from pygame import\*

from random importrandint

from time importtime astimer

font.init()

font1 = font.Sysfount("Arial", 80)

win = font1.render('YOU WIN!', True, (255, 255, 255))

lose = font1.render('YOU LOSE!', True, (180, 0, 0))

font2 = font.Sysfount("Arial", 36)

#фоновая музыка

mixer.init()

mixer.music.load('space.ogg')

mixer.music.play()

fire\_sound = mixer.Sound('fire.ogg')

#нам нужны такие картинки:

img\_back = "galaxy.jpg"#фон игры

img\_bullet = "bullet.png"#пуля

img\_hero = "rocket.png"#герой

img\_enemy = "ufo.png"#враг

img\_ast = "asteroid.png"#астероид

score = 0#сбито кораблей

goal = 20#столько кораблей нужно сбить для победы

lost = 0#пропущено кораблей

max\_lost = 10#проиграли, если пропустили столько кораблей

life = 3 #очки жизни

#класс-родитель для других спрайтов

classGameSprite(sprite.Sprite):

#конструктор класса

def\_\_init\_\_(self, player\_image, player\_x, player\_y, size\_x, size\_y, player\_speed):

#вызываем конструктор класса (Sprite):

sprite.Sprite.\_\_init\_\_(self)

#каждый спрайт должен хранить свойство image - изображение

self.image = transform.scale(image.load(player\_image), (size\_x, size\_y))

self.speed = player\_speed

#каждый спрайт должен хранить свойство rect - прямоугольник, в который он вписан

self.rect = self.image.get\_rect()

self.rect.x = player\_x

self.rect.y = player\_y

#метод, отрисовывающий героя на окне

defreset(self):

window.blit(self.image, (self.rect.x, self.rect.y))

#класс главного игрока

classPlayer(GameSprite):

#метод для управления спрайтом стрелками клавиатуры

defupdate(self):

keys = key.get\_pressed()

ifkeys[K\_LEFT] andself.rect.x > 5:

self.rect.x -= self.speed

ifkeys[K\_RIGHT] andself.rect.x < win\_width - 80:

self.rect.x += self.speed

#метод "выстрел" (используем место игрока, чтобы создать там пулю)

deffire(self):

bullet = Bullet(img\_bullet, self.rect.centerx, self.rect.top, 15, 20, -15)

bullets.add(bullet)

#класс спрайта-врага

classEnemy(GameSprite):

#движение врага

defupdate(self):

self.rect.y += self.speed

globallost

#исчезает, если дойдёт до края экрана

ifself.rect.y > win\_height:

self.rect.x = randint(80, win\_width - 80)

self.rect.y = 0

lost = lost + 1

#класс спрайта-пули

classBullet(GameSprite):

#движение врага

defupdate(self):

self.rect.y += self.speed

#исчезает, если дойдёт до края экрана

ifself.rect.y < 0:

self.kill()

#создаём окошко

win\_width = 700

win\_height = 500

display.set\_caption("Shooter")

window = display.set\_mode((win\_width, win\_height))

background = transform.scale(image.load(img\_back), (win\_width, win\_height))

#создаём спрайты

ship = Player(img\_hero, 5, win\_height - 100, 80, 100, 10)

#создание группы спрайтов-врагов

monsters = sprite.Group()

fori inrange(1, 6):

monster = Enemy(img\_enemy, randint(80, win\_width - 80), -40, 80, 50, randint(1, 5))

monsters.add(monster)

#создание группы спрайтов-астероидов ()

asteroids = sprite.Group()

fori inrange(1, 3):

asteroid = Enemy(img\_ast, randint(30, win\_width - 30), -40, 80, 50, randint(1, 7))

asteroids.add(asteroid)

bullets = sprite.Group()

