

# Clash of Elements - Videojuego

**Sergio Artal de Benito**

Grado en Ingeniería Informática

TFG - Videojuegos

**Consultora: Ester Arroyo Garriguez**

**Profesor: Javier Luis Cánovas Izquierdo**

9 de junio de 2019



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

**Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)**

**A) Creative Commons:**



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](#)

**B) GNU Free Documentation License (GNU FDL)**

Copyright © 2019 Sergio Artal.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free

Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

### **C) Copyright**

© (Sergio Artal)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Clash of elements (board game)</i>
<b>Nombre del autor:</b>	<i>Sergio Artal de Benito</i>
<b>Nombre del consultor/a:</b>	<i>Ester Arroyo Garriguez</i>
<b>Nombre del PRA:</b>	<i>Javier Luis Cánovas Izquierdo</i>
<b>Fecha de entrega (mm/aaaa):</b>	06/2019
<b>Titulación:</b>	<i>Grado en Ingeniería Informática</i>
<b>Área del Trabajo Final:</b>	<i>TFG – Videojuegos</i>
<b>Idioma del trabajo:</b>	<i>Español</i>
<b>Palabras clave</b>	<i>Tablero, Turnos, Estrategia</i>
<p><b>Resumen del Trabajo (máximo 250 palabras):</b> <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>El objetivo del proyecto es la creación de un videojuego basado en el juego de mesa ELEMENTOS, un juego moderno que no tiene actualmente adaptación digital. La elección de este juego se fundamenta en dos pilares: mi gusto personal por este tipo de juegos y las limitaciones de tiempo del proyecto que condicionan la elección de proyectos más ambiciosos.</p> <p>El proyecto incluye todas las fases del desarrollo, iniciando por el estudio del juego real para su adaptación, hasta la obtención de un producto final jugable. El motor elegido para el desarrollo del juego es Unity, debido a su sencillez, amplia documentación y comunidad, y las capacidades tanto técnicas como multiplataforma que ofrece.</p> <p>Utilizando el documento de diseño del juego creado al inicio del proyecto, se han ido completando las fases del proyecto, logrando los hitos marcados en tiempo. Cabe destacar que, al ser mi primer proyecto con Unity, a los primeros hitos de desarrollo he llegado muy ajustado y con mucho trabajo por delante.</p> <p>La entrega final comprende la aplicación para Android plenamente funcional para dos jugadores utilizando el mismo terminal. La planificación inicial del GDD y el estudio de las reglas del juego para su adaptación han facilitado el trabajo, y las mecánicas diseñadas inicialmente han funcionado correctamente.</p> <p>El proyecto se podría completar incluyendo la posibilidad de jugar contra la máquina o en línea contra otros jugadores. El arte sería otro punto por mejorar.</p>	

**Abstract (in English, 250 words or less):**

The project goal is to create a video game based on ELEMENTOS board game, a modern game that does not currently have digital version. This game was chosen due to two main reasons: I love this kind of games, and the project time constraints, that avoid choosing more ambitious projects.

The project includes all phases of development, starting with the study of the real game for its digital adaptation, until obtaining a playable final product. The chosen engine for the game development is Unity, due to its simplicity, extensive documentation and community, and the technical and multiplatform capabilities it offers.

Using the game design document created at the beginning of the project, the project phases have been completed, achieving the milestones in time. It should be noted that, being my first project with Unity, it was difficult to meet starting milestones goals, leaving a lot of work ahead.

Final delivery includes a fully functional Android application for two players on the same terminal. The GDD initial planning and the game rules analysis have facilitated the work for its digital adaptation, and the mechanics initially designed have worked properly.

The project could be completed later including the possibility of playing against the machine or online against other players. Art could be reviewed to improve it.

# Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	2
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. Resto de capítulos.....	5
2.1 Adaptación del juego original.....	5
2.2 Plataforma de desarrollo.....	5
2.3 Menús e interfaz de usuario.....	6
2.4 Componentes del juego.....	6
2.5 Programación del juego.....	9
2.5.1 Scripts.....	9
2.5.2 Control de menús.....	9
2.5.3 Preparación de fichas.....	10
2.5.4 Objetos de la partida.....	11
2.6 Proceso de una partida.....	14
3. Conclusiones.....	16
4. Glosario.....	18
5. Bibliografía.....	20
6. Anexos.....	21

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

Actualmente se dan una serie de circunstancias. La industria de los videojuegos es más importante que la del cine o la música. Ese auge ha venido apoyado por una gran cantidad de herramientas para el desarrollo de videojuegos accesibles a cualquiera con una conexión a internet. El desarrollo de los móviles está permitiendo que todo el mundo tenga en sus manos a diario un dispositivo con el que poder jugar videojuegos. Y por supuesto contamos con internet, con toda la información que podamos necesitar sobre cómo desarrollar videojuegos.

Recuerdo cómo allá por 1988 comencé a leerme el manual de Basic de mi Spectrum y aprendí a realizar mis primeros programas, simplemente mostrar una pregunta por pantalla, almacenar la respuesta del usuario, y que el programa reaccionase en base a esa respuesta me parecía algo increíble. Con el tiempo, un par de amigos y sin más guía que aquel manual y el código de algunos ejemplos que publicaban en la revista MicroHobby ([portada](#)), fuimos haciendo algún juego de plataformas, e incluso llegamos a programar un editor gráfico para diseñar los personajes de nuestros juegos. Durante varios años aparqué aquel Spectrum, hasta que decidí que quería dedicarme a la informática y volví a acercarme a la programación. Sin embargo, en el mundo laboral me decanté por la arquitectura de sistemas, y aquellos pinitos de desarrollo de videojuegos se quedaron en la infancia.

Cuando hace unos años decidí cursar el grado de Ingeniería Informática el objetivo era aprender y disfrutar de otras áreas de la informática que apenas conocía, no solo obtener un título que me capacitase para una profesión. De hecho, he escogido el itinerario que más se aleja de mi día a día profesional. Al seleccionar mi proyecto final de grado, mi primera opción era escoger un tema en el que mi desarrollo profesional me sirviera de ayuda, sin embargo, me recordé el motivo por el que comencé en este mundo de la informática, y decidí realizar un videojuego.

Una vez tomada la decisión tocaba pensar en el tipo de juego que quería desarrollar. De antemano ya sabía que el apartado gráfico y sonoro iban a ser un problema, porque no son precisamente mi fuerte, y además está la cuestión temporal, es un proyecto de pocos meses compaginado con la familia y el trabajo. Pensé que un juego de mesa al que pudiera jugar con mis hijas pequeñas sería un proyecto ideal. Tras buscar durante unos días encontré [ELEMENTOS](#) y al comprobar que no tenía versión digital decidí que sería el videojuego que realizaría.



