Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ciencias de la Computación

Reporte Tarea 1

Autor: Roberto Tapia
Profesor: Nancy Hitschfeld K.
Fecha de entrega: 15 de octubre
Santiago, Chile

Solución Propuesta

La solución implementada se basa en la arquitectura MVC. En el caso del modelo, se utiliza una clase Snake, que contiene la lógica del juego. Entre sus atributos de un objeto de esta clase se encuentran:

- piezas: Una lista de los "nodos"de la serpiente identificados por un par que indica la posición en ambos ejes de cada nodo.
- maxLength: El largo máximo actual de la serpiente, aumentara cada vez que la serpiente coma una manzana.
- snakeTextura: Un GPUShape que se utilizara para cada nodo de la serpiente.
- dirección: La dirección en la que se encuentra actualmente moviéndose la serpiente.
- grafo: Un grafo de escena que representa todo el juego.
- muerto: Un booleano que representa si el jugador ha perdido o no.
- grilla: Un entero que representa el tamaño de la grilla, debe ser recibido como parámetro del constructor
- manzana: Un par que representa la posición de la manzana actualmente.
- manzanaShape: Un GPUShape que se usara para dibujar la manzana.
- hasChangedDirection: Un booleano que indica si en el ïnstante.ªctual, la serpiente a cambiado de dirección.

La clase Snake cuenta con 4 métodos:

- drawFondo: Se encarga de dibujar el borde el centro blanco sobre el cual la serpiente se mueve.
- draw: Se encarga de dibujar la serpiente en cada instante, para ello primero checkea si la serpiente ha muerto, en caso de que así sea, muestra una textura con un mensaje de game over. En caso contrario, itera sobre los nodos de la serpiente y dibuja un nodo en la posición indicada por el par. Finalmente dibuja la manzana.
- move: Calcula los movimientos y colisiones de la serpiente, tanto como con si misma, con la manzana y con los bordes. Utiliza un contador para verificar que la serpiente solo se mueva 3 veces por segundo. Para generar el movimiento, se utiliza la lista de nodos de la serpiente, y en cada iteración que corresponda moverse, se agrega un nuevo nodo en al principio de la lista, para calcular su posición, se suma la dirección con el primer valor actual de la lista, luego se checkea si el largo de la serpiente es mayor al permitido, en caso de ser así, se elimina el ultimo elemento de la lista.
- changeDirection: Cambia la dirección en que se mueve la serpiente, para ello chequea que no haya cambiado de dirección en el instante actual, y además que la nueva dirección no se directamente contrario a la dirección actual.

En cuanto al archivo de vista, es muy similar a lo trabajado en clases auxilares, se centra en una iteración, donde en cada ciclo se reciben los eventos del controlador, luego se dibuja el fondo blanco, luego la serpiente y la manzana, luego se calculan los movimientos para la siguiente iteración, y finalmente se cambian los buffers para que el usuario solo vea la imagen terminada.

Instrucciones de ejecución

Para la ejecución del programa se utiliza el archivo view.py, en la llamada se debe incluir solo un argumento, el tamaño de la grilla del juego. Con respecto a los controles, se usan las teclas direccionales y WASD para cambiar la dirección en la que se mueve la serpiente.

Resultados

A continuación se muestran imágenes de un juego en proceso (en grilla de tamaño 10) y de un juego finalizado.