CC3501 Modelación y Computación Gráfica
Profesor: Daniel Calderón
Aulmno: Nicolás García Ríos

Reporte de Documentación

27 de abril de 2020

Identificación

El presente documento describe la arquitectura del problema y solución de la Tarea 1 del curso CC3501 - "Modelación y Computación Gráfica para Ingenieros". La opción elegida para resolver fué la Tarea 1-b, la creación de un prototipo de videojuego "SpaceWars", el cuál fué resuelto con funciones de archivos .py de Auxiliares y cátedras del curso, y codificado en Jupyter Notebook.

Solución Propuesta

Para poder designar funciones, se crea una clase Ship() compartida tanto por la nave principal como la enemiga, la cuál tiene componentes básicos como vida, posición, posición del rayo, matriz hitbox, etc. Además de funciones Ray() para mover el ataque de los enemigos, Rayv1() y Rayv2() como ataques de la nave principal, Colision() para detectar los disparos de un enemigo en el hitbox propio, Movimiento() para el movimiento en loop de los enemigos y finalmente la función Restart() para generar nuevos enemigos a partir de los tres enemigos base.

Además, se creó el modelo de la nave principal y de las naves enemigas, como también el de los rayos laser que ambas partes podrán usar. Estas fueron hechas con transformaciones y creadas como gpuShapes. También, la nave principal se controla con las teclas WASD y ataca con la tecla espacio del teclado para mayor comodidad.

Para el movimiento del fondo y las imagenes de victoria y derrota se usan texturas con imagenes enviadas con el código. El fondo se define con una transformación que depende del tiempo y que va bajando, esta además viene seguida de la misma imagen de fondo pero invertida en torno al eje Y para una visualización más suave para evitar ver los límites de la imagen. Para las imagenes de victoria se utiliza una imagen simple con el template Victory Achived, el cuál se genera al destruir a todas las naves y si la nave principal no muere. Por otro lado, la imagen de derrota utiliza una imagen básica de Game Over en el caso que la nave principal prierda sus tres vidas.

Para el movimiento en general del juego, se utiliza el programa de Shaders al ser hecho con gpuShapes y no texturas. Este dibuja dependiendo del tiempo transcurrido y de la vida de cada nave las naves y los rayos disparados. En el caso de la nave principal, esta se dibuja solo si tiene toda su vida, miestras que para las tres naves enemigas, estas se irán dibujando dependiendo del N recibido del usuario y de si estas están muertas del todo.

La generación de enemigos continuos se solucionó, como se mencionó anteriormente, con la función Restart(). Primero se crean tres naves separadas y a medida que mueren, estas reinician en su punto de inicio bajando el contador y manteniendo la cantidad de enemigos en pantalla constante (tres).

Instrucciones de Ejecución

Como ya se mencionó anteriormente, la nave principal se mueve utilizando las teclas W-A-S-D en el lado izquierdo del teclado, además de utilizar la tecla espacio para disparar dos rayos a la

vez, esto para mantener un equilibrio de dificultad en el prototipo y no permitir disparos continuos todo el tiempo. En el caso de las naves, la cantidad depende de lo entregado por el usuario, y estas dispararán de forma continua mientras se mueven en un loop en la parte superior de la pantalla. Como detalle final, la nave principal tiene sus movimientos restringidos a solo dentro de la ventana.

Resultados

El Prototipo de videojuego logra cumplir con los objetivos puestos y este funciona de forma continua. A continuación se muestran screenshots de la ventana de ejecución del código en distintos momentos del juego, estos siendo durante la ejecución y jugabilidad, cuando el jugador logra destruir todas las naves enemigas y cuando el jugador pierde:

Figura 1: Fotocapturas del prototipo en distintas situaciones