## 1.2 Objetivos del Trabajo

El objetivo principal es la creación de un videojuego totalmente operativo desde cero, pasando por todas las etapas desde la idea inicial hasta el programa terminado.

Para lograr este objetivo ha sido necesario cumplir con ciertos hitos:

- Diseñar el documento de diseño del juego (GDD)
- Aprender a desarrollar juegos de mesa con Unity
- Adaptar la jugabilidad del juego real al videojuego digital

## 1.3 Enfoque y método seguido

Teniendo en cuenta el tiempo para realizar el proyecto, mi poca experiencia en desarrollo, y que los apartados gráficos y sonoros podían ser un problema, opté por desarrollar un videojuego de estrategia por turnos del estilo de un juego de mesa. Dado que las mecánicas de este tipo de juegos deben ser muy cuidadas para que el juego sea interesante decidí apostar por un juego existente y hacer su versión digital.

Estas decisiones me han facilitado mucho el apartado estético y mi principal interés ha sido la adaptación del original manteniendo la jugabilidad. Para ello he aprovechado mi experiencia como jugador de [ajedrez](#) en el móvil, que me ha ayudado a la hora de diseñar la experiencia de usuario.

## 1.4 Planificación del Trabajo

La realización del proyecto ha requerido de los siguientes recursos:

- **Ordenador.** Para la programación del videojuego se requiere de un ordenador, en este caso se ha utilizado un equipo sobremesa con Windows 10.
- **Conexión a Internet.** Tanto para el acceso a recursos como imágenes o sonido, como para el acceso a manuales, guías y artículos, el uso de Internet ha sido continuo a lo largo de todo el proyecto.
- **Unity3d.** Motor de desarrollo de videojuegos, que, junto con Visual Studio, ha permitido la creación de los objetos y programas del juego.
- **Office 365.** Se ha utilizado para la edición de la documentación entregada a lo largo del proyecto.
- **Mischief.** Para la edición de algunos gráficos entregados en la documentación se ha utilizado esta herramienta.
- **Gantt Project.** Herramienta utilizada para la generación del Gantt del proyecto.

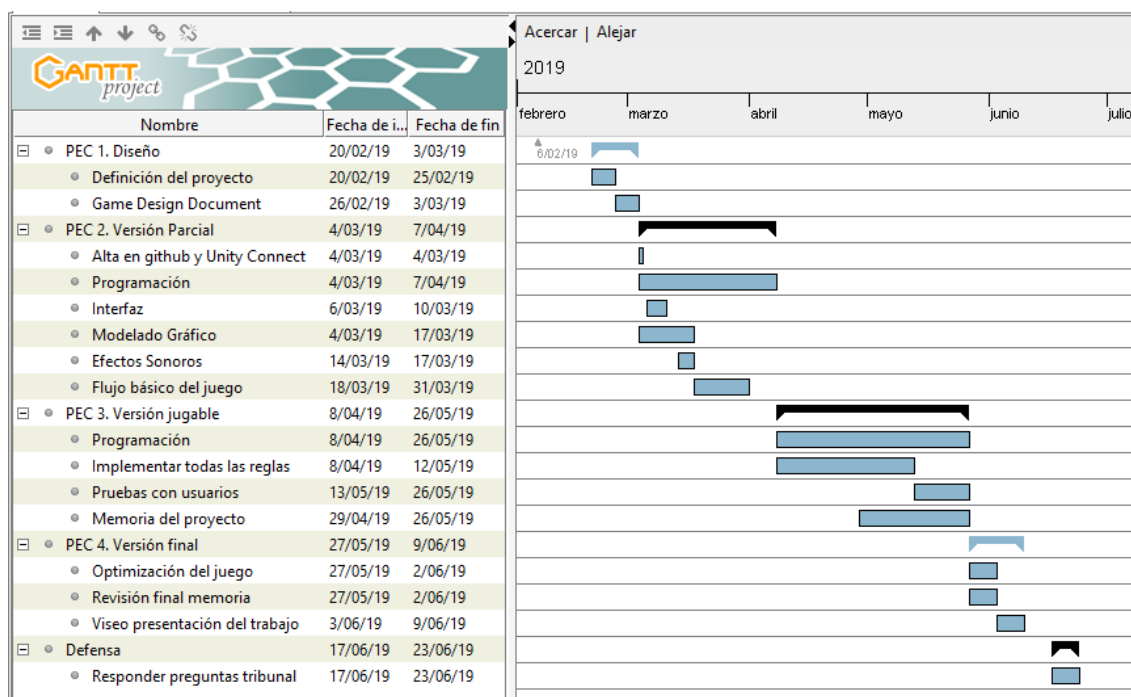
La planificación de entrega de las prácticas ha sido esta:

Fases	INICIO	FIN	DIAS
PEC 1. GDD	20/02/2019	03/03/2019	11
PEC 2. Versión Parcial	04/03/2019	07/04/2019	34
PEC 3. Versión jugable	08/04/2019	26/05/2019	48
PEC 4. Versión final	27/05/2019	09/06/2019	13
Defensa	17/06/2019	23/06/2019	6

Y la división de tareas se refleja en el siguiente cuadro:

	Tarea	INICIO	FIN	DIAS
Diseño	Definición del proyecto	20/02/2019	25/02/2019	6
	Game Design Document	26/02/2019	03/03/2019	6
Versión Parcial	Alta en github y Unity Connect	04/03/2019	04/03/2019	1
	Programación	04/03/2019	07/04/2019	35
	Interfaz	06/03/2019	10/03/2019	5
	Modelado gráfico	04/03/2019	17/03/2019	14
	Efectos sonoros	14/03/2019	17/03/2019	4
Versión jugable	Flujo básico del juego	18/03/2019	31/03/2019	14
	Programación	08/04/2019	26/05/2019	49
	Implementar todas las reglas	08/04/2019	12/05/2019	36
	Pruebas con usuarios	13/05/2019	26/05/2019	14
Versión final	Memoria del proyecto	29/04/2019	26/05/2019	28
	Optimización del juego	27/05/2019	02/06/2019	7
	Revisión final memoria	27/05/2019	02/06/2019	7
Defensa	Video presentación del trabajo	03/06/2019	09/06/2019	7
	Responder las preguntas del tribunal	17/06/2019	23/06/2019	7

De las tareas obtenemos el siguiente diagrama:



## 1.5 Breve resumen de productos obtenidos

Estos son los entregables que se han ido obteniendo y entregando a lo largo del proyecto:

- Game Design Document. Documento de diseño del juego.
- Gráfica de Gantt con la planificación del proyecto.
- Documento de diseño de pantallas con la navegación del juego.
- Documento de versión parcial sobre la primera entrega del juego.
- [Video](#) explicativo del estado del juego en su versión parcial.
- Documento de la versión final del juego.
- [Video](#) explicativo de la jugabilidad de la versión final.
- Proyecto en [Git hub](#) con todo el código fuente.
- [ShowCase](#) de Unity.
- Fichero APK con el ejecutable del juego para Android.
- Memoria final del proyecto.
- Documento de autoevaluación.
- [Video](#) de defensa del proyecto.

