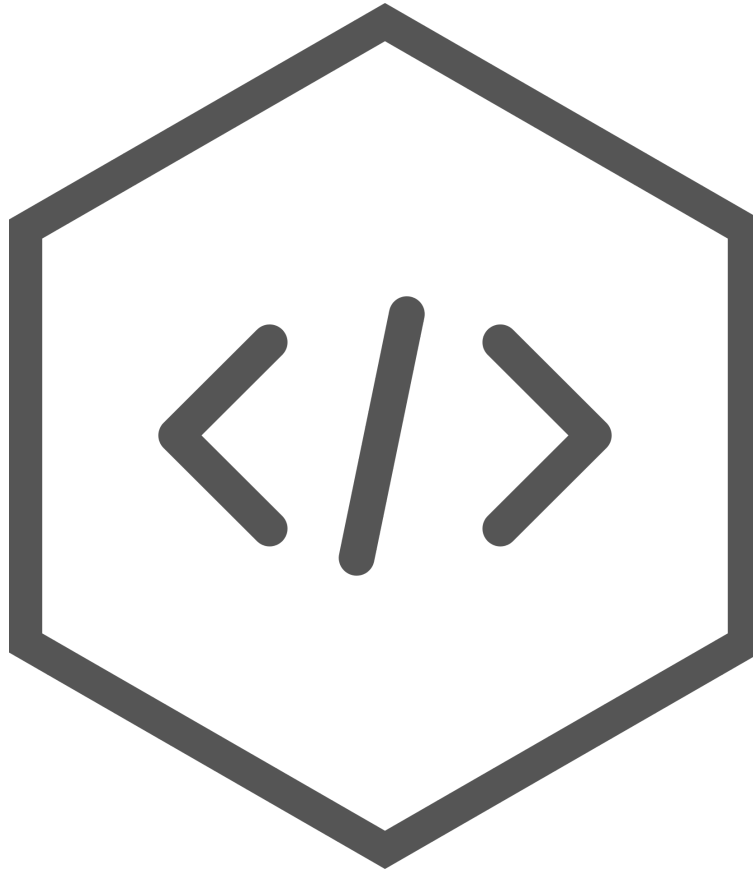


Creació d'un videojoc amb un llenguatge de programació



Nom: Xavier Montes Gallego
Grup: 4t ESO A
Tutor: Jaime Morcillo
Curs: 2020-2021

Índex

1. Introducció	4
2. Hipòtesi	6
3. Els videojocs	7
3.1. Línea cronològica dels videojocs	7
3.1.1. Primeres generacions	7
3.1.2. Innovacions generacionals	10
3.1.3. Últimes generacions i millores gràfiques	12
3.2. Etiquetes dels videojocs	13
3.2.1. Gèneres dels videojocs	13
3.2.1.1. Acció	13
3.2.1.2. Estratègia	14
3.2.1.3. Simulació	15
3.2.1.4. Treball mental	15
3.2.1.5. Aventura	16
3.2.1.6. Altres subgèneres	17
3.2.2. Qualificacions dels videojocs	17
4. Elaboració del videojoc	18
4.1. Introducció del videojoc	18
4.1.1. Història del videojoc	19
4.2. Tileset	19
4.2.1. Personatge principal	20
4.2.2. Primera versió del Tileset	20
4.2.3. Segona versió del Tileset	22
4.2.4. Última versió del Tileset	22
4.2.5. Objectes animats	23
4.3. Creació del videojoc en Unity	23
4.3.1. Primer contacte amb Unity	24
4.3.2. Programant amb Unity	25
4.3.2.1. Moviment	25
4.3.2.2. Creació del mapa	25
4.3.2.3. Col·lisions	26
4.3.2.4. Teletransports	26
4.3.2.5. Animacions del personatge principal	27
4.3.2.6. Sistema de vides	28
4.3.2.7. Sistema de diàlegs	28
4.3.2.8. Sistema de atac	29
4.3.2.9. Petits detalls	29
4.3.2.10. Ampliació del mapa	30
4.3.2.11. Enemics i atacs ofensius	31

4.3.2.12. Finals	31
4.4. Banda sonora	32
5. Conclusions	32
6. Agraïments	33
7. Webgrafia	33
8. Annexos	35
8.1 Script moviment	35
8.2 Script atac	37
8.3 Script Interficie	41
8.4 Script per mostrar els objectes de la interfície	46
8.5 Script Teletransport	52
8.6 Script Dany	54
8.7 Script Curació	55
8.8 Script Diàleg	55
8.9 Script Vida 1	56
8.10 Script Vida 2	57
8.11 Script Vida 3	58
8.12 Script del moviment d'una fletxa/objecte	59
8.13 Script Destructible	60
8.14 Script d'un enemic	60
8.15 Script Arc	62
8.16 Script de quan els enemics es poden moure	63
8.17 Script de l'enemic parat	64
8.18 Script de l'alarma quan s'entra a la sala de l'enemic final	65
8.19 Script de l'animació de l'enemic final	65
8.20 Script de l'enemic final	67
8.21 Script per mostrar la vida de l'enemic final	69
8.22 Script de la programació manual dels làsers	70
8.23 Script Bomba	83
8.24 Script de la destrucció del mur secret	86

1. Introducció

L'objectiu principal del treball, és veure si és possible que una persona que no té coneixements informàtics, amb l'ajuda d'internet, seria capaç de crear un videojoc.

Un altre objectiu no serà només crear un videojoc, també és veure on arriben les meves capacitats de programació amb aquest repte.

La idea és fer un videojoc *RPG*. Primer de tot faré el *tileset*¹. Fent una recerca en diverses pàgines per veure un programa on fer aquest *tileset*, i el que més en tinc familiaritzat és *Piskel*². Per crear tot el videojoc utilitzaré *Unity 2D*. Amb l'ajuda de *Microsoft Visual Studio Code* faré tot el codi per programar el videojoc, les accions que farà el personatge, les condicions que tindrà aquest, etc. En *Unity* també

¹ Tileset: El *tileset* és el conjunt de textures, blocs i dibuixos que es conformen el mapa i els personatges, generalment el disseny del videojoc.

² Piskel: Pàgina web per fer pixel-arts online.

exportaré el tileset i faré el mapa dins, bloc a bloc. A part, també exportaré el personatge principal amb les seves animacions, que també crearé en *Piskel*. Amb el mapa i el personatge faré les col·lisions, que formarà els límits del mapa, també faré les animacions del personatge amb el seu corresponent tileset, que contindrà les seves animacions. També faré un *NPC*³ amb el que es pugui parlar i prendre decisions, com que et preguntí si el deixes anar amb ell. Finalment faré una petita lluita entre uns monstres en una zona del mapa, que t'exigeixi un altre personatge per guanyar la batalla. L'últim que faré serà la banda sonora. Com no em va donar temps a crear la meua pròpia música, he decidit agafar una sense drets d'autor per complementar el joc.

La meua principal motivació per fer aquest treball va ser la meua curiositat per veure que tan difícil és crear un videojoc amb Unity, que és un programa que ofereix una àmplia branca de possibilitats per crear qualsevol mena de joc.

Un altre motiu que em va impulsar per aquest projecte va ser el fet de veure que una persona va poder crear un videojoc dels més coneguts a dia d'ara, anomenat *Undertale*. En 2016 aquest joc va vendre més d'un milió de còpies, només en un any.

Aquesta persona va crear un joc RPG amb una història impressionant i emotiva per moltes persones, amb unes 16 rutes alternatives, amb l'estil d'un videojoc del segle XX.



Imatge 1. *Undertale*. Font: www.xatakandroid.com

³NPC: No Playable Character. És un personatge dins del joc amb el que no es pot jugar

2. Hipòtesi

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

És possible crear un videojoc sense tenir coneixement de programació, una banda sonora i els personatges, en el temps d'un mes?

Per poder respondre a aquesta hipòtesi, vaig pensar si podria fer un videojoc amb els meus recursos.

Primer he de triar un tipus de joc de la nombrosa quantitat de jocs que existeixen, i vaig arribar a la conclusió que m'agradaria crear un videojoc tipus RPG bàsic, amb diferents idees, com per exemple, parlar amb una persona dins del joc, el fet de poder tomar decisions dins i crear diferents finals segons les teves accions, o un petit sistema de batalles amb enemics

Crear un joc amb un sistema de batalles o diferents finals dins d'aquest ja és un projecte molt complicat i ambiciós amb els coneixements de programació que tinc, però vaig pensar-lo moltes vegades, i la veritat, em motiva. Crear textures per mi mateix, la meva pròpia banda sonora, crear les animacions dels personatges i obstacles.

Per finalitzar vaig haver de trobar com poder programa totes aquestes idees i mirant per internet, vaig veure Unity 2D, ja que aquest programa té totes les opcions que necessitava.

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

- És possible crear un joc dinàmic?
- Seré capaç de crear els personatges?



Imatge 2. Hipòtesi

Font: <https://pxhere.com/es/photo/1403204>

3. Els videojocs

3.1. Línea cronològica dels videojocs

3.1.1. Primeres generacions

Els videojocs han anat evolucionant al llarg del temps. Els primers es van programar en la mateixa màquina, com les clàssiques màquines recreatives. Després es van crear petites màquines que es connectaven a un televisor i tenien els seus propis cartutxos per escollir el videojoc que voldries jugar. Amb un control podies jugar als clàssics videojocs d'aquella època, com per exemple, *Tetris* o *Mario Bros*.

La primera consola va ser la *Magnavox Odyssey* (1972, primera generació), on va ser una innovació per a tothom: podies interactuar amb imatges dins d'una televisió, el que avui dia s'anomena videojoc.

L'any 1981 va sortir la *Game and Watch* (segona generació), que va ser oficialment la primera consola portàtil que va sortir al mercat, creada per Nintendo. Aquesta tenia títols famosos com *Mario Bros*, *The Legend of Zelda*, o *Donkey Kong*.

Van anar sortint noves consoles per aquestes èpoques, com la *NES* (Nintendo Entertainment System, tercera generació, 1985), que, aquesta es connectava directament a un televisor de casa i en la consola podies anar posant cartutxos per cada videojoc.



Imatge 3. Magnavox Odyssey.
Font: www.retropalace.com



Imatge 4. Game and watch.
Font: monkeygames.es



Imatge 5. Nintendo Entertainment System
Font: hipertextual.com

La *Mega Drive* (quarta generació, 1988), en canvi, va ser creada per Sega, però va triomfar en el mercat gràcies a la seva gran velocitat i millora gràfica. Van sortir títols molt famosos i clàssics a d'avui dia com *Sonic the Hedgehog*.

Posteriorment va sortir la *Game Boy* (quarta generació, 1990). Va ser una consola portàtil que funcionava amb piles, i a causa d'aquest inconvenient no va vendre molt, però després de l'arribada de títols famosos com *Tetris*, es va disparar en vendes.



Imatge 6. Mega Drive.
Font: www.vidaextra.com



Imatge 7. Game Boy
Font: www.revistagq.com

La *SNES* (Super Nintendo Entertainment System, quarta generació, 1991) va ser la successora de la *NES* i va ser competència directa de la *Mega Drive*. Aquesta consola no tenia un dels millors processadors, però els xips gràfics i els sons van fer que no sigui de gran importància. En aquesta consola van sortir seqüeles de jocs clàssics com *Super Mario World*.



Imatge 8. Super Nintendo Entertainment System (SNES)

Font: wikipedia.org

La *Amiga CD32* (cinquena generació, 1993) va ser la primera consola a implementar directament els CD en la seva videoconsola, en canvi, va quedar obsoleta i va ser un fracàs. L'empresa Commodore International (creadora de la videoconsola) va quedar en la ruïna.

Seguidament va sortir la *PlayStation* (cinquena generació, 1995), per Sony que va utilitzar el format de CD i es va ser un total èxit, però també una de les consoles més piratejades de l'època. Va tenir grans títols, però molts altres van ser poc originals.

Any després va sortir la *Nintendo 64* (cinquena generació). Aquesta no va decidir utilitzar els CD i va seguir amb el format dels cartutxos. Això va fer que fossi un fracàs per part de Nintendo.



