

8/6/2015

4 DAM

BATTLE OF TERRORIST



VIDEOJUEGO BASADO EN JAVASCRIPT 2014/2015
DAVID SERRANO Y DANIEL TORRES

1. Índice.

1. ÍNDICE.	2
2. DESCRIPCIÓN DEL PROYECTO.	3
3. FUNCIONAMIENTO.	4
3.1 CONTROLES.	4
3.2 MAPAS.	5
3.3 MODO DE JUEGO.	6
3.4 PERSONAJES.	6
4. FICHA TÉCNICA.	7
4.1 SERVIDOR.	7
4.2 DOMINIO.	7
4.3 HARDWARE DEL USUARIO.	8
5. ANTECEDENTES Y DOCUMENTACIÓN.	8
5.1 CALL OF DUTY.	8
5.2 COUNTER STRIKE.	9
5.3 BATTLEFIELD.	9
5.4 MÉTODO DE FINANCIACIÓN.	9
5.5 CANVAS.	10
5.6 HTML5.	10
5.7 CSS3.	11
5.8 JAVASCRIPT.	11
5.9 PROCESSING.	12
5.10 HOVER.	12
6. ESTIMACIONES.	13
6.1 COCOMO.	13
6.1.1 Atributos de software.	13
6.1.2 Atributos de hardware.	13
6.1.3 Atributos de personal.	13
6.1.4 Atributos del proyecto.	14
6.1.5 Cálculo final del proyecto.	14
6.2 METODOLOGÍA.	14

6.2.1 Número de sprint.....	14
7. DISEÑO.....	17
7.1 INDEX.HTML	17
7.2 WORLD.....	18
7.3 PLAYER.....	18
7.4 CHARACTER.....	18
7.5 COLLISION.....	19
7.6 BULLET.....	19
7.7 SAVEIMAGES.	19
7.8 SPRITES.	19
7.9 TILE.....	20
7.10 PROCESSINGFUNCTION.	20
8. DIAGRAMAS.	21
8.1 DIAGRAMA DE CLASES.	21
8.2 ACTIVIDADES.	22
8.3 CASOS DE USO.	23
8.4 ESTADO.....	24
8.5 SECUENCIA.....	25
9. CONCLUSIONES.	26
10. BIBLIOGRAFÍA.....	27

2. Descripción del proyecto.

Dos jóvenes programadores, Daniel Torres y David Serrano, decidieron crear Battle of Terrorist, un shooter desarrollado en 2D, para que cualquier usuario con un ordenador pueda jugar con sus amigos, familiares o conocidos.

Ha sido desarrollado en JavaScript, pintado en un lienzo conocido como Canvas sobre una página Web con HTML5 y CSS3 y un menú creado con Processing basado en Java.

El juego consiste en elegir un personaje de los cuatro diferentes que hay, con su correspondiente color. Utilizando para cada usuario su control de botones para caminar por los dos mapas que se dispone. Sin olvidarnos de tener que disparar para poder terminar con el enemigo.

Los mapas, han sido cuidadosamente diseñados, para ofrecer una mejor jugabilidad al usuario, por ellos se encuentran diferentes escondites, ya sea debajo de una mesa, una esquina secreta, etc. Así que hay que tener cuidado por donde se va, no vaya a ser que el enemigo te sorprenda.

Cuando un jugador sea el mejor de cinco rondas, ganará y se pasará al siguiente mapa.

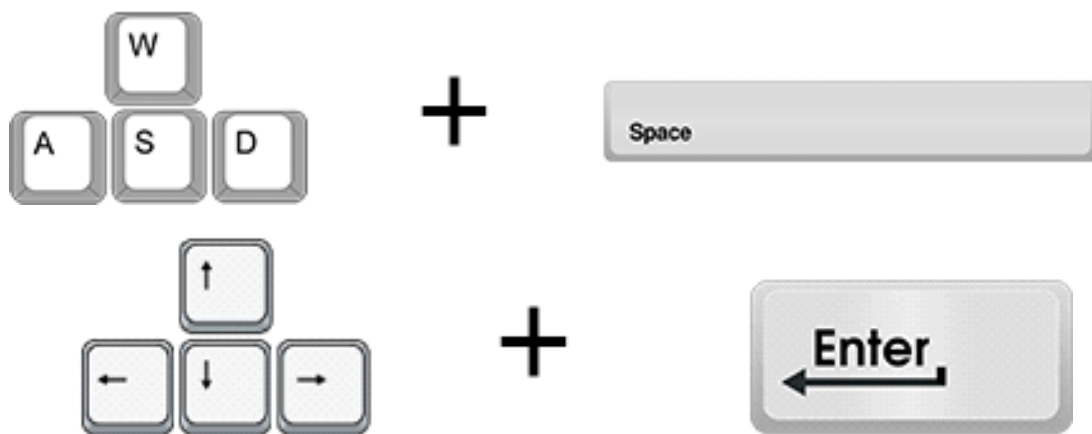
3. Funcionamiento.

Desde Battle of Terrorist, se ha intentado hacer un juego realista dentro de la realidad de 2D, entretenido e intuitivo.

3.1 Controles.

Los controles son sencillos de manejar, el jugador uno se maneja con el ya tradicional WASD

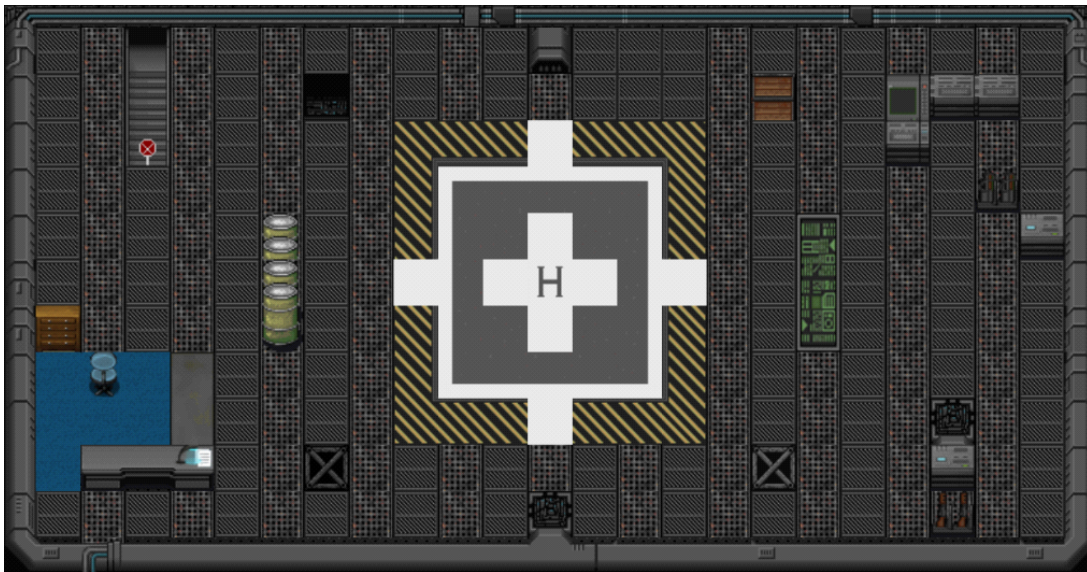
y con el espacio para poder disparar, mientras que el jugador dos, lo hará con las flechas y el intro para poder disparar.



3.2 Mapas.

Todo ello, en dos increíbles mapas, el primero en un edificio espía controlados por los comunistas, llamados Spetsnaz, donde los Navys Seals estadounidenses intentarán derrotar para salvar el mundo del ataque nuclear.

Además, se puede sorprender al enemigo en los escondites que hay por los diferentes mapas, habrá que ir con cuidado para que no acaben contigo.



El otro es desarrollado en una nave, donde los soldados comunistas se reúnen, investigan ataques químicos y viven para tramar el ataque para conquistar el mundo. Solo habrá una oportunidad para salvar el planeta.



3.3 Modo de juego.

El juego tiene el modo *Destruir*, que consiste en terminar con el enemigo, el que consiga matar más en 5 rondas es el que gana. Todo ello, habrá que conseguirlo antes de que finalice el tiempo, en cada ronda se dispondrá de 1 minuto 30 segundos para poder ganar.

3.4 Personajes.

Existen cuatro personajes a elegir:



- **Jack McGarret:** Después de que volviera de la guerra de Vietnam con los FiveZero, el antiguo compañero de Ellie en el escuadrón, se trasladó lejos de la ciudad y abrió una escuela para soldados novatos. La llamada que recibe de Ellie le trae de vuelta al combate.
- **Ellie Kurkov:** Una ex capitán que había ayudado a derrotar al enemigo dos veces anteriormente, ahora Ellie trabajaba como detective privada. Cuando es informada de que el próximo objetivo será el país de su viejo amigo y compañero, Jack , es entonces cuando inmediatamente organiza un grupo de combate para salvar el país o incluso algo peor.
- **Paul Hunter:** un soldado recién salido de la academia, donde fue el primero en todas las pruebas, batiendo todos los récords registrados. Jack fue su profesor.

- **Billy Fielding:** sufrió bastante en su infancia, ya que fue abandonado por sus padres a los 8 años. Una familia humilde a las afueras de la ciudad le adoptó, pero un día, apareció el cadáver de una chica cerca de la casa, y la policía le culpó de ello y tuvo que huir hasta que se topó con Ellie, quien le contrató para trabajar como detective.

Todos ellos, están disponibles para elegir en tres colores diferentes: rojo, azul y verde.

4. Ficha técnica.

4.1 Servidor.

Se ha necesitado de un servidor para poder instaurar el proyecto en la red. A través de www.000webhost.com, donde nos facilitan un servidor con las siguientes características:

- **Ancho de banda:** 100GB.
- **Disco duro:** 1.5GB.
- **Versión Apache:** 2.2.19 (Unix)
- **Versión PHP:** 5.2.*
- **Versión MySQL:** 5.1

4.2 Dominio.

Se han necesitado un dominio y un subdominio para poder utilizar el proyecto en la red.

En 000webhost, te dan la oportunidad de crear servidor y dominio, bien, pues el primer paso después de crear el servidor fue crear el dominio battleofterrorist.comli.com pero como queda poco estético a la hora de ver e introducir eso en la URL, se decidió crear un dominio de una sola extensión y si pudiera ser gratuito, en este caso, nos inclinamos por un dominio .tk.

La extensión .tk es un dominio de nivel superior geográfico para el archipiélago neozelandés de Tokelau. La idea era dar a conocer el archipiélago mediante el uso de sus dominios.

Al crear el dominio battleofterrorist.tk, hay que indicar a donde hay que redireccionar esa URL, y en esa ponemos nuestros subdominio battleofterrorist.comli.com . Es entonces cuando ya está nuestra web operativa.

4.3 Hardware del usuario.

Para poder jugar a Battle of Terrorist, no se precisa de una tecnología potente y de última generación, tan solo se precisa de:

- Un ordenador con acceso a Internet, su navegador web favorito (Mozilla Firefox, Google Chrome, Internet Explorer, etc.), un teclado y un ratón para elegir a los personajes que más le gusten.

5. Antecedentes y documentación.

5.1 CALL OF DUTY.

Call of Duty es una serie de videojuegos en primera persona, denominado shooter, creada por Ben Chichoski, desarrollado por Infinity Ward y distribuida por Activision.

La franquicia comenzó para ordenadores y después se expandió hacia videoconsolas. La serie inicialmente se ambientó en la Segunda Guerra Mundial, relatando personajes y combates acaecidos. Esto fue cambiando hasta la actualidad, donde los argumentos son en ambientes futuristas y/o ficticios.

La saga ha gozado de un gran éxito comercial y crítico, el cual ha ido aumentando desde su creación hasta la actualidad. Ha conseguido vender 55 millones de copias totales de sus juegos, recaudando más de 3 mil millones de dólares.

5.2 COUNTER STRIKE.

Counter-Strike es otra serie de videojuegos en primera persona, creada por Minh Le y Jess

Cliffe en 1999. Es una modificación del juego Half-Life.

La franquicia es solo accesible por ordenador para Windows, Apple y Linux desde su plataforma Steam.

5.3 BATTLEFIELD.

Battlefield es también otra serie de videojuegos del género. Desarrollado por la compañía David S.A. & ASIR y distribuido por Electronic Arts. Los juegos se centran en el combate por equipos en grandes mapas.

La franquicia es distribuida tanto para ordenador, como para videoconsolas. Centrándose en un principio en ambientes históricos hasta que en la actualidad, son ambientes más presentes e incluso futuristas.

Desde Battle of Terrorist, intentamos que la experiencia para el usuario sea la más divertida posible, cogiendo de los tres títulos, lo mejor de cada uno.

Por parte del Counter Strike, nos llamó la atención sobre como empezaron, su juego en 2D y sus sonidos realistas. De Call of Duty, sus formas de juego y su juego adictivo. Y, desde Battlefield, el hacer grandes mapas con gran realismo.

5.4 MÉTODO DE FINANCIACIÓN.

Por el momento, Battle of Terrorist, ha sido desarrollado sin ninguna financiación, todo lo creado ha sido con tecnología gratuita. En el día de mañana, si se necesita algún tipo de financiación, utilizaremos el método Kickstarter, se trata de una página web de financiación en masa para proyectos, de música, películas o videojuegos.

5.5 CANVAS.

Canvas ha sido el utilizado para representar el juego. Se trata de un elemento incorporado en HTML5 que permite la generación de gráficos dinámicamente. Permite generar gráficos estáticos y animaciones.

Fue implementado por Apple para su navegador Safari. Más tarde fue adoptado por otros navegadores, como Firefox a partir de su versión 1.5.

En nuestro caso, se ha utilizado para representar el juego. Todo ello va sobre el lienzo y representa lo que JavaScript le pide.

5.6 HTML5.

HTML5 es la quinta revisión de lenguaje básico de la World Wide Web, HTML. Se publicó en Octubre de 2014.

HTML (HyperText Markup Language) hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, vídeos, etc.

El desarrollo de este lenguaje de marcado es regulado por W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web.

HTML representa la página web donde va incluido el Canvas y el menú con el que poder interactuar a través de la página.

5.7 CSS3.

CSS (Cascading Style Sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML. El W3C es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores o usuarios.

La idea es separar la estructura de un documento de presentación. Es útil cuando solo quieres cambiar una cosa u otra. No todo en el mismo documento, así la estructura no depende de la presentación o viceversa.

La información de estilo puede ser definida en un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo `<style>`.

Es el encargado de todo el estilo de la página web, además se ha incluido una librería de GitHub, llamada Hover, con el cual el menú interacciona a la hora de poner el ratón por encima de las diferentes opciones del menú.

5.8 JavaScript.

Javascript es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se implementa el DOM (Document Object Model).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones solo en el marco de la aplicación cliente, sin acceso a funciones del servidor.

JavaScript es el núcleo donde se integra el proyecto, con él, ha sido posible hacer todo el juego. Se encuentra enlazado con Processing, a la hora de que el usuario elija un personaje con su color correspondiente y de lugar al inicio de la partida.

5.9 Processing.

Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Fue iniciado por Ben Fry y Casey Reas a partir de reflexiones en el Aesthetics and Computation Group del MIT Media Lab dirigido por John Maeda.

Con Processing ha sido posible hacer el menú, había otras alternativas, pero decidimos probar algo diferente.

5.10 Hover.

Se ha utilizado Hover, para el menú principal de la página web. Donde los botones cambian dinámicamente de tamaño y color a la hora de pasar el ratón por encima. Es una buena manera de conectar con el usuario.

6. Estimaciones.

6.1 COCOMO.

KLDC: 1,1

Lenguaje: JavaScript.

Proyecto de software: Orgánico.

6.1.1 Atributos de software.

- **Fiabilidad:** si hay algún fallo no genera ninguna pérdida en la empresa. 0,88.
- **Complejidad:** el proyecto no es demasiado complejo a la hora de calcular los diferentes aspectos que tiene el programa. 0,94.

6.1.2 Atributos de hardware.

- **Restricciones de tiempo de ejecución:** a la hora de cargar, no hay una restricción exigente. 1,00.
- **Tiempo de respuesta:** el juego debe ser rápido y que no se ralentice para que el usuario se lleve una mala visión del producto. 1,15.

6.1.3 Atributos de personal.

- **Capacidad de análisis:** capacidad baja, ya que nuestros programadores acaban de salir al mundo laboral. 1,19.
- **Experiencia en la aplicación:** es la primera vez que se introducen en este tipo de aplicación, por lo que irán un poco más lento que lo normal. 1,13.
- **Calidad de los programadores:** aunque no hayan tenido experiencia, saben programar perfectamente. 1,00.
- **Experiencia en el lenguaje:** no han utilizado este lenguaje en lo que orientación a objetos se refiere. 1,14.

6.1.4 Atributos del proyecto.

- **Técnicas actualizadas de programación:** se utilizarán técnicas de programación convencional. 1,00.
- **Utilización de herramientas de software:** se utilizarán herramientas estándar que no exigen una formación elevada. 1,00.
- **Restricciones de tiempo de desarrollo:** existe solo un límite, y es la fecha de entrega. 1,00.

6.1.5 Cálculo final del proyecto.

FAE: 1,45

Esfuerzo: $a \cdot KLDC^b \cdot FAE \rightarrow 3,2 \cdot 1,1^{1,05} \cdot 1,45 = 5,12$ personas / mes.

Tiempo: $c \cdot ESFUERZO^d \rightarrow 2,5 \cdot 5,12^{0,38} = 4,65$ meses.

Personal promedio: $\frac{ESFUERZO}{TIEMPO} \rightarrow \frac{5,12}{4,65} = 1,10$ personas.

