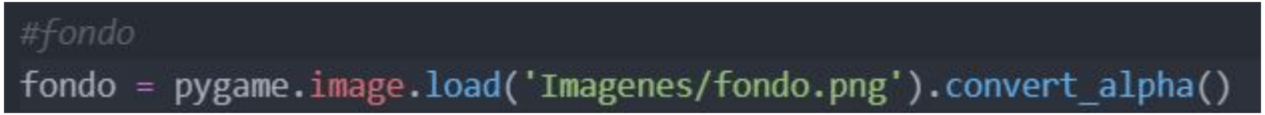
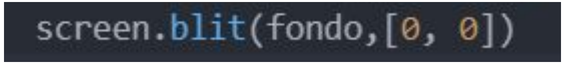
Snake game

Kevin Guaicha
Carlos Valladarez

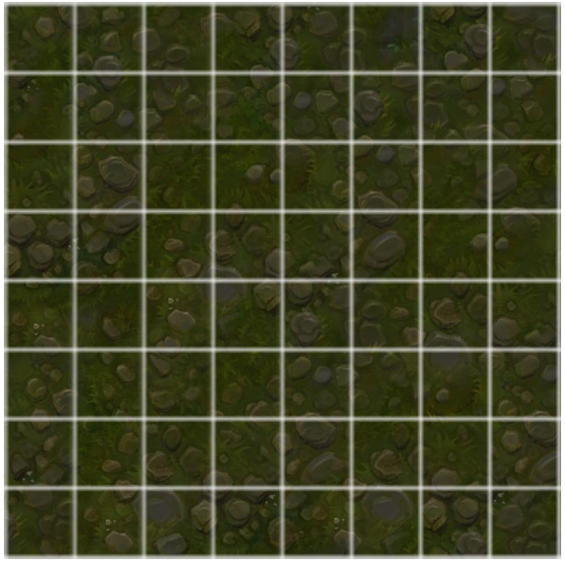
Mapa



```
#fondo
fondo = pygame.image.load('Imagenes/fondo.png').convert_alpha()
```

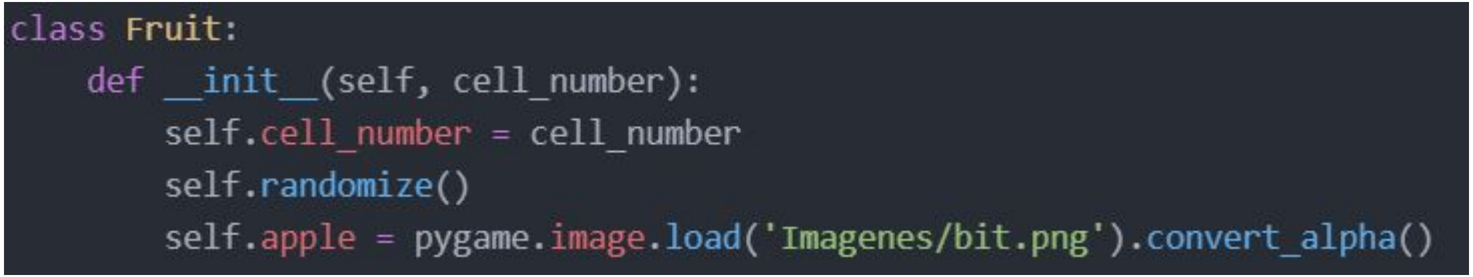


```
screen.blit(fondo,[0, 0])
```

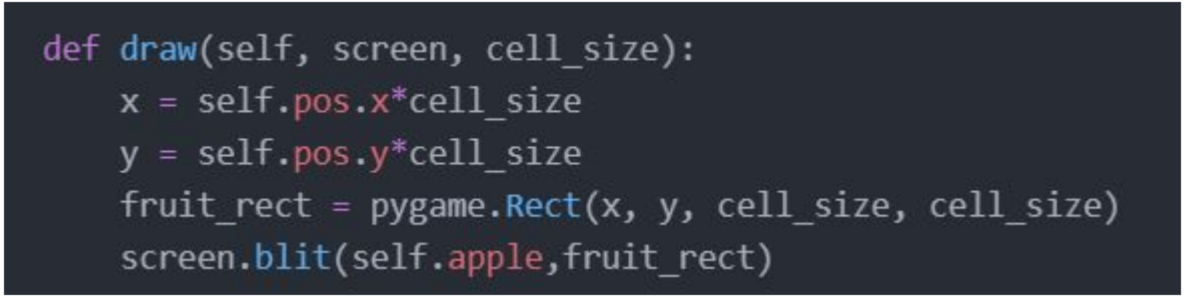


La pantalla está compuesta por 20* 20 cuadros que miden 40*40 píxeles.