## 1.6 Breve descripción de los otros capítulos de la memoria

Los detalles del desarrollo de todos los elementos del juego se detallarán a continuación en el resto de los capítulos. Los puntos que se tratan son:

- Adaptación del juego original
- Plataforma de desarrollo
- Componentes del juego
- Menús e interfaz de usuario
- Programación del juego
- Proceso de una partida

## 2. Resto de capítulos

### 2.1 Adaptación del juego original

Al realizar el proyecto sobre un juego ya existente uno de los puntos críticos ha sido la adaptación de las mecánicas del juego a las capacidades disponibles con un dispositivo móvil.

Para preparar la partida cada jugador debe seleccionar los elementos de sus piezas, ya que cualquier combinación es posible. Se ha preparado un sistema de pantallas en las que cada jugador puede realizar su selección de forma “privada” antes de comenzar la partida. Al comenzar la partida todas las fichas quedan a la vista sobre la cara principal.

Los jugadores deben poder girar sus piezas, para realizar este movimiento no se me ocurría ningún movimiento sobre la pantalla táctil que fuera intuitivo, por lo que finalmente el giro se realiza mediante un botón en la pantalla.

El resto de los movimientos no suponía ningún reto de diseño, si nos fijamos en cualquier juego de ajedrez o similar tenemos una mecánica muy intuitiva. En este punto lo más complejo ha sido calcular en cada momento qué casillas marcar como movimientos válidos o posibles fichas para tomar.

Con el desarrollo del juego ya avanzado, nos encontramos con la dificultad de mover la bandera, era demasiado pequeña para señalarla con facilidad, y poner un botón para mover la bandera no me gustaba. Finalmente, jugando con el tamaño de la bandera creo que el compromiso es bueno, podemos señalar con facilidad la ficha o su bandera independientemente, aunque es cierto que conforme la ficha avanza, alejándose de nuestro punto de vista, se complica un poco.

Por último, implementar la condición de victoria ha resultado bastante sencillo. No hay ningún análisis complejo que realizar, si tras un movimiento la bandera está en la última fila hemos ganado.

### 2.2 Plataforma de desarrollo

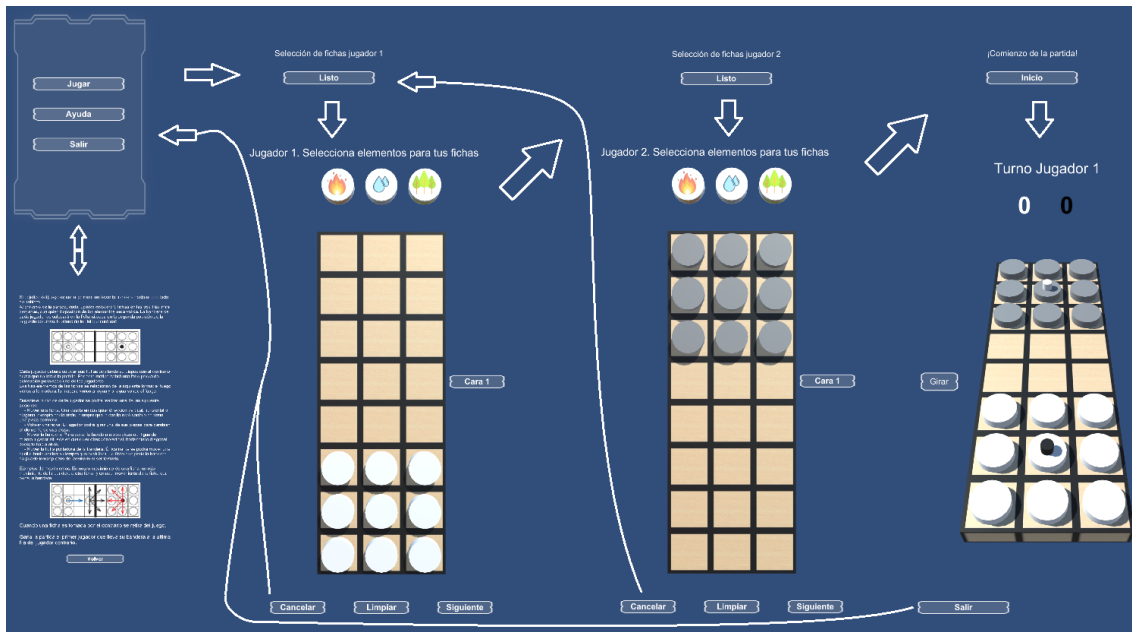
Actualmente hay muchos motores de videojuegos con los que se podría realizar “Clash of Elements” con solvencia. No es un juego especialmente complejo, no hay muchos elementos en pantalla al mismo tiempo, no hay animaciones y no debemos responder en tiempo real.

El motivo de utilizar Unity es que lleva muchos años en el mercado y hay muchísima documentación y ejemplos, lo que ha facilitado enormemente el desarrollo, especialmente al principio.

## 2.3 Menús e interfaz de usuario

La navegación de los menús es el apartado que primero se dejó terminado cuando se comenzó con la fase final de programación del juego. Es muy importante tener claro el flujo de menús y la funcionalidad de cada elemento.