Imatge 9. Amiga CD32

Font: karajaneadams.medium.com



Imatge 10. PlayStation

Font: es.ign.com



Imatge 11. Nintendo 64

Font: wikipedia.org

3.1.2. Innovacions generacionals

Al llarg del temps, van anar evolucionant aquestes màquines, sobretot l'empresa Nintendo, que va anar innovant noves videoconsoles, i Sony, que va anant millorant-les gambé. Sega es va retirar en 1999 amb la seva última consola, la *Dreamcast*, que també va ser un fracàs. Microsoft es va incorporar a la competència l'any 2001 amb la *Xbox*.



Imatge 12. Dreamcast. Última consola de Sega.
 Font: imgvideogame.wordpress.com



Imatge 13. Xbox. Primera consola de Microsoft.
 Font: www.3djuegos.com

A l'any 2000 va sortir la *Playstation 2* (sisena generació), que va ser una consola molt piratejada en l'època també, ja que els pirates informàtics anaven agafant experiència al llarg del temps. De totes maneres, aquesta videoconsola va portar al fracàs a Sega amb la seva *Dreamcast*, i definitivament l'empresa es va retirar de la competència. Va portar grans títols com *Grand Theft Auto San Andreas* o *Gran Turismo 4*.



Imatge 14. PlayStation 2.
 Font: www.abc.es

La *Nintendo DS* (sisena generació, 2004), va ser una consola portàtil plegable amb dues pantalles, una d'elles tàctil i s'utilitzava amb un llapis especialitzat per la consola. També va incorporar l'opció de la jugabilitat en línia. Va ser la primera videoconsola portàtil en aplicar totes aquestes característiques.

El mateix any va sortir la *PSP* (PlayStation Portable, sisena generació, 2004), creada per *Sony* i després de molt de temps, una de les poques consoles portàtils que no són de Nintendo. La *PSP*, encara tenint els seus cartutxos exclusius de *Sony*, va ser molt piratejada. Al contrari de Nintendo, aquestes consoles eren més tradicionals i eren més potents que innovadores.



Imatge 15. Nintendo DS.

Font: finalfantasy.fandom.com



Imatge 16. PlayStation Portable (PSP)

Font: wikipedia.org

La setena generació va començar amb la *Xbox 360* per part de Microsoft l'any 2005, i això li va beneficiar molt dins del mercat. A part, la qualitat gràfica ja era increïble per l'època i els jocs que tenien prometien molt.

La *Wii* (setena generació, 2006) va ser una consola molt innovadora, amb un control allargat amb què podies apuntar a la pantalla per seleccionar qualsevol opció dins de la videoconsola o jugar amb aquest. Va ser una consola molt piratejada també, però la més venuda de la generació.



Imatge 17. Xbox 360.
 Font: www.xataka.com

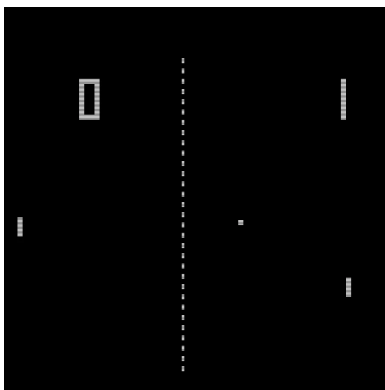
Imatge 18. Wii
 Font: muycomputer.com

Posteriorment van anar sacant més consoles de manera que la qualitat gràfica de totes les empreses ha pujat moltíssim aquests últims anys.

3.1.3. Últimes generacions i millores gràfiques

Avui dia moltes consoles han millorat moltíssim. Tant els seus components, com els videojocs. Actualment existeixen més de 100.000 títols, comptant els d'ordinador. Molts dels que existeixen ara tenen l'opció de poder jugar-los a 4k, fins i tot uns pocs amb l'opció de traçat de raigs, el *Ray Tracing*. Cada vegada van augmentant més la qualitat dels videojocs i ficant més opcions per veure d'una manera més realista aquests.

Videoconsoles com la *PlayStation 5* o *Xbox Series X* ja inclouen als seus components els nuclis dedicats al traçat de raigs per poder suportar els videojocs que inclouen aquesta opció, igual que algunes de les noves targetes gràfiques, concretament les de Nvidia. A part, els controls de les noves consoles inclouen característiques com ara sensor de moviment, micròfon, pantalla tàctil, etc.



Imatge 19. Els primers videojocs



Imatge 20. Traçat de raigs. Font: <https://www.wired.com/>

Font: <https://www.vice.com/es/>

3.2. Etiquetes dels videojocs

3.2.1. Gèneres dels videojocs

Al llarg de tota la història de la tecnologia s'han anat creant gèneres de videojocs com són els d'aventures, plataformes, misteri, etc. Ara explicaré els generals.

3.2.1.1. Acció

Aquests videojocs es caracteritzen per necessitar habilitat a l'hora de jugar, i majoritàriament la seva dificultat es progressiva al llarg del joc. Aquest és un gènere molt ampli i conformen molts dels videojocs més de moda actualment.

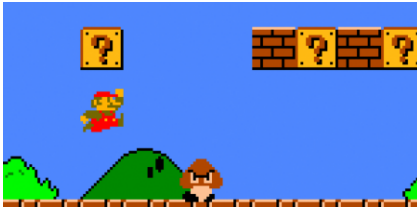
De plataformes: Aquest subgènere es caracteritza per ser jocs d'obstacles on has de intentar superar nivells fins a passar-te el joc.

Arcade: Van ser un dels primers gèneres a sorgir. Aquests es basen en nivells ràpids on has d'utilitzar la teva habilitat i els teus reflexos per aconseguir passar el nivell.

Lluita: Aquests són especialment de lluites. L'objectiu és derrotar a l'enemic amb armes, espases, poders o fins i tot a punys

Shooters: Els *shooters* són jocs d'armes a distància, semblants als de lluita.





Imatge 21. Exemple de joc de plataformes (Mario Bros)

Font: www.lavanguardia.com

Imatge 22. Exemple de shooter (Doom).

Font: steampowered.com

3.2.1.2. Estratègia

Els llocs d'estratègia usualment són llocs tàctics, on has de tomar decisions per guanyar el joc.

Estratègia a temps real: Consisteix en un joc on les decisions que tomes fan una resposta directa al temps real. Són jocs de paciència.

Tàctica: Aquest subgènere es basa en la tàctica militar.

Estratègia per torns: Com diu el nom, es basa en una estratègia per torns. Primer atakes tú, després l'enemic.

Tower Defense: Són videojocs on hi ha un camí on passen els enemics i tu has de construir torres per defensar el final del camí.

Construcció d'imperis: És simplement de construir el teu imperi administrant els teus propis recursos i tomant les teves pròpies decisions.



Imatge 23. Exemple de *Tower Defense* (Bloons TD 4)
Font: eliasbrasa.wordpress.com

Imatge 24. Exemple d'estratègia a temps real (*Age of Empires II*)
Font: www.hobbyconsolas.com

3.2.1.3. Simulació

Aquest gènere es basa en la simulació de situacions reals o fictícies, centrant-se en els detalls com els controls, els objectes del voltant, etc., de manera que puguis interactuar amb el teu entorn.



Imatge 25. Exemple de joc de simulació (*Cooking simulator*)
Font: steampowered.com

3.2.1.4. Treball mental

Aquest conjunt de videojocs es centren en els reflexes i l'enginy mental. No contenen res de violència i es treballa el cap.

Puzzle: Aquests jocs són especialment per treballar el cap i el teu enginy. Representen problemes que has de resoldre tu mateix.

Musicals: Es basen en el ritme de la música i els reflexos amb seqüències que el jugador ha de seguir.

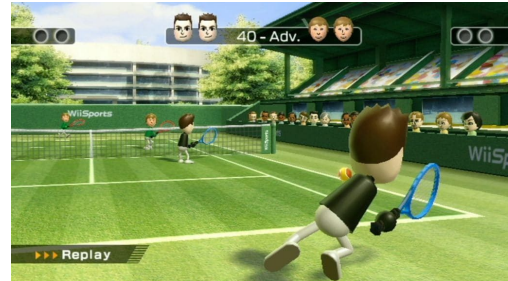
Esports: Són jocs d'esports. Simulen exercicis físics o jocs com el futbol, voleibol, etc.

Preguntes: Aquests jocs només són de preguntes que fan al jugador. Estan inspirats en programes televisius on la persona ha de sumar punts fins a obtenir la màxima puntuació.



Imatge 26. Exemple de joc musical (Guitar Hero)

Font: www.playerone.vg



Imatge 27. Exemple de joc esportiu (Wii Sports)

Font: www.nintendros.com

3.2.1.5. Aventura

Són jocs generalment amb història i mentre es va avançant el videojoc, es va desenvolupant la història.

Aventura conversacional: Són històries entre converses. El personatge principal no es mou lliurement. D'aquí vénen les novel·les visuals.

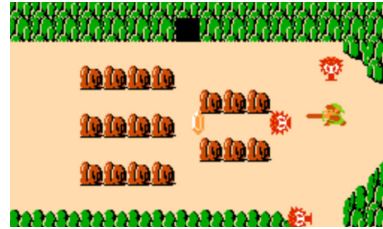
Aventura gràfica: Aquests jocs són aventures que se centren més en l'experiència gràfica, al contrari de les aventures conversacionals. Normalment hi ha món obert.

Jocs de rol (RPG): Dins d'aquest subgènere tu tries el teu personatge amb les teves habilitats i has de completar una sèrie de missions per pujar de nivell. Està basat en els jocs de rol de taula.

Aventura d'acció: És una mescla amb l'acció i l'aventura. És una aventura on necessites habilitat per superar cada zona on també hi ha història.



Imatge 28. Exemple de joc de rol (World of Warcraft)
 Font: www.vidaextra.com



Imatge 29. Exemple d'aventura d'acció (The Legend of Zelda)
 Font: www.tierragamer.com

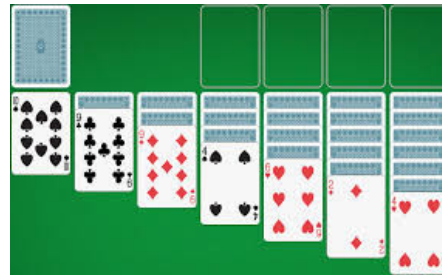
3.2.1.6. Altres subgèneres

Carreres: Aquests jocs es caracteritzen simplement per ser de carreres entre altres rivals. L'objectiu és quedar primer a la carrera amb diferents obstacles i circuits.

Jocs de taula: Com diu el nom, són inspirats en jocs de taula. Poden ser tant de cartes, de tauler, etc.



Imatge 30. Exemple de joc de carreres (Mario Kart)
 Font: wccftech.com



Imatge 31. Exemple de joc de taula (Solitario)
 Font: esports.xataka.com

3.2.2. Qualificacions dels videojocs

Abans de comprar un videojoc, ens hem de fixar de les seves qualificacions. Això inclou: l'edat recomanada per jugar-lo, si hi ha sang, escenes sexuals, etc. En

Europa s'utilitza el *Pegi* per dir l'edat recomanada d'un videojoc. Ara explicaré cada una de les etiquetes que et pots trobar en la informació d'un videojoc en Europa.



Pegi 3: L'edat mínima és de tres anys. Normalment aquesta etiqueta es posa quan no hi ha res dolent en el videojoc, de manera que un nen petit pot veure'l sense cap problema.



Pegi 7: L'edat mínima és de set anys. Potser en aquests videojocs ja surt una mica de violència o petits sorolls que poden espantar a nens petits.



Pegi 12: L'edat recomanada és de majors de dotze anys. Poden sortir més sorolls forts, una mica més de violència i potser paraules dolentes.



Pegi 16: En el videojoc ja surt bastant més violència i les paraules dolentes poden ser constants. Pot sortir sang i potser escenes sexuals censurades. Les activitats il·legals poden ser constants.



Pegi 18: És per majors de anys. Pot sortir sang i escenes sexuals explícites sense censura. Les activitats il·legals són comuns. Tota escena amb tràfic de drogues, armes, i tortures hi són permeses.

Imatge 32. Símbols de *Pegi*.

Font: <https://pegi.info/es>

4. Elaboració del videojoc

4.1. Introducció del videojoc

La història del videojoc és una part molt important, ja que és la part on interactuen amb l'usuari, si la història no ens motiva, farà que el nivell d'interès del joc sigui inferior, per altra part el joc tindrà quatre finals, perquè quan acabi sigui possible

jugar per veure un altre final. Personalment, penso que aquests detalls són els que fan triomfar un videojoc.

