

Лабораториска вежба 4

Wormy

Изработил: Стефан Бојациев – 181114

Барање 1:

По 20 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.

```
BLUE = (135, 203, 243) #Барање 1: боја на вториот црв
DARKBLUE = (0, 0, 255) #Барање 1: боја на вториот црв
```

Ги иницијализираме двете бои на вториот црв.

```
#Барање 1
startSecondX = random.randint(5, CELLWIDTH - 6)
startSecondY = random.randint(5, CELLHEIGHT - 6)

wormCoords = [{'x': startX, 'y': startY},
               {'x': startX - 1, 'y': startY},
               {'x': startX - 2, 'y': startY}]
direction = random.choice([UP, DOWN, RIGHT, LEFT])
```

Генерираме координати за појавување на првиот црв каде воедно ќе имаме 3 негови почетни сегменти. Додаваме случаен правец на движење.

```
#Барање 1
secondWormCoords = [{'x': startSecondX, 'y': startSecondY},
                    {'x': startSecondX - 1, 'y': startSecondY},
                    {'x': startSecondX - 2, 'y': startSecondY}]
secondWormDirection = random.choice([UP, DOWN, RIGHT, LEFT])
```

По примерот на генерирање на почетните координати на првиот црв, истото го правиме и за вториот, додавајќи случаен правец на движење.

```
#Барање 1
#Сите можности за правец се валидни освен спротивниот, затоа што не може црвот да се сврти 180 степени
if secondWormDirection == RIGHT:
    secondWormDirection = random.choice([UP, DOWN, RIGHT])
elif secondWormDirection == LEFT:
    secondWormDirection = random.choice([UP, DOWN, LEFT])
elif secondWormDirection == DOWN:
    secondWormDirection = random.choice([LEFT, RIGHT, DOWN])
elif secondWormDirection == UP:
    secondWormDirection = random.choice([RIGHT, UP, LEFT])

if secondWormDirection == UP:
    newSecondHead = {'x': secondWormCoords[HEAD]['x'], 'y': secondWormCoords[HEAD]['y'] - 1}
elif secondWormDirection == DOWN:
    newSecondHead = {'x': secondWormCoords[HEAD]['x'], 'y': secondWormCoords[HEAD]['y'] + 1}
elif secondWormDirection == LEFT:
    newSecondHead = {'x': secondWormCoords[HEAD]['x'] - 1, 'y': secondWormCoords[HEAD]['y']}
elif secondWormDirection == RIGHT:
    newSecondHead = {'x': secondWormCoords[HEAD]['x'] + 1, 'y': secondWormCoords[HEAD]['y']}
```

Додаваме услови и начин на движење, условот е дека црвот секогаш може да се движи во кој било од трите правци (настрана од спротивниот на тековниот, затоа што црвот не може да се сврти за 180 степени).

```
#Барање 1
if ((time.time() - start) > 20):
    if secondWormCoords[HEAD]['x'] == wormCoords[HEAD]['x'] and secondWormCoords[HEAD]['y'] == wormCoords[HEAD]['y']:
        pass
    else:
        del secondWormCoords[-1]
```

После 20 секунди од почетокот на играта (spawn time на вториот црв) почнуваме со неговите услови за растење. Доколку координатите за главите на двата црва се совпаднаат вториот црв расте, во спротива одиме со стандардно одземање на опашката на црвот (наменета за движење).

```
#Барање 1
if time.time() - start > 20:
    secondWormCoords.insert(__index: 0, newSecondHead)
    drawSecondWorm(secondWormCoords)
```

Услов за појавување на вториот црв. Броиме 20 секунди од почетокот на играта.

```
#Барање 1
1 usage
def drawSecondWorm(secondWormCoords):
    for coord in secondWormCoords:
        x = coord['x'] * CELLSIZE
        y = coord['y'] * CELLSIZE
        wormSegmentRect = pygame.Rect(x, y, CELLSIZE, CELLSIZE)
        pygame.draw.rect(DISPLAYSURF, DARKBLUE, wormSegmentRect)
        wormInnerSegmentRect = pygame.Rect(x + 4, y + 4, CELLSIZE - 8, CELLSIZE - 8)
        pygame.draw.rect(DISPLAYSURF, BLUE, wormInnerSegmentRect)
```

Го генерираме вториот црв со нова функција која што во принцип е идентична на функцијата за генерирање на првиот црв, бидејќи во основа двете функции треба да ја имаат истата функционалност (исцртување на двата црва во соодветната боја)

Барање 2:

Додадете два елемента што трепкаат (секој пат по три) на случајно избрани позиции со димензија од едно квадратче. Едниот се појавуваат во времетраење од 5 секунди на секои 5 секунди, а другиот еднократно, во времетраење од 7 секунди. Ако оригиналниот црв ги изеде, играчот добива дополнителни поени (по 3 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.

```
LIGHTPURPLE = (224, 176, 255) #Барање 2: боја на нов објект
DARKPURPLE = (145, 95, 109) #Барање 2: боја на нов објект
YELLOW = (255, 255, 0) #Барање 2: боја на нов објект
ORANGE = (248, 157, 12) #Барање 2: боја на нов објект
```

Ги поставуваме инцијалните RGB вредности за сите нови бои кои треба да ги употребиме во играта.

```
#Барање 2
yellowLemon = getRandomLocation()
purpleGrape = getRandomLocation()
extraPoints = 0 #нова променлива за броење на екстра поени
```

Поставуваме нова променлива extraPoints која ги брои поените додадени при собирањето на новите објекти.

```
#Барање 1
if wormCoords[HEAD]['x'] == purpleGrape['x'] and wormCoords[HEAD]['y'] == purpleGrape['y']:
    purpleGrape = {'x': -1, 'y': -1} #Го поместуваме објектот надвор од полето за игра
    extraPoints += 3 #Додаваме по 3 поени
if wormCoords[HEAD]['x'] == yellowLemon['x'] and wormCoords[HEAD]['y'] == yellowLemon['y']:
    yellowLemon = {'x': -1, 'y': -1} #Го поместуваме објектот надвор од полето за игра
    extraPoints += 3 #Додаваме по 3 поени
```

Доколку црвот-играч допре било кое од двете екстра овошја што сме ги додаде (грозје и лимон) објектот го преместуваме надвор од полето за игра и додаваме 3 поени на играчот (extraPoints).

```
#Барање 2
baseLength = len(wormCoords)-3
score = baseLength + extraPoints
drawScore(score)
```

Овде додаваме променлива и логика за растење на score променливата. Ја земаме должината на црвот во било кој момент, од неа одземаме 3 (со цел да не ги рачунаме првите 3 сегменти кои ги имаме од старт на играта) и ги додаваме екстра поените.

```
#Барање 2
if time.time() - start % 10 > 5:
    drawPurpleGrape(purpleGrape)
#Барање 2
if time.time() - start < 7:
    drawYellowLemon(yellowLemon)
pygame.display.update()
FPSLOCK.tick(FPS)
```

Ги додаваме условите за генерирање на другите два објекти како што е опишано во самото барање на вежбата.

```
#Барање 2
scoreFont = pygame.font.Font( name: 'freesansbold.ttf', size: 20)
scoreText = 'Score: ' + str(score)
scoreSurf = scoreFont.render(scoreText, antialias: True, YELLOW)
scoreRect = scoreSurf.get_rect()
scoreRect.midtop = (WINDOWWIDTH / 2, 500)
DISPLAYSURF.blit(scoreSurf, scoreRect)
```

Од тука го генерираме текстот за Score и ја додаваме неговата тековна вредност. Со помош на дефинираните променливи во овој дел можеме да експриментираме и да ги придвижуваме сите дела кои ги печатиме на екранот.

```

#Барање 2
1 usage
def drawYellowLemon(coord):
    color1 = YELLOW
    color2 = ORANGE
    x = coord['x'] * CELLSIZE
    y = coord['y'] * CELLSIZE
    for i in range(3):
        color1, color2 = color2, color1
        lemonYellowRect = pygame.Rect(x, y, CELLSIZE, CELLSIZE)
        pygame.draw.rect(DISPLAYSURF, color1, lemonYellowRect)
        pygame.display.update()
        FPSCLOCK.tick(FPS)

1 usage
def drawPurpleGrape(coord):
    color1 = LIGHTPURPLE
    color2 = DARKPURPLE
    x = coord['x'] * CELLSIZE
    y = coord['y'] * CELLSIZE
    for i in range(3):
        color1, color2 = color2, color1
        grapePurpleRect = pygame.Rect(x, y, CELLSIZE, CELLSIZE)
        pygame.draw.rect(DISPLAYSURF, color1, grapePurpleRect)
        pygame.display.update()
        FPSCLOCK.tick(FPS)

```

Двете функции за генерирање на дополнителните објекти. За да се обезбеди ефект на трепкање во еден for циклус го менуваме редоследот на двете одбрани бои (color1, color2).