Coste del proyecto: al ser una práctica el sueldo es nulo, por lo tanto el coste será de 0€.

Según los resultados, necesitaríamos un equipo de 2 personas trabajando alrededor de 5 meses.

6.2 Metodología.

6.2.1 Número de sprint.

Para poder llevar un orden sobre la estructura del proyecto, éste, tiene tres sprints. Todo estará representado en horas.

6.2.1.1 Sprint 1.

Este primer sprint se decidió realizar los siguientes puntos:

- **Representar mapa de prueba:** un pequeño lugar donde poder situar al personaje y poder interactuar. 5.
- **Imagen del personaje:** se decide que personaje se va a utilizar para realizar el juego. 5.
- **Movimiento personaje:** que el personaje se mueva por el mapa de prueba. 20.
- **Colisión con mapa:** el personaje no pueda salir del mapa y/o ir contra objetos que no debe traspasar. 8.

6.2.1.2 Sprint 2.

Este segundo sprint consta de lo siguiente:

- **Disparo personaje:** el personaje pueda disparar balas. 10.
- **Crear segundo personaje:** éste tendrá las mismas características que el primero. 0.5.
- **Colisión personaje:** cuando se choquen los dos personajes, que no se traspasen entre ellos. 1.
- **Posición personaje:** inicio de una posición aleatoria del personaje. 0.5.
- **Colisión personaje con bala:** la bala choque con el personaje y le debilite. 15.
- **Menú:** donde el usuario elige al personaje con el color que escoja. 30.

6.2.1.3 Sprint 3.

- **Mapas:** conseguir imágenes para los mapas y representarlos en el juego. 10.
- **Contador de tiempo:** contar el tiempo en el que finaliza una ronda. 1.

- **Escondites:** sitios donde el personaje pueda esconderse y pasar desapercibido para el enemigo. 2.
- **Mostrar marcador:** para mostrar el marcador que llevan los personajes. 1.
- **Mostrar barra vida:** cuanta vida tiene cada personaje. 2.
- **Menú página web:** acciones para el menú de la página web. 2.

6.2.1.4 Backlog.

- **Tipo de juego (detonar bomba):** realizar el modo de juego de detonar una bomba. 10.
- **Tipo de juego (gol):** cuando un personaje llegue a la "casa" del enemigo, anota 1 gol. Gana el que más goles meta en 2 minutos. 20.
- **Recompensas por el mapa:** aparecer recompensas(vida extra, diferente arma, más defensa, etc.) de forma aleatoria en una posición aleatoria. 10.
- **Online:** hacer el juego online para que el usuario pueda jugar en red. 30.
- **6 Jugadores:** una vez sea online, que puedan jugar hasta 6 jugadores en cada equipo. 10.
- **Cámara:** que la cámara se centre en el jugador. 5.
- **Vehículos:** poder conseguir un transporte como una moto o coche. 20.
- **Implementar seguridad:** para que el juego no sea hackeado. 20.
- **Modificación de disparo:** después de ser online, que el disparo pase a ser función del ratón para que haya una forma más fácil de mover. 20.
- **Diseño mapa:** que el usuario pueda crear su propio mapa. 10.
- **Mapas:** creación de mayor cantidad de mapas y más extensos. 8.

7. Diseño.

El diseño del juego ha sido estudiado para poder incrementar nuevos objetivos fácilmente, aunque siempre se puede mejorar el código y trabajamos duro para que eso sea así. El juego ha sido creado de la siguiente manera:

7.1 Index.html

El juego comenzará su andadura a partir de éste punto.

```
function gameInit()
{
    Images = new SaveImages();
    Sprites = new SaveSprites();

    for(var i=0; i<= numImages; i++)
        Images.load([ i, "images/game/"+i+".png"]]);
}
```

Donde se ejecutará la función *gameInit()* donde se cargarán todas las imágenes de los mapas, personajes, iconos y balas. Se van incluyendo con un array, en vez de ir escribiendo uno por uno todas las imágenes. Sería un trabajo innecesario.

```
</new>
<body onload="cargarCanvas();" > <!-- To load canvas -->
<div id = "main" > <!-- All the page -->
    <a href="index.html"><img src = "images/web/logo1.png" alt = "Logo" id = "Logo"></a>
    <ul id="menu">
        <li><a href = "index.html" class = "hvr-shutter-out-vertical"> HOME</a></li>
        <li><a href = "#modal" class = "hvr-shutter-out-vertical"> HOW TO PLAY </a></li>
        <li><a href = "#canvas1" class = "hvr-shutter-out-vertical"> PLAY </a></li>
    </ul>
    <div id = "footer"> </div>
    <div id = "gameplay"> <!-- The game goes here -->
        <div id="modal" class="modalmask">
            <div class="modalbox movedown">
                <a href="#close" title="Close" class="close">X</a>
                <h2>CONTROLLERS</h2>
                <h3 align="left"> PLAYER 1 </h3>
                <img src = "images/web/player1.png">
                <h3 align="left"> PLAYER 2 </h3>
                <img src = "images/web/player2.png">
                <h2> + </h2>
            </div>
        </div>
        <canvas id = "canvas1"> </canvas>
    </div> <!-- ***End gameplay*** -->
```

Cuando el navegador, detecte que la página web ha sido cargada totalmente, ejecutará automáticamente la función *cargarCanvas()*. La cual se encargará de mostrar el menú del juego, donde elegiremos los personajes que, tras ser elegidos, lucharán en el juego.

7.2 World.

Para que todas las entidades en un mismo espacio puedan interactuar entre sí, se crea la clase World. Que se encargará de calcular todos los datos donde se representará el juego. Donde será la encargada de administrar el tiempo para que se actualicen el resto de entidades.

Esta clase será además la encargada, de realizar el movimiento de los personajes, así como de sus balas, comprobar si existen colisiones entre balas y jugadores y entre jugadores. Después de realizar las comprobaciones, se dispone a poder pintar y actualizar hasta que o uno de los jugadores muera, o que termine el tiempo de la ronda.

