

Tarea 2

OpenGL 3D

Alumna: Fabiola Rivera

Rut: 19.673.174-1

Arquitectura de la Solución.-

Para llevar a cabo la implementación de la solución de la tarea 2C, se decidió aplicar la siguiente arquitectura basada en el patrón de diseño MVC, para de esta manera poder designar y encapsular las responsabilidades existentes en el juego. En la siguiente figura se puede ver las dependencias entre los paquetes y las clases, las cuales se encuentran anteceditas por un asterisco.

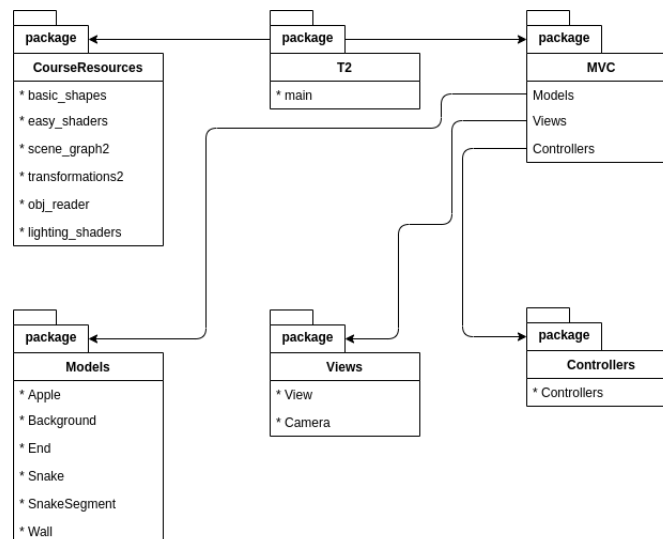


Figura 1: Organización de paquetes y clases

1. **Paquete CourseResources:** Es el encargado de almacenar todas las clases provistas por el curso para encargarse de las interacciones con OpenGL 3D. Además se incluyeron 2 clases con funciones provistas por el curso encargadas de leer un archivo tipo obj y los shaders encargados de la iluminación.
2. **Paquete T2:** Contiene las clases y paquetes relacionados a la implementación del snake 3D. Además contiene la clase main, la cual se encarga de correr el juego.
3. **Paquete MVC:** Contiene los paquetes relacionados a la estructura de MVC
4. **Paquete Models:** Contiene las entidades necesarias para el juego, relacionadas con los modelos:

- a) **Clase Apple:** Encapsula el comportamiento de la entidad Manzana y genera la representación gráfica implementada con OpenGL a partir de un archivo tipo obj y la iluminación de esta.
 - b) **Clase Background:** Genera la representación gráfica del fondo del juego mediante una textura para una mejor estética.
 - c) **Clase End:** Encapsula el comportamiento del final del juego y genera la representación gráfica de este en el juego mediante un cubo texturado.
 - d) **Clase Snake:** Encapsula el comportamiento de la entidad Serpiente en general, la que está compuesta por uno o más SnakeSegments.
 - e) **Clase SnakeSegment:** Encapsula el comportamiento de un segmento de la serpiente y la representación gráfica de esta en el juego. Esta clase busca aislar el comportamiento de un segmento y de esta manera simplificar las mecánicas de esta.
5. **Paquete Views:** Contiene la entidad necesaria para el juego relacionada con la vista de este
- a) **Clase View:** Es la encargada de iniciar todos los procesos relacionados a la parte gráfica del juego, como iniciar la ventana, establecer los distintos pipelines de OpenGL, shaders, entre otros. Además, es la encargada de representar en el juego los distintos modelos previamente señalados, utilizando las mecánicas propias de estos y las del juego propiamente tal.
 - b) **Clase Camera:** Encapsula el comportamiento de las 3 cámaras presentes en el juego, mediante la manipulación de las vistas y proyecciones entregadas por OpenGL.
6. **Paquete Controllers:** Contiene la entidad necesaria para el juego relacionada con el controlador de este
- a) **Clase Controller:** Es el encargado de manejar el comportamiento de la serpiente en el juego, mediante el método *on key* que recibe el input del jugador y en función de este, le indica a la serpiente que tipo de movimiento debe realizar y en el caso de estar con una cámara de primera persona modifica el movimiento para que este sea relativo al movimiento del Snake.

