**Proyecto**

Teddy Wars

**Grupo**

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](#_Toc462657844)

[1. Introducción 2](#_Toc462657845)

[1.1. Propósito 2](#_Toc462657846)

[2. Descripción general 2](#_Toc462657847)

[2.1. Ámbito del sistema (Contexto e historia). 2](#_Toc462657848)

[2.2. Funcionalidades generales. 2](#_Toc462657849)

[2.3. Características de los personajes. 2](#_Toc462657850)

[2.4. Escenarios. 2](#_Toc462657851)

[2.5. Requisitos (suposiciones y dependencias). 2](#_Toc462657852)

[2.6. Restricciones. 2](#_Toc462657853)

[2.7. Requisitos futuros. 2](#_Toc462657854)

[3. Requerimientos específicos 2](#_Toc462657855)

[3.1. Requerimientos funcionales. 2](#_Toc462657856)

[3.1.1. Mecánicas. 2](#_Toc462657857)

[3.1.1.1. De los jugadores 2](#_Toc462657858)

[3.1.1.2. De objetos y NPCs 2](#_Toc462657859)

[3.1.2. Técnicas y algoritmos a desarrollar. 2](#_Toc462657860)

[3.2. Requerimientos no funcionales. 3](#_Toc462657861)

[4. Apéndices 3](#_Toc462657862)

[4.1. Referencias 3](#_Toc462657863)

# Introducción

## 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.

# Descripción general

## Ámbito del sistema (Contexto e historia).

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

*Teddy Wars* se enmarca en los géneros ***shooter 2D, multijugador LAN, plataformas y frenético***.

Las **mecánicas** básicas del *Teddy Wars* están basadas en las del videojuego *Duck Game*, las cuales 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 ser el **peluche de un niño**. Harto de compartir la atención de “El Amo” con el resto de los tuyos, te adentrarás en una dura batalla por ser el peluche preferido de “El Amo”.

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

En acabar la cruenta batalla a lo largo de la casa de “El Amo” solo aquel que haya **acumulado la mayor cantidad de amor** será merecedor de la atención de “El Amo”.

## Funcionalidades generales.

## 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 aspectual.

Las **capacidades**, independientemente de su aspecto, son las mismas para todos los jugadores, por tanto, para todos los personajes (detalladas en el punto 2.1). Tanto el rol como el **comportamiento** de un personaje con IA en un juego 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



## 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 de “El Amo”, cada escenario 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 de “El Amo”** – La conforman elementos típicos de una habitación de un infante, como por ejemplo: juguetes, basura, decoración infantil, etc. Este escenario mantiene las físicas naturales del juego, sin añadir ningún modificador.
* **La 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: jabón de manos, cepillo de dientes, grifo, etc. Este escenario tiene el desagüe como elemento hostil significativo, capaz de absorber al jugador.
* **El Jardín** – Hallamos elementos como: flores, macetas, bichos, césped, regaderas, etc. Este escenario introduce el viento como modificador de físicas, añadiendo una fuerza lateral constante.
* **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 de “El Amo”, 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:

## 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 video. 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 ultimo en el apartado de realidad virtual se empleará **3DS Max**, **Blender** o **Maya**, en el modelado.

## 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 un entorno 2D y multijugador LAN.

## Requisitos futuros.

POR HACER

# Requerimientos específicos

* 1. Requerimientos funcionales.
     1. Mecánicas.
        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 ese 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 no te ataquen. Aunque simules estar muerto, si te colisiona una bala mueres 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.
  + - 1. De objetos y NPCs

*<Mecánicas de entidades y objetos controlados por el juego>*

Existirán diferentes tipos de armas, para las cuales existirán mecánicas comunes y luego especificas 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 3 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 rango corto de distancia de disparo y tendrán un cargador con entre 5 y 10 balas. Existirán dos tipos de escopetas: manuales y semiautomáticas.
  + 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.
* Rifle automático: Son armas de largo alcance, con muchas balas (entre 15 y 30 balas). Además son automáticas, por lo que el jugador podrá mantener el botón de disparar pulsado de manera continua.
* Rifle francotirador: Son armas de muy largo alcance, con pocas balas y muy poca desviación (prácticamente nula), con una cadencia muy alta y 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 limitado se recomienda lanzarla y alejarse.
* Lanzagranadas: Es un arma que puede lanzar granadas que explotan al colisionar con algún objeto colisionable.
* 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.

Mecánicas de los objetos:

Dentro del juego existirán distintos objetos, como cajas que podrán ser cogidos por los jugadores, bloquearán las balas que colisionen con las cajas.

Muelles, que impulsaran a los personajes cuando interactúen con ellos.

# Técnicas y algoritmos a desarrollar.

Para las físicas vamos a usar un motor de físicas ya implementado que será Box2D.

Para la IA hemos pensado implementar un bot que haga uso de la lógica difusa ya que creemos que es el sistema de IA menos predecible y la mejor opción para nuestro juego.

Para los mapas usaremos algún programa tipo tiled.

* 1. Requerimientos no funcionales.

Implementaremos algún sistema de partículas para las colisiones de las balas, casquillos, algodón de los ositos.

Implementación de césped, que se meza con el viento.

# Apéndices

* 1. Referencias

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

<https://www.youtube.com/watch?v=ojcdoI3bRj8>