

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по ДЗ

Выполнил:

студент группы ИУ5-34Б

Ступницкий Михаил

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2023 г.

Задание:

Реализовать игру с использованием библиотеки Kivy

Текст программы main.py:

```
from kivy.app import App
from kivy.uix.widget import Widget
from kivy.properties import (
    NumericProperty, ReferenceListProperty, ObjectProperty
)
from kivy.vector import Vector
from kivy.clock import Clock
from kivy.graphics import *
from random import randint
from kivy.core.window import Window
import math

class Fire(Widget):
    score = NumericProperty(0)
    velocity_x = NumericProperty(0)
    velocity_y = NumericProperty(0)

    velocity = ReferenceListProperty(velocity_x, velocity_y)

    def bounce_ball(self, ball):
        if self.collide_widget(ball):
            vx, vy = ball.velocity
            ball.velocity = vx + self.velocity_x/4, vy + self.velocity_y /4
            self.velocity_x *= -100000
    def move(self):
        self.pos = Vector(*self.velocity) + self.pos

    def check_bounds(self, height):
        if (self.y < 0 or self.top > height):
            self.velocity_y *= -1

class Gun(Widget):
    rot = NumericProperty(0)

    def updater1(self):
        self.rot += 2

    def updater2(self):
        self.rot -= 2

class Ball(Widget):
    velocity_x = NumericProperty(0)
    velocity_y = NumericProperty(0)

    velocity = ReferenceListProperty(velocity_x, velocity_y)

    def move(self):
        self.pos = Vector(*self.velocity) + self.pos

class GunGame(Widget):
    ball = ObjectProperty(None)
    gun1 = ObjectProperty(None)
    gun2 = ObjectProperty(None)
    fire1 = ObjectProperty(None)
```

```

fire2 = ObjectProperty(None)

def set(self):
    self.gun1.rot = 90
    self.gun2.rot = 180
def serve_ball(self):
    self.ball.center = self.center
    self.ball.velocity = Vector(0, 0)

    def serve_fire1(self):
        self.fire1.center = (self.gun1.pos[0] + self.gun1.width/2 *
math.cos(math.radians(self.gun1.rot + 270)) + self.gun1.height/8 *
math.cos(math.radians(360 - self.gun1.rot)),
self.gun1.pos[1] + self.gun1.width/2 *
math.sin(math.radians(self.gun1.rot + 270)))
        self.fire1.velocity = Vector(3, 0).rotate(self.gun1.rot + 270)

    def serve_fire2(self):
        self.fire2.center = (self.gun2.pos[0] + self.gun2.width / 2 *
math.cos(
math.radians(self.gun2.rot + 270)) + self.gun2.height / 8 *
math.cos(math.radians(360 - self.gun2.rot)),
self.gun2.pos[1] + self.gun2.width / 2 *
math.sin(math.radians(self.gun2.rot + 270)))
        self.fire2.velocity = Vector(3, 0).rotate(self.gun2.rot + 270)

def update(self, dt):
    self.ball.move()
    self.fire1.move()
    self.fire2.move()

    self.fire1.bounce_ball(self.ball)
    self.fire2.bounce_ball(self.ball)

    self.gun1.updater1()
    self.gun2.updater2()

    if (self.ball.y < 0 or self.ball.top > self.height):
        self.ball.velocity_y *= -1

    self.fire1.check_bounds(self.height)
    self.fire2.check_bounds(self.height)

    if self.ball.x < self.width * 1 / 6 - self.ball.width:
        self.fire2.score += 1
        self.serve_ball()

    if self.ball.x > self.width * 5 / 6 :
        self.fire1.score += 1
        self.serve_ball()

def __init__(self, **kwargs):
    super(GunGame, self).__init__(**kwargs)
    self._keyboard = Window.request_keyboard(None, self)
    self._keyboard.bind(on_key_down=self.key_action)

def key_action(self, keyboard, keycode, text, modifiers):
    if keycode[1] == 'a': # Например, если вы хотите запустить функцию
при нажатии клавиши "a"
        self.serve_fire1()

    if keycode[1] == 'l': # Например, если вы хотите запустить функцию
при нажатии клавиши "a"
        self.serve_fire2()

```

```

class GunApp(App):
    def build(self):
        game = GunGame()
        game.serve_ball()
        Clock.schedule_interval(game.update, 1.0 / 120)
        return game

if __name__ == '__main__':
    GunApp().run()

```

Текст файла gun.kv

```
#:kivy 1.11.1
```

```

<Gun>:
    canvas.before:
        Color:
            rgba: 1, 1, 1, 1
        PushMatrix:
        Rotate:
            angle: -90 + self.rot
            origin: self.pos
        Color:
            rgba: 1, 0, 0, 1
        Rectangle:
            pos: root.pos
            size: 50, 20
    canvas.after:
        PopMatrix

```

```

<Ball>:
    size: 100, 100

    canvas.before:
        Color:
            rgba: 0, 0, 1, 1
        Ellipse:
            pos: self.pos
            size: self.size

```

```

<Fire>:
    size: 20, 20

    canvas.before:
        Color:
            rgba: 1, 1, 1, 1
        Ellipse:
            pos: self.pos
            size: self.size

```

```

<GunGame>:
    ball: main_ball
    gun1: left_gun
    gun2: right_gun
    fire1: left_fire
    fire2: right_fire

    canvas.before:
        Color:

```

```

        rgba: 0.04, 0.85, 0.316, 1
Rectangle:
    pos: self.pos
    size: self.size
Color:
    rgba: 1, 1, 1, 1
Rectangle:
    pos: self.center_x - 5, 0
    size: 10, self.height
Rectangle:
    pos: self.center_x - self.width/3, 0
    size: 10, self.height
Rectangle:
    pos: self.center_x + self.width/3, 0
    size: 10, self.height

Label:
    font_size: 70
    center_x: root.width / 4
    top: root.top - 50
    text: str(root.fire1.score)

Label:
    font_size: 70
    center_x: root.width * 3 / 4
    top: root.top - 50
    text: str(root.fire2.score)

Ball:
    id: main_ball
    center: root.center

Gun:
    id: left_gun
    x: root.x + self.width/2
    y: root.center_y

Gun:
    id: right_gun
    x: root.width - self.width/2
    y: root.center_y

Fire:
    id: left_fire
    x: root.x + self.width/2
    y: root.center_y

Fire:
    id: right_fire
    x: root.width
    y: root.center_y - 10

```

Результат выполнения программы:

