

Лабораториска Worm – Ведран Крстевски 152030

1. По 45 секунди од почетокот на играта додадете друг црв. Оригиналниот црв треба да го избегнува вториот црв. Ако го допре со главата, неговото тело расте за еден сегмент. Ако вториот црв го допре оригиналниот, тогаш неговото тело се продолжува за еден сегмент. Движењето на вториот црв е случајно.

```
startx1 = random.randint(5, CELLWIDTH - 6)
starty1 = random.randint(5, CELLHEIGHT - 6)
wormCoords2 = [{'x': startx, 'y': starty},
                {'x': startx - 1, 'y': starty},
                {'x': startx - 2, 'y': starty}]
```

-генерираме нов црв, понатаму имав проблеми со детекцијата на колизија на црвот и главната големина на екранот и не можев никако да ја напраам оваа точка, па затоа ја пракам лабораторискава без нејзе.

2. Додадете елементи што трепкаат (секој пат по три) на случајно избрани позиции со димензија од едно квадратче. Тие се појавуваат во времетраење од 5 секунди. Ако оригиналниот црв ги изеде, играчот добива дополнителни поени (по 2 за секој изеден елемент). Овие поени треба да се вклучат во конечниот резултат на начин што вие ќе го изберете. Треба да обезбедите објаснување за формулата што ја користите за пресметување на резултатот. Резултатот треба да се прикаже на екранот што се појавува кога играта ќе заврши.

-Најпрвин тука искористиве 3 нови објекти кои што ги правам на овој начин, со тоа што и иницијализирав променлива што ќе ги чува поените.

```
square1 = getRandomLocation()
square2 = getRandomLocation()
square3 = getRandomLocation()
squareScore = 0
```

-Потоа проверувам дали воопшто црвот ќе го собере елементот со својата глава за да додадеме 2 поени за истата активност

```

if wormCoords[HEAD]['x'] == square1['x'] and wormCoords[HEAD]['y'] == square1["y"]:
    square1 = getRandomLocation()
    squareScore += 2

if wormCoords[HEAD]['x'] == square2['x'] and wormCoords[HEAD]['y'] == square2["y"]:
    square2 = getRandomLocation()
    squareScore += 2

if wormCoords[HEAD]['x'] == square3['x'] and wormCoords[HEAD]['y'] == square3["y"]:
    square3 = getRandomLocation()
    squareScore += 2

```

-Потоа проверувам дали тајмерот има изминато 5 секунди за да ги пуштиме новите полиња кои што би донеле поени

```

if (round(pygame.time.get_ticks()/1000) % 5 == 0) or (round(pygame.time.get_ticks()/1000) % 6 == 0) \
    or (round(pygame.time.get_ticks()/1000) % 7 == 0):
    drawRandomObject(square1)
    drawRandomObject(square2)
    drawRandomObject(square3)

drawApple(apple)
drawScore(len(wormCoords) - 3 + squareScore)
pygame.display.update()
FPSLOCK.tick(FPS)

```

-Ги генерираме на 5,6,7 секунда и тој е нивниот животен циклус, односно според модул гледаме на кои временски интервали ќе се генерираат , а ја додаваме променливата на главниот скор, односно доколку немаме јадено јаболко само +2 поена ќе имаме од шарените форми.

-Создаваме 2 функции за цртање на новите објекти , и ги правиме со функцијата за вртење на RGB боите , односно правиме shuffle на истите.

