Yes, I know you all have played the Snake Game and definitely, you never wanted to lose. As kids, we all loved looking for cheats in order to never see the “Game Over” message but as techies, I know you would want to make this ‘Snake’ dance to your beats. This is what I will be showing you all in this article on Snake Game in [Python](https://www.edureka.co/blog/python-basics/).  
Before moving on, let’s have a quick look at all the sub-bits that build the Snake Game in Python:

1. [Installing Pygame](https://www.edureka.co/blog/snake-game-with-pygame/#install)
2. [Create the Screen](https://www.edureka.co/blog/snake-game-with-pygame/#screen)
3. [Create the Snake](https://www.edureka.co/blog/snake-game-with-pygame/#createthesnake)
4. [Moving the Snake](https://www.edureka.co/blog/snake-game-with-pygame/#move)
5. [Game Over when Snake hits the boundaries](https://www.edureka.co/blog/snake-game-with-pygame/#boundaries)
6. [Adding the Food](https://www.edureka.co/blog/snake-game-with-pygame/#food)
7. [Increasing the Length of the Snake](https://www.edureka.co/blog/snake-game-with-pygame/#increaselength)
8. [Displaying the Score](https://www.edureka.co/blog/snake-game-with-pygame/#score)

Installing Pygame:

The first thing you will need to do in order to create games using Pygame is to install it on your systems. To do that, you can simply use the following command:

pip install pygame

Once that is done, just import Pygame and start off with your game development. Before moving on, take a look at the Pygame functions that have been used in this Snake Game along with their descriptions.

|  |  |
| --- | --- |
| Function | Description |
| init() | Initializes all of the imported Pygame modules (returns a tuple indicating success and failure of initializations) |
| display.set\_mode() | Takes a tuple or a list as its parameter to create a surface (tuple preferred) |
| update() | Updates the screen |
| quit() | Used to uninitialize everything |
| set\_caption() | Will set the caption text on the top of the display screen |
| event.get() | Returns list of all events |
| Surface.fill() | Will fill the surface with a solid color |
| time.Clock() | Helps track time time |
| font.SysFont() | Will create a Pygame font from the System font resources |

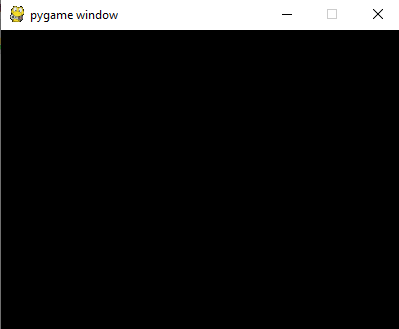
Create the Screen:

To create the screen using Pygame, you will need to make use of the display.set\_mode() [function](https://www.edureka.co/blog/python-functions). Also, you will have to make use of the init()  and the quit() methods to initialize and uninitialize everything at the start and the end of the code. The update() method is used to update any changes made to the screen. There is another method i.e flip() that works similarly to the update() function. The difference is that the update() method updates only the changes that are made (however, if no parameters are passed, updates the complete screen) but the flip() method redoes the complete screen dis again.

CODE:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | import pygame  pygame.init()  =pygame.display.set\_mode((400,300))  pygame.display.update()  pygame.quit()  quit() |

OUTPUT:

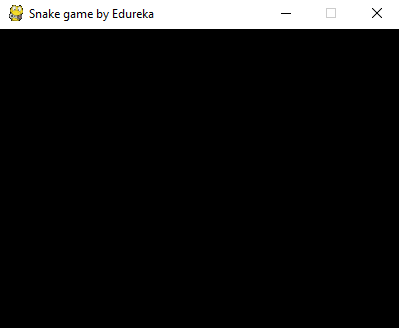


But when you run this code, the screen will appear, but it will immediately close as well. To fix that, you should make use of a game loop using the [while loop](https://www.edureka.co/blog/while-loop-in-python/) before I actually quit the game as follows:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | import pygame  pygame.init()  dis=pygame.display.set\_mode((400,300))  pygame.display.update()  pygame.display.set\_caption('Snake game by Edureka')  game\_over=False  while not game\_over:      for event in pygame.event.get():          print(event)   #prints out all the actions that take place on the screen    pygame.quit()  quit() |

When you run this code, you will see that the screen that you saw earlier does not quit and also, it returns all the actions that take place over it. I have done that using the event.get() function. Also, I have named the screen as “Snake Game by Edureka” using the display.set\_caption() function.

