

Proyecto Last Bear Standing

Grupo Estudio Rorschach

"DOCUMENTO DE ESPECIFICACIÓN"

Hito: 0

Fecha entrega: 06-10-2016

Versión: 1

Componentes:

- Miguel Paniagua Muela
- Miguel Córdoba Alonso
- José María Ortiz García
- José Roberto Martínez Gras
- Jorge Puerto Esteban
- Manuel Gómez Cámara

Contenido

Contenido	1
1. Introducción	2
1.1. Propósito	2
2. Descripción general	2
2.1. Ámbito del sistema (Contexto e historia).	2
2.2. Funcionalidades generales.	2
2.3. Características de los personajes.	3
2.4. Escenarios.....	4
2.5. Requisitos (suposiciones y dependencias).	5
2.6. Restricciones.	5
2.7. Requisitos futuros.	5
3. Requerimientos específicos	6
3.1. Requerimientos funcionales.	6
3.1.1. Mecánicas.....	6
3.1.1.1. De los jugadores	6
3.1.1.2. De objetos y NPCs	6
3.1.2. Técnicas y algoritmos a desarrollar.....	6
3.2. Requerimientos no funcionales.	8
4. Apéndices	9
4.1. Referencias.....	9

1. Introducción

1.1. Propósito

A continuación, se expondrá en detalle la **contextualización del videojuego**, su papel en el panorama actual de los videojuegos e historia, **características, contenido, funcionalidades, requisitos y restricciones** de éste. Además, se hará un desglose de los **requerimientos funcionales y no funcionales** necesarios para su funcionamiento, **mecánicas** tanto de personajes como de los NPC y por último los **algoritmos a desarrollar** para el videojuego.

2. Descripción general

2.1. Ámbito del sistema (Contexto e historia).

El nombre que se ha elegido para este videojuego es “**Last Bear Standing**”, haciendo referencia a cómo el jugador encarnará a un osito de peluche que tomará parte en una guerra frenética.

Last Bear Standing se enmarca en los géneros **shooter 2D, multijugador LAN, plataformas y frenético**.

Las **mecánicas** básicas del *Last Bear Standing* están basadas en las del videojuego *Duck Game*, que consisten en que cada jugador puede desplazarse por el escenario, saltar, agacharse, hacerse el muerto, deslizarse, disparar e interactuar con los distintos objetos que componen el mapa.

Las características que distinguen a este juego son lo **frenético** de éste, las partidas son **rápidas, caóticas** y cargadas de **confusión**, consiguiendo así que la atención del jugador se desborde. Además, dada la libertad de acción de cada jugador y lo distinto de los escenarios entre sí, cada partida es completamente distinta a la anterior.

HISTORIA

Encarnas el papel de un oso de un **oso peluche**. Harto de compartir la atención del niño al que llamáis *El Amo* con el resto de los tuyos, te adentrarás en una dura batalla por ser su peluche preferido, solo puede quedar uno ¿Serás tú?

Desde el momento que decidas formar parte de esta masacre, deberás recorrer el escenario cuidadosamente, evitando trampas y ataques de los otros peluches en busca de todo tipo de armas que puedan ayudarte a conseguir tu objetivo, acabar con tus oponentes antes de que ellos acaben contigo y ser el único superviviente del escenario e **ir acumulando así amor** del *Amo*.

En acabar la cruenta batalla a lo largo de los escenarios que abarcarán todas las partes de la casa, solo aquel que haya **acumulado la mayor cantidad de amor** será merecedor de la atención y podrá ser el oso favorito.

2.2. Funcionalidades generales.

Las funcionalidades del juego están divididas según los elementos del propio juego. Por un lado, los jugadores podrán hacer uso de las funcionalidades y mecánicas de movimiento y acción de las armas.

Las armas tendrán distintas funcionalidades dependiendo del tipo del arma, ya sea por la cadencia, el alcance, si es automática o manual o si es de tipo explosiva, dependiendo en este caso si es una granada, una mina o un cohete.

