|  |
| --- |
|  |
| Idea trabajo semanal |
|  |
| Pirate Pwnage! |

**Hurricane**

19 de diciembre de 2013

Autor: Aris

Contenido

[1. Resumen 2](#_Toc375252038)

[1.1. Concepto 2](#_Toc375252039)

[1.2. Jugabilidad 2](#_Toc375252040)

[1.3. Funcionalidad 2](#_Toc375252041)

[1.4. Estética 2](#_Toc375252042)

[1.4.1. Inspiración 3](#_Toc375252043)

[1.5. Música 2](#_Toc375252044)

[2. Diseño 1](#_Toc375252045)

[2.1. Concepto general 1](#_Toc375252046)

[2.2. Mapa 1](#_Toc375252047)

[2.3. Barcos 1](#_Toc375252048)

[2.4. Sonido 2](#_Toc375252049)

[2.5. Interfaz 2](#_Toc375252050)

[3. Desarrollo 2](#_Toc375252051)

[3.1. Consideraciones generales 2](#_Toc375252052)

[3.2. Funcionalidad 2](#_Toc375252053)

[4. Otras cosas 2](#_Toc375252054)

Idea trabajo semanal

Pirate Pwnage!

# Resumen

## Concepto

Videojuego en el que controlas a un barco en un escenario naval, pero con obstáculos. Cada barco, incluyendo los enemigos, puede llevar dos armas, una principal y una secundaria, y cada una tiene efectos diferentes. Todas las armas principales se disparan automáticamente, pero no las secundarias. Un arma secundaria puede ser un EMP y otra por ejemplo una invulnerabilidad temporal.

## Jugabilidad

Cada jugador controla el movimiento de su barco con el teclado y dispara su arma secundaria con un botón (probablemente espacio o enter). Hay un botón de menú que permite cambiar las armas, pero si se hacen cambios se tardan unos segundos en recuperar el control del barco tras ellos.

## Funcionalidad

Cada arma se programa en un script diferente, y cada barco va cambiando de componentes según se decida. Las armas tienen valores como ratio de fuego, distancia máxima y daño. Las armas especiales van completamente aparte, aunque también son scripts separados.

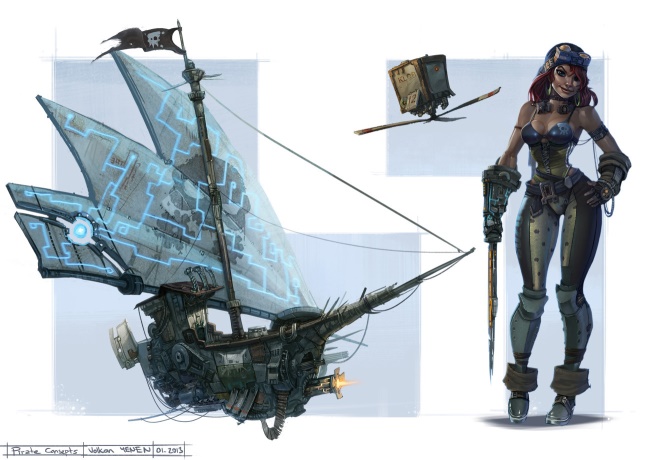
Con tener 4-5 armas diferentes principales y 3-4 armas secundarias disponibles es más que suficiente. Algunos ejemplos de armas principales pueden ser cañones (lentos pero muy destructivos), láseres (muy rápidos pero poco daño por golpe y con menos rango), catapultas (mucho rango y daño pero muy lentas) o ametralladoras (bastante rápidas y daño medio). Como armas secundarias y como ejemplo podrían ser escudos (invulnerabilidad unos segundos), cohetes de propulsión (mayor velocidad unos segundos) o PEM (desactiva las armas de los barcos en rango unos segundos).

## Estética

Estética pirata pero futurista, con cohetes, naves a propulsión, etc. Mezclando elementos piratas más clásicos con elementos steampunk o futuristas directamente. Un poco “mashup”, mezcla sin sentido.

### Inspiración

Guia visual de estilos para los barcos.



Inspiración para las armas.



Inspiración para el ambiente.



## Música

Música de fondo con temática pirata, ruidos de cañonazos para las armas y algún sonido especial para las armas especiales.

# Diseño

## Concepto general

El juego pretende ser un juego de acción muy sencillo (tipo MOBA, como puede ser el LoL, DotA 2, etc.) con un escenario costero en el que se encuentran varios lugares de reposo. El objetivo es ser el último barco en pie en cada ronda.

En cuanto al diseño de control del juego, nos basamos en algo muy simple, con unas teclas de movimiento “estándar” (como podrían ser las teclas WASD o las flechas de dirección) y un único botón de interacción para disparar las armas secundarias.

Se busca un componente estratégico poco profundo mediante el cual eligiendo un barco diferente, un arma principal y un arma secundaria diferente se abran nuevas posibilidades de juego. Así por ejemplo elegir un barco más lento pero con más aguante junto con armas de corto alcance puede ser una buena combinación con un arma secundaria que te permita llegar antes a tus enemigos, como un propulsor. Otra combinación puede ser un barco ligero y veloz junto con armas de gran alcance y un arma secundaria de invulnerabilidad temporal.