OUTPUT:



Now, you have a screen to play your Snake Game, but when you try to click on the close button, the screen does not close. This is because you have not specified that your screen should exit when you hit that close button. To do that, Pygame provides an event called “QUIT” and it should be used as follows:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | import pygame  pygame.init()  dis=pygame.display.set\_mode((400,300))  pygame.display.update()  pygame.display.set\_caption('Snake game by Edureka')  game\_over=False  while not game\_over:      for event in pygame.event.get():          if event.type==pygame.QUIT:              game\_over=True    pygame.quit()  quit() |

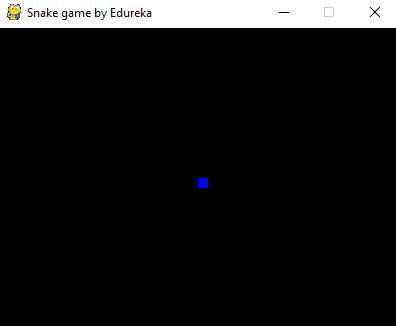
So now your screen is all set. The next part is to draw our snake on the screen which is covered in the following topic.

Create the Snake:

To create the snake, I will first initialize a few color variables in order to color the snake, food, screen, etc. The color scheme used in Pygame is RGB i.e “Red Green Blue”. In case you set all these to 0’s, the color will be black and all 255’s will be white. So our snake will actually be a rectangle. To draw rectangles in Pygame, you can make use of a function called draw.rect() which will help yo draw the rectangle with the desired color and size.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | import pygame  pygame.init()  dis=pygame.display.set\_mode((400,300))    pygame.display.set\_caption('Snake game by Edureka')    blue=(0,0,255)  red=(255,0,0)    game\_over=False  while not game\_over:      for event in pygame.event.get():          if event.type==pygame.QUIT:              game\_over=True      pygame.draw.rect(dis,blue,[200,150,10,10])      pygame.display.update()  pygame.quit()  quit() |

OUTPUT:



As you can see, the snakehead is created as a blue rectangle. The next step is to get your snake moving.

Moving the Snake:

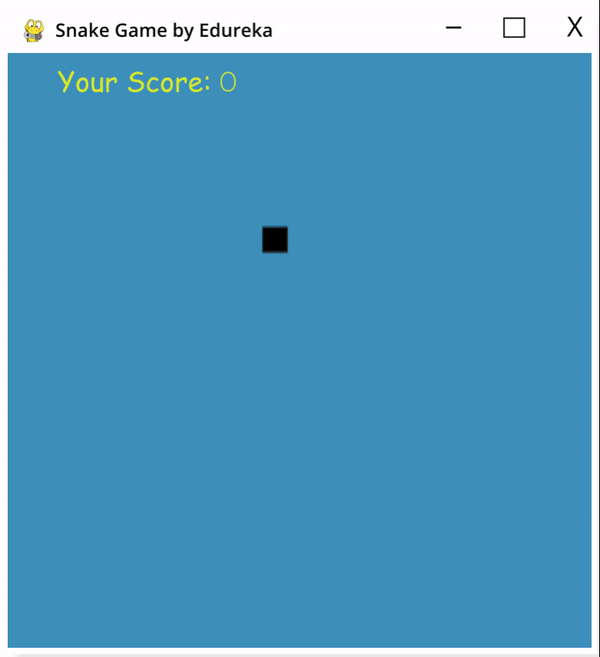
Next

To move the snake, you will need to use the key events present in the KEYDOWN class of Pygame. The events that are used over here are, K\_UP, K\_DOWN, K\_LEFT, and K\_RIGHT to make the snake move up, down, left and right respectively. Also, the display screen is changed from the default black to white using the fill() method.