En cuanto a los escenarios encontremos también distintas funcionalidades dependiendo del tipo de escenario, como distintos rozamientos y otros modificadores de físicas, así como elementos concretos.

2.3. Características de los personajes.

Todos los **personajes** que conforman el videojuego son **ositos de peluche**. Cada uno de los peluches lleva puesto una **máscara que lo identifica** y lo diferencia del resto, es decir, es un mero componente visual.

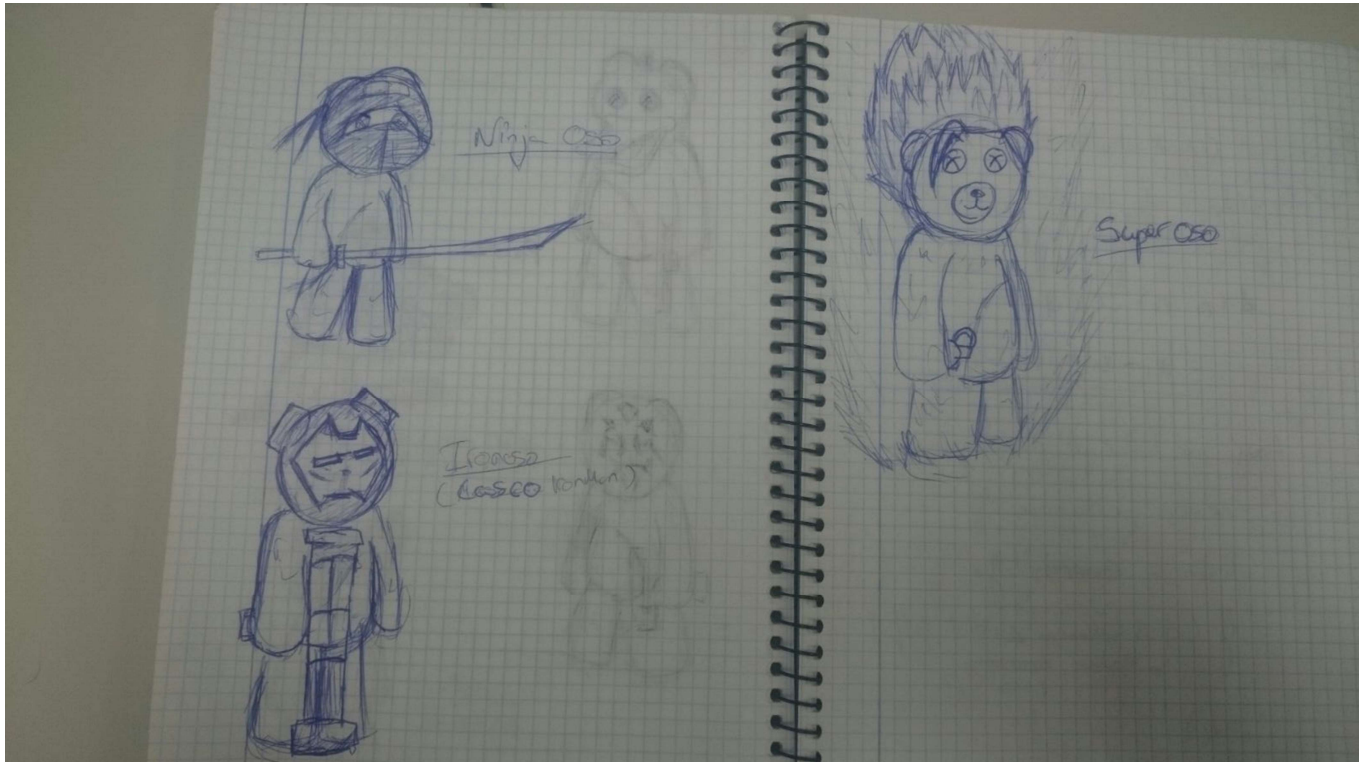
Las **capacidades**, independientemente de su aspecto son las mismas para todos los personajes y jugadores (detalladas en el punto 2.1). Tanto el rol como el **comportamiento** de los personajes con IA es aleatorio e independiente de su aspecto.

