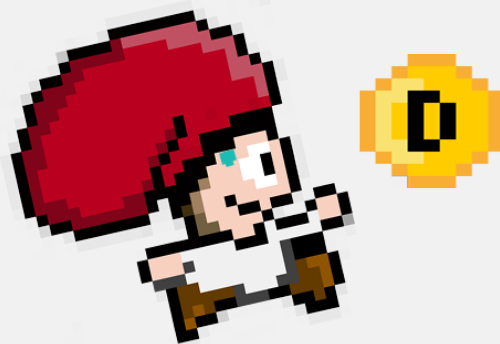


Memòria del projecte final de Grau

En Super Patufet

Creació d'un videojoc de plataformes
en 2D amb Unity3D



Marc Sugranyes Margalef
Grau en Multimèdia

Consultor: Joel Servitja Feu
Professor: Joan Arnedo Moreno

11 de juny de 2017

Aquesta obra està subjecta a una llicència de Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons



ÍNDEX

I RESUM	1
I.1 PARAULES CLAU	1
II ABSTRACT	2
1. INTRODUCCIÓ	3
1.1 CONTEXT I JUSTIFICACIÓ DEL TREBALL	3
1.2 DESCRIPCIÓ	3
1.3 OBJECTIUS DEL TREBALL	4
1.4 ENFOCAMENT I MÈTODE SEGUIT	5
1.5 PLANIFICACIÓ DEL TREBALL	7
1.6 CONTINGUTS	8
2. EL JOC	9
2.1 LA JUGABILITAT	9
2.2 EN PATUFET	10
2.3 ELS ENEMICS	11
2.4 ELS OBJECTES	11
2.5 MENÚ DE PAUSA	12
2.6 PANTALLA PRINCIPAL	12
2.7 ELS NIVELLS	12
3. DISSENY GRÀFIC	14
3.1 ELS PERSONATGES	14
3.2 ELS OBJECTES	18
3.3 LA INTERFÍCIE	19
3.4 TIPOGRAFIA	19
4. PROGRAMACIÓ DEL CODI. SCRIPTS.	20
4.1 MOVIMENT DEL PERSONATGE	20
4.2 VIDA D'EN PATUFET	24
4.3 DANYS	26
4.4 DINERETS	26
4.5 ARRIBADA A META	27
4.6 AGAFAR LA CAIXA	28
4.7 PERSECUSIÓ DEL BOU	29
4.7 ANIMACIONS	31

5. EFECTES SONORS	33
6. CONCLUSIONS	34
7. GLOSSARI	36
8. BIBLIOGRAFIA	37

I RESUM

Aquest document conforma la memòria del Treball de Final de Grau, consistent en la creació des de zero d'un videojoc 2D del gènere de plataformes i puzle, amb un estil retro de manera que ens recordi als videojocs de finals de la dècada dels 80. Aquest procés engloba les etapes de conceptualització, creació, disseny i programació del joc.

El fet d'haver treballat amb un programari com Unity i d'utilitzar el llenguatge de programació C#, ha estat una bona oportunitat per introduir-me en la creació de projectes multi plataforma, ja que m'ha semblat una eina molt versàtil i potent.

Per acabar, m'agradaria aclarir els objectius finals proposats, ja que el temps per executar un projecte d'aquestes característiques seria bastant més extens del que s'ha disposat per aquest treball. Així doncs, el projecte s'ha limitat a produir un videojoc tipus demo jugable i funcional, en la qual s'han implementat les mecàniques bàsiques, el personatge principal, plantejat el funcionament dels enemics i creat algun dels escenaris tipus, sense que això hagi de limitar el desenvolupament posterior del producte després de l'entrega d'aquest TFG.

L'aplicació s'executarà en un entorn Windows.

I.I PARAULES CLAU

videojoc, Unity, Photoshop, 2D, píxel, plataformes, puzle

II ABSTRACT

The present document constitutes the end of degree report, which develops the creation from scratch of a retro style 2D platformer and puzzle game, with reminiscent of the late 80's video games. This process includes the stages of game's conceptualization, creation, design and programming.

Working with Unity software and using the C# programming language has been a good opportunity to introduce myself to the creation of multi-platform projects, because Unity seems a very versatile and powerful game develop platform.

Finally, I would like to clarify the final objectives proposed, as the time to run a project like this would be much more extensive than the time I have had for this work. So, the project has been limited to producing a playable and functionally game demo type, in which I implemented the basic mechanics, the main character, the operation of the enemies and created some type of scenarios. But in no way that might imply any limitation of the further development of the product after delivery of this document.

The application will run in a Windows environment.

1. INTRODUCCIÓ

1.1 CONTEXT I JUSTIFICACIÓ DEL TREBALL

Aquest treball final de Grau (TFG) pretén augmentar i adquirir coneixements sobre desenvolupament de videojocs per part de l'autor. En aquest TFG s'implementarà un videojoc en 2D de tipus plataforma *side-scroller*, intentant fer una mirada enrere als videojocs que vaig gaudir quan jo era un adolescent. Jocs que per un servidor han estat i seguiran sent mítics com per exemple: *Super Mario World*, *Ghosts n' Goblins*, *Megaman*, *Golden Axe*, *Rayman*, *Metal Slug*, i un llarg etc de títols publicats en aquella època. Tot això, sense deixar d'observar l'escena actual pel que fa a la creació de jocs en 2D, que tot i que no gaudeix de tanta popularitat com jocs en 3D, estan sorgint nous estudis de creació de videojocs anomenats *indies*, que sense estar destinats a les consoles d'última generació amb motors gràfics molt potents, obtenen certa popularitat, gràcies a una fàcil distribució i a una bona experiència de joc, que no requereix d'un gran aprenentatge de les mecàniques.

1.2 DESCRIPCIÓ

El projecte consisteix en la creació d'un videojoc en 2D per a plataformes Windows mitjançant el motor de creació de Unity3D, el qual disposa de l'opció de desenvolupament en 2D.

Es podria catalogar el joc com de tipus plataformes, on els elements de jugabilitat que es poden incloure són els de córrer, saltar, escalar parets o esquivar obstacles i espadats, encara que també hi apareix algun element de tipus puzzle.

En aquest treball s'han implementat fins a tres nivells que inclouen algunes d'aquestes característiques pròpies dels videojocs de plataformes. El joc està pensat per a un únic jugador, que haurà de recórrer la plataforma i arribar al final, alhora que cerca tots els objectes recol·lectables de tipus moneda.

1.3 OBJECTIUS DEL TREBALL

L'objectiu principal del TFG és el desenvolupament d'una demo d'un videojoc en 2D amb el motor Unity3D, havent de construir l'estructura del joc, amb els seus components i característiques que permetin la creació de múltiples nivells diferents. Per tal d'arribar a aconseguir-ho, s'hauran de realitzar diferents tasques:

Disseny de les imatges i gràfics del joc

- Disseny de la plataforma: l'escenari per on es mou el personatge.

- Disseny del personatge principal.

- Disseny del personatge enemic.

- Disseny dels objectes a recol·lectar i objectes per interaccionar.

- Creació d'una interfície gràfica d'usuari senzilla i clara.

Programació de les funcionalitats del joc i animacions

- Moviment del personatge.

- Interacció objectes recol·lectables i objectes movibles.

- Comptador dels punts de salut i monedes.

- Moviment personatge enemic.

- Moviment de la càmera i efecte paral·laxis.

Implementació dels sons

Creació de diferents nivells que suposin un repte al jugador

Un altre objectiu més secundari que m'agradaria poder implantar és el desenvolupament d'un mode història per dotar de guió i finalitat al joc.

1.4 ENFOCAMENT I MÈTODE SEGUIT

Fase de definició del videojoc

En aquesta primera fase es decideix la història i ambientació en que es basa el videojoc. Aquest s'inspira en l'argument i en el personatge del conte popular **d'En Patufet**, però adaptant-lo al format típic dels jocs de plataformes. En el joc, el personatge ha de recol·lectar els dinerets, en forma de monedes que es troben en cada nivell del joc, per tal de poder passar al següent nivell. El que s'haurà d'evitar, és caure pels forats o sortir fora dels límits de la plataforma.

