МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Новосибирский национальный исследовательский государственный университет»

(Новосибирский государственный университет, НГУ)

**Структурное подразделение Новосибирского государственного университета -**

**Высший колледж информатики Университета**

Кафедра информатики

**КУРСОВОЙ ПРОЕКТ**

Прикладная информатика

Тема: Разработка компьютерной игры «ВКИ Dungeon»

Выполнил: Киселев Д.Д.

Студентк группы: 007б2*.*

Преподаватель: Кошкарева С.В.

Работа допущена к защите:

Оценка:

Подпись: */ /*

ОПИСАНИЕ ИГРЫ

Сюжет

Искатель приключений исследует темные лабиринты Подземелья ВКИ в поисках сокровищ. Иногда он встречает врагов. Сможет ли он победить всех и дойти до конца?

Правила игры

Изначально игрок не видит всего игрового уровня (лабиринта), чтобы не знать заранее, где выход. Он ходит по коридорам в поисках игровых очков и выхода на следующий уровень. Когда ему встречаются враги, он может сразиться с ними. Для победы в схватке ему нужно ввести комбинаю букв, которую случайным образом выберет игра. Если он введёт кобинацию неверно, враг отнимет здоровье у игрока. Здоровье восстанавливается при переходе между уровнями.

Управление

Управление в игре реализовано при помощи клавиатуры. Для передвижения персонажа используются стрелки. Для битв с врагами все буквы клавиатуры.

Нажатие клавиши Escape в модуле меню равносильно нажатию кнопки Exit. Нажатие кнопки Escape в модуле игры вернет игрока в меню. Нажатие клавиши Enter в меню запустит модуль игры.

КРАТКОЕ ОПИСАНИЕ ПРОЦЕДУР И ФУНКЦИЙ

Модуль ВКИ Dungeon запускает меню

Модуль menu

Функция menu() запускает меню

Функция help() показывает на экране краткую информацию об игре и управлении

Модуль game

Функция main() запускает игру

Функция draw\_map() создает уровень

Функция control(I, j, ip, jp) отвечает за управление

Функция moving(I, j, ip, jp, lvl, lives, spell\_change, spell\_result) отслеживает передвижение игрока

Функция pve(lives, sp\_in, spell\_change, spell\_result) отвечает за бой с врагом

ЛИСТИНГ ПРОГРАММНОГО КОДА

level\_counter = 0 #level counter счетчик уровней

counter = level\_counter

score = 0

def main():

global level\_counter

global counter

global score

import menu

import random

import pygame as pg

import pygame

pg.init()

from pygame import font

pg.time.set\_timer(pg.USEREVENT, 1000)

game\_name = 'ВКИ Dungeon'

pg.display.set\_caption(game\_name)

levels = ['level\_1.txt', 'level\_2.txt', 'level\_3.txt']

level\_quality = len(levels)

timer = 10

cs = 40

#создание матрицы для карты уровня

cells = 16

lvl = [0]\*cells

for i in range(cells):

lvl[i] = [0]\*cells

lvl\_file = open(levels[level\_counter], 'r')

for i in range(cells):

strings = lvl\_file.readline()

clean = strings.split()

for j in range(cells):

lvl[i][j] = int(clean[j])

## lvl\_file.close()

darkness = [1]\*cells #создание препятсвия для глаз поверх лабиринта, чтобы игрок не мог сразу узнать, куда ему идти, а исследовал уровень

for i in range(cells):

darkness[i] = [1]\*cells

player = pg.image.load('player.png')

wall = pg.image.load('wall.png')

floor = pg.image.load('floor.png')

exit\_lvl = pg.image.load('door\_ex.png')

money = pg.image.load('money.png')

dark = pg.image.load('dark.png')

enemy = pg.image.load('enemy.png')

game\_bg = pg.image.load('game\_bg.jpg')

live = pg.image.load('live.png')

screenX = 1200

screenS = cs\*cells

screen = pg.display.set\_mode((screenX, screenS))

screen.blit(game\_bg, [0, 0])

text\_color = [240, 250, 255]

