? ?????????? ?????? ???

//currentFrame += 0.005 \* time;

//if (currentFrame > 6) currentFrame -= 6;

//p.sprite.setTextureRect(IntRect(40 \* int(currentFrame) + 40, 244, -40, 50)); //????? ?????? p ?????? player ?????? ??????, ????? ???????? (????????? ???????? ?????)

}

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

state =right;

speed = 0.2;//??????????? ??????, ?? ????

//currentFrame += 0.005 \* time;

//if (currentFrame > 6) currentFrame -= 6;

//p.sprite.setTextureRect(IntRect(40 \* int(currentFrame), 244, 40, 50)); //????? ?????? p ?????? player ?????? ??????, ????? ???????? (????????? ???????? ?????)

}

if ((Keyboard::isKeyPressed(Keyboard::Up)&&(onGround))) {

state = jump;

dy = -0.5;//высота прыжка

onGround = false;//??????????? ????, ?? ????

//currentFrame += 0.005 \* time;

////if (currentFrame > 3) currentFrame -= 3;

////p.sprite.setTextureRect(IntRect(96 \* int(currentFrame), 288, 96, 96)); //????? ?????? p ?????? player ?????? ??????, ????? ???????? (????????? ???????? ?????)

}

if ((Keyboard::isKeyPressed(Keyboard::Down))) { //???? ?????? ??????? ??????? ????? ??? ???? ????? ?

state=down;

speed = 0.1;//??????????? ?????, ?? ????

//currentFrame += 0.005 \* time; //?????? ??? ??????????? ?? "??????". ?????????? ??????? ?? ???? ???????? ???????????? ??????? ? ????????. ??????? 0.005 ????? ???????? ???????? ????????

//if (currentFrame > 3) currentFrame -= 3; //?????????? ?? ?????? ? ??????? ?? ?????? ????????????. ???? ?????? ? ???????? ????? - ???????????? ?????.

//p.sprite.setTextureRect(IntRect(96 \* int(currentFrame), 0, 96, 96)); //?????????? ?? ??????????? ?. ?????????? 96,96\*2,96\*3 ? ????? 96

}

}

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

{

for (int i = y / 32; i < (y + h) / 32; i++)//проходимся по элементам карты

for (int j = x / 32; j < (x + w) / 32; j++)

{

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

{

if (Dy > 0) { y = i \* 32 - h; dy = 0; onGround = true; }//по Y вниз=>идем в пол(стоим на месте) или падаем. В этот момент надо вытолкнуть персонажа и поставить его на землю, при этом говорим что мы на земле тем самым снова можем прыгать

if (Dy < 0) { y = i \* 32 + 32; dy = 0; }//столкновение с верхними краями карты(может и не пригодиться)

if (Dx > 0) { x = j \* 32 - w; }//с правым краем карты

if (Dx < 0) { x = j \* 32 + 32; }// с левым краем карты

}

else { onGround = false; }

if (TileMap[i][j] == 's') { //если символ равен 's' (камень)

playerScore++;//какое то действие... например телепортация героя

TileMap[i][j] = ' ';//убираем камень, типа взяли бонус. можем и не убирать, кстати.

}

if (TileMap[i][j] == 'f') {

health -= 40;

TileMap[i][j] = ' ';

}

if (TileMap[i][j] == 'h') {

health += 20;

TileMap[i][j] = ' ';

}

}

}

};

int main()

{

RenderWindow window(VideoMode(400, 500), "Test!");//??????? ???? 200x200

///////////////////РАБОТА С ТЕКСТОМ////////////////

Font font;//шрифт

font.loadFromFile("CyrilicOld.ttf");//передаеим нашему шрифту файл

Text text1("", font, 20);

Text text2("", font, 20);

Text textM("", font, 20);//текст миссия

//textM.setColor(Color::White);//все еще не работает. Побробую потом другой шрифт

//text1.setStyle(sf::Text::Bold | sf::Text::Underlined);//жирный подчеркнутый шрифт

#pragma warning(suppress : 4996) text1.setColor(Color::Black);//не работает замена цвета!!!

#pragma warning(suppress : 4996) text2.setColor(Color::Black);

view.reset(sf::FloatRect(0, 0, 640, 480));//размер "вида" камеры при создании объекта вида камеры. (потом можем менять как хотим) Что то типа инициализации.

Image map\_image;//?????? ??????????? ??? ?????

map\_image.loadFromFile("images/map.png");//????????? ???? ??? ?????

Texture map;//???????? ?????

map.loadFromImage(map\_image);//???????? ???????? ?????????

Sprite s\_map;//??????? ?????? ??? ?????

s\_map.setTexture(map);//???????? ???????? ????????