Per tal de dur-ho a terme el personatge pot moure's a la dreta i esquerra de la pantalla, saltar i córrer, a més de poder interactuar agafant un objecte i moure'l per tal de poder agafar monedes que queden, a priori, fora de l'abast.

Fase de documentació

En aquesta segona fase es fa una valoració de quins són els motors gràfics i kits d'edició per tal de desenvolupar un videojoc. El candidat ha estat Unity3D degut a la seva facilitat d'ús i de ser gratuït en la versió per a crear continguts no comercials.

Unity, a més, ens ofereix una àmplia gamma de característiques amb una interfície a priori bastant senzilla. La seva principal característica és la integració multi plataforma, és a dir, els jocs es poden portar de forma fàcil i ràpida a Android, iOS, WindowsPhone 8 i BlackBerry.

Des de Unity 4.3 ha estat optimitzat el desenvolupament amb gràfics 2D nadius que simplifica el desenvolupament en 2D, oferint suport per a *sprites* d'imatges, característiques físiques en 2D, càmeres 2D, ...

Per tant és un bon motor per utilitzar-lo en el desenvolupament de jocs 2D.

Fase d'aprenentatge

Aquesta tercera fase s'ha anat realitzant de manera simultània amb la fase de realització i desenvolupament, consultant i buscant la informació a mida que sorgien els dubtes i problemes per a dur a terme el projecte.

Un dels avantatges més grans de Unity és la gran documentació, molta en anglès però també en castellà [<https://docs.unity3d.com/Manual/index.html>], per als diferents llenguatges de programació (c# o Javascript). A més també disposem de molts tutorials i guies existents a la xarxa així com de fòrums d'ajuda especialitzats per tal de poder resoldre dubtes referents a la programació i ús d'Unity3D.

[<https://stackoverflow.com/documentation/unity3d/topics>, <https://forum.unity3d.com/>].

Fase de realització

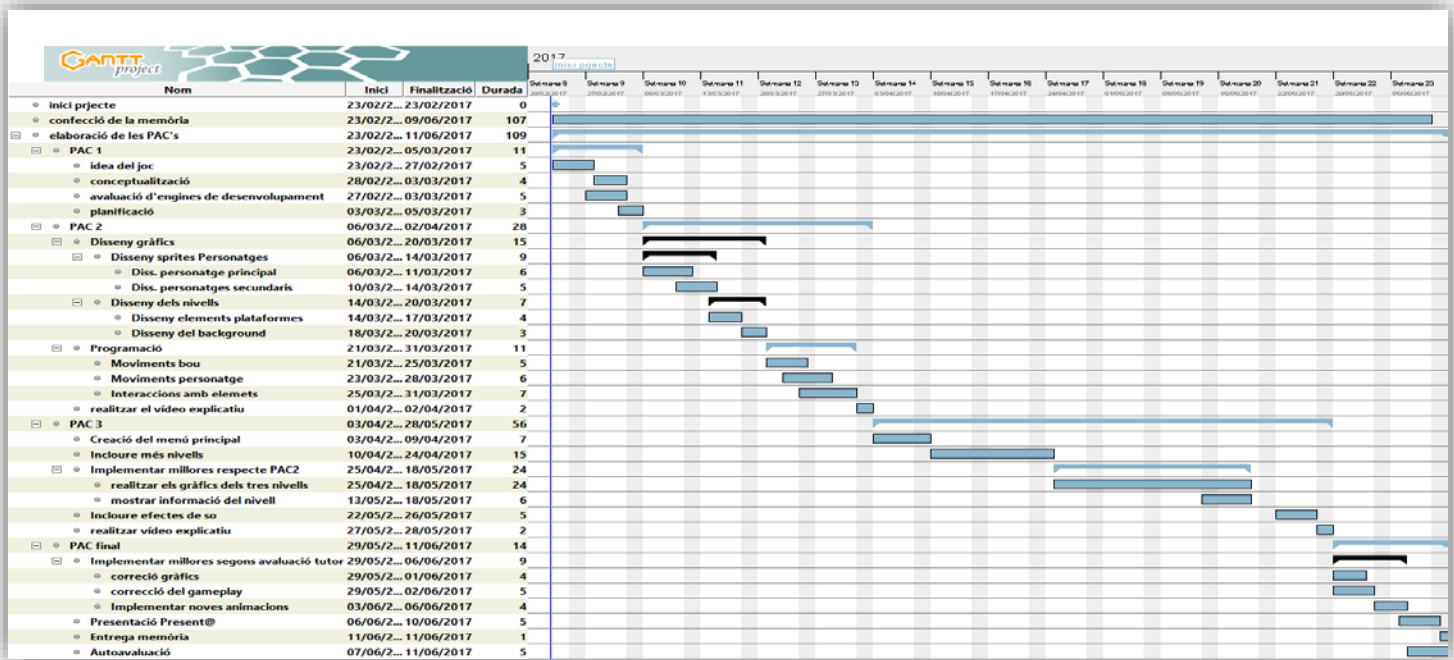
Aquesta fase és la fase on s'han anat aplicant els coneixents adquirits, i s'ha treballat de manera iterativa, revisant la feina feta i intentant millorar-ne cada part.

Primerament, amb l'ajuda del programari d'Adobe Photoshop com a eina d'edició d'imatges, s'ha creat els dissenys dels gràfics del personatge i dels diferents elements que formen la plataforma. En una primera fase s'utilitzen gràfics senzills que es substitueixen i milloren a mesura que s'han d'ajustar a la mecànica del joc.

Un cop obtinguts els primers dissenys de les imatges, s'ha realitzat les corresponents animacions dels elements mitjançant la programació i l'eina d'animació d'Unity3D. Aquestes animacions s'han hagut d'anar adaptant a mida que també s'implementaven les mecàniques del joc, com per exemple el salt simple o els salts entre parets de la plataforma. El llenguatge de programació

escollit per tal de crear els *scripts* encarregats de controlar els moviments, esdeveniments, la física i la programació del joc en general, ha estat el C#.

1.5 PLANIFICACIÓ DEL TREBALL



Un projecte com aquest consta d'una planificació amb moltes subtasques, que poden afectar el progrés en el desenvolupament d'un videojoc. Així que s'ha de tenir en compte que hi ha parts del projecte que s'han d'anar treballant de manera paral·lela. en cas d'endarreriments o imprevistos.

Les parts més importants de la que consta el projecte són:

- Definir el concepte del videojoc
- Disseny gràfic
- Disseny de la programació
- Test i proves
- Realització de la memòria



1.6 CONTINGUTS

Els diferents objectes i arxius que conformen els elements creats durant el desenvolupament del joc segons la categoria a que pertanyen, són els següents:

Imatges. Contenen tota la part gràfica i visual del joc.

patufet.png	window3.png
climb1.png	hearts.png
vaca1.png	arbre1.png
vaca2.png	barril.png
ground.png	portada.png
wood.png	Tip Sign.png
background.png	CAIXA.png
dineret.png	cama.png
dialog.png	fons casa.png
col.png	finish.psd

Escenes. Cadascuna de les escenes, pantalles o nivells que conformen el joc.

pantalla principal



main.unity

primer nivell



level1.unity

segon nivell



level2.unity

tercer nivell



level3.unity

Scripts. Arxius amb la el codi que permet la interacció amb el elements del joc.



Vida.cs



PauseMenu.cs



Gameinfo.cs



Instruccions.cs



CaixaCheck.cs



CamFollow.cs



Moviment.cs



Finish.cs



Hearts.cs



FollowPlayer.cs



ScrollinParallax.cs



MovimentNivell3.cs



Falling.cs



Cames.cs



AspectUtility.cs



PressStart.cs



SoundMGT.cs



Dinerets.cs

Font. Tipografia emprada per a mostrar els textos i la informació durant el joc.

Pixel font-7 [http://www.styleseven.com/php/get_product.php?product=pixel%20font-7]

abcdefghijklmnopqrstuvwxy 1234567890

ABCDEFGHIJKLMN OPQRSTUVWXYZ 1234567890

Sons. Arxius de música i sons per ambientar.

