

Exonite

Víctor Manuel Quesada Cobos
Grau en Enginyeria Informàtica

Joel Servitja Feu

10 de junio de 2018

© Víctor Manuel Quesada Cobos

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Exonite</i>
Nom de l'autor:	<i>Víctor Manuel Quesada Cobos</i>
Nom del consultor:	<i>Joel Servitja Feu</i>
Data de lliurament (mm/aaaa):	<i>06/2018</i>
Àrea del Treball Final:	<i>Videojocs Educatius</i>
Titulació:	Grau en Enginyeria Informàtica
Resum del Treball (màxim 250 paraules):	
<p>El presente proyecto de final de grado consiste en el desarrollo de un videojuego de navegador multijugador masivo online con componentes de rol y un sistema de combate basado en turnos.</p> <p>El juego pone al jugador en el rol de que ha sido el elegido por su civilización para llevar a cabo la búsqueda de un material llamado Exonita. Este material es muy valioso y es totalmente necesario para el sustento de la vida en el universo.</p> <p>Bajo este rol, el jugador dispone de personajes con los que entablar combates por turnos contra otros jugadores en un escenario 2D con perspectiva isométrica. Como resultado de estos enfrentamientos los jugadores podrán ganar Exonita y de esta forma ser mejor puntuados en el ranking.</p> <p>El resultado de este trabajo es una aplicación web con diseño adaptativo creada con tecnología Microsoft ASP.NET MVC. Esta aplicación web, incluye una serie de secciones en la que los jugadores pueden gestionar las diferentes funcionalidades del juego, así como, el escenario de combate.</p> <p>Este escenario y el juego, han sido creados usando el renderizador 2D Pixi js y un motor gráfico isométrico llamado Triviso.</p> <p>Esta memoria incluye las fase de planificación, diseño y desarrollo de todo el proyecto, así como la historia y descripción del videojuego.</p>	

Abstract (in English, 250 words or less):

This project is about the development of a massively multiplayer online browser game with role components and a turn-based combat system.

The player of this game, has the role of being the chosen by his civilization to carry out the search of a material called Exonite. This material is very valuable and it is absolutely necessary and vital for the life in the universe.

Under this role, the player has some characters to fight in turn-based confrontation against other players in an isometric 2D scenario. As a result of these confrontations the players will be able to win Exonite and as a consequence improve their position in the ranking.

The result of this project, is a web application with responsive design created with Microsoft ASP.NET MVC technology. This web application includes different kind of sections where the players can manage the different functionalities of the game and the combat scenario as well.

The scenario and the gameplay, have been created using the 2D renderer Pixi js and an isometric graphic engine called Triviso.

This essay includes the planning, design and development phases of the entire project, as well as the history and description of the video game.

Paraules clau (entre 4 i 8):

Videojuego, multijugador, online, rol, turnos, web, javascript, asp.net

Índice

1. Introducción	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método a seguir	1
1.4 Planificación del Trabajo.....	2
1.5 Breve resumen de productos obtenidos.....	4
1.6 Breve descripción de los otros capítulos de la memoria.....	4
2. Idea del juego	5
2.1 Descripción.....	5
2.2 Género y referencias	5
2.3 Interacción con el jugador.....	6
2.4 Plataforma destino.....	6
3. Conceptualización.....	7
3.1 Historia y ambientación.....	7
3.2 Definición de los personajes	7
3.3.1 Nave espacial	8
3.3.2 Estación Espacial	9
3.4 Interacción entre los actores del juego.....	9
4. Sistema de combate	10
4.1 Acciones.....	10
4.2 Pantalla de combate	12
4.3 Pantalla de resultados	13
4.4 Reproducción turno anterior	14
5. Modos de juego	14
5.1 Jugador contra Jugador	14
5.2 Jugador contra bot.....	15
6. Secciones del juego.....	16
6.1 Página de aterrizaje/acceso.....	16
6.2 Raids	16
6.3 Tripulación	18
6.4 Bitácora	19
6.5 Módulos de la nave.....	20
6.6 Mercado de Exonita.....	20
6.7 Tienda de Raiders	21
6.8 Muele de Exonita	22
6.9 Ranking	22
6.10 Tutorial	23
7. Material multimedia.....	24
7.1 Diseño de personajes	24
7.2 Mapas; objetos y tiles	26
7.3 Imágenes.....	27
7.4 Música	27
7.5 Efectos habilidades.....	28
8. Diseño y desarrollo	28
8.1 Programación web.....	30
8.1.1 Front-end.....	30
8.1.2 Back-end	31
8.2 Persistencia y acceso a datos.....	32
8.3 Utilización de material gráfico	32
8.4 Convirtiendo una aplicación web en una aplicación web en tiempo real	34
9. Conclusiones	35
10. Glossario	36
11. Bibliografía	38

Lista de figuras

Ilustración 1: diagrama de Gantt parte 1	3
Ilustración 2: diagrama de Gantt parte 2	3
Ilustración 3: HUD de acciones	10
Ilustración 4: selección movimiento	10
Ilustración 5: selección ataque	11
Ilustración 6: selección habilidad	11
Ilustración 7: HUD; acciones usadas	11
Ilustración 8: pantalla de combate	12
Ilustración 9: indicador daño	13
Ilustración 10: pantalla de resultados	13
Ilustración 11: modos de juego	14
Ilustración 12: tipos de juego modo JcJ	15
Ilustración 13: página de aterrizaje/acceso	16
Ilustración 14: Tus Raids	17
Ilustración 15: Raids interceptables	17
Ilustración 16: Raids detectadas	18
Ilustración 17: detalles del personaje	18
Ilustración 18: formulario subir de nivel	19
Ilustración 19: Bitácora	19
Ilustración 20: Módulos	20
Ilustración 21: Mercado de Exonita	21
Ilustración 22: mensaje Mercado de Exonita	21
Ilustración 23: Tienda de Raiders	21
Ilustración 24: Muelle de Exonita	22
Ilustración 25: Rankings	23
Ilustración 26: variaciones en el ranking	23
Ilustración 27: botón acceso tutorial	23
Ilustración 28: tutorial	24
Ilustración 29: esbozo personajes	25
Ilustración 30: animación personaje	25
Ilustración 31: objetos mapa, pruebas realizadas	26
Ilustración 32: proceso creación objetos mapa	27
Ilustración 33: aspecto final objetos mapa	27
Ilustración 34: animación aura ataque	28
Ilustración 35: arquitectura del proyecto	29
Ilustración 36: gestión de eventos en el escenario	30
Ilustración 37: dominio de datos	31
Ilustración 38: controladores	32
Ilustración 39: array tiles Traverso	33
Ilustración 40: aplicación MapImporter	33
Ilustración 41: Tiled imagen de ejemplo	34
Ilustración 42: notificación	35

1. Introducción

1.1 Contexto y justificación del Trabajo

El mundo de los videojuegos actualmente se encuentra en su momento de mayor relevancia y con una industria que año a año ve como sus beneficios económicos van al alza en líneas generales. Bajo esta premisa, los videojuegos son un mercado altamente mercantilizado y en el que queda poco espacio para los proyectos personales y nostálgicos.

Pese a este hecho, la justificación principal para el desarrollo de este trabajo nace de una motivación personal por parte del autor. El trabajo ha sido desarrollado con el fin de producir un videojuego que acerque y haga recordar la época en la que los videojuegos de rol multijugador masivo online en navegador pegaban a miles de niños y adolescentes a las pantallas CRT de sus ordenadores.

Todo el desarrollo y decisiones estratégicas sobre que producto se quiere conseguir, han sido claramente influenciados por la idea de producir un videojuego que sea del agrado para las personas que jugaron a este tipo de videojuegos.

1.2 Objetivos del Trabajo

- Planificar, estimar y llevar a cabo todas las fases de desarrollo de un videojuego.
- Crear un gameplay simple pero adictivo.
- Crear una aplicación web para acompañar el gameplay que aporte objetivos y recompensas más allá de éste.
- Publicar el juego y la aplicación web en internet de forma gratuita.
- Poner en práctica las diferentes aptitudes adquiridas por el autor durante su vida tanto académica como profesional.
- Satisfacción personal del autor.

1.3 Enfoque y método a seguir

La idea principal para llevar a cabo el trabajo y que ha decidido las estrategias a seguir es el objetivo que se quiere lograr. Un juego actual, pero que conserve la esencia de los primeros juegos de este tipo.

El producto a desarrollar es un producto nuevo, que está fuertemente inspirado en su género, pero que no puede ser considerado una adaptación de ningún producto existente.

Partiendo de que la plataforma de destino del proyecto es el navegador web, han sido evaluados varios engines y kits de desarrollo con el fin de encontrar el idóneo para las características deseadas.

Por una parte, han sido valorados engines y kits multiplataforma como Unity[1] y RPGMaker[2], pero no han sido los elegidos principalmente por no estar orientados a crear videojuegos web nativos y el uso de los cuales en gran medida resultaría aceptable pero desmedido por lo que no sería el idóneo.

Por otra parte, también han sido valoradas distintas alternativas para producir juegos nativos. En este caso, engines javascript. Han sido valoradas tres alternativas. Isogenic[3], un engine pensado para crear juegos html5, ha sido descartado por no ofrecer una comunidad amplia de desarrolladores. Phaser[4], un framework javascript, ha sido descartado por que su uso para el objetivo del proyecto resultaría desmedido. Por lo que la tercera alternativa, la suma del renderizador 2D Pixi js [5] i el engine javascript Triviso[6], han sido la elegida.

Esta combinación resulta idónea para el proyecto que se quiere realizar ya que facilita la creación y gestión del gameplay sin más opciones que las puramente necesarias.

1.4 Planificación del Trabajo

Hace falta diferenciar entre dos partes dentro del proyecto. Por un lado está el gameplay, que podría ser considerado el núcleo del proyecto, y por otro lado la aplicación web.

Las dos partes han sido desarrolladas de forma paralela, con el objetivo de tener el videojuego funcional lo antes posible e ir añadiendo capas de funcionalidad y mejoras posteriormente. La metodología usada ha sido iterativa e incremental, teniendo el claro objetivo de tener en el mismo punto de desarrollo las dos partes del proyecto.

