

1. 20 seconds after starting the game add a second worm. The original worm should avoid the second one. If the original worm touches the second worm with the head, its body grows for one segments. If the second worm touches the original, than its body grows for one segment. The movement of the second worm is random.

```
import random, pygame, sys, time
```

- We import time so we can measure the time passed and current time

```
BLUE      = (137, 207, 240) #1st req: color for the 2nd worm  
DARKBLUE  = ( 0, 0, 255) #1st req: color for the 2nd worm
```

- We set values for the colors we are going to use for the new worm that we need to create

```
#1st req: setting start coordinates for 2nd worm  
startx1 = random.randint(5, CELLWIDTH - 6)  
starty1 = random.randint(5, CELLHEIGHT - 6)
```

```
#1st req: setting coordinates and direction for 2nd worm  
wormCoords1 = [{'x': startx1, 'y': starty1},  
                {'x': startx1 - 1, 'y': starty1},  
                {'x': startx1 - 2, 'y': starty1}]  
direction1 = LEFT
```

- We set random starting coordinates for the new worm startx1 and startx2, and then we give it coordinates and direction in which it will move wormCoords1 and direction1. The second worm moves randomly, changing its directions at intervals

```
#1st req: random moving direction

if direction1 == LEFT:
    direction1 = random.choice([UP, DOWN, LEFT])
elif direction1 == RIGHT:
    direction1 = random.choice([UP, DOWN, RIGHT])
elif direction1 == UP:
    direction1 = random.choice([LEFT, RIGHT, UP])
elif direction1 == DOWN:
    direction1 = random.choice([RIGHT, DOWN, LEFT])

if direction1 == UP:
    newHead1 = {'x': wormCoords1[HEAD]['x'], 'y': wormCoords1[HEAD]['y'] - 1}
elif direction1 == DOWN:
    newHead1 = {'x': wormCoords1[HEAD]['x'], 'y': wormCoords1[HEAD]['y'] + 1}
elif direction1 == LEFT:
    newHead1 = {'x': wormCoords1[HEAD]['x'] - 1, 'y': wormCoords1[HEAD]['y']}
elif direction1 == RIGHT:
    newHead1 = {'x': wormCoords1[HEAD]['x'] + 1, 'y': wormCoords1[HEAD]['y']}
```

```
#1st req: function for drawing the second worm
def drawWorm1(wormCoords1):
    for coord in wormCoords1:
        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)
```

- Finally the function we use to draw the second worm, we give it color, size etc

```
#1st req: draw the second worm
if ((time.time() - start) > 20):
    wormCoords1.insert(0, newHead1)
    drawWorm1(wormCoords1)
```

- The worm is supposed to show up 20sec into the game

```
#1st req: if second worm touches the first worm add plus one segment to the second worm
if ((time.time() - start) > 20):
    if wormCoords1[HEAD]['x'] == wormCoords[HEAD]['x'] and wormCoords1[HEAD]['y'] == wormCoords[HEAD]['y']:
        pass
    else:
        del wormCoords1[-1]
```

- We check if the second worm touches the first worm after 20 seconds into starting the game, and if it does it gets a new segment added to it
2. Add 2 elements that blink (3 each time) at randomly selected positions with dimensions of 1 cell. The first one appears every 5 seconds, and it lasts 5 seconds. The second one appears only once, lasting 7 seconds. If the original worm eats any of these elements, the player gets additional points (3 for each eaten element). These points should be included in the end result in a way that you will choose. You need to provide an explanation for the formula that you will use for calculating the result. The result should be shown on the screen after the game ends.

```
YELLOW = (255, 255, 0) #2nd req: color for the new object

#2nd req: start two new object in random places
yellowApple = getRandomLocation()
blueApple = getRandomLocation()
extraPoints = 0 #new variable for adding extra points to the score
```

- We introduce two new objects that start at a random location

```
#2nd req: appear for 5 seconds every 5 seconds
if ((time.time() - start) % 10 > 5):
    drawBlueApple(blueApple)
#2nd req: appear once for 7 seconds
if ((time.time() - start) < 7):
    drawYellowApple(yellowApple)
```

- We set the time so the blue apple can appear for 5 seconds every 5 seconds and the yellow one once for 7 seconds

```
#2nd req: add score to the gameOver screen
scoreFont = pygame.font.Font('freesansbold.ttf', 16)
scoreSurf = scoreFont.render('Score: ' + str(score), True, WHITE)
scoreRect = scoreSurf.get_rect()
scoreRect.midtop = (WINDOWWIDTH / 2, 450)
DISPLAYSURF.blit(scoreSurf, scoreRect)
```

- We add the score to the gameover screen as requested and we also set the font for it.

```
score = (len(wormCoords) - 3) + extraPoints #2nd req: new formula for score
drawScore(score)
```

- This is where we calculate the score for each element the worm eats we need to add 3 additional points so we use this formula

```
#2nd req: draw two new objects
def drawYellowApple(coord):
    color1 = YELLOW
    color2 = GREEN
    x = coord['x'] * CELL_SIZE
    y = coord['y'] * CELL_SIZE
    for i in range(3):
        color1, color2 = color2, color1
        appleYellowRect = pygame.Rect(x, y, CELL_SIZE, CELL_SIZE)
        pygame.draw.rect(DISPLAYSURF, color1, appleYellowRect)
        pygame.display.update()
```

```
def drawBlueApple(coord):
    color1 = BLUE
    color2 = DARKBLUE
    x = coord['x'] * CELL_SIZE
    y = coord['y'] * CELL_SIZE
    for i in range(3):
        color1, color2 = color2, color1
        appleBlueRect = pygame.Rect(x, y, CELL_SIZE, CELL_SIZE)
        pygame.draw.rect(DISPLAYSURF, color1, appleBlueRect)
        pygame.display.update()
```

- The functions used for drawing the two new elements, their colors and the appearance of these elements is controlled based on the time passed.
3. Add two buttons "Start from the beginning" and "Quit" to the game over screen. When the user clicks the first button, the game should start over (without showing the starting screen). When the player clicks the second button, the game should terminate.

```
#3rd req: add start and quit buttons
buttonFont = pygame.font.SysFont('comicsansms', 25)
startSurf = buttonFont.render('Start from the beggining', True, BLUE, DARKGRAY)
quitSurf = buttonFont.render('Quit', True, RED, DARKGRAY)
startRect = startSurf.get_rect()
quitRect = quitSurf.get_rect()
startRect.midtop = (WINDOWWIDTH / 2, 350)
quitRect.midtop = (WINDOWWIDTH / 2, 400)
DISPLAYSURF.blit(startSurf, startRect)
DISPLAYSURF.blit(quitSurf, quitRect)
```

- After the game ends the player needs to see a game over screen with the score. We create two buttons "Start from the beginning" in blue and "Quit" in red with a dark gray background.

```
#3rd req: start and quit buttons func
for event in pygame.event.get():
    if event.type == MOUSEBUTTONDOWN:
        if startRect.collidepoint(pygame.mouse.get_pos()):
            return
        if quitRect.collidepoint(pygame.mouse.get_pos()):
            terminate()
```

- The showGameOverScreen() function displays the game over screen with the score, the new buttons are implemented with mouse click event handling