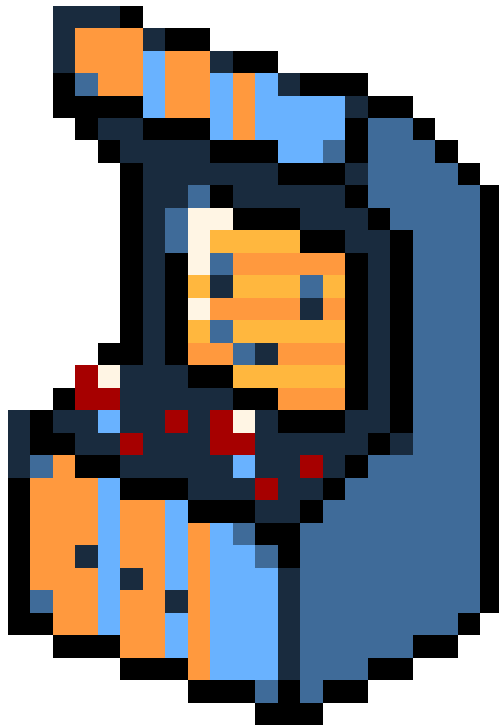


IES PUIG CASTELLAR

Creació d'un videojoc i d'una pàgina web



TREBALL DE RECERCA

Ferran Benéitez Borrut

2n de Batxillerat (201)

Tecnologia

Anna Sampedro Vizcaino / Jaime Morcillo

IES Puig Castellar

26/01/2022

- La esperanza es lo que nos hace más fuertes. Es la razón por la que estamos aquí. Es con lo que luchamos cuando todo lo demás se ha perdido - God of War

- Al final, ¿Qué distingue al hombre libre del esclavo? ¿Dinero? ¿Poder? No. El hombre elige. El esclavo obedece - Bioshock

- Hay cosas que olvidamos... Y hay cosas que no podemos olvidar, es gracioso, no sé cual es mas triste. - Silent Hill 3

Resumen

Hoy en día los videojuegos son una de las formas de entretenimiento más populares entre los jóvenes, llegando a ser el trabajo de muchos. Sin embargo, no se le da mucha visibilidad al proceso de creación de uno. El objetivo de este trabajo es elaborar un videojuego y documentar todo el proceso y trabajo que conlleva. Además, también tiene como objetivo elaborar una página web, otro elemento muy presente en la vida de casi cualquier persona, pero que se le da muy poca visibilidad a todo el trabajo que hay detrás. El videojuego está inspirado en el famoso videojuego Geometry Dash, el cual formó parte de la infancia de muchos adolescentes de hoy en día, entre los cuales me encuentro yo. La página web servirá para permitir a los jugadores descargar el juego y enviar feedback al creador. Durante el proceso me he topado con muchos problemas y he podido darme cuenta de la dificultad que conlleva un proyecto tan grande como es un videojuego y una página web. Todo esto me ha hecho valorar más los videojuegos a los que juego y las páginas que visito y me ha servido para no criticar a la ligera el trabajo de otros debido a que durante 9 meses he estado en su piel y he tenido que lidiar con todos los problemas que surgen haciendo algo así.

Abstract

Today video games are one of the most popular forms of entertainment among young people, becoming the work of many. However, not much visibility is given to the process of creating one. The objective of this work is to develop a video game and document the entire process and work involved. In addition, it also aims to create a web page, another element that is very present in almost anyone's life, but which gives very little visibility to all the work behind it. The video game is inspired by the famous Geometry Dash video game, which was part of the childhood of many teenagers today, including myself. The website will serve to allow players to download the game and send feedback to the creator. During the process I have run into many problems and I have been able to realize the difficulty of a project as large as a video game and a website. All this has made me value the video games that I play and the pages that I visit more and it has helped me not to lightly criticize the work of others because for 9 months I have been in their shoes and I have had to deal with all the problems that arise doing something like that.

Índex

1. INTRODUCCIÓ	1
2. HIPÒTESI	2
3. DESENVOLUPAMENT	3
3.1 Què són els llenguatges de programació?	3
3.1.1 Breu introducció a HTML	6
3.2 Com es fa un videojoc?	8
3.2.1 Creació de la idea	9
3.2.2 Concepció	10
3.2.3 Disseny	11
3.2.4 Planificació	12
3.2.5 Preproducció	13
3.2.6 Producció	13
3.2.7 Proves	14
3.2.8 Manteniment	15
3.3 Com es fa una pàgina web?	16
3.3.1 Recopilació de dades	17
3.3.2 Planificació	17
3.3.3 Disseny	18
3.3.4 Redacció del contingut i del SEO	18
3.3.5 Programació	18
3.3.6 Proves	19
3.3.7 Llançament	19
3.4 Seria possible penjar el meu videojoc a Steam o Epic Games?	19
3.5 Què és el SEO i com funciona?	21
3.6 Què és el Pixel Art?	24
3.6.1 Història del Pixel Art	24
3.6.2 Tècnica	29
3.6.3 Categories	30
3.7 Història dels videojocs	30
4. CREACIÓ DEL VIDEOJOC	39
4.1 Brainstorm	40
4.2 Disseny	50
4.2.1 Sprites del jugador	50
4.2.2 Sprites dels nivells	52
4.3 Programació de les mecàniques bàsiques	54
4.3.1 Moviment constant en l'eix x	54
4.3.2 Salt	55
4.3.3 Mort	55
4.4 Creació del primer nivell beta	56
4.5 Creació de les mecàniques més avançades	59

4.6 Creació dels nivells	64
4.7 Creació del menú	65
4.8 Retocs	67
5. CREACIÓ DE LA PÀGINA WEB	68
5.1 Estructura	68
5.2 Part visual de la pàgina web	69
5.3 Programació	70
5.4 Resultats de la pàgina web	72
6. CONCLUSIONS	73
7. AGRAÏMENTS	76
8. WEBGRAFIA	77
9. ANNEX	79
9.1 Sprites de la primera idea de videojoc	79
9.2 Sprites del videojoc final	85
9.3 Scripts del videojoc	87
9.3.1 Script del protagonista	87
9.3.2 Scripts d'altres parts del videojoc	96
9.4 Scripts de la pàgina web	101

1. INTRODUCCIÓ

La indústria dels videojocs se situa actualment en les 10 indústries que més diners generen al món. Arribats a aquest punt és sensat afirmar que ha tingut un impacte important en les nostres vides, especialment en els més joves. A causa d'això i a la meva passió pels videojocs, he decidit fer aquest Treball de Recerca.

El meu treball consisteix en la creació d'un videojoc 2D, en l'estil Pixel Art. Fet en la plataforma Godot Engine i després en la creació d'una pàgina web, mitjançant els llenguatges de programació HTML, Javascript i CSS per tal de promocionar el meu videojoc com si fos una empresa per poder fer que tothom pugui descarregar aquest videojoc de forma gratuïta a internet.

He escollit aquest treball perquè personalment em fascina el món de la programació i disseny de videojocs, passo una part considerable del meu temps jugant-los i crec que seria interessant posar-me en la pell de la gent que els crea. A més a més, és un tema que engloba dos àmbits molt diferents com són l'àmbit tecnològic i l'àmbit artístic. Tot i que forma més part de l'àmbit tecnològic i aquest treball va més encaminat per aquesta banda, també hi ha una gran importància en l'àmbit artístic. Això, engloba tot el disseny de personatges i escenaris mitjançant il·lustracions fetes a ordinador.

Els objectius d'aquest treball són diversos:

1. Aprendre a fer un videojoc en Godot.
2. Aprendre a fer una pàgina web.
3. Aprendre a publicar aquesta pàgina a internet.
4. Aprendre sobre el SEO.
5. Aprendre a patentar un videojoc.
6. Aprendre Javascript, HTML, CSS i el llenguatge propi de Godot Engine.
7. Aprendre pixel art i PhotoShop.

2. HIPÒTESI

Per poder començar amb la recerca del treball, m'he plantejat la següent hipòtesi:

Serà possible la creació d'una pàgina web i un videojoc sense tenir coneixements previs de programació?

A partir d'aquí em sorgeixen unes qüestions, que són les següents:

- **Què són els llenguatges de programació?**
- **Com es fa un videojoc?**
- **Com es fa una pàgina web?**
- **Seria possible penjar el meu videojoc a Steam o Epic Games?**
- **Com es posiciona una pàgina web a internet?**
- **Què és el SEO?**
- **Què és el Pixel Art?**
- **Com és el disseny de personatges i entorns en un videojoc?**

Per poder assolir els meus objectius, és completament necessari respondre a totes aquestes qüestions. Primer duré a terme una etapa de recerca on el propòsit serà adquirir tots els coneixements necessaris per respondre aquestes qüestions i per poder crear el meu videojoc i pàgina web.

3. DESENVOLUPAMENT

3.1 Què són els llenguatges de programació?

Per tal de definir que són els llenguatges de programació primer de tot he d'introduir uns altres conceptes abans.

Algoritme: Un algoritme és un conjunt d'operacions, ordres o instruccions que ens permeten duu a terme una acció per tal d'arribar a un propòsit. En matemàtiques són utilitzats per trobar la solució a un problema i està compost per un conjunt de càlculs ordenats.

Sistema informàtic: Un sistema informàtic és un sistema que permet emmagatzemar i processar informació. Està format per hardware, software i el personal informàtic.

Hardware: El hardware és el conjunt de parts físiques d'un ordinador, és a dir, el teclat amb el qual escrivim (el virtual no), les pantalles, el ratolí i l'ordinador en si (placa base, gràfica, processador, ram, ventiladors de refrigeració, font d'alimentació i el gabinet en si).

Software: El software són les instruccions que li diuen a un ordinador que ha de fer. Està format per programes, procediments i rutines associades amb el funcionament d'un sistema informàtic.

Sistema Operatiu: Un sistema operatiu és un conjunt de programes que controla tot l'ordinador, és el que ens permet utilitzar-ho. S'encarrega de la memòria, dels nostres perifèrics, etc. Exemples: Ubuntu, MacOS, Windows, Linux, Fedora, Black Berry OS, etc.

Els llenguatges de programació són la manera que tenim d'escriure un seguit d'algoritmes formats per ordres o instruccions que ens permeten controlar el comportament físic o lògic d'un sistema informàtic. En altres paraules, un llenguatge de programació és un sistema de comunicació, una manera de comunicar-se, formada per paraules, símbols, números..., que ens permet comunicar-nos amb els sistemes informàtics o màquines. Cal recalcar que aquest sistema de comunicació també conté regles semàntiques i regles sintàctiques que han de ser respectades per tal de garantir el correcte funcionament.

Cada llenguatge de programació està especialitzat o dissenyat per una funció en concret encara que qualsevol llenguatge pot ser utilitzat per qualsevol cosa encara que no sigui el més eficient.

Els 6 llenguatges més utilitzats són els següents:

- Java: És el llenguatge de programació més usat del món i el més estandarditzat. És un llenguatge de programació completament fonamental i imprescindible que tot programador ha de conèixer. Una de les seves utilitats és la capacitat que té per fer servir el mateix programa en diversos sistemes operatius.

- Python: Actualment, és el segon llenguatge més utilitzat, però cada vegada està agafant més popularitat. Aquest llenguatge pot ser emprat per emmagatzematge i extracció de dades, disseny web, serveis de pagament, intel·ligència artificial, etc.

- Javascript: Javascript és un dels llenguatges més importants a l'hora de dissenyar pàgines web, més del 95% de les pàgines a internet han estat programades amb Javascript.

- C++: C++ o "C Plus Plus" és un llenguatge orientat a objectes, és a dir, orientat a programar el comportament físic d'aquests, un exemple seria programar les físiques d'un videojoc o calcular una paràbola. És molt utilitzat en la creació de videojocs conjuntament amb C#.

- PHP: Va ser creat sense la intenció de convertir-se en un llenguatge web, sinó que va ser creat amb la intenció de ser utilitzat com una sèrie d'eines per al manteniment de les pàgines web.

- Swift: És un llenguatge creat per la companyia Apple i està enfocat en el desenvolupament d'aplicacions macOS i iOS (iPhone i iPad).

Els llenguatges de programació que jo utilitzaré són:

-HTML

-Javascript

-CSS

-Llenguatge propi de Godot

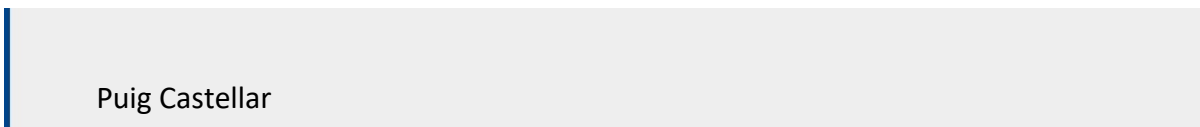
3.1.1 Breu introducció a HTML

En aquest apartat faré una breu introducció a HTML per tal que qualsevol persona sense coneixements previs pugui entendre la meva pàgina web.

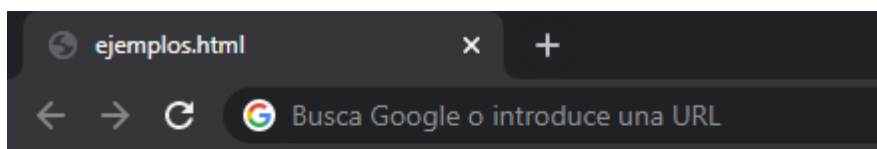
HTML

HTML (Hypertext Markup Language) no és considerat llenguatge de programació totalment, sinó que està dedicat a l'estructura de la pàgina web. En altres paraules, només és disseny, indica a la pàgina web com s'ha d'estructurar.

Per exemple:



Aquesta línia de codi donaria el següent resultat:



Puig Castellar

Imatge 1. Captura del *input* del codi anterior. *Imatge amb autor propi.*

Malgrat això, no és molt recomanable escriure el codi sense especificar res, en aquest cas si volguéssim especificar que és un paràgraf hauríem de fer el següent encara que tingui el mateix resultat:

```
<p> Puig Castellar </p>
```

Aquests símbols són etiquetes utilitzades per definir elements. Els elements són el conjunt de les etiquetes i el contingut. Les etiquetes principals són les següents:

```
<!DOCTYPE html>
```

Aquesta etiqueta defineix que és un document HTML5.

```
<html>
```

És l'element arrel d'una pàgina HTML.

```
<head>
```

És on el títol de la pàgina es posa i conté "meta" informació de la pàgina

```
<title>
```

Especifica el títol per a la pàgina que es mostra en la pestanya de la pàgina.

```
<body>
```

Defineix el cos del document i és on s'introdueixen tots els continguts visibles com els paràgrafs, imatges, enllaços, taules, llistes...

```
<h1>
```

S'utilitza per definir una secció gran, també existeix `<h2>` , `<h3>` , `<h...>`.

```
<p>
```

Defineix un paràgraf.

L'estructura més bàsica d'una pàgina web és la següent:

```
<html>
  <head>
    <title> Títol de la pàgina </title>
  </head>

  <body>
    <h1> Això és una secció </h1>
    <p> Això és un paràgraf </p>
    <p> Això és un altre paràgraf </p>
  </body>
</html>
```

3.2 Com es fa un videojoc?

Un videojoc és un projecte complex on intervenen moltes persones i requereix planificació, coordinació i molt treball. Com a qualsevol altre projecte és necessària una fase de planificació per a poder treballar adequadament.

Per elaborar un bon videojoc s'han de passar per unes etapes que garanteixen un ordre a seguir per tal d'evitar problemes. Seguir un ordre no només serveix per evitar dificultats, sinó, que també serveix per garantir l'assoliment dels objectius establerts prèviament. A més a més, en estudis grans on participen un gran nombre de persones això permet que tots treballin coordinats. Normalment en els estudis també existeix un director que s'encarrega de que tots els subequips actuïn coordinats, al dia, en bones condicions i que tot connecti entre si. Per exemple, si els equips de disseny estan dissenyant un personatge que no permet implementar certes mecàniques en la fase del codi del videojoc hauran de tornar a començar. El director s'encarrega precisament de que tot el treball tingui relació entre si i es pugui adaptar a les altres fases.

Les etapes de la creació d'un videojoc són les següents:

1. Creació de la idea sobre la qual es construirà el videojoc
2. Concepció
3. Disseny
4. Planificació
5. Preproducció
6. Producció
7. Proves
8. Manteniment

3.2.1 Creació de la idea

En aquesta etapa és on comença tot, primer de tot sorgeix una idea, una motivació, una necessitat de crear un videojoc, de contar una història, etc. No només existeixen videojocs per entretenir, passar una bona estona o guanyar diners. Hi ha videojocs creats amb finalitats més diverses, videojocs que serveixen per contar històries, per posar en evidència problemes que ens afecten a tots, videojocs personals que serveixen per contar vivències pròpies de qui l'ha creat, etc.

3.2.2 Concepció

Una vegada la idea del videojoc ja està definida cal plantejar-se els aspectes generals del videojoc.

- **Gènere:** Hi ha una infinitat de gèneres de videojocs completament diferents, és per això que cal especificar. En cas de no ser un gènere molt conegut cal especificar les seves característiques.

Exemples:

- Acció: Lluita, arcade, plataforma...
 - Trets: Primera persona, tercera persona, "shoot 'em up"...
 - Estratègia: En temps real, per torns...
 - "Hack and Slash"
 - Simulació: Simulació de vehicles, simulació de construcció, simulació de vida, simulació de combat...
 - Esports
 - Carreres
 - Aventura: Conversacional, gràfica, acció-aventura (horror de supervivència, sigil, "Battle Royale")
 - Rol
 - Sandbox
 - Musical
 - Agilitat mental
 - "Party games"
 - Educació
-
- **Gameplay:** En aquest apartat cal definir quin serà l'atractiu del videojoc, és a dir, que farà aquest joc diferent dels altres i farà que la gent el jugui. Aquest és un apartat molt ampli i variat, ja que pot ser des de la manera en la qual es juga, les mecàniques, la història del videojoc "lore", el missatge que vol transmetre, etc.

- **Conceptes:** Per últim, en aquest apartat cal començar a fer els primers esbossos i a tenir les primeres idees sobre com funcionarà el videojoc, la seva música, personatges, ambientació, etc.

3.2.3 Disseny

Una vegada ja es té una idea de com serà el videojoc i es tenen els primers esbossos arribem a la part de disseny. Aquesta és una de les parts més importants, en aquesta etapa és on realment es crea la base sobre la que es construirà el videojoc. En aquesta part cal definir els següents punts:

1. **Història del videojoc:** És a dir, el "lore", la història del món on té lloc, la història de cada un dels personatges i com evolucionaran amb el temps. Cal recalcar que hi ha videojocs arcade com el Tetris que no tenen història i aquesta part no és necessària.
2. **Guió:** En aquest punt l'equip fa una reunió per tal que tothom pugui aportar les seves idees i debatre-les entre si. Després, un equip s'encarrega de posar uns objectius establerts anteriorment, a més a més, també s'encarreguen de definir les parts en les quals es dividirà el videojoc, el context on es desenvoluparà i els personatges principals.
3. **Art:** En aquesta part un grup d'artistes s'encarreguen de crear conceptes dels personatges, escenaris, enemics, paisatges, objectes, etc. Per crear aquests conceptes es basen en les idees originals dels creadors i també aporten modificacions. Una vegada fets els conceptes el director d'art s'encarrega d'escollir les idees que s'adaptin millor al videojoc i a les idees originals dels creadors.

4. **So:** Es detallen tots els sons que seran necessaris a l'hora de fer el videojoc, veus, sons d'ambient, efectes sonors i música.
5. **Mecàniques:** S'especifica el funcionament general del videojoc. Aquesta part és molt similar al que seria la física de la realitat, aquí és on es concreta la manera en la qual interactuaran els cossos i els objectes entre si.
6. **Disseny de programació:** Per últim, es fa una reunió entre els programadors i decideixen quina plataforma de creació de videojocs donarà vida a la idea i quins llenguatges de programació s'utilitzaran.

Al final d'aquesta etapa es crea un document que conté totes les especificacions d'art, mecàniques i programació. Això ajuda al fet que si en cap moment de la producció es desvien, tornen a aquest document i recuperen les idees principals.

3.2.4 Planificació

En aquesta fase és on s'identifiquen les tasques necessàries i es reparteixen entre els diferents equips de desenvolupament del videojoc. També es posen dates i terminis per l'entrega de diferents tasques i s'estableixen dies de reunió claus per avaluar l'evolució del projecte i si arribaran a la data límit.

3.2.5 Preproducció

Una vegada que està tot planificat i es té el document de disseny, és important comprovar la viabilitat del videojoc. Aquesta part s'encarrega a un equip petit la tasca de crear un nivell del videojoc o una síntesi per comprovar que és viable la seva realització.

3.2.6 Producció

En aquesta fase és on es crea el videojoc en si. Se segueixen totes les especificacions donades en la fase de planificació seguint el document elaborat a la fase de disseny per tal de no desviar-se de les idees originals.

Les parts més importants de la producció són:

- **Il·lustració:** En videojocs 2D els il·lustradors han de ser persones experimentades en el camp. Això és degut al fet que han de tenir en compte les limitacions que pot presentar el hardware on el videojoc serà jugat. És a dir, no poden fer animacions o "sprites" amb resolució 4K si el videojoc està plantejat per ser jugat en una Nintendo del 2010. Han de tenir molts factors en conte i és per això que és necessari que siguin il·lustradors experimentats no tan sols en el seu camp sinó, en l'àmbit dels videojocs en general.
- **Interfície:** Són els elements de la interfície gràfica i HUD del videojoc. Exemple: barra de vida, objectes, menú, etc.