Gameplay

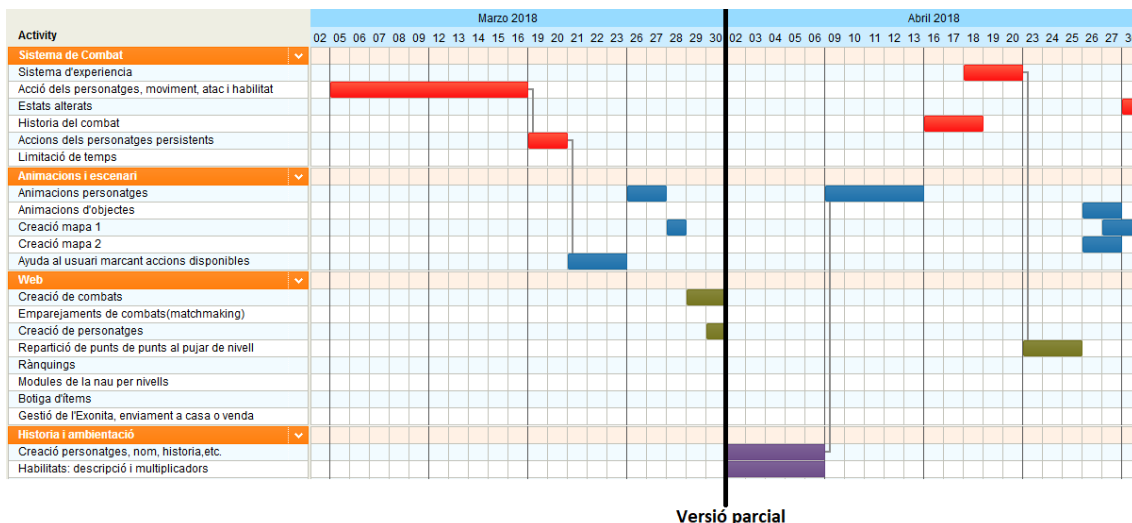
- Animaciones y movimientos de los personajes.
- Sistema de habilidades propias para cada personaje.
- Sistema de selección de acción a realizar por cada personaje.
- Mapa jugable.
- Sistema para recordar el historial de la partida en juego.
- Sistema de estados alterados para los personajes, como paralizado, etc.
- Limitación de tiempo para el jugador.
- Sistema de experiencia para los personajes.
- Interfaz de usuario dentro del gameplay (HUD).

Web

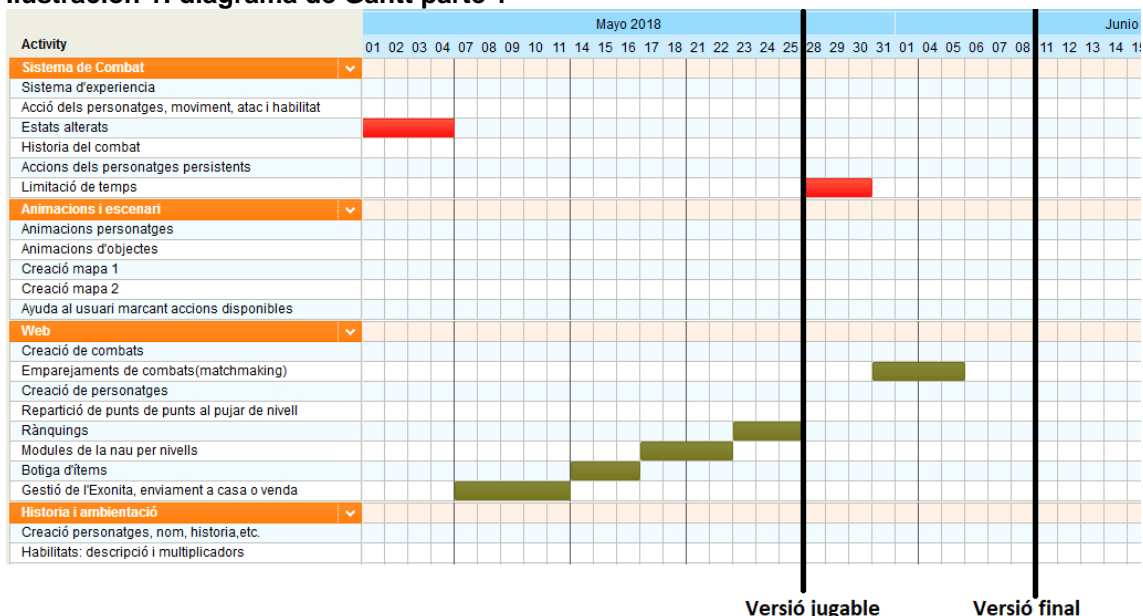
- Funcionalidad para crear partidas abiertas o limitadas por nivel.
- Creación y gestión de personajes.
- Sistema de puntuación para los jugadores.
- Sistema para desarrollar la nave del jugador.
- Tienda de items.
- Tienda de personajes.
- Sistema de gestión de la Exonita y Exons de los jugadores.

El núcleo del proyecto es la parte en la que dos jugadores están jugando una partida, y es la parte en la que mas tiempo se planeó invertir. Esta parte, no sólo es la interfaz del usuario en un navegador (front-end), si no que, conlleva toda la parte del servidor, donde realmente se realiza juego (back-end).

El proceso de desarrollo seguido, y las dependencias entre partes del proyecto, se puede observar en el siguiente diagrama de Gantt:



Il·lustración 1: diagrama de Gantt parte 1



Il·lustración 2: diagrama de Gantt parte 2

1.5 Breve resumen de productos obtenidos

El resultado del proyecto es una aplicación web que permite a los jugadores registrarse de forma gratuita y así, tener su cuenta de usuario con la que poder competir con otros usuarios.

En esta aplicación web, se encuentra incrustado el juego, al que se puede acceder a través de una de las secciones y de esta forma poder competir en el mapa isométrico. La web consta de todas las secciones necesarias para dar objetivos, profundidad y argumento al gameplay.

El producto obtenido es un juego de navegador multijugador masivo online con componentes de rol y un sistema de combate basado en turnos.

1.6 Breve descripción de los otros capítulos de la memoria

En los siguientes capítulos se explica el proceso seguido para producir el juego, la aplicación web y el sistema de combate descritos. Estos capítulos pasan desde la conceptualización e historia del juego hasta la tecnología usada, siguiendo éstos un orden de menos a más en lo relativo al nivel técnico de lo que se describe.

- Idea del juego: se describe el juego y sus características.
- Conceptualización: se describe la historia y ambientación del juego.
- Sistema de combate: se describe lo que el jugador puede ver y hacer dentro de una partida.
- Modos de juego: se describen los diferentes tipos de partidas.
- Secciones del juego: se describen las secciones de la web que envuelven al gameplay y el valor que aportan.
- Material gráfico: se describe el proceso seguido para producir la parte visual del juego.
- Diseño y desarrollo: se describe el software usado y el proceso seguido para diseñar y desarrollar todo el software del proyecto.

2. Idea del juego

2.1 Descripción

Exonite es un juego de navegador online donde los jugadores compiten en combates tácticos con equipos de 3 personajes. Cada jugador dispone de un equipo formado por diferentes personajes para poder elegir con cuales participar.

El combate finaliza cuando uno de los dos jugadores se queda sin personajes a utilizar.

Estos personajes, a medida que van siendo usados por el jugador van adquiriendo experiencia. Cuando acumulen la experiencia necesaria, el jugador podrá elegir que características de sus personajes quiere aumentar, haciendo que éstos mejores según su criterio.

El escenario donde estos combates se llevan a cabo, es en 2 dimensiones bajo una perspectiva isométrica i está dividido por casillas o por su término en ingles, tiles.

2.2 Género y referencias

El género del juego es un multijugador masivo online en navegador con un sistema de combate táctico basado en turnos con componentes de rol. Cabe diferenciar tres partes del juego para justificar el género mencionado, éstas son las siguientes:

- Multijugador masivo online (MMO): todos los jugadores compiten en el mismo universo al mismo tiempo de forma online entre ellos, por lo que el universo del juego siempre estará activo, independientemente de si un jugador está conectado o no.
- Componentes de rol: el sistema de experiencia que permite desarrollar a los diferentes personajes es muy frecuente en los videojuegos de rol y en cómo se desarrollan los personajes en éstos.
- Combate táctico basado en turnos (TTB): los combates entre los jugadores de Exonite son por turnos, un jugador realiza su jugada y cede su turno a su rival y espera movimientos. Este sistema de combate requiere un análisis previo de la situación de la partida, así como la de los personajes en juego, antes de realizar un movimiento, por lo que se requiere que el jugador aplique una estrategia o táctica para intentar vencer a su oponente.

Por la parte de MMO, como referencia existe el juego de navegador más popular, Ogame[7]. En este juego todos los jugadores compiten por recursos y tienen que ir desarrollando una serie de edificaciones para ser más poderosos.

Por otra parte, sobre el combate por turnos tácticos, las referencias son la saga Xcom[8] o la saga Final Fantasy Tactics[9], dos sagas donde los combates son por turnos y requieren un análisis estratégico de los movimientos a realizar.

2.3 Interacción con el jugador

Exonite, dispone de dos tipos de interacción con el jugador diferentes, por un lado tenemos el sistema de combate y por el otro las diferentes secciones de la web. Aun así, los dos tipos de interacción están fuertemente relacionados ya que las decisiones que se toman fuera del sistema de combate tienen un impacto importante en el combate y viceversa.

El jugador, para poder realizar diferentes estrategias dentro del sistema de combate, debe tomar unas decisiones en las diferentes secciones de la web que le permitan llevar a cabo las estrategias que desee. Como ejemplo, evolucionar uno de sus personajes sólo a nivel defensivo, le permitirá que dicho personaje se comporte diferente en el mapa de combate y pueda ser utilizado bajo un tipo de rol diferente.

A su vez, los resultados de los combates afectan a la interacción del jugador con la web, ya que si no resulta vencedor en las partidas, no podrá evolucionar rápidamente las diferentes características de su cuenta de usuario al no disponer de los recursos necesarios.

2.4 Plataforma destino

El objetivo final de un videojuego, es ser jugado y disfrutado por cuantos más jugadores mejor. Bajo esta premisa, la plataforma de destino elegida, como ya se ha comentado, es un navegador web.

Hoy en día, gracias al diseño adaptativo, una aplicación web puede ser usada en casi todos los dispositivos disponibles en el mercado por lo que no limita el alcance del videojuego a ninguna plataforma en exclusiva.

El diseño del juego ha priorizado el aspecto de la aplicación web en ordenadores personales con resolución full hd, aunque también se han dedicado esfuerzos en adaptar el resultado a las resoluciones más populares en teléfonos inteligentes.

3. Conceptualización

3.1 Historia y ambientación

La acción del juego sucede en un universo ficticio donde existe un mineral, que es el que da nombre al juego, llamado Exonita. Este mineral es comparable al agua en el planeta Tierra, es el principal sustento de la vida existente.

La importancia de la Exonita en el universo es tal, que tanto la vida como la economía se basan en ella. El sistema monetario del universo, y por lo tanto del juego, es llamado Exon, y está ligado a la Exonita con un factor de comercio, el precio de un Exon en Exonita es variable según el mercado.

El juego pone al jugador dentro del rol de que ha sido el elegido por su civilización para dirigir la misión más importante de su pueblo, la búsqueda de Exonita. El valioso y útil mineral escasea en su planeta y la vida está seriamente comprometida, por lo que el destino de toda su especie está en sus manos.

Bajo esta historia, el jugador recibe una nave espacial que le servirá para poder ir desplazándose por el universo en busca de Exonita. Esta nave, puede ser mejorada para aportar nuevas funcionalidades al jugador a cambio de Exons.

Así mismo, el jugador también recibe un presupuesto inicial de Exons para poder formar un equipo de protección con el que poder realizar incursiones (Raids) en el universo. Este equipo, está formado por clones de los que fueron considerados los mejores incursores (Raiders) de Exonita del universo.

3.2 Definición de los personajes

Los clones que el jugador puede comprar con la moneda del juego, le otorgarán protección cuando llegue a un filón de Exonita, ya que ante la escasez de dicho material la explotación de un yacimiento comporta una casi segura confrontación con otro equipo de exploradores.

Estos personajes, son copias exactas de héroes del universo que en su día fueron considerados los mejores incursores y que por sus habilidades, la Estación Espacial decidió clonar.

Cada uno de ellos, son únicos i tienen un nombre, historia y habilidades propias. Pese a que son clones, no deben ser considerados el mismo personaje o miembros de una clase, ya que cada una de las copias es única y dependerá de cómo se evolucione.

3.3 Definición de los elementos del juego

3.3.1 Nave espacial

La nave del jugador, dispone de la capacidad de ser mejorada de forma modular. Cada una de las funcionalidades base de las que dispone puede ser potenciada bajo un sistema de niveles y de esta manera ofrecer una nueva funcionalidad o una mejora en la funcionalidad existente.

La nave dispone de 6 módulos.

- Módulo de mando: contiene los mandos del sistema de propulsión de la nave y de los suministros. Este módulo permite que un mismo integrante de la tripulación pueda participar en diferentes raids al mismo tiempo. Cada nivel de mejora aumenta las raids simultáneas en las que los personajes pueden participar.
- Módulo de Tripulación: este módulo proporciona diferentes elementos para el soporte vital dentro de la nave. El módulo de tripulación, permite poder alojar a más raiders en la nave. Cada nivel de mejora aumenta el número total de personajes que puede tener el jugador en su equipo.
- Módulo de Aterrizaje: es una cápsula que lleva a la tripulación a la superficie de los planetas. El módulo de Aterrizaje permite enviar a la tripulación a más raids de forma simultánea. Cada nivel de mejora aumenta las raids simultáneas en las que un jugador puede participar.
- Módulo de Escape: también conocido como sistema de aborto de misión, es un mecanismo que pone en órbita rápidamente el módulo de aterrizaje en caso de emergencia. El módulo de Escape, permite abandonar una partida en curso sin tener que esperar a que finalice. Cada nivel de mejora permite abandonar la partida con menos turnos de espera.
- Módulo de Carga: este módulo permite aumentar la capacidad de transporte de la nave, facilitando el envío de Exonita al planeta de origen y el comercio. Cada nivel de mejora aumenta la capacidad de carga del módulo.
- Radar: es un sistema avanzado de rastreo espacial que permite escanear el universo para poder ver la actividad de flotas enemigas. Cada nivel de mejora permite acceder a un tipo de ranking diferente.

3.3.2 Estación Espacial

Para poder formar un equipo, el jugador deberá dirigirse a la Estación Espacial, donde se encuentra la tienda que pone a la venta los personajes. Esta estación, es un ente común en todo el universo construido por los grandes poderes fácticos en la galaxia siguiendo un plan con el único objetivo de amasar toda la Exonita posible y de esta manera conservar y concentrar todo el poder.

Con este fin, ofrecen una serie de servicios como son la venta de clones para proteger las misiones de búsqueda de Exonita, un mercado donde poder comerciar con el material y el servicio de envío de Exonita al planeta natal por un porcentaje del envío.

3.4 Interacción entre los actores del juego

Los jugadores pueden interactuar con otros jugadores creando partidas o accediendo a partidas creadas por otros jugadores. Una vez una partida dispone de los dos jugadores necesarios comienza el combate para ver quién se queda con la Exonita localizada en ese planeta.

Durante el turno de cada jugador, este, debe decidir las acciones que quiere que todos sus personajes realicen. Cada personaje, en un turno puede realizar un movimiento y usar el ataque básico o una de sus habilidades. Estas acciones pueden ser ejecutadas en el orden que el jugador decida.

Cuando todos los personajes de uno de los jugadores han sido vencidos, la partida finaliza y se reparten los beneficios. Tanto el ganador como el perdedor, consiguen Exonita y experiencia para sus personajes, pero cada uno en diferente medida.

También, los jugadores pueden interactuar con la Estación Espacial. En este apartado, los jugadores pueden vender Exonita por Exons en el mercado de la Estación y/o enviar la Exonita a casa. Las dos acciones sólo pueden ser realizadas una vez cada 12 horas.

3.5 Objetivos del jugador

El principal objetivo del jugador es ser el número 1 del ranking principal. Este ranking está basado en la Exonita que cada jugador ha podido enviar a su pueblo para que la procese y de esta manera sustentar la vida en su planeta.

Como se ha dicho, cada vez que un jugador combate, gane o pierda, recibe Exonita y con ésta podrá decidir si la envía a casa usando el muelle de carga de la Estación Espacial o si la intercambia por Exons en el mercado de Exonita.

A parte del ranking principal, existen tres rankings alternativos. Estos rankings están basado en los Exons invertidos en mejoras de la nave espacial, Exons invertidos en comprar nuevos personajes, y en los puntos de combate ganados.

4. Sistema de combate

4.1 Acciones

Los personajes del jugador dentro de una partida pueden realizar diferentes acciones, siendo el jugador quien decide el orden bajo su propio criterio.

Para realizar estas actividades, el jugador debe seleccionar el personaje que quiere usar y acto seguido seleccionar el tipo de acción. Dependiendo de la acción seleccionada, el sistema de combate mostrará al jugador de forma visual los posibles movimientos que puede realizar cada uno con su código de colores. Para confirmar la acción, deberá seleccionar la casilla sobre la cual quiere realizar la acción.

Para poder realizar las acciones, el jugador dispone de un hud en el que puede ver la información de cada uno de sus personajes y las acciones que tiene disponibles.

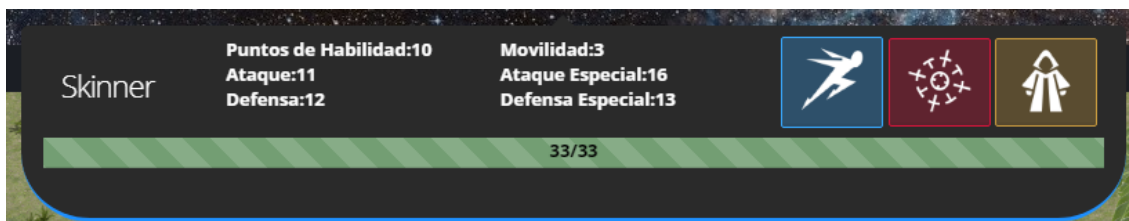


Ilustración 3: HUD de acciones

Estas acciones son las siguientes:

- Movimiento: permite al personaje desplazarse a una casilla diferente. El indicador visual de a dónde puede desplazarse es en color azul igual que el botón a seleccionar en el hud.



Ilustración 4: selección movimiento

- Cancelar movimiento: permite al personaje que acaba de realizar un movimiento deshacerlo y volver a su ubicación inicial. El botón para cancelar un movimiento es el mismo que para moverse pero en color negro.
- Ataque básico: realiza un ataque básico sobre un enemigo, realizándole un daño. El indicador visual de a donde alcanza el personaje a atacar aparece en color rojo, igual que el botón a seleccionar en el hud.



Ilustración 5: selección ataque

- Usar habilidad: cada personaje dispone de dos habilidades únicas, con diferente potencia y efectos. Para poder seleccionar una habilidad, se dispone de un botón amarillo en el HUD que abre un desplegable con las habilidades disponibles. El indicador visual también es amarillo.



Ilustración 6: selección habilidad

Cuando el personaje ya ha realizado una de las acciones disponibles, el botón de tal acción desaparece de su menú.



Ilustración 7: HUD; acciones usadas

Para las acciones de ataque básico y habilidades, los personajes tienen una animación de cast en forma de aura con el color de la acción seleccionada.

4.2 Pantalla de combate

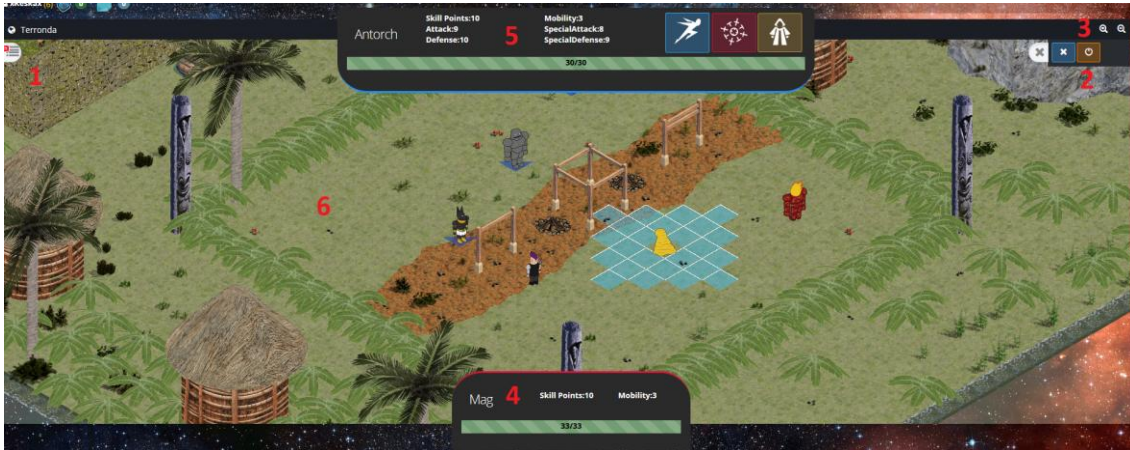


Ilustración 8: pantalla de combate

En la pantalla de combate aparece toda la información de los personajes del jugador activo (5) y parte de la información de los personajes de su enemigo (4). En esta pantalla, aparte de controlar los personajes y sus acciones, se dispone de un historial de las acciones realizadas por los jugadores (1).

También, en esta pantalla, se encuentra una sección con las opciones del combate (2). En estas opciones, el jugador puede deselegccionar la acción marcada, ceder su turno o si tiene los permisos necesarios, abandonar la partida en curso rindiéndose. Justo encima de las opciones del combate se encuentran los botones para ampliar o reducir el mapa (3).

Toda la demás área visible (6), es el mapa interactivo de juego donde se encuentran los personajes de los dos jugadores. Cabe destacar que los jugadores del usuario activo, tienen una señal azul en sus pies para facilitar su reconocimiento cuando hay más de un personaje igual dentro de una partida. Esta señal, también está indicando al usuario el área donde debe pulsar cuando quiere seleccionar a su personaje.

Como control adicional sólo disponible cuando se accede al desde un dispositivo con un ratón disponible, la pulsación del botón derecho de éste es análoga al botón de deselegccionar acción de la sección de opciones de combate.

Finalmente, en esta misma pantalla, cuando un personaje es dañado, justo encima de su posición aparece un indicador visual con el daño que ha recibido.



Ilustración 9: indicador daño

4.3 Pantalla de resultados

Cuando uno de los jugadores pierde todos sus personajes y finaliza la partida, aparece la pantalla de resultados. En esta pantalla se muestran todos los datos de la partida, de forma general para cada jugador y de forma detalla para los personajes del jugador activo.

