Documentación: Snake-Game

Tarea 1: OpenGL 2D

Autor: Felipe Urrutia Vargas
Fecha de realización: 10 de octubre de 2020, Fecha de entrega: 14 de Octubre 2020.

1. Solución propuesta

1.1. Elementos

- *Serpiente:* Modelo principal que es controlado por el usuario. Compuesta de:
  - *Cabeza:* Elemento principal definido por posición y dirección. Movimiento controlado por el usuario. Interactua con los elementos de colisión.
  - *Cuerpo:* Elemento secundario definido por posiciones. Inicialmente vació, y luego un elemento de colisión.
- *Comida:* Modelo secundario definido por posición. Es un elemento de colisión estático, salvo actualización (eliminar y agregar nuevo), la cual es aleatoria en el *espacio vació* del **mapa**.
- *Mapa:* Conjunto de espacio (fondo o grilla) y borde. El **borde** es un elemento de colisión estático dado por la geometría del mapa. El *espacio vació* es el conjunto de posiciones en el espacio que no contienen a la **serpiente** ni a la **comida**.

Observación: Todo elemento es temporal, i.e depende del instante de tiempo.

1.2. Lógica

Para este videojuego se considero la siguiente dinámica (*ver Figura 1*):

- *Estado actual:* Configuración (posición, dirección) de la **serpiente** en un instante de tiempo.
- *Actualizar:* Configuración de la **serpiente** en el siguiente instante de tiempo.
- *¿Colisiona?:* Si la **serpiente** no interactua con un elemento de colisión (borde, cuerpo, comida), entonces *Mover*. En caso contrario, preguntar si el elemento de colisión es **comida**.
- *Mover:* Visualizar la configuración dada en el modulo *Actualizar*. Luego, considerar dicha configuración como el *Estado actual*, y repetir la dinámica.
- *¿Comida?:* Si el elemento de colisión no es **comida**, entonces es el **borde**, o bien, el **cuerpo**, en tal caso el nuevo estado de la **serpiente** es *Muerte*. En caso contrario, preguntar si existe espacio en el **mapa** para agregar un nuevo elemento **comida**.

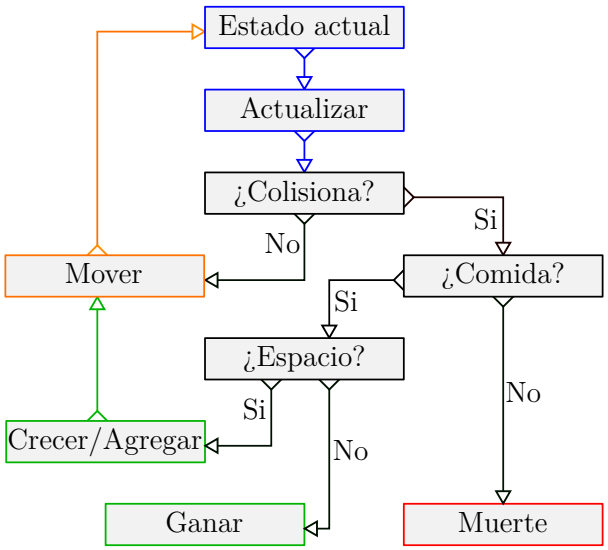

Figura 1: Diagrama de flujo principal.

- *Muerte:* La sesión de juego termina como una derrota.
- *¿Espacio?:* Si no existe espacio en el **mapa** para agregar un nuevo elemento **comida**, entonces la serpiente obtiene su largo máximo, en tal caso el nuevo estado de la serpiente es *Ganar*. En caso contrario, enviar a *Crecer/Agregar*.
- *Ganar:* La sesión de juego termina como una victoria.
- *Crecer/Agregar:* El **cuerpo** de la **serpiente** crece en una unidad de largo y se agregar un nuevo elemento **comida** con ubicación aleatoria en el espacio vació del **mapa**. Luego, enviar a *Mover*.

2. Instrucciones de ejecución

Ejecución paso a paso:

- *Extraer de .zip:* Consideremos `directorio` a la ubicación de la carpeta abierta `SnakeGame_FUV` después de extraer todo de `SnakeGame_FUV.zip`.
- *Inicialización:* Ahora, al iniciar `anaconda prompt` y dirigir con comando `cd directorio`, entregamos un parámetro N que indica la dimension de la grilla junto al archivo de *python* `view.py` que contiene al juego. Esto es: `-> python view.py N`, con N natural, se recomienda valores entre 10 y 30 como dimension estándar, o bien entre 4 y 10 para familiarizarse, y en general a lo mas 50.

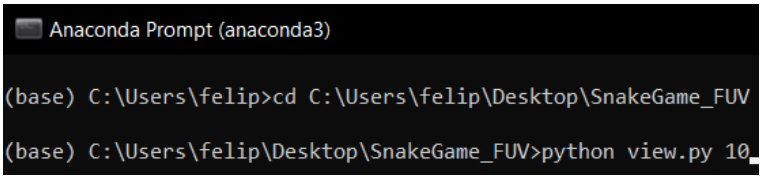

Figura 2: Ejemplo, para $N = 10$.

2.1. ¿Como jugar?

Luego, de iniciar correctamente, aparecerá una imagen rotando (*ver Figura 4*) que indica usar *barra espaciadora* para comenzar a jugar.

En caso de comenzar, deberá de aparecer una imagen como el ejemplo del lado derecho.

Para moverte se pueden utilizar tanto las "flechas" (Up, Left, Down, Right) como las teclas WASD, esto es: Arriba−Izquierda−Abajo−Derecha respectivamente.