Tras las pruebas iniciales con los botones estándar disponibles, se utilizó un ejemplo del Asset Store llamado [Unity Samles: UI](#) para mejorar la estética de los menús. El flujo de los menús es el siguiente:



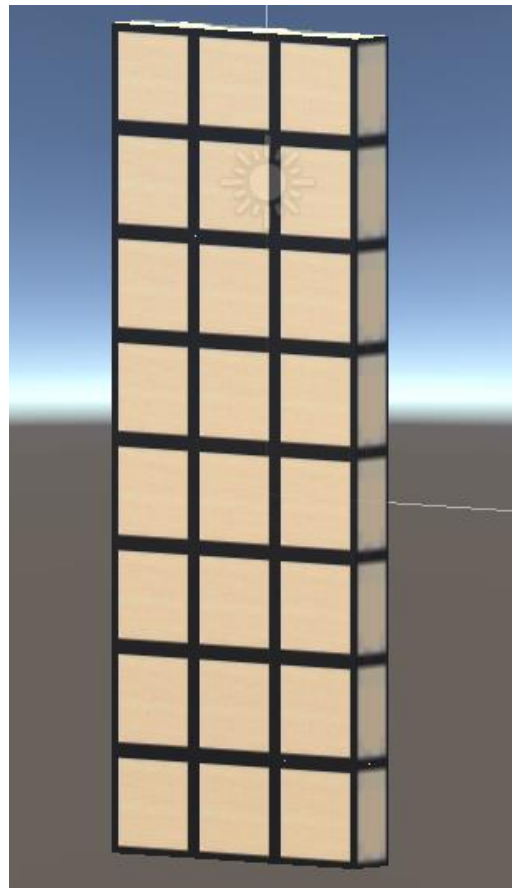
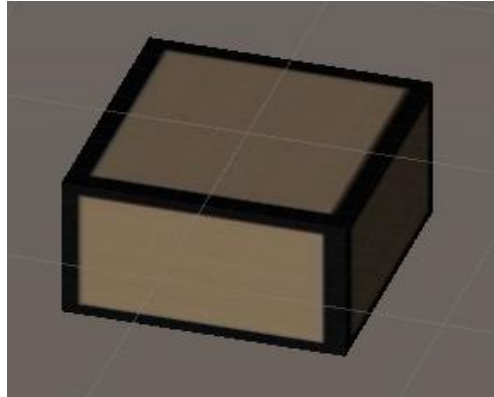
## 2.4 Componentes del juego

Los distintos elementos gráficos del juego se han creado con objetos 3d utilizando las herramientas que ofrece Unity, a los que se han agregado imágenes como texturas, colores y sonidos. Con la unión de los distintos elementos se han creado los Prefabs necesarios para construir el juego.

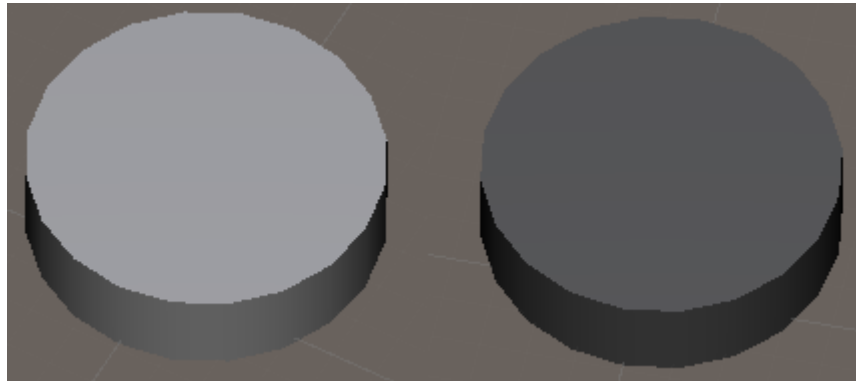
- **Fichas para selección de elementos.** En las pantallas de preparación, cada jugador debe elegir los elementos (fuego, agua o madera) que asignará a sus fichas. Las tres fichas que permiten la selección son tres Prefabs: Jugador – fuego, Jugador – Agua y Jugador – Madera.



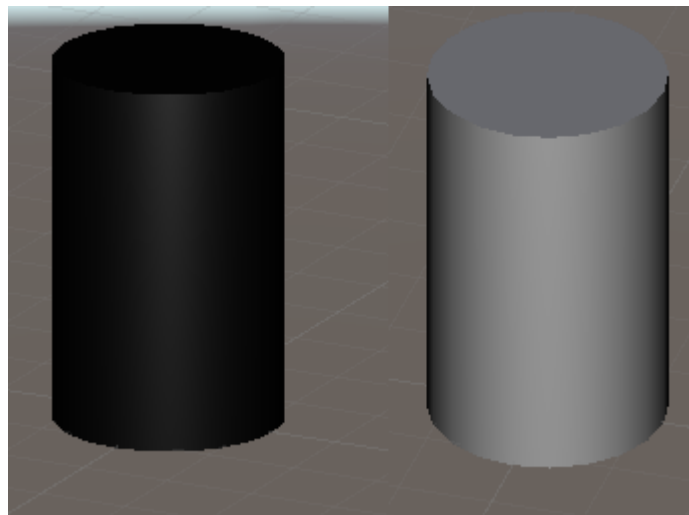
- **Casillas del tablero.** Son cubos creados con Unity a los que se les ha añadido un material que simula la madera con los bordes pintados de negro. Cada cubo se encuentra dentro del objeto Casilla, y juntos forman el tablero. Hay dos tipos de Prefab, Casilla y Casilla 2d, el motivo es que en la selección de fichas las casillas tienen menos funcionalidad.



- **Fichas de jugadores.** En este caso se utilizaron cilindros que se achataron hasta tomar la forma de fichas de madera. También distinguimos en los Prefabs las fichas para la partida y las que se muestran en 2d para la selección de elementos, ya que tienen funcionalidad muy diferente en los scripts. Las fichas están sin textura, ya que dependerá de la decisión del jugador.



- **Bandera.** Son, al igual que las fichas, cilindros creados con Unity. Para las banderas nos interesaba una figura cilíndrica alta y de cierto grosor para que fuera posible seleccionarlo. Para poder ver la bandera sobre las fichas de cada jugador, se ha buscado el máximo contraste, de forma que las fichas blancas tienen una bandera negra y viceversa.



Todos los objetos descritos tienen añadidos los componentes **Rigidbody**, que nos permite realizar acciones sobre ellos al seleccionarlos, y **Audiosource**, a través del que emitimos los sonidos deseados en cada momento del juego.

Además, estos objetos tienen scripts asociados en los que está gran parte de la lógica del juego. En el siguiente apartado veremos los detalles de la programación.