7.3 Player.

Desciende de la clase World. Esta clase es la encargada de determinar el movimiento de los futuros personajes. Recoge la dirección que indica el usuario a través del teclado. Al recoger la dirección, se va incrementando o disminuyendo la posición X e Y de cada personaje.

7.4 Character.

Desciende de la clase Player. Se determina la posición de los personajes en el mapa, la velocidad a la que caminan a través del mapa, la imagen (sprite) que usará en cada momento y que empieza siempre por el primero cuando cambie de dirección para darle más sensación de realismo, la vida que tendrá cada personaje, y el tiempo que transcurre entre sprite y sprite.

Se comprobará si el personaje choca con algo o con el enemigo llamando a la clase *Collision*.

Después de realizar todas las comprobaciones, lo pinta en el tablero.

7.5 Collision.

Desciende de dos clases, de *Bullet* y de *Character*. En ésta clase se tiene una coordenada x y una coordenada y, y hace las comprobaciones de si tanto las balas o los personajes se topan con objetos que no se pueden traspasar, comprobando en el mundo si en esa coordenada se puede caminar o de lo contrario se choca.

7.6 Bullet.

Desciende de *World*. En ésta clase la bala tendrá una velocidad, un tamaño, unas coordenadas x e y, un dueño para comprobar quien es el que dispara y a quien, y un daño aleatorio donde existe más probabilidad que salga un menor número que un mayor número.

Se detecta si choca con un jugador, y si es así, a éste le daña hasta terminar con su vida.

Después de todas las comprobaciones, se pinta en el tablero.

7.7 SaveImages.

Se llama cuando se ejecuta el *index.html*, donde se le pasarán todas las imágenes que se utilizarán en el juego. Con los métodos se cargarán las imágenes, con otro se comprueba si se han cargado todas y con otro se puede conseguir la imagen que se solicite.

7.8 Sprites.

Llamado también en *index.html*, donde después de que el usuario elija que usuario y color desea, se recoge a través del *SaveImages* la imagen que solicita y con ello se añade a *Sprites* junto con la dirección.

Con el método de *createDirection* se indican de la imagen el número de filas que tiene, el número de columnas, el espacio que hay que dejar entre los márgenes de las imágenes y las direcciones que existen.

Dependiendo de la dirección tendrá que recoger el número de imágenes que están asignadas a dicha dirección.

Además del *get* y *add* para conseguir y añadir imágenes de los sprites.

7.9 Tile.

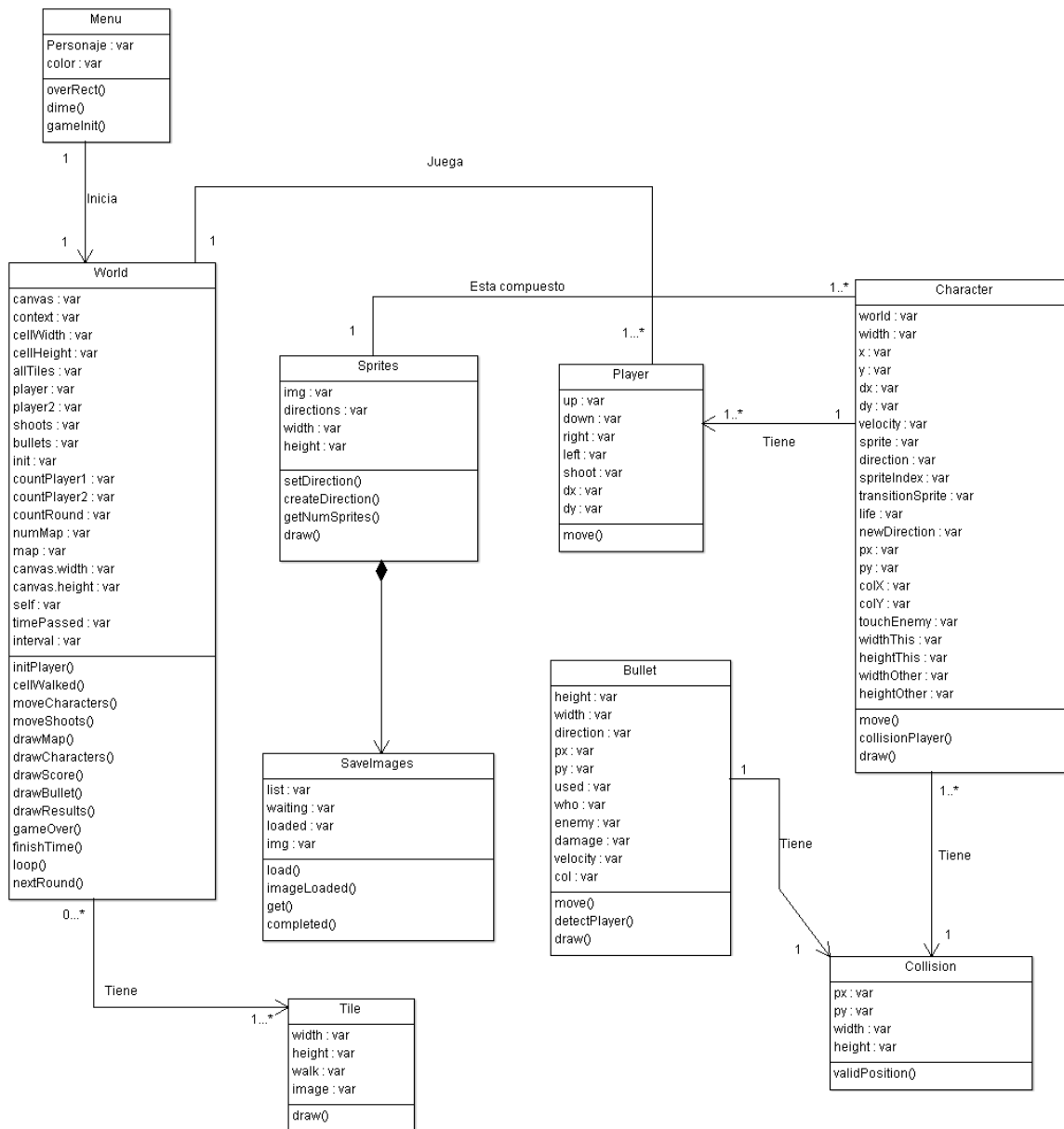
Desciende de la clase *World*. Con esta clase se consigue tener el realismo del mapa, y poder colocar las imágenes en nuestro tablero. Cada Tile corresponde a un número y se pinta en el tablero.

