Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

TRABAJO PRÁCTICO
TÉCNICAS DE GRÁFICOS POR COMPUTADORA

Docente: **Leandro Barbagallo**

Grupo: **BEFS**

Curso:

Cuatrimestre: **1°C 2015**

Año: **2015**

Integrantes:

- Halblaub, Braian 149.330-9
- Kalinowski, Eric 149.285-8
- Leira, Santiago 150.720-5
- Milano Pesce, Franco 149.156-8

Introducción:

Este juego consiste en un Sniper en primera persona que comienza en un bosque con un clima de nieve y ventoso. Se tiene un bosque de 2000 metros cuadrados, en los cuales se encuentra un conjunto de árboles y pastos que se mueven a partir del viento que hay. También hay barriles explosivos, un muro que limita el mapa y 20 enemigos.

Objetivo:

El objetivo del juego es poder eliminar a los 20 enemigos que aparecen a lo largo de todo el mapa, se tiene un número fijo de balas que es 10, las cuales se pueden ir recargando con la letra R. El jugador principal cuenta con las balas antes mencionadas y los barriles que se encuentran en el mapa para poder matar a sus enemigos.
Cuando el jugador principal mate a los 20 enemigos aparecerá un cartel con la leyenda "Ganaste".

Idea:

La idea del juego es que la persona que maneja al Sniper, pueda moverse a lo largo del mapa haciendo uso del contexto y de los elementos inmersos en el para lograr cumplir el objetivo. Que se pueda desenvolver en este mapa implica hacer uso de los barriles y de los árboles para beneficio propio

Controles:

W, A, S, D: Mover la cámara por el escenario
Mouse: Rotar la cámara
R: Recargar el arma.
Click izquierdo: Disparar
Click derecho: Zoom
Space Bar: Inicia el juego.

Escenario:

El escenario consta de los siguientes elementos:

- 500 árboles distribuidos mediante una función random.
- 250 meshes para el pasto.
- 30 barriles distribuidos.
- Un skybox para representar el cielo.

Enemigos:

Los enemigos aparecerán a lo largo de todo el mapa. Ellos permanecerán en una espera pasiva, hasta que nuestro personaje ingrese en un área de proximidad. A partir de ese momento, comenzaran a perseguirnos a lo largo del mapa hasta alcanzarnos, o volverán a estado pasivo si nos alejamos demasiado. Un disparo disminuye en 50 la vida de un enemigo.
Todos los enemigos poseen una velocidad variable haciendo que algunos sean más rápidos que otros mediante un random. También hay dos enemigos particulares que duplican el tamaño de los enemigos comunes.
Los enemigos comunes poseen una vida de 100, es decir que con 2 disparos mueren. En cambio los enemigos más grandes quintuplican la vida de estos, es decir 500, y morirían con 10 disparos.
Los 20 enemigos se irán acercando al personaje sacándole 5 de vida cada vez que le pegan al personaje principal.

Optimización:

La optimización la manejamos de la siguiente forma. Utilizamos una grilla regular para los objetos denominados estáticos ya que estos se distribuyen de manera uniforme a lo largo del mapa, y de esta forma cada celda de la grilla va a tener más o menos la misma cantidad de objetos. Por este motivo, seleccionamos este método en lugar de otro que quizás sea más complicado de implementar, como el KD-TREE por ejemplo, y que en este caso nos daría una optimización similar.
Primero se chequea si hay colisión entre una celda de la grilla y el frustum, si se llega a cumplir esta condición, se fija los elementos de esa celda.
Los enemigos están optimizados utilizando frustum culling porque al moverse, utilizar el método de las celdas seria poco eficiente, ya que es mejor aprovechado por elementos estáticos como se explicó anteriormente.
Por último, una optimización más sobre los elementos del mapa es que aquellos que se encuentren lejos, no se rendericen. Esto es posible ya que al disponer de un mapa extenso y en su mayoría cubierto por árboles, estos obstruyen la visión de los que están más lejos. Se obtiene una percepción muy similar si estuviesen o no esos elementos renderizados.

Shader:

Utilizamos Shaders cuando se realiza la colisión entre la bala y el barril se produzca una iluminación roja en el suelo que debido al color de la nieve se ve más claro. Esto le da un poco más de realidad, porque toda explosión debería producir una iluminación como efecto visual en el ambiente.

Colisiones:

Las colisiones en su mayoría están planteadas de BoundingBox a BoundingBox. Para el caso del personaje definimos un Box en la posición de la cámara, que al colisionar con otro elemento estático, como puede ser los árboles, barriles, límites del mapa, se redirecciona en la posición de un instante previo.
Una lógica similar fue utilizada para la colisión entre el Box de los enemigos con los elementos estáticos.
La otra colisión que tenemos es para las balas contra los barriles, los enemigos y demás elementos del mapa.
Para las balas usamos un Tgc ray o rayo, que tienen un alcance máximo, y se toma la primera colisión contra el BoundingBox de otro elemento, esto es así para que la superposición de elementos evite que el rayo le llegue a un enemigo si este esta obstruido, ya que no tendría sentido que por ejemplo si hay un enemigo detrás de otro se mueran ambos, solo debería morirse el primero que colisiona con la bala.
Para la colisión de las balas con los barriles lo que hicimos fue que cada vez que una bala colisiona con un barril que este mismo explote, creando un efecto visual y sonoro. Para el primer efecto lo que hicimos fue crear una esfera que va incrementando en tamaño por un determinado tiempo aplicándole una textura fuego, y para el segundo hicimos que cada vez que colisiona una bala con un barril se escuche un audio .wav . También lo que hicimos fue que cuando las balas colisionan con los barriles que los enemigos o bien el personaje mueran siempre y cuando se encuentran en un radio menor definido en el código, es decir si se encuentran en un entorno del barril cuando este explota deberían morirse.

Modifiers:

Utilizamos modifiers principalmente para poder calibrar las medidas relacionadas con los factores climáticos de una forma más sencilla. Así podemos ir probando y viendo la sensibilidad frente a los cambios de por ejemplo la caída de la nieve, el viento y la velocidad. Esto nos permitió poder darle un efecto más realista el aspecto del mapa con la interacción de los fenómenos climáticos.

Finalización del Juego:

La finalización del juego viene dada porque se le termina la vida al personaje como consecuencia del ataque de los enemigos, o por cumplimiento del objetivo, que vendría dado por lograr matar a los 20 enemigos distintos que se distribuyen a lo largo del mapa.