|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| **Федеральное государственное бюджетное образовательное учреждение****высшего образования**«МИРЭА - Российский технологический университет»РТУ МИРЭА |

Кафедра ИКБСП

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине

«Интерпретируемый язык программирования высокого уровня»

**Отчёт выполнил студент группы ББСО-03-20 Караблин Илья**

**Задание проекта**

Реализовать игру в жанре scroll shooter

**Программный код с коментариями**

import pygame  
import random  
  
from os import path  
  
img\_dir = path.join(path.dirname(\_\_file\_\_), 'img')  
snd\_dir = path.join(path.dirname(\_\_file\_\_), 'soun')  
txt\_dir = path.join(path.dirname(\_\_file\_\_), 'txt')  
  
#Размеры окна  
WIDTH = 480  
HEIGHT = 600  
FPS = 60  
POWERUP\_TIME = 5000  
  
# Задаем цвета  
WHITE = (255, 255, 255)  
BLACK = (0, 0, 0)  
RED = (255, 0, 0)  
GREEN = (0, 255, 0)  
BLUE = (0, 0, 255)  
YELLOW = (255, 255, 0)  
  
# Создаем игру и окно  
pygame.init()  
pygame.mixer.init()  
screen = pygame.display.set\_mode((WIDTH, HEIGHT))  
pygame.display.set\_caption("SpaceKiller")  
clock = pygame.time.Clock()  
  
font\_name = pygame.font.match\_font('arial')  
# Запись нового значения в таблицу рекордов  
def new\_record(filename, newstate):  
 # Загрузка таблицы рекордов  
 record = open(path.join(txt\_dir, filename), 'a+')  
 record.write(newstate+'\n')  
 record.close()  
# Выборка лучшего рекорда  
def best\_record(filename):  
 record = open(path.join(txt\_dir, filename), 'r')  
 scoreList = record.readlines()  
 scoreList = [line.rstrip() for line in scoreList]  
 print(scoreList)  
 bestscore = max(scoreList)  
 record.close()  
 return bestscore  
# Вывод текста на экран  
def draw\_text(surf, text, size, x, y):  
 font = pygame.font.Font(font\_name, size)  
 text\_surface = font.render(text, True, WHITE)  
 text\_rect = text\_surface.get\_rect()  
 text\_rect.midtop = (x, y)  
 surf.blit(text\_surface, text\_rect)  
# Вывод жизней на экран  
def draw\_lives(surf, x, y, lives, img):  
 for i in range(lives):  
 img\_rect = img.get\_rect()  
 img\_rect.x = x + 30 \* i  
 img\_rect.y = y  
 surf.blit(img, img\_rect)  
# Создание новых мобов на экран  
def newmob():  
 m = Mob()  
 all\_sprites.add(m)  
 mobs.add(m)  
# Вывод щита на экран  
def draw\_shield\_bar(surf, x, y, pct):  
 if pct < 0:  
 pct = 0  
 BAR\_LENGTH = 100  
 BAR\_HEIGHT = 10  
 fill = (pct / 100) \* BAR\_LENGTH  
 outline\_rect = pygame.Rect(x, y, BAR\_LENGTH, BAR\_HEIGHT)  
 fill\_rect = pygame.Rect(x, y, fill, BAR\_HEIGHT)  
 pygame.draw.rect(surf, GREEN, fill\_rect)  
 pygame.draw.rect(surf, WHITE, outline\_rect, 2)  
  
# Класс игрока  
class Player(pygame.sprite.Sprite):  
 def \_\_init\_\_(self):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.image = pygame.transform.scale(player\_img, (100, 100))  
 self.lives = 3  
 self.hidden = False  
 self.hide\_timer = pygame.time.get\_ticks()  
 self.image.set\_colorkey(BLACK)  
 self.rect = self.image.get\_rect()  
 self.radius = 20  
 #pygame.draw.circle(self.image, RED, self.rect.center, self.radius)  
 self.rect.centerx = WIDTH / 2  
 self.rect.bottom = HEIGHT - 10  
 self.speedx = 0  
 self.shield = 100  
 self.shoot\_delay = 250  
 self.last\_shot = pygame.time.get\_ticks()  
 self.power = 1  
 self.power\_time = pygame.time.get\_ticks()  
  