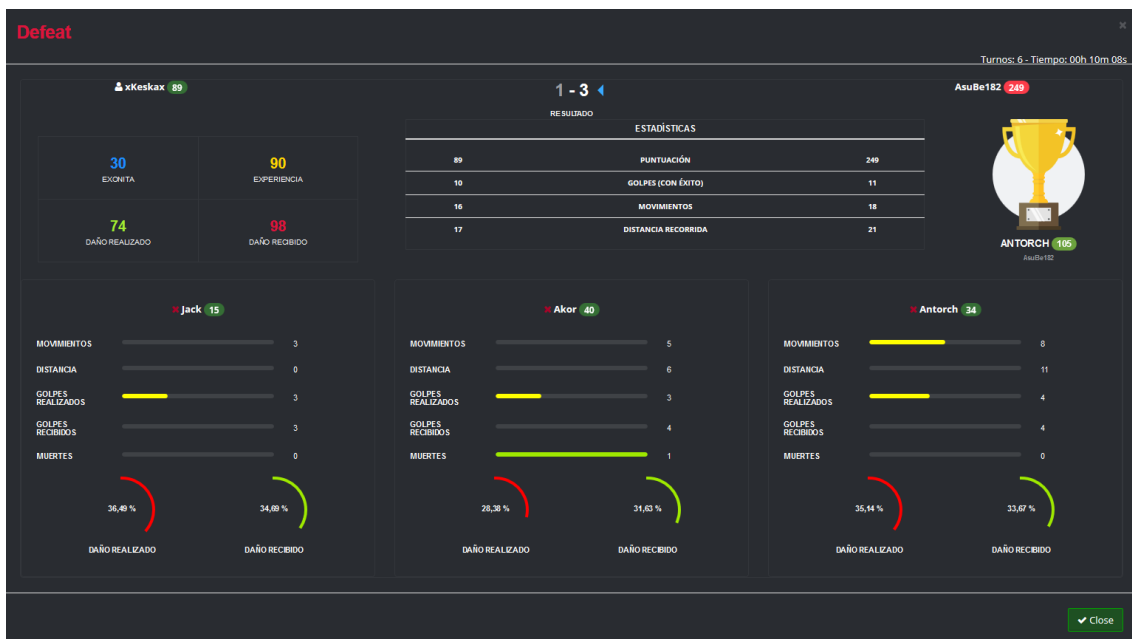


Ilustración 10: pantalla de resultados

Entre estos datos, se encuentra la Exonita que ha obtenido el usuario y la experiencia que han ganado sus personajes. También, existe el rol del personaje MVP, que se trata del personaje entre los 6 totales que mas puntuación ha conseguido en base a su actuación personal.

4.4 Reproducción turno anterior

Para dar una sensación más inmersiva al juego, una vez que un jugador accede a una partida cuando es su turno tras el turno de su rival, podrá ver los movimientos que ha realizado éste durante su turno, como si de una repetición se tratase. Esta funcionalidad ayuda a incrementar la sensación de estar jugando contra otro jugador y hace el juego más dinámico.

Esta repetición, puede ser reproducida tantas veces como el usuario desee, sólo es necesario actualizar la página en el navegador sin haber realizado ninguna acción en su turno.

Además, en el caso de que un usuario intente acceder a una partida en la que su rival ya ha salido vencedor, la primera vez que acceda verá la repetición de cómo su rival terminó el combate.

5. Modos de juego

La idea inicial para el proyecto sólo incluía el modo de juego jugador contra jugador, pero bajo la recomendación del equipo de consultores se ha decidido implementar un modo de juego contra un usuario bot.

Este modo de juego, está pensado para cuando no hay jugadores humanos disponibles y también, para facilitar el testeo y corrección del proyecto. El bot, pese a que no dispone una inteligencia avanzada, ha demostrado poder poner en apuros a un jugador humano.

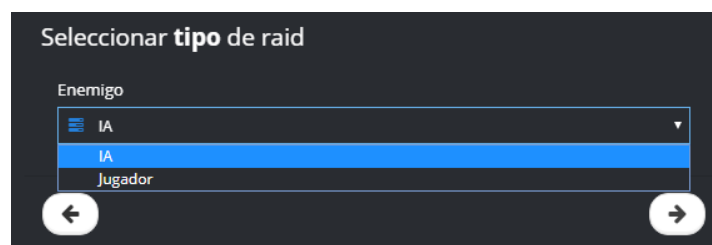


Ilustración 11: modos de juego

5.1 Jugador contra Jugador

Es el modo de juego principal, dos jugadores humanos compiten por la Exonita que se reparte en cada partida. El ganador obtiene 100 unidades de material, mientras que el perdedor sólo 30. Los personajes del ganador obtienen cada uno 100 puntos de experiencia mientras que en el caso de los del perdedor sólo son 30 puntos.

Dentro de este modo de juego, hay dos variantes. Por un lado existe el tipo de juego Abierto. Este tipo crea una partida sin limitaciones de nivel para los rivales.

Por otro lado, el tipo de juego Limitado sólo permite acceder a la misma partida a jugadores que seleccionen los personajes en el mismo rango de nivel que el usuario que ha creado la partida.

El tipo de juego, es una elección del jugador que inicia la partida y éste se puede seleccionar en el momento de creación de la partida, en la misma pantalla donde el jugador selecciona con que personajes quiere participar en la partida.

Este modo de juego, produce una serie de notificaciones emergente para los usuarios que están participando en una partida. Cuando un rival del usuario activo ha finalizado su movimiento, se envía una notificación al navegador del usuario para infórmale rápidamente de que puede empezar su turno. De esta manera, se intenta agilizar el ritmo de juego por parte de los usuarios, que serán informados tan pronto como sea posible hacer un movimiento.

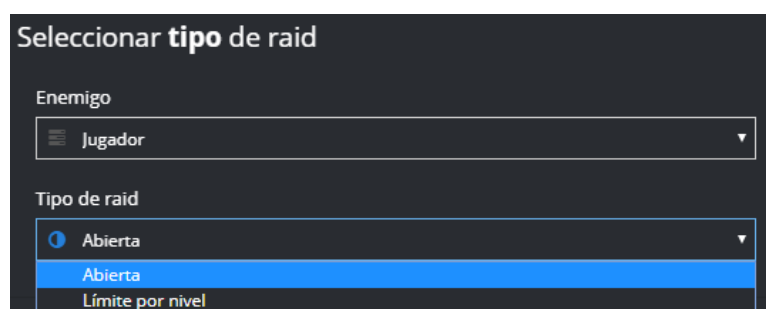


Ilustración 12: tipos de juego modo JcJ

5.2 Jugador contra bot

En este modo de juego, el flujo de la partida se ve modificado en comparación al modo JcJ. Cuando un jugador inicia una partida contra el bot, automáticamente ésta queda disponible para empezar a ser jugada, sin tener que esperar a que un segundo jugador aparezca.

A su vez, cuando el jugador humano finaliza su turno, el usuario bot realiza su turno automáticamente y el jugador puede ver los movimientos que el bot ha decidido realizar sin tener que salir de la partida ni tener que cargar la pantalla de combate de nuevo.

Por lo demás, los dos modos de juego disponible, JcJ y JcE, son exactamente iguales, y reportan al usuario el mismo tipo de beneficios tanto como si gana o pierde la partida.

Respecto al funcionamiento del bot y la estrategia que sigue en combate, su directriz principal es siempre intentar atacar al rival con menos vida que se encuentre dentro de su rango de acción.

Para lograr este objetivo, durante el turno del bot y mediante el algoritmo A* se determinan las distancias entre los personajes del bot y los del jugador humano. Tras tener a los objetivos localizados, se busca siempre al rival más flojo para ser atacado con la habilidad o ataque más potente que disponga el personaje controlado.

6. Secciones del juego

6.1 Página de aterrizaje/acceso

Esta es la primera toma de contacto del usuario con el juego y está pensada para atraer a los posibles usuarios con mensajes claros y directos.

Consta de un slider en el que en su estado inicial muestra imágenes del juego y algunos mensajes comerciales. En el segundo estado del slider, el visitante de la web puede ver un video demostración de la aplicación con más mensajes comerciales superpuestos.

Para poder llevar a cabo su principal objetivo, captar usuarios y que accedan al juego, consta de un formulario para poder crear una cuenta de usuario y el formulario de acceso al juego para usuarios ya registrados.



Ilustración 13: página de aterrizaje/acceso

6.2 Raids

En la sección de Raids es donde los usuarios pueden crear y unirse a partidas. Con este objetivo, la sección dispone de 3 partes diferenciadas.

- Tus Raids: en este apartado, el jugador puede ver las partidas en las que se encuentra activo, tanto si son partidas a las que él se ha unido como si han sido creadas por el propio jugador. También dispone del formulario de creación de partidas. En esta sección un jugador puede cancelar una partida siempre y cuando ningún otro jugador se haya unido.

Tus raids

Raids que han sido iniciadas por tu tripulación.

CREADA	PLANETA	ENEMIGO	NIVEL	TIPO	
27/05/2018 09:13:12	Terronda	seprasam	3	Abierta	
27/05/2018 21:44:50	Terronda	Sin enemigos a la vista	3	Abierta	

[Crear Raid](#)

Ilustración 14: Tus Raids

- Raids interceptables: aquí el jugador encuentra las partidas de tipo abierto que han creado otros jugadores y puede unirse a ellas. Están ordenadas de forma cronológica, mostrando antes las partidas que más tiempo hace que fueron creadas. A su vez, en esta sección, está disponible el formulario para buscar partidas de tipo limitado.

Raids interceptables

Raids de límite abierto iniciadas por tripulaciones enemigas.

CREADA	PLANETA	ENEMIGO	NIVEL	
27/05/2018 21:11:30	Terronda	AsuBe182	3	
27/05/2018 21:11:41	Terronda	AsuBe182	3	
28/05/2018 02:09:52	Terronda	VictorSbD	6	
28/05/2018 16:44:55	Terronda	kloyrT	3	
28/05/2018 17:11:28	Terronda	VictorSbD	6	
28/05/2018 17:11:38	Terronda	kloyrT	3	
28/05/2018 20:38:13	Terronda	Aureli23	3	
28/05/2018 20:47:15	Terronda	qneb82	3	
28/05/2018 20:54:31	Terronda	xPepCatx	3	
28/05/2018 20:54:40	Terronda	xPepCatx	3	

← « 1 2 » →

[Buscar Raids](#)

Ilustración 15: Raids interceptables

- Raids detectadas: una vez que se ha usado el formulario para buscar partidas de tipo limitadas, aparece esta sección. En ésta, las partidas que coinciden con el nivel del equipo que el usuario ha introducido en la búsqueda son mostradas.

