

Proyecto

"Last Bullet"

Grupo

"Paradox Studios"

"DOCUMENTO DE ESPECIFICACIÓN"

Hito: 0

Fecha entrega: 04-10-2016

Versión: 1.0

Componentes:

- Zamora Pastor, Julio
- Rebollo Berná, Antonio
- Pérez Cristo, Rubén
- Muñoz Periñán, José Luis
- Moltó Ferré, Enrique

Contenido

Contenido

1. Introducción

1.1. Propósito

2. Descripción general

2.1. Ámbito del sistema (Contexto e historia).

2.2. Funcionalidades generales.

2.3. Características de los personajes.

2.4. Escenarios.

2.5. Requisitos (suposiciones y dependencias).

2.6. Restricciones.

2.7. Requisitos futuros.

3. Requerimientos específicos

3.1. Requerimientos funcionales.

3.1.1. Mecánicas.

3.1.1.1. De los jugadores

3.1.1.2. De objetos y NPCs

3.1.2. Técnicas y algoritmos a desarrollar.

3.2. Requerimientos no funcionales.

4. Apéndices

4.1. Referencias

1.

Introducción

1.1. Propósito

Nuestro propósito es crear un videojuego multijugador, que consista en un shooter en primera persona, divertido, dinámico y funcional.

2. Descripción general

2.1. Ámbito del sistema (Contexto e historia).

El videojuego se llamará Last Bullet, es un shooter multijugador en primera persona que se basará en las mecánicas de Quake 3, aunque tendrá algunas modificaciones.

En el videojuego se disparará con diferentes armas a otros jugadores, cuando la barra de vida de los contrincantes llegue a cero, morirá, se le dará un punto a aquel que lo ha eliminado y el jugador que ha sido eliminado reaparecerá en otra parte del mapa. Cuando un jugador alcance el máximo de puntos este ganará la partida.

2.2. Funcionalidades generales.

El juego será un juego competitivo player vs player (PvP), un usuario anfitrión creará una partida y la podrá iniciar cuando se alcance el máximo de jugadores. Se podrán unir a la partida otros jugadores, pero también existirá la posibilidad de que el anfitrión asigne jugadores controlados por la IA y jugar una partida contra la máquina.

Los jugadores podrán moverse en los tres ejes, disparar y saltar. Habrá diferentes tipos de armas que se encontrarán repartidas por el mapa.

Habrán tres tipos de armas disponibles: un rifle de asalto que tendrá una cadencia de fuego alta pero poco daño, una pistola que tendrá poca cadencia pero mucho daño y un lanzacohetes que tendrá poca cadencia pero hará daño en área.

Los jugadores recogerán paquetes de vida que habrá repartidos por el escenario.

La partida se acabará cuando uno de los jugadores alcance el límite de puntos, cada jugador ganará puntuación al matar a otro jugador.

2.3. Características de los personajes.

Las capacidades de todos los personajes que interactuarán en una partida serán las mismas, pero tendrán diferentes posibilidades de afrontar las situaciones en función del arma que tengan.

Esto queda más claro si ponemos un ejemplo. En un determinado momento el jugador tendrá la posibilidad de recoger un arma diferente del suelo, por ejemplo un arma lanzamisiles. De esta forma las habilidades de los personajes serán diferentes a lo largo de la partida. Es decir, si un enemigo tiene 2 armas una escopeta y un lanzacohetes aunque tenga equipado el lanzacohetes si se encuentra con un oponente en un rango cercano entonces el enemigo sacará la escopeta en vez de usar el lanzamisiles.

En cuanto a las habilidades que tendrán todos los personajes a lo largo de toda la partida estará el de recoger vida del suelo, recoger armas del suelo, desplazarse por el mapa, disparar, y hacer salto o doble salto.

Los roles serán los mismos en todos los personajes aunque el aspecto de cada personaje será diferente. El jugador podrá jugar competitivo contra otros jugadores 1vs1vs1vs1 aunque si no se completa el máximo de jugadores los restantes serán controlados por la máquina.

2.4. Escenarios.

Los escenarios serán cerrados, donde habrán obstáculos para poder esconderse del fuego cruzado de los distintos jugadores.

En principio por reducir complejidad los suelos serían planos. No obstante habrán plataformas a distintas alturas para hacer una jugabilidad más dinámica.

Además por el escenario se encontrarán diferentes tipos de elementos con los que podrás interactuar como son armas para recoger por el suelo o paquetes de vida.

2.5. Requisitos (suposiciones y dependencias).

El juego será realizado en ordenadores con Windows 10, por lo que será reproducido en equipos con este sistema operativo.

También Incorporaremos el juego en un dispositivo de realidad virtual como Oculus Rift.

2.6. Restricciones.

Lenguaje de programación será C++.

La resolución mínima será de 1024x720

Los controles se adaptarán dependiendo dependiendo de si se juega con mando o no.