 #pygame.draw.circle(self.image, RED, self.rect.center, self.radius)  
  
 def update(self):  
 self.speedx = 0  
 keystate = pygame.key.get\_pressed()  
 if keystate[pygame.K\_LEFT]:  
 self.speedx = -8  
 if keystate[pygame.K\_RIGHT]:  
 self.speedx = 8  
 if keystate[pygame.K\_SPACE]:  
 self.shoot()  
 self.rect.x += self.speedx  
 if self.rect.right > WIDTH:  
 self.rect.right = WIDTH  
 if self.rect.left < 0:  
 self.rect.left = 0  
 # показать, если скрыто  
 if self.hidden and pygame.time.get\_ticks() - self.hide\_timer > 1000:  
 self.hidden = False  
 self.rect.centerx = WIDTH / 2  
 self.rect.bottom = HEIGHT - 10  
 # тайм-аут для бонусов  
 if self.power >= 2 and pygame.time.get\_ticks() - self.power\_time > POWERUP\_TIME:  
 self.power -= 1  
 self.power\_time = pygame.time.get\_ticks()  
 # функция для стрельбы  
 def shoot(self):  
 now = pygame.time.get\_ticks()  
 if now - self.last\_shot > self.shoot\_delay:  
 self.last\_shot = now  
 # в случае без улучшения создаёт одну пулю  
 if self.power == 1:  
 bullet = Bullet(self.rect.centerx, self.rect.top)  
 all\_sprites.add(bullet)  
 bullets.add(bullet)  
 shoot\_sound.play()  
 # иначе 2  
 if self.power >= 2:  
 bullet1 = Bullet(self.rect.left+30, self.rect.centery)  
 bullet2 = Bullet(self.rect.right-30, self.rect.centery)  
 all\_sprites.add(bullet1)  
 all\_sprites.add(bullet2)  
 bullets.add(bullet1)  
 bullets.add(bullet2)  
 shoot\_sound.play()  
 # подбор улучшения пули  
 def powerup(self):  
 self.power += 1  
 self.power\_time = pygame.time.get\_ticks()  
 def hide(self):  
 # временно скрыть игрока  
 self.hidden = True  
 self.hide\_timer = pygame.time.get\_ticks()  
 self.rect.center = (WIDTH / 2, HEIGHT + 200)  
 # Класс астеройда  
class Mob(pygame.sprite.Sprite):  
 def \_\_init\_\_(self):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.image\_orig = random.choice(meteor\_images)  
 self.image\_orig.set\_colorkey(BLACK)  
 self.image = self.image\_orig.copy()  
 self.rect = self.image.get\_rect()  
 # позиция задаётся случайным образом на промежутках  
 self.rect.x = random.randrange(WIDTH - self.rect.width)  
 self.rect.y = random.randrange(-100, -40)  
 # Так же скорость  
 self.speedy = random.randrange(1, 8)  
 self.speedx = random.randrange(-3, 3)  
 self.rect = self.image.get\_rect()  
 # Так же задаётся колизия объекта в виде окружности  
 self.radius = int(self.rect.width / 4)  
 self.rot = 0  
 self.rot\_speed = random.randrange(-8, 8)  
 self.last\_update = pygame.time.get\_ticks()  
 #pygame.draw.circle(self.image, RED, self.rect.center, self.radius)  
 # функция для анимации вращения астеройда  
 def rotate(self):  
 now = pygame.time.get\_ticks()  
 if now - self.last\_update > 50:  
 self.last\_update = now  
 self.rot = (self.rot + self.rot\_speed) % 360  
 new\_image = pygame.transform.rotate(self.image\_orig, self.rot)  
 old\_center = self.rect.center  
 self.image = new\_image  
 self.rect = self.image.get\_rect()  
 self.rect.center = old\_center  
  