I have created new variables x1\_change and y1\_change in order to hold the updating values of the x and y coordinates.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | import pygame    pygame.init()    white = (255, 255, 255)  black = (0, 0, 0)  red = (255, 0, 0)    dis = pygame.display.set\_mode((800, 600))  pygame.display.set\_caption('Snake Game by Edureka')    game\_over = False    x1 = 300  y1 = 300    x1\_change = 0  y1\_change = 0    clock = pygame.time.Clock()    while not game\_over:      for event in pygame.event.get():          if event.type == pygame.QUIT:              game\_over = True          if event.type == pygame.KEYDOWN:              if event.key == pygame.K\_LEFT:                  x1\_change = -10                  y1\_change = 0              elif event.key == pygame.K\_RIGHT:                  x1\_change = 10                  y1\_change = 0              elif event.key == pygame.K\_UP:                  y1\_change = -10                  x1\_change = 0              elif event.key == pygame.K\_DOWN:                  y1\_change = 10                  x1\_change = 0        x1 += x1\_change      y1 += y1\_change      dis.fill(white)      pygame.draw.rect(dis, black, [x1, y1, 10, 10])        pygame.display.update()        clock.tick(30)    pygame.quit()  quit() |

OUTPUT:

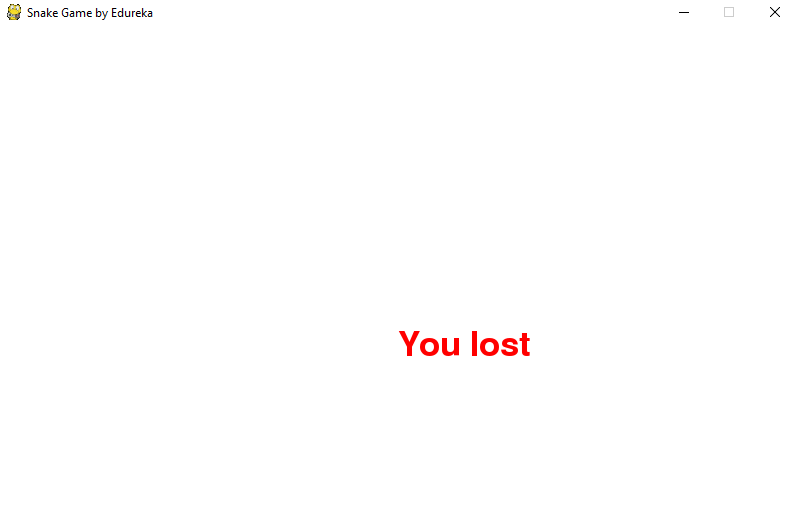


Game Over when Snake hits the boundaries:

In this snake game, if the player hits the boundaries of the screen, then he loses. To specify that, I have made use of an ‘if’ statement that defines the limits for the x and y coordinates of the snake to be less than or equal to that of the screen. Also, make a not over here that I have removed the hardcodes and used variables instead so that it becomes easy in case you want to make any changes to the game later on.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68 | import pygame  import time  pygame.init()    white = (255, 255, 255)  black = (0, 0, 0)  red = (255, 0, 0)    dis\_width = 800  dis\_height  = 600  dis = pygame.display.set\_mode((dis\_width, dis\_width))  pygame.display.set\_caption('Snake Game by Edureka')    game\_over = False    x1 = dis\_width/2  y1 = dis\_height/2    snake\_block=10    x1\_change = 0  y1\_change = 0    clock = pygame.time.Clock()  snake\_speed=30    font\_style = pygame.font.SysFont(None, 50)    def message(msg,color):      mesg = font\_style.render(msg, True, color)      dis.blit(mesg, [dis\_width/2, dis\_height/2])    while not game\_over:      for event in pygame.event.get():          if event.type == pygame.QUIT:              game\_over = True          if event.type == pygame.KEYDOWN:              if event.key == pygame.K\_LEFT:                  x1\_change = -snake\_block                  y1\_change = 0              elif event.key == pygame.K\_RIGHT:                  x1\_change = snake\_block                  y1\_change = 0              elif event.key == pygame.K\_UP:                  y1\_change = -snake\_block                  x1\_change = 0              elif event.key == pygame.K\_DOWN:                  y1\_change = snake\_block                  x1\_change = 0        if x1 >= dis\_width or x1 < 0 or y1 >= dis\_height or y1 < 0:          game\_over = True        x1 += x1\_change      y1 += y1\_change      dis.fill(white)      pygame.draw.rect(dis, black, [x1, y1, snake\_block, snake\_block])        pygame.display.update()        clock.tick(snake\_speed)    message("You lost",red)  pygame.display.update()  time.sleep(2)    pygame.quit()  quit() |