Imagen de la comida

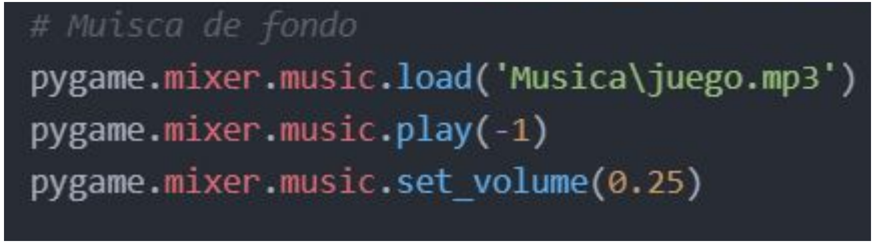


```python
class Fruit:
    def __init__(self, cell_number):
        self.cell_number = cell_number
        self.randomize()
        self.apple = pygame.image.load('Imagenes/bit.png').convert_alpha()
```



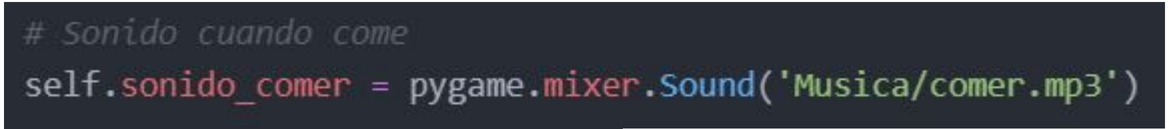
```python
def draw(self, screen, cell_size):
    x = self.pos.x*cell_size
    y = self.pos.y*cell_size
    fruit_rect = pygame.Rect(x, y, cell_size, cell_size)
    screen.blit(self.apple,fruit_rect)
```

Música de fondo

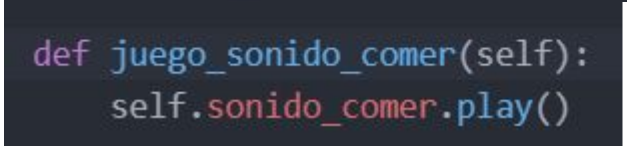


```python
# Muisca de fondo
pygame.mixer.music.load('Musica\juego.mp3')
pygame.mixer.music.play(-1)
pygame.mixer.music.set_volume(0.25)
```

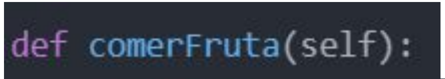
Sonido cuando come



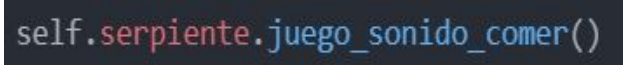
```python
# Sonido cuando come
self.sonido_comer = pygame.mixer.Sound('Musica/comer.mp3')
```



```python
def juego_sonido_comer(self):
    self.sonido_comer.play()
```