## 2.5 Programación del juego

### 2.5.1 Scripts

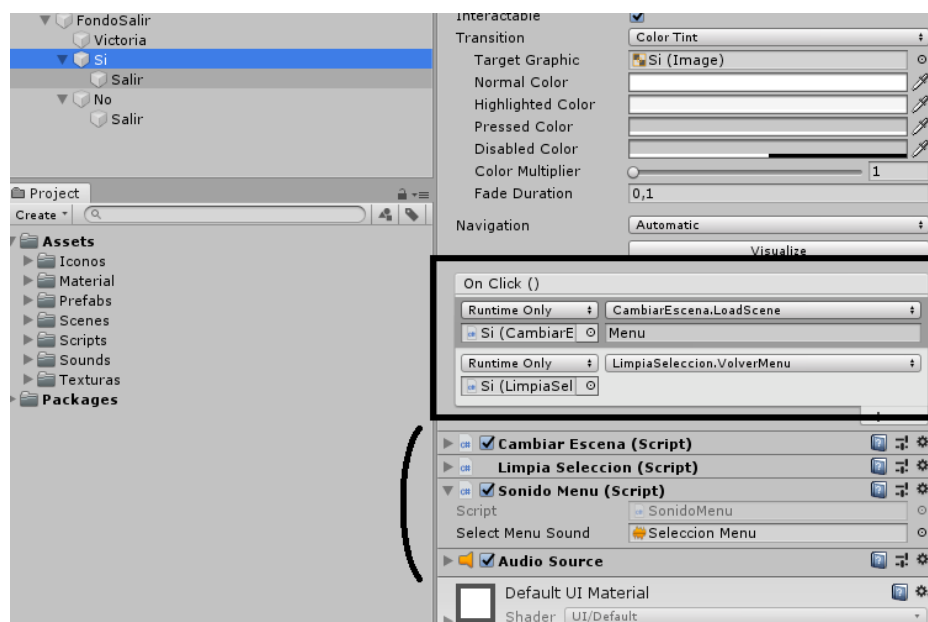
Para que todos los elementos descritos en los apartados anteriores se comporten como se espera en el juego, necesitamos añadir scripts donde programaremos los comportamientos deseados. En nuestro proyecto todos los scripts están asociados a un objeto del juego, aunque en algunos casos sea un objeto pasivo utilizado únicamente para ejecutar código al cargar una nueva escena. Solo hay una excepción, el script “CambiaObjeto” lo utilizamos a modo de librería, con una serie de funciones a las que podemos llamar pasando un objeto para que se haga alguna transformación. Lo utilizamos para cambiar el elemento de una pieza o el color de una casilla, por ejemplo, algunas de sus funciones son:

- CambiaObjeto.PintaRojo(Casilla)
- CambiaObjeto.AsignaMaderaNegra(Ficha)
- CambiaObjeto.QuitaMaterial(Ficha)

### 2.5.2 Control de menús

El control de flujo de los menús se hace con los scripts “CambiarEscena”, “Siguiente” y “Salir”, que se asocian a los distintos botones y realizan la tarea correspondiente. Por ejemplo, en la siguiente captura se observa que el botón “Sí” del menú de salida de la partida tiene asociados tres scripts:

- “Cambiar Escena” se utiliza para volver a la escena “Menú”
- “Limpia Selección” para inicializar todos los datos del juego
- “Sonido Menú” emite un sonido al pulsar un botón



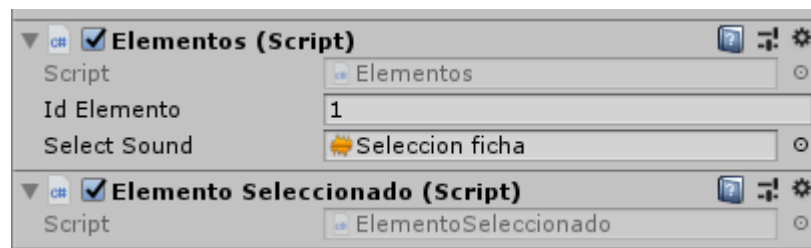


### 2.5.3 Preparación de fichas

En las diferentes escenas, cada elemento se encarga de reaccionar a las acciones del jugador mediante los scripts. Además, en estos scripts se añade información adicional sobre cada elemento. Vamos a detallar los scripts de cada elemento comenzando por las escenas de selección de fichas y terminando por la escena de la partida.

Las dos escenas de selección de fichas son muy similares, únicamente cambia el jugador que selecciona y las fichas que se eligen (incluido el color). Los objetos del juego son los siguientes:

- **Casillas.** En las fases de selección de fichas las casillas no son seleccionables y no tienen ningún tipo de comportamiento, por lo tanto, no tienen ningún script asociado.
- **Fichas de elementos.** Las fichas de los elementos que podemos seleccionar tienen asociados dos scripts cada uno:



El primero de ellos, “Elementos”, lo utilizamos para saber que elemento representa y el sonido que tiene asignado. El fuego es el 1, el agua es el 2 y el fuego el 3.

En el script “Elemento Seleccionado” controlaremos que solo haya una ficha de elemento seleccionada en un momento dado, pudiendo desmarcar la que ya estuviera seleccionada. En el caso de que se hubiera seleccionado alguna ficha del jugador previamente, al pulsar una ficha de elemento se asignaría el elemento a las fichas del jugador marcadas.

- **Fichas de Jugador.** Se han asignado también dos scripts a las fichas del jugador, en los que se añade información y lógica de comportamiento:



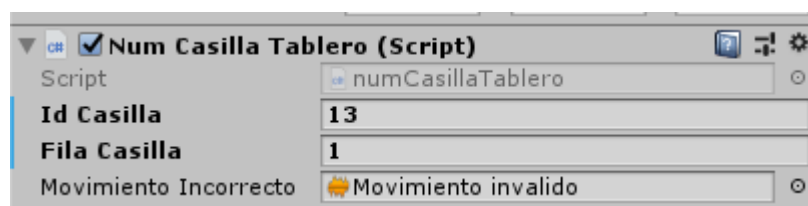
El script “Num Ficha Jugador” nos identifica cada ficha con un número del 1 al 9, tiene asignado el sonido que emitirá la ficha al ser seleccionada y almacena la información de todas las posibles texturas que puede llegar a tener una ficha. Estas texturas son las que se aplicarán con “CambiaObjeto”.

Con el script “Ficha Seleccionada” añadimos el comportamiento al seleccionar una de estas fichas de jugador. Al seleccionar una ficha, si alguna de las fichas de elementos está seleccionada, a la cara expuesta de esa ficha se le asignará el elemento. Si no hay elementos seleccionados, podremos ir marcando o desmarcando fichas, que tenga ya elementos o no, para poder cambiarlas en bloque al seleccionar un elemento.