Mientras la serpiente se mueve tan solo puedes girar en 90 grados, o lo que es lo mismo, girar en la dirección ortogonal a la actual.

Como la serpiente al inicio no se mueve, puedes comenzar a avanzar en cualquier sentido de los ejes vertical u horizontal.

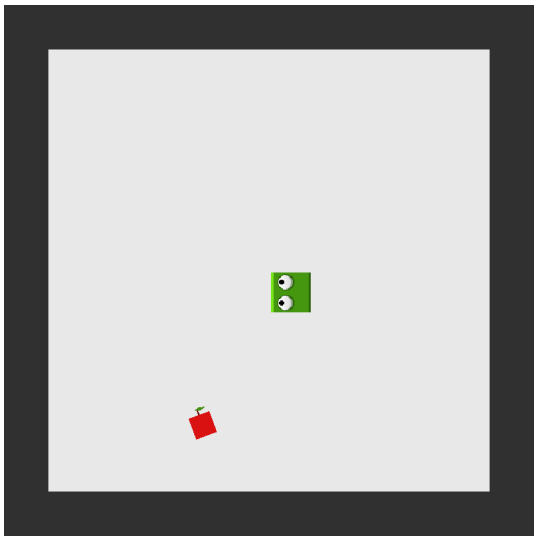

Figura 3: Inicio de juego para $N = 10$.

En esta version del videojuego se puede pausar la partida actual con la tecla ESC, y en caso de morir se puede cambiar la rapidez de la serpiente con la tecla H donde se puede seleccionar alguna con los números 1, 2 y 3, ordenada de menor a mayor c/r a la rapidez en una imagen (*ver Figura 6*).

Para morir, basta chocar con el borde del mapa o bien chocar con el cuerpo de la serpiente (*ver Figuras 9, 10*), lo cual se indicara con una imagen (*ver Figura 5*). Siempre que la serpiente muera se puede presionar la *barra espaciadora* para volver a comenzar (una nueva partida).

Ahora bien, para ganar es necesario conseguir toda las frutas las cuales harán crecer a la serpiente, aumentando la dificultad. Eventualmente no habrá mas espacio para nuevas frutas (*ver Figura 11*), en tal caso se indicara con una imagen (*ver Figura 7*).

Para conseguir una fruta, basta pasar con la cabeza de la serpiente por sobre la comida (*ver Figura 8*), inmediatamente una nueva fruta aparecerá de manera aleatoria en un lugar donde no este la serpiente.

3. Resultado

Imágenes pop−up que indican el estado del juego, antes de:

Figura 4: Intro y al pausar.

Figura 5: Al morir.

Figura 6: Al cambiar rapidez.

Figura 7: Al ganar.

Algunos instantes importantes de la funcionalidad del juego son:

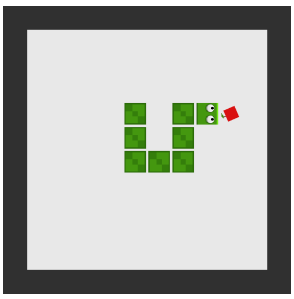

Figura 8: Comer una fruta.

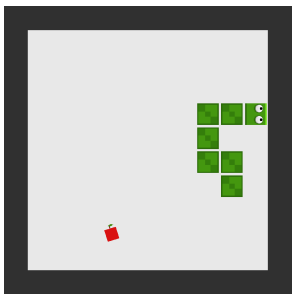

Figura 9: Chocar con el borde.

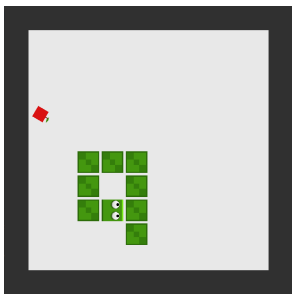

Figura 10: Chocar con el cuerpo.

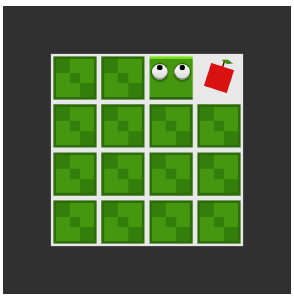

Figura 11: Ganar.

Otro ejemplo, el modelo de la serpiente y la comida (fruta):

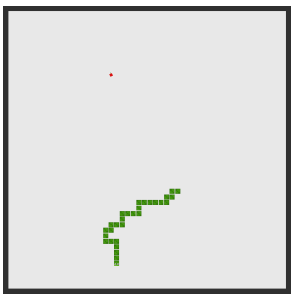

Figura 12: $N = 50$.

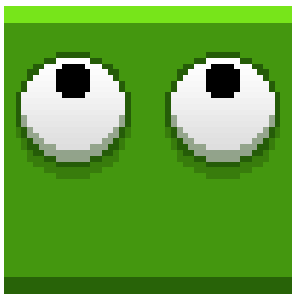

Figura 13: Cabeza.

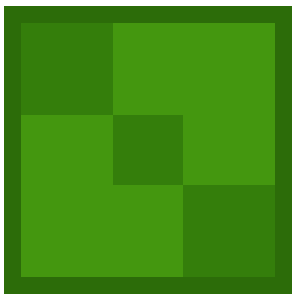

Figura 14: Cuerpo.

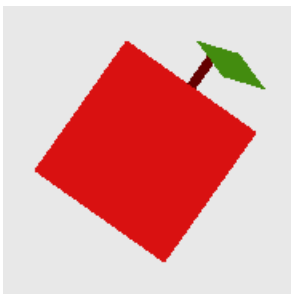

Figura 15: Fruta.