Барање 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.

```

#Барање 3
for event in pygame.event.get():
    if event.type == MOUSEBUTTONDOWN:
        mouse_pos = pygame.mouse.get_pos()
        if startRect.collidepoint(mouse_pos):
            return
        if quitRect.collidepoint(mouse_pos):
            terminate()

```

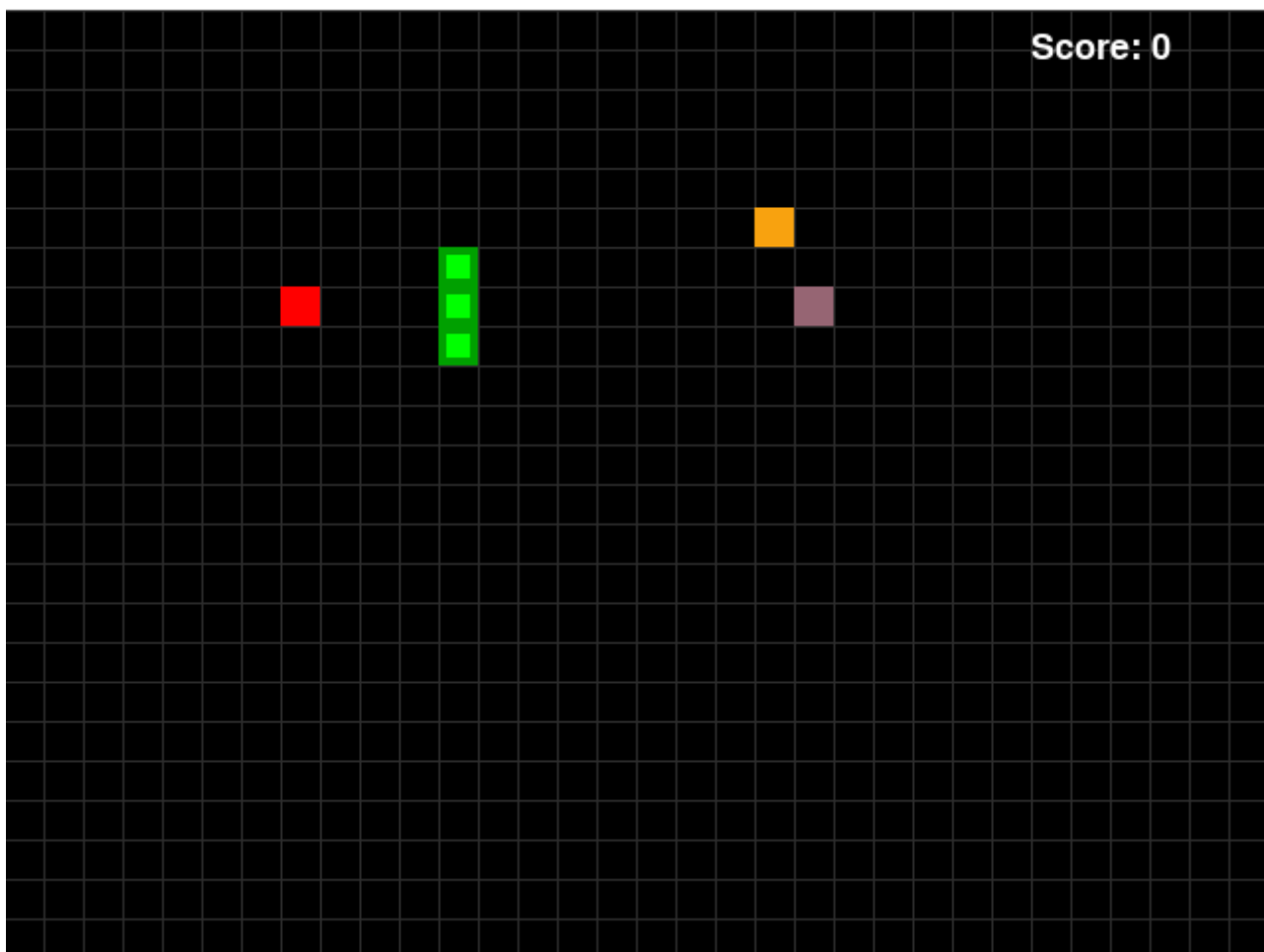
Ја додаваме функционалноста за двете копчиња (start и quit) со тоа што при start се почнува од почеток, додека quit ја гаси целата програма.

```
#Барање 3
buttonFont = pygame.font.Font( name: 'freesansbold.ttf', size: 35)
startText = 'Start from the beggining'
quitText = 'Quit'
startSurf = buttonFont.render(startText, antialias: True, GREEN, DARKGRAY)
quitSurf = buttonFont.render(quitText, antialias: True, GREEN, DARKGRAY)
startRect = startSurf.get_rect()
quitRect = quitSurf.get_rect()
startRect.midtop = (WINDOWWIDTH / 2, 400)
quitRect.midtop = (WINDOWWIDTH / 2, 450)
DISPLAYSURF.blit(startSurf, startRect)
DISPLAYSURF.blit(quitSurf, quitRect)
```

Со овој дел од кодот ги конфигурираме истите копчиња и додаваме променливи за самиот текст што би ни помогнале за полесна промена доколку е потребна.

