

# Tarea 1

OpenGL 2D

Integrantes: Esteban Zúñiga S.  
Profesor: Nancy Hitschfeld K.  
Auxiliar: Andrés Cerda  
Pablo Pizarro R.  
Alonso Utreras  
Ayudantes: Tomás Calderón R.  
Beatriz Grabolosa M.  
Sebastián Olmos H.  
Nadia Decar  
Ayudante de laboratorio: Sergio Alvarez

Fecha de entrega: 14 de octubre de 2020  
Santiago, Chile

# 1. Solución propuesta

Para la realización de esta tarea, se escogió desarrollar el clásico juego “Snake”.

El desarrollo de la tarea consistió en la realización de dos scripts, “modelo” y “vista”. El primero, contiene algunas funciones con las cuales se crearon las figuras. Para las manzanas se utilizaron dos círculos, luego fueron modeladas utilizando modelación jerárquica, al igual que la serpiente, aunque esta es un poco más compleja, ya que se creó una función que creaba una serpiente como un cuadrado con textura, luego otra función que se encargaba de dibujar la serpiente de largo  $N$ , siendo  $N - 1$  las manzanas que se ha comido la serpiente.

En el script “vista” posee todo lo correspondiente a lo que ocurre en pantalla, es decir, dibuja el fondo y las figuras (serpiente y manzanas), evalúa una posible colisión, despliega el mensaje de “Game Over”, etc. Algunas funcionalidades importantes a destacar son:

¿Cómo crece la serpiente?: La serpiente, a medida que come una manzana, crece su tamaño en una unidad, esta unidad corresponde a un cuadrado del tamaño del espacio de juego, 1.9, dividido en el número  $N$  ingresado por el jugador, es decir, un cuadrado de lado  $\frac{1.9}{N}$ . La forma en que crece es simple, se agrega una unidad extra en la cola de la serpiente, posición que se encuentra guardada en una variable.

¿Cómo se mueve la serpiente?: Luego de observar que mover todos los cuadrados de la serpiente no era muy eficiente, se encontró una forma de hacer este proceso más eficiente, el cual consiste en poner el último cuadrado de la serpiente, es decir, la cola, delante de la cabeza, así, solo se traslada solo un cuadrado por movimiento. Esto se logra accediendo y trasladando al último nodo de la serpiente creada, luego al penúltimo y así sucesivamente, cuando se ya se han trasladado todos los nodos, se recorre nuevamente desde el último. El siguiente dibujo ilustra de mejor manera el proceso:

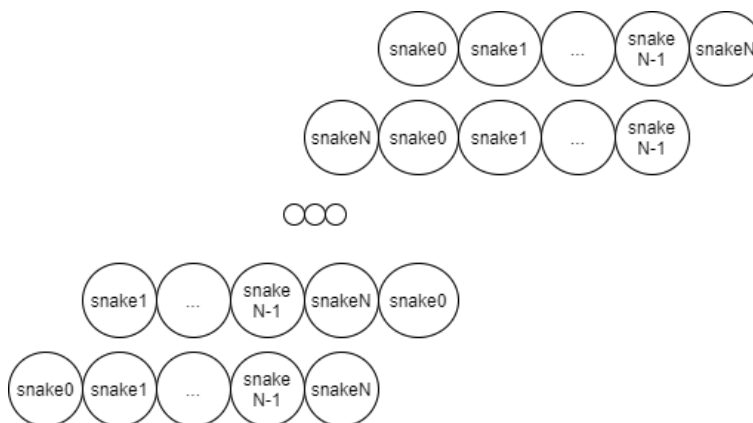


Figura 1: Movimiento de serpiente

Función update: Se creó una función update, que dependiendo de la variable *state*, indica hacia donde se debe mover la serpiente, por ejemplo si *state* == "RIGHT", la serpiente se moverá hacia la derecha. La función update se llama cada *i* ciclos *while*, siendo  $i = 30 - (5 * largoSerpiente) // 8$ . Se escogió este valor para *i* debido a que si se escoge solo 30, la serpiente se avanza más lento, esto debido a que mientras más grande la serpiente, el ciclo *while* demora más en ejecutarse.