CREADA	PLANETA	ENERGIA	NIVEL	
27/05/2018 21:43:02	Terronda	AsaBe182	3	<input type="checkbox"/>
28/05/2018 17:11:48	Terronda	kloyT	3	<input type="checkbox"/>
28/05/2018 20:36:25	Terronda	Auril823	3	<input type="checkbox"/>
28/05/2018 20:46:59	Terronda	qieB82	3	<input type="checkbox"/>
28/05/2018 03:14:40	Terronda	qieB82	3	<input type="checkbox"/>

Ilustración 16: Raids detectadas

6.3 Tripulación

La tripulación de la nave está formada por los raiders que el usuario ha comprado en la tienda de la Estación Espacial. En esta sección el usuario puede consultar y gestionar todos los datos relativos a todos y cada uno de sus personajes.

Puede consultar las propiedades de sus personajes y las estadísticas de éstos en combate. A su vez, en esta sección, el usuario puede ver los puntos de experiencia de los personajes, y los puntos necesarios para llegar al siguiente nivel.

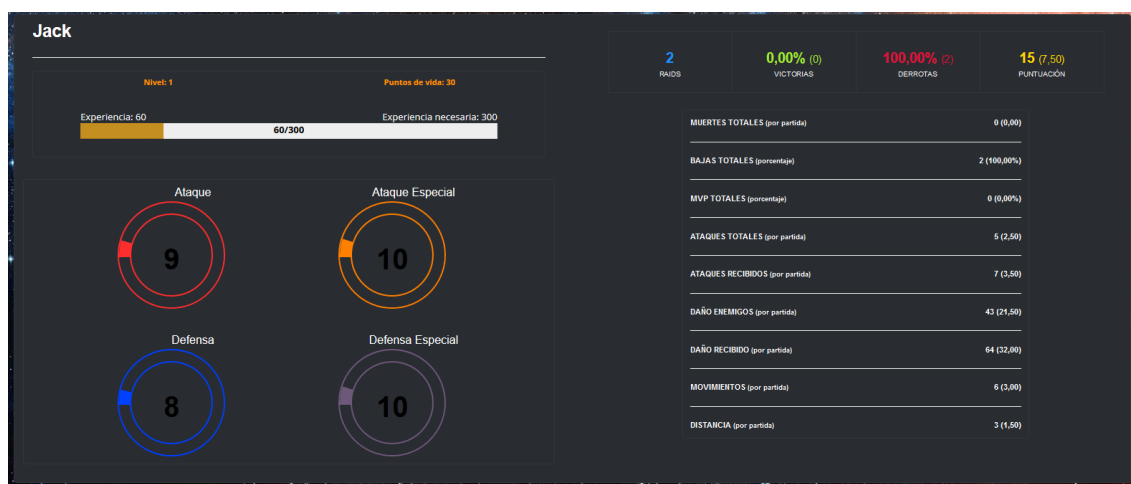


Ilustración 17: detalles del personaje

Llegado el momento, los personajes tendrán la suficiente experiencia para subir de nivel, apareciendo de esta manera el formulario para poder repartir los puntos de habilidades otorgados. Estos puntos son utilizados libremente bajo el criterio del usuario.



Ilustración 18: formulario subir de nivel


6.4 Bitácora

Esta sección funciona a modo de registro para los resultados del usuario. Permite volver a ver los resultados de todas y cada una de las partidas en las que ha participado. A su vez, el cuaderno de bitácora, registra el porcentaje de éxito de las actuaciones y puntuaciones del usuario en combate.

Bitácora

El cuaderno de bitácora registra los datos de lo acontecido durante tus raids. Este cuaderno permite volver a ver los resultados de todas y cada una de las raids en las que has participado.

A su vez, el cuaderno de bitácora de tu nave, registra el porcentaje de éxito de tus actuaciones y tu puntuación de combate.



8 RAIDS	100.00% (6) VICTORIAS	0.00% (0) DERROTAS	2.004 (334.00) PUNTUACION	
FINALIZADA	NIVEL	MAPA	TIPO	JUGADOR
29.05.2018 14:38:07	6	Terronda	Abierta	J1
29.05.2018 14:24:14	6	Terronda	Abierta	J1
28.05.2018 17:59:46	3	Terronda	Abierta	J1
27.05.2018 23:51:16	3	Terronda	Abierta	J1
27.05.2018 23:59:24	3	Terronda	Abierta	J1
27.05.2018 09:26:07	3	Terronda	Abierta	J1

Ilustración 19: Bitácora

6.5 Módulos de la nave

En esta sección el usuario puede consultar el nivel de todos y cada uno de los módulos de su nave. Si además, el usuario dispone de los Exons suficientes para poder comprar un módulo, y dicho módulo tiene un nivel de mejora disponible, el usuario podrá invertir los Exons necesarios en mejorar la nave.

Cuando el usuario no puede mejorar un módulo, porque o bien no dispone de Exons suficientes o dicho módulo ha alcanzado su máximo nivel, la imagen en miniatura del módulo es mostrada en escala de grises.

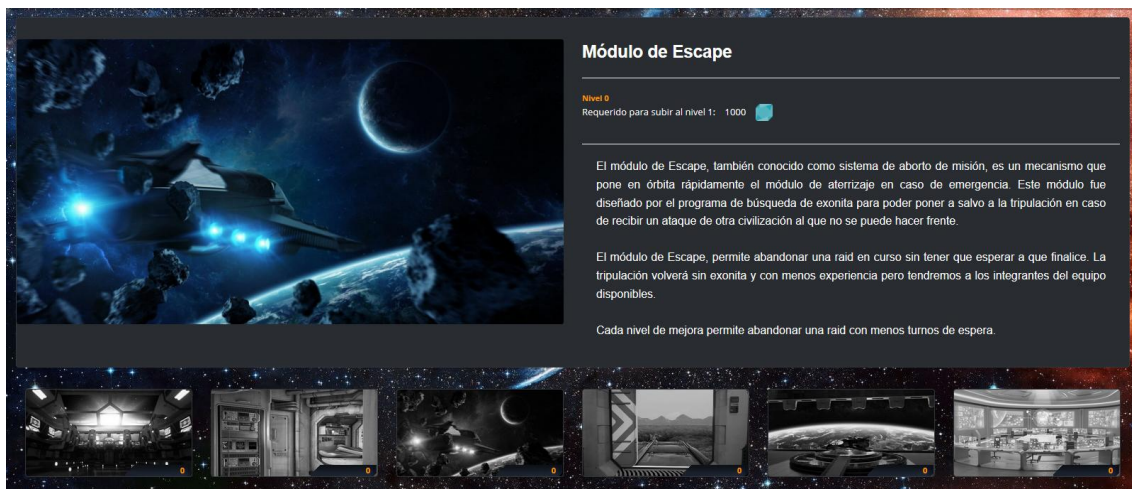


Ilustración 20: Módulos

6.6 Mercado de Exonita

El Mercado de Exonita dentro de la Estación Espacial es la sección donde los usuarios pueden intercambiar sus reservas de Exonita por Exons.

La cantidad de Exons por la que la Exonita puede ser vendida depende de un factor de comercio que se renueva de forma diaria. El valor oscila entre 0.8 y 2.2, mostrándose en verde cuando el valor actual esté cerca del máximo y en rojo cuando el valor esté cerca del mínimo.

De esta manera, los usuarios podrán identificar cuando sale más rentable vender sus reservas de Exonita en el caso de que quieran hacerlo.

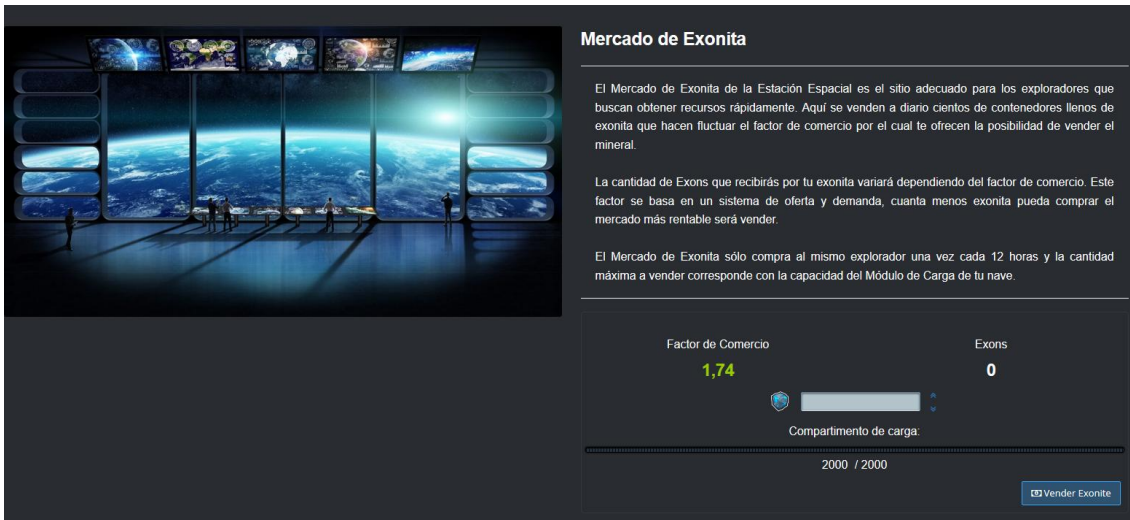


Ilustración 21: Mercado de Exonita

La capacidad máxima que el Mercado de Exonita compra a cada usuario depende del Módulo de Carga de su nave, siendo su capacidad el límite de la transacción. A su vez, el mercado, sólo compra Exonita a un mismo usuario cada 12 horas.

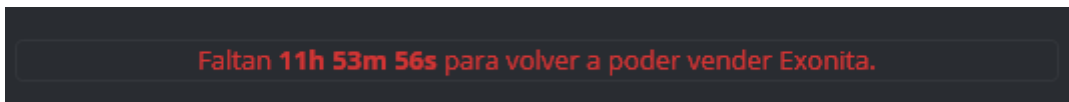


Ilustración 22: mensaje Mercado de Exonita

6.7 Tienda de Raiders

En sección ofrece la funcionalidad al usuario de poder comprar nuevos personajes para que se unan a su tripulación. Como en la sección de módulos, si el usuario no dispone de los Exons suficientes para comprar un personaje, su miniatura aparece en escala de grises.

También, en esta sección es posible consultar la historia personal de cada uno de los personajes del juego, así como las características y descripción de sus habilidades especiales.