🐛 Wormy

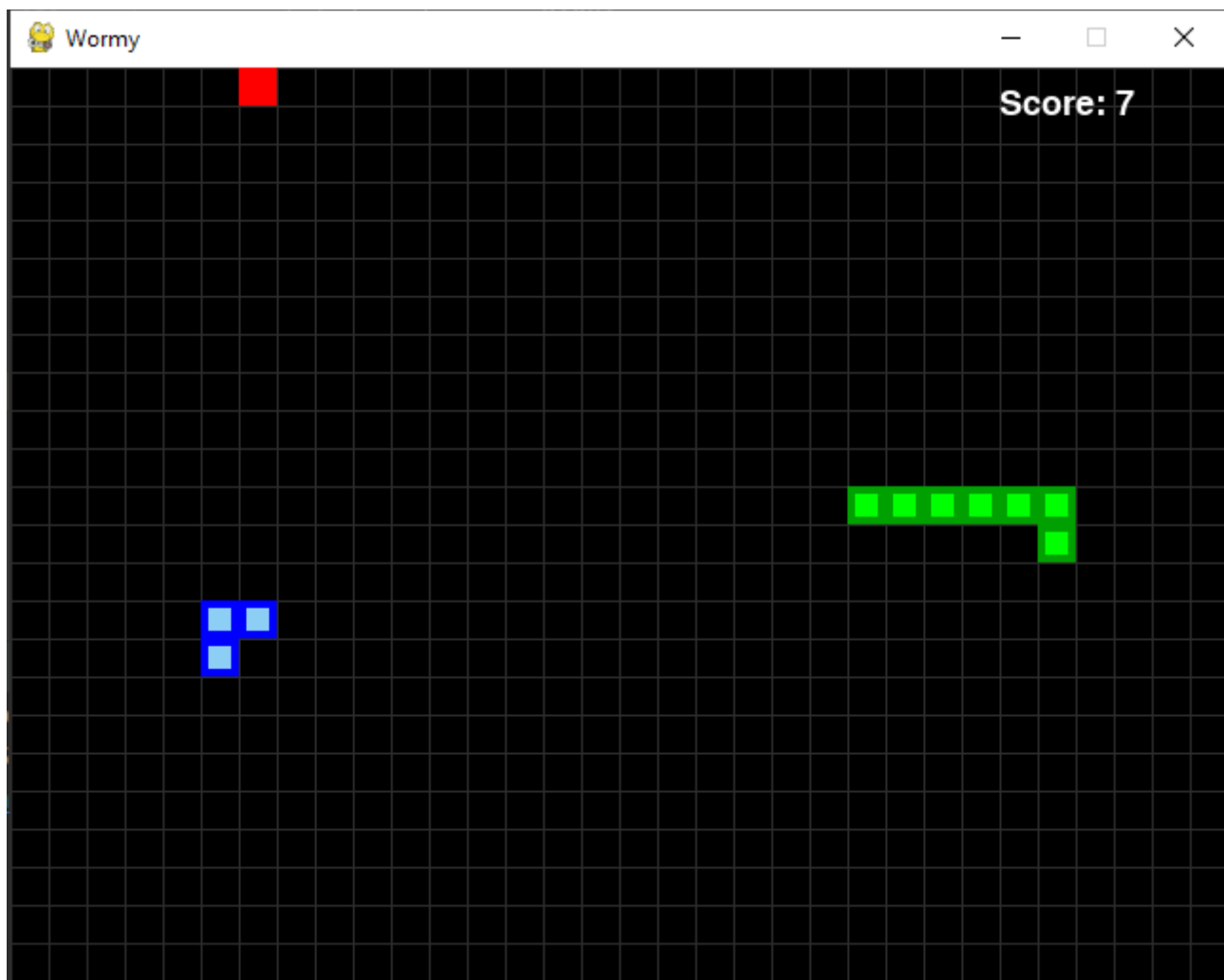
— □ ×



Ова е изгледот на играта на самиот почеток



Ова е изгледот на Game Over screen



Вториот црв (темно и светло сив) во време на прво појавување во играта.