4.1.1. Història del videojoc

A l'illa principal hi viuen feliçment dos germans. El protagonista s'adona que a l'est de la seva casa n'hi ha una separació de terra que formarà un terratrèmol, i a partir d'aquí té dues diferents opcions: o informar el seu germà, o escapar-se ell sòl.

Si vol informar sobre el succés al germà, aquest li donarà un arc i una espasa per defensar-se ell sòl, i el germà intentarà buscar una altra manera de sortir. Haurà d'entrar al castell i derrotar al drac que està defensant la sortida del castell.

Quan el protagonista derroti al drac, aquest podrà sortir de l'illa, i es retrobarà posteriorment amb el seu germà, on viuran feliços per sempre.

Si no vol informar el seu germà, aquest haurà de passar el castell també. Amb una roca que hi ha fora del castell, haurà de matar als enemics i derrotar al drac, on també podrà sortir de l'illa. Quan passi això, sortirà com que el protagonista ha aconseguit sortir, però el seu germà ha mort perquè no va ser informat del succés.

Hi ha dos altres finals; dins del castell n'hi haurà un camí secret que només es pot accedir amb una bomba. Aquesta bomba apareixerà quan derrotis tots els enemics del castell, exceptuant el drac. Posteriorment apareixerà la bomba al principi del castell i tindràs l'opció d'agafar-la i explotar-la en on estigui el camí secret. El final depèn de si has informat al teu germà o no, però la diferència és que el drac segueix viu i aconsegueix sortir de l'illa. Va ser despertat pel soroll de la bomba.

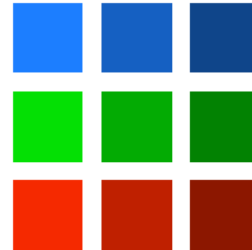
4.2. Tileset

Primer de tot vaig haver de fer un *Tileset*. Un *Tileset* és tota la part gràfica del videojoc. És com una paleta per anar construint les diferents zones del videojoc, totes les animacions que tindrà aquest, personatges, decoració, etc. Vaig decidir fer

aquest *Tileset* en una pàgina anomenada *Piskel*. Aquesta és una pàgina per fer crear *Píxel-Arts* d'una manera molt visual i senzilla.



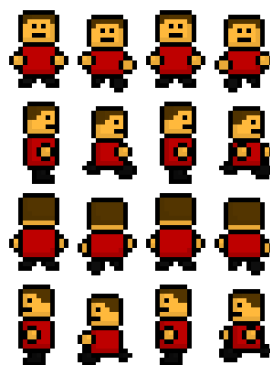
Imatge 33. Exemple d'un *Tileset*.
Font: <https://www.deviantart.com/>



Imatge 34. Logo de *Piskel*.
Font: <https://www.piskelapp.com/>

4.2.1. Personatge principal

La creació dels personatges i les seves corresponents animacions per caminar cap a dalt, a baix, dreta i esquerra. Vaig fer un personatge bastant senzill, perquè no volia que em tomés molt temps fer les animacions, ja que s'hauria de repetir aquest personatge moltes vegades.



Imatge 35. Personatge principal

4.2.2. Primera versió del *Tileset*

Després de la creació del personatge i em vaig posar a crear la paleta per construir tot el mapa.

Em vaig fer una idea de com seria el mapa, un esquema mental. Tindria el punt d'inici. Aquest seria una mena d'illa que portés a tres diferents llocs. Per la dreta n'hi hauria un camí on es pugui accedir a una cova secreta, empenyent una roca. Aquesta cova només tindrà un cartell que digui que no serveix per a res. Seguint el camí n'hi haurà una zona que hagués passat una mena de terratrèmol, o un gran canyó. Això serà la zona de la dreta. A dalt del punt d'inici hi haurà un camí que portés a la casa del teu germà, que puguis parlar amb ell i interactuï amb la història del videojoc. Fora de la casa hi haurà una pilota per jugar una mica. Per l'esquerra del punt d'inici hi haurà un pont que porti a una zona perillosa, concretament un castell. Aquest castell tindrà diferents zones, amb una lluita amb un monstre gegant. Després de derrotar aquest monstre, el joc termina. Vaig començar a fer aquesta paleta. Em va trigar unes 3 hores la primera versió del *Tileset*, que s'anirà actualitzant al pas del temps.

Aquesta primera versió, tenia el més bàsic: blocs d'herba, aigua, camins de sorra, camins de terra, unes tanques, el pont, una pilota, una roca, i la casa per fora.



Imatge 36. Primera versió del *Tileset*.

Vaig intentar crear el mapa amb aquest *Tileset*, però faltaven moltes coses. Vaig crear el punt d'inici, el pont de l'esquerra, el camí i la casa del nord, però era insuficient per continuar.

4.2.3. Segona versió del *Tileset*

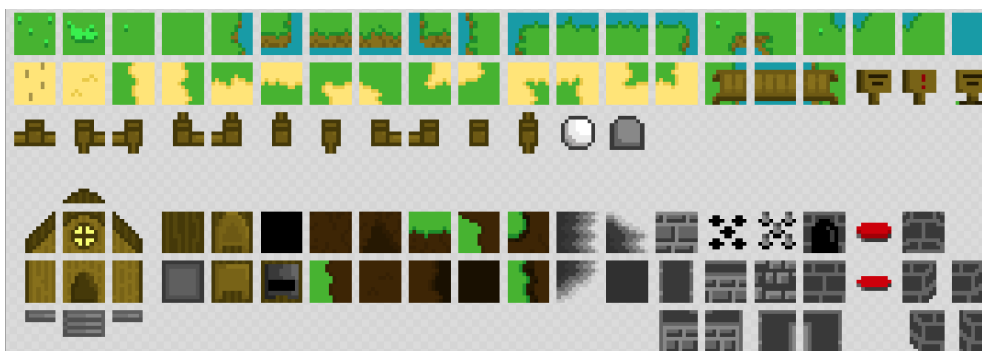
Era necessari afegir més coses al *Tileset* si volia crear el que volia realment fer. La segona versió va ser afegir tot l'anterior, l'interior de la casa, parets de terra per la cova, una zona de terra per la zona del terratrèmol i uns cartells per donar informació al jugador.



Imatge 37. Segona versió del *Tileset*.

4.2.4. Última versió del *Tileset*

Bé, aquesta versió va ser creada per detalls finals o alguns objectes que vaig agregar els últims dies que vaig estar creant el videojoc, especialment el castell. Vaig afegir tot el necessari per construir-lo, i també vaig afegir un conjunt de pinxos de sòl i un gran botó vermell.



Imatge 38. Última versió del *Tileset*.

4.2.5. Objectes animats

Durant el procés de creació del *Tileset* i el videojoc, vaig necessitar detalls que complementessin l'espai, o directament personatges, objectes o enemics que donessin dificultat al joc o afectessin directament a la història d'aquest. Principalment vaig crear el *NPC*, els enemics, el drac, uns arbustos destructibles, els atacs del drac, les vides, una bomba, i també totes les animacions del personatge principal disparant o atacant.

El que més se'm va complicar van ser les animacions d'atac o disparar, perquè quan l'exportava al *Unity*, em sortien de diferent mesura, i quan intentava canvia-les, no feia res. Vaig haver de crear aquestes animacions amb el mateix nombre de píxels que el personatge principal base. Em va sortir pitjor, però ja funcionava. En alguns dibuixos vaig posar ratlles per guiar-me amb els límits de la imatge i que tots siguin iguals perquè no sortissin malament les animacions.



Imatge 39. Alguns personatges, enemics i objectes animats del joc

4.3. Creació del videojoc en Unity

Després de crear la primera versió del *Tileset*, vaig començar a exportar-lo tot al *Unity*. Realment no sabia com començar el videojoc, i em vaig decantar per entendre una mica el codi.

4.3.1. Primer contacte amb *Unity*

Primer de tot vaig veure vídeos de com funcionava *Unity* i el seu codi. Aquesta va ser l'etapa d'informar-se de com funciona tot, perquè jo sabia que si no entenia com funcionava el programa, se m'aniria a complicar moltíssim el treball, ja que hauria de dependre d'internet. Vaig veure vídeos de com funcionaven els objectes i els components bàsics que podia tenir un, com els cossos rígids, les capses de col·lisions, els renderitzadors de sprites⁴ o els scripts⁵.

Després d'informar-me una mica, vaig aconseguir retallar el *Tileset* automàticament de manera que tots els objectes de la imatge tinguin la mateixa mesura, i vaig començar a programar per veure si tot anava bé.

Primer de tot vaig provar de fer moviment en un objecte. A partir d'aquell punt, si em va bé, podré començar a programar el videojoc. Vaig crear un script, i el vaig intentar obrir, però no feia res. Vaig navegar per internet per veure una possible solució, i després d'estar unes hores intentant solucionar-lo, vaig trobar quin era el problema: resulta que el programa on s'havia de posar el codi, no s'havia instal·lat, i tampoc es podia. Vaig provar d'instal·lar un altre i aparentment aquest anava, però quan copiava el codi, no feia res sobre l'objecte i no sortia cap error a la consola. Vaig provar amb un tercer programa, el *Microsoft Visual Studio Code*, i anava perfectament. Vaig trigar entre 6 i 8 hores en solucionar aquest error.



Imatge 40. Logo de Microsoft Visual Studio.
Font: <https://www.eleconomista.es/>

⁴ sprite: Cadascun dels dibuixos del *Tileset*.

⁵ script: Document per programar les línies de codi.

4.3.2. Programant amb *Unity*

Després de solucionar aquest error, em vaig animar a començar a programar el videojoc.

4.3.2.1. Moviment

Primer de tot havia de crear moviment. Aquests primers scripts vaig haver de veure com funcionaven pels vídeos, perquè realment no entenia res del codi.

El script era senzill, funcionava de la següent manera: els quatre botons de moviment són les fletxes de baix a la dreta del teclat, o també es pot utilitzar els botons W, A, S, D. En un vector de dues dimensions, si un d'aquests botons s'està pressionant, el personatge es mourà. Si s'està pressionant el de la dreta, el cos s'actualitzarà a la posició $x+1$ ⁶, si s'està pressionant de l'esquerra, el cos es mourà a $x-1$, si s'està pressionant el de dalt, el cos es mourà a $y+1$ i finalment si s'està pressionant el de baix, el cos es mourà a $y-1$. En el mateix script he creat la velocitat, que només multiplicaria la distància que recorre per la velocitat.

4.3.2.2. Creació del mapa

Després de veure que el cos es podia moure i els scripts anaven perfectament, vaig crear el mapa amb el *Tileset* que tenia. Primer ho vaig provar amb un programa extern, que era molt més fàcil, però em va donar problemes.

Amb el poc que tenia, vaig crear el punt d'inici, anomenat *Spawn*, on el personatge apareixeria. Pel nord, vaig crear un camí que arribaria a la casa. Per l'oest vaig crear el pont i seguint-lo, una zona salvatge on hi hagi arbustos o herba alta. No vaig poder crear més amb aquest *Tileset*, i el vaig millorar.

⁶ x / y : Són les dues coordenades d'un moviment de dos vectors. x es la coordenada horitzontal, i y , la vertical.

Després de fer la segona versió del *Tileset*, vaig crear l'interior de la casa, i una zona per l'oest on hi hauria una cova, i més a la dreta, l'origen del futur terratrèmol. Amb això ja tenia prou en aquell moment.

4.3.2.3. Col·lisions

Aquesta part va ser senzilla, només havia de col·locar objectes estàtics dins del mapa que tinguin caps de col·lisions i que el personatge principal tingui un cos rígid dinàmic, ja que es mou.

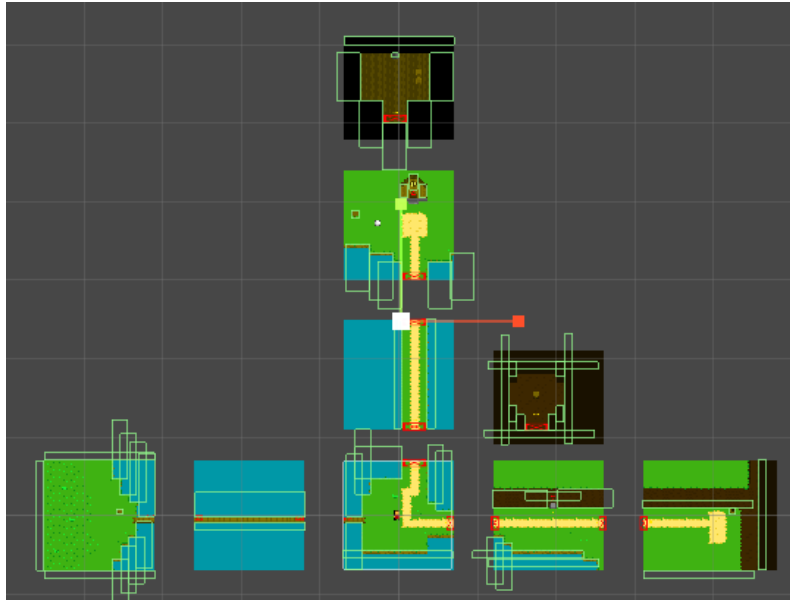
Posteriorment això em va donar problemes perquè va haver-hi moments que el personatge evitava les col·lisions, però no em va portar molt de temps arreglar aquests errors. Normalment eren per algun detall que em despistava.

4.3.2.4. Teletransports

El meu mapa es basa en teletransports. Si el jugador toca una cantonada o un costat del mapa, aquest es teletransportarà a la zona del costat corresponent. Això va ser molt complicat per al meu nivell d'informàtica, perquè no sabia controlar els objectes, i vaig haver de mirar una altra vegada internet.

El sistema és el següent: N'hi ha dos objectes en cada punt de teletransport, un petit, i un gran. L'objecte gran és el punt en el qual el jugador ha de tocar per teletransportar-se, i l'altre petit és el punt on el jugador apareixerà quan sigui teletransportat a aquella zona. Bé, aquests punts de teletransport tindran una caps de col·lisions de tipus desencadenant; significa que, si es toca, amb una línia d'arguments de codi, pot desencadenar una sèrie d'ordres escrites en un script. Si el jugador toca el punt gran, el teletransportarà a la posició del punt petit de l'altre costat. Sempre hi hauran d'haver dos punts de teletransport enllaçats entre ells.

Seguidament vaig fer que quan un teletransport s'utilitzés, es posi la pantalla negra, es teletransportés, i torni la pantalla, també parant el script de moviment.



Imatge 41. Mapa amb les col·lisions i teletransports.

Vaig intentar que la càmera no sortís dels límits del joc, però amb el mètode que vaig utilitzar per crear el mapa se'm va fer impossible.

4.3.2.5. Animacions del personatge principal

Les animacions les vaig entendre ràpidament, ja que el sistema que utilitza *Unity* per crear-les és molt visual.

Per crear les animacions de moviment, simplement s'havia de posar una variable que digués si estava caminant o no, i si estava caminant, amb unes altres dues variables (x, y) dir la posició on s'estava movent, semblant al mètode de moviment que vaig utilitzar.

Posteriorment utilitzaré aquest mateix mètode per crear les animacions d'atac i disparar, agregant altres variables per reconèixer si el personatge està atacant o disparant.

4.3.2.6. Sistema de vides

El sistema de vides va ser el primer sistema que vaig crear per mi mateix, ja començava a dominar el codi.

Per crear aquest sistema de vides havia de fer diverses coses. Primer, evidentment, declarar una variable de nombre enter que siguin les vides. El meu personatge tindrà tres vides. Vaig crear un script on registrés tot això i els canvis que faria al llarg del joc. També vaig crear un script que si un objecte de tipus desencadenant toqués al personatge, aquest perdi una vida. Seguidament vaig posar que quan es perdin les tres vides, reutilitzant la funció de la pantalla en negra del script de teletransportació, que es posi tot en negre. Finalment vaig crear una pantalla de *Game Over*, i que s'hagi de pressionar la R per reiniciar el joc.

4.3.2.7. Sistema de diàlegs

Aquest vaig haver que mirar per internet per com fer-ho, perquè no sabia res de posar textos i diàlegs.

Funciona de la següent manera: es declara una variable pública múltiple (en programació anomenada *array*) que siguin les frases que el personatge vagi a posar. Si un personatge entra a una àrea en concret, el diàleg comença, i cada vegada que es pulsi la barra d'espai del teclat, continuarà a la frase següent. Si no queden frases, no passarà de frase.



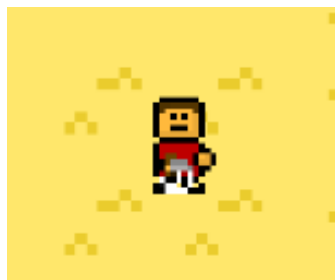
Imatge 42. Diàleg amb el NPC.

Imatge 43. Diàleg amb un cartell.

4.3.2.8. Sistema de atac

Vaig haver de mirar per internet una mica, sobretot per l'arc i la rotació de la fletxa a l'hora de disparar.

Si el personatge polsa la lletra E, el personatge farà una animació d'atac i durant un temps determinat farà aparèixer una àrea d'atac mentre passa l'animació. L'arc és més complex. Cada vegada que es pulsi la R, el personatge farà una animació de tensió amb l'arc, i si es deixa de pressionar, farà una altra animació de llençar la fletxa. Bé, per llençar la fletxa, s'ha de crear una fletxa base, i el jugador el que farà es duplicar aquesta fletxa tantes vegades com es llanci. La fletxa base no es llençarà, només farà còpies d'aquestes. Depèn de l'angle en el qual estiguis mirant, el programa el detectarà i duplicarà la fletxa mirant a aquella direcció, i per llençar-la, només ha de seguir aquella direcció on està mirant.



Imatge 44. Atac amb espasa.



Imatge 45. Atac amb arc.

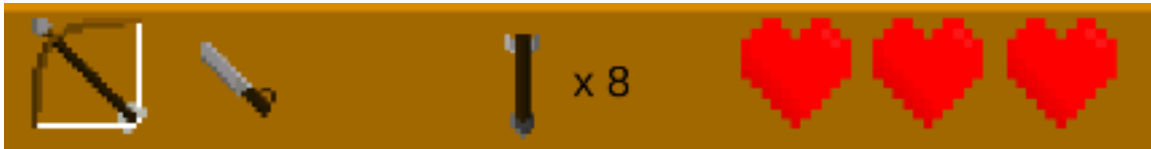
4.3.2.9. Petits detalls

Els detalls en un joc són importants, i a part, poden facilitar la jugabilitat la persona que el descarregui. Aquestes coses secundàries poden ser fonamentals en un videojoc tan senzill.

Primerament vaig posar que quan parles amb el teu germà dins del joc i et doni tot l'equipament, et surti en una mena d'interfície i la quantitat de fletxes que tens.

Un altre detall que vaig posar, es una cova que has d'empènyer una roca per introduir-te.

També vaig posar arbustos destructibles que, si els hi dónes amb l'espasa o amb una fletxa, es trenquen.

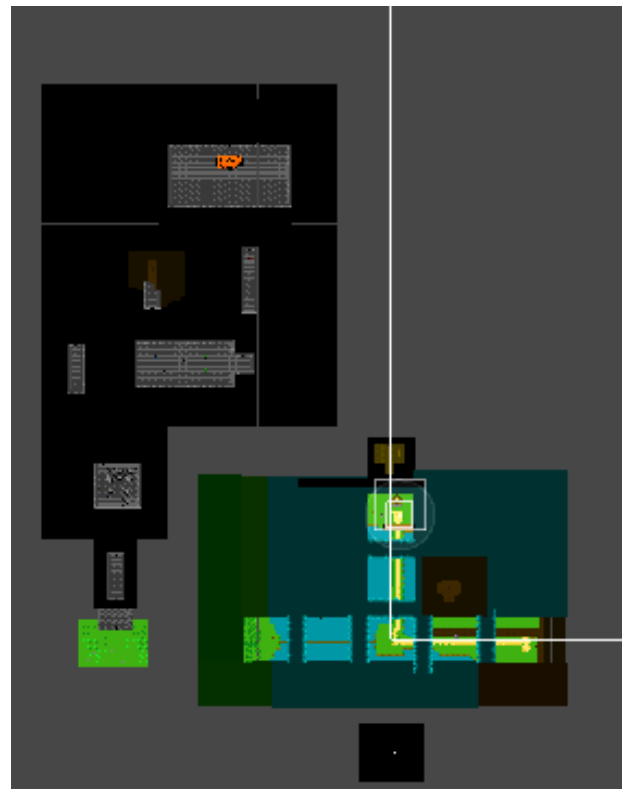


Imatge 46. Interfície del joc.

4.3.2.10. Ampliació del mapa

Després de tot això, vaig pensar que el joc necessitava un objectiu, una història complexa, i un final. Vaig decidir fer un castell per la zona de l'est on hi haurà diferents sales amb diferents obstacles. En aquest moment vaig fer servir la versió final del *Tileset*.

Una sala seria de trampes. Tindria fletxes i pinxos a terra. Una altra sala seria de monstres, que tinguis que derrotar-los sigui gastant fletxes o amb l'espasa. I l'última sala que faria seria la de l'enemic final, que seria un drac i que la seva dificultat sigui progressiva.



Imatge 47. Mapa final del joc.

4.3.2.11. Enemics i atacs ofensius

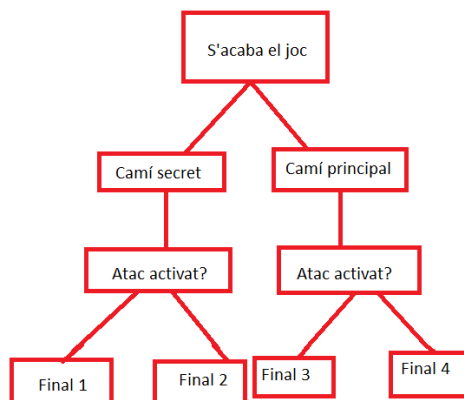
Una sala del castell vaig decidir que tindria enemics. Aquests et quitarien un cor només un, però si t'ataquen tots de cop et poden matar, i per això el jugador ha d'evitar que s'agrupin tots si els vol matar amb l'espasa.

Això va ser molt senzill. Només havia de crear un monstre amb animació (només vaig fer animació caminant a la dreta i esquerra) i que aquest es mogui cap al jugador. Si la coordenada X es negativa, l'animació seria cap a l'esquerra, i si la coordenada X es positiva, cap a la dreta. Vaig posar una col·lisió de tipus desencadenant perquè els monstres es mogui només si el jugador entra a la sala, perquè no s'acumulin abans d'arribar.

L'enemic final es basa en làsers i boles de foc. Tots els atacs els he programat jo de manera que la dificultat sigui progressiva al llarg de la batalla.

4.3.2.12. Finals

Quan el jugador arriba a un dels dos punts finals del joc, el programa detecta si té l'atac activat. Si el té activat, significa que ha parlat amb el NPC. D'aquesta manera, el joc es pot dividir en quatre finals diferents



Imatge 48. Esquema dels quatre possibles finals.

4.4. Banda sonora

De banda sonora no vaig tenir gens de temps per crear la música per mi mateix amb un programa per fer-la, així doncs, vaig haver de buscar música per internet sense drets d'autor. Vaig trobar una pàgina molt bona per descarregar música gratuïtament se manera legal, d'una persona anomenada Patrick d'Arteaga i em va ajudar a buscar música pels entorns del videojoc, com per exemple, pel castell o la batalla contra l'enemic final.

5. Conclusions

Bé, després de tot aquest treball he arribat a diverses conclusions, comptant la resposta a la hipòtesi.

És possible crear un videojoc sense tenir coneixement de programació, una banda sonora i els personatges, en el temps d'un mes?

La resposta és que sí. Internet és l'eina més gran del món, i amb molta paciència s'hi pot. N'hi ha moltíssims tutorials i cursos per YouTube o pàgines web explicant com funciona, i si tens qualsevol error, sempre pots buscar el motiu d'aquest.

Aproximadament he trigat unes 100-120 hores en fer aquest projecte, perquè realment em va motivar la idea de fer un videojoc, però aquest temps va limitar moltes coses que vaig pensar a fer les últimes setmanes. Realment un dissenyador i programador de videojocs que ja té un nivell de fama elevat pels seus videojocs, està sotmès a aquesta situació, ja que un videojoc necessita actualitzacions per mantenir al seu públic.

En aquest projecte també he après que la programació és lògica. La majoria de les línies de codi treballen amb funcions lògiques, i això em va ajudar a entendre fàcilment com és programar un videojoc.

6. Agraïments

He d'agrair al meu tutor, Jaime, per donar-me consells de com anar fent el projecte i per anar corregint-me el treball, o si podria estar millor.

Vull també agrair especialment a la meva família per animar-me a donar-me suport al llarg de tota la creació del videojoc. Realment no em vaig imaginar que fossi capaç de terminar aquest gran projecte ni arribar tan lluny com he arribat avui.

I també vull agrair a totes aquelles persones que han servit de banc de recursos per aprendre tot el que he après en aquest projecte. No m'esperava aprendre tan d'informàtica com he après amb la creació del meu propi videojoc.

7. Webgrafia

Fonts utilitzades per les imatges:

Imatge 1 - Undertale. Consultada el 2 d'abril de 2021 - www.xatakandroid.com

Imatge 2 - Hipòtesi. Consultada el 25 de febrer de 2021 -

<https://pxhere.com/es/photo/1403204>

Imatge 3 - Magnavox Odyssey. Consultada el 8 d'abril de 2021 -

www.retroplace.com

Imatge 4 - Game and Watch. Consultada el 8 d'abril de 2021 - monkeygames.es

Imatge 5 - Nintendo Entertainment System. Consultada el 8 d'abril de 2021 -

hipertextual.com

Imatge 6 - Mega Drive. Consultada el 8 d'abril de 2021 - www.vidaextra.com

Imatge 7 - Game Boy. Consultada el 8 d'abril de 2021 - www.revistagq.com

Imatge 8 - Super Nintendo Entertainment System. Consultada el 8 d'abril de 2021 - [wikipedia.org](https://www.wikipedia.org)

Imatge 9 - Amiga CD32. Consultada el 8 d'abril de 2021 - [karajaneadams.medium.com](https://www.karajaneadams.medium.com)

Imatge 10 - PlayStation. Consultada el 8 d'abril de 2021 - es.ign.com

Imatge 11 - Nintendo 64. Consultada el 8 d'abril de 2021 - [wikipedia.org](https://www.wikipedia.org)

Imatge 12 - Dreamcast. Consultada el 8 d'abril de 2021 - imgvideogame.wordpress.com

Imatge 13 - Xbox. Consultada el 8 d'abril de 2021 - www.3djuegos.com

Imatge 14 - PlayStation 2. Consultada el 8 d'abril de 2021 - www.abc.es

Imatge 15 - Nintendo DS. Consultada el 8 d'abril de 2021 - finalfantasy.fandom.com

Imatge 16 - PlayStation Portable. Consultada el 8 d'abril de 2021 - [wikipedia.org](https://www.wikipedia.org)

Imatge 17 - Xbox 360. Consultada el 8 d'abril de 2021 - www.xataka.com

Imatge 18 - Wii. Consultada el 8 d'abril de 2021 - muycomputer.com

Imatge 19 - Els primers videojocs. Consultada el 10 de març de 2021 - www.vice.com

Imatge 20 - Traçat de raigs. Consultada el 10 de març de 2021 - www.wired.com

Imatge 21 - Exemple de joc de plataformes (Mario Bros). Consultada el 7 d'abril de 2021 - www.lavanguardia.com

Imatge 22 - Exemple de shooter (Doom). Consultada el 7 d'abril de 2021 - steampowered.com

Imatge 23 - Exemple de *Tower Defense* (Bloons TD 4). Consultada el 7 d'abril de 2021 - eliasbrasa.wordpress.com

Imatge 24 - Exemple d'estratègia a temps real (Age of Empires II). Consultada el 7 d'abril de 2021 - www.hobbyconsolas.com

Imatge 25 - Exemple de joc de simulació (Cooking simulator). Consultada el 7 d'abril de 2021 - steampowered.com

Imatge 26 - Exemple de joc musical (Guitar Hero). Consultada el 7 d'abril de 2021 - www.playerone.vg

Imatge 27 - Exemple de joc esportiu (Wii Sports). Consultada el 7 d'abril de 2021 - www.nintenderos.com

Imatge 28 - Exemple de joc de rol (World of Warcraft). Consultada el 7 d'abril de 2021 - www.vidaextra.com

Imatge 29 - Exemple d'aventura d'acció (The Legend of Zelda). Consultada el 7 d'abril de 2021 - www.tierragamer.com

Imatge 30 - Exemple de joc de carreres (Mario Kart). Consultada el 7 d'abril de 2021 - wccftech.com

Imatge 31 - Exemple de joc de taula (Solitario). Consultada el 7 d'abril de 2021 - esports.xataka.com

Imatge 32 - Símbols de Pegi. Consultada el 7 d'abril de 2021 - <https://pegi.info/es>

Imatge 33 - Exemple d'un *Tileset*. Consultada el 19 de març de 2021 - <https://www.deviantart.com/>

Imatge 34 - Logo de *Piskel*. Consultada el 19 de març de 2021 - <https://www.piskelapp.com/>

Imatge 40 - Logo de Microsoft Visual Studio. Consultada el 20 de març de 2021 - <https://www.eleconomista.es/>

Fonts utilitzades el dia 7 d'Abril de 2021 per fer la part teòrica dels gèneres dels videojocs:

https://videojuegos.fandom.com/es/wiki/G%C3%A9neros_de_videojuegos#Deportes

<https://www.3dcube.es/generos-de-videojuegos/>

Fonts utilitzades el dia 8 d'Abril de 2021 per fer la part teòrica de l'evolució de les videoconsoles:

https://www.elotrolado.net/wiki/Evoluci%C3%B3n_de_las_Consolas

<https://www.gaboxreviews.site/opinion/las-consolas-mas-pirateadas-de-la-historia/>

<https://es.wikipedia.org/wiki/Videoconsola>

8. Annexos

8.1 Script moviment

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;
```

```

public class Movimiento : MonoBehaviour
{
    Animator anim;

    new Rigidbody2D rigidbody;

    Vector2 mov;

    public float speed = 4f;

    CircleCollider2D espadapos;

    void Start()
    {
        espadapos = transform.GetChild(0).GetComponent<CircleCollider2D>();

        espadapos.enabled = false;

        anim = GetComponent<Animator>();

        rigidbody = GetComponent<Rigidbody2D>();

        GetComponent<Ataque>().enabled = false;
    }

    void Update()
    {
        mov = new Vector2(
            Input.GetAxisRaw("Horizontal"),
            Input.GetAxisRaw("Vertical")
        );
        if (mov != Vector2.zero)
        {
            anim.SetFloat("movX", mov.x);
            anim.SetFloat("movY", mov.y);
            anim.SetBool("walking", true);
        }
    }
}

```

```

    }
    else
    {
        anim.SetBool("walking", false);
    }
}

void FixedUpdate()
{
    rigidbody.MovePosition(rigidbody.position + mov * speed * Time.deltaTime);
}
}

```

8.2 Script atac

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Ataque : MonoBehaviour
{
    Vector2 mov;

    Animator anim;

    public GameObject flechaPrefab;

    new Rigidbody2D rigidbody;

    CircleCollider2D espadapos;

    public float speed = 4f;
}

```

```

public static int flechas = 10;

void Start()
{
    espadapos = transform.GetChild(0).GetComponent<CircleCollider2D>();

    espadapos.enabled = false;

    anim = GetComponent<Animator>();

    rigidbody = GetComponent<Rigidbody2D>();
}

// Update is called once per frame
void Update()
{
    Movimiento();

    Animaciones();

    AtaqueEspada();

    if (flechas > 0) Disparo();
}

void FixedUpdate()
{
    rigidbody.MovePosition(rigidbody.position + mov * speed * Time.deltaTime);
}

void Movimiento()
{
    mov = new Vector2(

```

```

    Input.GetAxisRaw("Horizontal"),
    Input.GetAxisRaw("Vertical")
);
}

void Animaciones()
{
    if (mov != Vector2.zero)
    {
        anim.SetFloat("movX", mov.x);
        anim.SetFloat("movY", mov.y);
        anim.SetBool("walking", true);
    }
    else
    {
        anim.SetBool("walking", false);
    }
}

void AtaqueEspada()
{
    if (Input.GetKeyDown(KeyCode.E))
    {
        anim.SetBool("atacking", true);
    }
    else
    {
        anim.SetBool("atacking", false);
    }
    if (mov != Vector2.zero) espadapos.offset = new Vector2(mov.x / 2, mov.y / 2);

    AnimatorStateInfo stateInfo = anim.GetCurrentAnimatorStateInfo(0);
    bool atacking = stateInfo.IsName("Atacando");

```



```

    if (atacking)
    {
        float playbackTime = stateInfo.normalizedTime;
        if (playbackTime > 0.66 && playbackTime < 1)
        {
            espadapos.enabled = true;
        }
        else espadapos.enabled = false;
    }
}

void Disparo()
{
    AnimatorStateInfo stateInfo = anim.GetCurrentAnimatorStateInfo(0);
    bool charging = stateInfo.IsName("Cargando_disparo");

    if (Input.GetKeyDown(KeyCode.R))
    {
        anim.SetTrigger("charging");
    } else if (Input.GetKeyUp(KeyCode.R))
    {
        anim.SetTrigger("shooting");
        float angle = Mathf.Atan2(
            anim.GetFloat("movY"),
            anim.GetFloat("movX")
        ) * Mathf.Rad2Deg;
        GameObject Flechaobj = Instantiate(
            flechaPrefab, transform.position,
            Quaternion.AngleAxis(angle, Vector3.forward));
        Disparo flecha = Flechaobj.GetComponent<Disparo>();
        flecha.mov.x = anim.GetFloat("movX");
        flecha.mov.y = anim.GetFloat("movY");
    }
}

```

```
        flechas = flechas - 1;
    }
}
}
```

8.3 Script Interfície

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;
```

```
public class Interfaz : MonoBehaviour
{
```

```
    public static int vida = 3;
```

```
    bool start = false;
```

```
    bool isFadeln = false;
```

```
    float alpha = 0;
```

```
    float fadeTime = 1f;
```

```
    public Transform Spawn;
```

```
    public Transform Muerte;
```

```
    public GameObject Objeto;
```

```
    public float speed = 100000;
```

```
private Vector3 Mover;

private bool confirmation = false;

public TextMeshProUGUI displayArrows;

public static bool modogenocida = false;

public static int numberz = 0;

public static bool bomba;

static AudioSource musicaCastillo;

static AudioSource musicaBoss;

static AudioSource musicaFinal;

private void Start()
{
    musicaCastillo = transform.GetChild(4).GetComponent<AudioSource>();

    musicaBoss = transform.GetChild(5).GetComponent<AudioSource>();

    musicaFinal = transform.GetChild(6).GetComponent<AudioSource>();

    musicaCastillo.enabled = false;

    musicaBoss.enabled = false;

    musicaFinal.enabled = false;
```

```

        Mover = Muerte.position;
    }
    IEnumerator Restart()
    {
        FadeIn();
        GetComponent<Movimiento>().enabled = false;
        yield return new WaitForSeconds(1);
        Objeto.transform.position = Vector3.MoveTowards(Objeto.transform.position,
Mover, speed * Time.deltaTime);
        FadeOut();
        confirmation = true;
    }
    private void Update()
    {
        if (Input.GetKey(KeyCode.R) && confirmation == true)
        {
            Mover = Spawn.position;
            vida = 3;
            Objeto.transform.position = Vector3.MoveTowards(Objeto.transform.position,
Mover, speed * Time.deltaTime);
            SceneManager.LoadScene("SampleScene");
            GetComponent<Movimiento>().enabled = true;
            Ataque.flechas = 10;
            confirmation = false;
        }

        if (modogenocida == true) GetComponent<Ataque>().enabled = true;

        if (GetComponent<Ataque>().enabled == true) modogenocida = true;

        if (GetComponent<Ataque>().enabled == true)
        {

```

```

        displayArrows.text = "x" + " " + Ataque.flechas;
    }
    else
    {
        displayArrows.text = " ";
    }

    if(Input.GetKeyDown(KeyCode.Escape))
    {
        Application.Quit();
    }
    if (vida <= 0)
    {
        StartCoroutine("Restart");
    }
}

private void OnTriggerEnter2D(Collider2D col)
{
    if (col.CompareTag("MusicaCastillo"))
    {
        GetComponent<AudioSource>().enabled = false;
        musicaCastillo.enabled = true;
    }
    if (col.CompareTag("Anim_boss"))
    {
        musicaCastillo.enabled = false;
        musicaBoss.enabled = true;
    }
}

public void OnGUI()
{

```

```

if (!start)
    return;

GUI.color = new Color(GUI.color.r, GUI.color.g, GUI.color.b, alpha);

Texture2D tex;
tex = new Texture2D(1, 1);
tex.SetPixel(0, 0, Color.black);
tex.Apply();

GUI.DrawTexture(new Rect(0, 0, Screen.width, Screen.height), tex);

if (isFadeIn)
{
    alpha = Mathf.Lerp(alpha, 1.1f, fadeTime * Time.deltaTime);
}
else
{
    alpha = Mathf.Lerp(alpha, -0.1f, fadeTime * Time.deltaTime);

    if (alpha < 0) start = false;
}
}
public void FadeIn()
{
    start = true;
    isFadeIn = true;
}
public void FadeOut()
{
    isFadeIn = false;
}

```

}

8.4 Script per mostrar els objectes de la interfície

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DisplayInfo : MonoBehaviour
{
    SpriteRenderer flechaprincipal;
    SpriteRenderer arco;
    SpriteRenderer cuchillo;
    SpriteRenderer bomba;

    // Update is called once per frame
    void Update()
    {
        flechaprincipal = transform.GetChild(0).GetComponent<SpriteRenderer>();
        arco = transform.GetChild(1).GetComponent<SpriteRenderer>();
        cuchillo = transform.GetChild(2).GetComponent<SpriteRenderer>();
        bomba = transform.GetChild(3).GetComponent<SpriteRenderer>();

        if (Interfaz.modogenocida == true)
        {
            flechaprincipal.enabled = true;
            arco.enabled = true;
            cuchillo.enabled = true;
        }
        else
        {
            flechaprincipal.enabled = false;
            arco.enabled = false;
        }
    }
}
```

```
        cuchillo.enabled = false;
    }

    if (Bomba.bum == true && Interfaz.vida > 0) bomba.enabled = true;
    else bomba.enabled = false;
}
}
```

Script Control del diàleg

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class ControlDialogo : MonoBehaviour
{
    public Dialogo dialogue;

    Queue<string> sentences;

    public GameObject dialoguePanel;

    public TextMeshProUGUI displayText;

    public GameObject Mapa1;

    public GameObject Mapa2;

    public GameObject Mapa3;

    public GameObject Mapa4;
```



```
public Transform Final1;

public Transform Final2;

public Transform Final3;

public Transform Final4;

string activeSentence;

public float typingSpeed;

AudioSource myAudio;

public AudioClip speakSound;

private bool continuar;

Animator anim;

private Vector3 Mover;

private void Start()
{
    continuar = false;
    anim = GetComponent<Animator>();
    sentences = new Queue<string>();
    myAudio = GetComponent<AudioSource>();
}
```

```

void StartDialogue()
{
    sentences.Clear();

    foreach (string sentence in dialogue.sentencesList)
    {
        sentences.Enqueue(sentence);
    }

    DisplayNextSentence();
}

private void DisplayNextSentence()
{
    if (sentences.Count <= 0)
    {
        continuar = false;
        displayText.text = activeSentence;
        if (tag == "NPC") anim.SetBool("hablando", false);
        if (Final.finalSecreto == true && Interfaz.modogenocida == true &&
Input.GetKey("space"))
        {
            Mover = Final3.position;
            Mapa3.transform.position =
Vector3.MoveTowards(Mapa3.transform.position, Mover, 100000 * Time.deltaTime);
            displayText.enabled = false;
        }
        if (Final.finalSecreto == true && Interfaz.modogenocida == false &&
Input.GetKey("space"))
        {
            Mover = Final4.position;

```

```

        Mapa4.transform.position =
Vector3.MoveTowards(Mapa4.transform.position, Mover, 100000 * Time.deltaTime);
        displayText.enabled = false;
    }
    return;
}
activeSentence = sentences.Dequeue();
if (tag == "NPC") anim.SetBool("hablando", true);
displayText.text = activeSentence;
StopAllCoroutines();
StartCoroutine(Typing(activeSentence));
}

private void OnTriggerEnter2D(Collider2D col)
{
    if (col.CompareTag("Player"))
    {
        dialoguePanel.SetActive(true);
        StartDialogue();
        continuar = true;
    }
}

private void Update()
{
    if(continuar == true)
    {
        if (Input.GetKeyDown("space") && displayText.text == activeSentence)
        {
            DisplayNextSentence();
        }
    }
}

```

```
if (tag == "NPC" && displayText.text == "Es un lugar peligroso. Toma " && tag
== "NPC") anim.SetBool("cuchillo", true);
```

```
if (tag == "NPC" && displayText.text == "Es un lugar peligroso. Toma este
cuchillo para tu viaje [Pulsa E para usarlo]." && tag == "NPC")
anim.SetBool("cuchillo", false);
```

```
if (tag == "NPC" && displayText.text == "También" && tag == "NPC")
anim.SetBool("arco", true);
```

```
if (tag == "NPC" && displayText.text == "También te doy este arco [Pulsa R] y
10 flechas." && tag == "NPC")
```

```
{
    anim.SetBool("arco", false);
    Interfaz.modogenocida = true;
}
}
```

```
IEnumerator Typing(string sentence)
```

```
{
    displayText.text = "";

    foreach(char letter in sentence.ToCharArray())
    {
        displayText.text += letter;
        myAudio.PlayOneShot(speakSound);
        yield return new WaitForSeconds(typingSpeed);
    }
}

private void OnTriggerExit2D(Collider2D col)
{
    if (col.CompareTag("Player"))
    {
        dialoguePanel.SetActive(false);
        StopAllCoroutines();
    }
}
```

```

    }
}

}

```

8.5 Script Teletransport

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Teletransporte : MonoBehaviour
{
    public GameObject target;

    bool start = false;

    bool isFading = false;

    float alpha = 0;

    float fadeTime = 1f;

    private void Awake()
    {
        GetComponent<SpriteRenderer>().enabled = false;
        transform.GetChild(0).GetComponent<SpriteRenderer>().enabled = false;
    }

    IEnumerator OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.tag == "Player")
        {

```

```

collision.GetComponent<Animator>().enabled = false;
collision.GetComponent<Movimiento>().enabled = false;
FadeIn();

yield return new WaitForSeconds(fadeTime);

collision.transform.position = target.transform.GetChild(0).transform.position;

FadeOut();

collision.GetComponent<Animator>().enabled = true;
collision.GetComponent<Movimiento>().enabled = true;
}
else
{
    collision.transform.position = target.transform.GetChild(0).transform.position;
}
}

public void OnGUI()
{
    if (!start)
        return;

    GUI.color = new Color(GUI.color.r, GUI.color.g, GUI.color.b, alpha);

    Texture2D tex;
    tex = new Texture2D(1, 1);
    tex.SetPixel(0, 0, Color.black);
    tex.Apply ();

    GUI.DrawTexture(new Rect(0, 0, Screen.width, Screen.height), tex);

```

```

    if(isFadeIn)
    {
        alpha = Mathf.Lerp(alpha, 1.1f, fadeTime * Time.deltaTime);
    }
    else
    {
        alpha = Mathf.Lerp(alpha, -0.1f, fadeTime * Time.deltaTime);

        if (alpha < 0) start = false;
    }
}

public void FadeIn ()
{
    start = true;
    isFadeIn = true;
}
public void FadeOut ()
{
    isFadeIn = false;
}
}

```

8.6 Script Dany

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Damage : MonoBehaviour
{
    void OnTriggerEnter2D(Collider2D collision)
    {

```

```
        if (collision.tag == "Player")
        {
            Interfaz.vida = Interfaz.vida - 1;
        }
    }
}
```

8.7 Script Curació

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Heal : MonoBehaviour
{
    void OnTriggerEnter2D(Collider2D col)
    {
        if (col.tag == "Player")
        {
            Interfaz.vida = 3;

            GetComponent<SpriteRenderer>().enabled = false;

            GetComponent<BoxCollider2D>().enabled = false;
        }
    }
}
```

8.8 Script Diàleg

```
using System.Collections;
```



```
using System.Collections.Generic;  
using UnityEngine;
```

```
[System.Serializable]  
public class Dialogo  
{  
    public string nombre;  
  
    [TextArea(3, 10)]  
  
    public string[] sentencesList;  
  
}
```

8.9 Script Vida 1

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Life1 : MonoBehaviour  
{  
    Animator anim;  
  
    void Start()  
    {  
        anim = GetComponent<Animator>();  
    }  
  
    void Update()  
    {
```

```

        if (Interfaz.vida < 1)
        {
            anim.SetBool("died3", true);
        }
        else
        {
            anim.SetBool("died3", false);
        }
    }
}

```

8.10 Script Vida 2

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Life2 : MonoBehaviour
{
    Animator anim;

    void Start()
    {
        anim = GetComponent<Animator>();
    }

    void Update()
    {
        if (Interfaz.vida < 2)
        {
            anim.SetBool("died2", true);
        }
    }
}

```

```
    }  
    else  
    {  
        anim.SetBool("died2", false);  
    }  
}  
}
```

8.11 Script Vida 3

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Life3 : MonoBehaviour  
{  
    Animator anim;  
  
    void Start()  
    {  
        anim = GetComponent<Animator>();  
    }  
  
    void Update()  
    {  
  
        if (Interfaz.vida < 3)  
        {  
            anim.SetBool("died", true);  
        }  
        else  
        {
```

```

        anim.SetBool("died", false);
    }
}
}

```

8.12 Script del moviment d'una fletxa/objecte

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FlechaMov : MonoBehaviour
{
    public GameObject Objeto;

    public Transform Final;

    public float speed;

    private Vector3 Mover;

    private void Start()
    {
        Mover = Final.position;
    }
    private void Update()
    {
        Objeto.transform.position = Vector3.MoveTowards(Objeto.transform.position,
Mover, speed * Time.deltaTime);
    }
}

```

8.13 Script Destructible

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Destructible : MonoBehaviour
{
    Animator anim;

    BoxCollider2D arbusto;

    void Start()
    {
        anim = GetComponent<Animator>();
        arbusto = transform.GetComponent<BoxCollider2D>();
        anim.SetBool("breaking", false);
    }

    void OnTriggerEnter2D(Collider2D col)
    {
        if (col.tag != "Player")
        {
            anim.SetBool("breaking", true);
            arbusto.enabled = false;
        }
    }
}
```

8.14 Script d'un enemy

```
using System.Collections;
```

```
using System.Collections.Generic;
using UnityEngine;

public class Enemigo : MonoBehaviour
{
    public GameObject Objeto;

    public Transform Jugador;

    public float speed;

    private Vector3 Mover;

    Animator anim;

    public int Zlife = 10;

    private void Start()
    {
        anim = GetComponent<Animator>();
        Zlife = 10;
    }

    private void Update()
    {
        Mover = Jugador.position;
        Objeto.transform.position = Vector3.MoveTowards(Objeto.transform.position,
Mover, speed * Time.deltaTime);
        anim.SetBool("caminando", true);
        anim.SetFloat("movX", Mover.x);

        if (Zlife <= 0)
```

```

    {
        GetComponent<SpriteRenderer>().enabled = false;
        GetComponent<BoxCollider2D>().enabled = false;
        Interfaz.numberz = Interfaz.numberz + 1;
        speed = 0;
    }
}

private void OnTriggerEnter2D(Collider2D col)
{
    if (col.CompareTag("Ataque"))
    {
        Zlife = Zlife - 2;
    }
    if (col.CompareTag("Disparo"))
    {
        Zlife = Zlife - 4;
    }
    if (col.CompareTag("Roca"))
    {
        Zlife = Zlife - 5;
    }
}
}

```

8.15 Script Arc

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Disparo : MonoBehaviour

```

```

{
    [Tooltip("Debo recargar...")]
    public float Espera;

    [HideInInspector]
    public Vector2 mov;

    public float speed;

    void Update()
    {
        transform.position += new Vector3(mov.x, mov.y, 0) * speed * Time.deltaTime;
    }

    private IEnumerator OnTriggerEnter(Collider col)
    {
        if(col.tag == "Object")
        {
            yield return new WaitForSeconds(Espera);
            Destroy(gameObject);
        } else if (col.tag != "Player" && col.tag != "Attack" && col.tag != "Zombi")
        {
            Destroy(gameObject);
        }
    }
}