## Mapa

El mapa es una parte muy importante del juego, pues define varios aspectos clave. El mapa incluirá obstáculos que no se podrán atravesar, amplias zonas que propiciarán el combate directo, recovecos por los que huir y zonas de descanso donde no se podrá atacar a otros jugadores y donde se permitirá elegir otro barco/arma principal/arma secundaria. Estas últimas zonas serán muelles o astilleros, y tendrán un modelo único estándar para reconocerlos fácilmente.

Hay que tener en cuenta a la hora del diseño de mapas que el jugador solo verá una porción del mapa por la posición de la cámara, y que para explorar otras zonas deberá moverse hasta ellas.

# Barcos

Los barcos representan al jugador, y por tanto son el recurso más importante del juego. Se debe buscar que tengan todos una estética única e icónica (no obstante esto no debe entrar en conflicto con el ahorro de esfuerzo que se describirá más adelante).

Los barcos deben tener ciertos puntos en común unos con otros, al menos los modelos. Estos puntos en común son que no debe tener velas y/o mástil, al menos en la cubierta principal. Es decir, la cubierta principal debe estar libre para poder colocar en el centro el arma principal. El siguiente punto en común es la zona trasera, que debe ser en todos donde se coloque el arma secundaria. Esta zona debe ser más diferenciada que las demás y poder alojar siempre un modelo que en todos los casos envolverá parte de la misma.



Al tener los barcos dos puntos donde se deben colocar otros modelos, es necesario que en todos los modelos se mantenga una forma estándar que implique una parte delantera y una parte trasera diferenciadas, una cubierta libre de cosas que puedan interferir, etc.

En cuanto al diseño del modelo o “prefab”, se necesitarán como mínimo 3 cosas:

* El modelo del barco en sí, posicionado dentro de un GameObject que variará sus coordenadas (las del propio GO, no las del modelo del barco, que deberían ser siempre 0,0,0).
* Un GO vacío que sería el punto donde colocar el arma principal y que se coloca siempre en la cubierta del barco.
* Un último GO vacío que se coloca en la parte trasera del barco y que sería donde se colocaría el arma secundaria.

# Sonido

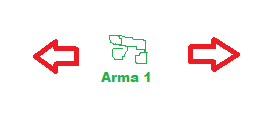
El diseño del sonido del juego es puramente funcional. La intención es sonorizar todo el juego cubriendo los siguientes aspectos:

* Música de ambiente. Una única o varias melodías aleatorias.
* Sonidos de disparos para cada arma principal.
* Sonidos de impacto para algunas armas.
* Sonidos de disparo de las armas secundarias.
* Sonidos para menús y selecciones.

# Interfaz

El diseño de menús e interfaz del juego debe ser lo más austero posible para ahorrar costes. Se representará la vida de cada barco con un sprite encima del mismo, puesto de forma horizontal y agotándose a medida que se pierde salud (la misma funcionalidad que en el anterior proyecto de desarrollo). El diseño de menús se hará con texto y sin mostrar apenas información más allá de la estrictamente necesaria para entender la opción elegida. No se podrá pausar el juego ni modificar el volumen/sonido. Las únicas opciones del menú serán continuar jugando (salir del menú), salir a la sala de espera o salir del juego.

En puntos especiales del mapa (los astilleros) se permitirá acceder a un menú especial pulsando una tecla designada que permitirá cambiar de barco, arma principal y arma secundaria. Este menú ensombrecerá toda la pantalla y tendrá 3 elementos de la GUI interactivos mas un botón para salir.



# Desarrollo

# Consideraciones generales

Lo primero a tener en cuenta es que en este proyecto va a primar el desarrollo lo más rápido posible reutilizando assets y scripts de otros proyectos y/u otras fuentes. Se recomienda el uso de la comunidad “[unify](http://wiki.unity3d.com/index.php/Main_Page)” para buscar scripts y de la “[asset store](https://www.assetstore.unity3d.com/)” u otras fuentes para modelos con los que trabajar al menos como punto de partida.

En cuanto a dificultades técnicas, este proyecto debería ser bastante sencillo de implementar. La única complicación a nivel de scripts podría ser el script de movimiento para hacer que el barco tuviera que girar desplazándose hacia adelante siempre, mientras que otra de las dificultades podría ser el cambio de prefabs en una jerarquía en tiempo de ejecución. Ambos problemas no deberían ser difíciles de solventar, sobre todo teniendo la ayuda de internet para buscar soluciones de terceros.

El otro problema, de mayor envergadura, es la cantidad de assets a generar y su variedad. Para paliar este problema, optaremos por reusar assets en la medida de lo posible de otras fuentes. Simplemente con el hecho de obtener un modelo de barco y adaptarlo ligeramente para cambiarle la textura un poco, eliminar el mástil etc., podemos obtener resultados que cumplan las expectativas del proyecto en menos tiempo.

# Funcionalidad

La interacción con el juego se lleva a cabo mediante movimientos en las direcciones cardinales con las teclas elegidas, una única tecla para disparar el arma secundaria y las teclas de movimiento mas las teclas “enter” y “escape” para navegar entre menús.