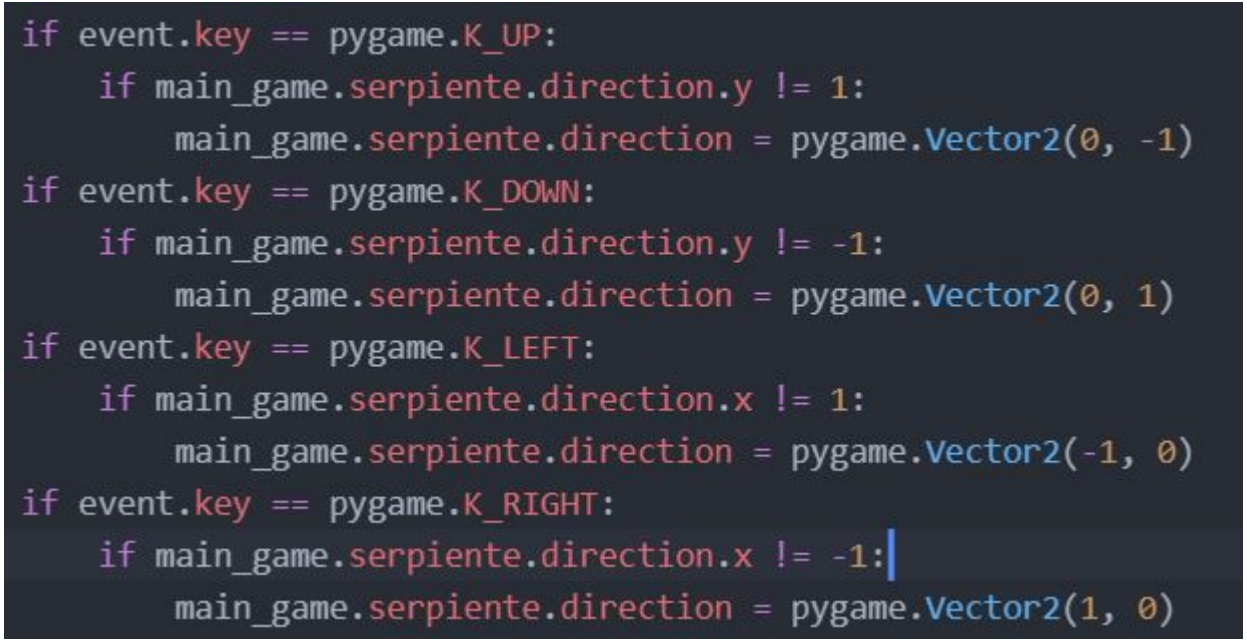


```python
def comerFruta(self):
```

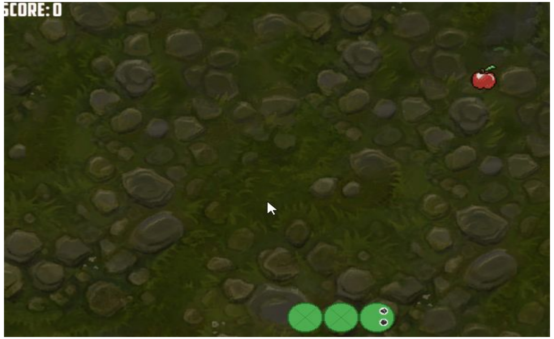


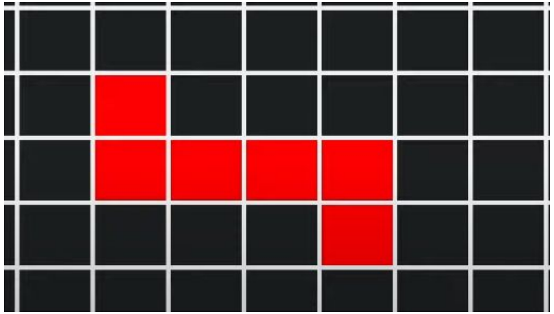
```python
self.serpiente.juego_sonido_comer()
```

Movimiento

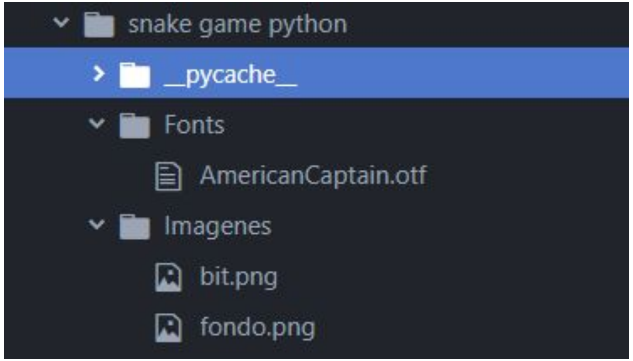


```python
if event.key == pygame.K_UP:
    if main_game.serpiente.direction.y != 1:
        main_game.serpiente.direction = pygame.Vector2(0, -1)
if event.key == pygame.K_DOWN:
    if main_game.serpiente.direction.y != -1:
        main_game.serpiente.direction = pygame.Vector2(0, 1)
if event.key == pygame.K_LEFT:
    if main_game.serpiente.direction.x != 1:
        main_game.serpiente.direction = pygame.Vector2(-1, 0)
if event.key == pygame.K_RIGHT:
    if main_game.serpiente.direction.x != -1:
        main_game.serpiente.direction = pygame.Vector2(1, 0)
```