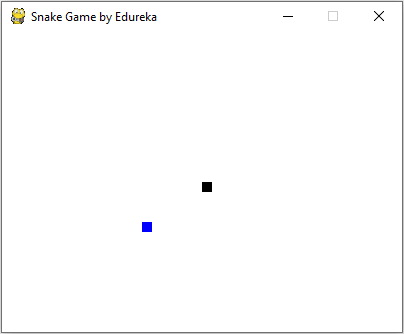
OUTPUT:

Adding the Food:

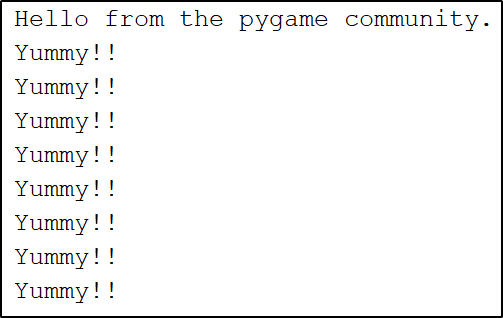
Here, I will be adding some food for the snake and when the snake crosses over that food, I will have a message saying “Yummy!!”. Also, I will be making a small change wherein I will include the options to quit the game or to play again when the player loses.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94 | import pygame  import time  import random    pygame.init()    white = (255, 255, 255)  black = (0, 0, 0)  red = (255, 0, 0)  blue = (0, 0, 255)    dis\_width = 800  dis\_height = 600    dis = pygame.display.set\_mode((dis\_width, dis\_height))  pygame.display.set\_caption('Snake Game by Edureka')    clock = pygame.time.Clock()    snake\_block = 10  snake\_speed = 30    font\_style = pygame.font.SysFont(None, 30)      def message(msg, color):      mesg = font\_style.render(msg, True, color)      dis.blit(mesg, [dis\_width/3, dis\_height/3])      def gameLoop():  # creating a function      game\_over = False      game\_close = False        x1 = dis\_width / 2      y1 = dis\_height / 2        x1\_change = 0      y1\_change = 0        foodx = round(random.randrange(0, dis\_width - snake\_block) / 10.0) \* 10.0      foody = round(random.randrange(0, dis\_width - snake\_block) / 10.0) \* 10.0        while not game\_over:            while game\_close == True:              dis.fill(white)              message("You Lost! Press Q-Quit or C-Play Again", red)              pygame.display.update()                for event in pygame.event.get():                  if event.type == pygame.KEYDOWN:                      if event.key == pygame.K\_q:                          game\_over = True                          game\_close = False                      if event.key == pygame.K\_c:                          gameLoop()            for event in pygame.event.get():              if event.type == pygame.QUIT:                  game\_over = True              if event.type == pygame.KEYDOWN:                  if event.key == pygame.K\_LEFT:                      x1\_change = -snake\_block                      y1\_change = 0                  elif event.key == pygame.K\_RIGHT:                      x1\_change = snake\_block                      y1\_change = 0                  elif event.key == pygame.K\_UP:                      y1\_change = -snake\_block                      x1\_change = 0                  elif event.key == pygame.K\_DOWN:                      y1\_change = snake\_block                      x1\_change = 0            if x1 >= dis\_width or x1 < 0 or y1 >= dis\_height or y1 < 0:              game\_close = True            x1 += x1\_change          y1 += y1\_change          dis.fill(white)          pygame.draw.rect(dis, blue, [foodx, foody, snake\_block, snake\_block])          pygame.draw.rect(dis, black, [x1, y1, snake\_block, snake\_block])          pygame.display.update()            if x1 == foodx and y1 == foody:              print("Yummy!!")          clock.tick(snake\_speed)        pygame.quit()      quit()      gameLoop() |

OUTPUT:



Terminal:

Increasing the Length of the Snake:

The following code will increase the size of our sake when it eats the food. Also, if the snake collides with his own body, the game is over and you ill see a message as “You Lost! Press Q-Quit or C-Play Again“. The length of the snake is basically contained in a list and the initial size that is specified in the following code is one block.