¿Qué ocurre cuando el jugador pierde?: En el momento que la serpiente choca contra algún borde, o contra ella misma, el jugador pierde, desaparecen las figuras y se despliega una textura animada con un giro parcial con la frase "GAME OVER".

## 2. Instrucciones de ejecución

Para ejecutar el código, debe dirigirse al directorio *tarea1c* y escribir el comando que aparece a continuación (*N* indica la dimensión del espacio de juego, se recomienda un número entre 15 y 25):

```
1 python vista.py N
```

La serpiente se puede controlar con las teclas *WASD* o las *flechas* de su computador, para efectos de revisión, se activó la posibilidad de ver las figuras sin relleno, presionando *espacio*.

## 3. Resultados

A continuación, se adjuntan algunas imágenes del juego en funcionamiento:

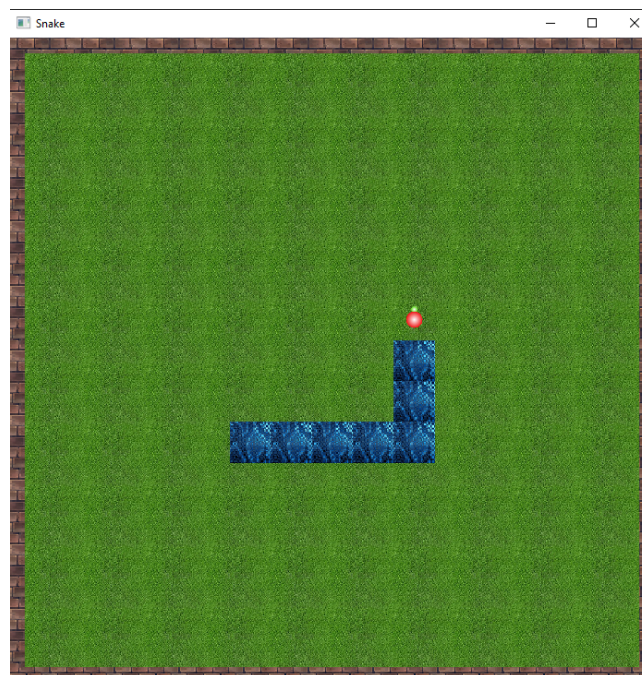


Figura 2: Serpiente comiendo una manzana

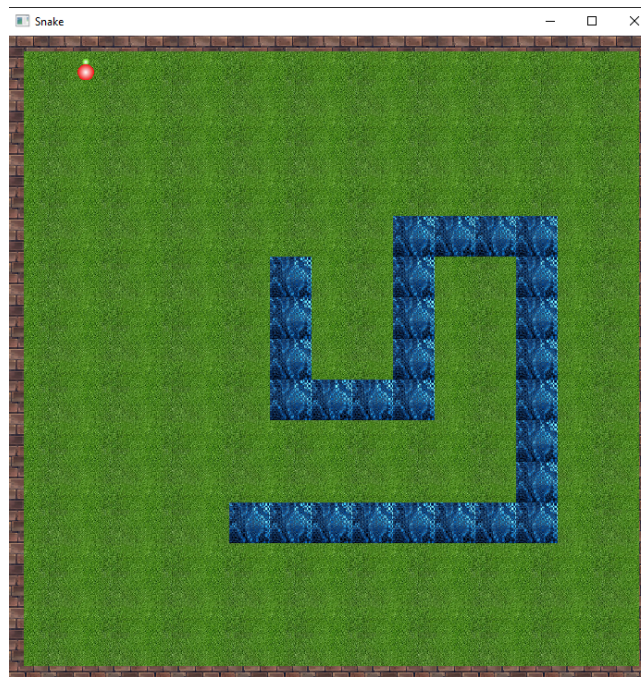


Figura 3: Serpiente de largo 28



Figura 4: GAME OVER

*\*En caso de encontrar algún bug, por favor avisar.*

*\*Link al repositorio en GitHub : [https://github.com/estebanzuniga/my\\_snake](https://github.com/estebanzuniga/my_snake)*