#переменная "игра закончилась": как только там True, в основном цикле перестают работать спрайты

finish = False

#основной цикл игры:

run = True#флаг сбрасывается кнопкой закрытия окна

rel\_time = False#флаг, отвечающий за перезарядку

num\_fire = 0 #переменная для подсчёта выстрелов

whilerun:

#событие нажатия на кнопку “Закрыть”

fore inevent.get():

ife.type == QUIT:

run = False

#событие нажатия на пробел - спрайт стреляет

elife.type == KEYDOWN:

ife.key == K\_SPACE:

#проверяем, сколько выстрелов сделано и не происходит ли перезарядка

ifnum\_fire < 5andrel\_time == False:

num\_fire = num\_fire + 1

fire\_sound.play()

ship.fire()

ifnum\_fire >= 5andrel\_time == False: #если игрок сделал 5 выстрелов

last\_time = timer() #засекаем время, когда это произошло

rel\_time = True#ставим флаг перезарядки

#сама игра: действия спрайтов, проверка правил игры, перерисовка

ifnotfinish:

#обновляем фон

window.blit(background,(0,0))

#производим движения спрайтов

ship.update()

monsters.update()

asteroids.update()

bullets.update()

#обновляем их в новом местоположении при каждой итерации цикла

ship.reset()

monsters.draw(window)

asteroids.draw(window)

bullets.draw(window)

#перезарядка

ifrel\_time == True:

now\_time = timer() #считываем время

ifnow\_time - last\_time < 3: #пока не прошло 3 секунды выводим информацию о перезарядке

reload= font2.render('Wait, reload...', 1, (150, 0, 0))

window.blit(reload, (260, 460))

else:

num\_fire = 0 #обнуляем счётчик пуль

rel\_time = False#сбрасываем флаг перезарядки

#проверка столкновения пули и монстров (и монстр, и пуля при касании исчезают)

collides = sprite.groupcollide(monsters, bullets, True, True)

forc incollides:

#этот цикл повторится столько раз, сколько монстров подбито

score = score + 1

monster = Enemy(img\_enemy, randint(80, win\_width - 80), -40, 80, 50, randint(1, 5))

monsters.add(monster)

#если спрайт коснулся врага, уменьшает жизнь

ifsprite.spritecollide(ship, monsters, False) orsprite.spritecollide(ship, asteroids, False):

sprite.spritecollide(ship, monsters, True)

sprite.spritecollide(ship, asteroids, True)

life = life -1

#проигрыш

iflife == 0orlost >= max\_lost:

finish = True#проиграли, ставим фон и больше не управляем спрайтами.

window.blit(lose, (200, 200))

#проверка выигрыша: сколько очков набрали?

ifscore >= goal:

finish = True

window.blit(win, (200, 200))

#пишем текст на экране

text = font2.render("Счет: "+ str(score), 1, (255, 255, 255))

window.blit(text, (10, 20))

text\_lose = font2.render("Пропущено: "+ str(lost), 1, (255, 255, 255))

window.blit(text\_lose, (10, 50))

#задаём разный цвет в зависимости от количества жизней

iflife == 3:

life\_color = (0, 150, 0)

iflife == 2:

life\_color = (150, 150, 0)

iflife == 1:

life\_color = (150, 0, 0)

text\_life = font1.render(str(life), 1, life\_color)

window.blit(text\_life, (650, 10))

display.update()

#бонус: автоматический перезапуск игры

else:

finish = False

score = 0

lost = 0

num\_fire = 0

life = 3

forb inbullets:

b.kill()

form inmonsters:

m.kill()

fora inasteroids:

a.kill()

time.delay(3000)

fori inrange(1, 6):

monster = Enemy(img\_enemy, randint(80, win\_width - 80), -40, 80, 50, randint(1, 5))

monsters.add(monster)

fori inrange(1, 3):

asteroid = Enemy(img\_ast, randint(30, win\_width - 30), -40, 80, 50, randint(1, 7))

asteroids.add(asteroid)

time.delay(50)