txt\_bg\_color = [45, 40, 43]

font = pg.font.Font('progresspixel.ttf', 20)

pg.draw.rect(screen, txt\_bg\_color, [screenS+cs, cs, screenX - screenS - 2\*cs, screenS - 2\*cs])

def draw\_map(): #рисование уровня

for i in range(cells):

for j in range(cells):

x = cs\*j

y = cs\*i

if lvl[i][j] == 6: #игрок

screen.blit(floor, [x, y])

screen.blit(player, [x, y])

ip = i

jp = j

darkness[i][j] = lvl[i][j] #в области, близкой к игроку препятствие для глаз исчезает

darkness[i+1][j] = lvl[i+1][j]

darkness[i-1][j] = lvl[i-1][j]

darkness[i][j+1] = lvl[i][j+1]

darkness[i][j-1] = lvl[i][j-1]

darkness[i+1][j-1] = lvl[i+1][j-1]

darkness[i-1][j+1] = lvl[i-1][j+1]

darkness[i-1][j-1] = lvl[i-1][j-1]

darkness[i+1][j+1] = lvl[i+1][j+1]

if i+2 < cells: #чтобы игрок изначально знал, что впереди враг или выход с уровня

if (lvl[i+2][j] == 3 or lvl[i+2][j] == 10) and lvl[i+1][j] != 0:

darkness[i+2][j] = lvl[i+2][j]

if (lvl[i-2][j] == 3 or lvl[i-2][j] == 10) and lvl[i-1][j] != 0:

darkness[i-2][j] = lvl[i-2][j]

if j+2 < cells:

if (lvl[i][j+2] == 3 or lvl[i][j+2] == 10) and lvl[i][j+1] != 0:

darkness[i][j+2] = lvl[i][j+1]

if (lvl[i][j-2] == 3 or lvl[i][j-2] == 10) and lvl[i][j-1] != 0:

darkness[i][j-2] = lvl[i][j-2]

if lvl[i][j] == 0: #стена

screen.blit(wall, [x, y])

elif lvl[i][j] == 2: #пол

screen.blit(floor, [x, y])

elif lvl[i][j] == 3: #выход с уровня

screen.blit(exit\_lvl, [x, y])

elif lvl[i][j] == 5: #игровые очки

screen.blit(floor, [x, y])

screen.blit(money, [x, y])

elif lvl[i][j] == 10: #враг

screen.blit(floor, [x, y])

screen.blit(enemy, [x, y])

if darkness[i][j] == 1: #препятсвие для глаз

screen.blit(dark, [x, y])

return(ip, jp)

def control(i, j, ip, jp): #передвижение игрока

if event.key == pg.K\_UP:

i = ip - 1

elif event.key == pg.K\_DOWN:

i = ip + 1

elif event.key == pg.K\_LEFT:

j = jp - 1

elif event.key == pg.K\_RIGHT:

j = jp + 1

return(i, j)

def moving(i, j, ip, jp, lvl, lives, sp\_in, spell\_change, spell\_result): #передвижение

global level\_counter

global score

if lvl[i][j] == 0:

i = ip

j = jp

if lvl[i][j] == 2:

lvl[i][j] = 6 #игрок переходит на следующую клетку

lvl[ip][jp] = 2 #предыдущая клетка закрывается полом

ip = i

jp = j

if lvl[i][j] == 5:

lvl[i][j] = 6

lvl[ip][jp] = 2

ip = i

jp = j

score += 1

if lvl[i][j] == 3: #переход на новый уровень

level\_counter += 1

lvl[i][j] = 6

lvl[ip][jp] = 2

ip = i

jp = j

if lvl[i][j] == 10:

lenemy, lives, sp\_in, spell\_change, spell\_result = pve(lives, sp\_in, spell\_change, spell\_result)

if lenemy <= 0:

lvl[i][j] = 6

lvl[ip][jp] = 2

ip = i

jp = j

ip, jp = draw\_map()

return(i, j, ip, jp, lvl, lives, sp\_in, spell\_change, spell\_result)

liters = str('qwertyuiasdfghjklzxcvbnm')

