***Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования***

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет***  ***имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

**Факультет ИУ**

**Кафедра ИУ10**

**Отчет**

**по лабораторной работе № 5**

**Дисциплина: Информатика**

**Тема: «Игра “Шарик”»**

Работу выполнил: Андреев Александр Вадимович ИУ10-23

Проверил: Буркацкий Кирилл Александрович

Москва, 2023

Содержание:

* **Отчет**
* **Цель работы**
* **Ход работы**

**Цель работы:** Написать код игры.

import pygame

from pygame.draw import \*

from random import randint

pygame.init()

FPS = 200

screen = pygame.display.set\_mode((800, 600))

RED = (255, 0, 0)

BLUE = (0, 0, 255)

YELLOW = (255, 255, 0)

GREEN = (0, 255, 0)

MAGENTA = (255, 0, 255)

CYAN = (0, 255, 255)

BLACK = (0, 0, 0)

WHITE = (255, 255, 255)

COLORS = [RED, BLUE, YELLOW, GREEN, MAGENTA, CYAN]

class Ball:

def \_\_init\_\_(self, screen, x, y, radius, color, speed, lifetime):

self.screen = screen

self.x = x

self.y = y

self.radius = radius

self.color = color

self.speed = speed

self.lifetime = lifetime

def draw(self):

pygame.draw.circle(self.screen, self.color, (self.x, self.y), self.radius)

def update(self):

self.x += self.speed[0]

self.y += self.speed[1]

if self.x - self.radius < 0 or self.x + self.radius > 800:

self.speed[0] = -self.speed[0]

if self.y - self.radius < 0 or self.y + self.radius > 600:

self.speed[1] = -self.speed[1]

self.lifetime -= 1

return self.lifetime > 0

class Game:

def \_\_init\_\_(self):

self.score = 0

self.balls = []

self.screen = pygame.display.set\_mode((800, 600))

self.started = False

def restart(self):

self.score = 0

self.balls = []

self.started = False

def new\_ball(self):

radius = randint(30, 50)

x = randint(10 + randint(20,60), 800 - randint(20,60))

y = randint(10 + randint(20,60), 600 - randint(20,60))

color = COLORS[randint(0, 5)]

speed = [randint(-1, 2), randint(-1, 2)]

ball = Ball(self.screen, x, y, radius, color, speed, 600)

self.balls.append(ball)

def delete\_ball(self, ball):

self.balls.remove(ball)

def check\_click(self, x, y):

for ball in self.balls:

if (x - ball.x) \*\* 2 + (y - ball.y) \*\* 2 <= ball.radius \*\* 2:

self.delete\_ball(ball)

self.score += 1

return

def game\_loop(self):

finished = False

clock = pygame.time.Clock()

font = pygame.font.Font(None, 36)

while not finished:

clock.tick(FPS)

for event in pygame.event.get():

if event.type == pygame.QUIT:

finished = True

elif event.type == pygame.MOUSEBUTTONDOWN:

if not self.started:

self.started = True

self.check\_click(\*event.pos)

if randint(1, 90) == 1:

self.new\_ball()

self.screen.fill(BLACK)

for ball in self.balls:

if not ball.update():

self.delete\_ball(ball)

for ball in self.balls:

ball.draw()

text = font.render(f"Score: {self.score}", True, WHITE)

self.screen.blit(text, (10, 10))

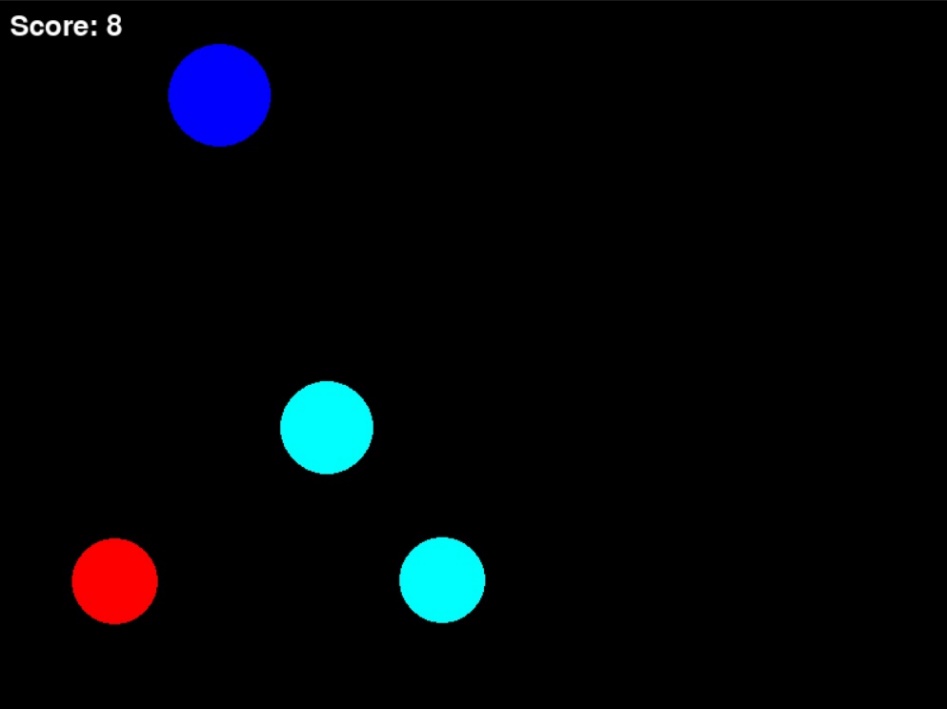
pygame.display.update()

pygame.quit()

if \_\_name\_\_ == "\_\_main\_\_":

game = Game()

game.game\_loop()

****

**Вывод: Мы написали код игры. Сделали передвижение целей и их отскок от стенок. Код понятен и легко редактируем.**