Coin2.mp3

Hit_02.mp3

Final.mp3

Jump_03.mp3

Hit_01.mp3

Patufet.mp3.WAV

2. EL JOC

2.1 LA JUGABILITAT

El joc consta de tres pantalles o nivells, més una pantalla principal amb la portada del joc i un menú de pausa.

Per poder passar de nivell el jugador ha de recol·lectar totes les monedes que es troben repartides en cada nivell. Un marcador ens mostra el número de monedes que portem


recol·lectades i el numero total de monedes a recol·lectar. Un cop s'han agafat totes les monedes, o dinerets, ens hem de situar damunt la plataforma de meta del nivell. Damunt d'aquesta meta, un text i un so ens informará sobre si ho hem aconseguit o si per contra, encara ens fan falta monedes per poder passar a la següent pantalla .

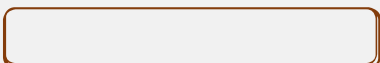
En Patufet disposa de tres cors que ens indiquen el nivell de vida que té. Cada cop que rep un impacte d'algun dels enemics, cama o bou, es perd un d'aquests cors. Si els perd tots, es reinicia el nivell on es trobem, de la mateixa manera, si cau per algun forat o espatat de la plataforma, també reiniciarem el nivell.


2.2 EN PATUFET


Els moviments que pot realitzar són els de moure's a la dreta i a l'esquerra, córrer i saltar. A més pot interactuar amb un element present al nivell per tal de poder recol·lectar tots els dinerets. Relació de tecles que podem utilitzar per a interactuar amb l'aplicació:


 moure's cap a la dreta

 moure's cap a l'esquerra



 saltar

 córrer, mentre es manté premuda

 permet agafar l'objecte caixa per moure'l juntament amb el personatge mentre és manté premuda

 tecla start per començar el joc a la pantalla d'inici

 accedim al menú de pausa

 +  apretades de manera correlativa permeten moure's cap a la dreta en el tercer nivell del joc

2.3 ELS ENEMICS

Els enemics són de dos tipus, les cames que ens trobem en el segon nivell i el bou que ens trobem al tercer nivell. Cada cop que es rep l'impacte amb un d'aquest dos elements, es perd un dels tres cors de vida d'en Patufet.

2.4 ELS OBJECTES

Troblem els dinerets, com a icona típica de monedes a recol·lectar en molts jocs de tipus plataforma i un element amb el qual es pot interactuar agafant-lo i movent-lo juntament amb el personatge per tal d'ajudar-nos a passar el nivell agafant tots els dinerets.

En el tercer nivell, hi trobem les cols que ens dificultaran l'avançament del personatge i que s'hauran d'anar saltant, fent que no sigui massa fàcil arribar al final de la plataforma.

2.5 MENÚ DE PAUSA

Disposem d'un menú de pausa accessible des de qualsevol dels tres nivells i en qualsevol moment mitjançant la tecla d'escapada. En aquest menú hi ha quatre opcions:

Reprendre

reprenem la partida

Nou

reiniciem el nivell que en que ens trobem

Menu Inici

tornem a la pantalla inicial del joc, la portada

Surt

tanquem l'aplicació

2.6 PANTALLA PRINCIPAL

2.7 ELS NIVELLS

Són un total de tres nivells ,en els quals la finalitat és recol·lectar totes les monedes presents a l'escena. Per dur-ho a terme, el jugador podrà moure el personatge, a més d'interactuar amb un objecte per aconseguir-ho i poder passar al següent nivell

En els tres, un cartell informatiu en farà aparèixer informació sobre el què hem de fer i com ho hem de fer, per tal de poder superar el nivell.

Nivell I

En aquest primer nivell es tracta de familiaritzar-se amb la mecànica del joc i amb els moviments d'en Patufet. El jugador haurà de recol·lectar totes les monedes de l'escena i després situar-se damunt la plataforma de meta per poder passar al següent nivell. Per tal de poder realitzar-ho, haurà de saltar i evitar caure al fons de l'escena. A més, haurà de moure un objecte present al nivell amb forma de caixa i situar-lo, de tal manera, que permeti a en Patufet pujar a buscar un dineret que es troba a certa alçada.

Nivell II

A la segona pantalla, de la mateixa manera que en primer, el jugador haurà de recol·lectar els dinerets i ajudar-se de l'objecte caixa per aconseguir-ho, però a més aquest nivell incorpora la dificultat d'haver d'esquivar les cames per evitar ser trepitjat. En Patufet disposa de tres oportunitats per ser trepitjat i poder seguir, si no haurà de reiniciar el nivell II.

Nivell III

Aquest es podria classificar com a un nivell de supervivència, en el qual en Patufet haurà de córrer saltant per sobre les cols, evitar caure pels forats de la plataforma i no ser atrapat pel bou que el persegueix. En un primer moment el moviment es realitzava com en els altres nivells, però vaig veure que es podria augmentar la dificultat havent d'utilitzar dues tecles alternativament per poder fer avançar en Patufet. Si parem de prémer les tecles o ho fem lentament, el protagonista perd velocitat i serà atrapat pel bou.



Track & Field [1983]



Per implementar aquest tipus de moviment, em vaig inspirar en un joc mític del 80, *Track & Field*, amb el qual hi vaig passar moltes estones prement la *tecla de córrer*.

3. DISSENY GRÀFIC

3.1 ELS PERSONATGES

Per realitzar les imatges s'ha utilitzat el programa Photoshop amb l'ajuda del seu sistema de capes per tal de poder realitzar les animacions del personatge i del bou. L'estil que he triat per realitzar aquest joc ha estat el dels gràfics tipus píxel, recordant una mica l'estètica que ens podíem trobar en els jocs de finals dels anys 80, principis dels 90.



Adventure Island
2 [1991]



The Oregon Trail
[1990]

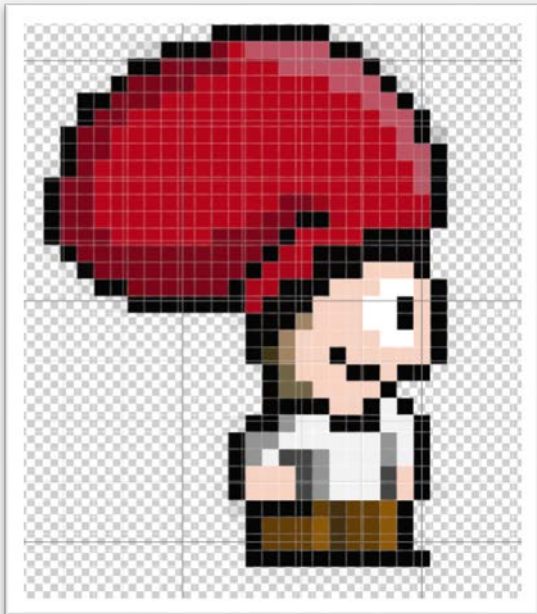


Leisure Suit Larry
[1987]

El procediment ha estat el següent:

Configurem la graella del Photoshop per tal de que es mostri amb una mida entre les línies d'un sol píxel, per veure bé cada punt.

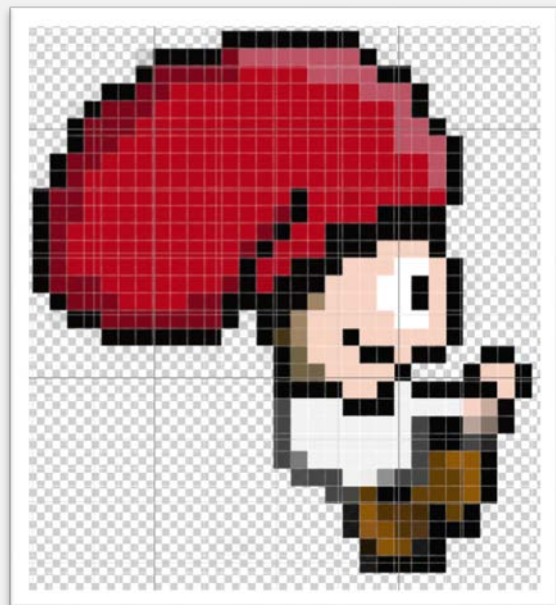
Amb el pinzell del Photoshop a mida 1px es crea la silueta del personatge que es vulgui dibuixar i omplim amb colors les diferents parts. Per tal de que no quedi tant monòton, es ressegueix el contorn amb tons més foscos del mateix color del que hem farcit la figura, per tal de fer com si fossin ombres.