7.10 ProcessingFunction.

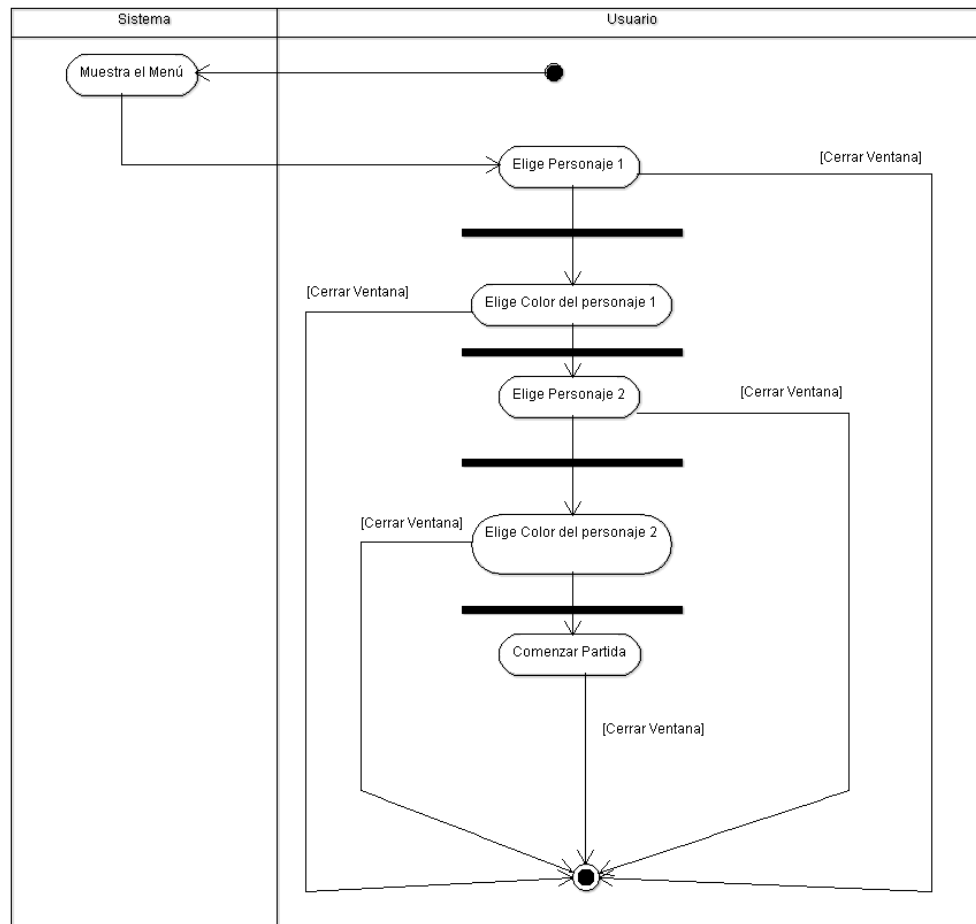
Es la clase encargada de mostrar el menú y de que el usuario elija los personajes que desee. Puede elegir de entre cuatro jugadores y tres colores diferentes cada uno. Para dar un toque de personalización al usuario.

8. Diagramas.

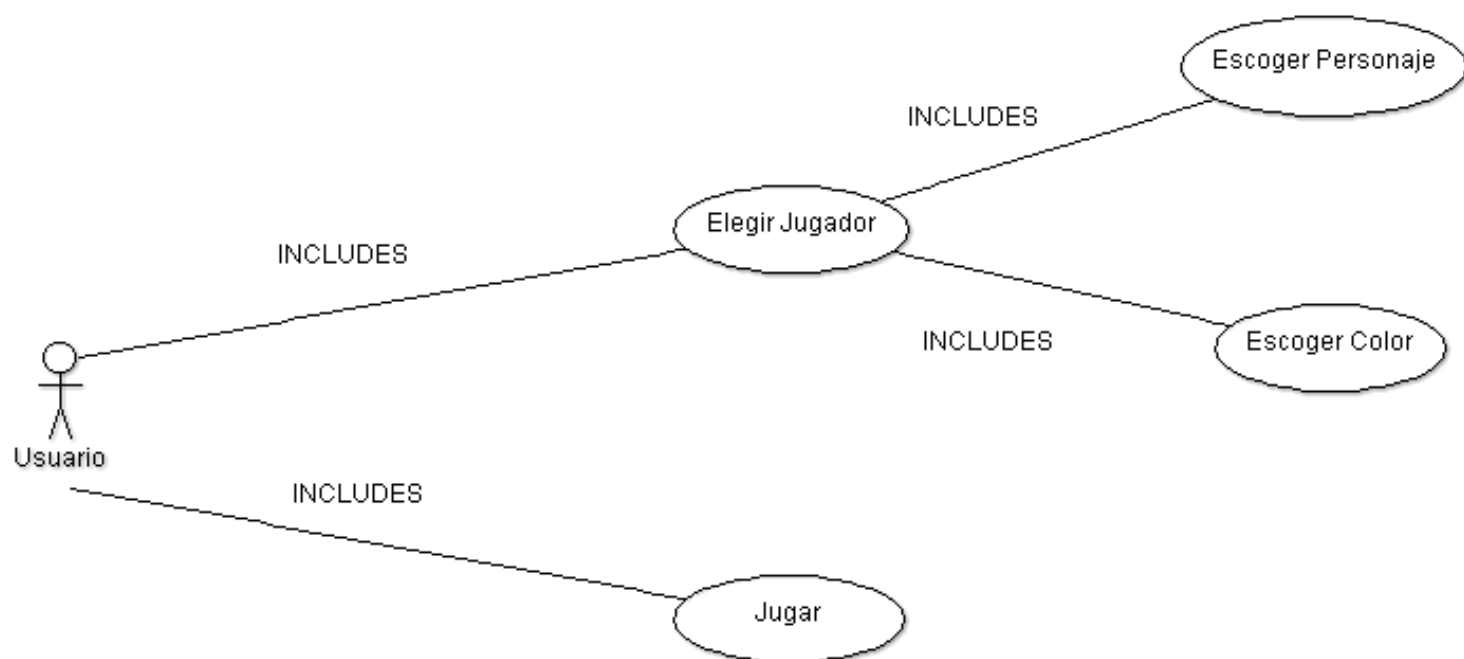
8.1 Diagrama de clases.



