

Universidad Nacional Autónoma de México



Facultad de Ingeniería



Estructura de Datos y Algoritmos I

Proyecto Final: Snake en Python

Herrera Alcántara Emilio Ramsés

M.I. Marco Antonio Martínez Quintana

13/08/2021

Resumen

Este es el documento de mi proyecto final de la materia de Estructura de Datos y Algoritmos I. Como primera parte se podrá leer la Introducción en dónde hablo de temas muy interesantes como la posible problemática de los videojuegos, una parte sobre el supuesto aislamiento de los niños por estar utilizando este tipo de tecnologías, también hablo un poco sobre la historia de los videojuegos, describir que las consolas potentes de hoy ya se están enfocando en ser portátiles y menciono un par de cosas sobre los juegos mas clásicos.

Posterior a la introducción viene la parte en dónde menciono un enfoque básico sobre el juego como: el objetivo del juego, cómo sumar puntos, las reglas y cómo se puede perder.

En la siguiente parte escribí un algoritmo básico con el cuál me pude guiar para pensar en cómo desarrollaría el programa y el orden que tendría que llevar para no confundirme durante la codificación.

La siguiente parte tiene cada una de las líneas de código que hice para el juego y cada una de ellas está comentada para poder hacer más claro el código.

En el código podemos ver que utilicé una librería llamada Turtle que hace más sencillo la parte de ventanas, figuras y movimientos, aparte de también usar funciones, listas, objetos, asignar funciones a teclas y un sistema básico de puntos (entre otras cosas).

En la siguiente parte podemos ver capturas sobre el funcionamiento del juego, una con la posición inicial de la cabeza, otra sobre la forma de perder al chocar con una pared, la otra es la forma de chocar con el cuerpo y otra sobre cómo se pasa la puntuación a la mejor puntuación.

La tabla de recursos tiene algunas cosas muy básicas como un teclado y el código del sistema para llevar a cabo la ejecución.

La tabla de costos tiene el desglose de lo que se podría pedir tanto por el desarrollo como por el manual de usuario.

El diagrama de Gannt es con el cuál se aprecia las horas que le dediqué al desarrollo de este proyecto desde el algoritmo como al video de YouTube y se separa por los tres días en los que estuve trabajando.

Al final del documento vienen las conclusiones en las cuales doy mi opinión sobre la importancia de los algoritmos y cómo fue mi perspectiva al terminar el proyecto, entre otras cosas. Y finalmente las referencias utilizadas.

Introducción

En este momento los videojuegos son la puerta de entrada de niños y jóvenes en las TIC. Mediante el videojuego los niños adquieren capacidades y desarrollan habilidades diversas, las más importantes de las cuáles son la familiarización con las nuevas tecnologías, su aprecio y su dominio. Por este motivo el videojuego es en estos momentos un elemento determinante para socializarse en el mundo de las nuevas tecnologías.

El impacto que los videojuegos han tenido en la sociedad en los últimos treinta años es un fenómeno que todavía no se estudia en profundidad. Sobre todo, por la presión generada por la idea de que las tecnologías “impactan” sobre las personas y sobre la sociedad.

Hay que pensar también en los videojuegos como herramientas de relación y no de aislamiento. Es decir, al entramado de relaciones, de diálogos y emociones que se pueden fomentar a través de estas prácticas de juego, tanto en el ámbito familiar como en otros.

Jugamos a los videojuegos en espacios que pueden ser públicos o privados, en casa o en cibercafés, y sobre todo, varias tipologías de lugares y diferentes modalidades de jugar.

Los videojuegos pueden ser considerados como una nueva forma de expresión artística en las últimas décadas, pero al igual que ocurre con las expresiones artísticas contemporáneas, la sociedad todavía no es capaz de asimilarlas.

Muchas personas ven a la práctica de jugar videojuegos como aislarse, pero puedo decir cómo experiencia personal que los juegos hacen todo lo contrario, nos ayudan a quitar el estrés del día, a relajarnos o hasta ayudarnos a distraernos de un momento complicado en el ámbito personal.

Para las personas que creen que los juegos nos hacen no tener amigos les puedo decir que hoy los juegos son muy avanzados lo que nos permite jugar con amigos o hasta personas de todas las partes del mundo y nosotros podemos hablar con ellos y agregarlos para conocer gente nueva con los mismos gustos que tenemos nosotros.

Hoy en día muchas personas critican cuándo una aplicación o un videojuego arroja un error y no ven la dificultad que tiene elaborar esos programas y estar atentos a corregir cualquier error. Es por esto que mi proyecto se basa en el clásico juego de snake, un juego que parece sencillo de programar, pero hasta este tipo de juegos tienen un grado de dificultad.