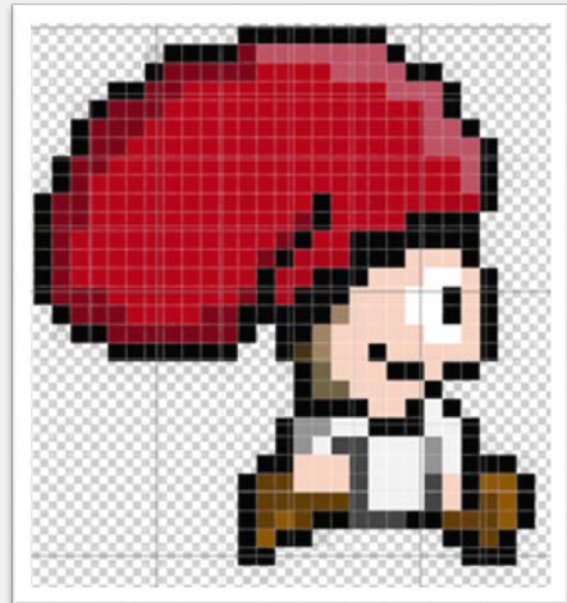
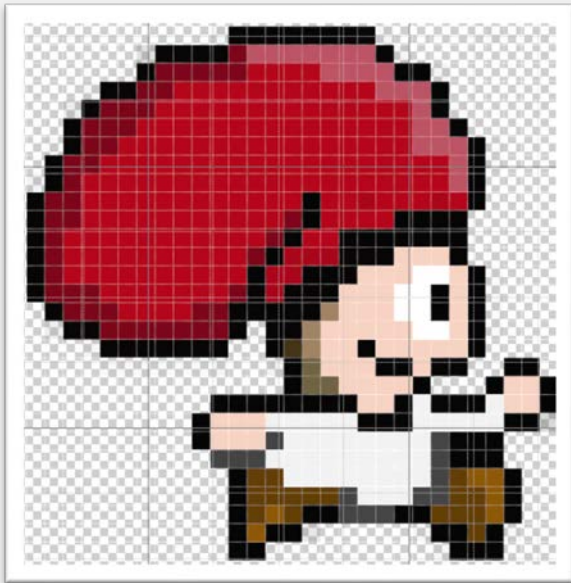


En Patufet

Per al protagonista es tenia bastant clar que hauria de portar una barretina molt gran, que a més s'hauria de moure quan es mogués també el personatge.

Un cop tenim la figura, es canvia les posicions del les extremitats per tal de poder fer-ne les animacions.

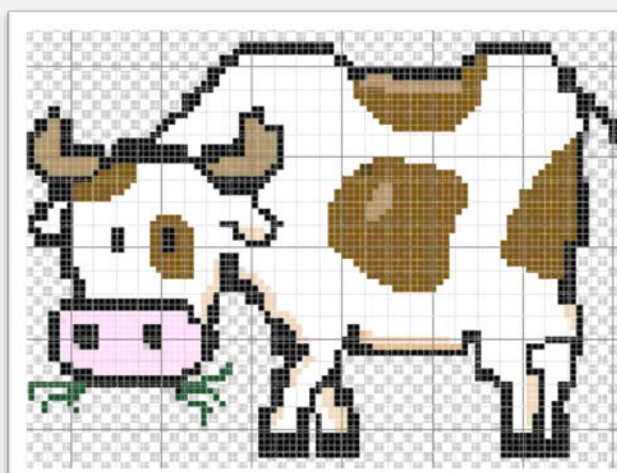
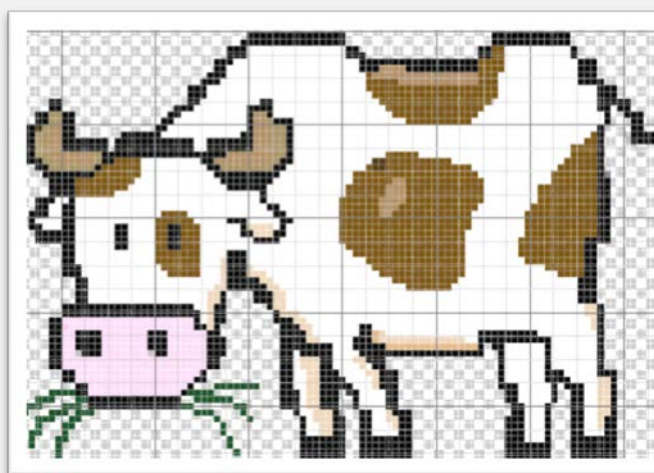
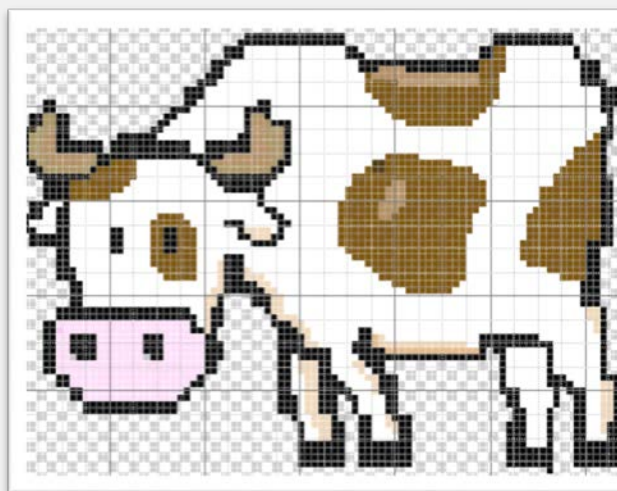




Per provar i testejar el moviment de les animacions, el mateix Photoshop incorpora un animador de capes que ens permet visualitzar com quedarà la possible animació un cop muntem les imatges a l'animador d'Unity3D.



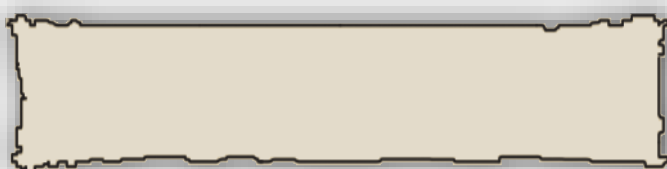
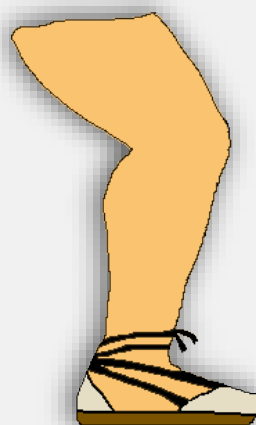
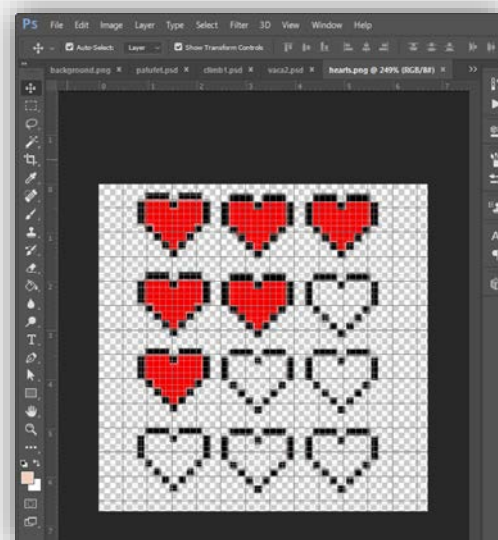
El bou



Per dibuixar el bou s'ha agafat una imatge de la típica vaca amb taques, ja que tot el bou marró quedava poc vistós.