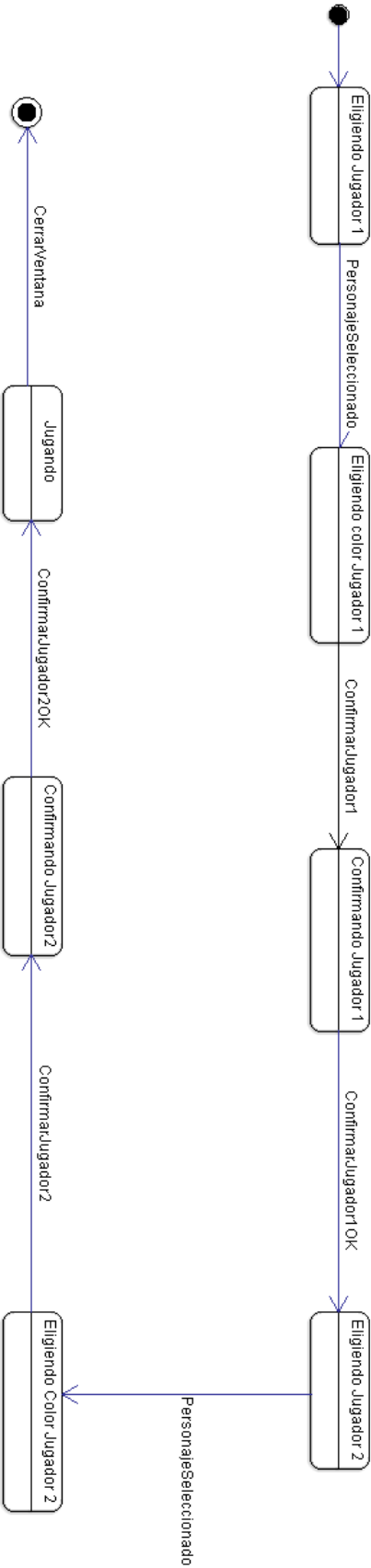
8.2 Actividades.



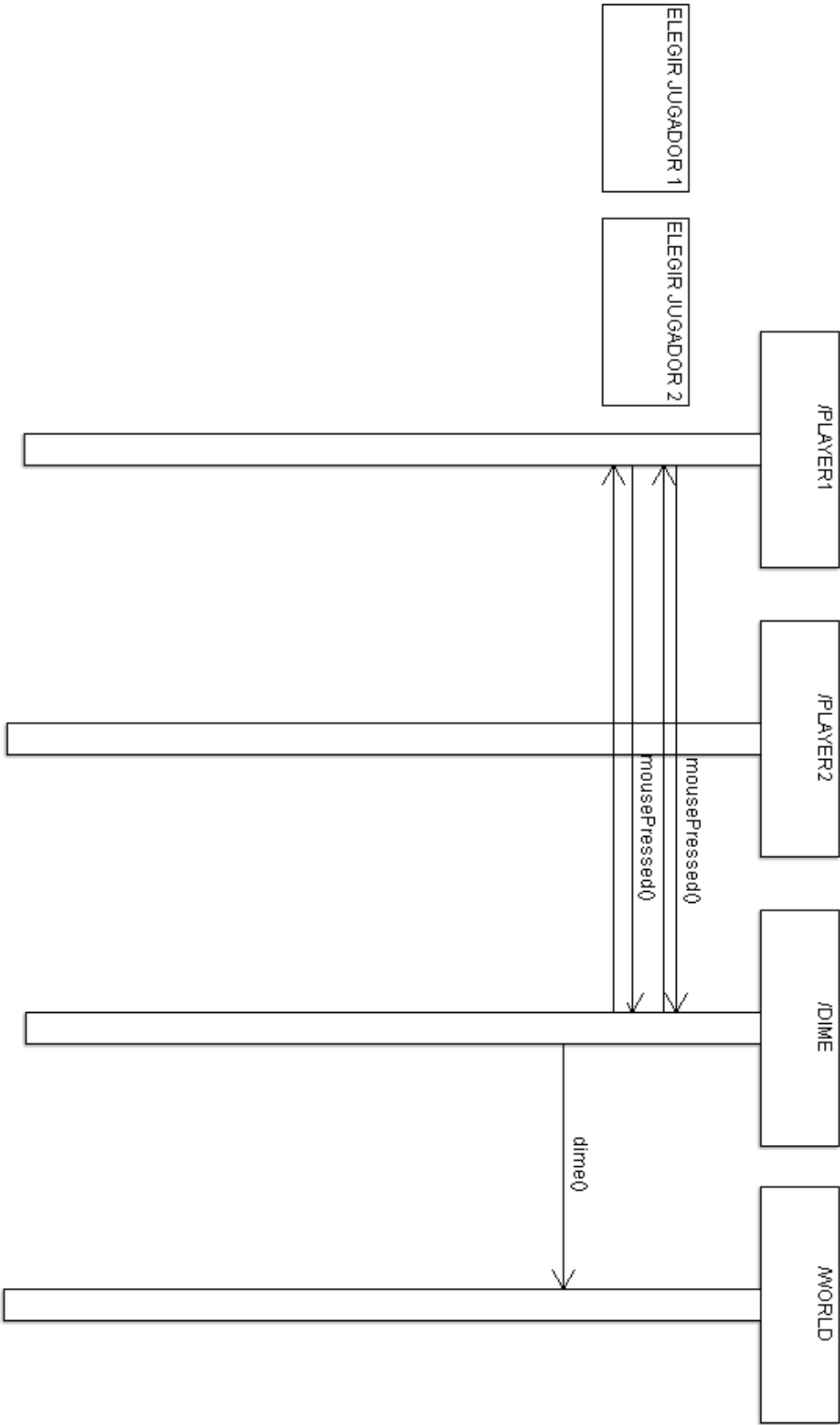
8.3 Casos de uso.