#### 2.5.4 Objetos de la partida

Los objetos de la partida tienen mucha más lógica asociada, ya que podemos interactuar potencialmente con todas las casillas, fichas y banderas del tablero. Para facilitar al usuario sus movimientos, el juego le propone los movimientos disponibles, incluyendo la posibilidad de tomar una pieza del contrario. Los scripts que tienen asociados los distintos objetos en la escena de juego son los siguientes:

- **Casillas del tablero.** En la partida las casillas del tablero tienen un script asociado que nos aporta información sobre cada casilla y además contiene lógica cuando una casilla es pulsada.



En el script almacenamos el sonido en caso de movimiento incorrecto, información del número de la casilla y la fila en la que se encuentra, información útil a la hora de calcular qué casillas se deben iluminar. También almacenamos información sobre si la casilla está iluminada, ocupada y qué jugador tiene una ficha en esa casilla. Esta información ayuda al calcular los posibles movimientos de otras fichas.

En este script controlamos cuando se pulsa en una casilla, si no está iluminada el movimiento no es correcto. En caso contrario moveremos la ficha, y la bandera si la lleva. La ficha seleccionada se mueve a la posición de la casilla (y la bandera también), además se actualiza la información de las casillas origen y destino, cambiando su ocupación y quien las ocupa. Finalmente deseccionamos todas las fichas y quitamos la iluminación de todas las casillas. Si la ficha que se ha movido lleva la bandera y ha llegado a la ultima fila se activa el panel de victoria, sino se pasa el turno.

- **Banderas.** Durante la partida, un jugador puede mover la bandera durante su turno a otra ficha adyacente siempre que sea en horizontal o en cualquier sentido hacia adelante. Para calcular ese posible movimiento se utiliza el script “Mover Bandera” asociado a las banderas de ambos jugadores.

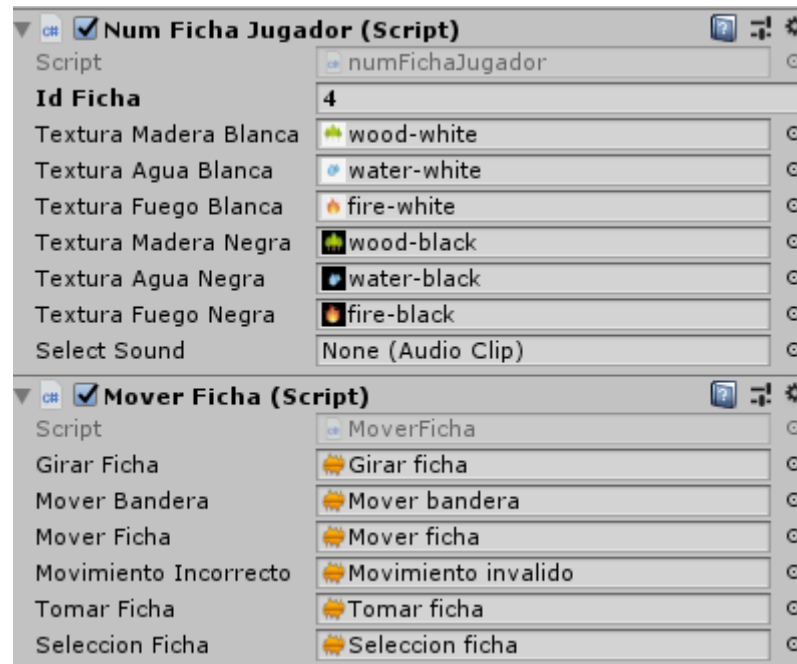


En este script almacenamos información sobre si la bandera está seleccionada o no y en que casilla se encuentra la bandera. Además, tiene asociado el audio de selección de la bandera.

Cuando un jugador selecciona su bandera, la bandera se ilumina y se calculan las fichas a las que se puede mover, iluminándolas para que el jugador pueda indicar a cuál de ellas quiere moverla. Antes de marcar las fichas destino, quitamos la iluminación de todas las casillas por si antes se hubiera seleccionado alguna ficha, y se quita la selección de todas las fichas. También desactivamos el botón de giro, ya que la bandera no puede girar.

Al calcular qué fichas se deben iluminar como destinos posibles, lo que hacemos es recorrer las fichas del jugador, y comparando la posición de cada ficha con la posición de la bandera decidimos si puede ser receptor de la bandera. Para cada jugador hacemos una comparación diferente, porque el cálculo lo hacemos en base al número de la casilla.

- **Fichas de los jugadores.** Durante la partida las fichas de los jugadores tienen mucha importancia en los procesos del juego, dependiendo de la ficha seleccionada hay muchas opciones a evaluar. Las fichas tienen dos scripts asociados:



El script “Num Ficha Jugador” nos permite almacenar información del número de ficha y las posibles texturas que puede llegar a tener.

El script “Mover Ficha” es con el que controlamos los efectos de la selección del jugador sobre sus fichas. De cada ficha almacenamos la cara que está activa (visible), el elemento que tiene en cada cara, si está o no seleccionada, si tiene la bandera, y en la casilla en la que se ubica. Además, tiene asociados los efectos sonoros para el giro de ficha, el movimiento de la bandera, el movimiento de la ficha, si intentamos un movimiento incorrecto, al tomar una ficha contraria y la selección de ficha.

Al seleccionar una ficha, podemos estar realizando tres acciones: seleccionarla para moverla, querer mover a esa ficha nuestra bandera, o queremos tomar la ficha. Para distinguir cada una de las distintas opciones comprobamos si la ficha sobre la que hemos pulsado es nuestra, si no lo es queremos tomarla. Si es nuestra y tenemos seleccionada la bandera, queremos mover la bandera. Y finalmente, si no ocurre ninguna de las anteriores, queremos seleccionar la ficha para moverla.

Mover la bandera implica mover gráficamente el objeto “Bandera” de nuestro jugador a la posición de la ficha seleccionada. Para que sea posible tenemos que tener señalada la bandera y nuestra ficha debe estar iluminada. Cambiaremos la información de las fichas para indicar cual de ellas tiene ahora la bandera, y desmarcaremos fichas,

bandera y casillas. Si la bandera se ha movido a la última fila, habremos ganado, en caso contrario, se pasa el turno.