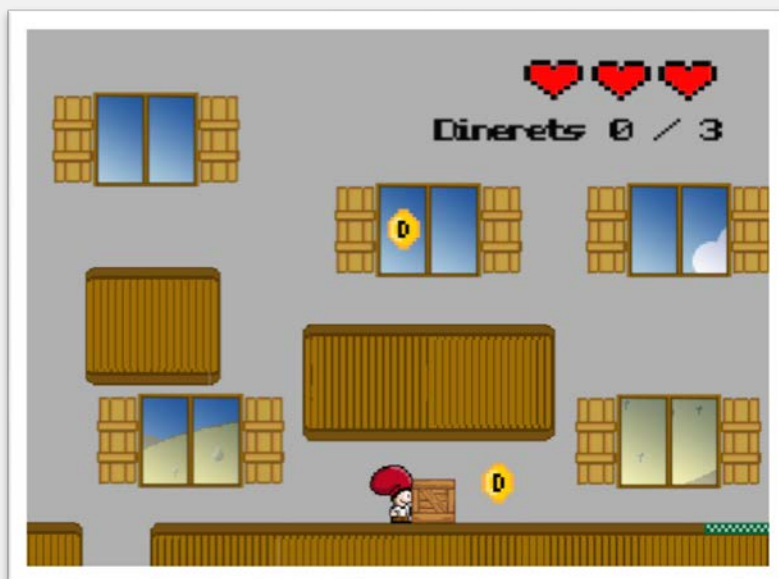
3.2 ELS OBJECTES

La resta d'elements s'han anat creant de la mateixa manera amb les eines del programari d'edició d'imatges.



3.3 LA INTERFÍCIE

La interfície no conté res que no tinguin altres jocs d'aquesta tipologia. De manera molt senzilla, ens mostra a la part superior dreta de la pantalla un comptador dels dinerets recollits i per recollir. Just a sobre hi trobem els tres cors que ens mostren la vida o el número d'impactes que pot rebre en patufet, abans de que es reinicialitzi el nivell en que ens trobem.



3.4 TIPOGRAFIA

La font emprada és una font tipus píxel, que ens recorda a les tipografies usades en aquests primers videojocs desenvolupats en els anys 80.

Es tracta de la **Pixel font-7**. Una font amb llicència *freeware* que es pot descarregar des de la web del seu autor:

http://www.styleseven.com/php/get_product.php?product=pixel%20font-7

abcdefghijklmnopqrstuvwxy 1234567890

ABCDEFGHIJKLMNopqrstuvwxyz 1234567890

4. PROGRAMACIÓ DEL CODI. SCRIPTS.

A continuació s'explicaran les parts dels scripts més rellevants del joc, per tal de dotar d'interactivitat els elements del joc. Tots els scripts del projecte es poden trobar a la carpeta *Assets/scripts*.

4.1 MOVIMENT DEL PERSONATGE

Per tal de poder moure l'element `Player` a través del teclat, és necessari afegir-li un script. Aquest s'ha anomenat `Moviment.cs`.

Només necessitem moure'l en l'eix de les `X`, l'horitzontal. Així que s'utilitza una variable `inputX` per emmagatzemar els valors de `getAxis("Horizontal");`

Si agafa valors positius mourem a `Player` cap a la dreta i si són negatius ho farà cap a l'esquerra.

```
float inputX = Input.GetAxis("Horizontal");
if (inputX > 0) {
    movX = transform.position.x + (inputX * velX);
    transform.position = new Vector3(movX, transform.position.y, 0);
    transform.localScale = new Vector3(1, 1, 1);
}
if (inputX < 0) {
    movX = transform.position.x + (inputX * velX);
    transform.position = new Vector3(movX, transform.position.y, 0);
    transform.localScale = new Vector3(-1, 1, 1);
}
```

Amb `transform.localScale = new Vector3(-1, 1, 1);` voltegem en horitzontal a `Player` per tal de que el personatge miri cap a la direcció a que es desplaça.

La variable `velX` ens ajudarà a canviar la velocitat a que es canviarà de posició, així podem fer-lo "córrer" quan polsem una tecla a més de les de direcció, en aquest cas la 'X'.

```
if (inputX > 0 || inputX < 0) {
    if (Input.GetKey(KeyCode.X) && aTerra) {
        isRun = true;
        salt = 380f;
        saltX = inputX * 100f;
        velX = 0.1f;
        animator.SetBool("isRun", true);
    } else {
        isRun = false;
        velX = 0.03f;
        salt = 320f;
        saltX = 0f;
        animator.SetBool("isRun", false);
    }
}
```

Hem de saber quan `Player` es troba tocant al terra, per tal de que només salti quan es trobi a terra i no a l'aire. Així que a `Player` li afegim un fill anomenat `peu` en forma de cercle, que es mourà juntament amb ell. I fem que la variable `aTerra` que ens comprovarà això, sigui `true` quan `peu` toqui a la capa que he anomenat `terra` (`public LayerMask terra;`).

```
aTerra = Physics2D.OverlapCircle(peu.position, radiPeu, terra +
caixa);

    if (aTerra && !aParet && Input.GetKeyDown(KeyCode.Space)) {
        SoundMGT.sm.PlaySalt();
        GetComponent<Rigidbody2D>().AddForce(new Vector2 (saltX,
salt));
    }
```

Quan sigui veritable i premem la tecla espai, afegim una força al RigidBody2D només en l'eix y igual a la variable `salt`, a més de reproduir un so.

Per tal de que pugui pujar per les parets de la plataforma, es comprova d'igual manera si es troba tocant a la capa paret. Amb un element fill anomenat `paretCheckPoint`.

```
aParet = Physics2D.OverlapCircle(paretCheckPoint.position,
radiCheckPointParet, paret);

        if (aParet && !aTerra) {
ParetSliding();
        }
        if (aParet && aTerra )
{
        aTerra = false;
        ParetSliding();
}
}
```

Amb la funció `ParetSliding()` apliquem a `Player` una força en direcció contrària a on mira, a més de voltejar-lo.

```
void ParetSliding()
{
    rb2d.velocity = new Vector2(rb2d.velocity.x, -0.7f);
    aParet = true;
    animator.SetBool("aParet", true);
    if (Input.GetKeyDown(KeyCode.Space)) {
        if (miraDreta)
        {
            rb2d.AddForce(new Vector2(-0.2f, 1.2f) * salt);
            transform.localScale = new Vector3(-1, 1, 1);
            miraDreta = false;
            animator.SetBool("aParet", false);
        }
        else
        {
            rb2d.AddForce(new Vector2(0.2f,1.2f) * salt);
            transform.localScale = new Vector3(1, 1, 1);
            miraDreta = true;
            animator.SetBool("aParet", false); }}}
}
```

Moviment d'en Patufet per al Nivell III utilitzant l'*script* `MovimentNivell13.cs`, per tal de córrer alternant dues tecles.

Amb aquest codi haurem de prémer alternativament dues tecles, per tal de que `Player` es vagi movent en l'eix de les X. En cas de que es deixin de prémer, o es faci de manera lenta, un comptador `counter` torna a establir el valor baix de `veloX`.

```
saltX = veloX * 100f;
if (Input.GetKeyDown(KeyCode.G) && aTerra && keyAlternate == false)
{
    counter = 0.5f;
    veloX = veloX + 0.2f;
    movX = transform.position.x + (veloX * 0.1f);
    transform.position = new Vector3(movX, transform.position.y, 0);
    isRun = true;
    keyAlternate = true;
}
else if (Input.GetKeyDown(KeyCode.H) && aTerra && keyAlternate == true)
{
    counter = 0.5f;
    veloX = veloX + 0.2f;
    movX = transform.position.x + (veloX * 0.1f);
    transform.position = new Vector3(movX, transform.position.y, 0);
    isRun = true;
    keyAlternate = false;
}
else
{
    counter -= Time.deltaTime;
}
if (counter <= 0f)
{
    veloX = 0.2f;
}
if (veloX > 1)
{
    veloX = 1;
}
```

4.2 VIDA D'EN PATUFET

Dues variables emmagatzemen el nivell de vida, l'actual i el màxim, a l'*script* `vida.cs`. Quan iniciem el nivell de vida està al màxim. Com a molt el nivell de vida serà igual al valor màxim de la variable `maxHealth`, en aquest cas s'ha establert a tres.