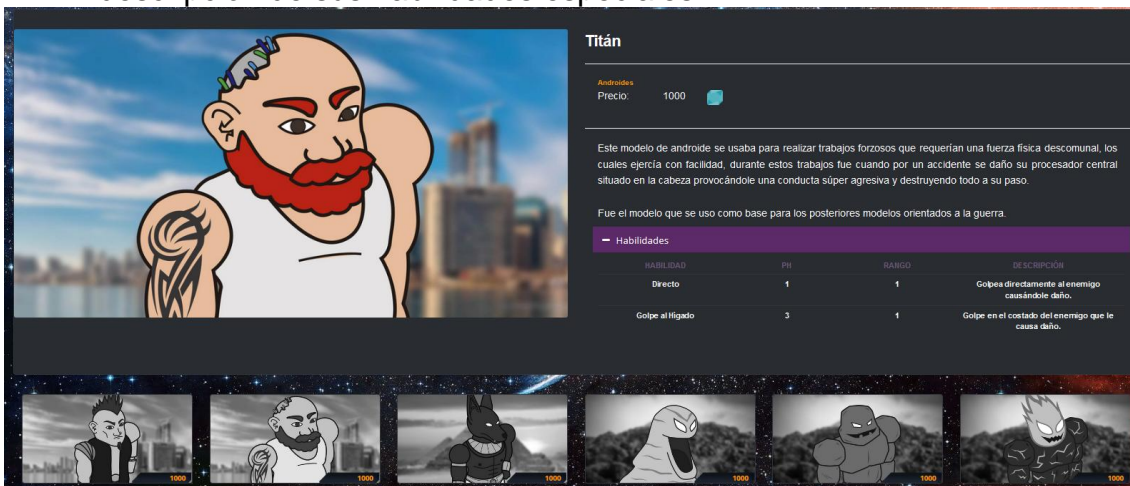


Ilustración 23: Tienda de Raiders

6.8 Muelle de Exonita

El Muelle de Carga de la Estación Espacial ofrece la posibilidad al usuario de enviar a su planeta la Exonita que ha conseguido durante las partidas. De esta manera, podrá conseguir puntos computables en el ranking principal.

La Estación Espacial no ofrece este servicio de forma gratuita para el usuario, por lo que se queda con una parte de lo que el usuario decide enviar. De esta manera, 1 unidad de Exonita corresponde a 0.1 puntos del ranking, o lo que es lo mismo, para obtener 1 punto en el ranking es necesario enviar 10 unidades de Exonita.

El Muelle de Carga, como el Mercado de Exonita, sólo permite un envío cada 12 horas y la capacidad máxima a enviar corresponde con la capacidad del Módulo de Carga de la nave del usuario.



Ilustración 24: Muelle de Exonita

6.9 Ranking

En la sección de rankings, el usuario puede consultar hasta 4 tipos de rankings diferentes. Éstos, serán visibles para cada usuario según el nivel del módulo de Radar de su nave.

Los tipos de ranking disponibles son los siguientes:

- Principal: los puntos que determinan la posición de un jugador en este ranking se obtienen de la Exonita que ha sido procesada en su planeta de origen.
- Nave: los puntos que determinan la posición de un jugador en este ranking se obtienen de los Exons invertidos al mejorar los módulos de la nave.
- Ejército: los puntos que determinan la posición del jugador en este ranking se obtienen de los Exons invertidos en su tripulación.

- Combate: los puntos que determinan la posición de un jugador en este ranking se obtienen de los puntos obtenidos tras cada partida.

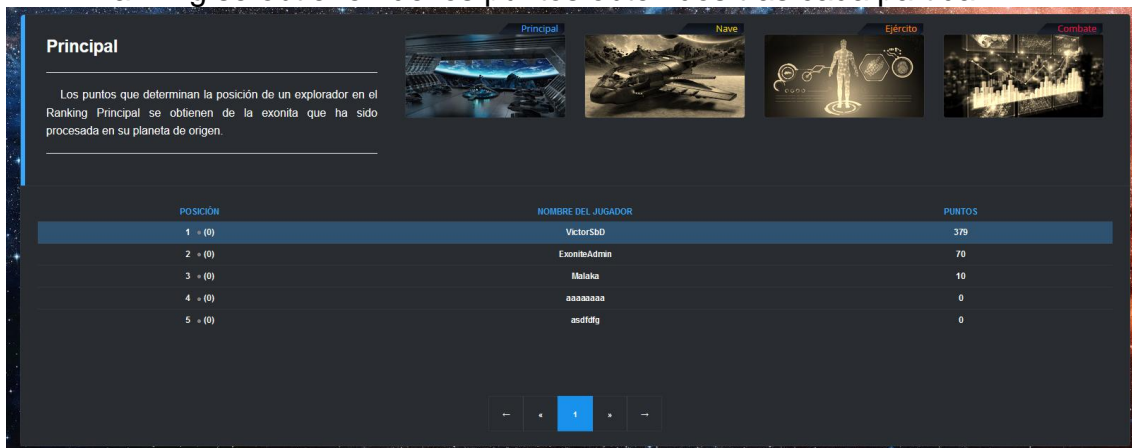


Ilustración 25: Rankings

Todos los rankings, son actualizados cada 5 minutos y de esta manera, muestran las variaciones ocurridas en estos periodos.

3 = (0)	ermionistranger	36
4 < (-1)	qneb82	13
5 < (-1)	AsuBe182	10
6 < (-1)	Aurel23	7
7 < (-1)	xKeskax	6
8 < (-1)	kloyrT	3
9 < (-1)	ExoniteAdmin	0
10 < (-1)	Malaka	0

Ilustración 26: variaciones en el ranking

6.10 Tutorial

El tutorial del juego consta de 6 tareas en las que el usuario deberá cumplir una serie de misiones para recibir una recompensa.

De esta manera, se va explicando a los usuarios como jugar de forma gradual y motivándoles a seguir jugando. El objetivo del tutorial es intentar reducir la sensación de no saber lo que hay que hacer y que por consiguiente los jugadores acaben abandonando el juego, desperdiciando la oportunidad que le dieron al registrarse.



Ilustración 27: botón acceso tutorial

El botón de acceso a la sección tutorial siempre estará presente ya que aunque el usuario lo haya completado, siempre servirá de sección de ayuda, ya que cada tarea dispone de los pasos a seguir para completarla, que son los mismos pasos a seguir para aprender a jugar.

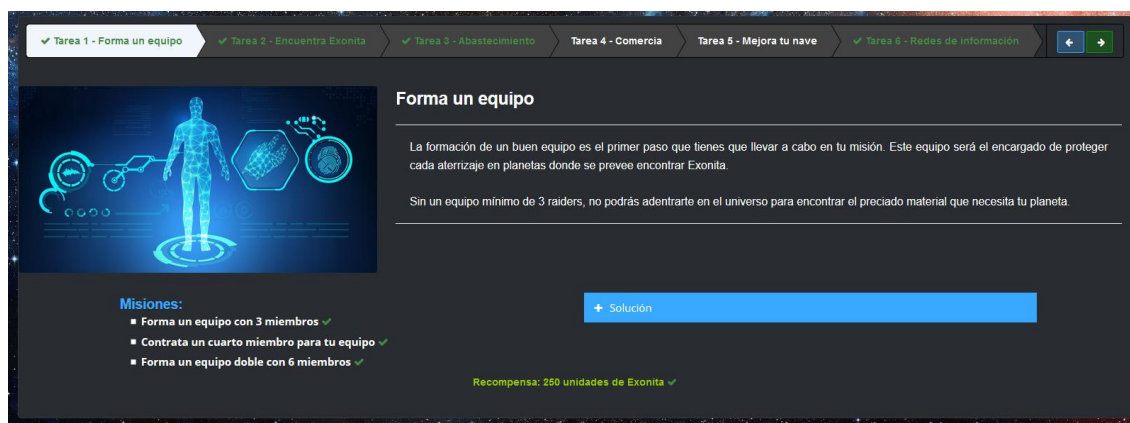


Ilustración 28: tutorial

7. Material multimedia

Para el apartado multimedia se han barajado dos posibilidades desde el inicio del proyecto, realizar todo el material desde cero, pensado y creado exclusivamente para el juego o utilizar material gratis y/o de pago disponible en internet.

Con el objetivo de crear un proyecto con identidad propia, la decisión que se ha tomado ha sido la de realizar en la medida de lo posible todo el material de forma exclusiva para el juego y en los casos en los que no sea posible, mayoritariamente por falta de conocimientos en la materia por parte del autor, acudir a utilizar el material de terceros.

En consecuencia a lo expuesto, este capítulo del juego, es el único en el que el autor no es la única persona que está involucrada.

7.1 Diseño de personajes

Para este apartado, el autor ha contado con la colaboración del ilustrador Albert Pérez Verdugo. Su colaboración ha consistido en dar vida y movimiento a los personajes.

Dado que todos los personajes han sido creados de forma exclusiva para el juego, tanto sus animaciones como apariencia e historia, es posible relatar el proceso seguido. Dicho proceso ha sido el siguiente:

1. Esbozos de los personajes en papel.



Ilustración 29: esbozo personajes

2. Escaneo del esbozo elegido para poder ser tratado digitalmente.
3. Proceso de vectorización usando el software Adobe Illustrator.
4. Realización de las animaciones usando el mismo software.
5. Exportación de los assets producidos para poder ser tratados en Adobe Photoshop.
6. Retoques y ajustes finales a los personajes para producir la imagen final que es utilizada en el mapa isométrico.

En la siguiente imagen se puede apreciar el resultado final del proceso relatado:



Ilustración 30: animación personaje

7.2 Mapas; objetos y tiles

Para la realización del material multimedia usado en la creación del mapa, el autor ha contado con la colaboración del ilustrador Adrián Velarde Torres. Su colaboración ha consistido en crear objetos e imágenes con perspectiva 2D isométrica para poder ser utilizados en el mapa.

Para el diseño del mapa, una mayor parte del material ha sido creado partiendo de imágenes reales. Pese a la decisión final tomada de crear texturas a partir de imágenes reales, también se realizaron pruebas creando el contenido desde cero con Adobe Illustrator.

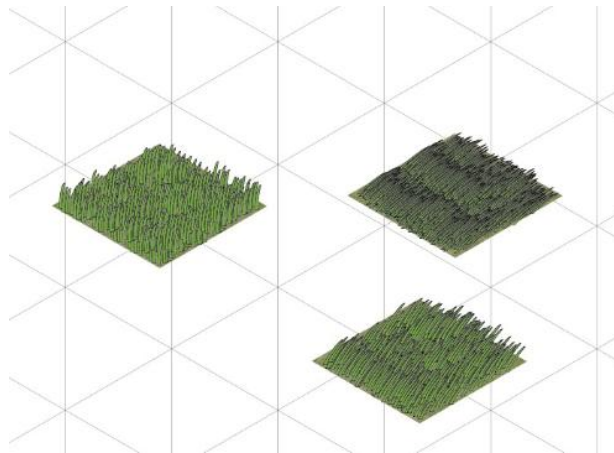


Ilustración 31: objetos mapa, pruebas realizadas

Como se puede ver en la ilustración 29, este fue el resultado obtenido utilizando dicha estrategia. No fue satisfactorio tanto para el ilustrador como para el autor, por lo que, finalmente, se decidió crear las texturas para los objetos partiendo de imágenes reales.

El proceso seguido ha sido el siguiente:

1. Esbozo en papel del objeto a producir.
2. Búsqueda de las texturas necesarias, bien en internet o produciéndolas.
3. Creación de imágenes con perspectiva isométrica usando el software Adobe Illustrator.
4. En el caso de tiles, separación de objetos complejos en imágenes del tamaño de las casillas.