Desarrollo del Proyecto

El proyecto se basa en el clásico juego de snake, juego en el cuál tú manejas a una pequeña serpiente la cuál tiene que comer la fruta que sale de manera aleatoria en el plano del juego y cada vez que comes una fruta, el cuerpo de la serpiente irá creciendo. Puedes perder de dos formas, una de ella es si choca tu cabeza con uno de los bordes del plano y la otra es chocar tu cabeza con alguna parte de tu cuerpo.

Algoritmo

INICIO

Definir las medidas de la ventana

Dar formato a la ventana

Crear la cabeza de la serpiente

Darle formato a la cabeza

Asignar posición inicial para la cabeza

Definir función para el movimiento

Asignar de cuántos pixeles será el movimiento de la serpiente

Asignar a los movimientos una tecla

Crear la comida

Hacer que la comida desaparezca al hacer contacto con ella

Hacer que aparezca la comida en lugares aleatorios de la pantalla

Hacer que el cuerpo de la serpiente crezca después de comer

Definir los bordes del juego

Hacer que el juego se reinicie al chocar con los bordes o golpear el cuerpo

Hacer que el cuerpo que teníamos en la partida anterior desaparezca al perder

Generar un score en pantalla

Sumar 10 puntos al comer

Borrar datos del score si perdemos

Almacenar el score como high score si se supera

FIN

Código Fuente comentado

posponer = 0.1

aumento = []

score = 0

high_score = 0

#Funciones

def arriba():

 cserp.direction = "up"

def abajo():

 cserp.direction = "down"

def izquierda():

 cserp.direction = "left"

def derecha():

 cserp.direction = "right"

def mov():

 if cserp.direction == "up": #función para mover hacia arriba

 y = cserp.ycor() #definimos la variable y como la el eje y

 cserp.sety(y + 20) #el movimiento será de 20 pixeles hacia arriba

 if cserp.direction == "down": #función para mover hacia abajo

 y = cserp.ycor() #definimos la variable y como la el eje y

 cserp.sety(y - 20) #el movimiento será de 20 pixeles hacia abajo

 if cserp.direction == "left": #función para mover hacia izquierda

 x = cserp.xcor() #definimos la variable x como la el eje x

 cserp.setx(x - 20) #el movimiento será de 20 pixeles hacia la izquierda

 if cserp.direction == "right": #función para mover hacia la derecha

 x = cserp.xcor() #definimos la variable x como la el eje x

 cserp.setx(x + 20) #el movimiento será de 20 pixeles hacia la derecha

#Personalizar la Pantalla

pantalla = turtle.Screen() #Abrimos una ventana

pantalla.title("Bienvenido a mi juego de Snake") #Título de la ventana

pantalla.bgcolor("cyan") #color de la ventana

pantalla.setup(width =600, height=600) #Dimensiones de la ventana

pantalla.tracer(0) #Mejorar la animación

#Marcador

texto = turtle.Turtle()

texto.speed(0)

texto.color("white")

texto.penup()

texto.hideturtle()

texto.goto(0,260)

texto.write("Score: 0 High Score: 0", align = "center", font =("Courier", 24, "normal"))

#teclado

pantalla.listen() #decirle al programa que esté preparado para las indicaciones

pantalla.onkeypress(arriba, "Up") #Movimiento con las flechas del teclado

pantalla.onkeypress(abajo, "Down") #Movimiento con las flechas del teclado

pantalla.onkeypress(izquierda, "Left") #Movimiento con las flechas del teclado

pantalla.onkeypress(derecha, "Right") #Movimiento con las flechas del teclado

#Creación de la cabeza de la serpiente

cserp = turtle.Turtle()#Definimos un objeto

cserp.speed(0)#Mostramos la cabeza sin movimiento

```
cserp.shape("square")#Definimos la forma de la cabeza
cserp.color("green") #Color de la cabeza
cserp.penup() #Funcion para que la cabeza no deje rastro al avanzar
cserp.goto(0,0)#Posición inicial de la cabeza en mi plano
cserp.direction="stop"#Cabeza en reposo en las coordenadas (0,0)
```