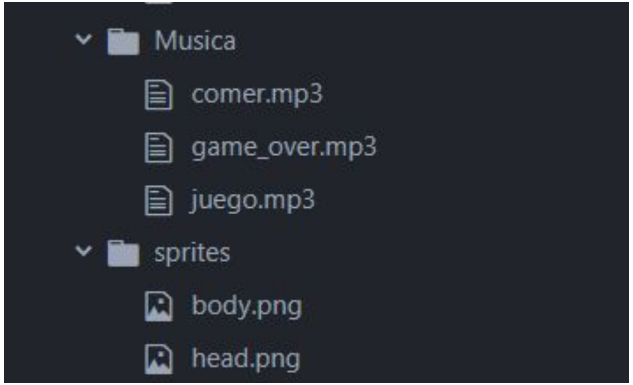




Recurso utilizados

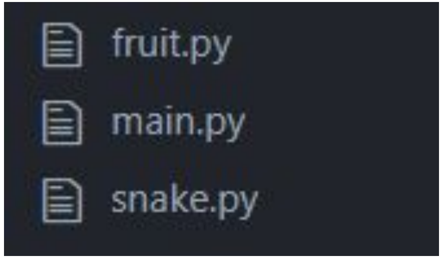


```
snake game python
  __pycache__
  Fonts
    AmericanCaptain.otf
  Imagenes
    bit.png
    fondo.png
```



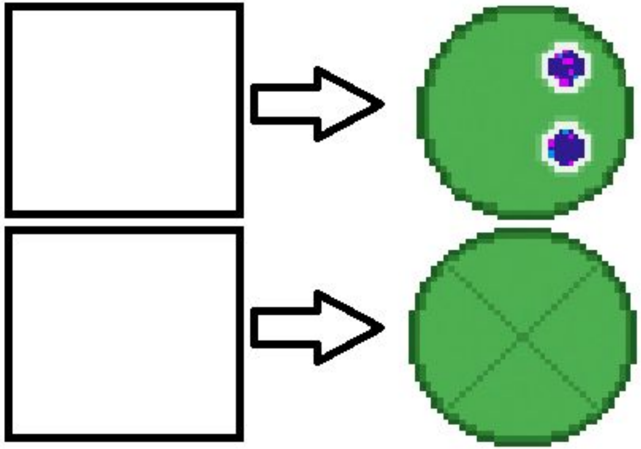
```
  Musica
    comer.mp3
    game_over.mp3
    juego.mp3
  sprites
    body.png
    head.png
```

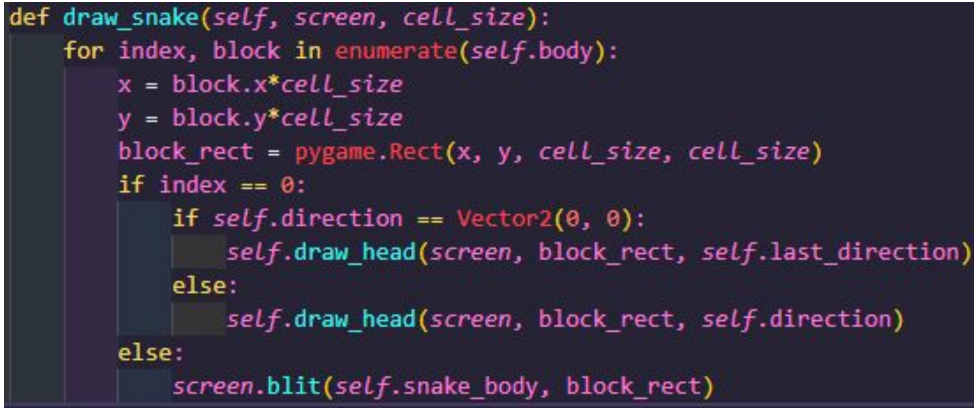
Clases



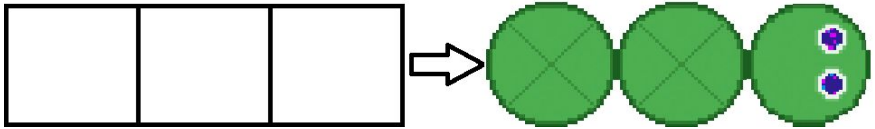
```
fruit.py
main.py
snake.py
```

Serpiente

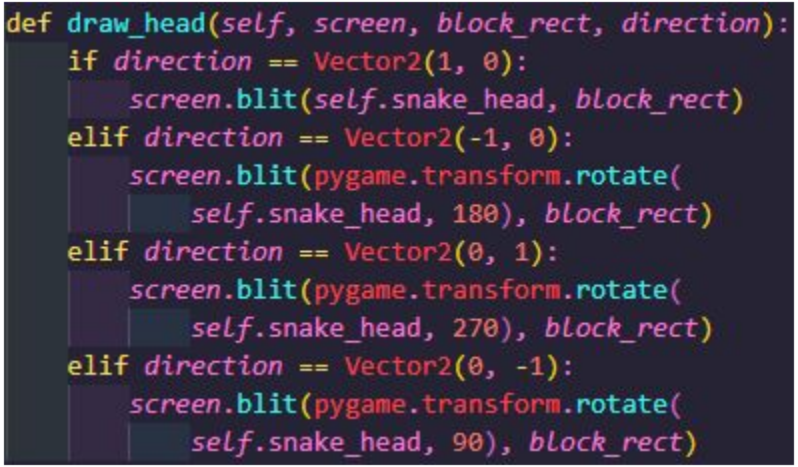




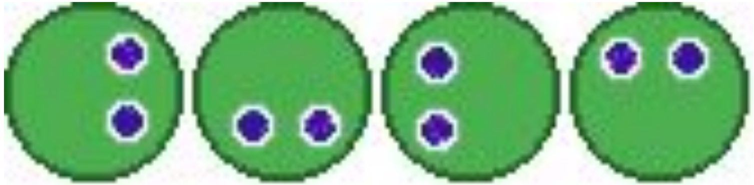
```python
def draw_snake(self, screen, cell_size):
    for index, block in enumerate(self.body):
        x = block.x*cell_size
        y = block.y*cell_size
        block_rect = pygame.Rect(x, y, cell_size, cell_size)
        if index == 0:
            if self.direction == Vector2(0, 0):
                self.draw_head(screen, block_rect, self.last_direction)
            else:
                self.draw_head(screen, block_rect, self.direction)
        else:
            screen.blit(self.snake_body, block_rect)
```



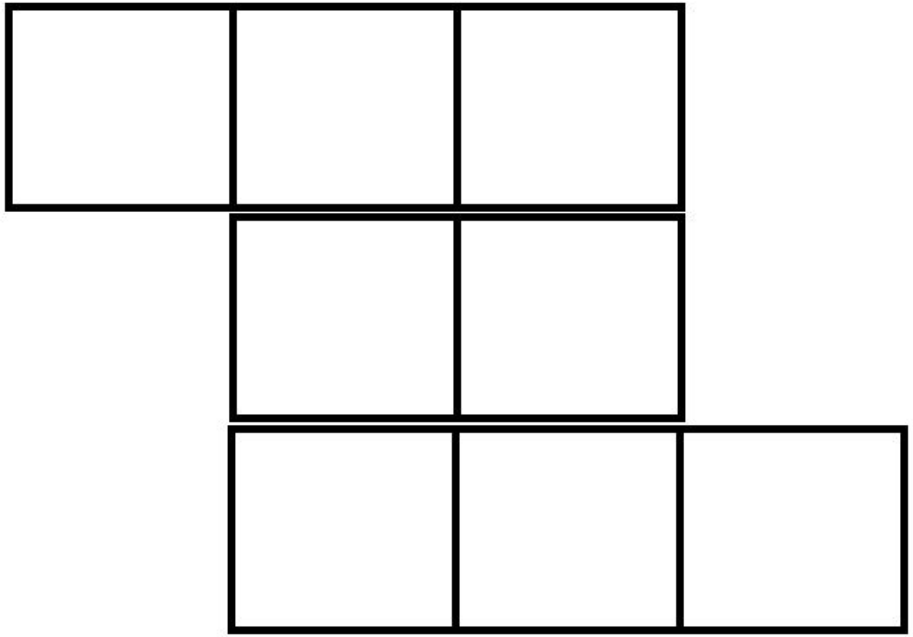
Serpiente

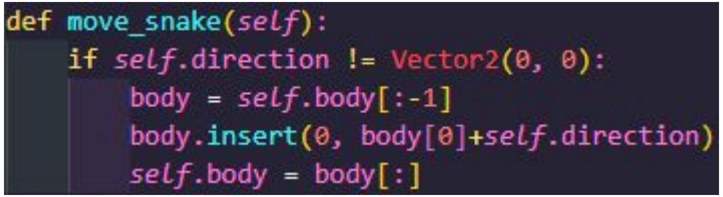


```python
def draw_head(self, screen, block_rect, direction):
    if direction == Vector2(1, 0):
        screen.blit(self.snake_head, block_rect)
    elif direction == Vector2(-1, 0):
        screen.blit(pygame.transform.rotate(
            self.snake_head, 180), block_rect)
    elif direction == Vector2(0, 1):
        screen.blit(pygame.transform.rotate(
            self.snake_head, 270), block_rect)
    elif direction == Vector2(0, -1):
        screen.blit(pygame.transform.rotate(
            self.snake_head, 90), block_rect)
```



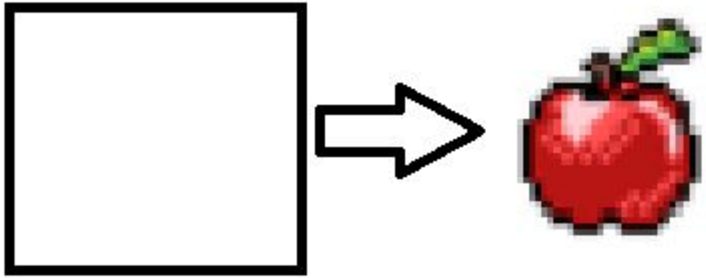
Movimiento de la serpiente

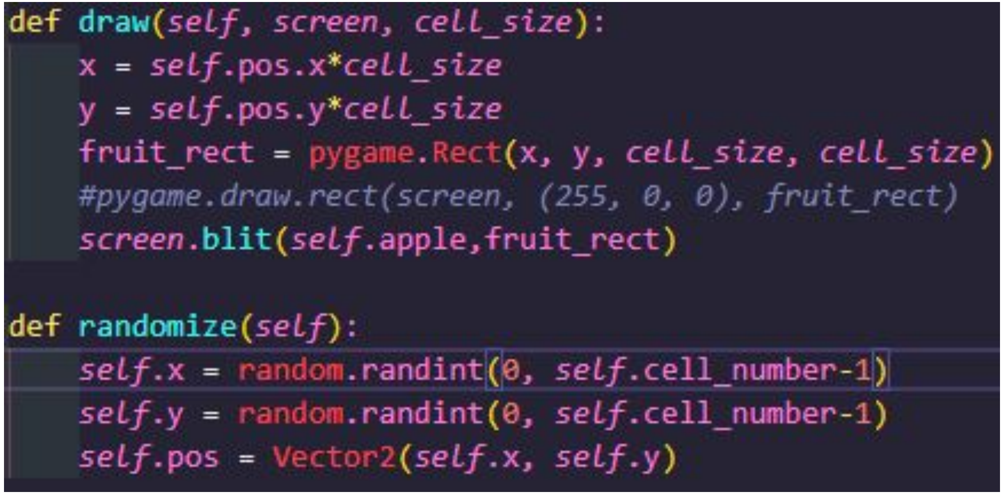




```
def move_snake(self):
    if self.direction != Vector2(0, 0):
        body = self.body[:-1]
        body.insert(0, body[0]+self.direction)
        self.body = body[:]
```

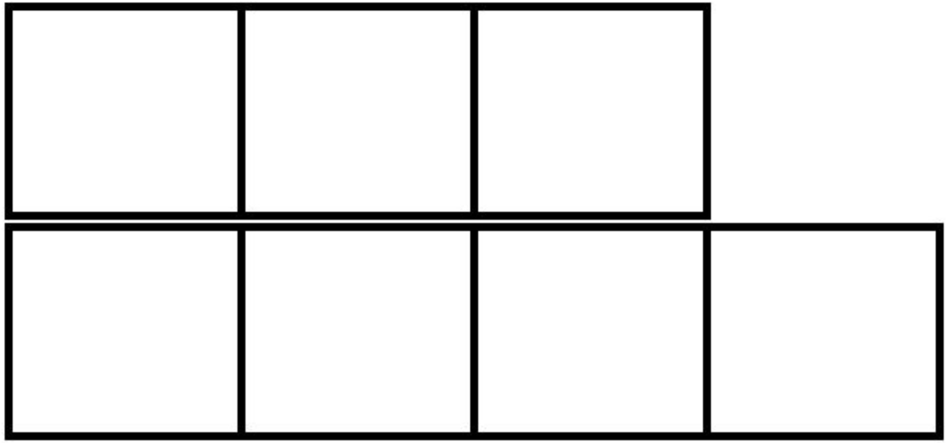
Fruta

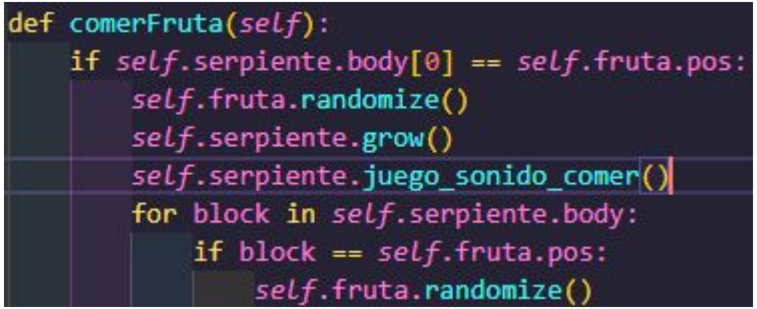




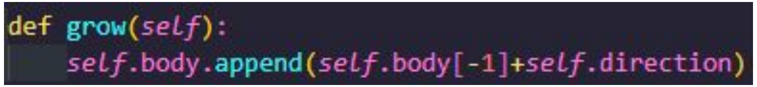
```python
def draw(self, screen, cell_size):
    x = self.pos.x*cell_size
    y = self.pos.y*cell_size
    fruit_rect = pygame.Rect(x, y, cell_size, cell_size)
    #pygame.draw.rect(screen, (255, 0, 0), fruit_rect)
    screen.blit(self.apple,fruit_rect)

def randomize(self):
    self.x = random.randint(0, self.cell_number-1)
    self.y = random.randint(0, self.cell_number-1)
    self.pos = Vector2(self.x, self.y)
```

Comer y crecer

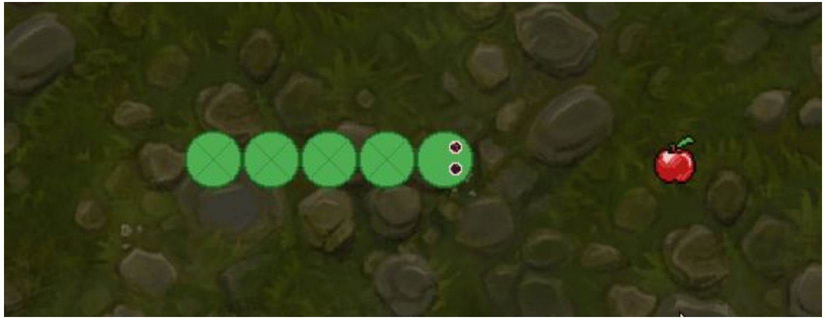




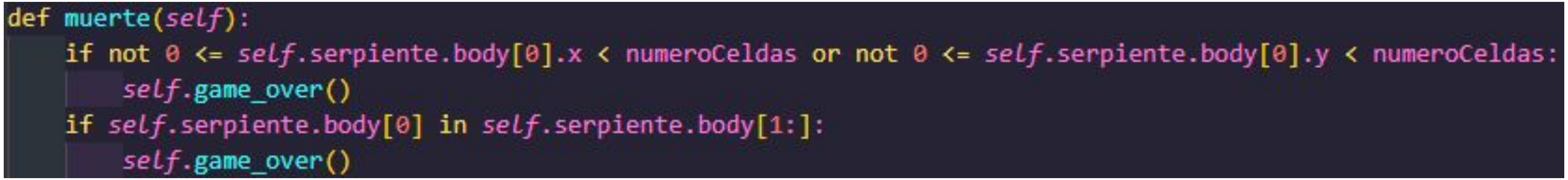
```
def comerFruta(self):
    if self.serpiente.body[0] == self.fruta.pos:
        self.fruta.randomize()
        self.serpiente.grow()
        self.serpiente.juego_sonido_comer()
        for block in self.serpiente.body:
            if block == self.fruta.pos:
                self.fruta.randomize()
```



```
def grow(self):
    self.body.append(self.body[-1]+self.direction)
```