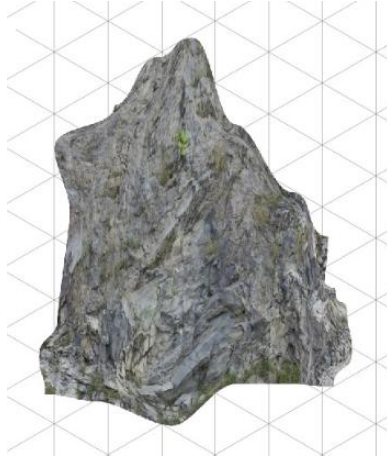


Ilustración 32: proceso creación objetos mapa



Ilustración 33: aspecto final objetos mapa

7.3 Imágenes

Para las imágenes usadas en el juego, la decisión inicial era la misma que se ha seguido con los objetos y personajes. Pero, los resultados obtenidos creando las imágenes desde cero de forma exclusiva para el juego, desde una fase muy temprana en el desarrollo se pudo observar de forma clara que no tendrían el nivel mínimo exigido por el autor.

De esta manera, se replanteó la decisión inicial y se cambió la estrategia. Se buscaron imágenes de autor sin royalties que ofrecieran una visión similar a la que el autor tenía proyectada en mente. En esta búsqueda, se encontró una colección que satisface lo requerido.

7.4 Música

Para el apartado musical del proyecto, desde el inicio no se ha barajado la posibilidad de incluir música en el juego ya que la inclusión de un apartado sonoro en una página web no suele despertar una buena aceptación entre los usuarios. Aun y así, la posibilidad de añadir sonidos a las acciones de los personajes si que se tuvo en cuenta de forma positiva, pero se sacó fuera del alcance del proyecto al realizar las estimaciones iniciales.

7.5 Efectos habilidades

Para los efectos de las habilidades, tanto el efecto que ocurre en la casilla donde se lanza el ataque como el aura que envuelve al personaje atacante, han sido adaptados para ser usados en vista isométrica recursos gratuitos de la plataforma footagecreate[10]. En esta web se pueden descargar algunos recursos de forma gratuita, con el copyright adecuado para ser usados, en formato video. Estos videos con los efectos han sido separados frame por frame y convertidos en un sprite para poder ser usado en el juego.

Para separar los frames de los videos se ha usado la plataforma ezgif[11], mientras que para paquetizar los frames se ha usado el software TexturePacketGUI[12].

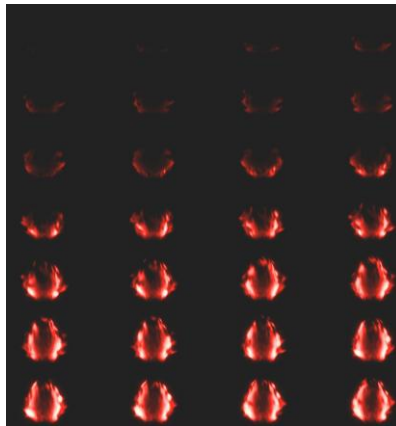


Ilustración 34: animación aura ataque

8. Diseño y desarrollo

Antes de empezar cualquier desarrollo, es vital realizar una fase de análisis y diseño previo. Las decisiones que se toman durante esta fase, serán las que faciliten en mayor medida las posibilidades de éxito del proyecto.

Conociendo este hecho, antes de empezar a desarrollar el proyecto, se han tomado una serie de decisiones que han marcado la arquitectura, diseño y desarrollo del mismo.

El modelo de arquitectura del proyecto, en parte viene impuesto por la plataforma de destino elegida para el juego, un navegador web. Pero aun y así, hay decisiones en este ámbito que deben ser tomadas por el equipo de desarrollo.

Una de estas decisiones, ha sido la de elegir la tecnología con la que llevar a cabo el proyecto. Como ya se ha indicado en puntos anteriores, el proyecto es una aplicación web nativa, y ha sido desarrollado utilizando tecnología del entorno Microsoft ASP.NET MVC.

Como alternativas a esta decisión, han sido barajadas las posibilidades de desarrollar el proyecto en un entorno Java EE o PHP, pero han sido descartadas, ya que el equipo de desarrollo, en este caso el autor como único miembro, pese a tener algún tipo de experiencia en las alternativas, dispone de más experiencia en el entorno ASP.NET MVC.

Por otro lado, una decisión no tan genérica como la tecnología a usar, ha sido decidir en qué parte del modelo cliente-servidor, impuesto como se ha dicho por la plataforma destino, se lleva a cabo la acción del juego. Por acción del juego, se entiende el cálculo de daños por ataque, gestión de turnos, gestión de movimientos disponibles por personaje, etc.

En este punto, existen dos posibilidades; que toda la acción del juego se lleve a cabo en el navegador del usuario y se transmita al servidor para que tenga persistencia, o que el navegador del usuario simplemente funcione como transmisor y ejecutor de las acciones que ocurren en el servidor.

La decisión tomada ha sido que toda la acción del juego ocurra en el servidor. De esta forma, el código ejecutado en el cliente se especializa en enviar peticiones http al servidor para transmitir los movimientos de los jugadores. A su vez, el código ejecutado en el servidor, controla que todas las peticiones por parte de los clientes sean correctas, evitando un uso fraudulento del juego, y posteriormente, calcula los resultados para cada movimiento y los persiste en la base de datos.

De esta forma, el desarrollo del proyecto es mas escalable y mantenible, ya que si el código del juego se lleva al cliente, el servidor al recibir los resultados, para poder validarlos debería volver a jugar la partida y comprobar que son correctos. Esta decisión añadiría más complejidad a la hora de mantener y escalar el juego.

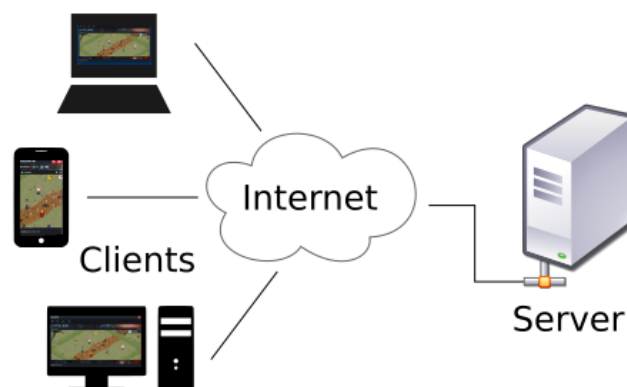


Ilustración 35: arquitectura del proyecto

8.1 Programación web

Este punto se divide en dos apartados con unas diferencias muy marcadas tanto a nivel de diseño y patrones de software utilizados como el de tecnologías o lenguajes de programación que se han hecho servir.

8.1.1 Front-end

El front-end del juego, principalmente ha sido desarrollado utilizando el lenguaje de programación javascript. Tanto el renderizador 2D Pixi js como el engine Trativo, están desarrollados en javascript por lo que toda la implementación para poder utilizar esta combinación en el proyecto ha sido realizada en el mismo lenguaje.

Trativo, para poder ser usado, expone una serie de funciones y eventos a los que el desarrollador de un juego puede subscribirse para gestionar las interacciones del usuario con el escenario.

De esta forma, alrededor de los eventos que expone Trativo ha sido desarrollado el front-end del sistema de combate del proyecto. Cuando el engine indica que un objeto ha sido seleccionado, se comprueba que tipo de objeto es y se actúa en consecuencia bajo el criterio del flujo de la partida.

```
1 var GameEvent = function ($) {
2
3     var priv = {
4         engine: null,
5         config: null,
6         skillId: null,
7         skillRange: null,
8         skillType: null,
9         castType: null,
10        animationOnTarget: null,
11        log: function (s)...,
16        initSceneCount: 0,
17        firstSceneMoves: function (...),
37        checkFirstSceneEnd: function(obj)...,
51        onSkillSelected: function (skillId, skillRange, skillType, castType, animationOnTarget)...,
58        setHighlightOnHeroNewFile: function(obj)...
62    }
63
64    return {
65        OnProcessFrameCallback: function (...),
70        OnObjectSelect: function (obj)...,
122        OnEngineInstanceReady: function (...),
152        OnFileSelect: function (r, c)...,
185        OnMouseOverObject: function (obj,event)...,
190        OnMouseOutObject: function (obj)...,
195        OnObjectReachedDestinationAfterMove: function (obj)...,
207        OnObjectUpdateCallback: function (obj)...,
214        Init: function (engine, config)...,
220        HeroMove: function(heroId)...,
230        HeroCancelMove: function(heroId)...,
240        HeroAttack: function(heroId)...,
249        HeroSkill: function (heroId, skillId, skillRange, skillType, castType, animationOnTarget)...
260    }
261 }(jQuery);
```

Ilustración 36: gestión de eventos en el escenario

En la ilustración 36, se puede observar la estructura de código javascript generada para reaccionar a los eventos del escenario. Esta parte del proyecto, y casi todas las demás dedicadas para gestionar el front-end en su totalidad, han sido desarrolladas utilizando el patrón de diseño de software javascript conocido como Module.

También, cabe destacar la utilización de la biblioteca javascript jQuery, que permite simplificar la manera de interactuar con todos los elementos del documento HTML en el navegador del usuario.

8.1.2 Back-end

El back-end del juego, principalmente ha sido desarrollado utilizando el lenguaje de programación orientado a objetos C#. Este proceso se ha llevado a cabo siguiendo el modelo arquitectónico Modelo-Vista-Controlador (MVC).

Esta arquitectura, separa una aplicación en tres componentes distintos, como son el modelo, la vista y el controlador. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento

Podemos considerar que el modelo de este proyecto es el dominio de datos del juego, el cual está formado por jugadores, personajes, habilidades, partidas, etc. En la siguiente ilustración se muestran todas las clases de dicho dominio.

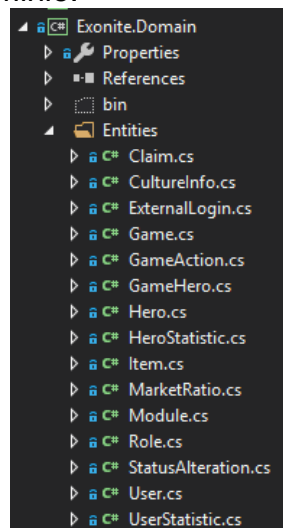


Ilustración 37: dominio de datos

Las vistas son los componentes que muestra la interfaz de usuario de la aplicación. Normalmente, esta interfaz de usuario se crea a partir de los datos del modelo, y como ejemplo de esto, tenemos las diferentes secciones de la web, que son generadas a partir de los datos que hay en el modelo

Finalmente, los controladores son los componentes que controlan la interacción del usuario, trabajan con el modelo y por último seleccionan una vista para representar la interfaz de usuario.

En una aplicación MVC, la vista solo muestra información; el controlador administra y responde a los datos proporcionados por el usuario y su interacción. De esta manera, los controladores dentro del proyecto, gestionan las acciones de los usuarios tanto en las diferentes secciones de la web como las peticiones http que genera el código del front-end del sistema de combate. En la ilustración 35 se muestran los diferentes controladores existentes en el proyecto.