#Creación de las manzanas

```
manzana = turtle.Turtle()#Definimos un objeto
manzana.speed(0)#Mostramos la comida sin movimiento
manzana.shape("circle")#Definimos la forma de la comida
manzana.color("red") #Color de la comida
manzana.penup() #Funcion para que la comida no deje rastro al avanzar
manzana.goto(0,200)#Posición inicial de la comida en mi plano
```

while True:

```
    pantalla.update() #actualizamos constantemente la pantalla
```

```
    if cserp.xcor() > 280 or cserp.xcor() < -280 or cserp.ycor() > 280 or cserp.ycor() < -280: #se genera un límite en los bordes para que la serpiente no pueda pasar
```

```
        time.sleep(1) #si choca con los bordes se pone un tiempo de 1 segundo
```

```
        cserp.goto(0,0) #la cabeza regresa a la posición inicial
```

```
        cserp.direction = "stop" #no hay movimiento
```

```
        for cuerpo in aumento:
```

```
            cuerpo.goto(1000,1000) #El cuerpo se va lejos de la pantalla
```

```
        aumento.clear() #Se borra el progreso que teníamos en el cuerpo
```

```
        score = 0
```

```
        texto.clear()
```

```
        texto.write(" Score: {}      High Score: {} ".format(score, high_score), align = "center", font = ("Courier", 24, "normal")) #formato
```

```
if cserp.distance(manzana) < 20: #si la distancia es menor a 20 pixeles entre la
serpiente y las manzanas
```

```
    x = random.randint(-280,280) #posición de las manzanas aleatorias entre esas
coordenadas
```

```
    y = random.randint(-280,280) #posición de las manzanas aleatorias entre esas
coordenadas
```

```
    manzana.goto(x,y) #posición de las manzanas aleatorias entre esas
coordenadas
```

```
    cuerpo = turtle.Turtle()
```

```
    cuerpo.speed(0)
```

```
    cuerpo.shape("square")
```

```
    cuerpo.color("green")
```

```
    cuerpo.penup()
```

```
    aumento.append(cuerpo)
```

```
    score += 10 #Aumentamos en 10 puntos el score
```

```
    if score > high_score:
```

```
        high_score = score #si nuestro score es mayor al high_score actualizamos
high_score
```

```
        texto.clear()
```

```
        texto.write(" Score: {}      High Score: {} ".format(score, high_score), align =
"center", font = ("Courier", 24, "normal")) #formato
```

```
totcuerpo = len(aumento)
```

```
for i in range(totcuerpo -1, 0, -1):
```

```
    x = aumento[i - 1].xcor()
```

```
    y = aumento[i - 1].ycor()
```

```
    aumento[i].goto(x,y)
```



```
if totcuerpo > 0:  
    x = cserp.xcor()  
    y = cserp.ycor()  
    aumento[0].goto(x,y)
```

```
mov() #llamamos a la función
```

```
for cuerpo in aumento:  
    if cuerpo.distance(cserp) < 20: #Condición para ver si la cabeza choca con el  
cuerpo  
        time.sleep(1) #1 segundo de espera  
        cserp.goto(0,0) #cabeza regresa a la posición inicial  
        cserp.direction = "stop" #la cabeza se encuentra sin moverse
```

```
for cuerpo in aumento:  
    cuerpo.goto(1000,1000) #se mueve el cuerpo acumulado  
    aumento.clear() #se elimina el progreso  
    score = 0  
    texto.clear()  
    texto.write(" Score: {}    High Score: {}".format(score, high_score), align =  
"center", font=("Courier", 24, "normal")) #formato  
    time.sleep(posponer) #delay para ver el movimiento
```

```
turtle.exitonclick() #Comando para que no se cierre la ventana
```