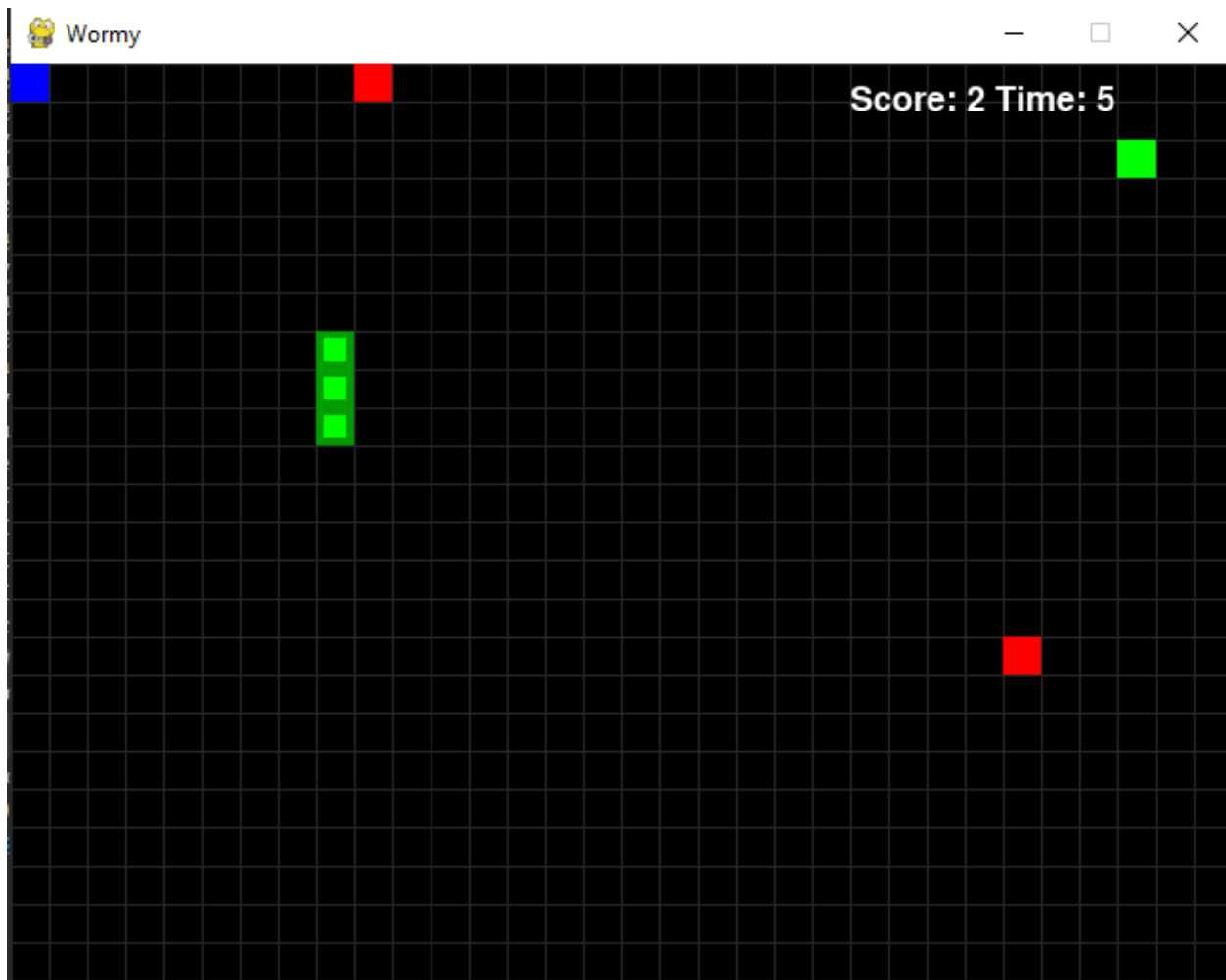
```

def drawRandomObject(coord):
    x = coord['x'] * CELL_SIZE
    y = coord['y'] * CELL_SIZE
    randomObject = pygame.Rect(x, y, CELL_SIZE, CELL_SIZE)
    pygame.draw.rect(DISPLAYSURF, random_color(), randomObject)

def random_color():
    rgb1=[255,0,0]
    random.shuffle(rgb1)
    return tuple(rgb1)

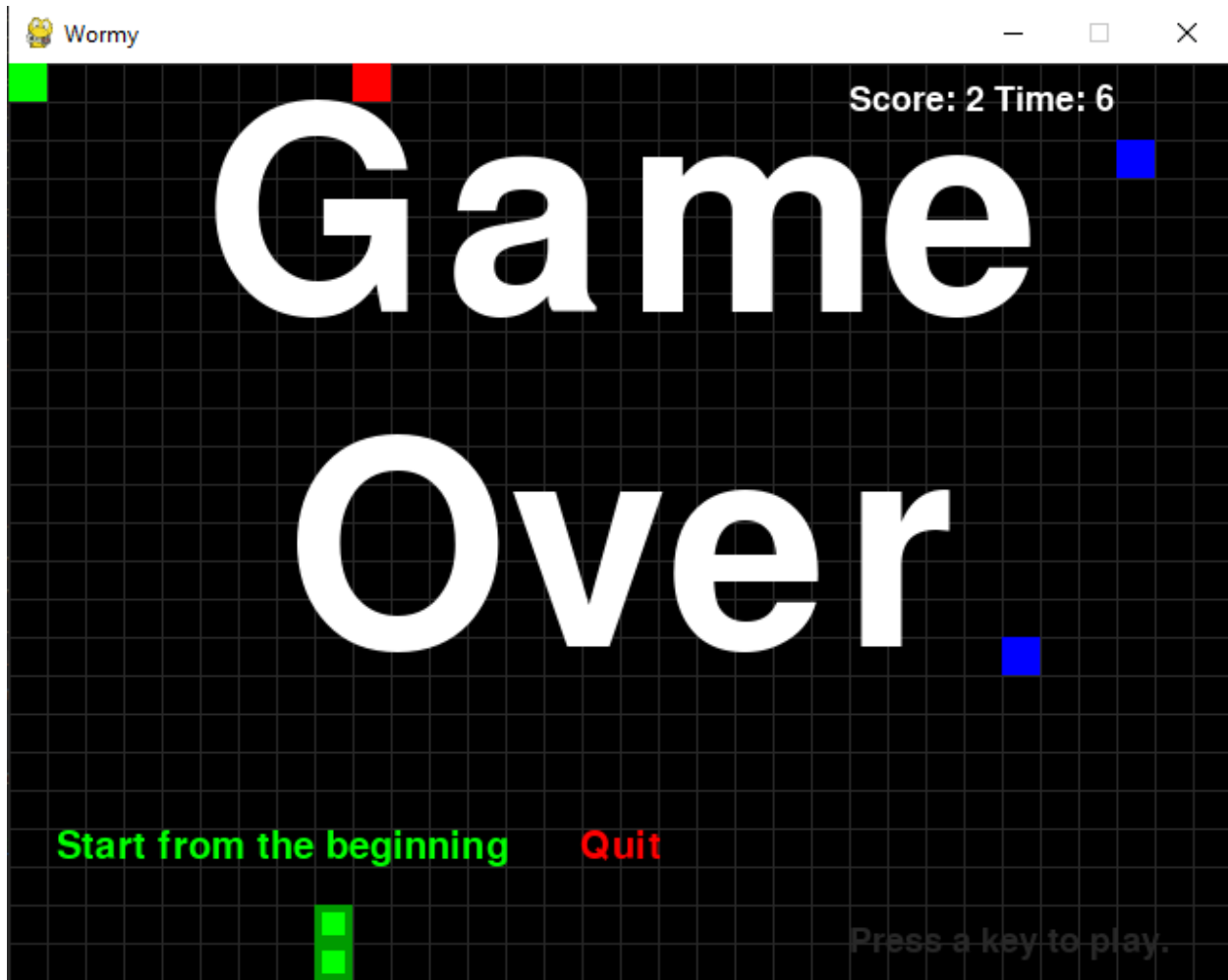
```