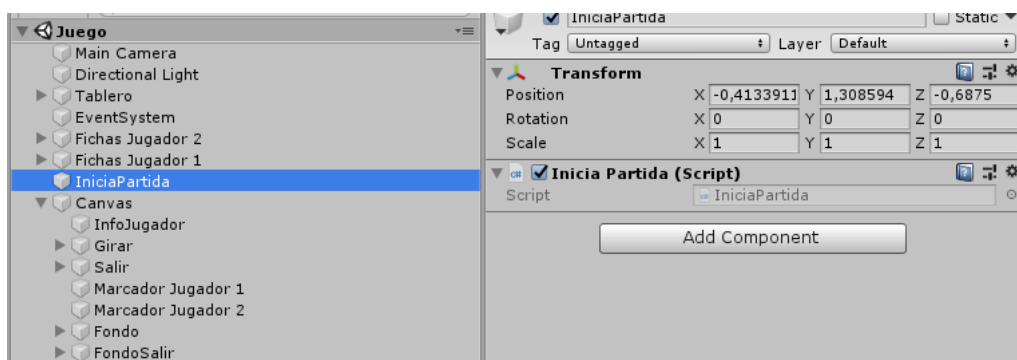
Si marcamos una ficha, debemos quitar la selección de cualquier otra de nuestras fichas, de la bandera y quitar la iluminación de las casillas. Marcaremos la ficha como seleccionada y dependiendo de si tiene o no la bandera calcularemos las posibles casillas de destino. De forma similar a la iluminación de fichas para el cambio de bandera, dependiendo de quien es el jugador, comprobaremos las casillas libres a las que la ficha se puede mover. Si una casilla destino está ocupada por una ficha del contrario, se comprobará si nuestro elemento puede con el del contrario para tomar la ficha, y se iluminará la casilla de la ficha enemiga en rojo en caso de que podamos tomarla.

Si marcamos una ficha contraria solo podremos tomarla si la casilla sobre la que se encuentra está marcada en rojo. Sino sonará que es un movimiento incorrecto. Al comer nuestra ficha ocupará la posición de la ficha contraria, y destruiremos la ficha tomada. Actualizaremos la información de la ficha y de las casillas origen y destino con la nueva información. Se cambia el turno, y se quitan todas las selecciones de fichas y casillas.

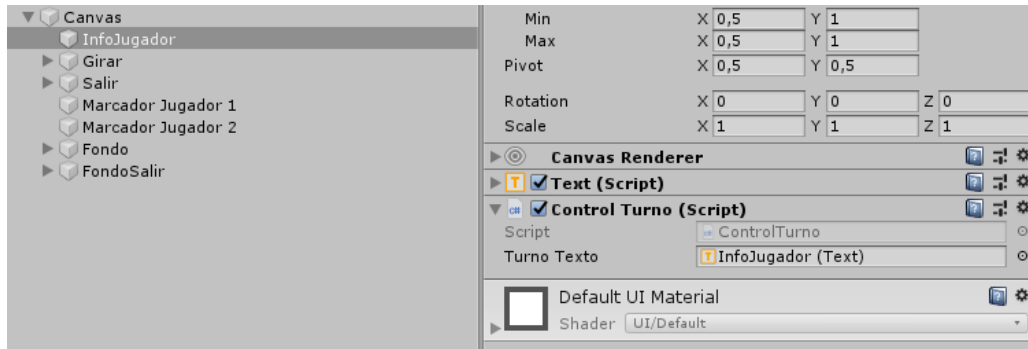
Al seleccionar una de nuestras fichas, se activa el botón que permite girarla. Este botón tiene asociado el script que realiza el giro de la ficha, cambiando la cara activa, actualizando el material que se muestra, deseleccionando la ficha y quitando la iluminación de las casillas.

## 2.6 Proceso de una partida

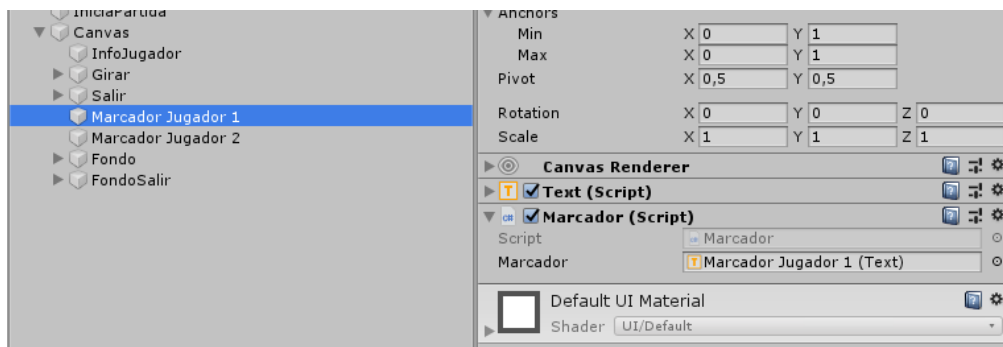
Cuando ambos jugadores terminan la asignación de elementos a ambas caras de sus fichas, toda la información se encuentra almacenada en un array. Al iniciar el juego, se ejecuta el script "IniciaPartida", que prepara los paneles de victoria y salida el juego, y con la información del array almacena la información de elementos en las fichas y les asigna los materiales correspondientes. Así mismo, se preparan las casillas indicando cuales están ocupadas y por qué jugador. Finalmente se inicializa el juego asignado el turno al jugador de las fichas blancas.



El panel informativo que indica de qué jugador es el turno, tiene asignado el script "Control Turno". Este script mantiene la información del panel y realiza el proceso de cambio de turno, realizando una mínima pausa para que veamos el efecto de nuestro movimiento, y cambiando la cámara de posición para que el otro jugador tenga la misma perspectiva de la partida.



Con el paso de los turnos, los jugadores irán tomando piezas, para mantener el marcador actualizado, en ambos elementos de texto hemos asociado el script "Marcador", que nos permitirá incrementar el valor al tomar una pieza contraria.



### 3. Conclusiones

Hace ya un par de semanas que tengo el juego terminado, y cada vez que lo enseño a alguien inicialmente da la sensación de que no esperan mucho, las pantallas en 2d donde seleccionas los elementos no parecen atraer su atención, sin embargo, cuando aparece la pantalla del juego en 3d siempre se sorprenden positivamente, mejora notablemente su percepción, y de repente prestan más atención.

Este es mi primer proyecto con Unity, no soy bueno en cuestiones artísticas, y solo he realizado programación orientada a objetos cursando el grado en la UOC. Tras el esfuerzo de aprendizaje y las horas invertidas en conseguir los comportamientos deseado, intentar que estéticamente sea mínimamente correcto, y que los efectos sonoros encajen... que a todo el mundo le empiece a gustar al verlo en 3d, cuando es exactamente lo mismo con la cámara girada, ha sido toda una lección sobre qué nos atrae inicialmente de un juego, aunque luego haya que mantener su atención con mecánicas entretenidas y que reten las habilidades del jugador. Posiblemente cuando me enfrento a un nuevo videojuego yo haga lo mismo, pero al desarrollarlo yo, había perdido esta perspectiva.