```

8.16 Script de quan els enemics es poden moure

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

```



```
public class LlamadaDeLosMuertos : MonoBehaviour
{
    public static bool Salid = false;

    private void Update()
    {
        if (Interfaz.vida <= 0) Salid = false;
    }

    private void OnTriggerEnter2D(Collider2D col)
    {
        if (col.tag == "Player")
        {
            Salid = true;
        }
    }
}
```

8.17 Script de l'enemic parat

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Espera : MonoBehaviour
{
    private void Start()
    {
        GetComponent<Enemigo>().enabled = false;
    }
    private void Update()
```

```

    {
        if (LlamadaDeLosMuertos.Salid == true)
        {
            GetComponent<Enemigo>().enabled = true;
            GetComponent<Espera>().enabled = false;
        }
    }
}

```

8.18 Script de l'alarma quan s'entra a la sala de l'enemic final

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Alarma : MonoBehaviour
{
    public static bool batalla = false;
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player")) batalla = true;
    }

    private void Update()
    {
        if (Interfaz.vida <= 0) batalla = false;
    }
}

```

8.19 Script de l'animació de l'enemic final

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class animacion_boss : MonoBehaviour
{

    public Camera CamaraPrincipal;

    public Camera CamaraBoss;

    public Camera CamaraGrande;

    private bool yano = true;

    private void Start()
    {
        CamaraPrincipal.enabled = true;
        CamaraBoss.enabled = false;
        CamaraGrande.enabled = false;
    }
    private void Update()
    {
        if (Alarma.batalla == true && yano == true)
        {
            yano = false;
            CamaraPrincipal.enabled = false;
            CamaraBoss.enabled = enabled = true;
            StartCoroutine(Animation());
        }

    }
}

```

```
IEnumerator Animation()  
{  
    yield return new WaitForSeconds(5);  
    CamaraBoss.enabled = false;  
    CamaraGrande.enabled = true;  
}  
}
```

8.20 Script de l'enemic final

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Dragon : MonoBehaviour  
{  
    Animator anim;  
  
    public static int Dlife = 10000;  
    public int Dvida;  
  
    private void Start()  
    {  
        anim = GetComponent<Animator>();  
        GetComponent<BoxCollider2D>().enabled = true;  
    }  
  
    private void Update()  
    {  
        Dvida = Dlife;  
    }  
}
```

```

    if (Alarma.batalla == true) anim.SetTrigger("despertando");
    if (Dlife > 4000)
    {
        StartCoroutine(CambioDeAnim());
    }
    if (Dlife <= 4000) anim.SetTrigger("fase2");
    if (Dlife <= 0) { anim.SetTrigger("muerte");
GetComponent<BoxCollider2D>().enabled = false; } else {
GetComponent<BoxCollider2D>().enabled = true; }
    if (Interfaz.vida <= 0) Dlife = 10000;
}

```

```

private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.CompareTag("Ataque"))
    {
        Dlife = Dlife - 700;
    }
    if (collision.CompareTag("Disparo"))
    {
        Dlife = Dlife - 500;
    }
    if (collision.CompareTag("Bomba"))
    {
        Dlife = Dlife - 4000;
    }
    if (collision.tag == "Player")
    {
        Interfaz.vida = Interfaz.vida - 1;
    }
    if (collision.CompareTag("Roca"))
    {

```

```

        Dlife = Dlife - 1000;
    }
}

IEnumerator CambioDeAnim()
{
    yield return new WaitForSeconds(20);
    anim.SetBool("izquierda", true);
    yield return new WaitForSeconds(20);
    anim.SetBool("izquierda", false);

}
}

```

8.21 Script per mostrar la vida de l'enemic final

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class VidaBoss : MonoBehaviour
{
    public GameObject dialoguePanel;

    public TextMeshProUGUI displayText;

    void Start()
    {
        displayText.text = "";
        dialoguePanel.SetActive(false);
    }
}

```

```

void Update()
{
    if (Dragon.Dlife > 0 && Alarma.batalla == true)
    {
        displayText.text = "Vida restante" + " " + Dragon.Dlife + "/10000";
        dialoguePanel.SetActive(true);
    } else
    {
        displayText.text = "";
        dialoguePanel.SetActive(false);
    }
}
}

```

8.22 Script de la programació manual dels làsers

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Laseres : MonoBehaviour
{
    Animator anim1;
    Animator anim2;
    Animator anim3;
    Animator anim4;
    Animator anim5;
    Animator anim6;
    Animator anim7;
    Animator anim8;
    Animator anim9;
    Animator anim10;
}

```

Animator anim11;
Animator anim12;
Animator anim13;
Animator anim14;
Animator anim15;
Animator anim16;
Animator anim17;
Animator anim18;
Animator anim19;
Animator anim20;
Animator anim21;
Animator anim22;
Animator anim23;
Animator anim24;
Animator anim25;

public GameObject laser1;
public GameObject laser2;
public GameObject laser3;
public GameObject laser4;
public GameObject laser5;
public GameObject laser6;
public GameObject laser7;
public GameObject laser8;
public GameObject laser9;
public GameObject laser10;
public GameObject laser11;
public GameObject laser12;
public GameObject laser13;
public GameObject laser14;
public GameObject laser15;
public GameObject laser16;
public GameObject laser17;


```
public GameObject laser18;
public GameObject laser19;
public GameObject laser20;
public GameObject laser21;
public GameObject laser22;
public GameObject laser23;
public GameObject laser24;
public GameObject laser25;
```

```
private bool rafaga1 = true;
private bool noparar = true;
private bool rafaga2 = true;
private bool rafaga3 = true;
private bool rafaga4 = true;
private bool rafagacomplementaria = true;
```

```
void Start()
```

```
{
    anim1 = transform.GetChild(0).GetComponent<Animator>();
    anim2 = transform.GetChild(1).GetComponent<Animator>();
    anim3 = transform.GetChild(2).GetComponent<Animator>();
    anim4 = transform.GetChild(3).GetComponent<Animator>();
    anim5 = transform.GetChild(4).GetComponent<Animator>();
    anim6 = transform.GetChild(5).GetComponent<Animator>();
    anim7 = transform.GetChild(6).GetComponent<Animator>();
    anim8 = transform.GetChild(7).GetComponent<Animator>();
    anim9 = transform.GetChild(8).GetComponent<Animator>();
    anim10 = transform.GetChild(9).GetComponent<Animator>();
    anim11 = transform.GetChild(10).GetComponent<Animator>();
    anim12 = transform.GetChild(11).GetComponent<Animator>();
    anim13 = transform.GetChild(12).GetComponent<Animator>();
    anim14 = transform.GetChild(13).GetComponent<Animator>();
```

```
anim15 = transform.GetChild(14).GetComponent<Animator>();
anim16 = transform.GetChild(15).GetComponent<Animator>();
anim17 = transform.GetChild(16).GetComponent<Animator>();
anim18 = transform.GetChild(17).GetComponent<Animator>();
anim19 = transform.GetChild(18).GetComponent<Animator>();
anim20 = transform.GetChild(19).GetComponent<Animator>();
anim21 = transform.GetChild(20).GetComponent<Animator>();
anim22 = transform.GetChild(21).GetComponent<Animator>();
anim23 = transform.GetChild(22).GetComponent<Animator>();
anim24 = transform.GetChild(23).GetComponent<Animator>();
anim25 = transform.GetChild(24).GetComponent<Animator>();
```

```
laser1.GetComponent<BoxCollider2D>().enabled = false;
laser2.GetComponent<BoxCollider2D>().enabled = false;
laser3.GetComponent<BoxCollider2D>().enabled = false;
laser4.GetComponent<BoxCollider2D>().enabled = false;
laser5.GetComponent<BoxCollider2D>().enabled = false;
laser6.GetComponent<BoxCollider2D>().enabled = false;
laser7.GetComponent<BoxCollider2D>().enabled = false;
laser8.GetComponent<BoxCollider2D>().enabled = false;
laser9.GetComponent<BoxCollider2D>().enabled = false;
laser10.GetComponent<BoxCollider2D>().enabled = false;
laser11.GetComponent<BoxCollider2D>().enabled = false;
laser12.GetComponent<BoxCollider2D>().enabled = false;
laser13.GetComponent<BoxCollider2D>().enabled = false;
laser14.GetComponent<BoxCollider2D>().enabled = false;
laser15.GetComponent<BoxCollider2D>().enabled = false;
laser16.GetComponent<BoxCollider2D>().enabled = false;
laser17.GetComponent<BoxCollider2D>().enabled = false;
laser18.GetComponent<BoxCollider2D>().enabled = false;
laser19.GetComponent<BoxCollider2D>().enabled = false;
laser20.GetComponent<BoxCollider2D>().enabled = false;
laser21.GetComponent<BoxCollider2D>().enabled = false;
```

```

laser22.GetComponent<BoxCollider2D>().enabled = false;
laser23.GetComponent<BoxCollider2D>().enabled = false;
laser24.GetComponent<BoxCollider2D>().enabled = false;
laser25.GetComponent<BoxCollider2D>().enabled = false;

}
void Update()
{
    if (Alarma.batalla == true)
    {
        if (rafaga1 == true) StartCoroutine(PrimeraRafaga()); rafaga1 = false;
        if (noparar == true) StartCoroutine(Noparar()); noparar = false;
        if (Dragon.Dlife <= 6000 && rafaga2 == true)
StartCoroutine(SegundaRafaga());
        if (Dragon.Dlife <= 4000 && rafaga3 == true)
StartCoroutine(TerceraRafaga());
        if (Dragon.Dlife <= 2000 && rafaga4 == true) StartCoroutine(UltimaRafaga());
        if (Dragon.Dlife <= 2000 && rafagacomplementaria == true)
StartCoroutine(Complemento());
        if (Dragon.Dlife <= 0) StopAllCoroutines();
    }
}

IEnumerator PrimeraRafaga()
{
    yield return new WaitForSeconds(8);
    anim1.SetTrigger("disparo");
    anim5.SetTrigger("disparo");
    anim9.SetTrigger("disparo");
    anim17.SetTrigger("disparo");
    yield return new WaitForSeconds(2);
    laser1.GetComponent<BoxCollider2D>().enabled = true;
    laser5.GetComponent<BoxCollider2D>().enabled = true;
}

```

```

laser9.GetComponent<BoxCollider2D>().enabled = true;
laser17.GetComponent<BoxCollider2D>().enabled = true;
anim3.SetTrigger("disparo");
anim25.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser1.GetComponent<BoxCollider2D>().enabled = false;
laser5.GetComponent<BoxCollider2D>().enabled = false;
laser9.GetComponent<BoxCollider2D>().enabled = false;
laser17.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
laser3.GetComponent<BoxCollider2D>().enabled = true;
laser25.GetComponent<BoxCollider2D>().enabled = true;
yield return new WaitForSeconds(1);
laser3.GetComponent<BoxCollider2D>().enabled = false;
laser25.GetComponent<BoxCollider2D>().enabled = false;

anim24.SetTrigger("disparo");
anim23.SetTrigger("disparo");
yield return new WaitForSeconds(1);
anim3.SetTrigger("disparo");
anim25.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser24.GetComponent<BoxCollider2D>().enabled = true;
laser23.GetComponent<BoxCollider2D>().enabled = true;
yield return new WaitForSeconds(1);
laser24.GetComponent<BoxCollider2D>().enabled = false;
laser23.GetComponent<BoxCollider2D>().enabled = false;
laser3.GetComponent<BoxCollider2D>().enabled = true;
laser25.GetComponent<BoxCollider2D>().enabled = true;
yield return new WaitForSeconds(1);
laser3.GetComponent<BoxCollider2D>().enabled = false;
laser25.GetComponent<BoxCollider2D>().enabled = false;

```

```

anim2.SetTrigger("disparo");
anim3.SetTrigger("disparo");
anim12.SetTrigger("disparo");
anim13.SetTrigger("disparo");
yield return new WaitForSeconds(1);
anim6.SetTrigger("disparo");
anim5.SetTrigger("disparo");
anim8.SetTrigger("disparo");
anim18.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser2.GetComponent<BoxCollider2D>().enabled = true;
laser3.GetComponent<BoxCollider2D>().enabled = true;
laser12.GetComponent<BoxCollider2D>().enabled = true;
laser13.GetComponent<BoxCollider2D>().enabled = true;
yield return new WaitForSeconds(1);
laser2.GetComponent<BoxCollider2D>().enabled = false;
laser3.GetComponent<BoxCollider2D>().enabled = false;
laser12.GetComponent<BoxCollider2D>().enabled = false;
laser13.GetComponent<BoxCollider2D>().enabled = false;
laser6.GetComponent<BoxCollider2D>().enabled = true;
laser5.GetComponent<BoxCollider2D>().enabled = true;
laser8.GetComponent<BoxCollider2D>().enabled = true;
laser18.GetComponent<BoxCollider2D>().enabled = true;
yield return new WaitForSeconds(1);
laser6.GetComponent<BoxCollider2D>().enabled = false;
laser5.GetComponent<BoxCollider2D>().enabled = false;
laser8.GetComponent<BoxCollider2D>().enabled = false;
laser18.GetComponent<BoxCollider2D>().enabled = false;
rafaga1 = true;
}

```

```
IEnumerator Noparar()
```

```
{
```

```

yield return new WaitForSeconds(16);
yield return new WaitForSeconds(1);
anim6.SetTrigger("disparo");
anim5.SetTrigger("disparo");
yield return new WaitForSeconds(1);
yield return new WaitForSeconds(1);
laser6.GetComponent<BoxCollider2D>().enabled = true;
laser5.GetComponent<BoxCollider2D>().enabled = true;
yield return new WaitForSeconds(1);
laser6.GetComponent<BoxCollider2D>().enabled = false;
laser5.GetComponent<BoxCollider2D>().enabled = false;
anim21.SetTrigger("disparo");
anim22.SetTrigger("disparo");
yield return new WaitForSeconds(1);
anim25.SetTrigger("disparo");
anim6.SetTrigger("disparo");
anim7.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser21.GetComponent<BoxCollider2D>().enabled = true;
laser22.GetComponent<BoxCollider2D>().enabled = true;
yield return new WaitForSeconds(1);
laser25.GetComponent<BoxCollider2D>().enabled = true;
laser6.GetComponent<BoxCollider2D>().enabled = true;
laser7.GetComponent<BoxCollider2D>().enabled = true;
laser22.GetComponent<BoxCollider2D>().enabled = false;
laser21.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
laser25.GetComponent<BoxCollider2D>().enabled = false;
laser6.GetComponent<BoxCollider2D>().enabled = false;
laser7.GetComponent<BoxCollider2D>().enabled = false;
noparar = true;
}

```

```
IEnumerator SegundaRafaga()
{
    rafaga2 = false;
    anim8.SetTrigger("disparo");
    yield return new WaitForSeconds(1);
    anim9.SetTrigger("disparo");
    yield return new WaitForSeconds(1);
    anim10.SetTrigger("disparo");
    laser8.GetComponent<BoxCollider2D>().enabled = true;
    yield return new WaitForSeconds(1);
    anim11.SetTrigger("disparo");
    laser9.GetComponent<BoxCollider2D>().enabled = true;
    laser8.GetComponent<BoxCollider2D>().enabled = false;
    yield return new WaitForSeconds(1);
    anim12.SetTrigger("disparo");
    laser10.GetComponent<BoxCollider2D>().enabled = true;
    laser9.GetComponent<BoxCollider2D>().enabled = false;
    yield return new WaitForSeconds(1);
    anim13.SetTrigger("disparo");
    laser11.GetComponent<BoxCollider2D>().enabled = true;
    laser10.GetComponent<BoxCollider2D>().enabled = false;
    yield return new WaitForSeconds(1);
    anim14.SetTrigger("disparo");
    laser12.GetComponent<BoxCollider2D>().enabled = true;
    laser11.GetComponent<BoxCollider2D>().enabled = false;
    yield return new WaitForSeconds(1);
    anim15.SetTrigger("disparo");
    laser13.GetComponent<BoxCollider2D>().enabled = true;
    laser12.GetComponent<BoxCollider2D>().enabled = false;
    yield return new WaitForSeconds(1);
    anim16.SetTrigger("disparo");
    laser14.GetComponent<BoxCollider2D>().enabled = true;
    laser13.GetComponent<BoxCollider2D>().enabled = false;
```

```

yield return new WaitForSeconds(1);
anim17.SetTrigger("disparo");
laser15.GetComponent<BoxCollider2D>().enabled = true;
laser14.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
anim18.SetTrigger("disparo");
laser16.GetComponent<BoxCollider2D>().enabled = true;
laser15.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
laser17.GetComponent<BoxCollider2D>().enabled = true;
laser16.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
laser18.GetComponent<BoxCollider2D>().enabled = true;
laser17.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
laser18.GetComponent<BoxCollider2D>().enabled = false;
rafaga2 = true;
}

```

IEnumerator TerceraRafaga()

```

{
    rafaga3 = false;
    anim6.SetTrigger("disparo");
    yield return new WaitForSeconds(1);
    anim1.SetTrigger("disparo");
    yield return new WaitForSeconds(1);
    anim2.SetTrigger("disparo");
    laser6.GetComponent<BoxCollider2D>().enabled = true;
    yield return new WaitForSeconds(1);
    anim3.SetTrigger("disparo");
    laser1.GetComponent<BoxCollider2D>().enabled = true;
    laser6.GetComponent<BoxCollider2D>().enabled = false;
    yield return new WaitForSeconds(1);
}

```



```

anim4.SetTrigger("disparo");
laser2.GetComponent<BoxCollider2D>().enabled = true;
laser1.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
anim5.SetTrigger("disparo");
laser3.GetComponent<BoxCollider2D>().enabled = true;
laser2.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
anim7.SetTrigger("disparo");
laser4.GetComponent<BoxCollider2D>().enabled = true;
laser3.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
laser5.GetComponent<BoxCollider2D>().enabled = true;
laser4.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
laser7.GetComponent<BoxCollider2D>().enabled = true;
laser5.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
laser7.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
rafaga3 = true;
}

```

```

IEnumerator UltimaRafaga()
{
    rafaga4 = false;
    anim8.SetTrigger("disparo");
    anim6.SetTrigger("disparo");
    anim18.SetTrigger("disparo");
    anim7.SetTrigger("disparo");
    yield return new WaitForSeconds(1);
    anim9.SetTrigger("disparo");
    anim1.SetTrigger("disparo");
}

```

```

anim17.SetTrigger("disparo");
anim5.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser8.GetComponent<BoxCollider2D>().enabled = true;
laser6.GetComponent<BoxCollider2D>().enabled = true;
laser18.GetComponent<BoxCollider2D>().enabled = true;
laser7.GetComponent<BoxCollider2D>().enabled = true;
anim10.SetTrigger("disparo");
anim2.SetTrigger("disparo");
anim16.SetTrigger("disparo");
anim4.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser8.GetComponent<BoxCollider2D>().enabled = false;
laser6.GetComponent<BoxCollider2D>().enabled = false;
laser18.GetComponent<BoxCollider2D>().enabled = false;
laser7.GetComponent<BoxCollider2D>().enabled = false;
laser9.GetComponent<BoxCollider2D>().enabled = true;
laser1.GetComponent<BoxCollider2D>().enabled = true;
laser17.GetComponent<BoxCollider2D>().enabled = true;
laser5.GetComponent<BoxCollider2D>().enabled = true;
anim11.SetTrigger("disparo");
anim3.SetTrigger("disparo");
anim12.SetTrigger("disparo");
anim13.SetTrigger("disparo");
anim14.SetTrigger("disparo");
anim15.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser9.GetComponent<BoxCollider2D>().enabled = false;
laser1.GetComponent<BoxCollider2D>().enabled = false;
laser17.GetComponent<BoxCollider2D>().enabled = false;
laser5.GetComponent<BoxCollider2D>().enabled = false;
laser10.GetComponent<BoxCollider2D>().enabled = true;
laser2.GetComponent<BoxCollider2D>().enabled = true;

```

```

laser16.GetComponent<BoxCollider2D>().enabled = true;
laser4.GetComponent<BoxCollider2D>().enabled = true;
yield return new WaitForSeconds(1);
laser11.GetComponent<BoxCollider2D>().enabled = true;
laser3.GetComponent<BoxCollider2D>().enabled = true;
laser12.GetComponent<BoxCollider2D>().enabled = true;
laser13.GetComponent<BoxCollider2D>().enabled = true;
laser14.GetComponent<BoxCollider2D>().enabled = true;
laser15.GetComponent<BoxCollider2D>().enabled = true;
laser10.GetComponent<BoxCollider2D>().enabled = false;
laser2.GetComponent<BoxCollider2D>().enabled = false;
laser16.GetComponent<BoxCollider2D>().enabled = false;
laser4.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
laser11.GetComponent<BoxCollider2D>().enabled = false;
laser3.GetComponent<BoxCollider2D>().enabled = false;
laser12.GetComponent<BoxCollider2D>().enabled = false;
laser13.GetComponent<BoxCollider2D>().enabled = false;
laser14.GetComponent<BoxCollider2D>().enabled = false;
laser15.GetComponent<BoxCollider2D>().enabled = false;
yield return new WaitForSeconds(1);
rafaga4 = true;
}
IEnumerator Complemento()
{
    rafagacomplementaria = false;
    anim25.SetTrigger("disparo");
    yield return new WaitForSeconds(1);
    anim23.SetTrigger("disparo");
    yield return new WaitForSeconds(1);
    laser25.GetComponent<BoxCollider2D>().enabled = true;
    anim20.SetTrigger("disparo");
    yield return new WaitForSeconds(1);
}

```

```

laser25.GetComponent<BoxCollider2D>().enabled = false;
laser23.GetComponent<BoxCollider2D>().enabled = true;
anim22.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser23.GetComponent<BoxCollider2D>().enabled = false;
laser20.GetComponent<BoxCollider2D>().enabled = true;
anim21.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser20.GetComponent<BoxCollider2D>().enabled = false;
laser22.GetComponent<BoxCollider2D>().enabled = true;
anim19.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser22.GetComponent<BoxCollider2D>().enabled = false;
laser21.GetComponent<BoxCollider2D>().enabled = true;
anim24.SetTrigger("disparo");
yield return new WaitForSeconds(1);
laser21.GetComponent<BoxCollider2D>().enabled = false;
laser19.GetComponent<BoxCollider2D>().enabled = true;
yield return new WaitForSeconds(1);
laser19.GetComponent<BoxCollider2D>().enabled = false;
laser24.GetComponent<BoxCollider2D>().enabled = true;
yield return new WaitForSeconds(1);
laser24.GetComponent<BoxCollider2D>().enabled = false;
rafagacomplementaria = true;
    }
}

```

8.23 Script Bomba

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

```

```
public class Bomba : MonoBehaviour
{
    public GameObject Objeto;

    public Transform Jugador;

    private Vector3 Mover;

    Animator anim;

    CircleCollider2D explosion;

    private bool agarrada = false;

    public static bool bum = false;

    private void Start()
    {

        explosion = transform.GetChild(0).GetComponent<CircleCollider2D>();

        anim = GetComponent<Animator>();

        explosion.enabled = false;

        GetComponent<SpriteRenderer>().enabled = false;
        GetComponent<BoxCollider2D>().enabled = false;
    }

    private void Update()
    {
```

```

if (Interfaz.numberz >= 5 && Interfaz.vida > 0)
{
    GetComponent<SpriteRenderer>().enabled = true;
    GetComponent<BoxCollider2D>().enabled = true;
}

if (agarrada == true && Input.GetKey(KeyCode.Q) && Interfaz.vida > 0)
{
    GetComponent<SpriteRenderer>().enabled = true;
    anim.SetBool("detonante", true);
    bum = false;
    agarrada = false;
    Mover = Jugador.position;
    Objeto.transform.position = Vector3.MoveTowards(Objeto.transform.position,
Mover, 1000000 * Time.deltaTime);
    StartCoroutine(Explosion());
}
}

private void OnTriggerEnter2D(Collider2D col)
{
    if (Interfaz.vida > 0)
    {
        GetComponent<SpriteRenderer>().enabled = false;
        GetComponent<BoxCollider2D>().enabled = false;
        agarrada = true;
        bum = true;
    }
}

IEnumerator Explosion()
{
    yield return new WaitForSeconds(3);
}

```

```
        explosion.enabled = true;
        yield return new WaitForSeconds(1);
        explosion.enabled = false;
    }
}
```

8.24 Script de la destrucció del mur secret

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Muro : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D col)
    {
        if (col.CompareTag("Bomba"))
        {
            GetComponent<BoxCollider2D>().enabled = false;
            GetComponent<SpriteRenderer>().enabled = false;
        }
    }
}
```