La única **propiedad** que tiene cada personaje es la del componente “Vida”, cuyo valor puede cambiar dependiendo de objetos en uso durante una partida. Los objetos con los que el jugador puede interactuar tienen sus propias características y se convierten en un elemento diferenciador.

Como ya se ha mencionado, existen **variaciones de aspecto** para los peluches y sirven de identificativo para el jugador. Las variaciones son las siguientes:

Primeros bocetos





2.4. Escenarios.

El videojuego consta de **seis escenarios**, cada uno asociado a una de las rondas o partidas en un juego. Los escenarios representan zonas de la casa y cada uno de ellos tendrá sus propias **trampas**, disposición de **objetos** con los que interactuar, **plataformas**, generadores de **armas** y modificadores de las **físicas** (viento, menos rozamiento, etc.).

Los escenarios que componen el juego son los siguientes:

- **Habitación:** la conforman elementos típicos de la habitación de un niño, como juguetes, basura, decoración infantil, etc. Este escenario mantiene las físicas naturales del juego, sin añadir ningún modificador.
- **La cocina/nevera:** está compuesta por elementos encontrados en un refrigerador, como puede ser: comida, bebidas, tupperwares, bricks, etc. Este escenario modifica el coeficiente de rozamiento haciéndolo más deslizante.
- **El Lavabo:** lo conforman elementos típicos de un lavabo, entre ellos jabón de manos, cepillo de dientes, grifo, etc. Este escenario tiene el desagüe como elemento hostil significativo, capaz de absorber al jugador.
- **El Jardín:** este escenario introduce el viento como modificador de físicas, añadiendo una fuerza lateral constante. Este mapa está compuesto de flores, macetas, bichos, césped, regaderas, etc.
- **El Salón:** lo conforman elementos como mando de televisión, revistas, muebles, etc. Las físicas se mantienen sin modificar.

- **La Oficina:** es la zona de los padres, un escenario lúgubre y especialmente hostil con objetos como archivadores, bolígrafos, flexos, libros, etc. Este escenario juega con la iluminación de la escena.

Ejemplo de escenario:



2.5. Requisitos (suposiciones y dependencias).

El sistema operativo empleado en el desarrollo del proyecto será **Windows**. Se hará uso de **Photoshop** para el tratamiento y edición de imágenes y **AfterEffects** y **Premiere** para la edición de vídeo. En cuanto al desarrollo del videojuego necesitaremos **NetBeans** como entorno de desarrollo y **Box2D** como motor de físicas del juego.

En cuanto al tratamiento de gráficos se hará uso de nuestro propio motor gráfico para el juego basándonos en alguno existente como **Ogre**, **Irrlicht** o **Unity**. Por último, en el apartado de realidad virtual se empleará **3DS Max**, **Blender** o **Maya** para el modelado.

2.6. Restricciones.

El proyecto está estrictamente limitado a un número de seis componentes y a una cantidad de tiempo limitada a dos cuatrimestres para su desarrollo. El videojuego debe hacer uso de modelos 3D con mecánicas 2D y multijugador LAN.

2.7. Requisitos futuros.

Como requisitos futuros necesitaremos hacer uso de un servidor para que el juego pueda ser completamente online. También sería interesante alguna plataforma web en la que los jugadores fueran capaces de comparar logros y puntuaciones, así como la compra y descarga de mapas y skins de personaje.

3. Requerimientos específicos

3.1. Requerimientos funcionales.

3.1.1. Mecánicas.

3.1.1.1. De los jugadores

Los jugadores podrán realizar las siguientes acciones:

- Disparar: cuando el jugador tenga un arma en la mano y ésta esté cargada, podrá realizar un disparo en la dirección en la que se encuentre mirando.
- Saltar: cuando el jugador se encuentre encima del suelo o de un objeto, podrá realizar un salto.
- Recoger un arma: cuando el jugador no tenga ningún arma en posesión, podrá recoger cualquier arma que se encuentre en el mapa siempre y cuando sea el primero en cogerla y se encuentre dentro del área cogible de esa arma.
- Soltar un arma: cuando el jugador tenga alguna arma en posesión, podrá soltar dicha arma en la dirección que lleve.
- Agacharse: el personaje podrá agacharse en cualquier momento para esquivar balas.
- Hacerse el muerto: el personaje podrá simular estar muerto para confundir al resto de jugadores y evitar que le ataquen. Aunque se simule estar muerto, si colisiona una bala con el personaje morirá igualmente.
- Deslizarse: cuando el jugador se mueve en una dirección y acto seguido se agacha, el personaje seguirá la trayectoria una determinada distancia o hasta que colisione con un objeto del mapa.

3.1.1.2. De objetos y NPCs

Existirán diferentes tipos de armas, para las cuales existirán mecánicas comunes y mecánicas específicas para cada tipo de arma.

Mecánicas generales de las armas:

- Aparición del arma: todas las armas estarán disponibles en una posición del mapa predeterminada (*spawn*) y el primer jugador que la coja la podrá usar. Pasado un tiempo, aparecerá una nueva arma en el spawn.
- Desaparición del arma: cuando el jugador se deshace de un arma tirándola al suelo, si su cargador está vacío, el arma desaparecerá al cabo de un tiempo (se puede seguir cogiendo a pesar de tener el cargador vacío).
- Cargador: algunos tipos de armas dispondrán de un cargador de balas que variará en función del tipo de arma. Las balas disparadas seguirán la trayectoria en la cual este apuntando el jugador hasta que haya colisión o llegué a rango máximo del arma (este rango variará en función del tipo de arma).
- Desviación del arma: todas las balas disparadas saldrán con una desviación aleatoria que variará en función del tipo de arma usada.
- Cadencia: cada arma tendrá una cadencia (tiempo entre dos disparos seguidos) que variará en función del tipo de arma.

Mecánicas específicas de cada arma:

- Pistola: las pistolas tendrán un cargador de entre 1 y 9 balas, un rango medio de distancia de disparo y serán semiautomáticas (teniendo que pulsar el botón de disparo cada vez que quieras realizar un disparo).
- Escopetas: las escopetas dispararán en cono con un rango corto de distancia de disparo, tendrán un cargador con entre 5 y 10 balas. Existirán dos tipos de escopetas:
 - Manuales: para disparar será necesario recargar el arma (volver a dar al botón de disparar) salvo si es el primer disparo, ya que cuando la coges del spawn aparecerá cargada.
 - Semiautomática: el jugador solamente tendrá que pulsar el botón de disparar sin necesidad de recargar el arma.
- Fusiles: son armas de medio/largo alcance, con muchas balas (entre 25 y 35 balas). Son armas automáticas, por lo que el jugador podrá mantener el botón de disparar pulsado de manera continua.
- Rifle francotirador: armas de muy largo alcance, con pocas balas y muy poca desviación (prácticamente nula) con una cadencia muy alta, existirán dos tipos:
 - Francotirador de cerrojo: este rifle será necesario hacer una recarga cada vez que el jugador realice un disparo, salvo si es el primer disparo (como la escopeta manual).
 - Francotirador semiautomático: el jugador solamente tendrá que pulsar el botón de disparar sin necesidad de recargar el arma.

Mecánicas de explosivos:

Dentro del juego existirán algunas armas explosivas, las cuales se comportan de forma totalmente diferente a las armas que usan balas y de las cuales ningún personaje es inmune a la explosión.

- Granada: el jugador podrá recoger granadas en los spawns de arma correspondiente y ésta no explotará hasta que el jugador la active con el botón de disparar, una vez activada la granada, el jugador tendrá un tiempo limitado hasta que explote. Dentro de este tiempo se recomienda lanzarla y alejarse.
- Lanzagranadas: es un arma que puede lanzar granadas que explotan al colisionar con algún jugador o pasado el tiempo de explosión.
- Lanzacohetes: arma que disparará un cohete, su funcionamiento es similar al de una pistola, salvo que la bala disparada explotará al colisionar con cualquier cosa.
- Minas: el jugador podrá recoger minas de los correspondientes spawns y tendrá que activarlas para luego dejarlas en el suelo. Las minas explotarán al entrar en contacto con un jugador u otro objeto cogible.