- **Animació i modelatge 3D:** En videojocs en tres dimensions els elements han de ser modelats, és a dir, han de tenir profunditat ja que són en 3 dimensions. Aquest és un procés molt més complicat que la il·lustració 2D d'imatges planes. Per dur a terme aquesta tasca s'utilitzen programes especialitzats en modelatge 3D com són: Blender, 3DS Max, Maya, etc. (El més conegut i utilitzat és Blender, a més a més és gratuït).
- **So:** Els enginyers de so s'encarreguen de crear tots els sons que necessita el videojoc.
- **Programació:** Aquesta part és una de les més importants perquè és el que permet que el videojoc funcioni, programar el codi. La gran majoria de videojocs es programen en C++ ja que és un llenguatge amb una dificultat no molt elevada i permet una compatibilitat molt bona amb el hardware de les consoles. Tanmateix, no és l'únic llenguatge usat, Java, Javascript i Lua són també molt populars.

3.2.7 Proves

Una vegada que el videojoc ja està creat ha de passar per una fase de proves abans de ser entregat al consumidor. En aquesta fase es corregeixen errors tant de codi, visuals, *bugs*, *exploits*, *glitches*, etc. A més a més, també es retoca el videojoc en general per tal de garantir la jugabilitat.

Aquesta fase de proves es divideix en tres fases:

1. **Proves físiques:** Els programadors i dissenyadors s'encarreguen de crear prototips d'escenes o situacions que poden ocórrer en el videojoc. Aquest tipus de proves poden prevenir problemes de programació. També és molt útil perquè permet a l'equip tenir feedback directe sobre l'opinió del jugador per tal de millorar o ajustar diversos aspectes del videojoc.
2. **Proves alpha:** Són proves fetes per persones que ja formaven part de l'equip de creació prèviament i s'utilitzen per solucionar errors greus i millorar característiques no contemplades en la fase de disseny.
3. **Proves beta:** Una vegada les proves alpha terminen es proporciona la possibilitat de provar el videojoc a un equip extern de jugadors per tal de jugar i portar al límit el videojoc. Aquestes proves tenen la finalitat d'eliminar tots els errors importants i reduir els més lleus.

3.2.8 Manteniment

Encara que el videojoc ja estigui a disposició de l'usuari cal mantenir-lo i corregir els errors que vagin sortint. Això es fa mitjançant la recopilació de dades dels jugadors, reports dels jugadors i analitzant el videojoc.

En el cas de trobar errors l'empresa o equip de desenvolupament els corregeix i els incorpora mitjançant l'ús d'actualitzacions. Aquestes actualitzacions poden ser correcció d'errors que afectin la jugabilitat, disseny o incorporar nou contingut. Sempre són gratuïtes.

Per una altra banda, també existeixen actualitzacions que es paguen, normalment són anomenades expansions o DLC (Downloadable content) i solen agregar bastant contingut nou al videojoc.

3.3 Com es fa una pàgina web?

De la mateixa manera que un videojoc necessita planificació una pàgina web també. Per tal de crear una pàgina web no només cal tenir coneixements de programació o disseny, a més a més també cal entendre el funcionament d'internet. Això és degut al fet que necessites donar visibilitat a la teva pàgina, necessites proporcionar servidors i has de tenir present que sigui compatible amb qualsevol dispositiu electrònic amb accés a internet. Per tant, per elaborar una pàgina web s'han de tenir uns coneixements molt més amplis que pels d'un videojoc.

La creació d'una pàgina web es divideix en set parts:

1. Recopilació de dades
2. Planificació
3. Disseny
4. Redacció del contingut i SEO
5. Programació
6. Proves
7. Llançament

3.3.1 Recopilació de dades

Aquesta és la primera fase i és molt important perquè ens permetrà dissenyar una pàgina web d'acord amb les nostres necessitats. Primer de tot hem d'identificar els objectius que volem aconseguir amb la pàgina web, el propòsit en si i el públic al qual va dirigida. A més a més, és rellevant tenir una idea general de quin serà el contingut de la pàgina.

3.3.2 Planificació

Després de la recopilació de dades és recomanable fer un mapa o esquema de la pàgina web. Això ens permet saber quina serà la magnitud de la web, ens permetrà crear l'estructura i podem saber quin tipus de tecnologies i/o els llenguatges de programació necessitarem. Per exemple, si fem l'esquema de la pàgina web i ens adonem que hi ha un mini joc en línia o models 3D animats ens haurem de plantejar usar Javascript o CSS. Un altre exemple seria si trobem que hi ha transaccions a la nostra pàgina, en aquest cas és molt important tenir present que haurem d'utilitzar el llenguatge Python per al tractament de dades personals.

Cal tenir en compte que l'esquema de la pàgina web no només s'ha de plantejar per a un ordinador, sinó que també s'ha de tenir present la compatibilitat amb dispositius mòbils. Això pot fer que la pàgina variï en funció del dispositiu, ja que no tots tenen la mateixa grandària ni la mateixa resolució i a vegades no és viable l'adaptació.

3.3.3 Disseny

L'etapa de disseny és potser si no la més una de les més importants, durant aquesta etapa és important centrar-se en el disseny visual de la pàgina web. És crucial que la comunicació sigui clara (user-friendly), colors que representin la marca o el missatge que es vol transmetre. Per exemple, una pàgina web de McDonald's hauria de ser blanca, groga i vermella, ja que són els colors emblemàtics de la marca. A part de tot això la prioritat és sempre que sigui fàcil d'entendre i d'usar per a l'usuari, d'aquesta manera la seva experiència millorarà i per conseqüència el SEO també.

3.3.4 Redacció del contingut i del SEO

En aquesta part cal redactar els continguts que es mostraran a la web i que seran a la vista dels usuaris. A més a més, també cal incloure imatges i vídeos per tal de fer la pàgina web més dinàmica. Normalment, aquests textos són escrits primer a mà per un redactor, són corregits, passen a la persona que està al cap de la pàgina web per revisar-los i per últim acaben a un programador qui és el que els implementa digitalment.

Per una altra banda, en aquesta fase també és important optimitzar el SEO de la pàgina web pels motors de cerca digital.

3.3.5 Programació

Una vegada s'arriba a aquest punt els programadors comencen a fer el seu treball. Programen la pàgina web, normalment en HTML, CSS o PHP. En aquesta part no només cal tenir present el seu funcionament a ordinador, sinó que com passa als videojocs, han de tenir en compte el (responsive size) què és el que els permetrà la compatibilitat amb dispositius mòbils. D'aquesta manera s'arriba a un major nombre d'usuaris.

3.3.6 Proves

L'última fase abans del llançament oficial de la pàgina web consisteix a fer proves per garantir el correcte funcionament de la pàgina web. La pàgina és analitzada des de diferents navegadors i des de diferents dispositius.

En aquesta part també es fa una reunió a la qual assisteixen tots els membres que han format part del desenvolupament per tal d'avaluar si s'ha arribat als objectius establerts anteriorment.

3.3.7 Llançament

Una vegada tot aquest procés acaba la pàgina és publicada oficialment per tal que tothom ho pugui veure. Amb el pas del temps continua sent necessari un manteniment tècnic i d'actualització dels continguts.

3.4 Seria possible penjar el meu videojoc a Steam o Epic Games?

Epic Games Store i Steam són les dues plataformes de distribució digital de videojocs més grans actualment.

Steam va ser creada el setembre de 2003 amb la finalitat de proporcionar actualitzacions per als videojocs de l'empresa Valve. En canvi, amb el pas del temps ha acabat sent una plataforma digital per comprar videojocs tant de Valve com d'altres companyies.

D'altra banda, Epic Games Store va aparèixer el desembre de 2018 amb un escàs estoc de videojocs, malgrat això va aconseguir augmentar exponencialment de tal manera que ha aconseguit fer competència a Steam.

Tant Steam com Epic Games són dues plataformes on qualsevol companyia o un SoloDev (una única persona que fa videojocs sola) pot penjar el seu videojoc. A l'hora de penjar el teu videojoc només hi ha dues grans diferències significatives a tenir en compte. Steam es queda amb un 30% dels ingressos que té un videojoc a la seva plataforma, 25% pels videojocs que venen entre 10 i 50 milions de dòlars i 20% pels que venen més de 50 milions de dòlars; en canvi, Epic Games només es queda amb un 12% dels ingressos de les vendes dels videojocs.

L'altra gran diferència és que a l'hora de publicar el teu videojoc en Steam has de pagar prèviament 100 € i a Epic Games Store és gratuït.

Aquesta diferència de preus és causada degut al fet que Epic Games Store no presenta beneficis monetaris per a l'empresa Epic Games. Malgrat això sí que els hi proporciona molta publicitat i usuaris que generen que altres dels seus productes sí que generin beneficis. D'aquesta manera es poden permetre perdre diners en Epic Games Store per tal de fer competència amb Steam i atreure més clients a altres dels seus productes.

Amb tota aquesta informació és evident que puc publicar el meu videojoc a qualsevol de les dues plataformes. Però, com el meu propòsit no és generar beneficis, sinó aprendre a fer un videojoc i passar per tot el procés pel qual passen totes les companyies, en cas de publicar-lo en alguna plataforma seria a la Epic Games Store.

3.5 Què és el SEO i com funciona?

SEO (Search Engine Optimization) és el procés que millora les cerques en els resultats orgànics. És a dir els que apareixen d'acord amb la cerca sense enllaç i no són pagats. És la manera que tenen els buscadors de filtrar els resultats.

Ad · <https://www.nike.com/> ▼

Nike - Online Store - The Official Nike Site

Any Place. Any Time. Any Condition. Be Your Best Every Time With **Nike** Shoes And Clothing.

Imatge 2. Resultat que ha estat pagat. *Fotografia amb autor propi.*

Resultat pagat

https://www.nike.com ▼

Nike. Just Do It. Nike.com

Nike delivers innovative products, experiences and services to inspire athletes.

Imatge 3. Resultat que no ha estat pagat. *Fotografia amb autor propi.*

Resultat orgànic trobat per SEO

Aquesta diferència es pot veure per l'etiqueta "Ad", a la primera imatge l'empresa ha pagat per estar, en canvi, a la segona és resultat d'una cerca orgànica.

Existeixen molts factors mitjançant els quals un motor de cerca filtra els seus resultats, però els més comuns són el d'autoritat o fama i el de rellevància.

1. Autoritat. Aquest factor consisteix a basar-se en la popularitat de la pàgina web. És a dir, com més popular és una web més probable, és que surti en una cerca orgànica. Aquest criteri segueix la lògica de, si la visita molta gent serà perquè és útil.
2. Rellevància. Aquest criteri se centra en la relació de la cerca amb el contingut de la pàgina. És a dir, si busques "restaurants Barcelona" et sortiran les pàgines on aparegui la combinació de paraules "restaurants Barcelona". Aquestes paraules són anomenades "Key Words", són paraules clau que tenen una relació molt concreta amb el que busques. A més a més també segueixen altres criteris on-site.

El SEO es divideix en dos grans grups

1. On-Site: El SEO On-Site es basa principalment en la rellevància de la pàgina web. També es basa en les "Key Words", temps de càrrega, optimització del codi i el format dels URL.
2. Off-Site: El SEO Off-Site al contrari que el SEO On-Site es basa en factors externs a la pàgina com el seu nom diu (Fora del lloc). Els factors més importants a tenir en compte en el SEO Off-Site són el número i la quantitat dels enllaços, presència en xarxes socials, rendiment en els resultats de cerca i l'autoritat de la marca.

Com es millora el posicionament d'una pàgina web?

Existeixen dues maneres de millorar el nostre SEO, es divideixen en Black Hat SEO i el White Hat SEO.

La principal diferència és que el Black Hat SEO utilitza tècniques considerades poc ètiques com per exemple "spam" de la seva pàgina en fòrums que no tenen res a veure. Aquestes tècniques són efectives a curt termini, però a la llarga no són efectives. A més a més, és una estratègia arriscada, ja que a la majoria de fòrums o xats aquesta activitat està penalitzada. En canvi, el White Hat SEO consisteix a millorar l'experiència de l'usuari i, per tant, fer més rellevant la pàgina.

Importància del SEO

El SEO compleix un paper fonamental en una pàgina web, el SEO permet donar visibilitat a les pàgines web i, per tant, atreure més usuaris. En el cas d'una empresa com Adidas per exemple, necessiten tenir un SEO molt ben fet i optimitzar per competir amb la seva competència Nike. Això vol dir, si l'usuari busca "sabates esportives" al buscador l'empresa que tingui millor SEO serà la primera que apareixerà i tindran més possibilitats que l'usuari compri a la seva pàgina web, per tant, hauran guanyat un client.

3.6 Què és el Pixel Art?

El Pixel Art és una forma de crear art digitalment, és un tipus d'art. Aquest tipus d'art és creat amb l'ús d'una computadora i programes específics d'edició. El Pixel Art consisteix a editar o crear imatges al nivell dels pixels. És a dir, pintes o edites pixel per pixel (similar al puntillisme).

Aquesta tècnica està molt present en videojocs per PC i videoconsoles antigues i també en obres actuals. Aquesta tècnica va ser publicada per primera vegada el 1982 per Adele Goldberg i Robert Flegal del Centre d'Investigació de Xerox en Pablo Alto.

3.6.1 Historia del Pixel Art

El Pixel Art és una forma d'art digital creada mitjançant ordinadors on les imatges són editades pixel a pixel.

El Pixel Art va ser creat especialment pels videojocs antics que utilitzaven resolucions baixes. Algunes de les consoles o videojocs més destacats són: Nintendo Entertainment System (consola), Space Invaders (videojoc), Pac-Man (videojoc), etc.

Encara que aquesta tècnica va ser publicada oficialment per primera vegada el 1982 per Adele Goldberg i Robert Flegal el concepte ja existia. Aquest concepte del Pixel Art es remunta al desenvolupament del sistema SuperPaint de Richard Shoup en 1972.

El Pixel Art ha estat una part molt important dels videojocs al llarg de la història, a més a més ha anat evolucionant amb el pas del temps fins a arribar al Pixel Art que coneixem actualment.

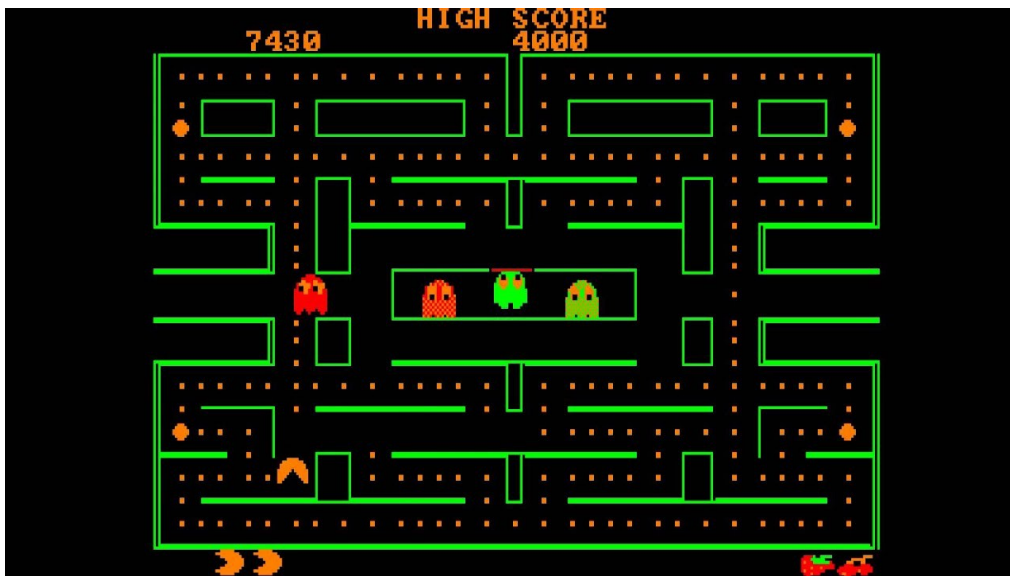
Els primers anys (1972 - 1983)

Els primers anys del sorgiment del pixel art van ser potser els més difícils pels desenvolupadors i dissenyadors. Dos grans causants d'aquesta dificultat són les limitacions tecnològiques i la falta d'experiència dels creadors de videojocs en aquest camp. Les limitacions tècniques no permetien crear les imatges en una qualitat adequada i les eines que tenien per fer-ho eren molt limitades. D'una altra banda, era una tècnica nova mai abans vista, tots eren pioners, estaven creant les bases del Pixel Art.

A causa de tot això les il·lustracions creades amb aquesta tècnica eren molt simples i requerien part de la imaginació del jugador per tal de cobrar sentit.

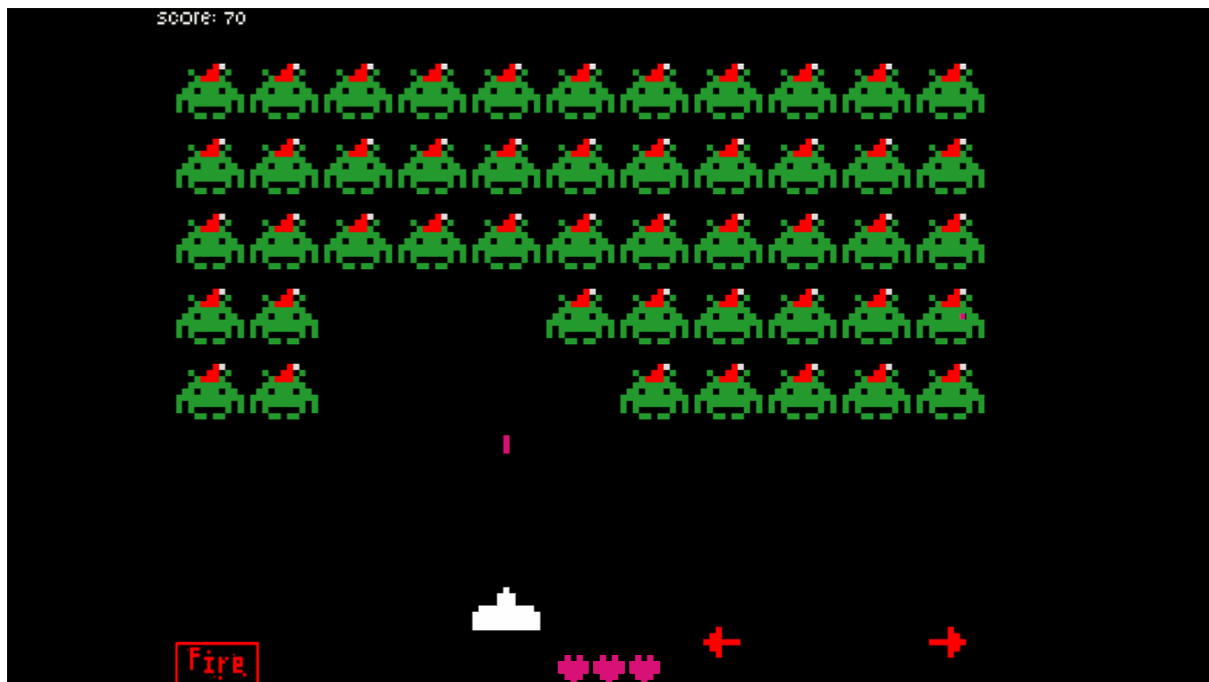
Alguns exemples d'aquests videojocs són els següents:

Videojoc Pac Man, Bandai Namco Entertainment



Imatge 4. Captura de pantalla del videojoc *Pac Man*. Imatge extreta de <https://www.youtube.com/watch?v=2u0427GdT10>

Videojoc *Space Invaders*, Taito



Imatge 5. Captura de pantalla del videojoc *Space Invaders*, Taito. Imatge extreta de <https://mundowin.com/>

L'era dels 8 bits (1983 - 1987)

Durant aquests anys l'art del Pixel Art ja havia evolucionat més i cada vegada els dissenyadors s'atreuen a crear il·lustracions més diferents mai abans vistes i començaven a experimentar. A més a més, les franquícies més importants dels videojocs estaven interessades en què els seus personatges fossin molt més reconeixibles. Gràcies a això va augmentar la qualitat de la imatge i la de les il·lustracions per tal de formar imatges més nítides.

Una altra raó que va causar en la millora i evolució del Pixel Art va ser la necessitat d'atraure nous jugadors, nou públic a tot aquest món. Això va portar a mons digitals molt més atractius i detallats i van replicar escenes cinematogràfiques per tal de recordar a pel·lícules.

Les consoles més destacades d'aquesta generació són considerades com el naixement dels videojocs en la seva esplendor.

Nintendo Entertainment System (NES)



Imatge 6. Consola Nintendo Entertainment System (NES). *Imatge extreta de <https://www.amazon.com/>*



Game Boy

Imatge 7. Consola Game Boy. *Imatge extreta de https://es.wikipedia.org/wiki/Game_Boy*

L'era dels 16 bits (1987 - 1993)

Després de tota la revolució que va suposar l'era dels 8 bits els videojocs van evolucionar a un ritme vertiginós. Tant va ser aquest avanç que les consoles van començar a poder competir amb les sales de videojocs amb màquines recreatives.

Els creadors de videojocs van refinar la seva tècnica de Pixel Art fins al punt el qual van començar a allunyar-se de l'estètic arcade i van començar a crear els seus mons. Fins i tot van intentar combinar el Pixel Art amb el modelatge 3D.

Alguns videojocs són:

Super Metroid, Nintendo

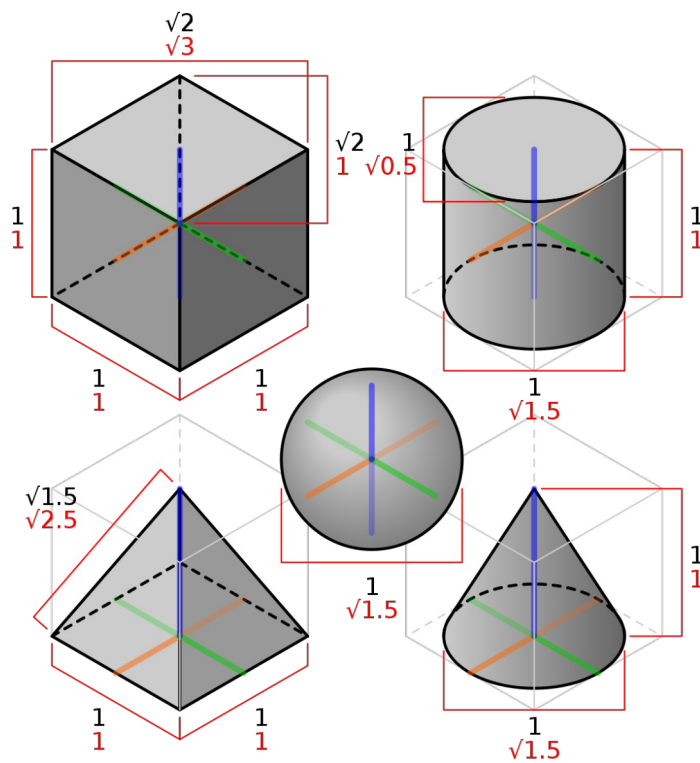


Imatge 8. Captura del videojoc *Super Metroid*, Nintendo. Imatge extreta de https://www.youtube.com/watch?v=0I34aHGb2_8

3.6.2 Tècnica

Si la imatge és 2D, simplement es pinta pixel per pixel d'una manera normal, en canvi, si la imatge és 3D, hem d'utilitzar la perspectiva isomètrica per donar aquest efecte de tres dimensions.

Una perspectiva isomètrica consisteix en una representació visual d'un objecte tridimensional representat en dues dimensions. En aquesta representació els tres eixos ortogonals en projectar-se formen angles de 120 graus.



Imatge 9. Perspectiva isomètrica. *Imatge extreta de*
https://es.wikipedia.org/wiki/Proyecci%C3%B3n_isom%C3%A9trica

Les il·lustracions creades amb aquesta tècnica solen començar amb una base, una imatge preestablerta denominada "line art" que constitueixen les línies bàsiques que defineixen el personatge o dibuix. Aquestes línies solen ser dibuixades primer a mà i més tard escanejades i importades al dibuix.

3.6.3 Categories

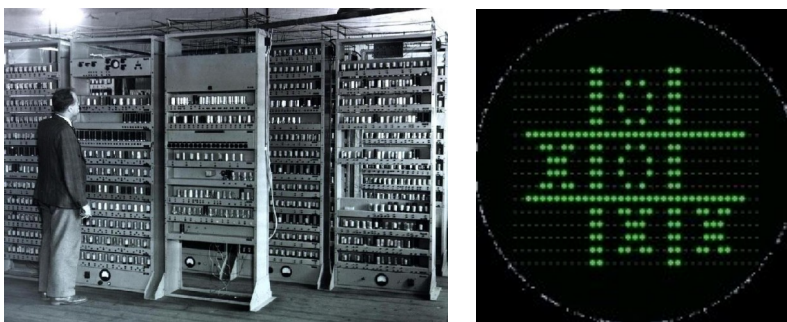
El Pixel Art es divideix en dues grans categories, l'isomètric i el no isomètric. L'estil isomètric és dibuixat en una projecció dimètrica (que serveix per representar volums) i és molt útil en l'entorn dels videojocs per produir una sensació de tres dimensions.

Tècnicament en una perspectiva isomètrica els angles haurien de ser de 30 graus des de l'horitzontal, però en pixel art no queda bé aquest pendent. Per solucionar-ho s'utilitza un pendent que consisteix a fer 1 píxel cap al costat, 2 cap amunt consecutivament formant així un angle de 26.565 graus.

3.7 Història dels videojocs

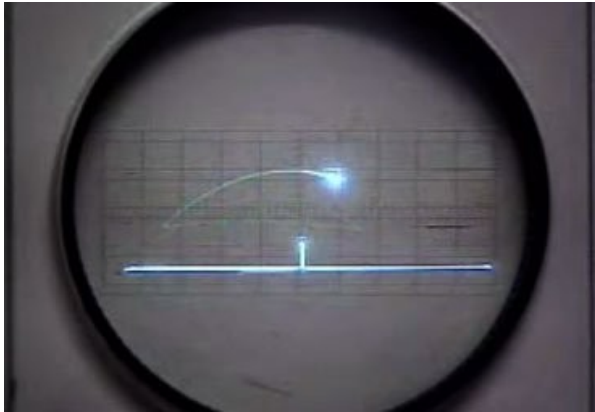
Els inicis

Des de l'inici dels videojocs mai es va tenir clar quin va ser l'origen, el primer videojoc que va iniciar una de les indústries més grans actualment. Malgrat això es creu que el primer videojoc oficial va ser Naught and Crosses o OXO, en altres paraules, el tres en ratlla. Consistia en el popular joc conegut com a tres en ratlla, es jugava en l'ordinador britànic conegut com a EDSAC, el jugador jugava contra la màquina.



Imatge 10 i 11. Ordinador EDSAC i imatge del tres en ratlla. *Imatges extretes de:*
<https://ca.wikipedia.org/wiki/EDSAC>
<https://www.fib.upc.edu/>

Més tard, el 1958 William Higginbotham va crear el videojoc Tennis for Two que consistia en una partida de tennis minimalista per dues persones.



Aquest videojoc va ser creat amb un programa de càlcul de trajectòries i un oscil·loscopi amb la finalitat d'entretenir els visitants del Brookhaven National Laboratory. Aquest videojoc va ser tot un abans perquè va ser el primer a implementar el PVP (player versus player), és a dir, una persona contra una altra.

Imatge 12. Imatge del videojoc Tennis for Two. *Imatge extreta de <https://proyectoidis.org/tennis-for-two/>*

Va passar el temps i quatre anys més tard un estudiant de l'Institut de Tecnologia de Massachussets anomenat Steve Russell va crear Spacewar. Un videojoc multijugador on dues persones s'enfrontaven en un combat entre dues naus espacials. Va tenir molt d'èxit en l'àmbit universitari, però mai va arribar a un públic fora d'aquest entorn universitari.

Malgrat tots els videojocs que hem vist fins ara cap d'ells podia ser jugat en una consola domèstica. El primer només es podia jugar en l'ordinador en el qual va estar creat, el segon només es podia jugar en l'exposició i l'últim es jugava en una màquina PDP-1.



Imatge 13. Imatge del DEC PDP-1, el primer ordinador del món on es va jugar al primer videojoc del món. *Imatge extreta de <https://gamersup.blogspot.com/>*

Màquina PDP-1

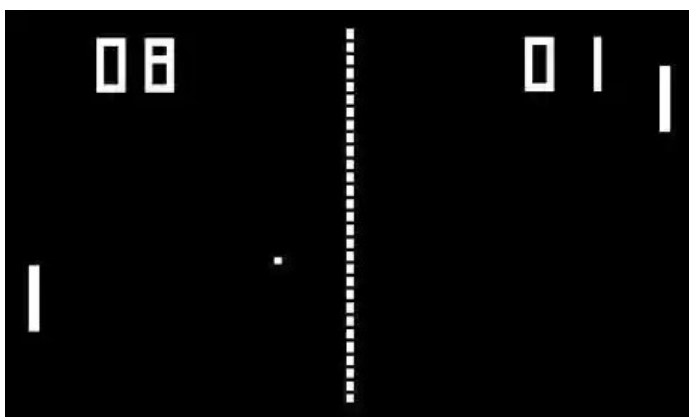
Després de veure les imatges es pot apreciar que els videojocs encara estaven lluny del que tenim avui dia, els quals podem jugar en ordinadors, consoles, mòbils, tauletes i fins i tot calculadores. No va ser fins al 1972 on Ralph Baer, Albert Maricon i Ted Dabney van llançar la primera consola domèstica, la Magnavox Odyssey.



Imatge 14. Imatge de la consola *Magnavox Odyssey*. *Imatge extreta de <https://www.vidaextra.com/>*

L'eclosió dels videojocs (1970 - 1979)

Durant aquesta etapa hi va haver molts nous videojocs i persones que es van interessar per aquesta nova indústria. Dos d'ells van ser Nolan Bushnell i Allan Alcorn. Nolan Bushnell va aconseguir comercialitzar el videojoc Space War, cal recalcar que no era el joc original sinó, que es tractava d'una versió diferent anomenada Computer Space. Alhora Allan Alcorn va crear el famós videojoc Pong, una versió comercial per a les recreatives basada en el videojoc Tennis for Two creat el 1958 per William Higginbotham.



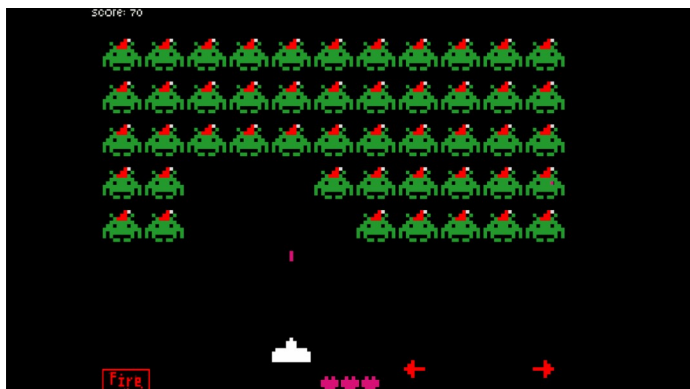
Imatge 15. Imatge del videojoc *Ping Pong*. Imatge extreta de <https://www.theguardian.com/>



Imatge 16. Imatge del videojoc *Computer Space*. Imatge extreta de <https://twitter.com/>

Pong va ser presentat el 1972 i va significar un abans i un després a la indústria dels videojocs. Durant els següents anys hi va haver molts avenços tècnics que permetran el desenvolupament i la millora de les consoles, els més significatius van ser els microprocessadors i els xips de memòria.

A més a més cal recalcar l'aparició de dos videojocs molt importants pels salons recreatius, Space Invaders i Asteroids.



Imatge 17. Captura del videojoc *Space Invaders*. Imatge extreta de <https://mundowin.com/>



Imatge 18. Captura del videojoc *Asteroids*. Imatge extreta de <https://www.pinterest.es/>

La dècada dels 8 bits (1980 - 1989)

A principis dels anys vuitanta va augmentar considerablement la popularitat dels salons de màquines recreatives, aquestes sales van aconseguir atreure un gran nombre de persones donant així a conèixer el món dels videojocs a un públic molt més gran. A més a més, les consoles també van patir un gran augment en la seva popularitat. Tot això va resultar en el creixement de la indústria dels videojocs, fent així que aquesta es desenvolupés molt més, malgrat tot això encara hi estava molt lluny de com la coneixem avui dia.



Imatge 19. Imatge de la consola *Odyssey 2*. Imatge extreta de <https://es.m.wikipedia.org/>



Imatge 20. Imatge de la consola *Intellivision*. Imatge extreta de <https://www.retrogamer.net/>



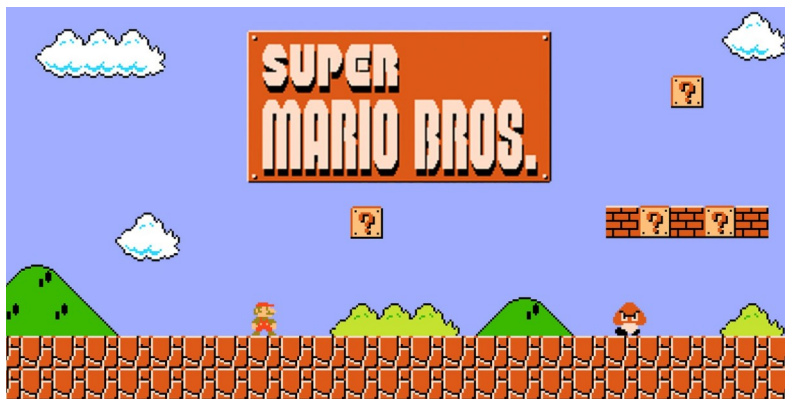
Aquestes van ser les tres consoles més populars d'aquesta època. Van ser consoles molt destacades en la indústria, però, no eren les úniques. També van destacar Colecovision (Coleco) i Turbografx (NEC). És molt important tenir-les present i conèixer-les perquè van ser l'inici del mercat mundial de consoles. Mercat que avui està dividit en dos, Xbox i PlayStation. Aquestes consoles són els seus predecessors.

Imatge 21. Imatge de la consola Atari 5200. Imatge exteta de <https://es.wikipedia.org>

Malgrat tot això, no tot eren beneficis i creixement, el 1983 va començar la crisi dels videojocs, una crisi principalment als Estats Units i Canadà que va durar dos anys i va suposar la pèrdua de 4000 milions de dòlars. Aquesta crisi va ser causada per la saturació dels mercats en el nombre de consoles i videojocs, i en el creixent interès en els ordinadors personals. Els ingressos anuals van passar dels 3200 milions de dòlars als 100 milions a la crisi, una pèrdua del 97%. La crisi va acabar gràcies a la segona generació de videojocs que va sortir i també degut a la popularitat que van perdre els salons recreatius, fent així que la gent fos a comprar més consoles domèstiques.

Per altra banda, Japó en comptes d'apostar pels ordinadors personals va apostar per les consoles, concretament per la Famicom, coneguda a Europa com la Nintendo Entertainment System que va sortir al mercat l'any 1983. A tot això Europa estava centrada en els microordinadors com el Commodore 64 o el famós Spectrum.

Després de la crisi als Estats Units i Canadà va triomfar la NES, va passar a ser la consola de videojocs per excel·lència amb llocs molt famosos com és el cas del Tetris. Un videojoc que consisteix a anar fent files amb figures geomètriques sense que aquestes arribin a dalt de la pantalla. Cal recalcar la sortida del videojoc Super Mario Bros l'any 1985, va ser un pioner en la indústria. Aquest videojoc proposava una història, proposava un inici, una meta i un final. Al contrari que tots els llocs que havien aparegut fins al moment els quals només consistien en pantalles repetitives on l'únic objectiu era aconseguir més puntuació.



Imatge 22. Imatge del videojoc *Super Mario Bros.* Imatge extreta de <https://www.nintendo.es/>

Super Mario Bros

Per últim cal mencionar la sortida de la consola portàtil Game Boy el 1989, va ser un èxit mundial i va suposar l'evolució definitiva dels portàtils com a plataformes de videojocs.

La revolució de les 3D (1990 - 1999)

A causa de la competència de l'anomenada "generació de 16 bits", que incloïa el Mega Drive, el Super Nintendo Entertainment de Nintendo, el PC Engine de NEC i el CPS Changer de Capcom, les consoles de videojocs van patir un gran salt tecnològic a partir dels anys noranta.

Juntament, va aparèixer la Neo Geo (SNK), una consola que coincidia amb Arcade en termes de capacitat tecnològica, però que era massa cara per al jugador mitjà de videojocs.

A causa de les noves capacitats tècniques, aquesta generació va experimentar un considerable creixement del nombre de jugadors i la introducció de noves tecnologies com el propi CD-ROM.

Durant aquest temps, algunes empreses van començar a treballar en videojocs amb entorns tridimensionals, especialment en el camp de la informàtica, i van obtenir resultats com "2 i mig" de Doom i el 3D complet.

Per a les consoles de 16 bits més antigues, la victòria més gran i final va ser per a l'SNES (Super Nintendo Entertainment System) gràcies a la tecnologia 3D de SGI, la màxima expressió de la qual van ser jocs com Donkey Kong Country o Killer Instinct. També apareix el primer joc de consola poligonal, la competició de SNES, MegaDrive, ha llançat Virtual Racing, tot un èxit i marca l'abans i el posterior èxit dels jocs en 3D.

Ràpidament, els videojocs 3D van prendre una posició dominant en el mercat, en gran part gràcies a l'anomenada "generació de 32 bits" de la qual es va destacar Sony PlayStation, Sega Saturn (que només va ser prominent al Japó i no va tenir molt èxit a la resta del món) i la "generació de 6 bits" on Nintendo va tornar a estar al cim de mercat amb la Nintendo 64.

La consola de Sony va néixer d'un projecte iniciat amb Nintendo (anomenat SNES PlayStation) que incloïa un perifèric per SNES amb reproductor de CD. Finalment, Nintendo va rebutjar la proposta de Sony perquè Sega havia desenvolupat una cosa similar, però va fallar, i Sony més tard va inventar la PlayStation, que va tenir èxit en el mercat.

Quant als recreatius, van començar una lenta, això no obstant, imparabile retirada a mesura que augmentaven els accessos a les consoles i els ordinadors més potents. En canvi, els videojocs portàtils, producte de les noves tecnologies més poderoses, van començar el seu veritable negoci. Es van unir a la Game Boy màquines com la Game Gear (Sega), Linx (Atari) o la Neo Geo Pocket (SNK), encara que cap d'elles va poder reprimir el boom de la Game Boy i els seus descendents (Game Boy Pocket, Game Boy Color, Game Boy Advance, Game Boy Advance SP).

Fins a finals de la dècada dels noranta, la consola més popular era la PlayStation amb jocs com Final Fantasy VII (Square), Resident Evil (Capcom), Winning Eleven 4 (Konami), Gran Turismo (Polyphony Digital) i Metal Gear Solid (Konami).

La revolució de les 3D (1990 - 1999)

Sony va deixar de fabricar la PlayStation 2 al 2000 i Sega va llançar al mercat una consola anomenada Dreamcast Drivers 2000 Sèries CX-1 amb les mateixes característiques tècniques que la Dreamcast. El que la diferenciava era que incorporava un televisor de 14 polzades, un teclat, altaveus i els comandaments, és a dir, venia amb els "drivers" d'aquí el nom. A més a més, el 2001 Microsoft va entrar a la indústria de les consoles amb la Xbox.

Nintendo va tornar al mercat de les consoles amb la Gamecube i la Game Boy Advance. Per una altra banda, Sega, el 2000, va veure que no podia competir i va deixar de produir hardware, va passar a dedicar-se només al hardware.

Malgrat l'esforç de totes aquestes companyies la plataforma que sempre ha estat la més potent és l'ordinador personal PC. Això és degut al fet que permet molta més flexibilitat i pot ajustar el preu i els seus components a les teves necessitats. A més a més, no l'has de canviar tan periòdicament com les consoles sinó que pots millorar només parts concretes, peça a peça.

4. CREACIÓ DEL VIDEOJOC

Una vegada que vaig fer tota la recerca necessària em tocava dissenyar el meu videojoc. Les fases que vaig seguir en el desenvolupament del meu videojoc van ser:

1. Brain Storm: Aquí és on vaig fer una pluja d'idees pel meu videojoc de les quals va sortir el videojoc final.
2. Disseny: Aquí és on vaig dissenyar tots els sprites i animacions que tindria el meu videojoc.
3. Programació de les mecàniques bàsiques: En aquest apartat entra tota la programació relacionada amb el moviment bàsic del jugador.
4. Creació del primer nivell beta: En aquesta part vaig dissenyar el primer nivell per comprovar la viabilitat del meu videojoc.
5. Creació de mecàniques més avançades
6. Creació dels nivells
7. Menú
8. Retocs

Com qualsevol projecte aquest procés no va ser lineal, és a dir, vaig haver de tornar enrere alguns cops per fer que les coses funcionessin o per canviar aspectes que no quadraven al meu videojoc. Això és una situació en la qual es troben moltes empreses dissenyadores de videojocs, per exemple, programen a un personatge per poder volar, però animació feta per l'equip de disseny no és adaptable al videojoc i els programadors els hi demanen una nova versió, fan un pas enrere. En el meu cas, com només soc una persona aquests errors no són tan freqüents, ja que la mateixa persona, és a dir jo, té a la seva ment tots els processos pels quals ha de passar i ja sap com reaccionarà el videojoc envers alguns dissenys o característiques. Malgrat tot això, sí que em vaig descobrir en algunes situacions on vaig haver de fer passos enrere.

4.1 Brainstorm

Aquesta etapa va ser en la qual vaig haver de pensar la temàtica i com faria el meu videojoc. Al principi tenia moltes idees i molt variades, tenia molts projectes en ment, malgrat que totes les idees eren molt diferents totes tenien característiques comunes:

1. Eren videojocs 2D. El fet de tractar-se d'un videojoc 2D reduïa la dificultat del modelatge i disseny de personatges considerablement. Ja que al no tenir eines preprogramades com a Roblox Studio, Rpg Maker..., havia de crear els personatges des de zero. Aquests programes et faciliten una sèrie de models que tu només has de retocar, en canvi, utilitzant Godot Engine els havia de crear des de zero. Òbviament, això té avantatges i desavantatges, per una part el procés és molt més complicat des de zero, però, per una altra banda, em dona més llibertat i creativitat perquè no m'he d'ajustar a cap model prefabricat. Una altra raó que em va portar a decantar-me per l'estil 2D és el record que tinc dels videojocs de la meva infància. Com la majoria dels nens i nenes de la meva edat vaig créixer jugant videojocs principalment a la Nintendo on tots eren 2D. Va ser això el que em va portar a decidir-me finalment pel 2D, fer un videojoc com els que jo jugava quan era petit.

2. Eren en l'estil Pixel Art. És una tècnica que consisteix en l'edició de les imatges pixel a pixel. Vaig escollir aquest estil perquè era un tipus d'art que des de sempre m'havia apassionat, però mai vaig tenir l'oportunitat de provar. Amb aquest treball de recerca vaig poder iniciar-me en el món del pixel art i vaig poder crear els meus primers dissenys.

Les dues idees principals que vaig tenir van ser, un RPG 2D o una rèplica d'uns dels meus videojocs de la meva infància anomenat Geometry Dash.

La primera idea consistia a crear un RPG, és a dir, un videojoc de rol on el jugador es troba en la situació d'un personatge fictici i ha de seguir una història. Un gran exemple d'aquest gènere conegut universalment és la saga Pokémon, on el jugador ha de seguir la història d'un entrenador que aspira a capturar tots els pokémons i a guanyar la lliga. La meua idea tractava sobre un món fantasiós, inspirat en l'edat mitjana, on el jugador era un cavaller que havia de passar per moltes aventures per tal de derrotar els seus majors enemics en batalles. Tot això mentre passava per "dungeons" i trampes que posarien a prova la seva habilitat.

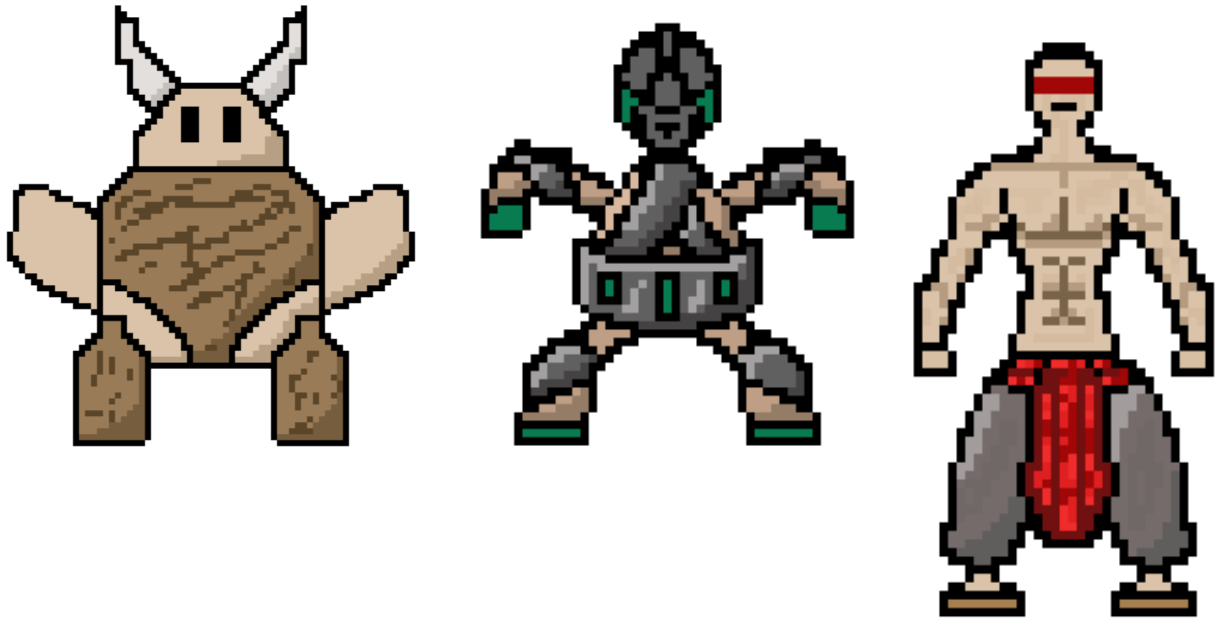


Aquest va ser el primer personatge que vaig fer del primer videojoc. Aquest seria el protagonista, un cavaller amb espasa i armadura. A més a més vaig crear tot un seguit d'animacions pel moviment i atac del jugador (aquestes animacions hi seran a l'Annex).

El personatge podia atacar amb l'espasa i també podia canviar d'arma depenent a quina part es trobés del videojoc.

Imatge 23. Imatge del protagonista d'un dels videojocs que tenia pensats. *Autor propi.*

A partir d'aquí vaig crear més personatges, vaig crear diversos enemics als quals el jugador havia de vèncer. Són els següents:

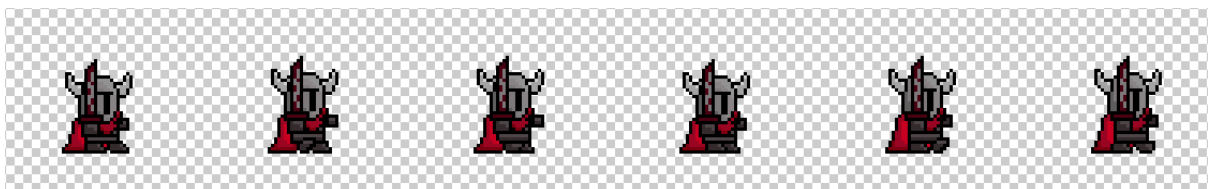


Imatges 24, 25 i 26. Imatges dels enemics que vaig crear pel videojoc. *Autor propi.*



Imatge 27. Imatge dels personatges de la primera idea de videojoc amb els seus perfils. *Autor propi.*

Vaig crear aquestes imatges utilitzant el programa Photoshop a principis de maig
Una vegada creat els personatges els vaig animar:



Imatge 28. Imatge dels sprites de l'animació del protagonista caminant a la dreta. *Autor propi.*



Imatge 29. Imatge dels sprites de l'animació del protagonista atacant cap a la dreta. *Autor propi.*



Imatge 30. Imatge dels sprites de l'animació del protagonista fent una voltereta cap a la dreta. *Autor propi.*



Imatge 31. Imatge dels sprites de l'animació d'un dels enemics caminant a la dreta. *Autor propi.*



Imatge 32. Imatge dels sprites de l'animació d'un dels enemics caminant a la dreta. *Autor propi.*

Una vegada tenia les animacions dels personatges vaig crear sprites dels objectes i de l'entorn de les escenes.



Imatges 33 i 34. Imatges de dos arbres que vaig dissenyar. *Autor propi.*

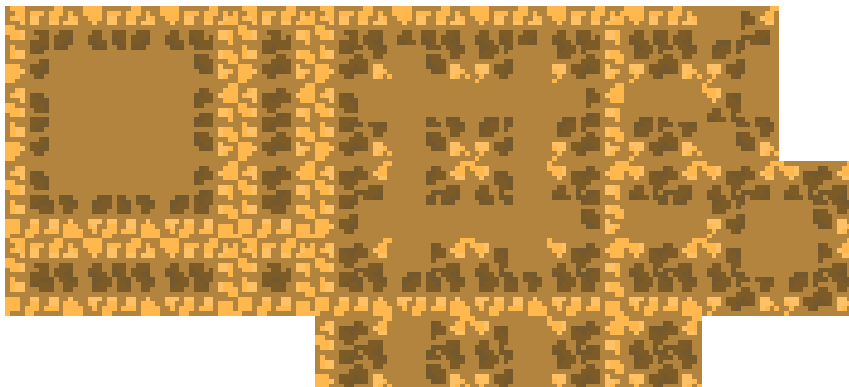


Imatge 35. Imatge d'un matoll que vaig dissenyar. *Autor propi.*



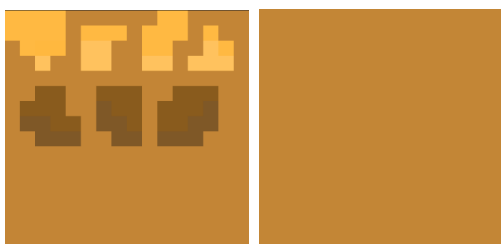
Imatge 36. Imatge del murs i les portes que vaig fer per les dungeons. *Autor propi.*

Una vegada vaig fer tots els sprites vaig crear el que s'anomena TileMaps. Els TileMaps són com el seu nom indica, mapes, on se seleccionen i s'ordenen un conjunt de sprites (imatges) i s'indica com s'hi han de col·locar. Per exemple:



Imatge 37. Imatge del TileMap de les coves. *Autor propi,*

Aquest TileMap està compost per dues imatges:



Imatges 38 i 39. Imatges dels dos sprites que conformen el TileMap.

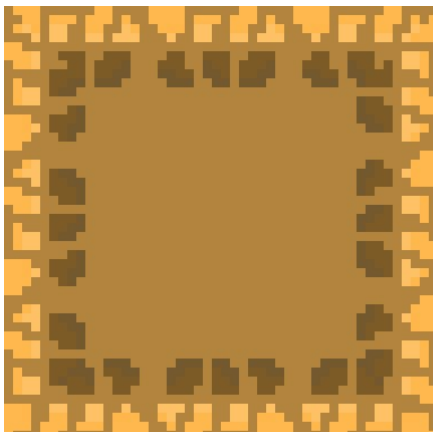
El TileMap està compost d'aquestes dues imatges, les altres són només rotacions i fusions que el programa s'encarrega de crear a partir de les dues originals. Estan col·locats d'aquesta manera amb un sentit, no és a l'atzar, aquest ordre permet al programa, en aquest cas Godot Engine, saber quin sprite hi ha de col·locar quan. Permetent dibuixar nivells d'una manera mil vegades més senzilla que col·locant un a un. Per exemple:

Si col·loquem aquest sprite:



Imatge 40. Imatge del sprite base del TileMap. *Autor propi.*

El programa sabrà que li falten 8 sprites als costats, quatre als costats i quatre als vèrtexs. D'aquesta manera:



Imatge 41. Imatge de com corregiria el programa en base del TileMap. *Autor propi.*

Llavors vaig crear tots els TileMaps necessaris. Cal recalcar que encara era el començament de la idea, eren els primers conceptes del videojoc.

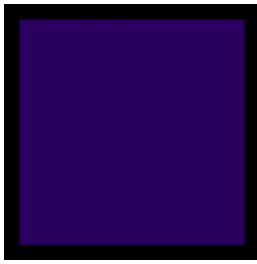
Malgrat tot l'esforç que vaig fer en aquesta etapa no vaig poder continuar amb aquesta idea a causa de la complexitat del desenvolupament del videojoc. Una vegada vaig començar a implementar les mecàniques al videojoc em vaig adonar que era molt complex per un treball de recerca, és a dir, no era viable en el meu marge de temps. Llavors vaig continuar amb la segona idea del Brain Storm.

La segona idea consisteix en la recreació d'un dels videojocs de la meua infància anomenat Geometry Dash. Geometry Dash és un videojoc creat per Robert Topala, també conegut com a Rob Top. El videojoc consisteix en un personatge que ha de superar nivells saltant.

El joc comença amb el personatge constantment movent-se cap a la dreta, mentre es mou ha d'anar esquivant obstacles sense morir per tal d'arribar al final del nivell. La idea del videojoc és fer diversos nivells cada vegada més complicats fins a arribar a l'últim el qual sigui molt difícil, obligant al jugador no tan sols a jugar més, sinó a millorar el seu joc.

Els sprites d'aquest projecte són molt més senzills que els de la idea anterior, ja que als videojocs arcade es valoren molt més les mecàniques, és a dir el moviment del jugador i com pot aquest millorar el seu nivell, que la qualitat visual.

El jugador consisteix en un sprite elaborat en Photoshop molt senzill:



Aquesta és la imatge base del jugador, tanmateix, a mesura que passa pantalles del videojoc i va guanyant desbloqueja nous personatges. D'aquesta manera el jugador pot escollir el que més li agrada.

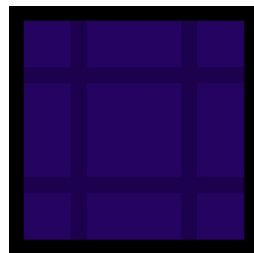
Imatge 42. Imatge del protagonista base.

Autor propi.

Una vegada que tenia el protagonista calia dissenyar l'entorn:



Imatge 43. Imatge dels pinxos que maten al jugador. *Autor propi.*



Imatge 44. Imatge del terra i els murs del videojoc. *Autor propi.*

Aquests van ser els primers models del videojoc, els pinxos maten al jugador si els toca i la terra serveix de plataforma per al jugador per poder saltar.

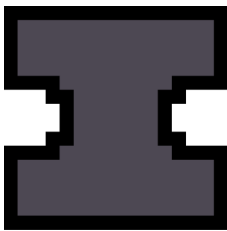
Una vegada que tenia aquests conceptes bàsics vaig poder comprovar la viabilitat del videojoc fent un nivell de prova i com aquest va complir les expectatives em vaig decantar per aquesta idea.

4.2 Disseny

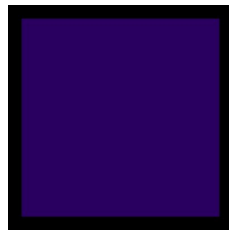
Una vegada ja estava decidida la idea vaig començar a dissenyar tots els sprites que necessitaria.

4.2.1 Sprites del jugador

En aquest apartat vaig dissenyar tots els diferents models de personatge que hi haurien disponibles al videojoc.

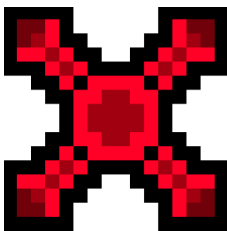


Imatge 45. Imatge d'un dels primers models del videojoc. *Autor propi.*

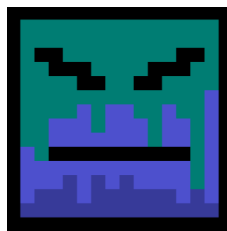


Imatge 46. Imatge d'un dels primers models del videojoc. *Autor propi.*

Després vaig dissenyar més models alternatius:



Imatge 47. Imatge d'un disseny de personatge. *Autor propi.*



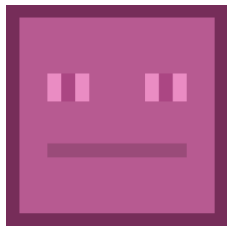
Imatge 48. Imatge d'un disseny de personatge. *Autor propi.*



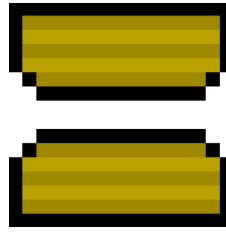
Imatge 49. Imatge d'un disseny de personatge. *Autor propi.*



Imatge 50. Imatge d'un disseny de personatge. *Autor propi.*



Imatge 51. Imatge d'un disseny de personatge. *Autor propi.*



Imatge 52. Imatge d'un disseny de personatge. *Autor propi.*

Aquests últims 6 sprites són els sprites que vas aconseguir a mesura que passes nivells. També vaig dissenyar dos sprites per una variació del nivell on el jugador comença a volar i canvia a una nau.

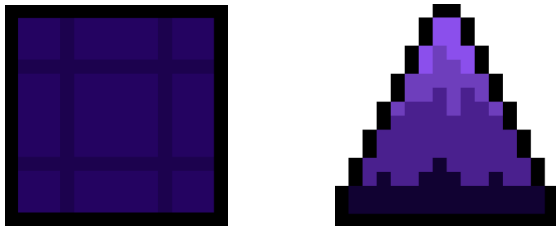


Imatge 53 i 54. Imatge dels dos models de coet del videojoc. *Autor propi.*

4.2.2 Sprites dels nivells

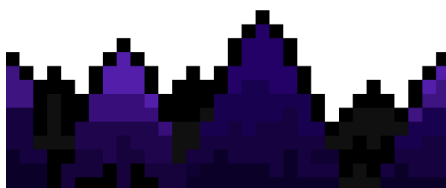
Una vegada vaig fer els sprites del personatge vaig començar a fer els sprites dels nivells, plataformes, trampes, etc.

Són aquests:



Imatge 53 i 54. Imatges de la plataforma i dels pinxos.

Autor propi.

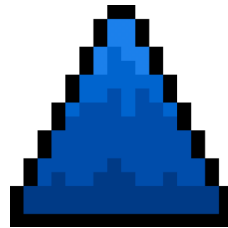
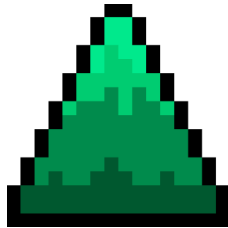
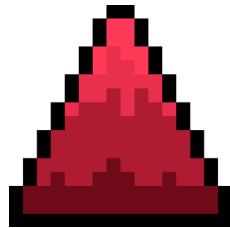
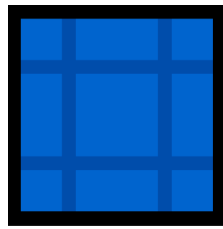
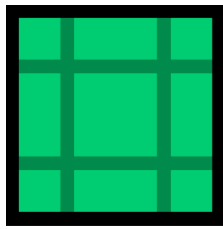
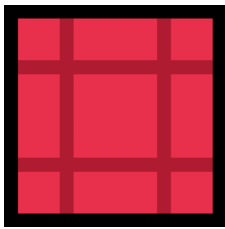


Imatge 55. Imatge dels pinxos de fons.

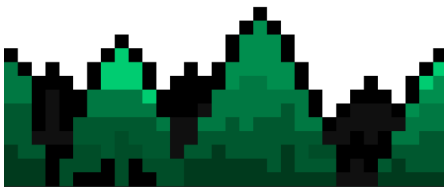
Autor propi.

Aquests són els 3 sprites bàsics, d'aquests dissenys vaig treure diferents variacions per diferents nivells.

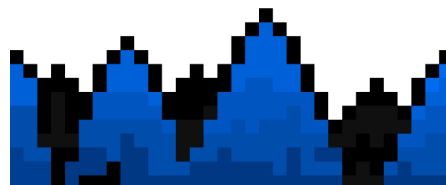
Variacions:



Imatges 56, 57, 58, 59, 60 i 61. Imatges de les variacions de les plataformes i dels pinxos. *Autor propi.*



Imatge 62, 63 i 64. Imatges de les dels pinxos del fons. *Autor propi.*



variacions

4.3 Programació de les mecàniques bàsiques

En aquesta part vaig elaborar el primer nivell i vaig implementar totes les mecàniques bàsiques al videojoc. En aquesta part era vital aconseguir el funcionament bàsic del videojoc, és a dir, que funcionin les físiques del videojoc, el moviment en l'eix x i en l'eix y. Que el personatge pogués morir i que el personatge pogués saltar.

4.3.1 Moviment constant en l'eix x

La primera mecànica més bàsica que necessitava programar per tal que funcionés el videojoc era mantenir el personatge en constant moviment cap a la dreta en l'eix x.

Això va ser una tasca molt fàcil, vaig implementar els sprites del personatge al videojoc, vaig incorporar-li una "hitbox", és a dir, una forma que s'integra als sprites per tal que el personatge pugui interactuar amb el seu entorn, pugui tenir col·lisions, es pugui moure..., i per últim vaig crear una càmera 2D que segueix al personatge a mesura que avança. Una vegada vaig tenir tot això i una petita plataforma amb els sprites dels murs fets anteriorment, per tal que tingui un lloc per on moure's i no ser al buit, vaig passar a la part del codi. En aquesta part vaig programar el seu moviment en el llenguatge propi de Godot.

El codi sencer serà a l'annex del treball de recerca amb totes les explicacions i anotacions necessàries perquè tothom ho pugui entendre fins i tot si no tens cap experiència programant.

4.3.2 Salt

Una vegada tenia el moviment del jugador calia donar-li de l'habilitat de saltar per poder esquivar obstacles i continuar sense morir al nivell. No només calia donar-li la possibilitat de saltar sinó que també havia d'establir una gravetat per tal que en saltar no comencés a volar sense tornar a tocar el terra.

Per fer això vaig establir una gravetat i vaig programar una funció que en el moment que detectes que el jugador hi havia polsat la tecla espai, apliqués una força en l'eix y i en sentit oposat a la gravetat i el personatge saltés.

A més a més, un dels problemes que vaig tenir va ser que es podia abusar d'això i fer salts una vegada que ja era en l'aire, per tant, el jugador podia mantenir-se constantment en l'aire sense caure i passar-se tots els nivells. El que vaig fer per solucionar aquesta dificultat va ser mitjançant la detecció de les col·lisions entre la "hitbox" del personatge i la "hitbox" del terra detectar si el jugador estava tocant terra o no. Una vegada que tenia això vaig implementar una condició per tal que només si aquesta condició es complia, si tocava el terra, pogués saltar. Amb tot això vaig solucionar el problema.

El codi de tot aquest procés serà a l'annex.

4.3.3 Mort

Una vegada que el personatge ja es podia moure i podia saltar calia fer que pogués morir, això va ser potser el procés més simple de tota la creació del videojoc. Simplement, vaig afegir els sprites dels pinxos al motor gràfic, vaig posar-los unes "hitbox", unes col·lisions, vaig crear una àrea 2D per detectar quan el jugador entrava en la "hitbox" i vaig programar una funció que fa que quan el jugador toqui un, la pantalla del videojoc es reinici.

A més a més, com al videojoc t'has de moure constantment cap a la dreta en el cas que el jugador trobi alguna manera de parar el seu personatge vaig fer que si el videojoc detecta que deixa d'avançar també el mati.

Una vegada vaig fer tot això només em calia repetir el procés fet pels pinxos a tots els elements possibles amb els quals el jugador, pot morir, vaig afegir sprites, hitbox, àrees 2D i vaig programar les funcions.

El codi de tot aquest procés serà a l'annex.

4.4 Creació del primer nivell beta

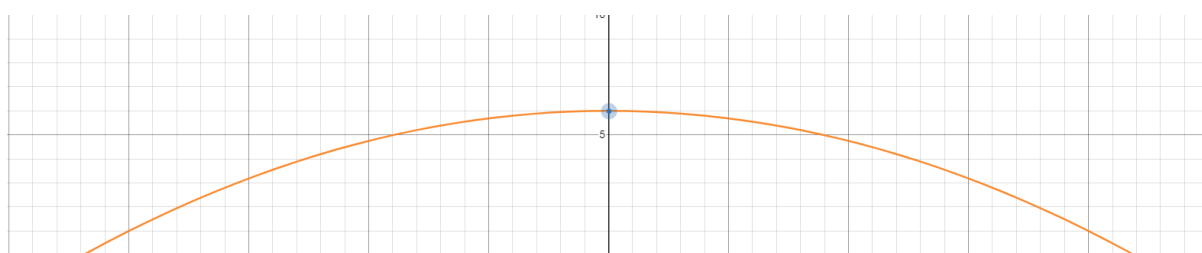
Una vegada tenia ja totes les mecàniques bàsiques programades em tocava crear el primer nivell del videojoc, el nivell beta, el nivell que em diria si el meu videojoc era viable.

El procés que vaig seguir va ser el següent:

1. Creació del mapa
2. Implementació del jugador
3. Ajustos de viabilitat

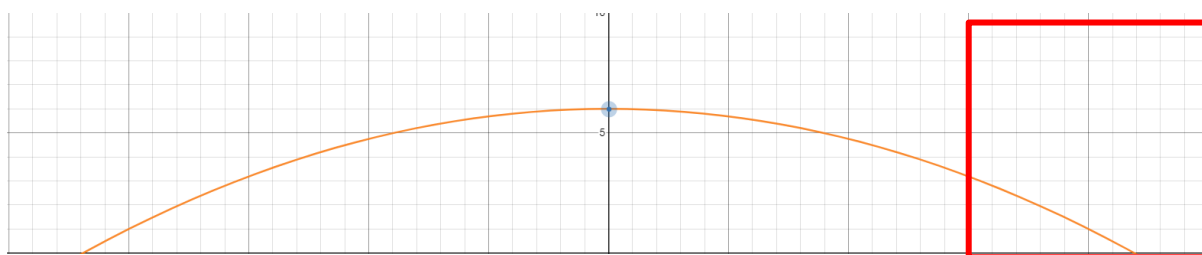
Primer de tot vaig crear el mapa del nivell, les plataformes, els espais, les trampes, els pinxos, etc. Al principi de tot vaig començar només amb les plataformes per intentar veure com és el moviment del meu personatge, vaig implementar el jugador al motor gràfic i ho vaig provar. És aquí on vaig trobar un dels problemes més grans de tot el desenvolupament del videojoc, el moviment no se sentia natural, no era un moviment natural pel jugador, ni predictable per poder esquivar ni s'ajustava bé a la manera en la qual el videojoc estava pensat.

És a dir, jo havia establert una gravetat que jo creia correcte pel meu videojoc, també havia establert una velocitat en l'eix horitzontal i una altra en l'eix vertical en saltar. La velocitat en l'eix horitzontal s'encarrega de moure el personatge cap a la dreta, la gravetat de mantenir-lo a terra i la velocitat a l'eix vertical de permetre al jugador saltar polsant la tecla espai. El problema ve perquè no eren valors que es relacionessin bé entre ells, és a dir, potser tenia una velocitat en l'eix horitzontal molt gran, una gravetat molt gran i una capacitat per saltar molt petita, això comportava salts d'aquest tipus:



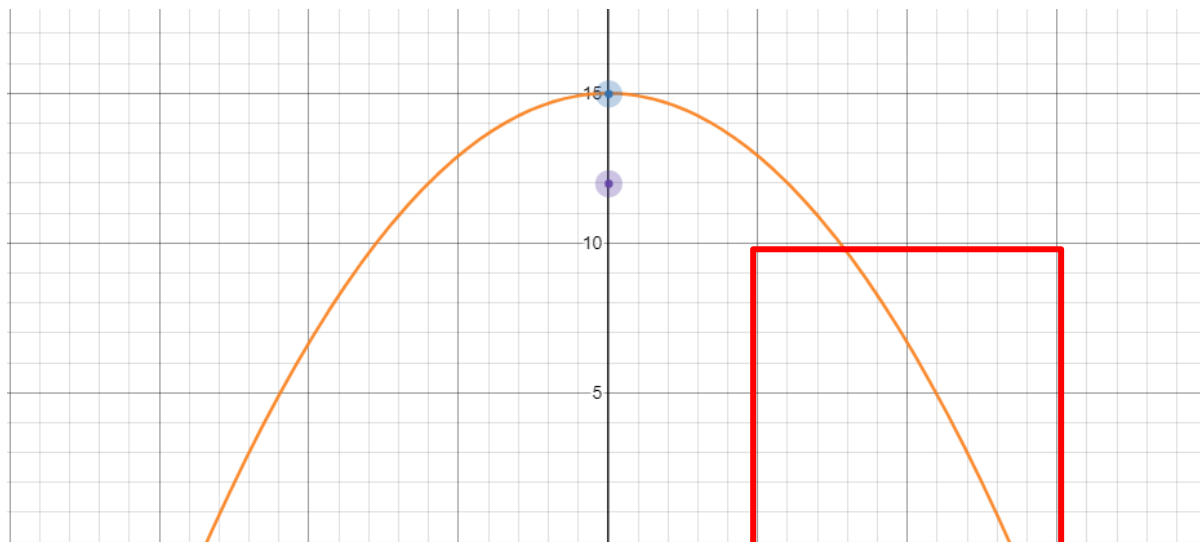
Imatge 65. Paràbola descrita pel jugador al saltar. *Autor propi.*

El problema amb aquests tipus de salts és que les plataformes tenen una altura equivalent a dos quadrats grans de la gràfica. El quadrat en vermell representa una plataforma i la paràbola la trajectòria que descriu el personatge en saltar. Això comportava que el jugador mai pogués arribar a la plataforma.



Imatge 66. Paràbola descrita pel jugador comparada amb l'altura d'un bloc. *Autor propi.*

En canvi, si augmentem la força amb la qual salta o disminuïm la força de la gravetat els salts canvien i passen a ser així:



Imatge 67. Paràbola descrita pel jugador comparada amb l'altura d'un bloc. *Autor propi.*

El problema que tenen aquests salts és que tenen una paràbola molt pronunciada, pugen molt i es desplacen molt poc en l'eix horitzontal, això fa que el videojoc perdi velocitat i perdi la seva gràcia. La idea del videojoc és que sigui un videojoc ràpid que necessitis tenir reflexos, aquest salt el fa molt lent i treu tota aquesta part.

Per tant, vaig haver d'anar provant diferents velocitats, gravetats i forces de salt fins a arribar a aconseguir la que per mi era la més natural i la que em permetia poder crear nivells ben adaptats i divertits.

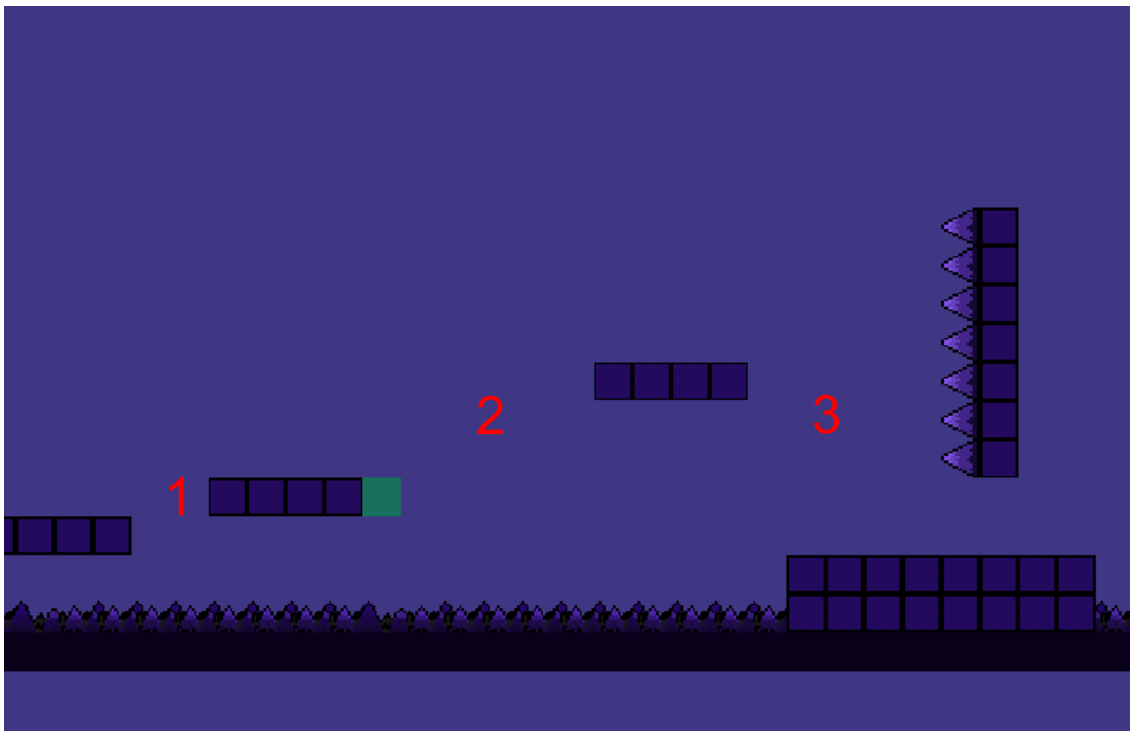
Aquest va ser un procés molt llarg a causa del fet que amb una petita variació de valors el moviment canviava molt, al final vaig assolir el moviment que volia, el codi serà al final, però la velocitat en l'eix horitzontal és 200, la gravetat 1200 i la força amb la qual salta -300. Aquests van ser els valors que em van permetre obtenir el moviment desitjat.

Una vegada vaig fer tot això vaig poder comprovar que el videojoc era possible i viable i vaig poder passar a la creació de les mecàniques més avançades.

4.5 Creació de les mecàniques més avançades

El meu videojoc necessitava mecàniques avançades, mecàniques que donessin dinamisme a l'experiència del jugador.

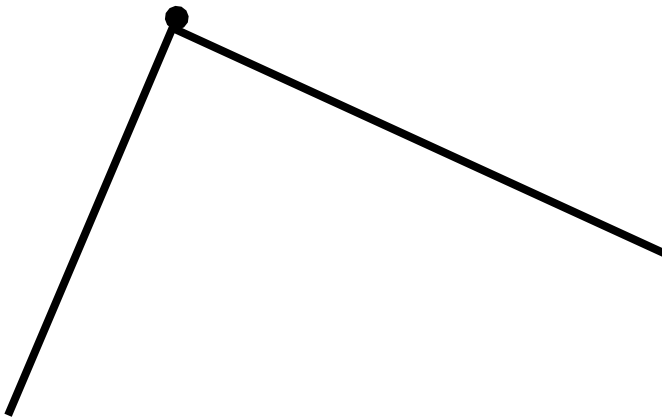
La primera mecànica és la més senzilla, consisteix en una plataforma amb la qual el jugador pot fer salts molt més grans que els que pot fer sense cap mena d'ajuda, això em permet crear obstacles i murs més grans els quals sense l'ajuda d'aquestes plataformes sigui impossible passar. A més a més, el fet que canviï el salt encara que sigui un moment, fa que la persona que juga ho hagi de tenir en compte i no despistar-se. Per exemple:



Imatge 68. Imatge d'una part del nivell on estan marcat els tres salts que ha de fer el jugador. *Autor propi.*

En aquesta part d'un dels nivells el jugador ha de fer el primer salt normal, però, ha de tenir en compte que ha de deixar de saltar abans de tocar l'altra plataforma, si no, se saltarà el quadrat verd que li dona la suficient força per fer el següent salt i morirà. Això és un exemple de com aquesta mecànica fa que el jugador hagi de canviar i estar atent al seu entorn. I per últim, a la tercera plataforma, al tercer salt, ha de saber que no ha de saltar, s'ha de deixar anar, sinó, xocarà amb els pinxos i morirà.

La segona mecànica és la de poder volar, aquesta mecànica consisteix que en arribar a un punt del nivell el jugador canvia de sprite i passa a ser una nau que pot volar. En aquest moment la gravetat desapareix, la velocitat canvia i el moviment es passa a controlar d'una altra manera. Al moment de canviar el personatge puja amb una velocitat gran i amb un angle de 65 graus i en polsar la tecla espai comença a anar cap a baix amb una velocitat vertical una mica menor i amb un angle de 35 graus i en tornar a polsar la tecla espai torna a canviar. Descriu un moviment d'aquest tipus:

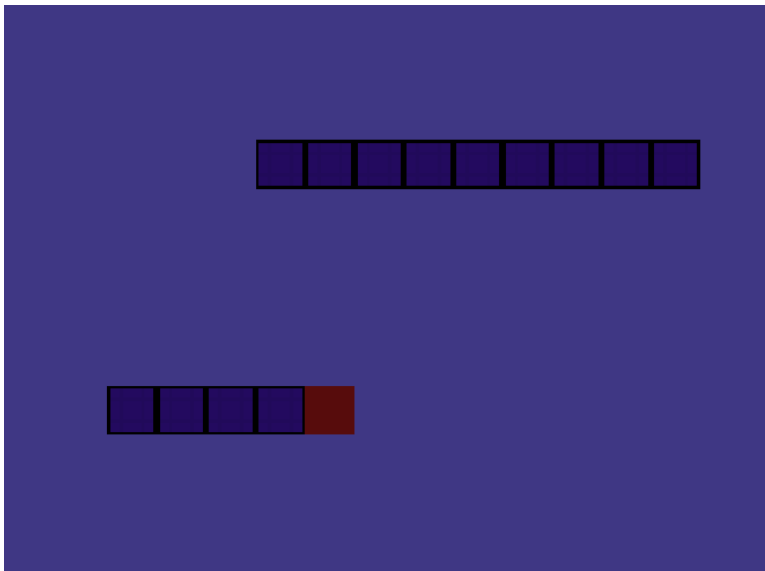


Imatge 69. Trajectòria descrita pel jugador volant. *Autor propi.*

Aquesta mecànica va ser la més difícil de programar, ja que comportava un canvi molt gran en el moviment i vaig haver de superar molts problemes i errors que se'm van presentar.

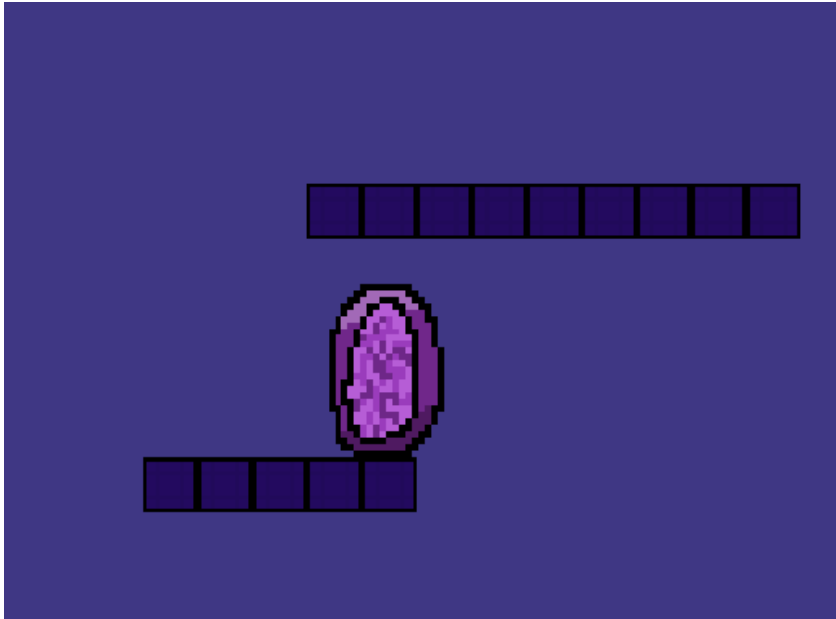
La tercera mecànica consisteix en el jugador invertit, és a dir, en comptes d'anar per terra el jugador camina pel sostre i en saltar en comptes d'anar cap amunt va cap a baix. És la meva mecànica preferida pel fet que si o intercales amb el moviment normal pots fer que la dificultat del videojoc augmenti considerablement. Per programar-ho vaig fer el següent.

Primer de tot necessitava saber en quins moments volia activar això, per tant, havia de tenir una manera d'identificar quan el personatge havia arribat a la part del nivell on s'activaria aquesta mecànica. La solució que vaig trobar a aquest problema va ser crear un quadrat invisible que posaria al nivell i que quan el jugador el toqués, el videojoc pogués saber que havia arribat i s'havia d'activar aquesta mecànica.



Imatge 70. Imatge de l'entitat, quadrat vermell, que canvia al mode invertit. *Autor propi.*

En aquesta imatge podem veure el quadrat que detecta quan el jugador arriba a aquesta part i el fa canviar, de tal manera que passa a caminar pel sostre de la segona plataforma. El quadrat només és visible en el mode d'edició, només el programador ho pot veure, els jugadors no. Els jugadors el que veuen és un sprite d'un portal que per ells és el que fa l'acció de transició entre les mecàniques.

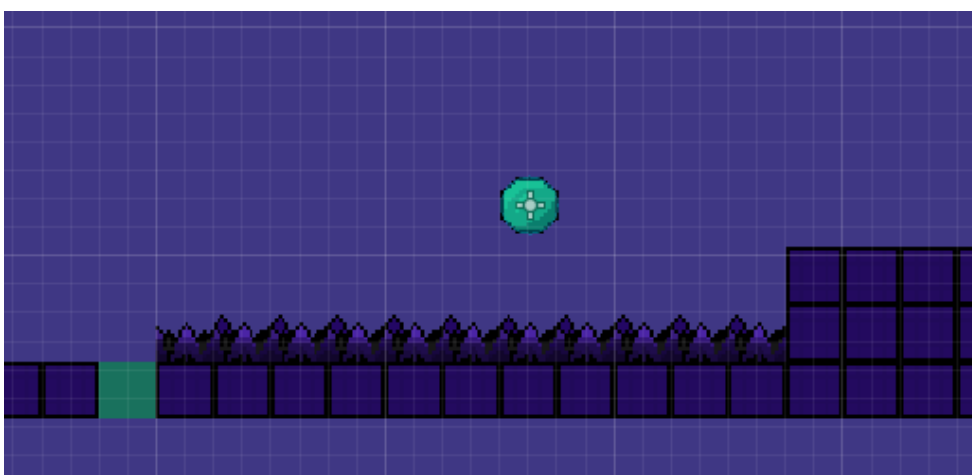


Imatge 71. Imatge del portal que substitueix al quadrat vermell. *Autor propi.*

Aquesta és la imatge real del que veu el jugador al videojoc en comptes del quadrat vermell.

Una vegada que la part invertida termina el jugador torna a passar per un altre portal igual que aquest i torna al seu moviment normal.

L'última mecànica avançada que vaig programar va ser el que s'anomena "Double Jump" o doble salt en català. Consisteix en una esfera que es col·loca en l'aire i que permet al jugador tornar a fer un salt sense la necessitat de ser a terra. És a dir, el jugador salta i una vegada en l'aire quan està passant per davant de l'esfera si prem la tecla espai, pot fer un segon salt en l'aire.



Imatge 72. Imatge de l'esfera encarregada del doble salt. *Autor propi.*

En aquesta imatge el jugador primer salta gràcies a la plataforma creada al principi de les mecàniques bàsiques i una vegada és en l'aire hi ha de pulsar la tecla espai justament en el moment exacte per tal de fer un segon salt en l'aire i evitar caure als pinxos.

Tot el codi serà a l'annex.

4.6 Creació dels nivells

En aquest apartat simplement em vaig dedicar a crear nivells pel videojoc amb les mecàniques que havia creat fins al moment, és important no fer els nivells monòtons i anar canviant tan dificultats i mecàniques.

Imatges del primer nivell:



Imatge 73, 74 i 75. Imatges del primer nivell del videojoc. *Autor propi.*

Aquestes tres imatges juntes formen el primer nivell del videojoc que serveix perquè el jugador s'acostumi a totes les mecàniques que podrà trobar.

4.7 Creació del menú

Una vegada que estaven les mecàniques programades i el primer nivell fet em calia programar el menú del meu videojoc. No es pot fer un videojoc sense menú. Per aquesta part vaig decidir mantenir el disseny simple perquè no és necessari complicar-se molt en un menú, l'atractiu del videojoc no és el menú, són els nivells. Llavors vaig preferir dedicar-hi més temps a les mecàniques i als nivells.

Vaig fer dos menús, el menú principal i el menú de pausa. El menú principal és el primer menú que apareix en començar el videojoc i el menú de pausa és el menú que apareix una vegada ja dins del nivell.



Imatge 76. Imatge del menú del videojoc. *Autor propi.*

És un menú molt senzill que només té l'opció "Start" per començar i l'opció "Quit" per sortir. Si li donem a "Quit" el programa es tanca i si li donem a "Start" l'escena canvia al primer nivell.

Una vegada a dins del nivell si li premem la tecla ESC, s'obre el menú de pausa.

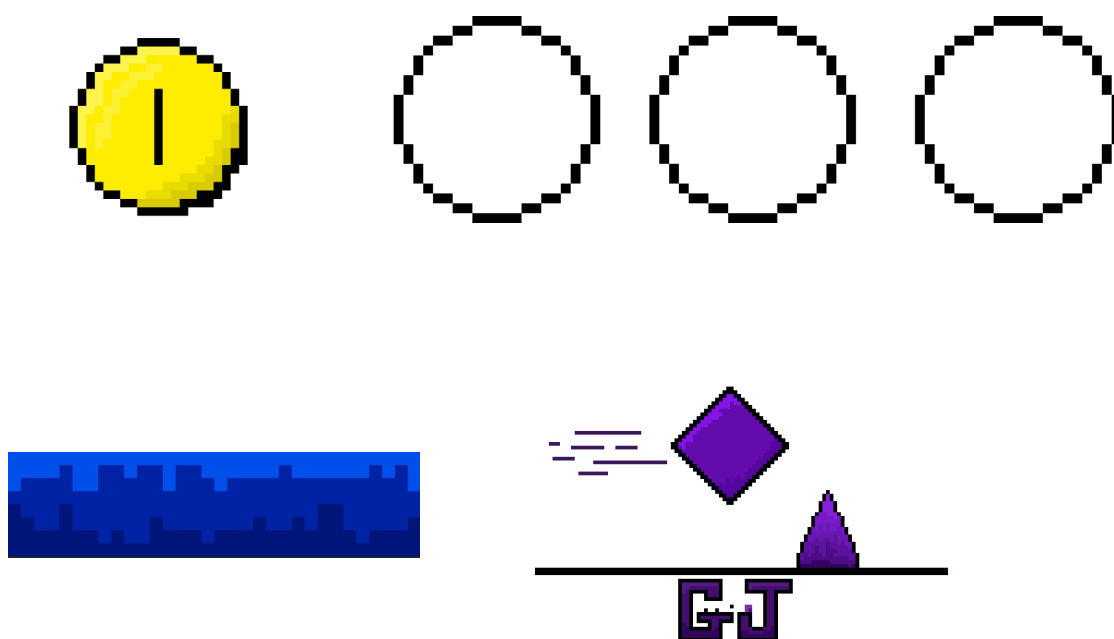


Imatge 77. Imatge del menú de pausa del videojoc. *Autor propi.*

Que conté les tecles "Continuar" que ens tanca el menú i tornem al nivell i la tecla "Salir" que tanca el videojoc. En aquest menú també vaig afegir una secció on ens indica quantes monedes hem agafat al nivell. La mecànica de les monedes va ser una que vaig afegir més tard, però que simplement consisteix en tres monedes col·locades estratègicament per tal que sigui molt difícil agafar-les. D'aquesta manera el mateix nivell té diverses dificultats segons com juguis, si no vols les monedes, et pots passar el nivell pel camí fàcil, això no obstant, si vols les monedes has d'arriscar.

4.8 Retocs

En aquesta part em vaig dedicar a arreglar els “bugs” del meu videojoc, vaig afegir són, les monedes i una petita mecànica més. També vaig posar-li nom al videojoc i vaig dissenyar la seva portada.



Imatges 78, 79, 80 i 81. Moneda, monedes buides, plataforma especial i logo. *Autor propi.*

Les monedes poden agafar-se durant el nivell i una vegada les tens apareixen al menú de pausa, quan no tens cap veus la figura de les 3 monedes buides i a mesura que les aconseguixes van apareixent.

El rectangle blau és una espècie de plataforma de salt com la que he creat al principi del videojoc, però que només apareix i serveix per agafar monedes. Passar per aquesta plataforma de salt és l'única manera d'agafar les monedes.

I per últim vaig fer la portada del videojoc, anomenat Geometry Jump, en honor al videojoc original Geometry Dash.

5. CREACIÓ DE LA PÀGINA WEB

Una vegada que vaig acabar el meu videojoc calia fer la meva pàgina web on la gent pogués descarregar-ho i saber qui sóc. Vaig acabar fent el projecte en el motor Visual Studio Code.

5.1 Estructura

Primer vaig planificar com seria la meva pàgina web, vaig pensar que volia posar i quina seria la seva funció. A la pàgina web havia de posar, l'enllaç per descarregar el meu videojoc, informació sobre qui sóc, una manera per tal que la gent pogués contactar amb mi i un breu resum del videojoc.

Vaig dividir l'estructura de la meva pàgina web en tres parts:

1. Un apartat sobre el seu creador, jo "Sobre mi".
2. Un apartat on descarregar el videojoc, "Proyectos".
3. Un apartat on els jugadors puguin contactar amb mi, "Contacta conmigo".

Sobretot volia que la pàgina web fos minimalista i fàcil d'entendre per a tothom, l'atractiu és el videojoc i la pàgina web ha de complir amb la seva funció de permetre una descàrrega fàcil i intuïtiva del videojoc.

5.2 Part visual de la pàgina web

Abans de posar-me a programar directament vaig elaborar i seleccionar les imatges que voldria que hi apareguessin:



Imatge 82, 83 i 84. Imatge del logo del meu videojoc, portada i foto d'estoc.



Autor propi. Autor propi. Imatge extreta de <https://www.freepik.com>.

Aquestes van ser les tres fotos que vaig posar a la pàgina web, les dues primeres fetes per mi i l'altra extreta d'internet. La primera és el logo que surt en la pestanya de la pàgina web, per exemple la del Puig Castellar:



Imatge 85. Imatge del logo de la pàgina web del Puig Castellar. *Extreta de <https://elpuig.xeill.net/>.*

La segona és el logo del meu videojoc i apareix quan ho descarregues. I l'última només és una imatge predeterminada de Google que em serveix per ambientar la pàgina web. Però, en el cas de tractar-se d'un projecte amb fins econòmics probablement hauria de ser canviada i substituïda per una fotografia que representés l'empresa. En el meu cas no ho necessito.

5.3 Programació

Una vegada tenia tot això fet em vaig dedicar a programar la pàgina web a Visual Studio Code, utilitzant principalment HTML i CSS i una mica de JavaScript. El mateix codi de programació està dividit d'una manera similar a com està dividida la pàgina web.

Explicaré el codi per tal que així a l'annex pugui ser entès encara el lector no tingui experiència prèvia.

Primer de tot cal saber que una pàgina web no es programa tot en un sol "script", un sol document, sinó que és la unió de diversos documents que es complementen entre si. És a dir, en el meu cas són tres documents, el document principal programat en HTML, un segon document programat en CSS que s'encarrega dels colors, fonts, estils... (la part visual de la pàgina web), i per últim un document en JavaScript encarregat de les funcions dinàmiques de la pàgina web, el comportament i la manera amb la qual interactuen amb la persona que navega la web.

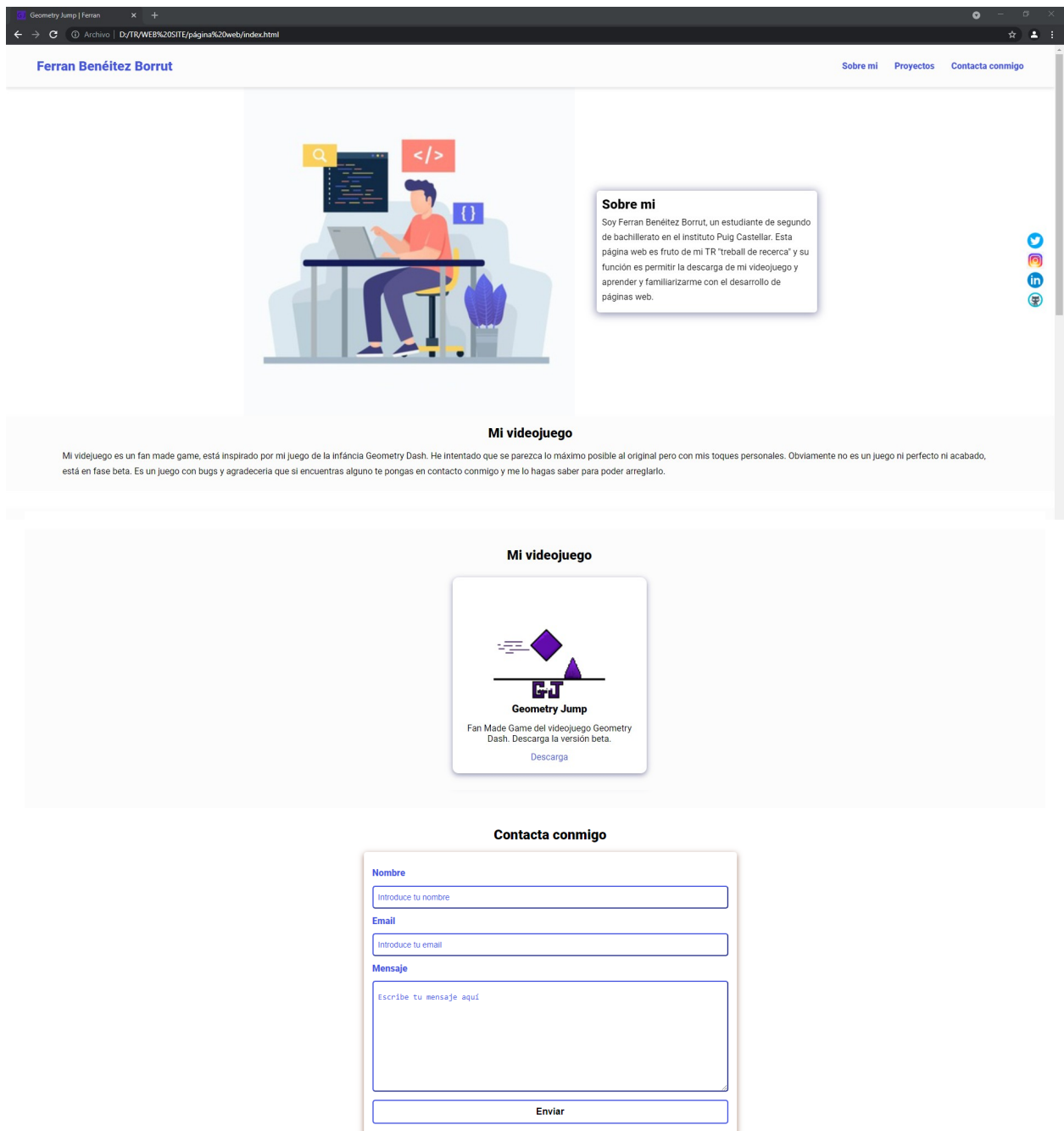
El primer document en HTML està subdividit en parts:

1. "Meta" informació o "meta" etiquetes: Informació general i necessària per a la pàgina web i pel mateix navegador. Exemples: en quin llenguatge està la pàgina web escrita (en el meu cas espanyol), el títol de la pàgina, definició de la font, descripció de la pàgina web, paraules clau, copyright, etc.
2. CSS link: Aquest és un apartat molt breu que només ocupa una línia de codi i que serveix perquè una vegada la pàgina web s'executi busqui l'arxiu, l'script, en CSS que controla l'estil visual. En aquesta línia indiquem a l'arxiu en HTML on es troba al nostre ordinador l'arxiu CSS per tal que el pugui anar a buscar i l'executiu.

3. Favicons: En aquesta part és on definim quina serà la icona que apareixerà a la pestanya.
4. Navbar: En aquesta part és on programo la barra que té la pàgina de web a dalt on surten els diferents apartats i si cliques en un, et porta directament sense la necessitat de baixar tu.
5. About me: Aquesta és la part on escric la informació disponible sobre mi.
6. More about: En aquesta part és on faig una descripció breu del meu videojoc i demano que en el cas que algú trobi un error m'ho notifiqui.
7. Projects section: Aquí és on poso la descàrrega del meu videojoc.
8. Contacts section: Aquí és on programo la missatgeria i així la gent es pugui comunicar amb mi.
9. Website scripts: Aquesta és la part on l'arxiu va a buscar l'altre arxiu, l'script en JavaScript que controla el comportament dinàmic de la pàgina web. El funcionament és exactament el mateix que al CSS enllaç.

5.4 Resultats de la pàgina web

Així va quedar la pàgina web:



Imatge 84. Imatge del resultat de la meva pàgina web. *Autor propi.*

6. CONCLUSIONS

Després d'haver acabat tot aquest treball i una vegada vaig acabar el meu videojoc i la meva pàgina web vaig tornar a la meva hipòtesi i als meus objectius per veure si havia completat els meus propòsits.

Els meus objectius van ser:

1. Aprendre a fer un videojoc en Godot.
2. Aprendre a fer una pàgina web.
3. Aprendre a publicar aquesta pàgina web.
4. Aprendre sobre el SEO.
5. Aprendre a patentar un videojoc.
6. Aprendre JavaScript, HTML, CSS i el llenguatge propi de Godot Engine.
7. Aprendre pixel art i Photoshop.

Ara una vegada acabat els puc analitzar un a un.

Primer de tot em vaig proposar aprendre a fer un videojoc en Godot Engine. He aconseguit aquest objectiu a la perfecció, amb l'ajuda de tutorials a YouTube i pàgines web vaig aconseguir aprendre els conceptes bàsics. Una vegada vaig adquirir aquests coneixements vaig començar a programar pel meu compte i vaig anar aprenent de manera autodidàctica, a base de prova i error. Puc afirmar que he après a fer videojocs amb una complexitat no molt elevada a Godot Engine i també he adquirit la capacita d'entendre el codi d'altres programadors.

El segon objectiu que em vaig proposar va ser aprendre a fer una pàgina web. També he aconseguit aquest propòsit, he aconseguit aprendre HTML i soc capaç d'elaborar una pàgina web no molt complicada d'una manera bastant sòlida.

El tercer objectiu va ser aprendre a publicar una pàgina web, malauradament això no va ser possible a causa del fet que es necessitava pagar una quantitat de diners de la qual no disposava. Tanmateix, sí que vaig poder entendre com són publicades i com funcionen els servidors.

El quart objectiu està molt relacionat amb el tercer i amb les pàgines web. Abans de començar aquest TR no coneixia aquest concepte i ni tan sols sabia que existia. Es tracta del SEO, era una cosa desconeguda per mi, però ara que he acabat el TR encara que sigui un concepte que encara em sembla difícil, puc dir que he aconseguit fer-me una idea de com funciona. El que si he après ha estat en el fet que consisteix i no tant en el seu funcionament i m'he adonat que és un factor molt important a l'hora de posicionar una pàgina web.

Aprendre a patentar una pàgina web va ser el cinquè objectiu. Aquest malauradament no ho vaig poder assolir, tot i això, tampoc em vaig veure en la necessitat de fer-ho. Potser en un futur projecte sí que ho hauré de fer. I de la mateixa manera amb el sisè objectiu, patentar un videojoc.

El setè objectiu va ser aprendre el llenguatge propi de Godot Engine, HTML, JavaScript i CSS. Respecte al llenguatge de Godot Engine i HTML he après molt i he obtingut molta experiència en aquest entorn. Pel que fa a JavaScript i CSS he après, però no he aconseguit el mateix nivell com amb els altres dos llenguatges.

L'últim objectiu va ser aprendre Pixel Art i Photoshop. Després d'haver fet tot els sprites del meu videojoc en l'estil Pixel Art i en el programa Photoshop he aconseguit molta experiència i he après tant la tècnica com el programa.

Una vegada vaig repassar tots els objectius calia reprendre la hipòtesi per veure el resultat.

La meva hipòtesi era la següent: Serà possible la creació d'una pàgina web i un videojoc sense tenir coneixements previs de programació?

Sí que és possible, és possible crear un videojoc i després una pàgina web des de zero sense cap coneixement previ. És més, així és com la majoria de programadors comencen, primer elaboren els seus projectes individuals a l'adolescència o fins i tot abans i l'utilitzen com a punt de partida en la seva carrera. Animo a qualsevol persona que li agradin els videojocs a començar el seu propi projecte, al principi sembla difícil i et trobes perdut, però val la pena l'esforç.

Malgrat tot això el meu treball no és perfecte i un futur treball de recerca d'algú podria ser agafar el meu videojoc i optimitzar-ho i actualitzar-ho partint de la base que he creat.

En conclusió, he aconseguit elaborar un videojoc del qual estic molt content i satisfet i una pàgina web la qual compleix amb el seu propòsit. És veritat que si tingués més temps faria el videojoc més complex i afegiria més coses, però dintre del temps que he tingut crec que he fet el millor videojoc possible. A més a més, estic molt content amb la meva pàgina web. Un altre punt a tenir en compte és que aquests dos projectes em permeten poder continuar-los en un futur quan tingui més experiència amb la programació.

7. AGRAÏMENTS

Agraeixo sobretot a la meva tutora del treball de recerca, Anna Sampedro Vizcaino, per guiar-me durant el treball i acceptar ser la meva tutora malgrat que no va ser des del principi. Agraeixo les correccions i el temps que l'hi ha dedicat encara que ha estat un treball tant extens, sense ella aquest treball no seria així.

També agraeixo al meu primer tutor del treball de recerca Jaime Morcillo, sense ell el videojoc no existiria, ell em va motivar a fer-ho quan no ho tenia molt clar i estava perdut. Estic molt agraït pel fet que aquest videojoc ha estat el meu primer projecte de programació i gràcies a això qui sap si acabaré treballant amb alguna cosa relacionada. Sense ell estaria fent un treball de recerca molt diferent.

Per últim vull agrair a dos amics que em van ajudar amb el disseny dels personatges, aportant idees i ajudant-me en el procés. Agraeixo molt els seus consells, els dos són grans dibuixants i em van ajudar molt. Gràcies, Olivier Offray Rodriguez i Miquel Gil Reixach.

8. WEBGRAFIA

Godot Engine, Documentation [en línea]. [Consultat: Abril 2021] Disponible a:

<<https://docs.godotengine.org/en/stable/about/introduction.html>>

Medium, How to start making pixel art [en línea]. [Consultat: Abril 2021] Disponible a:

<<https://medium.com/pixel-grimoire/how-to-start-making-pixel-art-2d1e31a5ceab>>

Adobe, Master these fundamental Photosop tools [en línea]. [Consultat: Maig 2021] Disponible a:

<<https://medium.com/pixel-grimoire/how-to-start-making-pixel-art-2d1e31a5ceab>>

Moz, What is SEO? [en línia]. [Consultat: Maig 2021] Disponible a:

<<https://moz.com/learn/seo/what-is-seo>>

Blog Candidatos [en línia]. [Consultat: Juny 2021] Disponible a:

<<https://blog.tecnoempleo.com/candidatos/2020/15420/las-7-etapas-del-desarrollo-de-un-videojuego/>>

Wikipedia, Proyección isométrica [en línea]. [Consultat: Juny 2021] Disponible a:

<https://es.wikipedia.org/wiki/Proyecci%C3%B3n_isom%C3%A9trica>

UPC, Història dels videojocs [en línea]. [Consultat: Juliol 2021] Disponible a:

<<https://www.fib.upc.edu/retroinformatica/historia/videojocs.html>>

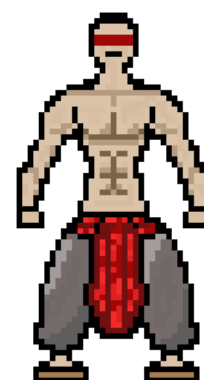
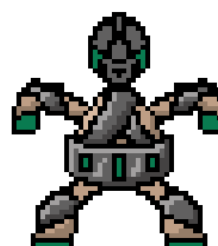
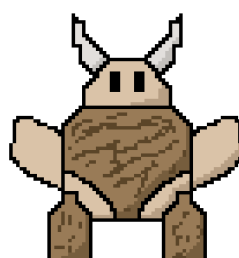
Wikipedia, Pixel Art [en línea]. [Consultat: Juliol 2021] Disponible a:

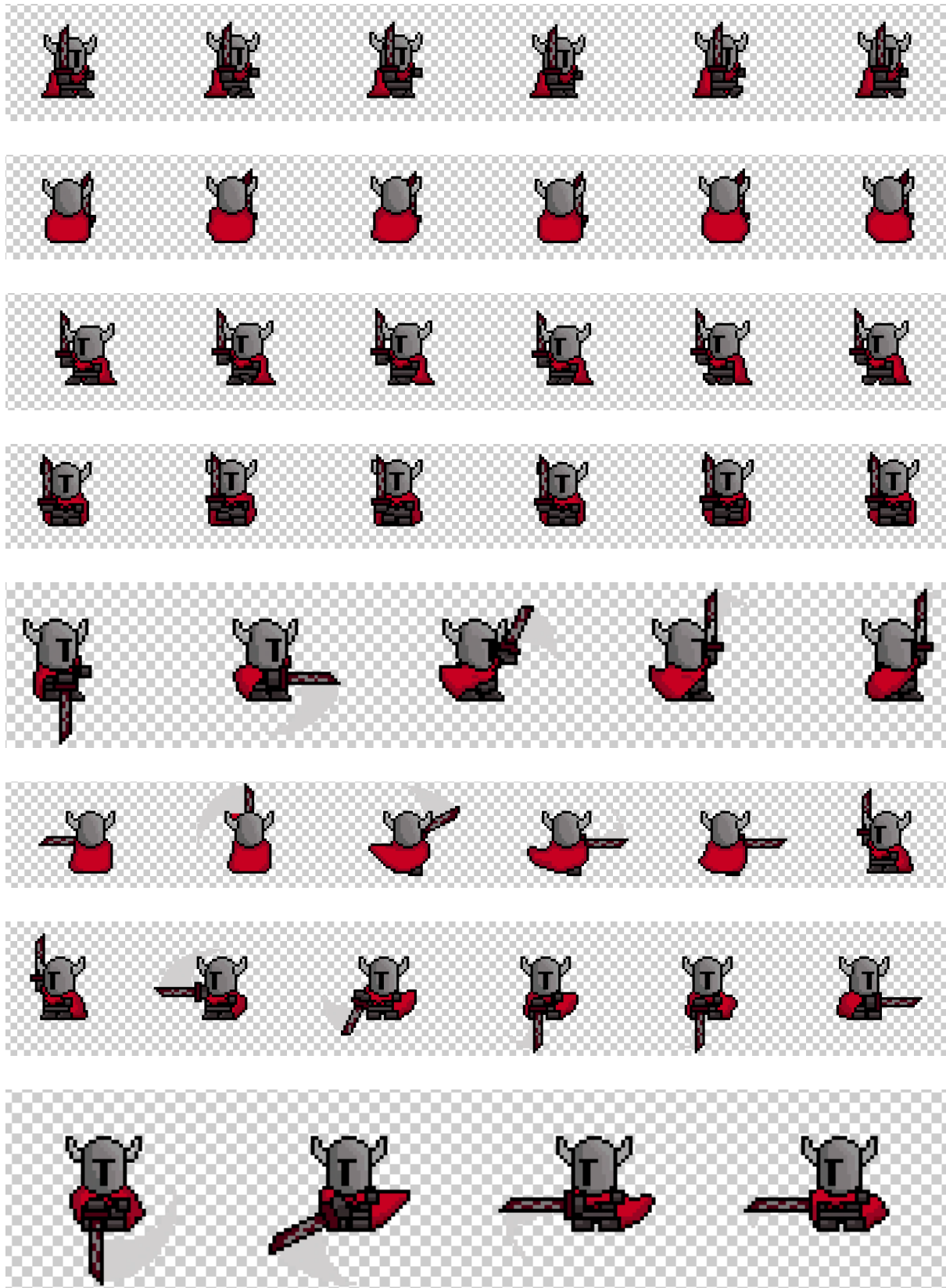
<[Youtube, HeartBeast \[en línea\]: \[Consultat: juliol 2021\] Disponible a:](https://es.wikipedia.org/wiki/Pixel_art#:~:text=El%20arte%20de%20p%C3%ADxel%20debe,Sega%20Master%20System%20(1985).></p></div><div data-bbox=)

<<https://www.youtube.com/channel/UCrHQNOyU1q6BFfkNq2CYMA>>

9. ANNEX

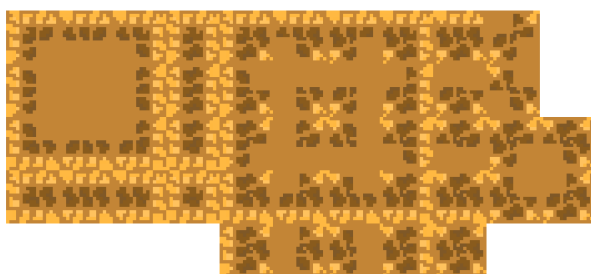
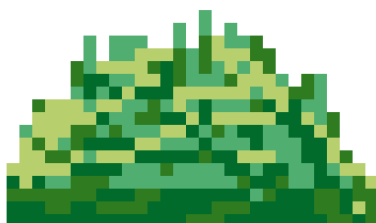
9.1 Sprites de la primera idea de videojoc





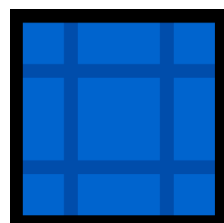
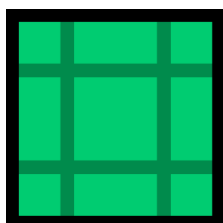
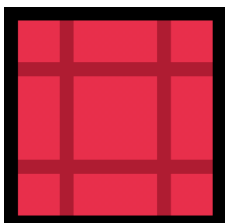
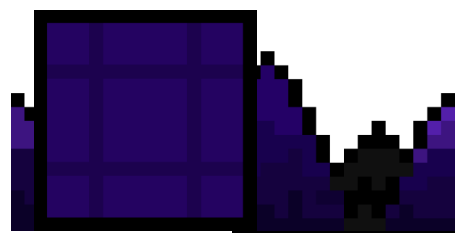
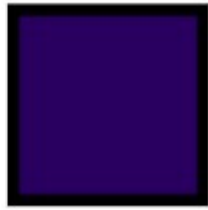


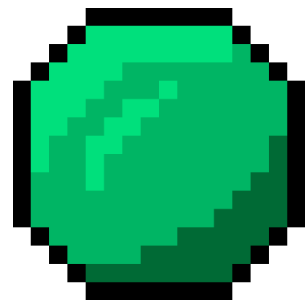
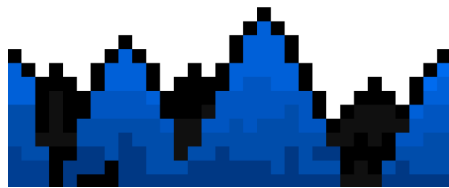
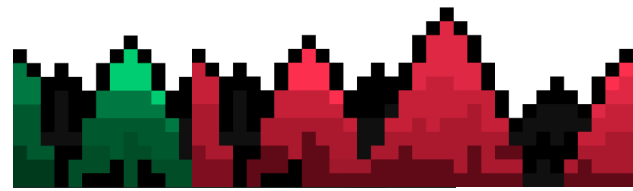






9.2 Sprites del videojoc final





9.3 Scripts del videojoc

9.3.1 Script del protagonista

```
1  extends KinematicBody2D
2
3  var bala_velocidad = 1000
4  var bala = preload("res://bala.tscn")
5  >| >|
6  export var pruebas = false
7  export (int) var speed = 200
8  export (int) var speed2 = 300
9  export (int) var jump_speed = -300
10 export (int) var gravity = 1200
11 export (int) var gravity2 = -1300
12 export (int) var jump_seed2 = -450
13 export (int) var jump_seed3 = -650
14 export (int) var b = 0
15 export (int) var c = 0
16 export (int) var z = 1
17 export (int) var k = 0
18 export (int) var yes = 1
19 export (int) var change = 1
20 export (int) var one = 0
21 export var check = true
22 export (int) var rotation_speed = 500
23 export var double = false
24 var velocity = Vector2.ZERO
25 var movimiento = Vector2()
26 export (int) var intentos = 1
27 var control = 1
28 var spider_change = 1
29
30 enum {
31 >| Normal,
32 >| Invertido,
33 >| Vuelo,
34 >| Spider,
35 >| Win
36 }
37
38 var state = Normal
```

```
40
41 ▾ func _physics_process(delta):
42     |
43     ▾ match state:
44     ▾     | Normal:
45     ▾     |     | Normal(delta)
46     ▾     |     |
47     ▾     |     | Invertido:
48     ▾     |     |     | Invertido(delta)
49     ▾     |     |     |
50     ▾     |     |     | Vuelo:
51     ▾     |     |     |     | Vuelo(delta)
52     ▾     |     |     |     |
53     ▾     |     |     |     | Spider:
54     ▾     |     |     |     |     | Spider(delta)
55     ▾     |     |     |     |     |
56     ▾     |     |     |     |     | Win:
57     ▾     |     |     |     |     |     | Win()
58     ▾     |     |     |     |     |     |
59     ▾     |     |     |     |     |     |
60     ▾     |     |     |     |     |     |
61     ▾     |     |     |     |     |     | if is_on_wall():
62     ▾     |     |     |     |     |     |     | matar()
63     ▾     |     |     |     |     |     |
64     ▾     |     |     |     |     |     | if is_on_floor() or is_on_ceiling():
65     ▾     |     |     |     |     |     |     | check = true
66     ▾     |     |     |     |     |     |     | $Sprite.rotation = false
67     ▾     |     |     |     |     |     | else:
68     ▾     |     |     |     |     |     |     | check = false
69     ▾     |     |     |     |     |     |     | $Sprite.rotate(7.5 * delta)
70     ▾     |     |     |     |     |     |
71     ▾ func Vuelo(delta):
72     ▾     |
73     ▾     |     | speed2 = 300
74     ▾     |     |
75     ▾     |     | #Este trozo detecta cuando el jugador pulsa la tecla espacio y hace que cambie
76     ▾     |     | #la dirección en el aire
77     ▾     |     | if Input.is_action_just_pressed("up"):
78     ▾     |     |     | fly()
79     ▾     |     |     |
80     ▾     |     | #Esta línea de aquí mantiene al jugador en movimiento constante hacia la derecha
81     ▾     |     | movimiento.x += 25
```

```

82 >| >|
83 >| >| if Input.is_action_just_pressed("up") and not change % 2 == 0:
84 >| >| >| movimiento = move_and_slide(movimiento * 0)
85 >| >| #Esta linea normaliza el vector en dos dimensiones del movimiento
86 >| >| #Esto se hace para que la velocidad del jugador no sea mayor moviendose en diagonal
87 >| >| #que en horizontal o vertical
88 >| >| movimiento = movimiento.normalized()
89 >| >|
90 >| >| #Esta linea modifica el vector del movimiento multiplicandolo por la velocidad preestablecida
91 >| >| #anteriormente, esta linea se ocupa del movimiento en horizontal solo
92 >| >| movimiento = move_and_slide(movimiento * speed2)
93 >| >|
94 >| >| #Estas dos lineas son las encargadas del movimiento en el eje y, vertical.
95 >| >| velocity.y += gravity2 * delta * yes
96 >| >| velocity = move_and_slide(velocity, Vector2.UP)
97 >| >|
98 >| >| #Control de la rotación del personaje al saltar
99 >| >| if Input.is_action_pressed("up") and not change % 2 == 0:
100 >| >| >| $Sprite2.rotation_degrees = -65 * yes
101 >| >| >|
102 >| >| elif Input.is_action_pressed("up") and change % 2 == 0:
103 >| >| >| $Sprite2.rotation_degrees = -30 * yes
104 >| >|
105 >| >| func Invertido(delta):
106 >| >|
107 >| >| speed2 = 0
108 >| >| #Velocidad del jugador al ir invertido
109 >| >| speed = 200
110 >| >|
111 >| >| #Gravedad negativa, porque va invertido, por lo tanto le hace subir.
112 >| >| gravity = -1500
113 >| >|
114 >| >| #Velocidad con la que se mueve en el eje y al saltar, es positivo porque está invertido,
115 >| >| #en el momento que vaya normal todos los signos cambiaran.
116 >| >| jump_speed = 350
117 >| >|
118 >| >| #Mantiene al jugador en movimiento
119 >| >| movimiento.x += 20
120 >| >|
121 >| >| #Normalización del vector y multiplicación por la velocidad.
122 >| >| movimiento = movimiento.normalized()
123 >| >| movimiento = move_and_slide(movimiento * speed)
124 >| >|

```

```

124 >| >|
125 >| >| #Control de la velocidad y del vector 2D en vertical.
126 >| >| velocity.y += gravity * delta
127 >| >| velocity = move_and_slide(velocity, Vector2.UP)
128 >| >|
129 >| >| #Condición que detecta cuando el jugador pulsa la tecla espacio para saltar.
130 >| >| if Input.is_action_pressed("up"):
131 >| >| >| salto_inverso(delta)
132 >| >| >|
133 >| >| if c != 1 and c != 3 and spider_change != 2:
134 >| >| >| state = Normal
135 >| >| >|
136 >| >| if spider_change == 2:
137 >| >| >| state = Spider
138 >| >| >|
139 >| >| func Normal(delta):
140 >| >| >|
141 >| >| >| speed2 = 0
142 >| >| >| #Cuando el jugador va normal
143 >| >| >| #Velocidad del jugador.
144 >| >| >| speed = 200
145 >| >| >|
146 >| >| >| #Gravedad del jugador, esta vez positiva porque va normal.
147 >| >| >| gravity = 1200
148 >| >| >|
149 >| >| >| #Velocidad que tiene en el eje y al saltar, negativa, porque tiene dirección opuesta
150 >| >| >| #a la gravedad.
151 >| >| >| jump_speed = -300
152 >| >| >|
153 >| >| #Línea que mantiene al jugador en constante movimiento hacia la derecha y normalización del vector.
154 >| >| movimiento.x += 20
155 >| >| movimiento = movimiento.normalized()
156 >| >| movimiento = move_and_slide(movimiento * speed)
157 >| >| >|
158 >| >| #Estas líneas controlan el movimiento del jugador en el eje y.
159 >| >| velocity.y += gravity * delta
160 >| >| velocity = move_and_slide(velocity, Vector2.UP)
161 >| >| >|
162 >| >| #Condición que detecta cuando el jugador pulsa la tecla espacio y efectúa el salto.
163 >| >| if Input.is_action_pressed("up"):
164 >| >| >| salto(delta)
165 >| >| >|

```



```

124 >| >|
125 >| >| #Control de la velocidad y del vector 2D en vertical.
126 >| >| velocity.y += gravity * delta
127 >| >| velocity = move_and_slide(velocity, Vector2.UP)
128 >| >|
129 >| >| #Condición que detecta cuando el jugador pulsa la tecla espacio para saltar.
130 >| >| if Input.is_action_pressed("up"):
131 >| >| >| salto_inverso(delta)
132 >| >| >|
133 >| >| if c != 1 and c != 3 and spider_change != 2:
134 >| >| >| state = Normal
135 >| >| >|
136 >| >| if spider_change == 2:
137 >| >| >| state = Spider
138 >| >| >|
139 >| >| func Normal(delta):
140 >| >| >|
141 >| >| >| speed2 = 0
142 >| >| >| #Cuando el jugador va normal
143 >| >| >| #Velocidad del jugador.
144 >| >| >| speed = 200
145 >| >| >|
146 >| >| >| #Gravedad del jugador, esta vez positiva porque va normal.
147 >| >| >| gravity = 1200
148 >| >| >|
149 >| >| >| #Velocidad que tiene en el eje y al saltar, negativa, porque tiene dirección opuesta
150 >| >| >| #a la gravedad.
151 >| >| >| jump_speed = -300
152 >| >| >|
153 >| >| >| #Línea que mantiene al jugador en constante movimiento hacia la derecha y normalización del vector.
154 >| >| >| movimiento.x += 20
155 >| >| >| movimiento = movimiento.normalized()
156 >| >| >| movimiento = move_and_slide(movimiento * speed)
157 >| >| >|
158 >| >| >| #Estas líneas controlan el movimiento del jugador en el eje y.
159 >| >| >| velocity.y += gravity * delta
160 >| >| >| velocity = move_and_slide(velocity, Vector2.UP)
161 >| >| >|
162 >| >| >| #Condición que detecta cuando el jugador pulsa la tecla espacio y efectúa el salto.
163 >| >| >| if Input.is_action_pressed("up"):
164 >| >| >| salto(delta)

```

```
166 >| >| if (c == 1 or c == 3) and k == 0 and b != 1 and spider_change != 2:
167 >| >| >| state = Invertido
168 >| >|
169 >| >| if spider_change == 2:
170 >| >| >| state = Spider
171 >| >| >|
172 >| func Spider(delta):
173 >|
174 >| >| var gravity_spider = 5000
175 >| >|
176 >| >| movimiento.x += 20
177 >| >| movimiento = movimiento.normalized()
178 >| >| movimiento = move_and_slide(movimiento * speed)
179 >| >|
180 >| >| velocity.y += gravity_spider * delta * control
181 >| >| velocity = move_and_slide(velocity, Vector2.UP)
182 >| >|
183 >| >| if Input.is_action_just_pressed("up"):
184 >| >| >| spider_control(delta)
185 >| >|
186 >| func Win():
187 >| get_tree().reload_current_scene()
188 >| >|
189 >| func spider_control(delta):
190 >| if is_on_floor() or is_on_ceiling():
191 >| >| control = control * (-1)
192 >| else:
193 >| >| pass
194
195 >| func double_jump():
196 >|
197 >| if c != 1 and c != 3:
198 >| >| velocity.y = jump_speed
199 >|
200 >| elif (c == 1 or c == 3) and k == 0 and b != 1:
201 >| >| jump_speed = 350
202 >| >| velocity.y = jump_speed>>|
203
204 >| func spin(delta):
205 >| rotation = true
206 >|
```

```

206 >|
207 #Función que es llamada cuando el jugador pulsa la tecla espacio y va en un movimiento normal
208 #que le permite saltar, la línea "if is_on_floor():" detecta si el jugador está en el suelo para
209 #evitar que el personaje pueda saltar estando en el aire, no tendría sentido.
210 < func salto(delta):
211 < >| >| if is_on_floor():
212 >| >| >| velocity.y = jump_speed
213 >| >| >|
214 #Función que controla el disparo del jugador, crea la variable bala, inserta la posición de la bala
215 #en la del jugador y en el momento que es disparada redirige la inclinación con la cual sale en base
216 #a la inclinación del jugador y le da una velocidad mayor a la que lleva en ese momento el personaje.
217
218 < func disparo():
219 >| var bala_instance = bala.instance()
220 >| var Jugador = get_parent().get_node("Jugador")
221 >| bala_instance.position = Jugador.position
222 >| bala_instance.rotation_degrees = rotation_degrees
223 >| bala_instance.apply_impulse(Vector2(),Vector2(bala_velocidad,0).rotated(rotation))
224 >| get_tree().get_root().call_deferred("add_child", bala_instance)
225 >|
226 #Función que una vez llamada reinicia el nivel, muerte del jugador.
227 < func matar():
228 >| c = 0
229 >| state = Normal
230 >| control = 1
231 >|
232 #Función que una vez llama hace que el jugador salte con una velocidad diferente a la que tiene y sin
233 #que él pulse la tecla.
234 < func boing():
235 >| velocity.y = jump_seed2
236 >|
237 #Función que una vez llama hace que el jugador cambie su velocidad de salto.
238
239 < func boing2():
240 >| velocity.y = jump_seed3
241
242 #Función que una vez llama cambia la dirección en la que se mueve el jugador volando y
243 #reinicia su velocidad en el eje y para que el cambio sea instantaneo y no tenga que frenar.
244

```

```

244
245 < func fly():
246 >|
247 >| yes = -yes
248 >| velocity.y = 0
249 >|
250 < >| if not change % 2 == 0:
251 >| >| >| speed2 = 400
252 >| >| >| gravity2 = -1200
253 >| >| >|
254 < >| elif change % 2 == 0:
255 >| >| >| speed2 = 300
256 >| >| >| gravity2 = -1300
257 >| >|

```

```

262
263 ▾ func _on_Area2D_body_entered(body):
264 ▾ >| if "TileMap8" in body.name:
265   >|   >| matar()
266 ▾ >| if "TileMap7" in body.name:
267   >|   >| boing()
268 ▾ >| if "TileMap9" in body.name and Input.is_action_pressed("up"):
269   >|   >| boing2()
270 ▾ >| if "Pinchos" in body.name:
271   >|   >| matar()
272 ▾ >| if "Pinchos normal" in body.name:
273   >|   >| matar()
274 ▾ >| if "TileMap5" in body.name:
275   >|   >| matar()
276 ▾ >| if "TileMap10" in body.name:
277   >|   >| matar()
278 ▾ >| if "TileMap2" in body.name:
279   >|   >| b += 1
280   >|   >| $Sprite2.visible = not $Sprite2.visible
281   >|   >| $Sprite.visible = not $Sprite.visible
282   >|   >| $Sprite2.rotation_degrees = -65 * yes
283 ▾ >| if "TileMap11" in body.name:
284   >|   >| matar()
285 ▾ >| if "TileMap3" in body.name:
286   >|   >| c += 1
287 ▾ >| if "TileMap13" in body.name:
288   >|   >| k += 1
289 ▾ >| if "Muro Inclinado6" in body.name:
290   >|   >| matar()
291 ▾ >| if "Muro Inclinado5" in body.name:
292   >|   >| matar()
293 ▾ >| if "Muro Inclinado" in body.name:
294   >|   >| matar()
295 ▾ >| if "Muro Inclinado" in body.name:
296   >|   >| matar()
297 ▾ >| if "Pinchos del reves" in body.name:
298   >|   >| matar()
299 ▾ >| if Input.is_action_pressed("up"):
300   >|   >| double_jump()
301 ▾ >| if "Idea" in body.name:
302   >|   >| Idea()
303   >|   >|

```

```
304 #Esta función una vez llamada efectua el salto en el movimiento invertido. La línea
305 #"if is_on_ceiling():" hace lo mismo que el "if is_on_floor():" pero esta vez con el
306 #techo no con el suelo.
307 func salto_inverso(_delta):
308     if is_on_ceiling():
309         jump_speed = 350
310         velocity.y = jump_speed
311
312 func _on_Spider_Portal_body_entered(body):
313     $Spider.visible = true
314     $Sprite.visible = false
315     if "Jugador" in body.name:
316         state = Spider
317
318 func _on_Spider_Portal_Exit_body_entered(body: Node) -> void:
319     $Spider.visible = false
320     $Sprite.visible = true
321     if "Jugador" in body.name:
322         state = Normal
323
324 func Idea():
325     velocity.y = -400
326
```

9.3.2 Scripts d'altres parts del videojoc

```
1 extends Label
2 export (int) var intentos = 1
3 >|
4 ▾ func _ready() -> void:
5   >| display()
6
7 ▾ func _on_Area2D_body_entered(body: Node) -> void:
8   >| if "TileMap7" in body.name:
9     >| >| contador()
10    >| >| display()
11  >| if "Pinchos normal" in body.name:
12    >| >| contador()
13    >| >| display()
14  >| if "TileMap10" in body.name:
15    >| >| contador()
16    >| >| display()
17  >| if "Pinchos del revés" in body.name:
18    >| >| contador()
19    >| >| display()
20  >| if "Fondo Pinchos" in body.name:
21    >| >| contador()
22    >| >| display()
23  >| if "Fondo Pinchos Al revés" in body.name:
24    >| >| contador()
25    >| >| display()
26
27 ▾ func contador():
28   >| intentos += 1
29
30 ▾ func display():
31   >| text = "Intentos " + str(intentos)
32   >|
33
```

```
1 extends CanvasLayer
2
3 func _ready():
4     >| pass
5     >|
6 func _input(event):
7     >| if event.is_action_pressed("ui_cancel"):
8     >| >| get_tree().paused = true
9 |
```

```
1 extends Control
2
3 func _ready():
4     >| $VBoxContainer/Button.grab_focus()
5     >|
6 →| func _on_Button_pressed() -> void:
7     >| get_tree().change_scene("res://Node2D.tscn")
8     >|
9
10 →| func _on_Button2_pressed() -> void:
11     >| get_tree().quit()
12 |
```



```

1  extends Node2D
2
3  export (int) var a = 1
4
5  func _ready() -> void:
6  >|  $Jugador/Control/VBoxContainer/Button.grab_focus()
7
8  func _input(event):
9  >|
10 >|  if event.is_action_pressed("ui_cancel") and a == 1:
11 >| >|  $AudioStreamPlayer.stream_paused = true
12 >| >|  a = 2
13 >| >|
14 >|  elif event.is_action_pressed("ui_cancel") and a == 2:
15 >| >|  $AudioStreamPlayer.stream_paused = false
16 >| >|  a = 1
17 >| >|
18 >|  if event.is_action_pressed("ui_cancel"):
19 >| >|  $Jugador/Control/VBoxContainer/Button.visible = true
20 >| >|  $Jugador/Control/VBoxContainer/Button2.visible = true
21 >| >|  $Jugador/Control/VBoxContainer/Pause.visible = true
22 >| >|  $Jugador/Control/VBoxContainer/Monedas_Vacias.visible = true
23 >| >|
24 func matar():
25 >|  $Jugador.position.x = 40
26 >|  $Jugador.position.y = -361
27 >|  $Coin/Coin_Sprite.visible = true
28 >|  $Coin_2/Coin2_Sprite.visible = true
29 >|  $Coin_3/Coin3_Sprite.visible = true
30 >|  $Jugador/Spider.visible = false
31 >|  $Jugador/Sprite.visible = true
32 >|  $Jugador/Control/VBoxContainer/Monedas_Vacias/Moneda_1_Menu.visible = false
33 >|  $Jugador/Control/VBoxContainer/Monedas_Vacias/Moneda_2_Menu.visible = false
34 >|

```



```
35
36 ▾ func _on_Area2D_body_entered(body: Node) -> void:
37 ▾ | if "TileMap8" in body.name:
38   | | matar()
39 ▾ | if "Pinchos" in body.name:
40   | | matar()
41 ▾ | if "Pinchos normal" in body.name:
42   | | matar()
43 ▾ | if "TileMap5" in body.name:
44   | | matar()
45 ▾ | if "TileMap10" in body.name:
46   | | matar()
47 ▾ | if "TileMap11" in body.name:
48   | | matar()
49 ▾ | if "Muro Inclinado6" in body.name:
50   | | matar()
51 ▾ | if "Muro Inclinado5" in body.name:
52   | | matar()
53 ▾ | if "Muro Inclinado" in body.name:
54   | | matar()
55 ▾ | if "Muro Inclinado" in body.name:
56   | | matar()
57 ▾ | if "Pinchos del reves" in body.name:
58   | | matar()
59
60 ▾ func _on_Coin_body_entered(body: Node) -> void:
61   | $Coin/Coin_Sprite.visible = not $Coin/Coin_Sprite.visible
62   | $Jugador/Control/VBoxContainer/Monedas_Vacias/Moneda_2_Menu.visible = true
63
64
65 ▾ func _on_Coin_2_body_entered(body: Node) -> void:
66   | $Coin_2/Coin2_Sprite.visible = not $Coin_2/Coin2_Sprite.visible
67   | $Jugador/Control/VBoxContainer/Monedas_Vacias/Moneda_3_Menu.visible = true
68
69
70 ▾ func _on_Coin_3_body_entered(body: Node) -> void:
71   | $Coin_3/Coin3_Sprite.visible = not $Coin_3/Coin3_Sprite.visible
72   | $Jugador/Control/VBoxContainer/Monedas_Vacias/Moneda_1_Menu.visible = true
73
74
```

```
74
75 ▾ func _on_Button_pressed() -> void:
76   >| get_tree().paused = false
77   >| $Jugador/Control/VBoxContainer/Button.visible = false
78   >| $Jugador/Control/VBoxContainer/Button2.visible = false
79   >| $Jugador/Control/VBoxContainer/Pause.visible = false
80   >| $Jugador/Control/VBoxContainer/Monedas_Vacias.visible = false
81
82 ▾ func _on_Button2_pressed() -> void:
83   >| get_tree().quit()
84
85 ▾ func _on_End_body_entered(body: Node) -> void:
86   >| print("Fin")
87
```

9.4 Scripts de la pàgina web

```

página web > <> index.html > ...
 1 <!DOCTYPE html>
 2 <html lang="es">
 3 <head>
 4 <meta charset="UTF-8" />
 5 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
 6 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
 7
 8 <!--CSS-->
 9 <link rel="stylesheet" href="assets/css/styles.css" />
10
11 <!--Favicons-->
12 <link
13 | rel="apple-touch-icon"
14 | sizes="180x180"
15 | href="assets/icons/apple-touch-icon.png"
16 | />
17 <link
18 | rel="icon"
19 | type="image/png"
20 | sizes="32x32"
21 | href="assets/icons/favicon-32x32.png"
22 | />
23
24 <!-- Animate CSS CDN -->
25 <link
26 | rel="stylesheet"
27 | href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css"
28 | />
29 <title>Geometry Jump | Ferran</title>
30 </head>
31
32 <body>
33 <!-- Navbar -->
34 <nav>
35 | <h1>Ferran Benéitez Borrut</h1>
36 | <ul class="navigation">
37 | | <li><a href="#about" class="nav-link">Sobre mi</a></li>
38 | | <li><a href="#projects" class="nav-link">Proyectos</a></li>
39 | | <li><a href="#contact" class="nav-link">Contacta conmigo</a></li>
40 | </ul>
41 | <button class="burger-menu" id="burger-menu">
42 | | <ion-icon class="bars" name="menu-outline"></ion-icon>
43 | | <!-- <ion-icon class="times" name="close-outline"></ion-icon> -->
44 | </button>
45 </nav>

```

```

47 <!-- About me -->
48 <section class="hero" id="about">
49   
55   <div class="bio animate__animated animate__shakeX">
56     <h2 class="bio-title">Sobre mi</h2>
57     <p class="bio-text">
58       Soy Ferran Ben itez Borrut, un estudiante de segundo de bachillerato en el instituto
59       Puig Castellar. Esta p gina web es fruto de mi TR "treball de recerca" y su funci n es permitir
60       la descarga de mi videojuego y aprender y familiarizarme con el desarrollo de p ginas web.
61     </p>
62   </div>
63 </section>
64
65 <!-- More about -->
66 <section class="more-about">
67   <h2>Mi videojuego</h2>
68   <p>
69     Mi videojuego es un fan made game, est  inspirado por mi juego de la infancia Geometry Dash. He intentado que
70     se parezca lo m ximo posible al original pero con mis toques personales. Obviamente no es un juego ni perfecto ni acabado,
71     est  en fase beta. Es un juego con bugs y agradecer a que si encuentras alguno te pongas en contacto conmigo y me lo hagas
72     saber para poder arreglarlo.
73   </p>
74 </section>
75

```

```

76 <!-- Projects section -->
77 <section class="projects" id="projects">
78   <h2 class="projects-title">Mi videojuego</h2>
79   <div class="projects-container">
80     <div class="project-container project-card">
81       
87       <h3 class="project-title">Geometry Jump</h3>
88       <p class="project-details">
89         Fan Made Game del videojuego Geometry Dash.
90         Descarga la versi n beta.
91       </p>
92       <a href="assets/images/prueba.bat" target="_blank" class="project-link">Descarga</a>
93     </div>
94 </section>
95

```

```
96 <!-- Contact section -->
97 <section class="contact" id="contact">
98   <h2>Contacta conmigo</h2>
99   <div class="contact-form-container">
100     <div class="contact-form">
101       <form action="https://formspree.io/f/xyylnggw" method="POST">
102         <div class="form-control">
103           <label for="name">Nombre</label>
104           <input
105             type="text"
106             id="name"
107             name="sender-name"
108             placeholder="Introduce tu nombre"
109             class="input-field"
110             required
111           />
112         </div>
113         <div class="form-control">
114           <label for="email">Email</label>
115           <input
116             type="email"
117             id="email"
118             name="sender-email"
119             placeholder="Introduce tu email"
120             class="input-field"
121             required
122           />
123         </div>
124         <div class="form-control">
125           <label for="message">Mensaje</label>
126           <textarea
127             id="message"
128             cols="60"
129             rows="10"
130             placeholder="Escribe tu mensaje aquí"
131             name="message"
132             class="input-field"
133             required
134           >>/textarea>
135         </div>
136         <input
137           type="submit"
138           value="Enviar"
139           id="submit-btn"
140           class="submit-btn"
141         />
142       </form>
143     </div>
144   </div>
145 </section>
```

```
1 @import url("https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,400;0,900;1,700&display=swap");
2
3 /* Variables */
4 :root {
5     --font-family: "Roboto", sans-serif;
6     --normal-font: 400;
7     --bold-font: 700;
8     --bolder-font: 900;
9     --bg-color: #f0f0f0;
10    --primary-color: #4756df;
11    --secondary-color: #ff7235;
12    --primary-shadow: #8b8eaf;
13    --secondary-shadow: #a17a69;
14    --bottom-margin: 0.5rem;
15    --bottom-margin-2: 1rem;
16    --line-height: 1.7rem;
17    --transition: 0.3s;
18 }
19 /* Variables end */
20
21 html {
22     scroll-behavior: smooth;
23 }
24
25 /* CSS Resets */
26 * {
27     margin: 0;
28     padding: 0;
29     box-sizing: border-box;
30     /* border: 2px solid red; */
31 }
32
33 ul {
34     list-style-type: none;
35 }
36
37 a {
38     text-decoration: none;
39     color: var(--primary-color);
40 }
41
42 a:hover {
43     color: var(--secondary-color);
44 }
45
46 body {
47     font-family: var(--font-family);
48 }
```

```
57  /* Navbar */
58  nav {
59    position: sticky;
60    top: 0;
61    left: 0;
62    z-index: 1;
63    display: flex;
64    align-items: center;
65    justify-content: space-between;
66    padding: 1.5rem 3.5rem;
67    background-color: var(--bg-color);
68    box-shadow: 0 3px 5px 0 rgba(0, 0, 0, 0.1);
69  }
70
71  nav h1 {
72    color: var(--primary-color);
73  }
74
75  nav a {
76    color: var(--primary-color);
77    transition: var(--transition);
78  }
79  nav a:hover {
80    color: var(--secondary-color);
81    border-bottom: 2px solid var(--secondary-color);
82  }
83
84  nav ul {
85    display: flex;
86    gap: 1.9rem;
87  }
88
89  nav ul li {
90    font-weight: var(--bold-font);
91  }
92
```

```
93  √ .burger-menu {
94    color: var(--primary-color);
95    font-size: 2rem;
96    border: 0;
97    background-color: transparent;
98    cursor: pointer;
99    display: none;
100 }
101 /* Navbar ends */
102
103 /* Hero section */
104 √ .hero {
105   display: flex;
106   align-items: center;
107   justify-content: center;
108   gap: 2.5rem;
109   max-width: 68.75rem;
110   margin: auto;
111 }
112
113 √ .hero img {
114   height: 37.5rem;
115   width: 37.5rem;
116 }
117
118 √ .bio {
119   width: 25rem;
120   padding: 0.625rem;
121   border-radius: 6px;
122   box-shadow: 0px 2px 15px 2px var(--primary-shadow);
123 }
124
125 √ .bio h1 {
126   margin-bottom: var(--bottom-margin);
127 }
```



```
129 .bio p {
130   line-height: var(--line-height);
131   padding: 0.3rem 0;
132 }
133 /* Hero section ends */
134
135 /* More about */
136 .more-about {
137   background-color: var(--bg-color);
138   padding: 1rem 6rem;
139 }
140
141 .more-about h2 {
142   margin-bottom: var(--bottom-margin);
143   text-align: center;
144 }
145
146 .more-about p {
147   line-height: var(--line-height);
148   padding: 0.4rem;
149 }
150 /* More about ends */
151
152 /* Skills section */
153 .skills {
154   max-width: 68.75rem;
155   margin: auto;
156   text-align: center;
157   margin-top: 2.5rem;
158 }
159
160 .skill-header {
161   margin-bottom: 1rem;
162 }
163
164 .skills-wrapper img {
165   padding: 1.25rem;
166 }
167
168 .icon {
169   width: 11.875rem;
170   height: 11.25rem;
171 }
172 /* Skills section ends */
```

```
174  /* Projects section */
175  .projects {
176    background-color: var(--bg-color);
177    padding: 32px 0;
178    margin-top: 2rem;
179  }
180
181  .project-pic {
182    width: 65%;
183    height: 60%;
184  }
185
186  .projects-container {
187    display: flex;
188    align-items: center;
189    justify-content: center;
190  }
191
192  .projects-title {
193    text-align: center;
194    margin-bottom: 1rem;
195  }
196
197  .project-container {
198    text-align: center;
199    width: 21.875rem;
200    padding: 1rem;
201  }
202
203  .project-container p {
204    padding: 0.4rem;
205  }
206
207  .project-title {
208    margin-bottom: var(--bottom-margin);
209  }
210
211  .project-details {
212    margin-bottom: var(--bottom-margin);
213  }
214
215  /* Projects section ends */
216
217  /* Contacts section */
218  .contact {
219    margin-top: 2rem;
220  }
```

```
222  ∨ .contact h2 {
223     text-align: center;
224     margin-bottom: var(--bottom-margin-2);
225  }
226
227  ∨ .contact-form-container {
228     max-width: 40.75rem;
229     margin: 0 auto;
230     padding: 0.938rem;
231     border-radius: 5px;
232     box-shadow: 0 3px 10px var(--secondary-shadow);
233  }
234
235  ∨ .contact-form-container label {
236     line-height: 2.7em;
237     font-weight: var(--bold-font);
238     color: var(--primary-color);
239  }
240
241  ∨ .contact-form-container textarea {
242     min-height: 6.25rem;
243     font-size: 14px;
244  }
245
246  ∨ .contact-form-container .input-field {
247     width: 100%;
248     padding-top: 10px;
249     padding-bottom: 10px;
250     border-radius: 5px;
251     border: none;
252     border: 2px outset var(--primary-color);
253     font-size: 0.875rem;
254     outline: none;
255  }
256
257  ∨ .input-field::placeholder {
258     padding: 0.5rem;
259     color: var(--primary-color);
260  }
261
```

```
262  ✓ .submit-btn {
263    width: 100%;
264    padding: 10px;
265    margin: 10px 0;
266    background-color: #fff;
267    border: 2px solid var(--primary-color);
268    border-radius: 5px;
269    font-size: 1rem;
270    font-weight: var(--bold-font);
271    transition: var(--transition);
272  }
273
274  ✓ .submit-btn:hover {
275    background-color: var(--primary-color);
276    border: 2px solid var(--primary-color);
277    cursor: pointer;
278  }
279  /* Contacts section ends */
280
281  /* Social buttons */
282  ✓ .socials {
283    display: flex;
284    flex-direction: column;
285    position: fixed;
286    right: 1%;
287    bottom: 50%;
288  }
289
290  ✓ .socicon {
291    width: 2rem;
292    height: 2rem;
293  }
294  /* Social button ends */
```

```
296  /* Scroll to top button */
297  √ .scroll-up {
298      position: fixed;
299      right: 0.5%;
300      bottom: 3%;
301      cursor: pointer;
302  }
303
304  √ .up-arrow {
305      width: 3rem;
306      height: 3rem;
307  }
308  /* Scroll to top button ends */
309
310  /* Footer section */
311  √ footer {
312      background-color: var(--bg-color);
313      padding: 1.25rem;
314      text-align: center;
315      margin: 2rem 0 0;
316  }
317  /* Footer section ends */
318
319  /* General (utilities) */
320  √ .icon-card {
321      background-color: ■ #fff;
322      border-radius: 11px;
323      box-shadow: 0 3px 10px var(--secondary-shadow);
324      padding: 20px;
325      margin: 10px;
326  }
327
328  √ .project-card {
329      background-color: ■ #fff;
330      border-radius: 11px;
331      box-shadow: 0 3px 10px var(--primary-shadow);
332      padding: 20px;
333      margin: 10px;
334  }
```

```
336 /* Media queries for responsiveness */
337 @media screen and (max-width: 720px) {
338     nav {
339         padding: 1.5rem 1rem;
340     }
341     nav ul {
342         position: fixed;
343         background-color: var(--bg-color);
344         flex-direction: column;
345         top: 86px;
346         left: 10%;
347         width: 80%;
348         text-align: center;
349         transform: translateX(120%);
350         transition: transform 0.5s ease-in;
351     }
352
353     nav ul li {
354         margin: 8px;
355     }
356
357     .burger-menu {
358         display: block;
359     }
360
361     nav ul.show {
362         transform: translateX(0);
363     }
364
365     .hero {
366         margin-top: -4rem;
367         flex-direction: column;
368         gap: 0;
369     }
370
371     .hero img {
372         height: 37.5rem;
373         width: 30rem;
374     }
375
376     .bio {
377         margin-top: -7rem;
378         width: 20.5rem;
379     }
380
381     .more-about {
382         margin-top: 2rem;
383         padding: 1rem 3.5rem;
384     }
```

```
386  ✓ .more-about h2 {
387    |   text-align: center;
388    | }
389
390  ✓ .more-about p {
391    |   text-align: justify;
392    | }
393
394  ✓ .icon {
395    |   width: 5.875rem;
396    |   height: 5.25rem;
397    | }
398
399  ✓ .projects-container {
400    |   flex-direction: column;
401    | }
402
403  ✓ .project-container {
404    |   width: 20.875rem;
405    | }
406
407  ✓ .contact-form-container {
408    |   max-width: 23.75rem;
409    | }
410  }
411
412  ✓ @media screen and (max-width: 420px) {
413  ✓ .hero img {
414    |   height: 37.5rem;
415    |   width: 23rem;
416    | }
417
418  ✓ .bio {
419    |   width: 18.3rem;
420    | }
421
422  ✓ .project-container {
423    |   width: 17.875rem;
424    | }
425
426  ✓ .contact-form-container {
427    |   max-width: 17.75rem;
428    | }
429  }
```

```
1  // Nav hamburgerburger selections
2  const burger = document.querySelector("#burger-menu");
3  const ul = document.querySelector("nav ul");
4  const nav = document.querySelector("nav");
5
6  // Scroll to top selection
7  const scrollUp = document.querySelector("#scroll-up");
8
9  // Select nav links
10 const navLink = document.querySelectorAll(".nav-link");
11
12 // Hamburger menu function
13 burger.addEventListener("click", () => {
14   ul.classList.toggle("show");
15 });
16
17 // Close hamburger menu when a link is clicked
18 navLink.forEach((link) =>
19   link.addEventListener("click", () => {
20     ul.classList.remove("show");
21   })
22 );
23
24 // scroll to top functionality
25 scrollUp.addEventListener("click", () => {
26   window.scrollTo({
27     top: 0,
28     left: 0,
29     behavior: "smooth",
30   });
31 });
32
```