

**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 ЛР№4

Выполнил:

студент группы ИУ5-34Б

Ступницкий Михаил

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2023 г.

Задание:

Написание игры “Pong” 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 random import randint

class PongPaddle(Widget):
    score = NumericProperty(0)

    def bounce_ball(self, ball):
        if self.collide_widget(ball):
            vx, vy = ball.velocity
            offset = (ball.center_y - self.center_y) / (self.height / 1.5)
            bounced = Vector(-1 * vx, vy)
            vel = bounced * 1.1
            ball.velocity = vel.x, vel.y + offset

class PongBall(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 PongGame(Widget):
    ball = ObjectProperty(None)
    player1 = ObjectProperty(None)
    player2 = ObjectProperty(None)

    def serve_ball(self):
        self.ball.center = self.center
        self.ball.velocity = Vector(4, 0).rotate(randint(0, 360))

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

        self.player1.bounce_ball(self.ball)
        self.player2.bounce_ball(self.ball)

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

        if self.ball.x < self.x:
            self.player2.score += 1
            self.serve_ball()

        if self.ball.x > self.width:
            self.player1.score += 1
            self.serve_ball()
```

```

def on_touch_move(self, touch):
    if touch.x < self.width / 3:
        self.player1.center_y = touch.y

    if touch.x > self.width - self.width / 3:
        self.player2.center_y = touch.y

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

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

```

Текст файла pong.kv

```

#:kivy 1.11.1

# Тут нарисуем наш шарик, называем этот объект точно также как и класс в коде
<PongBall>:
    size: 50, 50

    canvas:
        Ellipse:
            pos: self.pos
            size: self.size

# Тут нарисуем панельку игрока, называем этот объект точно также как и класс
в коде
<PongPaddle>:
    size: 25, 200
    canvas:
        Rectangle:
            pos: self.pos
            size: self.size

# А это игровое поле =)
<PongGame>:
    ## тут привязываем шарик к свойству ball
    ball: pong_ball
    player1: player_left
    player2: player_right

    canvas:
        Rectangle:
            ## Бординг посередине
            pos: self.center_x - 5, 0
            size: 10, self.height

    Label:
        ## Очки игрока слева
        font_size: 70
        center_x: root.width / 4
        top: root.top - 50
        text: str(root.player1.score)

    Label:
        ## Очки игрока справа
        font_size: 70
        center_x: root.width * 3 / 4

```

```

        top: root.top - 50
        text: str(root.player2.score)

## а тут создаем экземпляр нашего шарика в игровом поле
PongBall:
    id: pong_ball
    center: self.parent.center

## создаем игрока 1
PongPaddle:
    id: player_left
    x: root.x
    center_y: root.center_y

## создаем игрока 2
PongPaddle:
    id: player_right
    x: root.width - self.width
    center_y: root.center_y

```

Результат выполнения программы:

