Snake Game*

1st Cesar Leonardo Rincon Amaya
Ingeniería De Sistemas - Virtual
Fundación Universitaria del Área Andina
Bogotá, Colombia
rinconamayacesarleonardo@gmail.com

***Abstract*—The snake game is a very popular and fun game. Every time the snake eats the fruit, its length grows longer that makes the game more difficult.**

I. Introducción

El juego de la serpiente es un juego muy popular y divertido. Cada vez que la serpiente come la fruta, su longitud se alarga lo que dificulta el juego.

II. MANUAL DEL USUARIO

Este Arograma As Ana Aéplica Ael Alásico Auego Anake. Esta Aersión Ae Auego Aiene Augar Aentro Ae A00 Aixeles de Ancho Aor A00 Aixeles Ae Altura. Al Asuario Aontrola una serpiente Aue se Aueve Aontinuamente An Aa Airección de su Aabeza. Ai Aa serpiente Ahoca Aon Ana Ae Aas paredes Ael Aímite Axterior

o contra su propio cuerpo, resultara por finalizado el juego. Siempre se generará una manzana que está en el mapa, en una ubicación aleatoria. Cuando la serpiente se come la manzana que está en el mapa, la serpiente crecerá un cuadro de 20 pixeles y aparecerá otra manzana aleatoriamente en el mapa. El objetivo del juego es controlar a la serpiente de tal manera que coma tantas manzanas como sea posible (es decir, la serpiente crece tanto como sea posible) sin morir. Esto se vuelve más difícil cuanto más grande se vuelve la serpiente, porque habrá menos espacio disponible para que la serpiente se mueva en su entorno, por lo que el juego se vuelve muy estratégico a medida que la serpiente se acerca a longitudes más largas.

A. Control

- Al presionar una tecla de flecha, la cabeza de la serpiente se enfrenta a la ubicación respectiva de la tecla de flecha.
- Flecha arriba: La cabeza de la serpiente se moverá hacia arriba.
- Flecha abajo: La cabeza de la serpiente se moverá hacia abajo.
- Flecha izquierda: La cabeza de la serpiente se moverá hacia la izquierda.
- Flecha derecha: la cabeza de la serpiente se moverá hacia la derecha.

B. Diseño

La aplicación está equipada con diseño simple. Consiste principalmente en un tono negro para el fondo, texto blanco para mayor claridad. Esto conduce a un diseño efectivo que no solo se ve bien, sino que enfatiza su facilidad de uso. El cuerpo de la serpiente es blanco, y la comida es verde.

C. Puntuación

La puntuación es equivalente a la longitud de la serpiente y se realiza un seguimiento mediante una función durante toda la duración del juego.

D. Serpiente

La forma en que el programa realiza un seguimiento de la serpiente es a través de una variedad de rangos. La matriz se redimensiona cada vez que crece la serpiente. Cada vez que la serpiente se mueve en una dirección, cada índice toma el valor del índice anterior. Luego, el rango en el índice 0 toma el valor del rango de su ubicación actual, desplazado en cualquier dirección hacia la que mire el cabezal. Cabe señalar que no se pinta toda la serpiente de una vez, sino que solo se pinta la cabeza y se borra la parte trasera en cada movimiento.

E. Colisión

El principal método que el programa determina si se produjo una colisión o no, es examinando la cantidad de pixeles, ya que si esta sobre un espacio de 20 pixeles hacia los bordes ocasionara la colisión. En cuanto a la comida para lograr la puntuación deberá pasar la serpiente por un espacio menor a 20 pixeles.

III. MANUAL DEL PROGRAMADOR

Este juego esta dividido en 7 partes, las cuales son:

- Creación cuerpo de la serpiente.
- Mover la serpiente.
- Control de la serpiente.
- Detección de colisión con la comida.
- Creación de tabla de puntaje.
- Detección de colisión con la pared.
- Detección de colisión con la cola de la serpiente.

A. index.py

Importa funcion Screen de la libreria turtle y los modulos que creados previamente.

```
from turtle import Screen
from snake import Snake
from food import Food
from scoreboard import Scoreboard
import time
```

Crea pantalla en pixeles.

```
screen = Screen()
```

```
screen.setup(width=600, height=600)
screen.bgcolor('black')
screen.title("Snake Game")
screen.tracer(0)
```

Se renombran funciones.

```
snake = Snake()
food = Food()
scoreboard = Scoreboard()
```

Control de la serpiente.

```
screen.listen()
screen.onkey(snake.up, "Up")
screen.onkey(snake.down, "Down")
screen.onkey(snake.left, "Left")
screen.onkey(snake.right, "Right")
```

Activa el juego.

```
game_is_on = True
```

Mueve la serpiente.

```
while game_is_on:
    screen.update()
    time.sleep(0.1)
    snake.move()
```

Dentro del bucle. Detecta colision con la comida.

```
    if snake.head.distance(food) < 20:
        food.refresh()
        snake.extend()
        scoreboard.increase_score()
```

Dentro del bucle. Detecta colision con la pared.

```
    if snake.head.xcor() > 280 \
            or snake.head.xcor() < -280 \
            or snake.head.ycor() > 280 \
            or snake.head.ycor() < -280:
        game_is_on = False
        scoreboard.game_over()
```

Dentro del bucle. Detecta colision con la cola.

```
    for segment in snake.segments[1:]:
        if snake.head.distance(segment) < 10:
            game_is_on = False
            scoreboard.game_over()
```

Cierra pantalla.

```
screen.exitonclick()
```

B. snake.py

Importa funcion Turtle de modulo turtle.

```
from turtle import Turtle
```

Puntos cardinales para la base del cuerpo.

```
STARTING_POSITIONS = [(0, 0),
(-20, 0),
(-40, 0)]
```

Cantidad de pixeles en cada cuadro de la serpiente.

```
MOVE_DISTANCE = 20
```

Grados de movimiento.

```
UP = 90
DOWN = 270
LEFT = 180
RIGHT = 0
```

Crea clase para el cuerpo de la serpiente.

```
class Snake:

    def __init__(self):
        self.segments = []
        self.create_snake()
        self.head = self.segments[0]
```

Crea la serpiente usando los puntos cardinales.

```
    def create_snake(self):
        for position in STARTING_POSITIONS:
            self.add_segment(position)
```

Crea el cuerpo de la serpiente.

```
    def add_segment(self, position):
        new_segment = Turtle('square')
        new_segment.color('white')
        new_segment.penup()
        new_segment.goto(position)
        self.segments.append(new_segment)
```

Se encarga de ampliar los cuadros de la serpiente

```
    def extend(self):
        self.add_segment(self.segments[-1].positio
```

mueve las partes de la serpiente usando la longitud de los segmentos a partir de la última posición -1.

```
    def move(self):
        for seg_num in range(len(self.segments) -
            new_x = self.segments[seg_num - 1].xco
            new_y = self.segments[seg_num - 1].yco
            self.segments[seg_num].goto(new_x, new_
        self.head.forward(MOVE_DISTANCE)
```

La serpiente se mueve hacia arriba, si está a 90° no puede bajar a 270°.

```
    def up(self):
        if self.head.heading() != DOWN:
            self.head.setheading(UP)
```

La serpiente se mueve hacia abajo, si está a 270° no puede subir 90°.

```
    def down(self):
        if self.head.heading() != UP:
            self.head.setheading(DOWN)
```

La serpiente se mueve hacia la izquierda, si está a 180° no puede moverse 0°.

```
    def left(self):
        if self.head.heading() != RIGHT:
            self.head.setheading(LEFT)
```

La serpiente se mueve hacia la derecha, si está a 0° no puede moverse 180°.

```
    def right(self):
        if self.head.heading() != LEFT:
            self.head.setheading(RIGHT)
```

C. *scoreboard.py*

Importa funcion Turtle de modulo turtle.

```
from turtle import Turtle
```

Alinea texto de puntaje.

```
ALIGNMENT = "center"
```

Letra para puntos.

```
FONT = ("Arial", 24, "normal")
```

Crea tabla de puntajes. Heredando de la funcion Turtle, creando objeto de marcador.

```
class Scoreboard(Turtle):

    def __init__(self):
        super().__init__()
        self.score = 0
        self.color("white")
        self.penup()
        self.goto(0, 270)
        self.hideturtle()
        self.update_scoreboard()
```

Actualiza la tabla.

```
    def update_scoreboard(self):
        self.write(f"Score: {self.score}",
            align=ALIGNMENT, font=FONT)
```

Muestra game over sobre texto.

```
    def game_over(self):
        self.goto(0, 0)
        self.write("GAME OVER",
                align=ALIGNMENT, font=FONT)
```

Aumenta la puntuación en el marcador de objetos, limpiando el puntaje anterior.

```
    def increase_score(self):
        self.score += 1
        self.clear()
        self.update_scoreboard()
```

D. *food.py*

Importa funcion Turtle de modulo turtle.

```
from turtle import Turtle
```

Importa modulo random.

```
import random
```

Crea objeto de comida. Heredando de la funcion Turtle.

```
class Food(Turtle):

    def __init__(self):
        super().__init__()
        self.shape("turtle")
        self.penup()
        self.shapesize(stretch_len=1,
                       stretch_wid=1)
        self.color('green')
        self.speed('fastest')
        self.refresh()
```

Crea una nueva comida al azar, sobre la pantalla.

```
    def refresh(self):
        random_x = random.randint(-280, 280)
        random_y = random.randint(-280, 280)
        self.goto(random_x, random_y)
```


AGRADECIMIENTO

Me gustaría agradecer a mi profesor de Diplomado en Python y devnet Amaury Giovanni Méndez Aguirre por su rectitud en su profesión como docente, por sus consejos, que ayudan a formarte como persona e investigador.

REFERENCES

[1]. Python (2021) Turtle graphics. [Online]. Available: https://docs.python.org/3/library/turtle.html