-Илустрирано тоа би изгледало вака



3. На екранот што се појавува кога играта ќе заврши треба да се додадат две копчиња, "Start from the beginning" и "Quit". Кога играчот ќе кликне на првото копче, играта треба да почне од почеток (без да се појави почетниот екран). Кога играчот ќе кликне на второто копче, треба да се исклучи играта.

-За ова најдобро би било првен илустрацијата , односно



```
def showGameOverScreen():
    gameOverFont = pygame.font.Font('freesansbold.ttf', 150)
    quitFont = pygame.font.Font('freesansbold.ttf', 20)
    gameSurf = gameOverFont.render('Game', True, WHITE)
    overSurf = gameOverFont.render('Over', True, WHITE)
    quitSurf = quitFont.render('Quit', True, RED)
    restartSurf = quitFont.render('Start from the beginning', True, GREEN)
    gameRect = gameSurf.get_rect()
    overRect = overSurf.get_rect()
    quitRect = quitSurf.get_rect()
    restartRect = restartSurf.get_rect()
    gameRect.midtop = (WINDOWWIDTH / 2, 10)
    overRect.midtop = (WINDOWWIDTH / 2, gameRect.height + 10 + 25)
    quitRect.midtop = (WINDOWWIDTH / 2, 400)
    restartRect.midtop = (quitRect.width + 100, 400)
    DISPLAYSURF.blit(gameSurf, gameRect)
    DISPLAYSURF.blit(overSurf, overRect)
    DISPLAYSURF.blit(quitSurf, quitRect)
    DISPLAYSURF.blit(restartSurf, restartRect)
    drawPressKeyMsg()
    pygame.display.update()
    pygame.time.wait(500)
    checkForKeyPress() # clear out any key presses in the event queue
    while True:
```

-Тука генерираме целосно нов фонд за новите 2 копчиња, и ги цртаме им генерираме нивни должина и големина а и воедно каде ќе се наоѓаат на екранот.

-Со функциите `get_pressed` да е `TRUE` и функцијата за нивниот `rect`, `collidepoint`, проверуваме дали нашето глумче е во просторот на тој квадрат со `mouse.get_pos()` , и на тој начин доколку глумчето ни е на `quit` копчето и глумчето е притиснато се терминира играта, доколку е на `restart` копчето ,односно `Start from the beginning` и е притиснато копчето почнува нова игра.

```
while True:
    if pygame.mouse.get_pressed()[0] and quitRect.collidepoint(pygame.mouse.get_pos()):
        pygame.quit()
        quit()
    if pygame.mouse.get_pressed()[0] and restartRect.collidepoint(pygame.mouse.get_pos()):
        pygame.event.get()
        return
    if checkForKeyPress():
        pygame.event.get() # clear event queue
        return
```