Mecánicas de los objetos:

- Dentro del juego existirán distintos objetos, como cajas que podrán ser cogidos por los jugadores, estas cajas harán a la vez de escudo bloqueando las balas que colisionen con ellas.
- Muelles, que impulsaran a los personajes cuando interactúen con ellos.
- Existirán también elementos que podrán matar a los personajes cuando colisionen con ellos, como por ejemplo unos pinchos.

4. Técnicas y algoritmos a desarrollar.

Uno de los primeros pasos será la implementación de un motor de red que permita que nuestro juego pueda correr en una red local en tiempo real. Para ello debemos crear los algoritmos necesarios de forma que el cliente suministre los elementos necesarios para su computación y devuelva al cliente los elementos cambiantes del juego de modo que la información sea mínima. Intentaremos añadir capacidades de predicción en el cliente de forma que éste pueda realizar algunos de los cálculos y evitar la sobrecarga de paquetes en la red.

Para el desarrollo de las físicas utilizaremos el motor Box2D, que nos aporta todas las técnicas necesarias para implementar las físicas de nuestro mundo 2D. Haremos uso de este motor para desarrollar los sistemas de trazado de rayos, uso de joints, modificación de los distintos parámetros del mundo en ejecución, etc.

Nuestra IA correrá a cargo de unos bots que podrán ejecutar las mismas acciones que el resto de jugadores existiendo varias dificultades. Como queremos que las acciones de éstos sean lo más impredecibles posible haremos uso de algoritmos con lógica difusa, ya que creemos que es lo más adecuado a nuestras necesidades, puesto que la lógica difusa crea un comportamiento poco predecible al tener muchas combinaciones entre las distintas variables de acción.

Además, nuestro bot debe servirse de un sistema de búsqueda de caminos lo más óptimo posible, evitando los peligros de los jugadores para llegar hasta el arma más próxima corriendo los mínimos peligros posibles.

Para la definición de los mapas debemos crear un editor de niveles, de forma que podamos cargar nuestros mapas escritos en un formato propio de xml, en el que se especifique las posiciones de los objetos del mapa, los spawns y los distintos modelos usados para cada uno de los elementos del mapa.

También debemos implementar un algoritmo para el seguimiento de la cámara, de forma que siempre centre a los personajes vivos teniendo en cuenta los dos más alejados. De esta forma iremos haciendo *zoom in* si están muy próximos entre si y *zoom out* si se alejan.

Haremos uso de las técnicas de *rigging* para animar a los personajes que previamente modelaremos de forma que sea lo más natural posible para un oso de peluche.

Además, utilizaremos el sistema de captura de movimiento para grabar algunas de las animaciones que utilizarán nuestros personajes.

Utilizaremos técnicas de edición, montaje y postproducción para la creación de elementos visuales como el cartel, el tráiler, el menú, etc.

Crearemos nuestro propio motor gráfico, y utilizaremos diversas técnicas para su optimización. Añadiremos distintos niveles de detalle para nuestros modelos, de forma que podamos elegir la calidad visual de los elementos del juego.

4.1. Requerimientos no funcionales.

Dentro de los requisitos no funcionales tenemos pensado implementar un sistema de partículas que mejorará el aspecto visual del juego. Este sistema de partículas lo usaremos para las colisiones de las balas con el entorno o con los personajes de forma que cuando se dispare se aprecie el casquillo o si los personajes mueren suelten algodón. También se puede

usar este sistema de partículas para recrear el clima (viento, hielo o lluvia) de las distintas localizaciones.

Además, implementaremos un sistema para recrear vegetación, de forma que podemos apreciar en algunos escenarios césped u otras plantas y como son medidas por las condiciones climáticas del entorno.

5. Apéndices

5.1. Referencias

<https://www.youtube.com/watch?v=yvUUwUIHPA8>

<https://www.youtube.com/watch?v=ojcdol3bRj8>