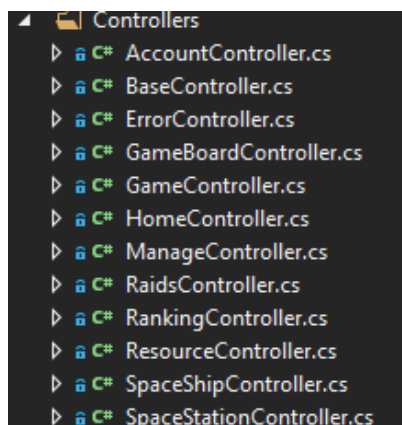


Ilustración 38: controladores

Cabe destacar, que para el desarrollo del back-end se ha seguido un diseño guiado por domino (DDD). Este diseño, es un enfoque para el desarrollo de software que provee una serie de prácticas y terminologías que facilitan las decisiones de diseño y aceleran el desarrollo de proyectos de aplicaciones con una lógica de dominio compleja, como es el caso de este proyecto.

Además, el proceso de análisis e implementación de todo el código se ha realizado siguiendo los principios SOLID, con el fin de crear un sistema que sea fácil de mantener y ampliar.

8.2 Persistencia y acceso a datos

En el apartado anterior, ya se ha visto el modelo de datos del proyecto, pero no se han explicado las tareas de persistencia y de acceso a los datos que este modelo produce.

Para tales fines, el proyecto cuenta con una capa de acceso a datos que proporciona un acceso simplificado a los datos almacenados en una base de datos relacional. Esta capa ha sido implementada siguiendo el patrón de software repositorio y usando un software de mapeo objeto-relacional (ORM).

8.3 Utilización de material gráfico

Para poder añadir en el escenario creado por la combinación Pixi js y Traveso el material gráfico creado para el proyecto, es necesario enviar al navegador del usuario un archivo json con todos los datos y objetos necesarios para renderizar el escenario completo.

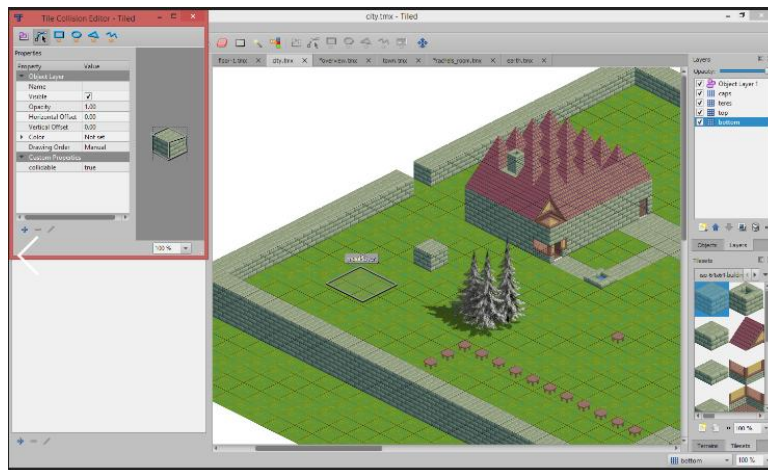


Ilustración 41: Tiled imagen de ejemplo

Tiled es un editor de mapas muy flexible de carácter general orientado a la creación de escenarios con tiles que soporta objetos con perspectiva isométrica.

8.4 Convirtiendo una aplicación web en una aplicación web en tiempo real

En las aplicaciones web tradicionales, el servidor siempre tiene un rol pasivo, respondiendo a las peticiones de los clientes. Este hecho, implica que ciertas funcionalidades requieran que los clientes deban preguntar al servidor cada cierto tiempo si deben recibir algún tipo de notificación o llamada. Este procedimiento dista de ser óptimo, ya que genera un tráfico totalmente innecesario.

En este punto es cuando tecnologías como Websockets entran en acción. Websockets es una tecnología que proporciona un canal de comunicación bidireccional sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor.

Conociendo la existencia de este tipo de tecnología, se ha añadido un sistema de notificaciones asíncrono al proyecto que no estaba incluido en la planificación inicial, ya que ni siquiera había sido tomado en consideración.

Utilizando la librería SignalR[14], creada para ser usada en entornos ASP.NET, se ha implementado un sistema de notificaciones que avisa a los jugadores cuando algún otro jugador se une a su partida o cuando es su turno en partidas ya iniciadas.

El sistema implementado permite al servidor del juego enviar información a los navegadores de los jugadores para ejecutar código javascript sin que el navegador del jugador tenga que realizar una petición previa.

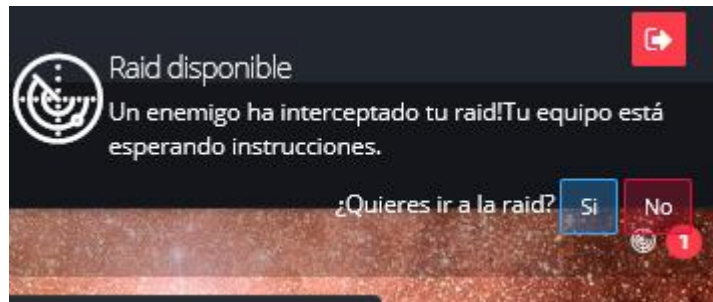


Ilustración 42: notificación

9. Conclusiones

La realización de este proyecto ha sido todo un reto personal para el autor, ya que el desarrollo de toda la plataforma web y el sistema de combate en el tiempo preestablecido fue una decisión muy ambiciosa.

Durante el desarrollo del mismo, el autor ha podido crecer como desarrollador y ha aprendido una serie de lecciones personales. Estas lecciones, van desde valorar mucho más el esfuerzo de otros desarrolladores sin tener tanto en cuenta el aspecto visual del resultado producido, a lecciones personales en el ámbito de la gestión de proyectos.

Aun y haberse completado la mayoría de los objetivos iniciales, cabe comentar que algunos objetivos fueron subestimados y que gracias al consejo de los consultores, el alcance del proyecto fue modificado. Los objetivos que fueron modificados, son los relativos a la cantidad de mapas y personajes disponibles en la entrega final.

Pese a esta modificación para quitar carga de trabajo al proyecto, también es necesario añadir, que se han conseguido objetivos que no fueron planificados. Estos objetivos son el modo de juego contra un bot y el sistema de notificaciones en tiempo real de la web.

Un videojuego, principalmente, es evaluado por su parte visual y artística y el público general no conoce el esfuerzo y dedicación que un proyecto de videojuego tiene detrás. Por este motivo, y después de la realización de este proyecto, el autor ha aprendido a valorar más el trabajo de los demás desarrolladores y a valorar, incluso, su propio esfuerzo cuando el resto de público general lo desconoce.

Por las razones mencionadas, la finalización del proyecto y la consecución de los objetivos de forma mayoritaria, son claramente un motivo de satisfacción.

10. Glossario

Algoritmo A*: es un algoritmo de búsqueda muy usado en los campos de la inteligencia artificial y videojuegos.

Assets: cada uno de los elementos que componen el juego(animaciones, sprites, imágenes, etc).

Back-end: hace referencia al lado del servidor.

Bot: término que describe a un usuario que no es humano que es dirigido por una inteligencia artificial de más o menos capacidad.

C#: lenguaje de programación de alto nivel orientado a objetos.

Cast: en el mundo de los videojuegos, hace referencia a cuando un personaje prepara un ataque o habilidad para lanzarlo a otro personaje o así mismo.

CRT: tecnología usada por los monitores de ordenadores personales antes de la aparición de pantallas planas.

Diseño adaptativo: es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visitarlas.

Engine: término que hace referencia a un software que provee de una serie de rutinas de programación que permiten el diseño, la creación y la representación de un videojuego.

Frame: hace referencia a una imagen individual.

Framework: un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.

Front-end: hace referencia a lo que el usuario ve en una aplicación web.

Gameplay: hace referencia a la jugabilidad de un juego.

HTTP: es un protocolo de transferencia de datos usado en internet.

HUD: hace referencia a la barra de estado que se muestra en pantalla durante una partida, en la que generalmente se encuentran botones e información.

Isometrico: es un método de representación gráfica que dota de aspecto 3D cuando realmente sólo hay 2D.

Javascript: lenguaje de programación interpretado usado en las paginas web para dotar a estas de dinamismo.

JcE: jugador contra entorno, designa una modalidad en la que uno o más jugadores se enfrentan contra la máquina.

JcJ: jugador contra jugador, designa una modalidad en la que dos o más jugadores se enfrentan entre ellos.

Json: acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos.

Kit de desarrollo: conjunto de herramientas de desarrollo de software.

MMO: usado para designar un género de videojuego en donde pueden participar, e interactuar en un mundo virtual, un gran número de jugadores del orden de cientos o miles simultáneamente conectados a través de la red.

MVP: siglas usadas para designar el jugador mas valioso de una partida.

ORM: técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia.

Raid: incursión, por su nombre en inglés.

Raiders: participantes en una incursión.

Renderizador: software encargado del proceso de generación de imagen.

Royalties: es el pago que se efectúa al titular de derechos de autor, patentes o marcas.

RPGMaker: serie de programas para el desarrollo de videojuegos de rol.
Slider: es un elemento de una página web que muestra múltiples imágenes y texto que se alternan entre ellas.

Socket TCP: designa un concepto abstracto por el cual dos programas pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

SOLID: acrónimo mnemónico que representa cinco principios básicos de la programación orientada a objetos y el diseño.

Sprite: hace referencia a una imagen generada por un mapa de bits.

Texturas: es una imagen de mapa de bits que se usa para cubrir la superficie de un objeto virtual, ya sea tridimensional o bidimensional, con un programa de gráficos especial.

Tile: casilla o azulejo en ingles, hace referencia a la separación de los mapas.

TTB: siglas en inglés para tactical turn-based.

Unity: es un motor de videojuego multiplataforma.

Vectorización: proceso que consiste en convertir imágenes que están formadas por píxeles en imágenes formadas por vectores.

WPF: es una tecnología de Microsoft que permite el desarrollo de aplicaciones con interfaz gráfica de usuario en Windows.

11. Bibliografía

- [1]: <https://unity3d.com/es> (22/02/2018)
- [2]: <http://www.rpgmakerweb.com/> (22/02/2018)
- [3]: <https://www.isogenicengine.com/> (24/02/2018)
- [4]: <https://phaser.io/> (24/02/2018)
- [5]: <http://www.pixijs.com> (24/02/2018)
- [6]: <http://www.travisajs.com/> (24/02/2018)
- [7]: <https://es.oqgame.gameforge.com/> (27/02/2018)
- [8]: <https://xcom.com/> (27/02/2018)
- [9]: <http://dlgames.square-enix.com/fft/en/> (27/02/2018)
- [10]: <http://footagecrate.com/> (17/04/2018)
- [11]: <https://ezgif.com/> (19/04/2018)
- [12]: <https://www.codeandweb.com/texturepacker> (08/03/2018)
- [13]: <https://www.mapeditor.org/> (15/03/2018)
- [14]: <https://www.asp.net/signalr> (28/05/2018)