A nivel técnico, aunque mi código todavía tiene mucho margen de mejora, he aprendido mucho de cómo programar con Unity y hacer uso de los scripts en los objetos, que es algo a lo que inicialmente me costó adaptarme. También he aprendido que por muy claro que tengas lo que quieres hacer, muchas veces hasta que no lo ves en la práctica no te das cuenta de que no funciona (estéticamente o por jugabilidad), o que te has dejado algo por tener en cuenta y debes volver al código o al arte para retocarlo.

Profesionalmente llevo años desarrollando y gestionando proyectos de tecnología en empresas, pero nunca me había enfrentado al desarrollo de un videojuego. Las fases de planificación y documentación no me han resultado especialmente ajenas, únicamente hay que adaptarse a las fases requeridas en este tipo de proyectos y medir los tiempos correctamente. Creo que en este sentido he ido cumpliendo con la planificación, aunque en la PEC2 llegué con el desarrollo muy poco avanzado. Todo el tiempo invertido en aprender cómo trabajar con Unity en la PEC2 permitió que para la siguiente entrega diera tiempo a corregir y limpiar algo el código, rediseñar toda la estética, e implementar la parte más compleja que era la partida en sí misma. En la PEC3 me sentí mucho más cómodo con Unity y pude disfrutar de reto que suponía desarrollar cada proceso del juego.

En cuanto al resultado final del proyecto estoy contento, tanto por haber conseguido trasladar no solo las reglas, sino también la jugabilidad del juego de mesa en el que me he basado; como por el cambio de aspecto general del juego, especialmente tras la entrega de la PEC2. Con los nuevos menús, paneles UI informativos, paneles flotantes y sonido, considero que la estética es practica y cumple su función.

Por otro lado, en el capítulo de líneas de mejora, lo primero a revisar es obviamente la estética, especialmente después de las reacciones tras enseñar el juego. Habría que rediseñar los fondos, los textos, añadir algún efecto (por ejemplo, al girar), y buscar ayuda para retocar la estética del tablero y elementos de las fichas.

Sin embargo, considero que el principal trabajo a futuro debería ser añadir la posibilidad de jugar contra la máquina (añadir IA) y partidas online. Ampliaríamos mucho las posibilidades del juego, aumentando el atractivo para más perfiles de jugadores y permitiendo otra temporalidad en el uso del juego. Posteriormente se podrían añadir clasificaciones y puntuaciones a los jugadores. Sería interesante investigar otras opciones para el giro de las fichas, como poder ver la otra cara antes de decidir si giramos o no, o la posibilidad de que la elección inicial de los elementos fuera automática al azar.



## 4. Glosario

**Elementos:** juego de mesa en el que se ha basado el desarrollo del videojuego entregado en este proyecto.

**Motor de videojuego:** es una plataforma que permite el diseño, creación y funcionamiento de videojuegos.

**Multiplataforma:** indica que una misma aplicación puede ser ejecutada en distintas plataformas como Windows, Linux, IOS o Android. Unity permite desarrollar juegos y compilarlos para distintas plataformas sin cambiar el juego desarrollado.

**Terminal:** dispositivo, puede ser un teléfono, ordenador, Tablet, etc.

**Basic:** (**B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode) lenguaje de programación de alto nivel desarrollado en los años 60 muy utilizado para el aprendizaje de la programación.

**Spectrum:** los Sinclair ZX Spectrum eran unos ordenadores de 8 bits que fueron muy populares en Europa en los años 1980.

**Editor gráfico:** aplicación que permite realizar o modificar imágenes con el ordenador.

**Diagrama de Gantt:** visualización gráfica que permite mostrar los tiempos de dedicación previstos a diferentes tareas dentro de un proyecto en un periodo de tiempo.

**Github:** plataforma de desarrollo corporativo con control de versiones, ampliamente utilizada para el desarrollo de código abierto.

**Showcase:** dentro de los servicios de Unity, se dispone de un área donde mostrar los videojuegos realizados con Unity.

**APK:** (**A**ndroid **A**pplication **P**ackage) es el formato de fichero utilizado para empaquetar aplicaciones en Android. Se utiliza para distribuir e instalar componentes empaquetados para plataformas Android.

**Asset store:** tienda de activos. En Unity se dispone de una tienda donde los propios usuarios pueden vender sus propios desarrollos de activos y entornos, para que otros usuarios los compren para sus proyectos.

**UI:** (**U**ser **I**nterface) interfaz de usuario. En Unity tiene componentes propios con los que el usuario puede interactuar. Su uso típico son los menús y durante el juego las puntuaciones e información adicional.

**Prefabs:** es Unity es un objeto predefinido que se utiliza para replicarlo luego en el juego, de forma que todas las “copias” tienen los mismos atributos y funcionalidad.

**Scripts:** son las secuencias de comandos que contienen la lógica de programación que asociaremos a los distintos objetos del videojuego en Unity. En este proyecto los scripts se han desarrollado en C# con Visual Studio de Microsoft.

**Escenas:** en Unity una escena puede ser un menú o un nivel de juego. En cada escena colocamos los entornos, obstáculos y decoraciones, diseñando y construyendo el juego.

**Textura:** es una imagen que se utiliza para cubrir la superficie de un objeto gráfico.

**IA:** inteligencia artificial. En Unity se programaría la IA en scripts asociados a objetos, que reaccionarán según se les indique.

## 5. Bibliografía

- Juego de mesa Elementos  
<http://www.tytoqames.com/elementos-board-game.html>
- Entorno de desarrollo Unity  
<https://unity.com/es>
- Sonidos  
<https://freesound.org/>
- Imágenes  
Fuego: Icon made by freepic from [www.flaticon.com](http://www.flaticon.com)  
Madera: Icon made by Freepik from [www.flaticon.com](http://www.flaticon.com)  
Aire: Icon made by freepic from [www.flaticon.com](http://www.flaticon.com)
- Diagrama de Gantt  
<https://www.ganttproject.biz/>
- Bocetos gráficos  
<https://www.madewithmischief.com/>
- Ayuda desarrollo y documentación  
<https://forum.unity.com/>  
<https://docs.unity3d.com/Manual/index.html>

## 6. Anexos

- Código fuente  
<https://github.com/sartalUOC/ClashOfElements>
- Presentación  
<https://youtu.be/OCGwbumKrr4>