```
Import pygame
Import random

Pygame.init()

Clock = pygame.time.Clock()
Fps = 60

# Game window
Tile_size = 50
Cols = 20
Margin = 100
Screen_width = tile_size * cols
Screen_height = (tile_size * cols) + margin

Screen = pygame.display.set_mode((screen_width, screen_height))
Pygame.display.set_caption('Endless Runner')

# Load images
Bg_img = pygame.image.load(r'C:\Users\sheeb\OneDrive\Desktop\Documents\sky.jpg')
Bg_img = pygame.transform.scale(bg_img, (screen_width, screen_height – margin))
Blob_img = pygame.image.load(r'C:\Users\sheeb\OneDrive\Desktop\Documents\blob.jpg')
Blob_img = pygame.transform.scale(blob_img, (tile_size, tile_size))
Coin_img = pygame.image.load(r'C:\Users\sheeb\OneDrive\Desktop\Documents\coin.jpg')
Man_img = pygame.image.load(r'C:\Users\sheeb\OneDrive\Desktop\Documents\man.jpg')
Man_img = pygame.transform.scale(man_img, (tile_size, int(tile_size * 1.5)))
```

```
# Define game variables
Score = 0
Gravity = 1

# Define colors
White = (255, 255, 255)
Green = (144, 201, 120)

Font = pygame.font.SysFont('Futura', 24)

# Player class
Class Player():
  Def __init__(self, x, y):
    Self.rect = pygame.Rect(x, y, tile_size, int(tile_size * 1.5))
    Self.vel_y = 0
    Self.jump = False
    Self.in_air = True

  Def move(self):
    Dx = 0
    Dy = 0

    # Get key presses
    Key = pygame.key.get_pressed()
    If key[pygame.K_UP] and not self.jump and not self.in_air:
```

```
    Self.vel_y = -15
    Self.jump = True
    Self.in_air = True
If key[pygame.K_LEFT] and self.rect.left > 0:
    Dx = -5
If key[pygame.K_RIGHT] and self.rect.right < screen_width:
    Dx = 5

# Gravity
Self.vel_y += gravity
If self.vel_y > 10:
    Self.vel_y = 10
Dy += self.vel_y

# Update player position
Self.rect.x += dx
Self.rect.y += dy

# Collision with ground
If self.rect.bottom > screen_height – margin:
    Self.rect.bottom = screen_height – margin
    Dy = 0
    Self.in_air = False

# Draw the player
Screen.blit(man_img, self.rect.topleft)
```

```
        Return dx, dy


# Obstacle class
Class Obstacle(pygame.sprite.Sprite):
    Def __init__(self, x, y):
        Pygame.sprite.Sprite.__init__(self)
        Self.image = blob_img
        Self.rect = self.image.get_rect()
        Self.rect.x = x
        Self.rect.y = y


    Def update(self):
        Self.rect.x -= 5
        If self.rect.right < 0:
            Self.kill()


# Coin class
Class Coin(pygame.sprite.Sprite):
    Def __init__(self, x, y):
        Pygame.sprite.Sprite.__init__(self)
        Self.image = coin_img
        Self.rect = self.image.get_rect()
        Self.rect.x = x
        Self.rect.y = y
```

```
    Def update(self):
        Self.rect.x -= 5
        If self.rect.right < 0:
            Self.kill()

# Create sprite groups
Obstacle_group = pygame.sprite.Group()
Coin_group = pygame.sprite.Group()
Player = Player(tile_size * 2, screen_height – margin – tile_size * 1.5)

# Function to draw text
Def draw_text(text, font, text_col, x, y):
    Img = font.render(text, True, text_col)
    Screen.blit(img, (x, y))

# Main game loop
Run = True
While run:

    Clock.tick(fps)

    # Draw background
    Screen.fill(green)
    Screen.blit(bg_img, (0, 0))

    # Draw player
```

```
Dx, dy = player.move()

# Update and draw groups
Obstacle_group.update()
Obstacle_group.draw(screen)
Coin_group.update()
Coin_group.draw(screen)

# Check for collision with obstacles
If pygame.sprite.spritecollide(player, obstacle_group, False):
   # Reset player position upon collision
   Player.rect.x = tile_size * 2
   Player.rect.y = screen_height – margin – tile_size * 1.5

# Check for collision with coins
If pygame.sprite.spritecollide(player, coin_group, True):
   Score += 1

# Draw the score
Draw_text(f’Score: {score}’, font, white, tile_size, screen_height – 60)

# Generate new obstacles and coins
If random.randint(1, 100) == 1:
   Obstacle_group.add(Obstacle(screen_width, screen_height – margin – tile_size))
If random.randint(1, 30) == 1:
   Coin_group.add(Coin(screen_width, screen_height – margin – tile_size * 2))
```

```
    # Event handler
    For event in pygame.event.get():
        # Quit game
        If event.type == pygame.QUIT:
            Run = False

    # Update game display window
    Pygame.display.update()

Pygame.quit()
```