Curso: CC3501-Modelación y Computación Gráfica para Ingenieros

Nombre: Javier Urrutia

Tarea 1B - Space War

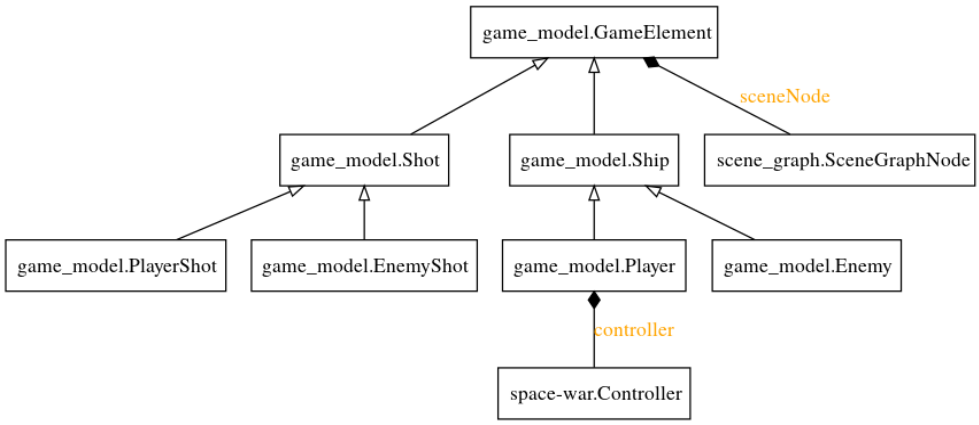
Solución propuesta

La solución propuesta para esta tarea se basa en el estilo de diseño Modelo-Vista-Controlador. Ésta implementación se realizó principalmente con programación orientada a objetos pues permite separar mejora cada pieza del juego e implementar las mecánicas pedidas.

En este trabajo se utilzan las librerías del curso *scene_graph, easy_shaders, basic_shapes* y *transformations,* y además se desarrollaron dos específicas para el juego. Estas librerías son *game_shapes* y *game_model*.

game_shapes contiene los modelos de los elementos del juego que son construidos como nodos para un grafo de escena. En éstos están la nave enemiga, la nave del jugador y el modelo de los disparos, todos hechos con polígonos. También contiene modelos con texturas como el *background*, y uno para cargar una imagen sobre la pantalla para mostrar *game over*. Las texturas fueron hechas con GIMP.

En *game_model* se implementa el modelo del juego. En primer lugar se crearon una serie de clases mostradas a continuación que representan cada elemento del juego. Como base cada elemento tiene una posición en el juego y un modelo visual dado por un nodo del grafo de escena. Luego por un lado, los disparos se diferencian en aquellos de los enemigos que se mueven hacia abajo y los del jugador que se mueve hacia arriba. Por otro lado, en las naves se diferencia en la del jugador que se mueve según el controlador, genera disparos *PlayerShot* y tiene 3 puntos de vida, mientras que las enemigas que se mueven según una trayectoria y generan disparos *EnemyShot* y solo tienen un punto de vida.



Todos estos elementos luego interactúan en una instancia de la clase *GameModel*. Ésta mantiene tres listas con los enemigos, disparos de enemigos y disparos del jugador en pantalla, además de una instancia del jugador. esta clase se encarga de llamar a los métodos que actualizan la posición de cada elemento y los que verifican colisiones entre los disparos y las naves. También se encarga de las naves enemigas de forma gradual en oleadas, hasta un máximo de 5 por cada una. Junto a esto, se genera una barra que representa la vida del jugador.

En el apartado visual ,el juego está implementado con *opengl* y *glfw*. Para dibujar los distintos elementos se creó la clase *ScreenDrawer* que maneja un *pipeline* para dibujar los elementos con texturas como el fondo y otro para dibujar los elementos que son solo polígonos. En éste se crea el efecto de movimiento infinito del fondo haciendo que dos copias de éste se deslicen hacia abajo en forma periódica. También dibuja todos los elementos presentes en la instancia de *GameModel* y cuando termina el juego dibuja la pantalla de *game over* o de juego ganado con un efecto de traslación hacia abajo.

Instrucciones de ejecución

Para ejecutar el juego se corre el script principal con python 3 de la siguiente forma (donde N especifica la cantidad de enemigos en el juego):

```
python space-war.py N
```

Los controles del juego son W[Arriba], S[Abajo], A[Izquierda], D[Derecha] y espacio[Disparar].

Resultados

El juego implementado se puede ver a continuación. Se puede ver que muestra la vida de la nave del jugador, la que puede recibir tres disparos antes de perder el juego. También el mensaje de *game over* que se desliza desde arriba. Además los enemigos pueden aparecen en tres colores distintos y se muestra un efecto de explosión cuando se destruye una nave.