Instrucciones de Ejecución.-

1. **Argumentos que recibe:** El programa recibe 1 argumento desde la clase main, el cual corresponde a la dimensión de la grilla del juego. Este argumento se espera que sea un valor entre 10 y 40, de caso contrario se asume un tamaño igual a 10.
2. **Teclas de control:** Para poder mover la serpiente en el juego se utilizan las teclas WASD o las flechas de teclado, para cambiar la cámara a una dinámica en primera persona se utiliza la tecla R. A su vez, para cambiar la cámara a una estática se tiene que al apretar la tecla E la cambia a una que le da una vista 2d y la tecla T cambia a una con vista perspectiva desde una de las esquinas de la grilla. La velocidad de la serpiente se mantiene constante, pero puede ser modificado para variar la dificultad en el archivo View.
3. **Consideraciones:** El programa tiende a demorarse en iniciar, ya que la mayoría de los modelos fueron realizados mediante texturas y a su vez inicia pygame solo para incluir música en el juego. Es por esto, que al principio es esperable ver la ventana negra por una cantidad de tiempo no menor.

Resultados.-

A continuación se presentan distintas vistas del juego resultante:

- Vistas en el juego

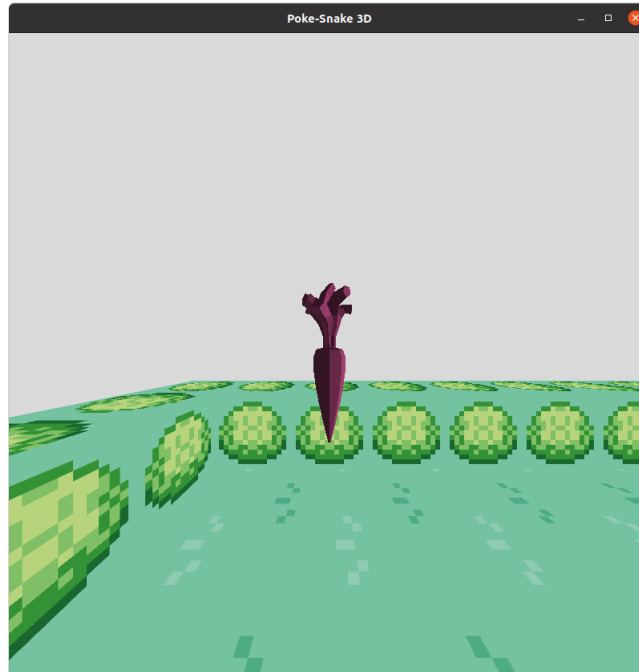


Figura 2: Vista en primera persona

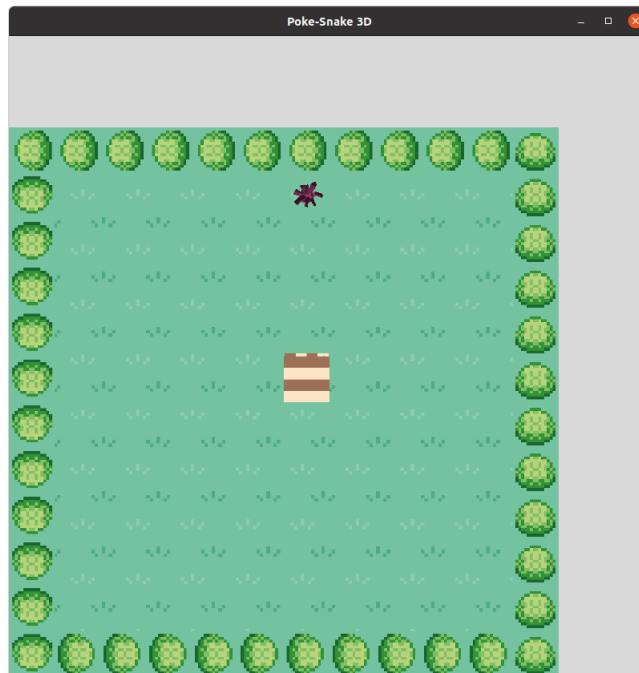


Figura 3: Vista 2D

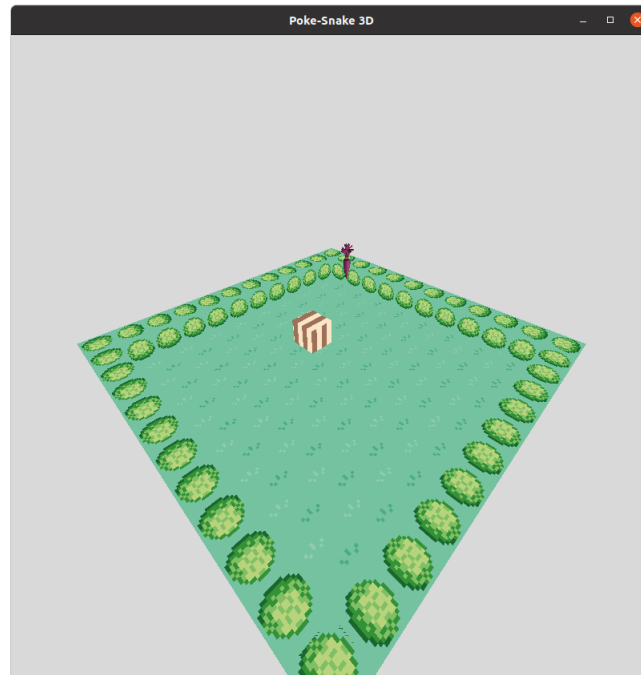


Figura 4: Vista perspectiva

■ Dinámicas del juego

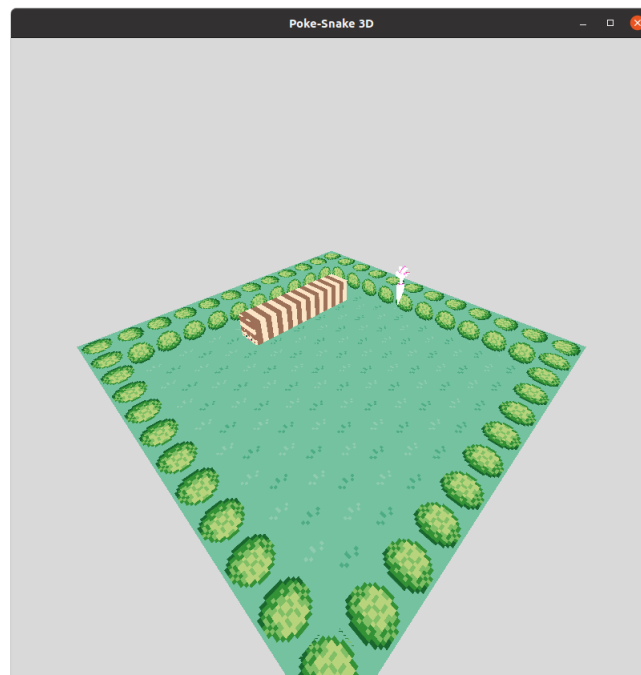


Figura 5: Snake tamaño 4 y mecánica de iluminación desde una vista en perspectiva

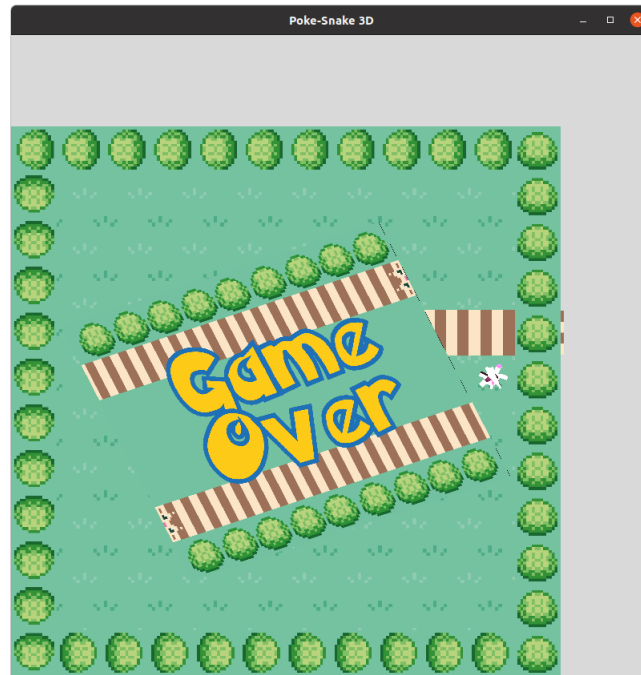


Figura 6: Final del juego por colisión de muro desde una vista 2D