  
 def update(self):  
 self.rotate()  
 self.rect.x += self.speedx  
 self.rect.y += self.speedy  
 #Уничтажается если заходит за пределы экрана  
 if self.rect.top > HEIGHT + 10 or self.rect.left < -120 or self.rect.right > WIDTH + 120:  
 self.rect.x = random.randrange(WIDTH - self.rect.width)  
 self.rect.y = random.randrange(-100, -40)  
 self.speedy = random.randrange(1, 8)  
  
# Класс пули  
class Bullet(pygame.sprite.Sprite):  
 def \_\_init\_\_(self, x, y):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.image = pygame.transform.scale(player\_img, (25, 25 ))  
 self.image = bullet\_img  
 self.image.set\_colorkey(BLACK)  
 self.rect = self.image.get\_rect()  
 self.radius = int(self.rect.width / 10)  
 self.rect.bottom = y  
 self.rect.centerx = x  
 self.speedy = -10  
  
  
 def update(self):  
 self.rect.y += self.speedy  
 # убить, если он заходит за верхнюю часть экрана  
 if self.rect.bottom < 0:  
 self.kill()  
# класс для создания анимции взрыва  
class Explosion(pygame.sprite.Sprite):  
 def \_\_init\_\_(self, center, size):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.size = size  
 self.image = explosion\_anim[self.size][0]  
 self.rect = self.image.get\_rect()  
 self.rect.center = center  
 self.frame = 0  
 self.last\_update = pygame.time.get\_ticks()  
 self.frame\_rate = 50  
  
 def update(self):  
 now = pygame.time.get\_ticks()  
 if now - self.last\_update > self.frame\_rate:  
 self.last\_update = now  
 self.frame += 1  
 if self.frame == len(explosion\_anim[self.size]):  
 self.kill()  
 else:  
 center = self.rect.center  
 self.image = explosion\_anim[self.size][self.frame]  
 self.rect = self.image.get\_rect()  
 self.rect.center = center  
#Класс бонусов  
class Pow(pygame.sprite.Sprite):  
 def \_\_init\_\_(self, center):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.type = random.choice(['bullet\_upgrade', 'HP'])  
 self.image = powerup\_images[self.type]  
 self.image.set\_colorkey(BLACK)  
 self.rect = self.image.get\_rect()  
 self.rect.center = center  
 self.speedy = 2  
  
 def update(self):  
 self.rect.y += self.speedy  
 # убить, если он сдвинется с нижней части экрана  
 if self.rect.top > HEIGHT:  
 self.kill()  
# Загрузка мелодий игры  
shoot\_sound = pygame.mixer.Sound(path.join(snd\_dir, 'shootlaser.wav'))  
shoot\_sound.set\_volume(0.3)  
pow\_sound = pygame.mixer.Sound(path.join(snd\_dir, 'pick\_up.wav'))  
pow\_sound.set\_volume(0.3)  
expl\_sounds = []  
for snd in ['boom.wav','boom1.wav','boom2.wav']:  
 expl\_sound = pygame.mixer.Sound(path.join(snd\_dir, snd))  
 expl\_sound.set\_volume(0.2)  
 expl\_sounds.append(expl\_sound)  
  