2.7. Requisitos futuros.

Plataformas de salto colocadas por el mapa que provoquen un impulso y permita a los jugadores moverse a una distancia mayor.

Como requisito futuro tenemos pensado añadir granadas para darle un mayor dinamismo al juego y una mayor variedad de armamento.

Otro de nuestros requisitos futuros es implementar varios mapas y varios modos de juegos, entre ellos queremos realizar un modo de capturar la bandera.

3. Requerimientos específicos

3.1. Requerimientos funcionales.

3.1.1. Mecánicas.

- Moverse alrededor del mapa: con el teclado serás capaz de moverte en cualquier dirección en el plano XY
- Disparar a los enemigos: debido a que tenemos 3 tipos de armas los disparos se harán de manera diferente. Por un lado tenemos las armas que disparan balas (fusil de asalto y pistola) que calcularán la dirección del proyectil si se le da al jugador y por otro la que lanzan misiles (lanzacohetes), que calculará también la dirección del proyectil y dónde tiene impacto con el suelo para hacer el daño en área
- Recoger paquetes de vida: habrán spawns fijos de vida alrededor de mapa que aumentarán tu barra de vida.
- Después de morir habrá unos segundos de castigo y volverá a aparecer con la pistola principal.
- Doble salto: se podrá dar un salto en el aire después de realizar uno desde el suelo para llegar a más altura.
- Habrá armas repartidas por todo el juego, cuando un jugador coge un arma, esta desaparecerá del mapa pero posteriormente volverá a aparecer para que otro jugador pueda recogerla.

3.1.1.1. De los jugadores

Los jugadores se moverán en todas las direcciones, pudiendo dar un doble salto en el eje Z. El jugador disparará a los contrincantes con algunas de las diferentes armas que ofrecemos, cuando consiga bajar la vida de los enemigos a cero estos desaparecerán y el jugador conseguirá un punto.

3.1.1.2. De objetos y NPCs

Habrán paquetes de vida cuando un jugador pase por encima de uno de esos paquetes, se le sumarán puntos de vida a su cuenta y el paquete desaparecerá, pasado un tiempo el paquete volverá a aparecer en el mismo sitio para que los jugadores puedan recogerlo de nuevo.

La IA se comportará de la siguiente manera:

- a. En primer lugar los enemigos te buscarán por el mapa para intentar matarte, si estás demasiado cerca podrán escucharte y aunque estés por su espalda se girarán. Cuando te puedan ver te dispararán.
- b. Si tienen poca vida en vez de buscarte a tí, buscarán los paquetes de vida, recorriendo el mapa en busca de estos. Si detectan que estás cerca de ellos huirán siempre que puedan y si no tendrán que luchar contra ti.
- c. Tendrán un arma por defecto que será la misma que la del jugador, este primer arma es la peor de todas así que si detectan que cerca hay algún arma mejor irán a por ella antes de buscarte a ti, siempre y cuando no estén en mitad de un tiroteo.
- d. Si el enemigo tiene varias armas podrá elegir entre ellas como mejor le convenga, es decir si tiene un lanzacohetes y una escopeta y los jugadores están muy lejos entonces elegirá el lanzacohetes en vez de la escopeta.

3.2. Técnicas y algoritmos a desarrollar.

Utilizaremos algoritmo de IA que permita que los enemigos puedan vernos y escucharnos y, sean capaces de pensar si les conviene encarar a un enemigo, esquivar o huir en busca de vida en caso de que sea necesario.

Por ello utilizaremos sistemas de fine state machine, logica difusa, behaviour trees, trigger de eventos y percepción sensorial.

Algoritmos de IA de pathfinding estratégico, sistemas de waypoints y navmeshes.

También crearemos un motor de red que permita conectarnos a varios jugadores a la misma vez en multijugador. Utilizaremos un middleware de red como RakNet.

El juego al ser en 3 dimensiones, utilizaremos un motor de físicas en 3D como bullet3D o Physx que permitirá generar todas las colisiones de los personajes y de las balas.

3.3. Requerimientos no funcionales.

Rendimiento de la aplicación. Al ser multijugador hay que intentar que el tiempo de respuesta de la aplicación sea mínimo.

Tasa fija de frames por segundo. Desde el equipo de desarrollo nos comprometemos a intentar conseguir una tasa de fotogramas estable que en principio situaríamos en 60 fps.

Aplicación compatible con varios sistemas operativos. No usaremos código dependiente del sistema operativo en el que desarrollemos con el objetivo de poder abarcar más plataformas.

Aplicación fiable y estable. La versión final de la aplicación debe estar libre de bugs, aunque no tendremos en cuenta temas de seguridad porque la aplicación no lo requiere.

4. Apéndices

4.1. Referencias

Nuestro juego se basa en el Quake: <https://www.youtube.com/watch?v=PitACVUaz8c>