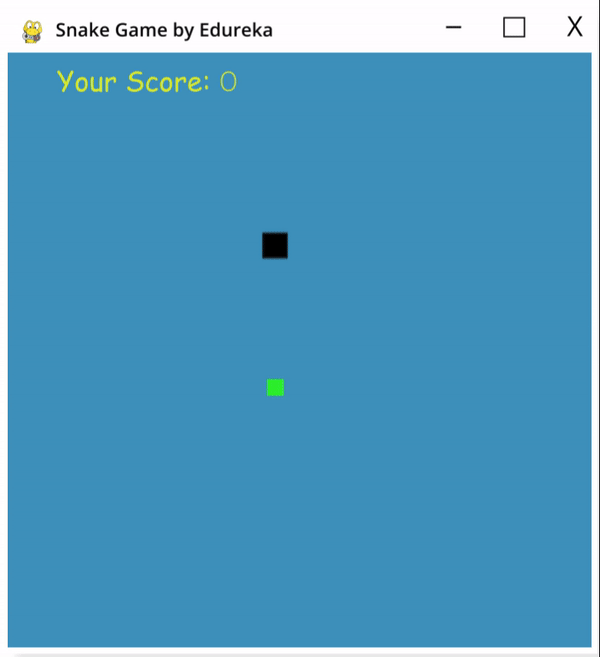
[[](https://www.edureka.co/data-science-python-certification-course)](https://www.edureka.co/data-science-python-certification-course" \t "_blank)

[Python Certification Training for Data Science](https://www.edureka.co/data-science-python-certification-course" \t "_blank)

[Weekday / Weekend BatchesSee Batch Details](https://www.edureka.co/data-science-python-certification-course" \t "_blank)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120 | import pygame  import time  import random    pygame.init()    white = (255, 255, 255)  yellow = (255, 255, 102)  black = (0, 0, 0)  red = (213, 50, 80)  green = (0, 255, 0)  blue = (50, 153, 213)    dis\_width = 600  dis\_height = 400    dis = pygame.display.set\_mode((dis\_width, dis\_height))  pygame.display.set\_caption('Snake Game by Edureka')    clock = pygame.time.Clock()    snake\_block = 10  snake\_speed = 15    font\_style = pygame.font.SysFont("bahnschrift", 25)  score\_font = pygame.font.SysFont("comicsansms", 35)    def our\_snake(snake\_block, snake\_list):      for x in snake\_list:          pygame.draw.rect(dis, black, [x[0], x[1], snake\_block, snake\_block])      def message(msg, color):      mesg = font\_style.render(msg, True, color)      dis.blit(mesg, [dis\_width / 6, dis\_height / 3])      def gameLoop():      game\_over = False      game\_close = False        x1 = dis\_width / 2      y1 = dis\_height / 2        x1\_change = 0      y1\_change = 0        snake\_List = []      Length\_of\_snake = 1        foodx = round(random.randrange(0, dis\_width - snake\_block) / 10.0) \* 10.0      foody = round(random.randrange(0, dis\_height - snake\_block) / 10.0) \* 10.0        while not game\_over:            while game\_close == True:              dis.fill(blue)              message("You Lost! Press C-Play Again or Q-Quit", red)                pygame.display.update()                for event in pygame.event.get():                  if event.type == pygame.KEYDOWN:                      if event.key == pygame.K\_q:                          game\_over = True                          game\_close = False                      if event.key == pygame.K\_c:                          gameLoop()            for event in pygame.event.get():              if event.type == pygame.QUIT:                  game\_over = True              if event.type == pygame.KEYDOWN:                  if event.key == pygame.K\_LEFT:                      x1\_change = -snake\_block                      y1\_change = 0                  elif event.key == pygame.K\_RIGHT:                      x1\_change = snake\_block                      y1\_change = 0                  elif event.key == pygame.K\_UP:                      y1\_change = -snake\_block                      x1\_change = 0                  elif event.key == pygame.K\_DOWN:                      y1\_change = snake\_block                      x1\_change = 0            if x1 >= dis\_width or x1 < 0 or y1 >= dis\_height or y1 < 0:              game\_close = True          x1 += x1\_change          y1 += y1\_change          dis.fill(blue)          pygame.draw.rect(dis, green, [foodx, foody, snake\_block, snake\_block])          snake\_Head = []          snake\_Head.append(x1)          snake\_Head.append(y1)          snake\_List.append(snake\_Head)          if len(snake\_List) > Length\_of\_snake:              del snake\_List[0]            for x in snake\_List[:-1]:              if x == snake\_Head:                  game\_close = True            our\_snake(snake\_block, snake\_List)              pygame.display.update()            if x1 == foodx and y1 == foody:              foodx = round(random.randrange(0, dis\_width - snake\_block) / 10.0) \* 10.0              foody = round(random.randrange(0, dis\_height - snake\_block) / 10.0) \* 10.0              Length\_of\_snake += 1            clock.tick(snake\_speed)        pygame.quit()      quit()      gameLoop() |

