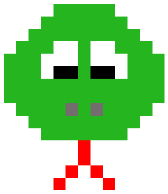
JCC Summer
Game Design
TechEm Studios



Instructions: The activity is broken up into four separate parts. Each part consists of crucial components that make up the snake game we will be programming in Python language. Once you have filled in each part, you can try to run the program and test out your game!

Pay very close attention to the syntax:

( ) [ ] { } , “ ‘ : ; .

Part One:

```
# Create variables that define our colors we will need
26 black = (0, 0, 0)
27 white = (255, 255, 255)
28 red = (255, 0, 0)
29 green = (0, 180, 0)

# Define variables that will define the level width and height
32 display_width = 800
33 display_height = 600
```

Part Two:

```
# Adding the functionality to keep track of the score
105 def score(score):
106       text = smallfont.render("Score: " + str(score), True, black)
107       gameDisplay.blit(text, [0,0])
```

Part Three:

```
146    gameDisplay.fill(white)

148    gameDisplay.blit(alive, [260,0])
```

```
message_to_screen("Welcome NEXTUP",
        green,
        -80,
        "large")
message_to_screen("The objective of the game is to eat red apples",
        black,
        -30,
        "small")
message_to_screen("The more apples you eat, the longer you get",
        black,
        10,
        "small")
message_to_screen("If you run into yourself or the edges, you lose!",
        black,
        50,
        "small")

message_to_screen("Press C to play, P to pause, or Q to quit.",
        black,
        180,
        "small")
pygame.display.update()
clock.tick(15)
```

Part Four:

```
while gameOver == True:
    gameDisplay.fill(white)
    message_to_screen("Game over",
        red,
        y_displace = -50,
        size = "large")

    gameDisplay.blit(dead, [280,20])

    message_to_screen("press C to play again or Q to quit",
        black,
        50,
        size = "medium")
    pygame.display.update()
```