TRABAJO PRÁCTICO
TÉCNICAS DE GRÁFICOS POR COMPUTADORA

Docente/Tutor: Leandro Barbagallo

Grupo: RenderMan

Integrantes:

Azrak, Nicolás	146.656-2
Esquenazi, Brian	
Martínez Fornaso, Ignacio	146.802-9
Mazzeo, Gastón	146.865-0
Sierra, Emiliano	146.561-2

Cuatrimestre: 2°C 2014

Introducción:

Nuestro juego es un FPS dentro de un bosque. El objetivo es matar la mayor cantidad de enemigos posibles. Tenemos un arma con balas limitadas y varios barriles que al explotar matan a los enemigos que están cerca.

Controles:

W, A, S, D: Mover la cámara por el escenario
Mouse: Rotar la cámara
Click izquierdo: disparar
Click derecho: zoom
R: recargar
C: liberar el mouse

Escenario:

El escenario está compuesto de 500 árboles, 200 meshes que representan el pasto y 40 barriles explosivos. Además cuenta con un skybox representando el cielo.

Shader:

Utilizamos Shaders para la simulación del movimiento de los árboles y del pasto debido al viento. Tomando cada de vértice de los mismos y, sumando una función seno que depende del tiempo, la frecuencia y la dirección del mismo pudimos lograr un efecto bastante similar al real.

Enemigos:

El juego se divide en niveles, cada uno con una dificultad diferente. Inicialmente aparecerán pocos, serán más fáciles de vencer y difíciles de encontrar. A medida que vayamos avanzando el juego se tornará más intenso y empezaran a aparecer más enemigos. No solo serán más, sino que también les crecerán los brazos y serán más difíciles de derrotar.

Colisiones:

Las colisiones son usadas en el juego de varias maneras diferentes. Una de esa es para la IA de los enemigos, más específicamente para saber que camino recorrer sin chocarse con los árboles. En ese caso se usa una colisión BoundingSphere con BoundingCylinder.

Otra colisión es para los disparos. En este caso se usa el PickingRay representando una recta que el centro de la pantalla con el punto de intersección. Se puede testear el disparo contra un barril, un enemigo o la cabeza de un enemigo (para detectar el headshot!).

Optimización:

Para optimizar decidimos usar 2 métodos. Por un lado un Octree y Oclussion culling. Podría parecer que usar un KD-Tree hubiera sido mejor, pero en nuestro caso no nos servía demasiado, ya que los árboles son distribuidos regularmente quedando correctamente distribuidos, por lo tanto un KD-Tree no nos daría tanto beneficio, pero si tardaría mucho más tiempo su construcción, ralentizando el inicio del juego.

Con respecto al occlusion culling nos aprovechamos del Skybox al cual lo movemos para que siga al personaje tapando los elementos no visibles. Si bien no esta hecho desde el lado de la aplicación, usar el Skybox permite a la gpu tener que renderizar muchos menos triángulos.