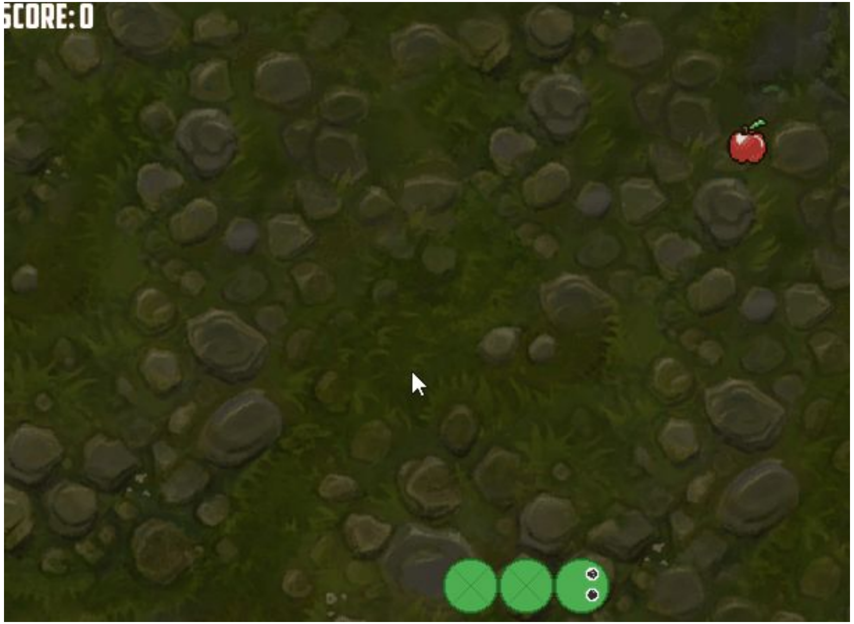


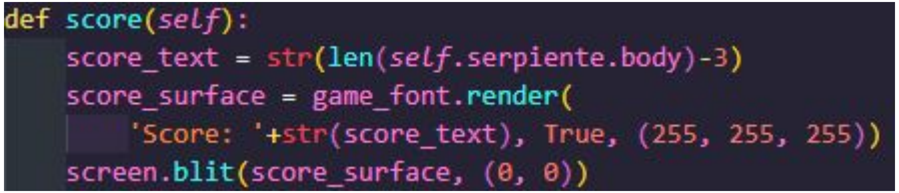
Muerte



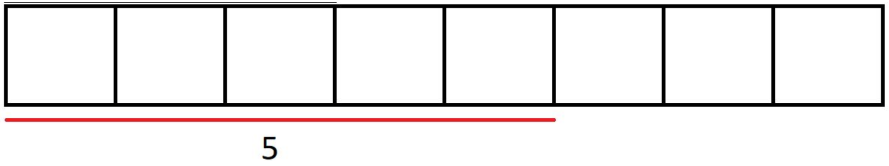
```
def muerte(self):
    if not 0 <= self.serpiente.body[0].x < numeroCeldas or not 0 <= self.serpiente.body[0].y < numeroCeldas:
        self.game_over()
    if self.serpiente.body[0] in self.serpiente.body[1:]:
        self.game_over()
```

Puntuación

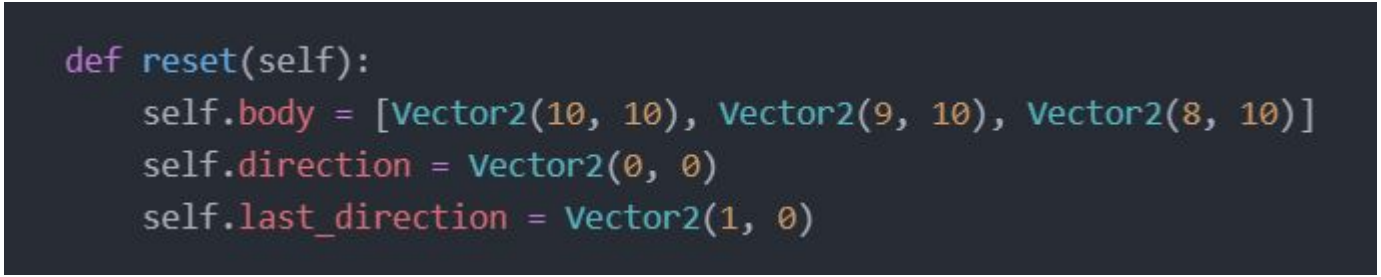




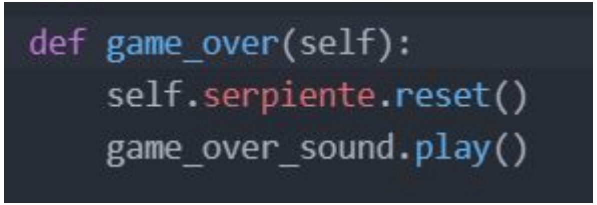
```
def score(self):
    score_text = str(len(self.serpiente.body)-3)
    score_surface = game_font.render(
        'Score: '+str(score_text), True, (255, 255, 255))
    screen.blit(score_surface, (0, 0))
```



Reiniciar y Game Over



```
def reset(self):
    self.body = [Vector2(10, 10), Vector2(9, 10), Vector2(8, 10)]
    self.direction = Vector2(0, 0)
    self.last_direction = Vector2(1, 0)
```



```
def game_over(self):
    self.serpiente.reset()
    game_over_sound.play()
```