  
pygame.mixer.music.load(path.join(snd\_dir, 'Megadrive\_-\_NARC\_Hotline\_Miami\_2\_Wrong\_Number\_OST\_63637466.ogg'))  
pygame.mixer.music.set\_volume(0.2)  
# Загрузка всей игровой графики  
background = pygame.image.load(path.join(img\_dir, 'Background1.png')).convert()  
player\_img = pygame.image.load(path.join(img\_dir, "spaceship1.png")).convert()  
bullet\_img = pygame.image.load(path.join(img\_dir, "bullet.png")).convert()  
powerup\_images = {}  
powerup\_images['HP'] = pygame.image.load(path.join(img\_dir, 'HP.png')).convert()  
powerup\_images['bullet\_upgrade'] = pygame.image.load(path.join(img\_dir, 'bullet\_upgrade.png')).convert()  
background\_rect = background.get\_rect()  
meteor\_images = []  
meteor\_list = ['asteroid1.png','asteroid1big.png']  
for img in meteor\_list:  
 meteor\_images.append(pygame.image.load(path.join(img\_dir, img)).convert())  
  
explosion\_anim = {}  
explosion\_anim['lg'] = []  
explosion\_anim['sm'] = []  
for i in range(3):  
 filename = 'regularExplosion0{}.png'.format(i)  
 img = pygame.image.load(path.join(img\_dir, filename)).convert()  
 img.set\_colorkey(BLACK)  
 img\_lg = pygame.transform.scale(img, (80, 80))  
 explosion\_anim['lg'].append(img\_lg)  
 img\_sm = pygame.transform.scale(img, (72, 72))  
 explosion\_anim['sm'].append(img\_sm)  
  
player\_img = pygame.image.load(path.join(img\_dir, "spaceship1.png")).convert()  
player\_img1 = pygame.image.load(path.join(img\_dir, "spaceship2.png")).convert()  
player\_img2 = pygame.image.load(path.join(img\_dir, "spaceship3.png")).convert()  
player\_mini\_img = pygame.transform.scale(player\_img, (45, 45))  
player\_mini\_img.set\_colorkey(BLACK)  
powerups = pygame.sprite.Group()  
all\_sprites = pygame.sprite.Group()  
mobs = pygame.sprite.Group()  
bullets = pygame.sprite.Group()  
player = Player()  
all\_sprites.add(player)  
for i in range(8):  
 newmob()  
score = 0  
pygame.mixer.music.play(loops=-1)  
#Меню игры при начале и смерти игрока  
def show\_go\_screen():  
 screen.blit(background, background\_rect)  
 draw\_text(screen, "Space Killer!", 64, WIDTH / 2, HEIGHT / 4)  
 draw\_text(screen, "Press a key to begin", 18, WIDTH / 2, HEIGHT \* 3 / 4)  
 if (firstPlay):  
 draw\_text(screen, "Arrow keys move, Space to fire", 22,  
 WIDTH / 2, HEIGHT / 2)  
 else:  
 new\_record('record.txt',str(score))  
 draw\_text(screen, "Your best score " + best\_record('record.txt'), 18, WIDTH / 2, HEIGHT \* 2 / 4)  
  
 pygame.display.flip()  
 waiting = True  
 while waiting:  
 clock.tick(FPS)  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 pygame.quit()  
 if event.type == pygame.KEYUP:  
 waiting = False  
# Цикл игры  
game\_over = True  
running = True  
firstPlay= True  
while running:  
 if game\_over:  
 show\_go\_screen()  
 all\_sprites = pygame.sprite.Group()  
 mobs = pygame.sprite.Group()  
 bullets = pygame.sprite.Group()  
 powerups = pygame.sprite.Group()  
 player = Player()  
 all\_sprites.add(player)  
 for i in range(8):  
 newmob()  
 score = 0  
 firstPlay = False  
 game\_over = False  
 # Держим цикл на правильной скорости  
 clock.tick(FPS)  
 # Ввод процесса (события)  
 for event in pygame.event.get():  
 # проверка для закрытия окна  
 if event.type == pygame.QUIT:  
 running = False  
  