Player p("fang.png", 100, 100, 40.0, 50.0);

//p.setPosition(50,100);

float currentFrame = 0; // ??????????, ? ??????? ???????? ????? ???????? ?????

bool showMissionText = true;//логическая переменная, отвечающая за появление текста миссии на экране

///////////////////Добавим изображение свитка!/////////////////////

Image quest\_image;

quest\_image.loadFromFile("images/missionbg.jpg");

//quest\_image.createMaskFromColor(Color(0, 0, 0));

Texture quest\_texture;

quest\_texture.loadFromImage(quest\_image);

Sprite s\_quest;

s\_quest.setTexture(quest\_texture);

s\_quest.setTextureRect(IntRect(0, 0, 340, 510)); //приведение типов, размеры картинки исходные

s\_quest.setScale(0.6f, 0.6f);//чуть уменьшили картинку, => размер стал меньше

Clock clock;//?????????? ??????? (?????? ? ??????????)

while (window.isOpen())

{

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

clock.restart();//????????????? ????. (???????? ????? ?????)

time = time / 800; //?????????? ???????? ????

Event event; //???????? ??????? ???????? ????

while (window.pollEvent(event))

{

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

window.close();

if (event.type == Event::KeyPressed)//событие нажатия клавиши

if ((event.key.code == Keyboard::Tab)) {//если зажата клавиша Tab

switch (showMissionText) {//переключатель, реагирующий на логическую переменную showMissionText

case true: {

std::ostringstream playerHealthString;//строка здоровья игрока

playerHealthString << p.health; //заносим в строку здоровье

std::ostringstream task;//строка текста миссии

task << getTextMission(getCurrentMission(p.getplayercoordinateX()));//вызывается функция getTextMission (она возвращает текст миссии), которая принимает в качестве аргумента функцию getCurrentMission(возвращающую номер миссии), а уже эта ф-ция принимает в качестве аргумента функцию p.getplayercoordinateX() (эта ф-ция возвращает Икс координату игрока)

textM.setString("Здоровье: " + playerHealthString.str() + "\n" + task.str());//задаем

showMissionText = false;//эта строка позволяет убрать все что мы вывели на экране

break;//выходим , чтобы не выполнить условие "false" (которое ниже)

}

case false: {

textM.setString("");//если не нажата клавиша таб, то весь этот текст пустой

showMissionText = true;// а эта строка позволяет снова нажать клавишу таб и получить вывод на экран

break;

}

}

}

}

setPlayerCoordinateForView(p.getplayercoordinateX(), p.getplayercoordinateY());

p.update(time);

//viewmap(time);//двигать камеру буковками

//changeview();

window.setView(view);//"оживляем" камеру в окне sfm

window.clear(); //???????? ?????

/////////////////////////////Рисуем карту/////////////////////

for (int i = 0; i < HEIGHT\_MAP; i++)

for (int j = 0; j < WIDTH\_MAP; j++)

{

if (TileMap[i][j] == ' ') s\_map.setTextureRect(IntRect(0, 0, 32, 32)); //если встретили символ пробел, то рисуем 1й квадратик

if (TileMap[i][j] == 's') s\_map.setTextureRect(IntRect(32, 0, 32, 32));//если встретили символ s, то рисуем 2й квадратик

if ((TileMap[i][j] == '0')) s\_map.setTextureRect(IntRect(64, 0, 32, 32));//если встретили символ 0, то рисуем 3й квадратик

if ((TileMap[i][j] == 'f')) s\_map.setTextureRect(IntRect(96, 0, 32, 32));//добавили цветок

if ((TileMap[i][j] == 'h')) s\_map.setTextureRect(IntRect(128, 0, 32, 32));//и сердечко

s\_map.setPosition(j \* 32, i \* 32);//по сути раскидывает квадратики, превращая в карту. то есть задает каждому из них позицию. если убрать, то вся карта нарисуется в одном квадрате 32\*32 и мы увидим один квадрат

window.draw(s\_map);//рисуем квадратики на экран

}

///////////РАБОТА С ТЕКСТОМ//////////////

std::ostringstream playerScoreString; // объявили переменную

playerScoreString << p.playerScore; //занесли в нее число очков, то есть формируем строку

text1.setString("Собрано камней:" + playerScoreString.str());

text1.setPosition(view.getCenter().x-195, view.getCenter().y-200);//задаем позицию текста, центр камеры///задаем позицию текста, отступая от центра камеры

std::ostringstream playerHealthString; // объявили переменную

playerHealthString << p.health; //занесли в нее число здоровья, то есть формируем строку

text2.setString("Здоровье:" + playerHealthString.str());//задаем строку тексту и вызываем сформированную выше строку методом .str()

text2.setPosition(view.getCenter().x - 165, view.getCenter().y - 230);//задаем позицию текста, отступая от центра камеры

window.draw(text1);//рисую этот текст

window.draw(text2);//рисую этот текст

if (!showMissionText)

{

window.draw(s\_quest); window.draw(textM);

textM.setPosition(view.getCenter().x + 125, view.getCenter().y - 130);//позиция всего этого текстового блока

s\_quest.setPosition(view.getCenter().x + 115, view.getCenter().y - 130);//позиция фона для блока

}//рисуем спрайт свитка (фон для текста миссии). а затем и текст. все это завязано на логическую переменную, которая меняет свое состояние от нажатия клавиши ТАБ

window.draw(p.sprite);//???????? ??????

window.display();//?????? ??? ?? ???????

}

return 0;

}