

# PRÀCTICA 2

## Battle Arena

Abel Selfa

Dani Valls

15/01/2021

## Introducció

L'objectiu d'aquesta pràctica era posar a prova tots els coneixements apresos a classe a llarg d'aquest primer semestre i ser capaços de realitzar un videojoc de l'estil DungeonCrawler 2D mitjançant HTML, CSS i JavaScript. A més a més, se'ns donava l'opció d'utilitzar un dels frameworks vistos a classe.

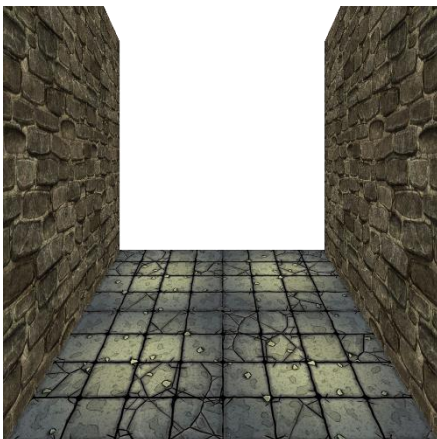
Aquest projecte l'hem afrontat amb moltes ganes perquè als dos ens agrada programar videojocs ja que els resultats obtinguts són més vistosos que no pas una pàgina web, l'usuari interactua d'una forma més activa amb el programa i a més a més el disseny de la interfície també és una cosa que ens apassiona, la qual cosa ha ajudat a fer menys pesada la realització de cada una de les tasques encomanades.

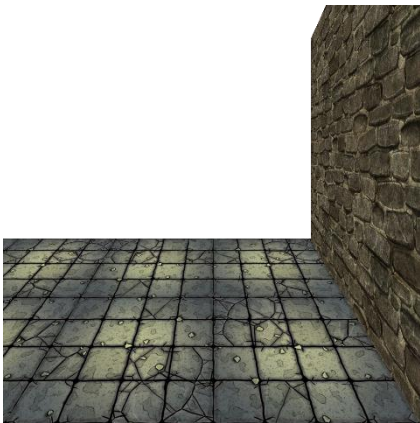
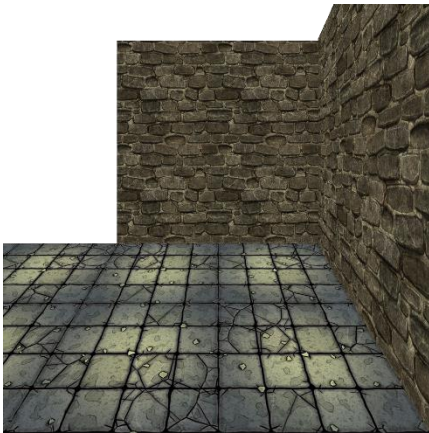
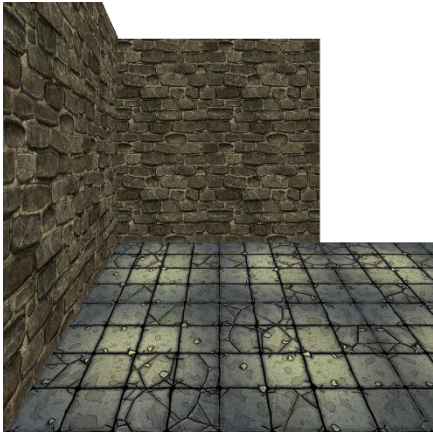
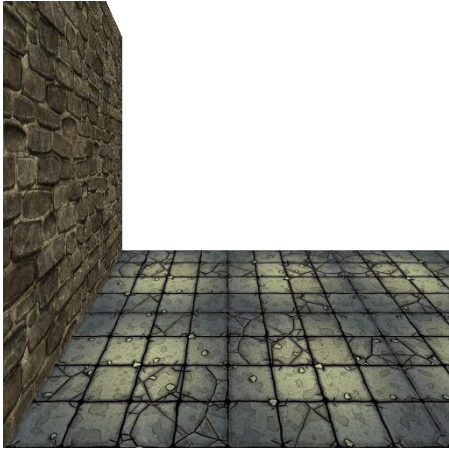
Les tasques obligatòries que s'havien de realitzar eren les següents:

- Moure el jugador pel mapa.
- Atacar als enemics. Es consideren enemics aquells jugadors que estan vius.
- Incloure un visor per poder visualitzar la casella que té davant el jugador i veure si hi ha o no enemics o objectes.
- Un minimapa que mostri la posició del jugador, dels enemics i les seves respectives direccions.
- Una brúixola que es vagi movent acord amb l'orientació del jugador.
- Un espai amb les característiques del jugador
- 3 opcions: crear un nou jugador, reviure i eliminar jugador

## Desenvolupament i resultats

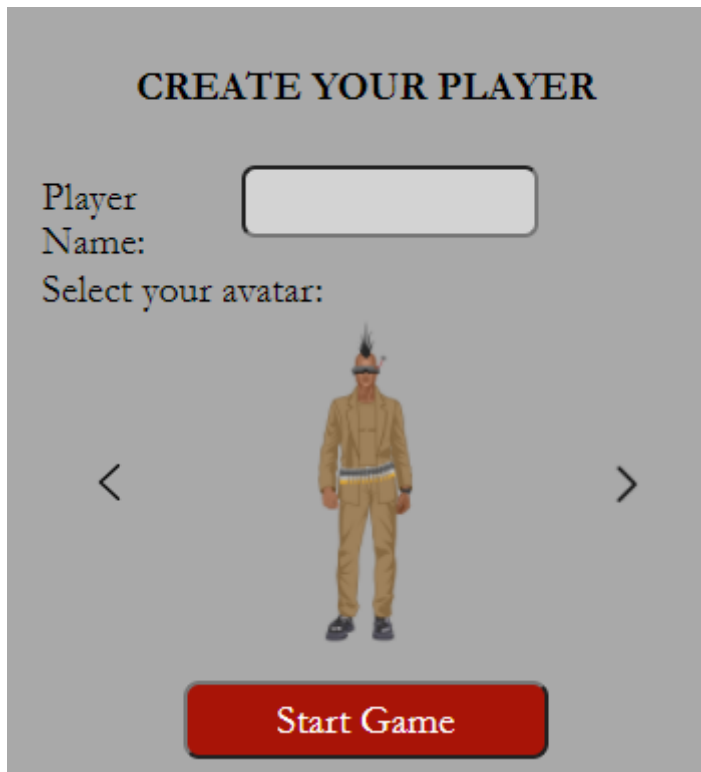
El que vam fer abans de res va ser la interfície. D'aquesta forma podríem saber com distribuir els diferents elements que se'ns imposava i veure si podíem afegir algun més d'opcional. A més a més, ens havíem de fer una idea de si el visor tindria forma quadrada o no per poder editar amb photoshop les diferents imatges de l'escenari per cada una de les casuístiques de la partida. Les imatges que vam haver de crear van ser les següents:





Etc.

Immediatament després d'haver realitzat una primera versió de la interfície en la que ja ens podíem fer una idea de com afrontar la lògica del joc, ens vam posar a fer les funcions que contenen els fetch per obtenir la informació de l'API de forma asíncrona. Aquest pas va ser un error no haver-lo fet abans, almenys llegir el que ens retornava cada una de les crides a l'API, ja que al crear el jugador havíem implementat que l'usuari pogués escollir quin avatar preferia, però al veure, més tard, que la crida Player de l'API ja ens retornava l'identificador de la fotografia pertanyent a l'avatar vam haver de reformular el codi. Així és com es veia la interfície al pitjar l'opció de "New Player":



Al pitjar sobre cada una de les fletxes et permetia canviar d'avatar.

Un cop realitzades les crides a l'API vam començar a implementar la lògica del joc. Havíem de controlar que al pitjar sobre cada un dels botons i cada una de les tecles, el visor, el minimapa i la brúixola s'actualitzessin de forma pertinent.

A continuació es mostren un seguit de captures d'algunes de les casuístiques de la partida:



Com es pot veure, a més a més del que es demanava, hem inclòs els punts de vida que tenen els enemics per tal de que el jugador sigui conscient de quant els queda de vida, si l'enemic està en mode fantasma (mort) apareix per pantalla amb menys opacitat, si en una casella hi ha més de tres jugadors (el màxim nombre que hem fixat perquè apareguin per pantalla) es mostra un missatge de quants mes hi ha, més informació actualitzada del que ens aporta la API com vindria a ser el percentatge de vida restant del jugador, els punts d'atac i defensa que se li ha atribuït i la imatge de l'avatar.

## **Tecnologies i eines**

Per la correcta realització de la pràctica hem hagut d'utilitzar molts dels coneixements apresos a classe i amb l'ajut d'internet ens hem pogut documentar de tots aquells aspectes més concrets com vindrien a ser alguns atributs de les classes de CSS.

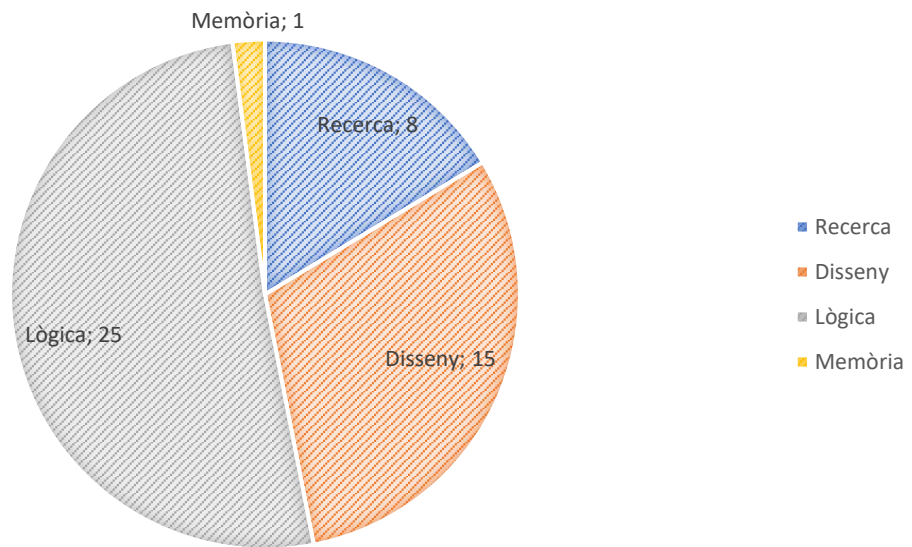
Hem desenvolupat tot el codi (inclosos els aspectes del disseny) des de JavaScript per mitjà del DOM. Hem creat una classe Player per guardar tota aquella informació relativa al jugador, com vindria a ser la vida, l'atac, la defensa, el nom... Hem utilitzat el Fetch per fer les crides a l'API, tornant-nos, d'aquesta forma Promises. Amb el async i el await fem que les funcions siguin asíncrones i no hi hagi retards de cap tipus.

## **Costos**

Per donar una estimació de les hores dedicades definirem les següents categories:

- Recerca: aquesta categoria reflexa el temps dedicat en obtenir informació, ja sigui a través dels apunts o d'internet.
- Disseny: tot el que correspon a la part visual del joc: la interfície i les imatges del visor realitzades a través de photoshop.
- Lògica: tota la part funcional del joc: crides a l'API, implementació de funcions, classes...
- Memòria: temps dedicat a redactar aquest document.

## TEMPS DEDICAT EN HORES



### Conclusions

Com hem dit abans aquesta pràctica ha sigut clau per acabar d'assolir tots els coneixements apresos al llarg del semestre i posar-los tots en pràctica en una única entrega que estem bastant satisfets del resultat. Creiem però, que amb una mica més de temps haguéssim pogut proporcionar un resultat millor. També si l'API ens proporcionés més informació ens hagués agradat poder mostrar estadístiques de la partida a temps real o informació relativa a les morts per exemple.