En cas de que esdevingui o llancem la funció `Die()`. Aquesta reinicia el nivell en que ens trobem. La funció que s'encarrega d'abaixar el valor del nivell de vida és `Damage(int)`; la qual rep un enter amb el valor a restar a `curHealth`. Llancem també una animació per representar el dany rebut.

```
public int curHealth;
public int maxHealth = 3;

// Use this for initialization
void Start()
{
    curHealth = maxHealth;
}

// Update is called once per frame
void Update()
{
    if (curHealth > maxHealth){
        curHealth = maxHealth;
    }
    if (curHealth <= 0){
        Die();
    }
}

//Die function
public void Die(){
    Application.LoadLevel(Application.loadedLevel);
}

public void Damage(int damage)
{
    curHealth = curHealth - damage;
    gameObject.GetComponent<Animation>().Play("tocat");
}
```

Una variable tipus array ens permet emmagatzemar diferents imatges.



Fent referència a l'*script* `Vida.cs` que conté la informació amb el nivell de vida, per poder accedir a les seves variables.

En aquest cas `curHealth`, variable ens servirà per passar el valor a l'array i mostrar una imatge diferent a l'UI, dependent del valor passat.

```
using UnityEngine.UI;

public class Hearts : MonoBehaviour {
    //imatges
    public Sprite[] HeartSprites;
    public Image HeartUI;

    private Vida vida;
    // Use this for initialization
    void Start () {
        vida=GameObject.FindWithTag("Player").GetComponent<Vida>();
    }

    // Update is called once per frame

    void Update () {
        HeartUI.sprite = HeartSprites[vida.curHealth];
    }
}
```

4.3 DANYS

Codi per tal de restar un punt de vida quan impacta el collider de la cama o del bou amb el RigidBody2D patufet.

```
private Vida vida;
private Moviment player;
void Start () {
    vida=GameObject.FindWithTag("Player").GetComponent<Vida>();
    player=GameObject.FindWithTag("Player").GetComponent<Moviment>(); }
private void OnTriggerEnter2D(Collider2D collision){
    if (collision.CompareTag( "Player")) {
        vida.Damage(1);
        StartCoroutine(player.Tocat(2f,player.transform.position));
    }
}
```

Si es detecta col·lisió es passa el valor de 1 a la funció `Damage()`; que recordem, resta aquest valor a `curHealth`.

Es llança també una animació i una funció que afegeix una força. Aquesta funció es troba a l'*script* `Moviment.cs`.

4.4 DINERETS

A l'*script* `Moviment.cs`, quan es detecta la col·lisió es fa desaparèixer l'objecte tocat i s'accedeix a la variable `dinCount` de `Dinerets.cs` per incrementar el valor en 1.

```
Private dinerets dn;
dn = GameObject.FindWithTag("Dineret").GetComponent<Dinerets>();
//col·lisió monedes
void OnTriggerEnter2D(Collider2D collision){
    if (collision.CompareTag ("dineret")) {
        Destroy (collision.gameObject);
        dn.dinCount += 1;
    }
}
```

A `Dinerets.cs` accedim a l'UI per canviar el text de `dineretsText` amb el valor de `dinCount`.

```
using UnityEngine.UI;
public class Dinerets : MonoBehaviour {
    public int dinCount;
    public Text dineretsText;
    // Update is called once per frame
    void Update () {
        dineretsText.text=("Dinerets:"+dinCount);
    }
}
```

4.5 ARRIBADA A META

Quan s'arriba al final del nivell es troba una plataforma que simula l'arribada a la meta. Per passar de nivell, el jugador s'ha de posicionar a sobre durant uns dos segons, a més d'haver recol·lectat totes les monedes existents a l'escena. Quan `Player` entra al collider es compara el valor de les monedes recollides amb el de les monedes totals presents a l'escena, depenent d'aquest valor es mostrarà un missatge o un altre i es reproduirà un so o un altre.

Script `Finish.cs`

```
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.CompareTag("Player") && (dn.dinCount <
dn.dinTotal)) {
        gi.finishText.text = ("Encara et falten " +(dn.dinTotal-
dn.dinCount) +" monedes !!");
        SoundMGT.sm.PlayNo();
    }
    if (collision.CompareTag("Player") && (dn.dinCount ==
dn.dinTotal))
    {
        SoundMGT.sm.PlayFinal();
    }
}
```

La variable `dinTotal` de `Dinerets.cs`, conté el nombre de fills de l'objecte `dinerets` d'Unity3D.

```
dinTotal = transform.childCount;
```

Si sortim del collider, el comptador torna a iniciar-se i canviem el text perquè no es mostri. Per contra, quan el comptador arriba a 0, carreguem el nivell.

```
private void OnTriggerExit2D(Collider2D collision)
{
    counter = 2f;
    gi.finishText.text = ("");
}

private void Update()
{
    //Debug.Log(counter);
    if (counter <= 0f)
    {
        Application.LoadLevel(level2Load);
    }
}
```

4.6 AGAFAR LA CAIXA

A l'*script* de `Moviment.cs` comprovem que realment la caixa estigui tocant el collider de `Player`. Si és cert, si premem la tecla `V`, afegim l'objecte caixa de l'escena com a fill de `Player`. En cas que deixem de prémer la tecla `V`, l'objecte *caixa* torna al seu estat anterior.

```
agafaCaixa = Physics2D.OverlapCircle(paretCheckPoint.position,
radiCheckPointParet, caixa);
if (agafaCaixa)
{
    if (Input.GetKey(KeyCode.V))
    {
        cx.Caixa.transform.parent = cx.Patufet.transform;
        cx.Caixa.GetComponent<Rigidbody2D>().gravityScale = 0;
    }
}
```



```

        cx.Caixa.GetComponent<Rigidbody2D>().isKinematic = true;
    }
    else
    {
        cx.Caixa.transform.parent = null;
        cx.Caixa.GetComponent<Rigidbody2D>().gravityScale = 1;
        cx.Caixa.GetComponent<Rigidbody2D>().isKinematic = false;
        if (cx.caixaTerra)
        {
            cx.Caixa.GetComponent<Rigidbody2D>().bodyType=RigidbodyType2D.Static;
        }
    }
}
}

```

4.7 PERSECUSIÓ DEL BOU

Amb aquest codi present a `FollowPlayer.cs`, fem que l'objecte `vaca` segueixi a l'objecte `Player`.

Si l'objecte `Player` es distancia bastant, l'objecte `vaca` agafa més velocitat per tal de no marxar de la pantalla. Establim també la variable per fer saltar l'animació a l'animador de Unity3D.

```

if (Running)
{
    if (Vector3.Distance(ThePlayer.position, this.transform.position) <
10)
    {
        Vector3 direction = ThePlayer.position - this.transform.position;
        direction.y = 0;
        transform.position+=direction.normalized*moveSpeed*Time.deltaTime;

        if (direction.magnitude > 5)
        {
            transform.position+=direction.normalized*moveSpeed*4*Time.deltaTime;
        }
    }
    animator.SetBool("running", true);
}

```

El següent codi es fa servir per parar el moviment i l'animació de l'objecte `vaca` quan ens trobem a sobre del cartell d'informació de la pantalla.

Codi per a l'*script* `FollowPlayer.cs`.

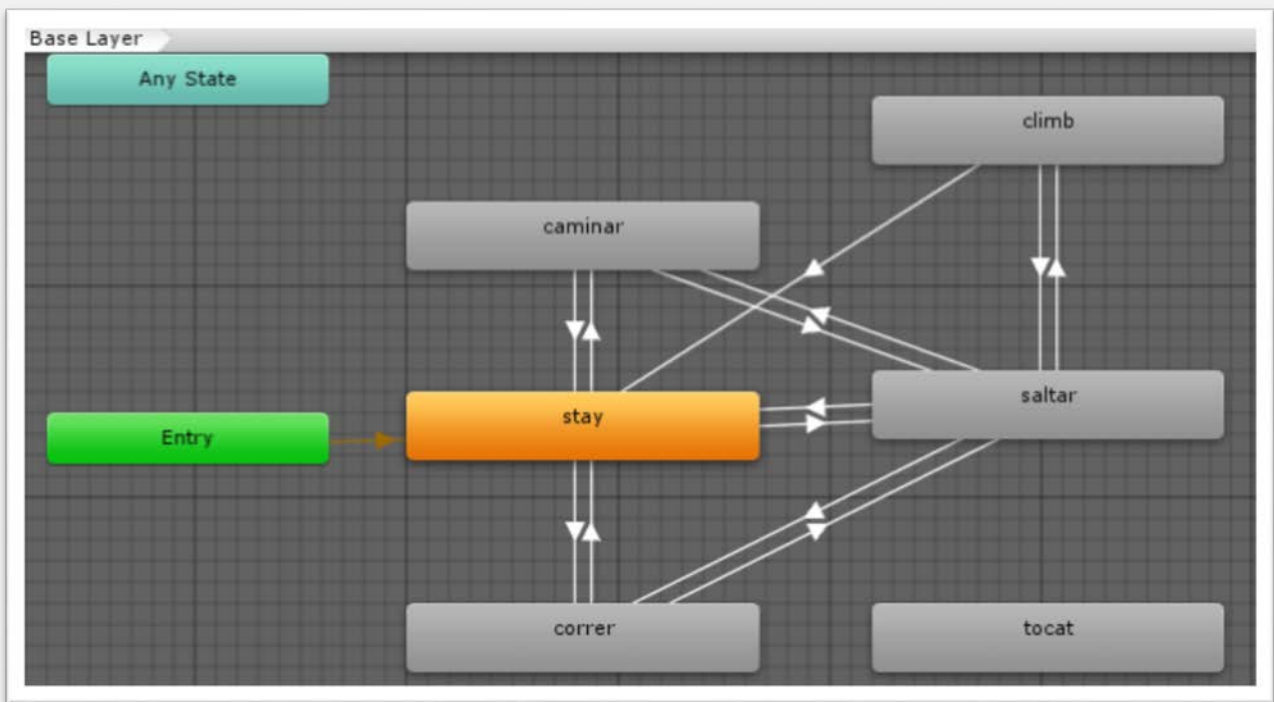
```
if (!Running)
{
    Vector3 direction = ThePlayer.position - this.transform.position;
    direction.y = 0;
    transform.position += direction.normalized * 0 * Time.deltaTime;
    animator.SetBool("running", false);
}
```

Codi per a l'*script* `Instruccions.cs`.

```
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.tag == "Player")
    {
        onTip = true;
        vaca.Running = false;
    }
}
private void OnTriggerExit2D(Collider2D collision)
{
    onTip = false;
    vaca.Running = true;
}
```

4.7 ANIMACIONES

Bàsicament, les animacions amb Unity3D s'han utilitzat per animar els moviments del personatge Patufet.



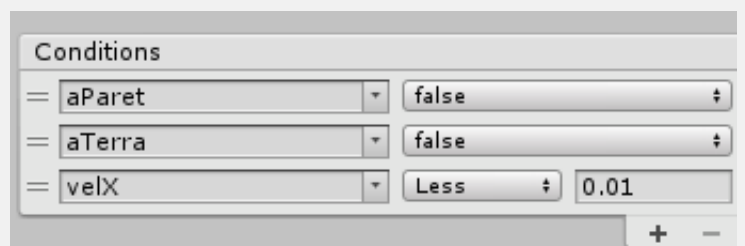
A través de l'animador de Unity es fan les transicions entre els diferents estats.

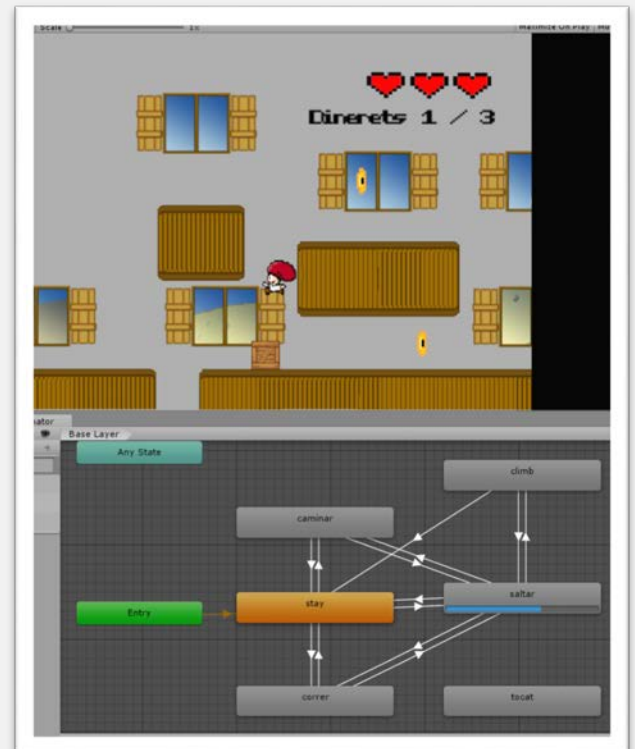
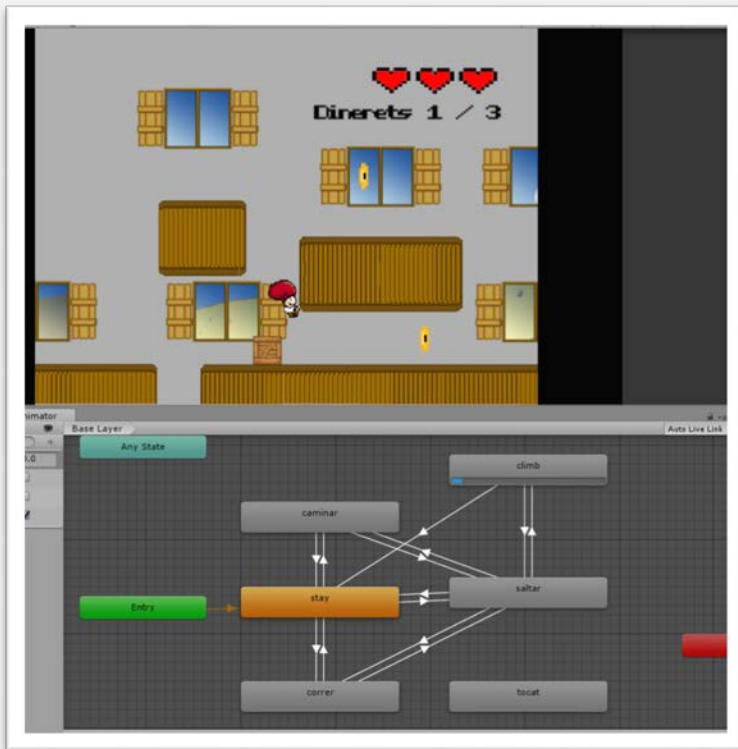
Quiet, caminar, córrer, saltar i escalar.

Per tal de fer les transicions s'utilitzen variables, bàsicament he utilitzat de tipus booleà, les quals es pot canviar el seu estat des dels script que fem servir per al moviment.

Per exemple, per passar de la posició d'escala a la de saltar, s'han de complir les següents condicions:

Que `aParet` sigui `false`, `aTerra` `false` i la velocitat en X sigui menor a `0.01`





Per accedir a les variables de l'animador des dels Scripts es fa amb la següent ordre:

```
GetComponent <Animator>().SetBool("variable", valor);
```

Per exemple, per passar entre animacions per al salt, s'ha utilitzat una variable de tipus booleà als paràmetres de l'animador anomenada aTerra. Per tal de que quan tingui com a valor true no es generi l'animació i quan tingui un valor false es reproduceixi.

Per tal de canviar aquest valor ho fem a l'script que genera el moviment:

```
if (aTerra) {
  GetComponent <Animator>().SetBool("aTerra", true);
} else {
  GetComponent <Animator>().SetBool("aTerra", false);
}
```

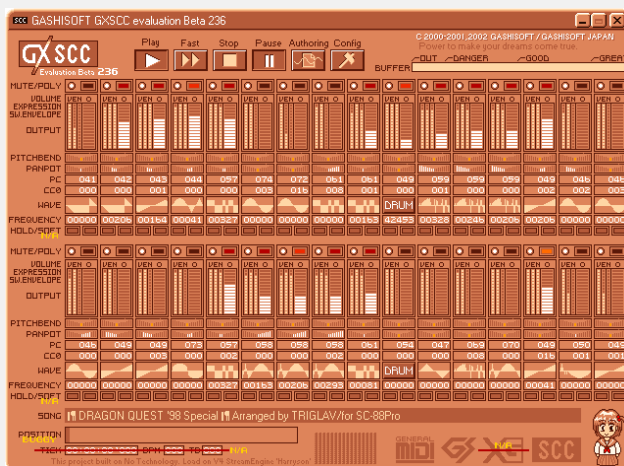
5. EFECTES SONORS

L'aplicació no destaca en aquest apartat. Els sons estan extrets de la base de dades col·laborativa de Freesound [<https://www.freesound.org/>], la qual disposa de llicència Creative Commons en els seus arxius.

S'han incorporat sons en el moviment de salt i quan es colpejat per un dels enemics per tal d'emfatitzar aquestes accions.

També al arribar a la plataforma de *meta*, per tal de que percebem de que hi estem a sobre, juntament amb un missatge de text que ens informa si ens falten o no falten monedes per recol·lectar. Si ens falten monedes es reproduïx un so, si es tenim totes se'n reproduïx un altre.

Per realitzar el so de la pantalla inicial s'ha agafat un arxiu d'àudio que es trobava en format mp3 i s'ha convertit a un arxiu de tipus midi, per tal d'emular el so s'ha utilitzat un programari gratuït anomenat GXSCC. Es tracta d'un reproductor midi desenvolupat per a Windows, que emula el so d'un ordinador antic de 8-bit, com un MSX o una NES, per a arxius midi. Es pot descarregar des de la seva pàgina web.



[<http://www.geocities.co.jp/SiliconValley-SanJose/8700/P/GsorigE.htm>]

6. CONCLUSIONS

Primer de tot, remarcar que aquest treball final de Grau m'ha obert el camí per descobrir i poder treballar amb un programari que no he vist durant la realització d'aquests estudis. Desconeixia el potencial dels motors per a la creació de videojocs com l'Unity3D i això ha resultat ser un factor molt motivant alhora d'aprendre'n el funcionament durant tot aquest procés de construcció d'un videojoc.

En el transcurs d'aquest dies, n'he pogut extreure certs resultats, tant de positius com negatius, que espero que em serveixin de cara a un futur a facilitar-me les tasques i a guanyar experiència afrontant les dificultats que vagin apareixent.

Aquest projecte m'ha servit per aprendre que realitzar un treball d'aquestes característiques comporta una feina enorme, sobretot si l'ha de realitzar una sola persona. Ja que és una tasca en la que participen diferents disciplines, des del disseny gràfic, a la programació, al disseny de l'experiència d'usuari, a les tècniques d'edició d'àudio, de vídeo i fins i tot, encara que en aquest projecte no s'ha treballat, en la creació de videojocs també hi participen els camps de publicitat i màrqueting.

Com a conclusions, podria destacar la gran importància en saber gestionar bé el temps així com també preveure els imprevistos que puguin anar sorgint. Un altre punt important, és saber posar límits al projecte, saber el què volem fer i el què no hem de fer, ens pot estalviar molt de temps en la realització d'aquest.

Pel que fa als objectius plantejats, crec que s'han assolit en gran part. He pogut crear un videojoc de tipus plataforma, amb un cert estil als productes que es creaven a finals dels anys 80 i principis de la dècada dels 90. El protagonista del videojoc, basat en un personatge popular, és bastant identificable. Pel que fa als moviments que realitza, el salt entre les parets de la plataforma crec que finalment han millorat respecte les primeres entregues, al afegir una nova animació que la diferencia de l'animació del salt.

Dels punts a millorar, faltaria millorar la jugabilitat del joc, creant els nivells més extensos per tal d'augmentar-ne el temps de joc. Així com també plantejar una major dificultat a l'hora de saber on haver de moure l'element caixa per a la resolució del nivell, plantejant aquest apartat com una resolució d'un problema més a l'estil dels videojocs tipus puzzle. De totes maneres, considero que aquest punt era tant important per assolir els objectius principals, pel que fa a la realització d'un treball on aplicar els fonaments i aprendre les bases a l'hora de programar i dissenyar un videojoc, més considerant l'espai de temps requerida per a la corba d'aprenentatge d'Unity3D i aconseguir uns resultats dignes, però evidentment, hagués ajudat molt a presentar un producte final molt més atractiu.

Per acabar, en cas de continuar aquest projecte o algun de similar basat en aquest, de cara a un futur quedaria pendent implementar algunes funcionalitats, que tot i que estaven pensades no han pogut dur-se a terme per falta de temps, com per exemple: l'adició d'un objecte que ens augmentes el nivell de vida en cas de que tinguéssim un dels cors per omplir; en els nivells en els quals la mecànica del joc fos com en el nivell III, establiria un cronòmetre per tal de controlar el temps amb que es finalitza el nivell, el qual quedaria emmagatzemat per a un registre dels rècords de la pantalla.

7. GLOSSARI

C#

Llenguatge de programació orientat a objectes utilitzat per implementar els scripts a Unity3D.

Script

Arxiu de text que conté la informació per realitzar els processos que donen interactivitat als gràfics del joc.

Sprite

Imatge plana que s'integra dins de l'escena del videojoc.

UI

Acrònim en anglès d'interfície d'usuari, el medi amb què l'usuari pot comunicar-se amb una màquina. Normalment solen ser fàcils d'entendre i fàcils d'accionar.

Gameplay

Forma específica en què els jugadors interactuen amb el videojoc.

8. BIBLIOGRAFIA

Unity Manual

<<http://docs.unity3d.com/Manual/index.html>>

Unity Scripting

<<https://docs.unity3d.com/Manual/ScriptingSection.html>>

Creating a Basic Platformer Game

<<https://unity3d.com/learn/tutorials/topics/2d-game-creation/creating-basic-platformer-game>>

Crear y animar un sprite (Tutorial paso a paso)

<<http://www.pixelsmil.com/2012/04/crear-y-animar-un-sprite-tutorial-paso.html>>

[data de consulta: 24 de febrer de 2017]

Unity 5 2D Platformer Tutorial

<<https://www.youtube.com/watch?v=dpXJrgRJRQI>>

[data de consulta: 10 de març de 2017]

How to make the Camera follow the Player

<<http://answers.unity3d.com/questions/1151519/how-to-make-the-camera-follow-the-player-1.html>>

[data de consulta: 3 de març de 2017]

How to make a 2D Platformer - MENU - Unity Tutorial

<<https://www.youtube.com/watch?v=Y0lg2TIDiTo>>

[data de consulta: 21 d'abril de 2017]

Creando un juego "Infinite Runner" en 2D con Unity 4.3 (Tutorial)

<<https://www.youtube.com/playlist?list=PLREdURb87ks2qkD9svllwYwN35FZ3Afv>>

[data de consulta: 3 de març de 2017]

How to create a basic follow AI

<<http://answers.unity3d.com/questions/26177/how-to-create-a-basic-follow-ai.html>>

[data de consulta: 12 de maig de 2017]

Unity Tips & Tricks - Parallax Scrolling

<<https://www.youtube.com/watch?v=30OuWY1UfcQ>>

[data de consulta: 5 de maig de 2017]

8-Bit Sound Effects Library by LittleRobotSoundFactory

<<https://www.freesound.org/people/LittleRobotSoundFactory/packs/16681/>>

[data de consulta: 7 d'abril de 2017]