```
import pygame
import time
import random
pygame.init()
# Define colors
WHITE = (255, 255, 255)
BLACK = (0, 0, 0)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)
# Game display dimensions
WIDTH = 900
HEIGHT = 500
# Snake parameters
SNAKE_SIZE = 10
SNAKE_SPEED = 15
# Setup the display
game_display = pygame.display.set_mode((WIDTH, HEIGHT), pygame.RESIZABLE)
pygame.display.set_caption('Snake Game - Press Enter to Start')
# Clock object for controlling the frame rate
clock = pygame.time.Clock()
# Font styles
font_style = pygame.font.SysFont("bahnschrift", 25)
score_font = pygame.font.SysFont("comicsansms", 35)
class Node:
    """A node in a linked list representing a segment of the snake."""
    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.next = None
class LinkedList:
    """A linked list to represent the snake's body."""
    def __init__(self):
        self.head = None
        self.tail = None
    def add_head(self, x, y):
        """Add a new segment to the head of the snake."""
        new_node = Node(x, y)
        if not self.head:
            self.head = self.tail = new_node
        else:
            new_node.next = self.head # type: ignore
            self.head = new_node
    def remove_tail(self):
```

```
        """Remove the last segment (tail) of the snake."""
        if not self.head:
            return
        if self.head == self.tail:
            self.head = self.tail = None
        else:
            current = self.head
            while current.next != self.tail: # type: ignore
                current = current.next # type: ignore
            self.tail = current
            self.tail.next = None # type: ignore
    def contains(self, x, y):
        """Check if any segment of the snake has the given coordinates (x, y)."""
        current = self.head
        while current:
            if current.x == x and current.y == y:
                return True
            current = current.next
        return False
    def get_positions(self):
        """Return a list of all segments' (x, y) coordinates."""
        positions = []
        current = self.head
        while current:
            positions.append((current.x, current.y))
            current = current.next
        return positions
    def draw(self, surface):
        """Draw the entire snake on the screen."""
        current = self.head
        while current:
            pygame.draw.rect(surface, BLACK, [current.x, current.y, SNAKE_SIZE,
SNAKE_SIZE])
            current = current.next
def display_score(score):
    """Displays the current score."""
    value = score_font.render("Score: " + str(score), True, BLUE)
    # Clear previous score area
    pygame.draw.rect(game_display, BLACK, [0, 0, 150, 50])
    game_display.blit(value, [0, 0])
def our_snake(snake_block, snake_list):
    """Draws the snake using a list of segments."""
    for x in snake_list:
```

```
        pygame.draw.rect(game_display, BLACK, [x[0], x[1], snake_block,
snake_block])
def message(msg, color):
    """Displays a message in the center of the screen."""
    mesg = font_style.render(msg, True, color)
    game_display.blit(mesg, [WIDTH / 6, HEIGHT / 3])
def game_loop():
    """Main game loop."""
    game_over = False
    game_close = False
    start_game = False
    # Initial position of the snake
    x1 = WIDTH / 2
    y1 = HEIGHT / 2
    # Movement coordinates
    x1_change = 0
    y1_change = 0
    # Snake body
    snake_list = []
    snake_length = 1
    # Initial food position
    foodx = round(random.randrange(0, WIDTH - SNAKE_SIZE) / 10.0) * 10.0
    foody = round(random.randrange(0, HEIGHT - SNAKE_SIZE) / 10.0) * 10.0
    # Start screen message
    game_display.fill(BLACK)
    message("Press Enter to Start the Game", WHITE)
    pygame.display.update()
    while not start_game:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                start_game = True
                game_over = True
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_RETURN:
                    start_game = True
    while not game_over:
        while game_close:
            game_display.fill(BLACK)
            message("You Lost! Press P to Play Again or Q to Quit", RED)
            display_score(snake_length - 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:
                game_loop()
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        game_over = True
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_w or event.key == pygame.K_UP:
            x1_change = 0
            y1_change = -SNAKE_SIZE
        elif event.key == pygame.K_a or event.key == pygame.K_LEFT:
            x1_change = -SNAKE_SIZE
            y1_change = 0
        elif event.key == pygame.K_s or event.key == pygame.K_DOWN:
            x1_change = 0
            y1_change = SNAKE_SIZE
        elif event.key == pygame.K_d or event.key == pygame.K_RIGHT:
            x1_change = SNAKE_SIZE
            y1_change = 0
# Snake position update
x1 += x1_change
y1 += y1_change
# Collision with boundaries
if x1 >= WIDTH or x1 < 0 or y1 >= HEIGHT or y1 < 0:
    game_close = True
# Fill the screen background
game_display.fill(GREEN)
# Draw the food
pygame.draw.rect(game_display, RED, [foodx, foody, SNAKE_SIZE, SNAKE_SIZE])
# Snake growing mechanism
snake_head = [x1, y1]
snake_list.append(snake_head)
if len(snake_list) > snake_length:
    del snake_list[0]
# Check collision with itself
for segment in snake_list[:-1]:
    if segment == snake_head:
        game_close = True
# Draw the snake
our_snake(SNAKE_SIZE, snake_list)
# Display the live score
display_score(snake_length - 1)
pygame.display.update()
```

```
        # Snake eats the food
        if x1 == foodx and y1 == foody:
            foodx = round(random.randrange(0, WIDTH - SNAKE_SIZE) / 10.0) * 10.0
            foody = round(random.randrange(0, HEIGHT - SNAKE_SIZE) / 10.0) * 10.0
            snake_length += 1
        # Control the frame rate
        clock.tick(SNAKE_SPEED)
    # Quit the game
    pygame.quit()
    quit()
# Run the game loop
game_loop()
```