 # Обновление  
 all\_sprites.update()  
 # Проверка, не ударила ли моба пуля  
 hits = pygame.sprite.groupcollide(mobs, bullets, True, True, pygame.sprite.collide\_circle)  
 for hit in hits:  
 score += 50 - hit.radius  
 random.choice(expl\_sounds).play()  
 expl = Explosion(hit.rect.center, 'lg')  
 all\_sprites.add(expl)  
 if random.random() > 0.9:  
 pow = Pow(hit.rect.center)  
 all\_sprites.add(pow)  
 powerups.add(pow)  
 newmob()  
  
 # Проверка, не ударил ли моб игрока  
 hits = pygame.sprite.spritecollide(player, mobs, True, pygame.sprite.collide\_circle)  
 for hit in hits:  
 player.shield -= hit.radius \* 2  
 expl = Explosion(hit.rect.center, 'sm')  
 all\_sprites.add(expl)  
 newmob()  
 # в зависимости от полученного урона меняется внешность корабля  
 if 50 <= player.shield < 70:  
 player.image = pygame.transform.scale(player\_img1, (100, 100))  
 player.image.set\_colorkey(BLACK)  
 if 0 < player.shield < 50:  
 player.image = pygame.transform.scale(player\_img2, (100, 100))  
 player.image.set\_colorkey(BLACK)  
 if player.shield <= 0:  
 expl\_player = Explosion(player.rect.center, 'lg')  
 all\_sprites.add(expl\_player)  
 player.image = pygame.transform.scale(player\_img, (100, 100))  
 player.image.set\_colorkey(BLACK)  
 player.hide()  
 player.lives -= 1  
 player.shield = 100  
 # Если игрок умер, игра окончена  
 if player.lives == 0 and not expl\_player.alive():  
 game\_over = True  
 # Проверка столкновений игрока и улучшения  
 hits = pygame.sprite.spritecollide(player, powerups, True)  
 for hit in hits:  
 pow\_sound.play()  
 if hit.type == 'HP':  
 player.shield += random.randrange(10, 30)  
 if 50 <= player.shield < 70:  
 player.image = pygame.transform.scale(player\_img1, (100, 100))  
 player.image.set\_colorkey(BLACK)  
 if 0 < player.shield < 50:  
 player.image = pygame.transform.scale(player\_img2, (100, 100))  
 player.image.set\_colorkey(BLACK)  
 if player.shield >= 100:  
 player.image = pygame.transform.scale(player\_img, (100, 100))  
 player.image.set\_colorkey(BLACK)  
 player.shield = 100  
 if hit.type == 'bullet\_upgrade':  
 player.powerup()  
 # Рендеринг  
 screen.fill(BLACK)  
 screen.blit(background, background\_rect)  
 all\_sprites.draw(screen)  
 draw\_text(screen, 'SCORE '+str(score), 18, WIDTH / 2, 10)  
 draw\_lives(screen, WIDTH - 100, 5, player.lives,player\_mini\_img)  
 draw\_shield\_bar(screen, 5, 5, player.shield)  
 # После отрисовки всего, переворачиваем экран  
 pygame.display.flip()  
  
pygame.quit()

**Листинг 1**

В данном проекте была использована библиотека Pygame. Pygame – это игровая библиотека, которая помогает создавать игры. С её помощью становится проще работать с графикой и анимацией, звуком и управлением. Сам проект из себя представляет high scroll shooter. Управление кораблём осуществляется стрелками, а стрельба пробелом. При попадании снарядов в метеоры начисляются очки соответствующими размерами метеора, а так же из разрушенного метеора с некоторым шансом выпадает улучшение. Подбирая улучшения, игрок может восстановить щиты или получить улучшение двойной стрельбы (в зависимости от выбитого улучшения), после которой игрок начинает стрелять сразу двумя оружиями. При столкновении с метеором у корабля отнимаются щиты. Когда щиты кончаются, у игрока отнимается жизнь, он прячется на некоторое время, и он появляется с полными щитами. Когда жизни закончатся, закончится и игра. После появится меню, в котором будет написан лучший рекорд. Вся игра выполнена в стиле пиксельарт.