litq = len(liters)

lives = 10

sp\_in = ''

spell\_change = True

spell\_result = ''

def pve(lives, sp\_in, spell\_change, spell\_result): #боевая система

lives\_enemy = 1

spell = ''

spell\_len = random.randint(5, 8)

if spell\_change == True and spell\_result == '': #предотвращает изменение выводимой на экран комбинации букв, если игрок вводит ее

for i in range(spell\_len):

f = random.randint(0, litq-1)

spell += str(liters[f])

spell\_result = spell

spell\_change = False

text = font.render('Введи ' + spell\_result + ' на клавиатуре', 0, text\_color)

pg.draw.rect(screen, txt\_bg\_color, [screenS+2\*cs, 6\*cs, 10\*cs, cs], 0)

screen.blit(text, [screenS+2\*cs, 6\*cs])

inp\_text = font.render('', 0, text\_color)

answer = ''

inp = True

kd = pg.key.name(event.key)

if kd == 'return':

answer = sp\_in

inp = False

spell\_change = True

if kd != 'return' and kd != 'backspace' and kd != 'left' and kd != 'right' and kd != 'up' and kd != 'down':

sp\_in = sp\_in + kd

if kd == 'backspace':

kd = ''

a = len(sp\_in)-1

sp\_in = sp\_in[0:a]

inp\_text = font.render(sp\_in, 0, text\_color)

pg.draw.rect(screen, txt\_bg\_color, [screenS+2\*cs, 7\*cs, 10\*cs, cs], 0)

screen.blit(inp\_text, [screenS+2\*cs, 7\*cs])

pg.draw.rect(screen, txt\_bg\_color, [screenS+2\*cs, 8\*cs, 10\*cs, cs], 0)

if inp == False:

if answer == spell\_result:

lives\_enemy -= 1

spell\_result = ''

sp\_in = ''

text = font.render('Ты изгнал монстра', 0, text\_color)

screen.blit(text, [screenS+2\*cs, 8\*cs])

elif lives > 0:

lives -= random.randint(1, 3)

spell\_result = ''

sp\_in = ''

text = font.render('Монстр ударил тебя', 0, text\_color)

screen.blit(text, [screenS+2\*cs, 8\*cs])

return(lives\_enemy, lives, sp\_in, spell\_change, spell\_result)

ip, jp = draw\_map()

i = ip

j = jp

gameprocess = True

while gameprocess:

text = font.render ('Сокровища: ' + str(score), 0, text\_color)

pg.draw.rect(screen, txt\_bg\_color, [screenS+3\*cs, 3\*cs, 10\*cs, cs], 0)

screen.blit(text, [screenS+3\*cs, 3\*cs])

screen.blit(money, [screenS + 10\*cs, 3\*cs])

text = font.render('Здоровье: ' + str(lives), 0, text\_color)

pg.draw.rect(screen, txt\_bg\_color, [screenS+3\*cs, 4\*cs, 10\*cs, cs], 0)

screen.blit(text, [screenS + 3\*cs, 4\*cs])

screen.blit(live, [screenS + 10\*cs, 5+4\*cs])

if level\_counter == counter + 1 and level\_counter < level\_quality: #когда счетчик уровней изменяется, программа запускается заново

counter = level\_counter

gameprocess = False

if level\_counter >= level\_quality:

lives = 0

for event in pg.event.get():

if event.type == pg.QUIT:

pg.quit()

if lives <= 0:

score = 0

level\_counter = 0

text = font.render("Game over", 0, text\_color)

screen.blit(text, [screenS+cs, 0])

if event.type == pg.USEREVENT:

timer -= 1

if timer <= 0:

menu.menu()

if event.type == pg.KEYDOWN:

i, j = control(i, j, ip, jp)

i, j, ip, jp, lvl, lives, sp\_in, spell\_change, spell\_result = moving(i, j, ip, jp, lvl, lives, sp\_in, spell\_change, spell\_result)

if event.key == pg.K\_ESCAPE:

score = 0

menu.menu()

pg.display.flip()

while True:

main()