8.4 Estado.



8.5 Secuencia.



9. Conclusiones.

La creación de Battle Of Terrorist comenzó con la idea de programar un juego que sirviera de método de aprendizaje, para ambos compañeros del proyecto, de JavaScript, un lenguaje que hemos estudiado a nivel básico orientado a objetos y que tiene en la actualidad una gran importancia.

En un principio, el videojuego iba a contar con varios escenarios, y con un modo multijugador que permitiera conectar hasta a 4 jugadores distintos, a luchar por la victoria en los diferentes escenarios. Además, teníamos pensado introducir diferentes cámaras para cada jugador, de manera que cada uno viera una parte del mapa en un radio alrededor de su personaje, y diferentes modos de juego que sirvieran para hacer más entretenido el desarrollo del juego y su posterior resultado final. Esto no ha sido posible por problemas de tiempo, y planificación, algo que a ambos componentes del proyecto nos deja un mal sabor de boca.

Después de terminar este proyecto, como ya hemos dicho, tenemos la sensación de haber podido hacer mucho más. De no haber demostrado todo lo que hemos aprendido en estos dos años en el modulo, y aunque hemos aprendido a manejar *Javascript* y *Processing.js*, nos quedamos con las ganas de haber introducido, y por tanto aprendido, el modo multijugador que teníamos pensado, y aunque en menor importancia, las diferentes cámaras para los jugadores.

Como ya hemos dicho, creemos que de no haber tenido problemas con la planificación de cada parte del proyecto, podríamos haber introducido las dos características anteriormente nombradas. Este problema no puede servir como una excusa, sino como un punto de inflexión que nos ha servido para hacer autocrítica y que nos servirá en el futuro a la hora de planificar cada tarea.

Asimismo, y con la espina clavada de mejorar un juego que comenzamos con mucha ilusión y grandes ideas, creemos que en este verano vamos a ir mejorando detalles del juego que nos ayuden tanto a aprender como a cumplir con las expectativas, aun hoy insatisfechas, del proyecto que comenzamos hace unos meses.

Para terminar, el proyecto también nos ha servido para aprender a trabajar en equipo. A lo largo del mismo, han surgido muchas dudas resultas por el compañero, muchas felicitaciones por cada parte que se iba consiguiendo, y también, alguna que otra crítica que ha servido para mejorar en el desarrollo del videojuego, nunca sobrepasando al terreno personal.

10. Bibliografía.

Para programar *Battle Of Terrorist* hemos utilizado como recurso de información y de dudas varias fuentes.

Para la confección del estilo del menú de la pagina web hemos utilizado este enlace:

<http://ianlunn.github.io/Hover/>

Para obtener las imágenes de las que está compuesto el mapa hemos visitado diversas paginas:

<https://www.pinterest.com/dolphinsheart/rpg-maker/>

<https://www.pinterest.com/pin/308215168219125505/>

<https://www.pinterest.com/pin/232498399487570113/>

<http://malinovskiydima519.blogspot.com.es/2014/02/rpg-maker-vx-tilesets-download.html>

<http://www.taringa.net/comunidades/rpgmakervx/9128969/Aporte-Futuristic-Tiles.html>

<https://www.pinterest.com/pin/396457573421585382/>

<http://www.universomaker.net/t1678-vx-ace-tilesets-celianna>

<http://www.alteredgamer.com/rpg-maker/105824-some-new-tilesets-vx/>

<http://www.taringa.net/comunidades/rpgmakervx/8857180/Aporte-Tilesets-de-Subterranos-y-Cloacas.html>

<http://projectxrpg.googlecode.com/svn/trunk/%20projectxrpg/ProjectX/Graphics/Characters/>

<https://vxresource.wordpress.com/category/resources/tilesets/>

<http://www.rpgmakervxace.net/topic/35-grannys-lists-ebtsukuru-blog-resources-for-vx/>

<http://www.rpgmakervxace.net/user/35640-quizicalgin/?tab=reputation>

<http://www.rpgmakervxace.net/topic/20236-quizzes-work-drawer-resources-530-new-sprite-sheet-guides/>

<http://painhurt.deviantart.com/art/RPG-Maker-Wooden-Floor-316114650>

<http://nicnubill.deviantart.com/art/RPG-Maker-VX-Ace-VX-like-XP-Tile-472063464>

<http://programavideojuegos.blogspot.com.es/2013/11/conceptos-basicos-de-rpg-maker-vx-ace.html>

Para el personaje y su movimiento hemos utilizado las imagenes del siguiente link:

<http://www.rpg-maker.fr/ressources/characters/futuriste/>

Para la elaboración del menú con processing hemos utilizado los siguientes links:

<https://processing.org/>

<http://processingjs.org/>

<https://github.com/processing>

<http://www.false.jp/motion/>

<https://scottdowne.wordpress.com/2010/07/16/processing-js-canvas-custom-fonts-and-font-face/>

<http://www.desarrolloweb.com/manuales/manual-processing.html>

Para descargar la fuente que hemos utilizado en el texto del menu, asi como en los contadores de tiempo y rondas ganadas, hemos utilizado el siguiente link:

<http://www.dafont.com/es/transformers-movie.font>

Para la elaboración del juego, como apoyo hemos usado los siguientes links:

<http://juegos.canvas.ninja/>

<http://www.jlabstudio.com/webgl/>