Resultados del Proyecto

Capturas de pantalla del Funcionamiento

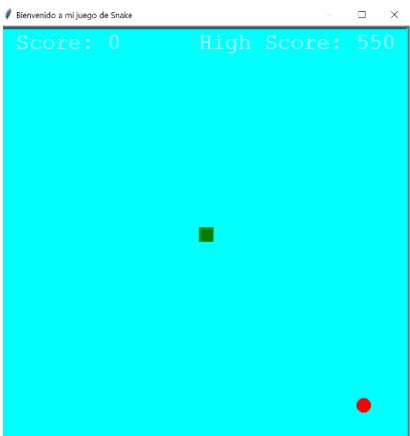
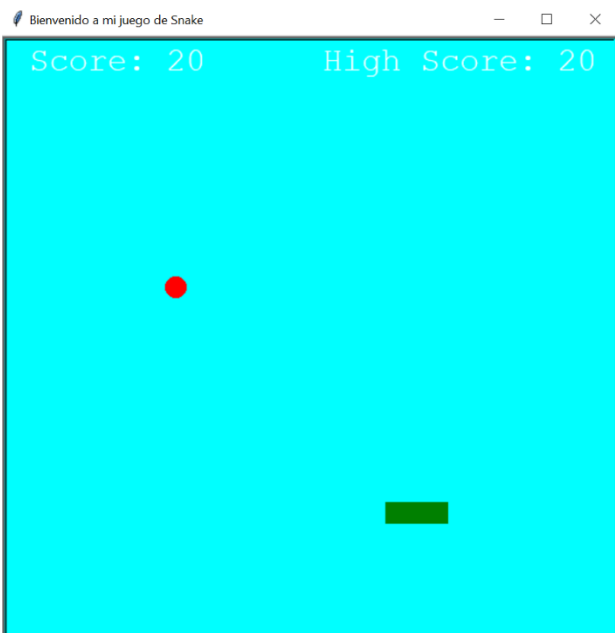
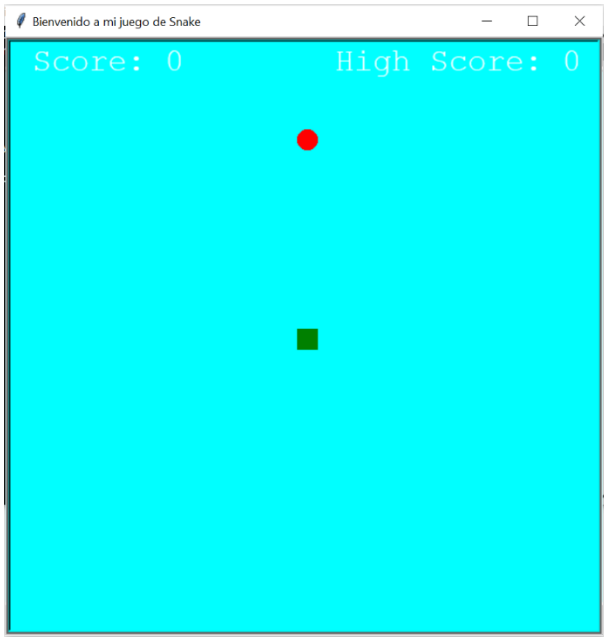


Tabla de recursos

Hardware	Software
Teclado con flechas	Código del juego
Pantalla	Código del sistema para ejecutarlo

Tabla de costos

Proyecto	Costos
Programa	\$400
Manual de Usuario	\$150

Diagrama de Gannt

Actividades	Tiempo de duración								
	11 de agosto			12 de agosto			13 de agosto		
	1	2	3	1	2	3	1	2	3
Algoritmo									
Codificación									
Diseño del documento									
Repositorio									
Video de YouTube									

Conclusiones

El manejo de algoritmos es fundamental hoy en día pues es la base para hacer cualquier programa, nos da las bases para empezar a codificar y sobre todo un buen orden para llevar a cabo las instrucciones. Por otra parte, un buen manejo de las estructuras de datos tiene que ser la base para todo futuro programador, ya que es el puente para demostrar que ya tenemos cierta experiencia al tener más ordenados nuestros programas. El impacto de las estructuras de datos es enorme, porque todo programa grande y profesional deja ver el uso de estos fundamentos, por ejemplo, los videojuegos que usan las pilas.

En lo personal, este proyecto no fue tan desafiante como el de la materia pasada, pero esto se debe a que tuve poco tiempo para realizarlo y decidí optar por este juego que utilizara cosas que aprendí este semestre como listas o Python, que es un lenguaje con el que nunca había trabajado. Lo que si es que gracias a este

proyecto me pude dar cuenta de que yo estaba muy “casado” con usar solo c++ porque ya tenía buenas bases y me daba un poco de miedo explorar estos lenguajes, pero al utilizar Python me di cuenta de que es mucho más sencillo de comprender y que tiene infinidad de cosas que lo hace más amigable con el programador. Me quedo con ganas de aprender más sobre este lenguaje y otros nuevos. Y finalmente me voy con los ojos abiertos de que la programación no se basa en solo un lenguaje y que no hay solo una forma de resolver un problema.

Referencias

- INTRO - Introducción a Turtle en Python, Mauricio Avilés, 31 de marzo de 2017, 11 de agosto de 2021, <https://www.youtube.com/watch?v=HLvifpyCk54>
- Alargando el cuerpo de la serpiente 🐍 | Cómo hacer un videojuego con Python, YouDevs, 7 de septiembre de 2019, 11 de agosto de 2021, <https://www.youtube.com/watch?v=jo6GK8aDaCw>
- Simone Belli y Cristian López Raventós. (2008). Breve Historia De Los Videojuegos. 12-08-2021, de Athenea Digital, 14(159-179)