OUTPUT:

Displaying the Score:

Last but definitely not the least, you will need to display the score of the player. To do this, I have created a new function as “Your\_score”. This function will display the length of the snake subtracted by 1 because that is the initial size of the snake.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127 | import pygame  import time  import random    pygame.init()    white = (255, 255, 255)  yellow = (255, 255, 102)  black = (0, 0, 0)  red = (213, 50, 80)  green = (0, 255, 0)  blue = (50, 153, 213)    dis\_width = 600  dis\_height = 400    dis = pygame.display.set\_mode((dis\_width, dis\_height))  pygame.display.set\_caption('Snake Game by Edureka')    clock = pygame.time.Clock()    snake\_block = 10  snake\_speed = 15    font\_style = pygame.font.SysFont("bahnschrift", 25)  score\_font = pygame.font.SysFont("comicsansms", 35)      def Your\_score(score):      value = score\_font.render("Your Score: " + str(score), True, yellow)      dis.blit(value, [0, 0])        def our\_snake(snake\_block, snake\_list):      for x in snake\_list:          pygame.draw.rect(dis, black, [x[0], x[1], snake\_block, snake\_block])      def message(msg, color):      mesg = font\_style.render(msg, True, color)      dis.blit(mesg, [dis\_width / 6, dis\_height / 3])      def gameLoop():      game\_over = False      game\_close = False        x1 = dis\_width / 2      y1 = dis\_height / 2        x1\_change = 0      y1\_change = 0        snake\_List = []      Length\_of\_snake = 1        foodx = round(random.randrange(0, dis\_width - snake\_block) / 10.0) \* 10.0      foody = round(random.randrange(0, dis\_height - snake\_block) / 10.0) \* 10.0        while not game\_over:            while game\_close == True:              dis.fill(blue)              message("You Lost! Press C-Play Again or Q-Quit", red)              Your\_score(Length\_of\_snake - 1)              pygame.display.update()                for event in pygame.event.get():                  if event.type == pygame.KEYDOWN:                      if event.key == pygame.K\_q:                          game\_over = True                          game\_close = False                      if event.key == pygame.K\_c:                          gameLoop()            for event in pygame.event.get():              if event.type == pygame.QUIT:                  game\_over = True              if event.type == pygame.KEYDOWN:                  if event.key == pygame.K\_LEFT:                      x1\_change = -snake\_block                      y1\_change = 0                  elif event.key == pygame.K\_RIGHT:                      x1\_change = snake\_block                      y1\_change = 0                  elif event.key == pygame.K\_UP:                      y1\_change = -snake\_block                      x1\_change = 0                  elif event.key == pygame.K\_DOWN:                      y1\_change = snake\_block                      x1\_change = 0            if x1 >= dis\_width or x1 < 0 or y1 >= dis\_height or y1 < 0:              game\_close = True          x1 += x1\_change          y1 += y1\_change          dis.fill(blue)          pygame.draw.rect(dis, green, [foodx, foody, snake\_block, snake\_block])          snake\_Head = []          snake\_Head.append(x1)          snake\_Head.append(y1)          snake\_List.append(snake\_Head)          if len(snake\_List) > Length\_of\_snake:              del snake\_List[0]            for x in snake\_List[:-1]:              if x == snake\_Head:                  game\_close = True            our\_snake(snake\_block, snake\_List)          Your\_score(Length\_of\_snake - 1)            pygame.display.update()            if x1 == foodx and y1 == foody:              foodx = round(random.randrange(0, dis\_width - snake\_block) / 10.0) \* 10.0              foody = round(random.randrange(0, dis\_height - snake\_block) / 10.0) \* 10.0              Length\_of\_snake += 1            clock.tick(snake\_speed)        pygame.quit()      quit()      gameLoop() |

OUTPUT:

