



Logotip de l'aplicació

Food Hunter

A platform game for culinary minds

Autor: Èric Anguera Caihuelas

Grau de Multimèdia

Consultor: Joel Servitja Feu

Data Lliurament:10/06/2017



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Food Hunter</i>
Nom de l'autor:	<i>Èric Anguera</i>
Nom del consultor:	<i>Joel Servitja Feu</i>
Data de lliurament (mm/aaaa):	<i>10/06/2017</i>
Àrea del Treball Final:	<i>Videojocs</i>
Titulació:	<i>Grau Multimèdia</i>

Resum del Treball (màxim 250 paraules):

Food Hunter ha estat una oportunitat per a iniciar-me en el món dels videojocs, creant un joc amb la plataforma Unity on he pogut posar en pràctica molts dels coneixements adquirits al llarg del grau Multimèdia.

Food Hunter és un joc de plataformes 2D tradicional amb reminiscències asiàtiques utilitzant tècniques i eines actuals.

El protagonista és un cuiner samurai que ha d'aconseguir una sèrie d'ingredients per a cuinar una recepta per a l'emperador. Ha de superar una sèrie de dificultats com plataformes, trampes, precipicis i enemics per aconseguir els ingredients. Un cop obtinguts els ingredients, el usuari obté una recepta gastronòmica.

El joc té una versió per a dispositius mòbils i una altra per a PC. És un joc divertit adictiu per aprendre exòtiques receptes de cuina.

El trobem disponible a la botiga Google Play.

<https://vimeo.com/219372647>

Abstract (in English, 250 words or less):

Food Hunter was my opportunity to make and step in the videogame world. I created a small game with the development platform Unity. It was a nice chance to prove all the skills I learned in this Multimedia degree.

Food Hunter is a Standard platform game 2D with some japan narrative and graphic influences.

The main character, Kai, is a Young samurai cooker of a castle. The old cooker ask him some ingredient to make a receipt for the emperor. Kai will have to go to a dangerous world with moving platforms, traps, enemies and find all the ingredients before he goes back to the castle. When he finish the mission the user will get a food receipt.

The game is available for PC and Mobile devices. It's a beautiful addictive game to learn cooking while having fun.

It's available in Google Play Store.

<https://vimeo.com/219372647>

Paraules clau (entre 4 i 8):

Videogame, Platform, Game, Unity, Food, GastroGame, 2D.

Índex Interactiu

1. Introducció

- 1.1 Context i Justificació
- 1.2 Objectius del treball
- 1.3 Enfocament i mètode seguit
- 1.4 Planificació del treball
- 1.5 Productes obtinguts
- 1.6 Breu descripció dels altres capítols de la memòria

2. Food Hunter

2.1 Presentació

- 1.1 Descripció
- 1.2 Gènere i referències
- 1.3 Target
- 1.4 Tests
- 1.5 Ambientació i trama
- 1.6 Definició e interacció dels personatges
- 1.7 Inspiració Gràfica

2.2 Elements gràfics del joc

- 2.1 Personatges
- 2.2 Ingredients
- 2.3 Elements de l'escenari
- 2.4 Escenaris

3.1 Continguts

- 3.2 Escenes del joc
- 3.3 Estructura dels continguts e interacció

4. Construcció dels escenaris

- 4.1 Base
- 4.2 Plataformes
- 4.3 Porta
- 4.4 Trampes
- 4.5 Enemics
- 4.6 Ingredients
- 4.7 Killzone

5. Control del Personatge

- 5.1 Personatge principal
- 5.2 Animator
- 5.3 Moviment
- 5.4 Saltar
- 5.5 Atacar

6. Càmera

7. Quest

8. Canvas

- 8.1 Comptador
- 8.2 Text
- 8.3 Botons Menu
- 8.4 Botons Control

9. Efectes

10. Àudio i Música

3. Conclusions

4. Presentació online

- 4.1 Google Play
- 4.2 Youtube

5. Paraules Clau

6. Bibliografia

7. Scripts

8. Annexos

- 7.1 Entregues anterior

1. Introducció

1.1 Context i justificació del Treball

Al llarg del grau Multimèdia he adquirit un conjunt de coneixements molt variats. Des de disseny gràfic a disseny d'interacció, passant per diversos llenguatges de programació, post-producció audiovisual, narrativa interactiva, animació 3D, empenedoria empresarial ,gestió de projectes...

A través dels anys he assimilat conceptes molt diversos, he après a generar , manipular i gestionar continguts molt diversos en l'àmbit Multimèdia, un dels sectors empresarials més importants, i que més ha crescut en els últims anys.

M'he format com a individu multimèdia, capaç de comprendre i participar en el actual mon digital. Un món digital que cada dia envaeix més la nostra realitat. La nostra quotidianitat cada cop depèn més de la tecnologia. Neveres intel·ligents amb pantalles OLED , dispositius amb intel·ligència artificial, rellotges intel·ligents, aplicacions mòbils que t'ajuden a fer dieta, videojocs per a fer tractaments fisioterapèutics de recuperació.

Una de les ultimes tendències és la ludificació (gamification), i ha estat la ultima empenta que faltava per acabar de fer arribar els videojocs a tots els públics. El sector dels videojocs no para de créixer, ha deixat enrere antics prejudicis i s'instal·la a les aules i a les empreses per a quedar-s'hi.

Si bé mai he estat un àvid jugador, sempre em van agradar alguns jocs una mica diferents que els referents de l'indústria. En els últims anys s'han encunyat alguns noms com indie games, o serious games i he descobert alguns jocs que m'han apassionat, i m'han proporcionat algunes emocions i sentiments únics.

Per aquest motiu vaig tenir la idea d'aprofitar el treball de fi de grau per a introduir-me al món dels videojocs a través de la eina Unity.

Vaig començar el grau Multimèdia al 2009, un any on el smartphone tot just començava a introduir-se en el mercat de masses i era difícil imaginar-se que acabaria convertint-se en un instrument vital en la nostre quotidianitat. Per aquell temps encara existien veus detractores que seguien defensant que el telèfon era un aparell per a realitzar trucades, i el fet de tenir càmera de fotos, escoltar música o tenir jocs no era necessari. Avui dia es inimaginable sortir de casa sense el smartphone. Utilitzem la aplicació de mapes per a trobar una adreça, ens entretenim al metro jugant a diversos jocs, realitzem fotos de 12 megapíxels que no tenen res a envejar de les càmeres reflex...

Per aquest motiu la meva intenció es realitzar un videojoc d'ordinador amb capacitat d'adaptabilitat per a IOS/ Android.

1.2 Objectius del Treball

1.2.1. Objectius generals

- Crear el meu primer joc d'ordinador.
- Realitzar un joc de tipus Plataforma.
- El joc serà compatible per a telèfons mòbils.

1.2.2. Objectius específics

- Posar en pràctica coneixements gràfics, 3D, d'interacció o de programació apresos en el grau.
- Idear una narrativa interactiva.
- Produir tots els objectes 2D del joc.
- Iniciar-me i aprendre a utilitzar el software de producció de Unity.
- Millorar les meves habilitats en el llenguatge C Sharp.

1.3 Enfocament i mètode seguit

Al ser el meu primer contacte amb Unity proposar un mètode a seguir era complicat, tot i així en la primera entrega em vaig marcar una sèrie d'objectius mínims que qualsevol joc de plataformes ha de complir. Gràcies a aquestes pautes que es poden veure a la següent imatge he aconseguit complir tots els objectius i fins i tot afegir alguns de nous.

A més des de l'inici del treball vaig definir de forma clara la estètica, i narrativa del joc, si bé ha patit algunes variacions no ha afectat a cap element del joc.

Un cop marcades les pautes, la meua metodologia s'ha basat en anar realitzant pas a pas tots els objectius de forma orgànica i fluida. Gràcies a aquesta metodologia he pogut anar aplicant coneixements apresos en diversos materials online , o tutorials d'Unity, implementant característiques de múltiples joc ja existents, i ampliant algunes idees pròpies.

En definitiva crec que la metodologia ha estat correcta tenint en compte que era el meu primer contacte amb el programa.

1.4 Planificació del Treball

Etapas	Durada	Inici	Fi
TFG	74d	01/03/2017	12/06/2017
Disseny inicial del projecte	5d	02/03/2017	08/03/2017
PAC1	1d?	08/03/2017	08/03/2017
Creacio elements grafics: illustrator	7d?	09/03/2017	17/03/2017
Narrativa i guió	5d	17/03/2017	23/03/2017
Inicialitzacio projecte Unity	10d?	17/03/2017	30/03/2017
Crear escena, importar elements 2d, background,	3d?	20/03/2017	22/03/2017
Llum i càmera. Incorporació del personatge	3d?	23/03/2017	27/03/2017
Colliders i cossos rígids.	2d?	27/03/2017	28/03/2017
Capes del joc i prefabs	4d?	27/03/2017	30/03/2017
PAC 02	1d?	31/03/2017	31/03/2017
Correccions, modificacions i millores de la primera entrega.	32d?	05/04/2017	18/05/2017
Programar els elements. Moviment d'obstacles, i afegir dificultats	15d?	10/04/2017	28/04/2017
Evolució moviment personatge	15d?	18/04/2017	08/05/2017
Exportar a ios/android	6d?	15/05/2017	22/05/2017
Disseny de nous nivells/dificultats	8d?	03/05/2017	12/05/2017
Adequació/modificacions	8d?	17/05/2017	26/05/2017
Crear Enemic	8d?	17/05/2017	26/05/2017
Inserir música i efectes de so	8d?	17/05/2017	26/05/2017
PAC 03	1d?	26/05/2017	26/05/2017
Revisió/ Correcció	5d?	05/06/2017	09/06/2017
Memoria Final	15d?	18/05/2017	07/06/2017
PAC FINAL	1d?	08/06/2017	08/06/2017

Figura 02. Diagrama de Gantt.

1.5 Breu sumari de productes obtinguts

FoodHunter.exe : versió executable d'un joc d'ordinador per a Windows.

FodHunter.apk: versió executable d'un joc per a Android.

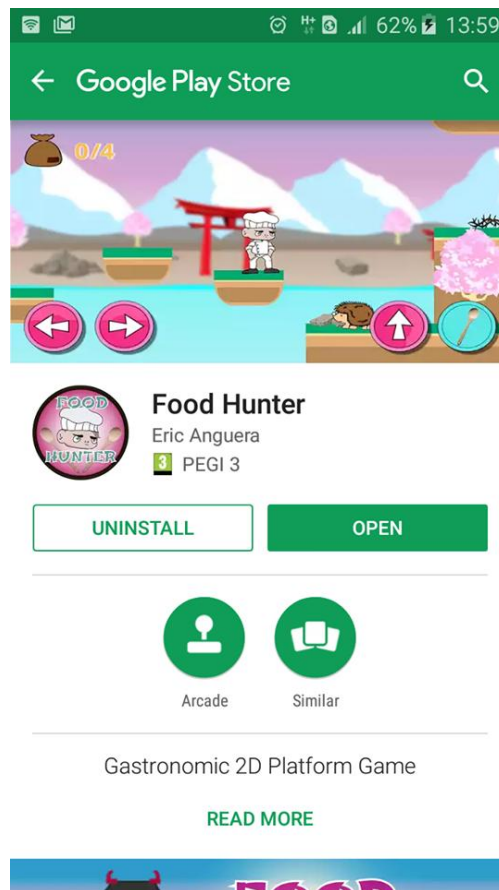


Figura 03. Imatge de l'aplicació a google play.

1.6 Breu descripció dels altres capítols de la memòria

En el següent capítol de la memòria descriure amb detall la meva producció.

Parlaré del target, el tipus de usuaris que podrien estar interessats en el producte.

Mostraré totes les captures de les diferents escenes i etapes del joc, així com la seva estructura i narrativa interactiva.

2. Food Hunter

FOOD HUNTER

2.1 Presentació

2.1.1 Descripció

Food Hunter és un videojoc de plataformes 2D clàssic per a ordinador, tablets i dispositiu mòbils. Es un joc gastronòmic i està ambientat en l'època feudal japonesa (període Edo).

El personatge protagonista és un aprenent de cuiner que haurà d'aconseguir una sèrie de ingredients que demana el personatge NPC (Cuiner savi) per a realitzar una recepta determinada.

El personatge NPC es troba a la part inicial del escenari, i indica a partir d'una conversació textual que necessita ingredients per a cuinar per a l'emperador. A continuació el personatge principal inicia l'aventura movent-se per l'escenari on té que superar una sèrie d'obstacles per aconseguir els ingredients i tornar a l'inici a entregar-los al cuiner .

Al aconseguir els ingredients desitjats, el jugador obtindrà la recepta per escrit i podrà passar al següent nivell.

El joc actualment té dos nivells. L'evolució lògica del joc serà un increment de la dificultat en els nivells, a traves de diverses característiques encara per determinar.

Durant l'aventura, el personatge tindrà que superar una sèrie d'obstacles com plataformes en moviment, trampes punxants, i un enemic. Tindrà que recollir els ingredients de la recepta.

2.1.2 Gènere i referències

Els jocs de plataforma son un gènere que es caracteritza per tenir que caminar, córrer , saltar i escalar en una sèrie de plataformes i espadats, on apareixen enemics i on s'ha de recollir una sèrie de objectes per a completar el joc.

Es un gènere que es va popularitzar a la dècada dels anys 80 amb títols com Donkey Kong o Space Panic, i va esclonar amb el famós primer títol de la saga Super Mario Bros.



Figura 04. Doney Kong 1981



Figura 05. Super Mario Bros 1985

Als anys 90 aquest gènere de joc es va estandarditzar tant en les consoles de 16 bits , com els les primeres consoles portàtils com la Game Boy o la Sega Game gear.

A finals de la dècada dels 90 es van introduir els primers jocs de plataformes 3D com Super Mario 64, tot i així mai superarà en popularitat als jocs 2D.

En els últims anys s'han recuperat moltes sagues, i el gènere va poder revitalitzar-se gràcies a consoles portàtils populars com Game Boy Advance o Ds.

En el context actual crec que és un gènere que es troba una mica paralitzat, ja que l'evolució del disseny del Smartphone actual, com en aparell tàctil , ha dificultat la interacció clàssica d'aquest gènere amb els seus usuaris.

2.1.3 Target

Es un joc tradicional, i bastant simple, així que té un Target molt variat. Es un joc sense violència explícita i es apte i recomanable a partir de 6 anys. AL no precisar d'habilitats especials es apte per a totes les edats. Si bé el target principal és una persona jove de 20 a 30 anys, interessada en aprendre receptes culinàries mentre juga.

2.1.4 Tests

Julia Andrea, 21 anys

“ He jugat a food hunter amb el meu Iphone 6s. Es el meu primer joc d'aquest tipus. Es molt fàcil de fer servir amb els botons, i enganxa molt”



-Mireia Moya, 26 anys

“He utilitzat el meu mòbil Samsung Galaxy s7. M’ha costa tuna mica passar-me el primer nivell, no acostumo a jugar amb el telefon. Es divertit, i he descobert una nova recepta culinaria.”



-Joan Ferrer , 52 anys

“Jugo des de el ordinador de l’empresa. M’agrada el joc, em recorda als jocs de plataforma de la meva època però molt definit. “



-Jonathan Levit, 28 anys

“ Se ve increíblemente bien con mi móvil full hd, aunque soy muy malo jugando y no consigo pasar el nivel”



2.1.5 Història, ambientació i/o trama

En un castell feudal, en l'època samurai als inicis de la restauració Meiji, un jove aprenent de cuina té la missió de preparar un banquet per l'emperador que està de visita. El xef oficial és un avi centenari, i envia al seu jove aprenent a recollir els ingredients per a realitzar cada recepta.

El joc estarà ambientat en un entorn gastronòmic amb estètica asiàtica.

L'objectiu del protagonista serà aconseguir els ingredients necessaris per a realitzar una determinada recepta que ens sol·licita el NPC.

2.1.6 Definició e interacció dels personatges/elements

Kai

PROTAGONISTA

Cuiner aprenent. Jove estudiant de cuina. Prové d'una família de pescadors i viu en un petit poble a les afores de la ciutat. El seu somni es convertir-se en el millor cuiner de la ciutat de Soul.

Han

NPC

Cuiner expert. Cuiner/a molt vella i exigent. Era un cuiner molt famós en els anys 60, però va decidir marxar a un petit poble on viu sol en una cabanya.

Ingredients: Serà el element que el protagonista ha de recollir a través de contacte.

Seràn ingredients de diferents aliments.

Han i Kai es troben en un escenari PREVI a la cuina del castell. Kai explica que per preparar una recepta pel banquet del senyor feudal necessita una sèrie de ingredients.

Kai tindrà que superar una sèrie de obstacles per a recollir els ingredients. Un cop té

tots els ingredients tornarà a la cuina del castell per entregar-li al cuiner savi i així obtenir la recepta.

2.1.7 Inspiració Gràfica

Tots els elements gràfics del joc, com els icones, els personatges o el fons seran realitzats amb programari d'Adobe. Seguiran una temàtica gastronòmica, amb influències asiàtiques i japoneses.

S'utilitzarà una gamma cromàtica de colors de tipus pastel.



Figura 06. Ramen Master APP



Figura 07. Missy Messy's midnight ramen



Figura 08. Llibreria d'Unity d'icones de menjar



Figura 09. Dibuix propi

2.2 Elements gràfics del joc

He realitzat o modificat tots els elements gràfics amb el programa Adobe Illustrator.

He realitzat dibuixos amb vectors utilitzant la tableta gràfica Wacom Intuos i el mouse.

2.2.1 Personatges

He dibuixat cada element del personatge per capes, així m'ha estat molt més fàcil realitzar les diferents versions de les animacions del personatge.

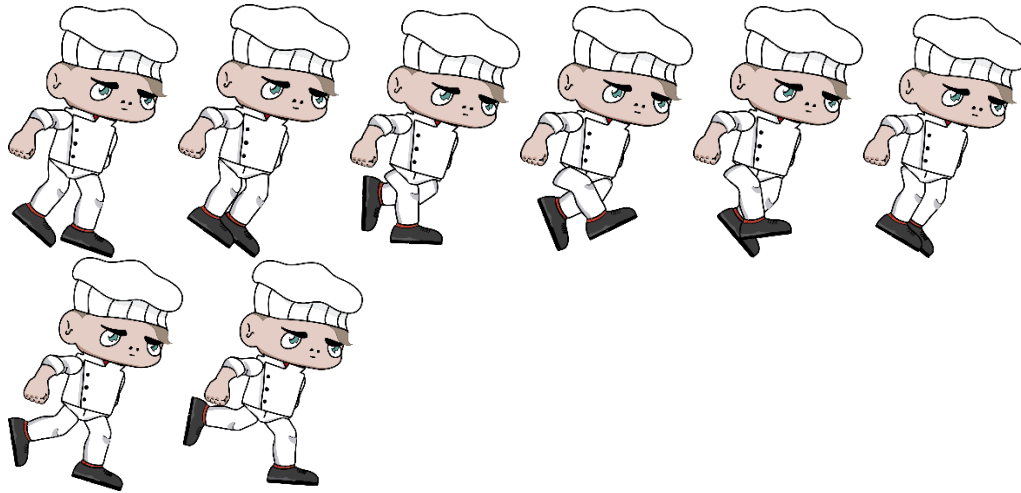


L'aplicació es basa en transicions de diverses animacions que simulen cada estat del personatge o els objectes.

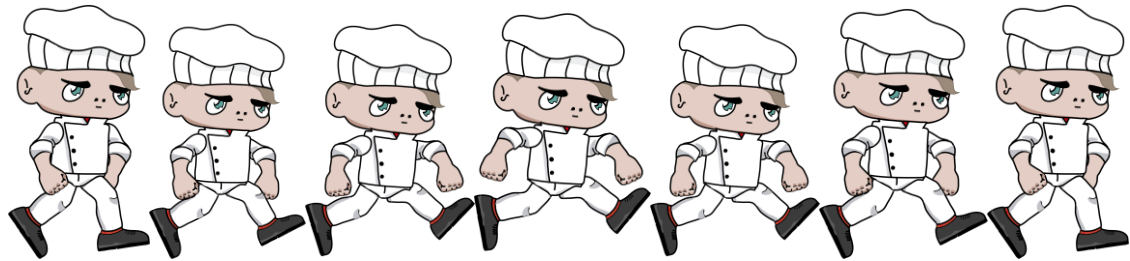
- **IDLE** : Animació quan el personatge esta quiet. Simula un moviment de respiració



- **RUNNING**: És el moviment del personatge al apretar les tecles direccionals.

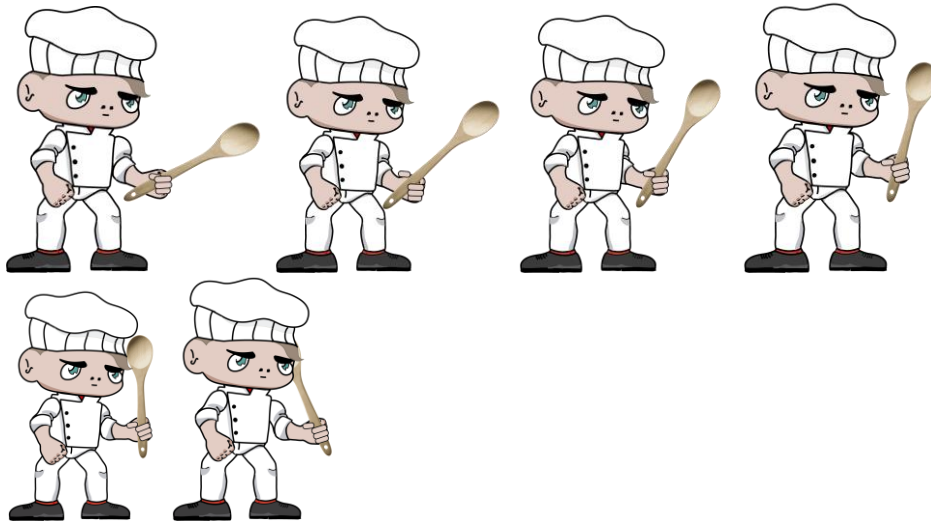


- **JUMP:** Saltar o caure.

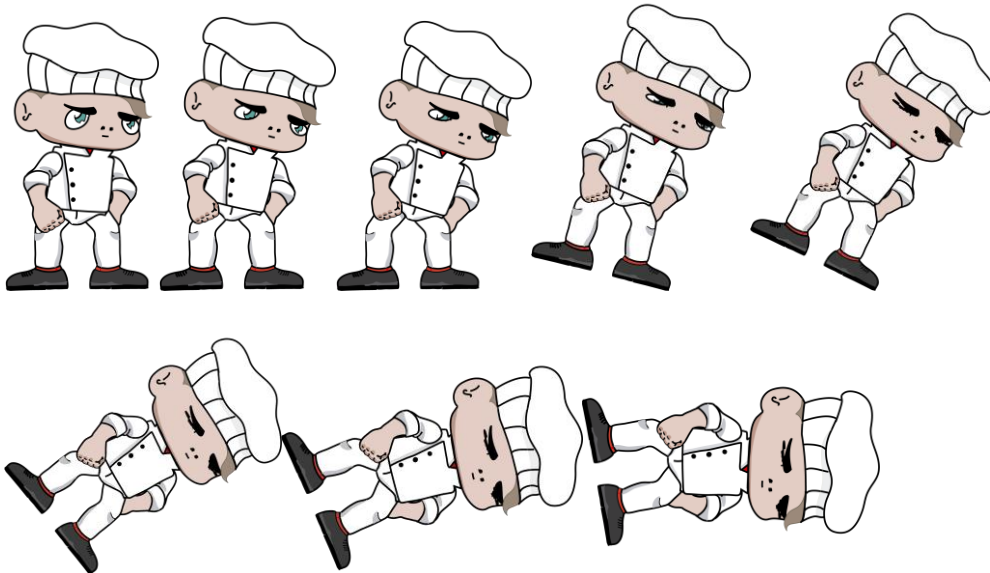


- **ATAC:** Moviment d'atac.

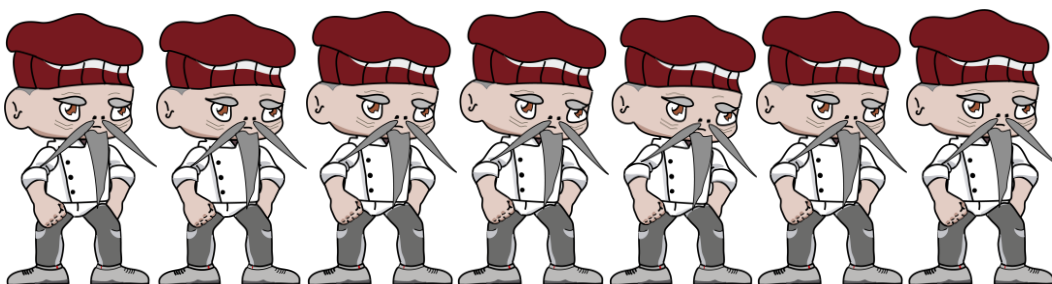




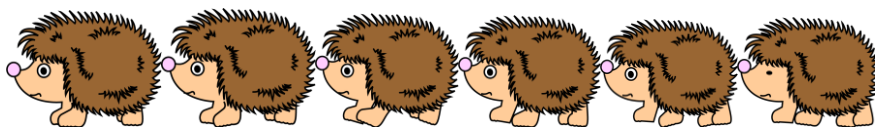
- **DEAD** : Acció del personatge al morir



- **NPC**



- Eriçó



2.2.2 Ingredients

Col



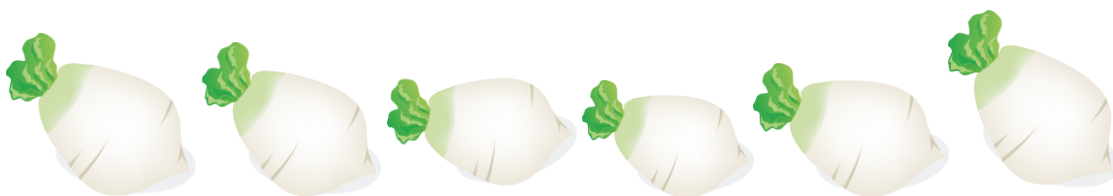
Pastanaga



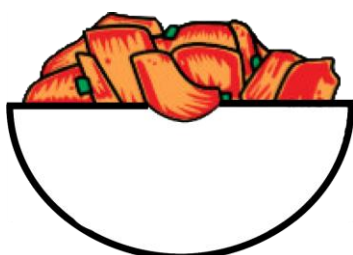
Chili



Rave

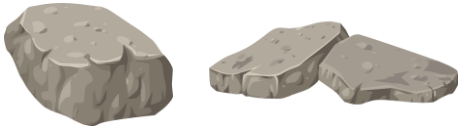


Kimchi



2.2.3 Elements del escenari

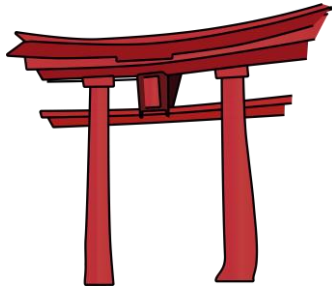
Pedres



Cirerer



Porta Temple



Taula cuina



Porta



Plataformes



Base terra



plataforma voladora



Plataforma terra

2.2.4 Escenaris

Fons escenari 01



Figura 011. Fons escenari 1

Fons escenari 2

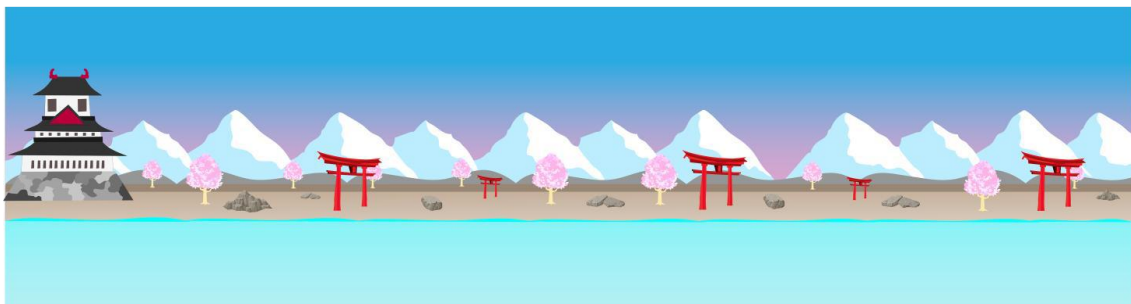


Figura 012. Fons escenari 2

2.3.1 Continguts

2.3.2 Escenes del joc

La primera escena que es troba el usuari al obrir l'aplicació. Trobem un escenari estàtic amb els personatges i el cartell del joc amb una explosió lumínica. Al mig de l'escenari hi han els quatre botons. Un botó d'inici del joc, dos botons desactivats fins que no es superi el primer nivell (segon nivell i receptes), i un botó de sortida de l'aplicació.



Figura 013. Fons Main Menu

Escena 1

En aquest escenari trobem el NPC, el cuiner savi, que ens invita a parlar amb ell. Un cop parlem amb ell ens explica que el emperador ve a sopar i necessita una sèrie d'ingredients per a la recepta. Aleshores s'inicia la quest i podem sortir per la porta per anar al següent escenari.



Figura 014. Escena 01

Escena 2

Es el escenari plataforma de l'aventura. S'han de superar una sèrie de plataformes, trampes i uns enemics en forma de eriço. Trobem una sèrie de ingredients que s'han de recol·lectar. A la barra superior trobem un dibuix d'una bossa i un comptador dels ingredients. Al arribar al final trobem una porta que ens retorna a la escena 1, on el NPC ens felicita si tenim tots els ingredients, o ens diu que encara falten ingredients si no.

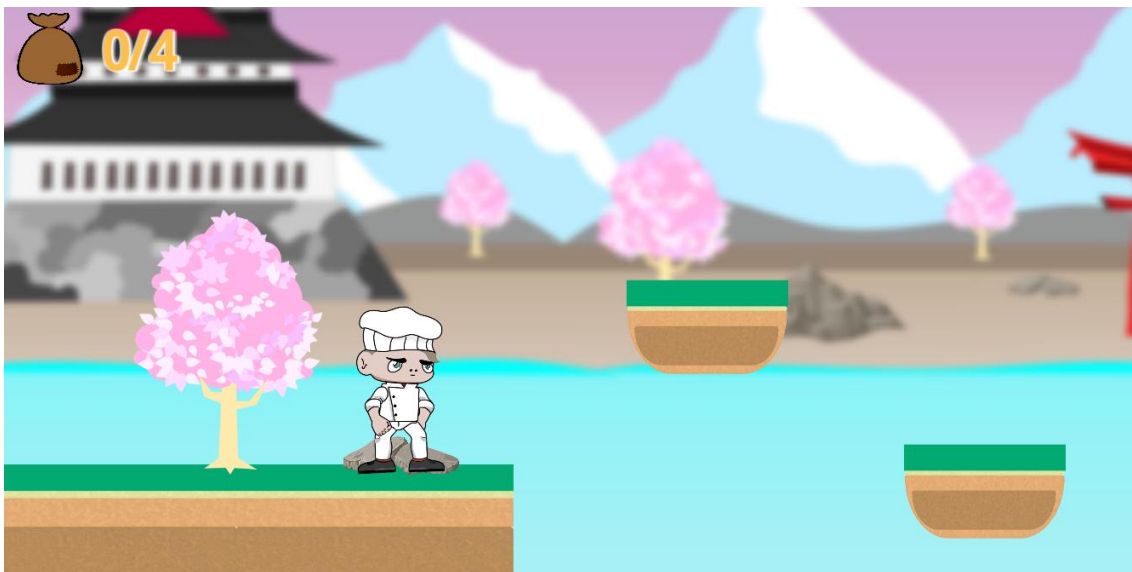


Figura 015. Escena 02

Escena Receptes

En aquesta escena trobem la cuina del Castell amb la recepta. Té un botó de tornar al inici.

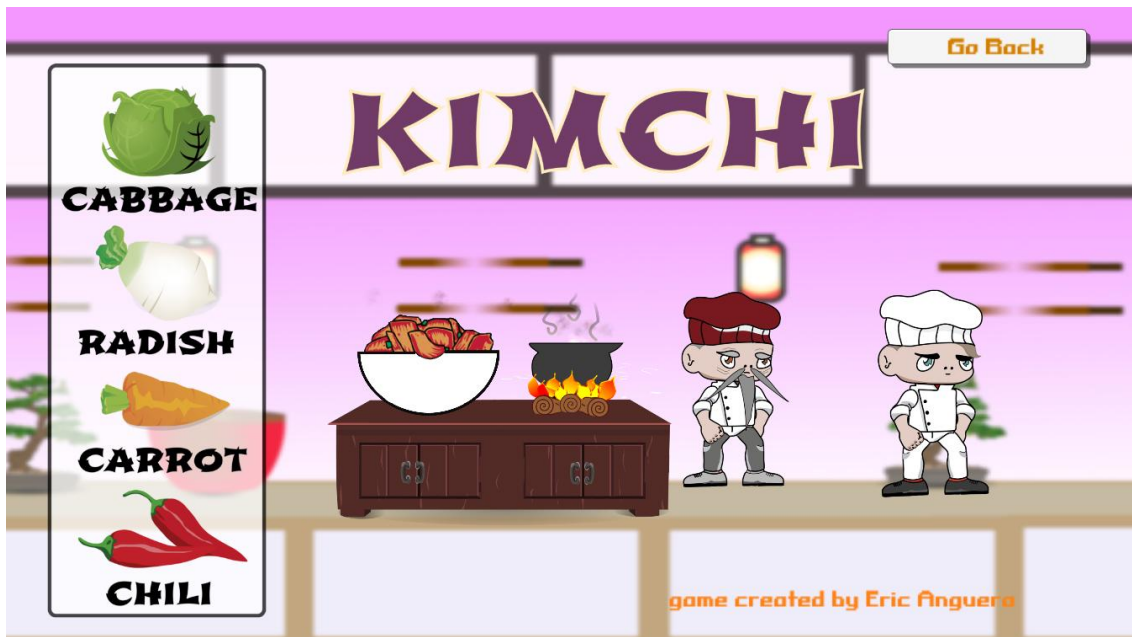


Figura 016. Escena Receptes

Escena Game Over

Apareix al precipitar-se per l'escenari, al caure a les trampes o topar amb els eriçons.



Figura 017. Escena Fi joc

Escena 3

Es la escena inicial com la escena 1, però dona pas al nivell 2. Trobem una petita modificació lumínica.

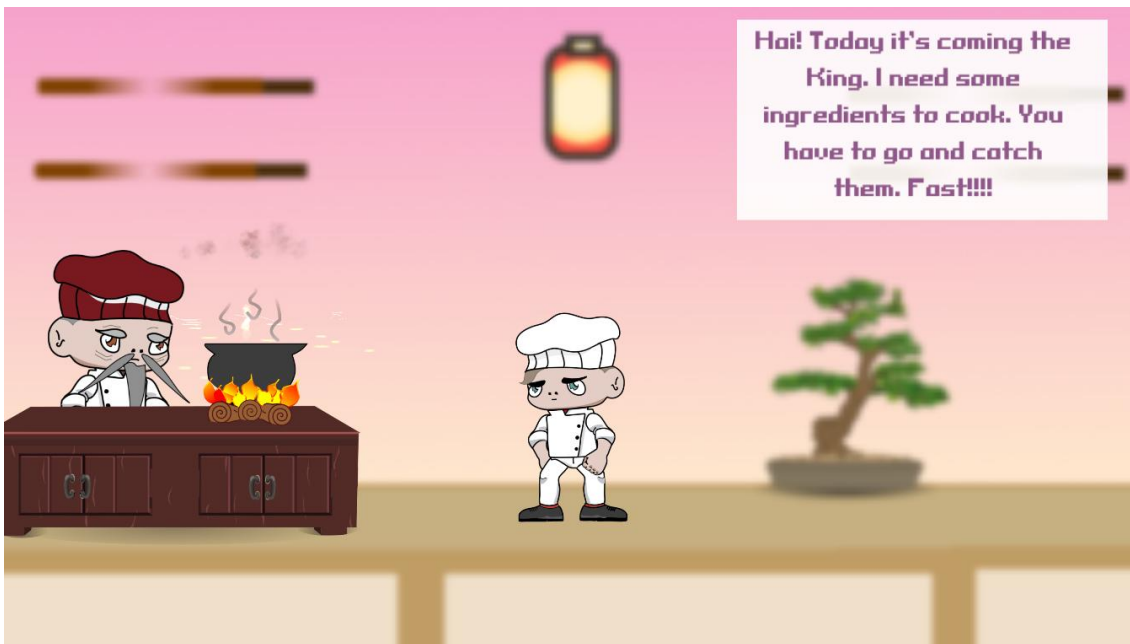


Figura 018. Escena 03

Escena 4

Es el escenari 2 però més complicat. Trobem més enemics, més ràpids, plataformes més complicades... També trobem algunes modificacions gràfiques.

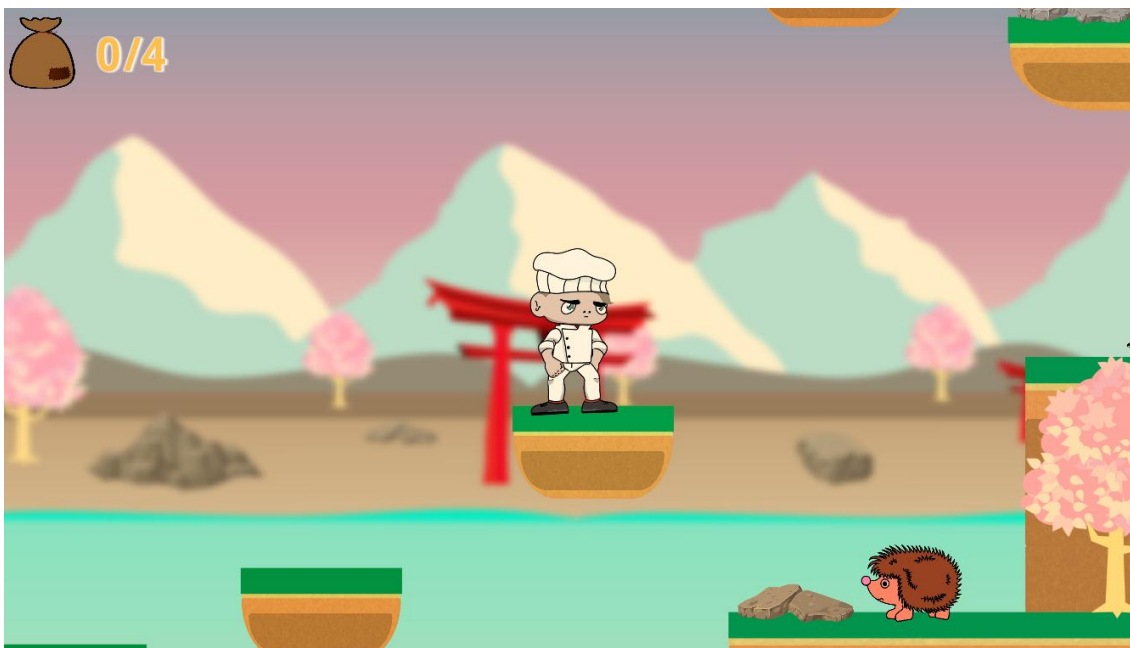


Figura 019. Escena 04

2.3.3 Estructura dels continguts e interacció

iniciem l'aplicació

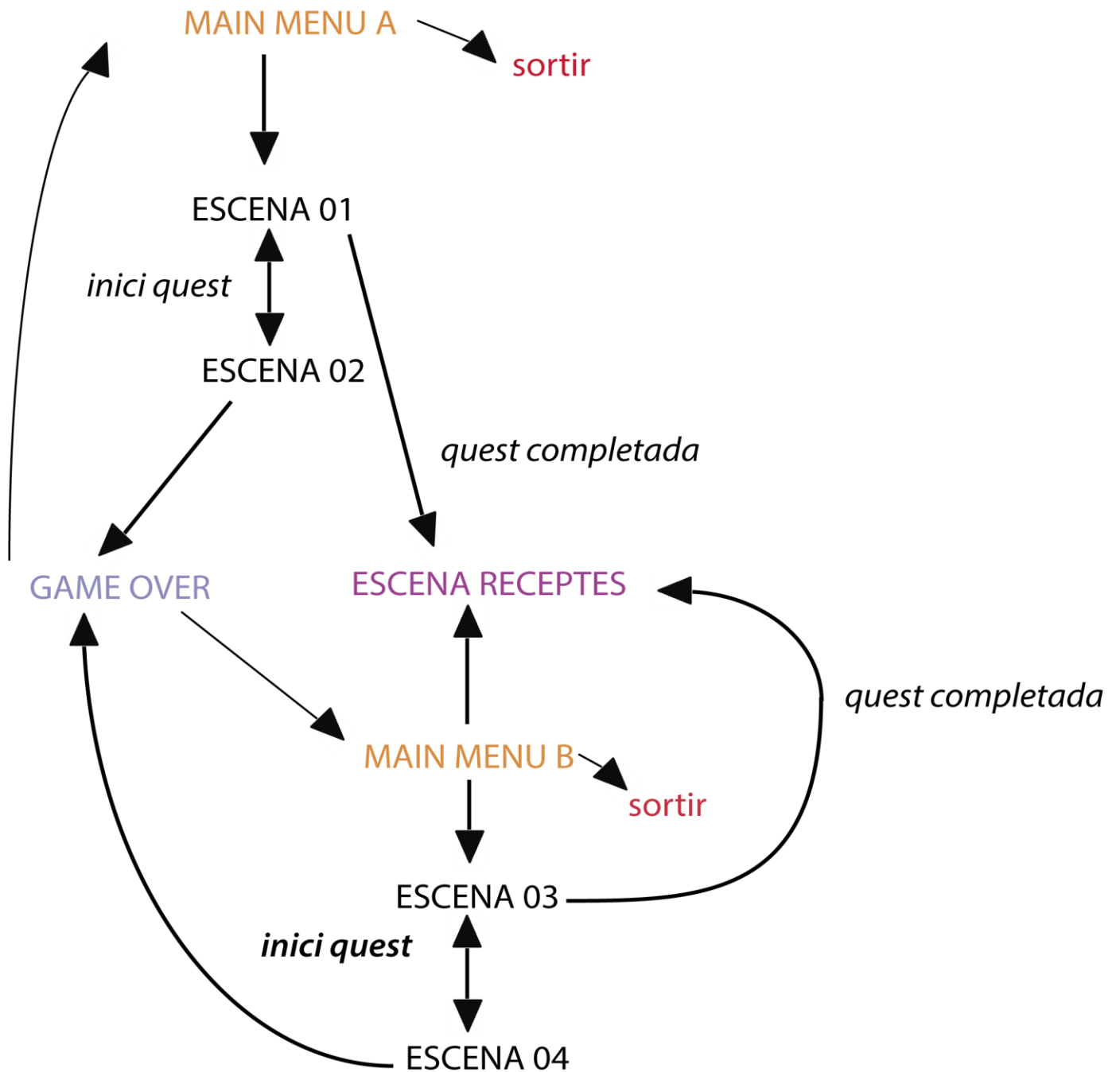


Figura 020. Interacció del joc

2.4 Construcció dels escenaris

El programa Unity carrega les textures (imatges) a través de capes. Totes les imatges tenen que ser convertides a Sprites (2D). Tot i ser un joc 2D s'ha treballat en espais tridimensionals.

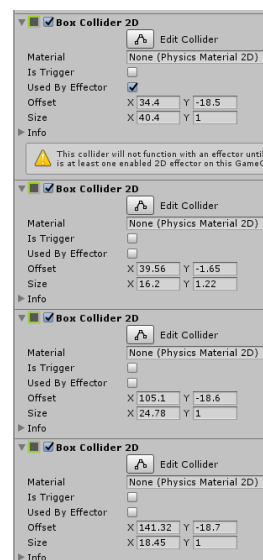
2.4.1 Base

Primer vaig carregar el fons de l'escenari en la capa 0, i a continuació la resta d'elements com les plataformes, o la base en una capa superior. Després vaig carregar els elements decoratius com les roques o els arbres.

Un cop construït el escenari teníem que dotar-lo de característiques físiques, és a dir, teníem que aclarir on esta el terra, on col·lidiran els objectes que col·locarem en el futur, i estaran condicionats per lleis físiques com la gravetat. Per a construir el terra s'han utilitzat una sèrie de colliders 2D posicionats en el nivell de color verd (gespa) de les plataformes i la base.



Figura 021,22. Collider de la Base



2.4.2 Plataformes

En les plataformes, apart del collider 2d per a delimitar el terra, s'ha afegit un collider extra, posionat lleugerament inferior al inicial. Aquest collider l'utilitzarem en un script anomenat `colplataforma`. En aquest script trobem dues funcions de tipus "trigger" (`trigger enter`, `trigger exit`). Un trigger és un mecanisme de tipus gallet, que funciona com a disparador i al col·lidir amb el objecte 2d del personatge permetrà que pugui saltar sobre la plataforma des de la part inferior, i després es mantingui sobre el primer collider.

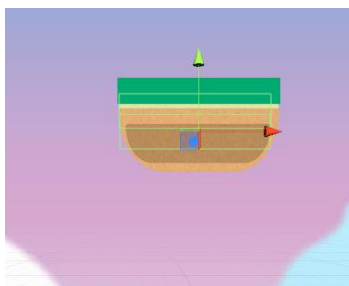


Figura 023 .Collider de la plataforma

Un cop teníem totes les plataformes col·locades en l'escenari, la intenció va ser dotar-les de moviment. Després de provar alguns mètodes es va optar per crear un script anomenat `Movplataforma`. En aquest script es declaren uns punts físics vectorials que s'utilitzen per a moure les posicions inicials de l'objecte a un nou destí amb una velocitat serialitzada. Aleshores es crea un "game object" nou amb un rigid body i col·loquem la plataforma dintre. Aquest cos rigid és de tipus kinematic. També creem un objecte esclau anomenat posició A, que el situem sobre les coordenades on voldrem que la nostra plataforma situada en la posició B, es desplaci.

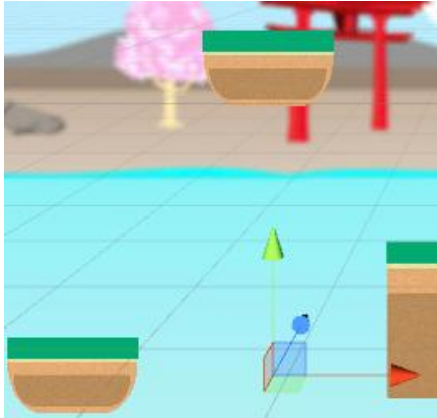


Figura 024 i 025. Moviment plataformes

Al llarg de l'escenari trobarem plataformes amb diferents velocitats, i una plataforma més difícil amb un color vermell.

2.4.3 Porta

Com em explicat anteriorment, al llarg del joc trobem diverses escenes, però com es mourà el personatge entre elles?

La solució va ser crear una sèrie de portes. Inicialment eren portes invisibles però seguint les recomanacions del consultor es va optar per a incorporar una imatge de porta.

Vaig crear un game object amb un collider allargat de tipus "trigger" i es va afegir el script Puerta. Vaig declarar una variable identitat del nivell , i una variable de tipus booleana de passar de nivell. Aleshores el seu funcionament es simple , quan el personatge col·lisiona el collider del objecte, es carrega el següent nivell desitjat. Finalment vaig tenir que aplicar una petita modificació en la porta de les escenes 1 i 3, perquè la porta només funcionés quan la quest hagués estat assignada.



Figura 026. Porta de la escena

2.4.4 Trampes

He afegit una sèrie de esbarzers en punts estratègics de l'escenari amb l'intenció de ser una trampa mortal pel personatge. Tenen afegida l'etiqueta de "pinchos" que crido des de el script del personatge principal ([Protagonista](#)), i que funciona a través d'un "OnCollisionEnter", és a dir si el collider del personatge toca aquest objecte, el personatge morirà (serà destruït).

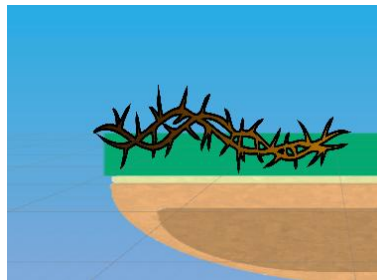


Figura 027. Trampes del joc

2.4.5 Enemics

[Erisomov](#)

Realitzar un personatge intel·ligent que ataca al nostre personatge era una tasca molt difícil sobretot tenint en compte que aquest era el meu primer joc. Per això vaig optar per a adaptar el codi de les plataformes sobre una animació. L'erició té una velocitat assignada i es desplaça d'un punt a un altre. A més té associada l'etiqueta de "erizo" que crido des de el script Protagonista , i que provoca que si coincideix amb el "collider" del personatge el "mata" però si el personatge ataca, i l'erició coincideix amb el "edge collider" de la cullera del personatge, succeeix al revés, i el erició és destruït.

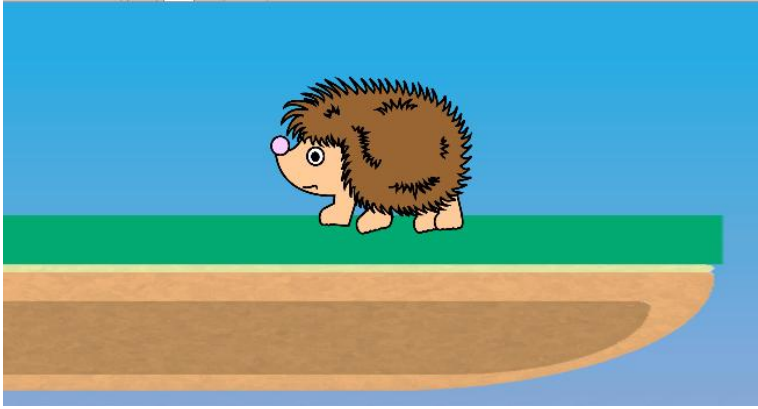


Figura 028. Eriçó enemic del joc

2.4.6 Ingredients

Els ingredients culinaris son les monedes del nostre joc, Son objectes col·lectables que el protagonista haurà d'aconseguir.

He col·locat els ingredients en diversos punts estratègics de l'escenari. Més endavant com explicaré en el punt 2.7 de la memòria, es crearà una funció en el script Protagonista, que permetrà al personatge recol·lectar-los, i que es vagin sumant en un comptador.

2.4.7 Killzone

La KillZone és la zona de la mort. És la zona inferior que simula el mar, on el personatge es precipita al caure de les plataformes. És una zona composta per un objecte rectangular molt gran, que té un box collider en forma de "trigger" que al traspasar el personatge el destrueix. Inicialment la idea era que li provoques un determinat mal , però al final no es va realitzar una barra de vida al personatge i vaig preferir que directament provoqués el game over.

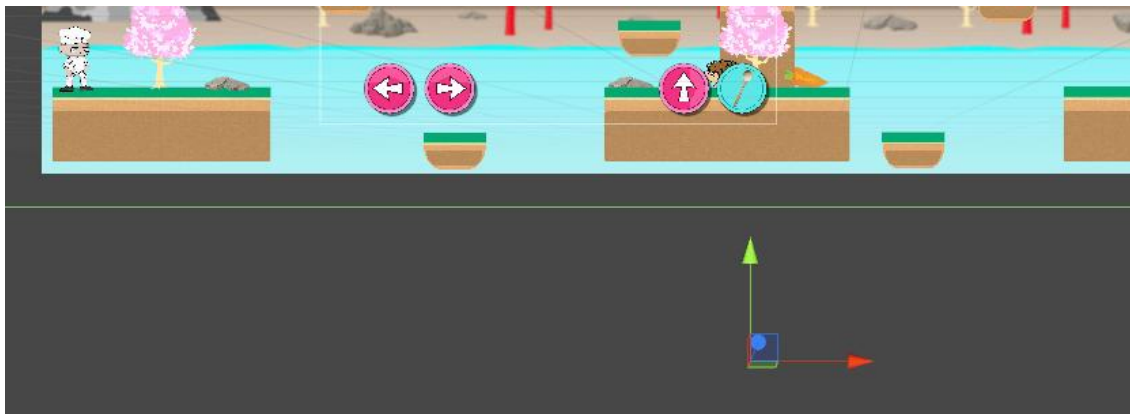


Figura 029. Zona Mort del joc

2.5 Control del personatge

2.5.1 Personatge Principal

Aquest punt el considero el més important i difícil del joc. El personatge és el protagonista, i l'element d'interacció entre el usuari i el joc.

Com s'ha vist anteriorment s'ha realitzat un conjunt d'imatge en forma de frame que simulen els diversos moviments del personatge. Al carregar al escenari un grup d'imatges se'ns genera una animació que és la que utilitzarem en el apartat "Animator". Amb el script [Protagonista](#) controlarem totes les accions del personatge a través dels diferents escenaris . A més he creat una instància de la classe principal que permet controlar les animacions a través de uns scripts de comportament ("behaviours").

[Jump Behaviour](#)

[Landd Behaviour](#)

[Layer Behaviour](#)

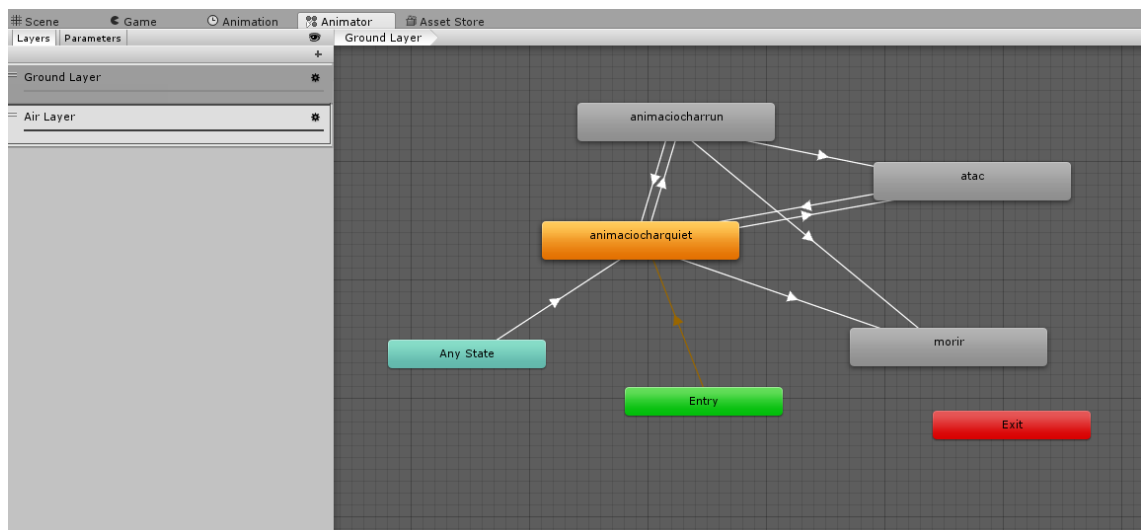
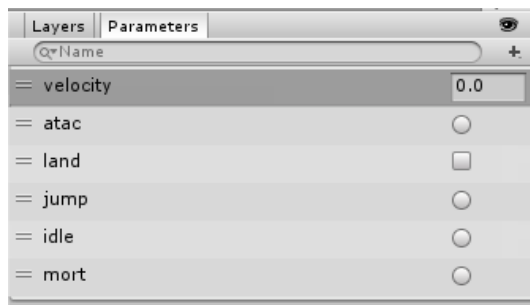
[Attack Behaviour](#)

2.5.2 Animator

L'animador és un component de Unity on es controlen les animacions. El punt d'origen és la animació del personatge quiet, conegut amb anglès com a "idle". A partir d'aquesta animació sorgeixen les demés. A més, en aquesta secció podrem controlar i programar la velocitat de cada animació i ajustar totes les transicions segons ens interessa.

He dividit l'animador en dues capes, una per quan el personatge està a terra i una per a controlar el comportament a l'aire, és a dir quan salta.

En les dues capes trobem tots els paràmetres, de tipus booleans, trigger i float, que s'utilitza en el codi per a controlar les animacions.



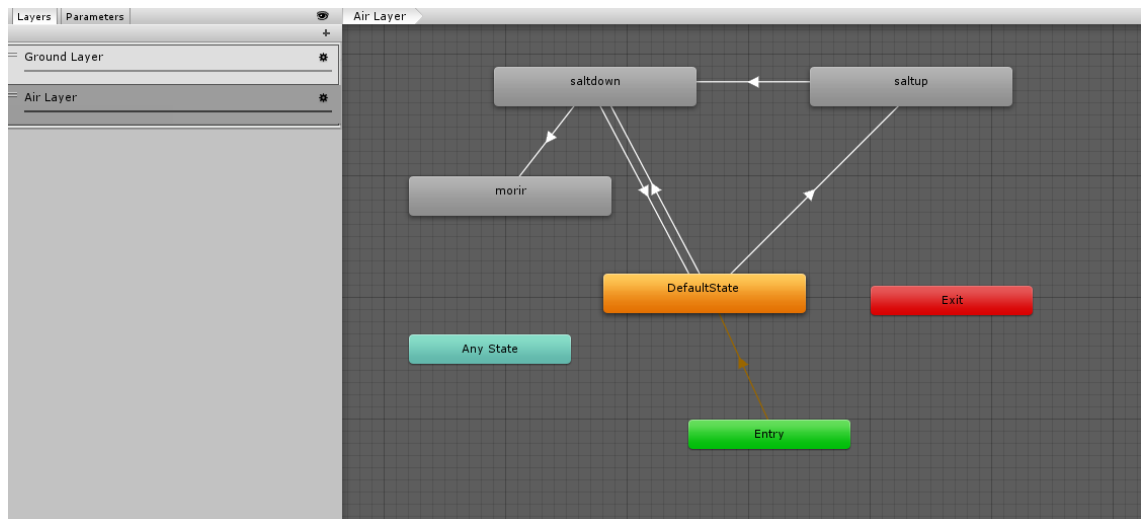


Figura 030 i 031 032. Animator del joc

2.5.3 Moviment

El personatge, que té un cos rígid 2D i gravetat, es mou a través dels escenaris de forma horitzontal, cridant els seus vectors. Té serialitzada la seva velocitat que controla el moviment del personatge en el eix de coordenades x, i respon a les tecles direccionals del teclat, o a les lletres w,s,d,a. També he afegit un efecte de “flip”, és a dir, que el personatge giri segons la direcció. Realitza el “flip” a través d’una funció simple posant en contraposició la propietat prèviament declarada (mirar dreta), i escalant un Vector 3.

2.5.4 Saltar

EL següent pas ha estat aplicar l’animació de saltar, a través d’un paràmetre booleà. Aquesta ha estat més complicada, i vaig acabar optant per a crear una altra capa com he explicat anteriorment. El salt té diferents estats, quan està saltant, o quan està tornant al terra, o bé caient d’una plataforma. Si cliquem la tecla “space” i el personatge no està caient, el personatge saltarà.

A més el salt tindrà una variable potencia de salt que afectarà al vector y, el vertical. Controlem la resta d'estats sabent si el personatge esta al terra (" is on ground") o no, i gràcies a la variable controlSalt.

2.5.5 Atacar

Una altra acció ha estat el atac, també a través d'un paràmetre booleà amb el codi del teclat shift. A més en el atac s'ha afegit un "edge collider" que només s'habilita al atacar durant uns segons, així es pot controlar com destruïm el enemic. És controla a través del script [mortspoon](#).

2.6 Càmera

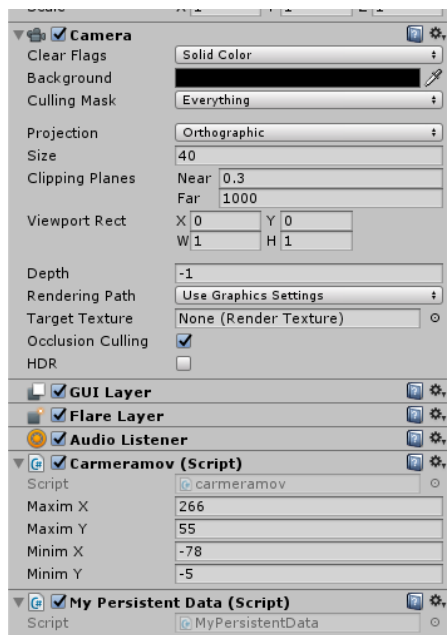


Figura 033. Preferències de la càmera

La càmera mostrarà la pantalla que visualitzarà el usuari al jugar. Al ser un joc de plataformes interessava un pla curt centrat en el personatge i que hi hagués un seguiment al seu moviment a través de l'escenari.

En el script [cameramov](#) he serialitzat quatre floats que utilitzaré com a punts mínims i màxims dels eixos de coordenades x i y . Per tant la càmera encara que es mogui a través del escenari mai superarà aquests punts dels límits de l'escena. Finalment he aplicat un target de seguiment del objecte "maincharacter", és a dir el personatge principal. Quan el personatge es desplaça a través de l'escenari la càmera el segueix, i mai supera els límits de l'escenari.

2.7 Quest

Un cop tenia el escenari, el personatge i la càmera funcional el següent pas va ser la construcció de l'escena 1, i l'assignament d'una missió al personatge.

En l'escena 1 trobem el NPC, el cuiner savi, i li he afegit un circle collider 2d extra 2D amb un radi força gran que al apropar-nos , i traspasar el "collider" s'inicia la missió.

Controlarem la quest amb els scripts [QuestManager](#) i [GameManager](#).

En el [QuestManager](#) controlem els estats de la Quest, assignant les condicions per a que estigui "sense assignar", "assignada" o "completada".

En el [GameManager](#) serveix per a serialitzar els objectes col·lectables, i el text que apareix en el canvas en cada estat de la quest.

2.8 Canvas

El canvas és una àrea designada on estan tots els elements d'UI (Interfície d'usuari). Aquesta àrea es mostra com un rectangle , i s'hi col·loquen els elements a dintre que després apareixen en la visió de la càmera del joc.

He acabat optant per un canvas que escala la mida segons la resolució. Per tant els elements dels canvas es veuran de forma molt semblant en diferents dispositius o resolucions. Aquesta solució ha estat necessària sobretot en la versió per a dispositius mòbils.

2.8.1 Comptador

El comptador es compon d'una imatge d'un sac, i un text numèric situats i ancorats a la part superior esquerra que augmenta cada cop que el personatge aconsegueix un ingredient, convertit a prefab. Trobem el codi al script [Protagonista](#).

Augmenta a partir d'un increment de punts que es manté a través de les escenes la funció `mypersistentdata`, però es reseteja al morir. Quan el personatge obté un mínim de 4, i retorna al collider circular del NPC en l'escena anterior, aconsegueix completar la quest.



Figura 034. Comptador dels ingredients

2.8.2 Text

Com em introduït anteriorment amb el GameManager controlàvem els estats de la quest. Per cada estat en les escenes inicials al apropar-nos al NPC ens apareixia un text. Aquest text el trobem en el canvas. És un text amb la tipografia Unipix, i està situat sobre un panell translúcid blanc.

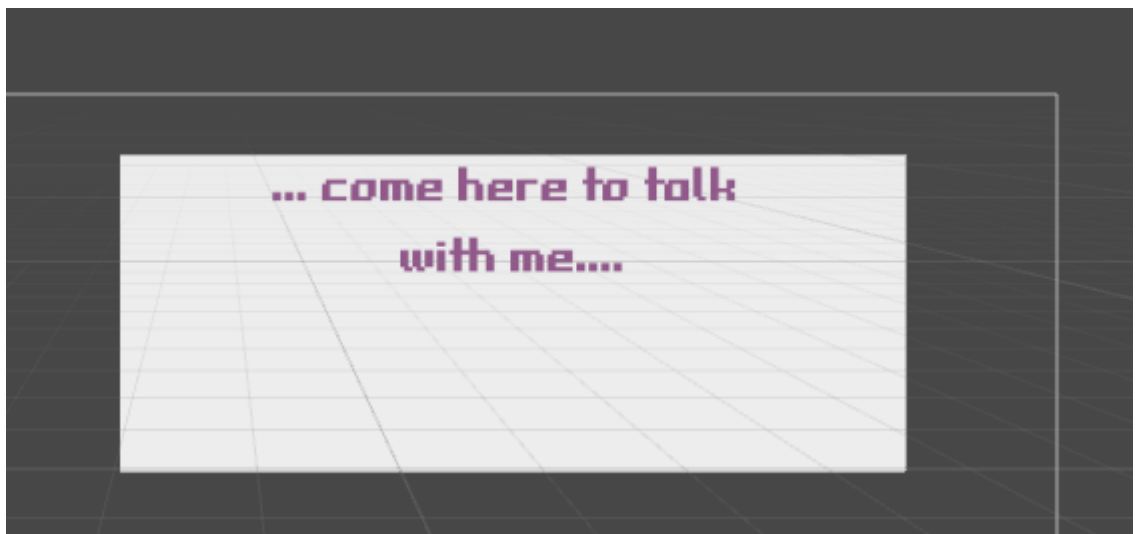


Figura 035. Panell i text del canvas

2.8.3 Botons Menu

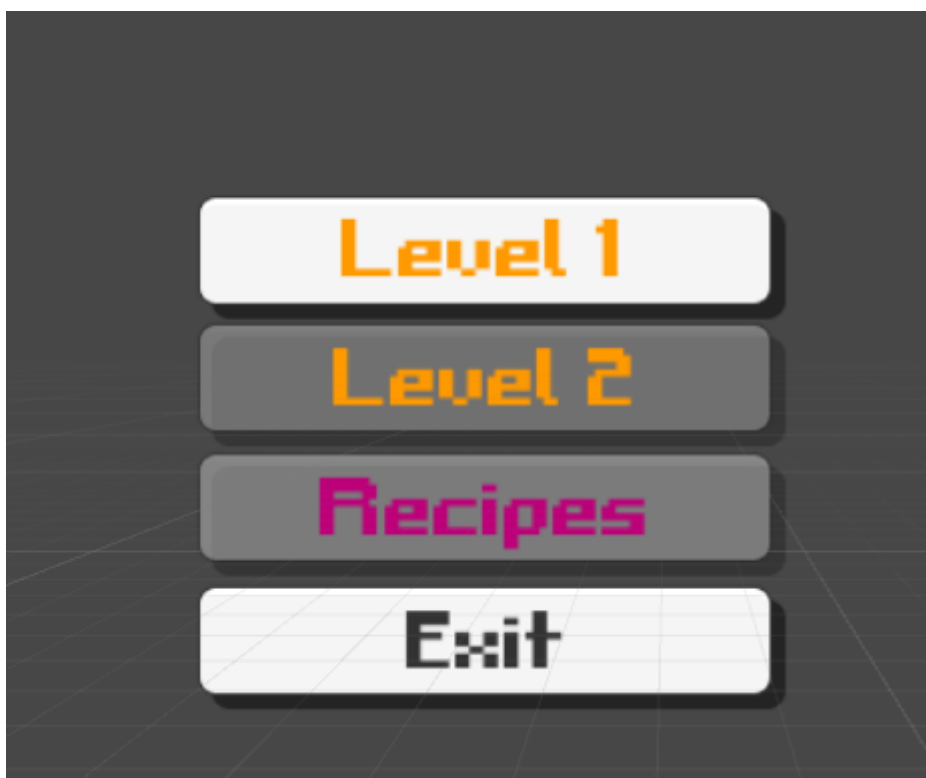


Figura 036. Botons del canvas del menu

Son uns botons molt simples. Funcionen amb el script basic de botó on click. Serveixen per a carregar les escenes o sortir. Només es té accés a les receptes o al nivell 2 un cop completat el nivell 1. Aquests valors es reinicien cada cop que s'obra el joc.

2.8.4 Botons Control

Primer es va produir una versió del joc per ordinador, i el control del personatge es realitzava gràcies a certs comandaments del teclat. En el cas de la versió per a dispositius mòbils o tablets, teníem un problema, ja que tenen un control tàctil. Es un tipus d'interacció complexa per als jocs de plataformes, i després de provar diverses opcions vaig optar per a crear uns botons tàctils amb els controls.

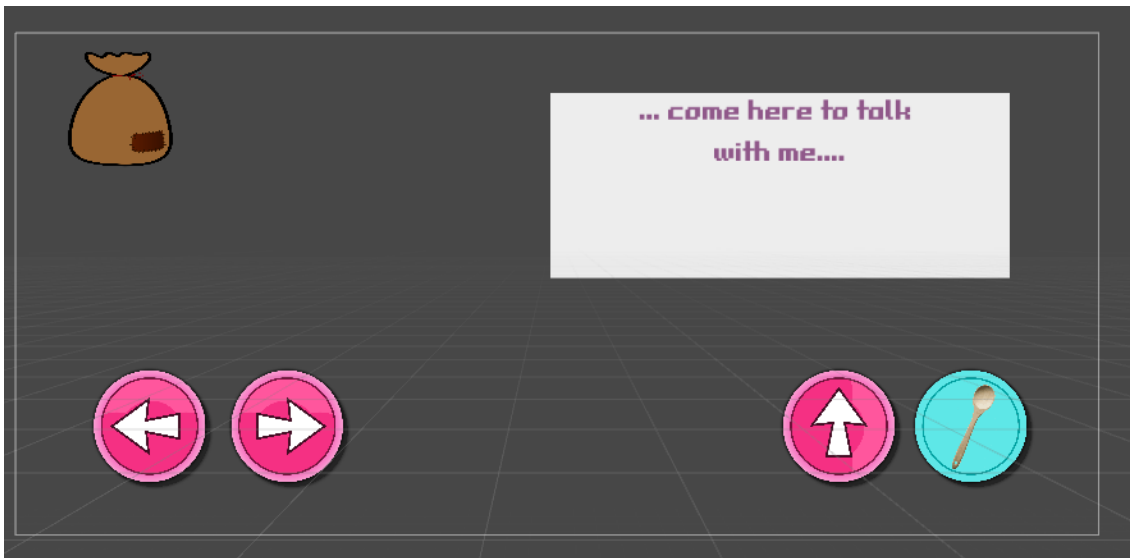


Figura 037 Botons de control per a dispositius mòbils

Els botons es controlen en el script [Protagonista](#). S'ha realitzat una sèrie de “voids” públics, on s'indiquen les accions de cada botó. Els botons direccionals son de tipus “event trigger”, és a dir, que quan es premen, l'acció es manté fins que es deixen anar.

En canvi les accions de atac i salt funcionen a través d'un “On Click”. Al prémer un boto es realitzar sol un cop l'acció.

2.9 Efectes

Un dels punts a destacar del meu joc és que he produït personalment gran part del joc, i no he utilitzat recursos ni “assets” addicionals. En el cas d'els efectes si que he fet servir alguns dels “Standard assets “ d'unity. He utilitzat un sistema de partícules de foc anomenat “fire complex” per a afegir l'efecte de fum i foc a la olla del cuiner.

En el Main Menu he utilitzat un efecte de “fireball” que genera un focus de llum darrera el títol del joc.



Figura 038. Efecte explosió de llum

2.10 Àudio i música

He produït les dues cançons del joc amb el programa Ableton Live 9. He utilitzat principalment un mòdul digital de Roland anomenat JV 1080 on tenia diversos bancs de sons de instruments asiàtics. També he utilitzat alguns samples de música japonesa dels anys 80. He carregat les cançons en les escenes desitjades.



Figura 039. Imatge pròpia dels instruments utilitzats.



Figura 039. Ableton Live

En quant als efectes de so, els he programat al script de Protagonista, cridant-los en les accions del joc (saltar, atacar, destruir al enemic o recollir un ingredient).

3. Conclusions

Estic molt satisfet amb el resultat final del treball de fi de grau.

Es normal acabar un grau universitari tant divers i polifacètic com Multimèdia amb la sensació que no s'ha après res. Després de realitzar assignatures tant diferents, podem dir que sabem una mica de tot, i a la vegada que no sabem res. En el cas del meu treball de fi de grau puc assegurar que al llarg del grau he assolit una sèrie de coneixements del funcionament digital que han afavorit i permès la realització del treball. Habilitats en disseny gràfic, coneixements bàsics de física, funcionament d'un escenari tridimensional, una base en programació suficient per introduir-me a un llenguatge nou com "csharp", tractament i publicació d'àudio, narrativa interactiva, disseny d'interacció...

Les primeres setmanes del treball van ser dures, doncs desconeixia per complet el funcionament de Unity, i qualsevol petit error en el codi, o descuit en el animator o les opcions de unity em suposava una pèrdua temporal molt alta. Poc a poc vaig anar agafant confiança amb el programa, gràcies a desenes de tutorials que trobem a la xarxa vaig anar assimilant coneixements, i resolent totes les propostes inicials de la primera pac. Aquests problemes amb el programa van provocar una entrega de la Pac 02 sense complir algunes de les fites proposades i un cert retràs temporal. Tot i així el retràs s'ha compensat, s'ha seguit el diagrama de Gant, i fins i tot he pogut afegir algunes característiques extres al joc.

Després de realitzar el meu primer joc , amb els coneixements actuals i confiança realitzaria una altra metodologia, i un altre tipus de joc, més enfocat als serious games, o jocs indies , amb un disseny i funcionalitats més úniques. Tot i així crec que realitzar un joc de plataformes , va ser una bona elecció ja que vaig tenir molt clar inicialment com seria el funcionament del joc, i els elements bàsics que formarien part. Un escenari, un personatge, plataformes en moviment, enemic i objectes col·leccionables. També m'agradaria destacar com he aconseguit gràficament l'estil japonès adequat al

argument del joc, i com he aconseguit dotar al joc de certa narrativa tot i ser un joc molt simple.

Al llarg del joc s'han aplicat molts canvis, s'han corregit tots els problemes i posat en pràctica totes les recomanacions del consultor. La variació principal la trobem en el script de control del personatge ([Protagonista](#)) que després de tenir problemes realitzant l'acció de salt, vaig decidir tornar a començar amb el script.

També vaig modificar el escenari, eliminant plataformes, i separant les existents per guanyar espai en l'escena, permetent així al personatge tenir més velocitat i salts més potents que afavoreixen la jugabilitat.

L'evolució del joc evidentment seria amb la creació de nous nivells, ingredients i receptes. Un joc on apeixessin noves receptes diàries en nous nivells i els usuaris les adquirissin mitjançant micropagaments. D'aquesta manera podríem tenir un públic fidel que juga diàriament a la nostre aplicació.

4. Presència Online

4.1 Google Play

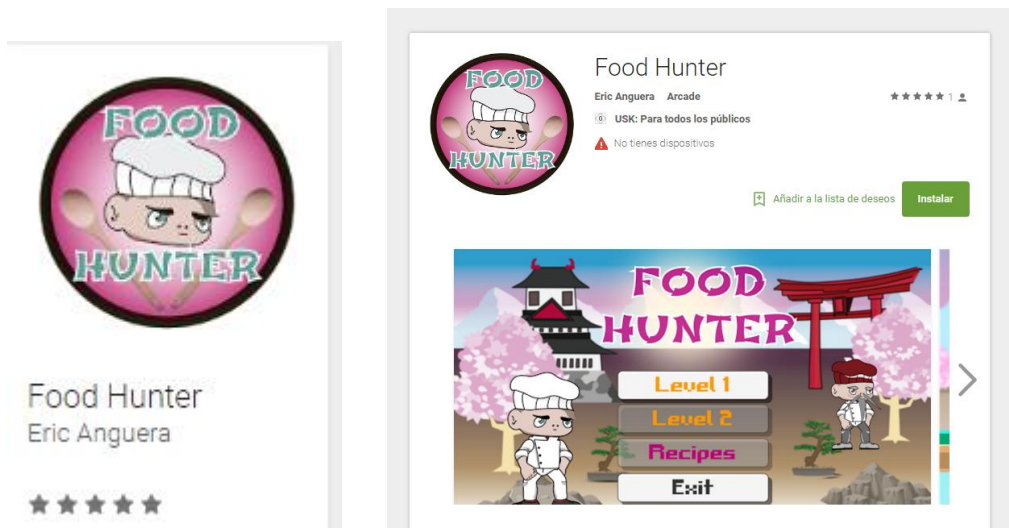


Figura 040 i 041 Imatges del joc a google Play

4.2 Youtube

Presentació publicitària del videojoc

<https://vimeo.com/221619139>

Explicació del funcionament Pac03

<https://vimeo.com/219372647>

5. Paraules Clau

ios: Sistema operatiu mòbil d'Apple.

Android. Sistema operatiu mòbil basat en el nucli de Linux.

Collider : Component que defineix la forma d'un objecte per a propòsits de col·lisions físiques.

Void: En programació, s'utilitza com a mètode que especifica que no retorna un valor

Rigid Body: Cos rígid que segueix unes lleis físiques.

Frame: Fotograma

Prefabs: Unió d'objectes per a crear un determinat Objecte del Joc.

Asset: Actius disponibles per utilitzar en la plataforma Unity.

Boolean: Computació que representa un valor de lògica binaria. (vertader/fals)

Float: Mètode per a declarar variables enteres decimals.

Kinematic: Estudi del moviment de cossos, sense les referències de la massa i la força.

6. Bibliografia

Real-Time Rendering. Tomas Akenine-Möller, Eric Haines, and Naty Hoffman, A.K. Peters Ltd., 3a edition, ISBN 987-1-56881-424-7, Year: 2008.

Game Engine Architecture . Jeff Lander, Jason Gregory. CRC Press, ISBN,978-1-43987-918-4. Year: 2009.

Tutorials:

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

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

Unity 5 2D Platformer Tutorial

https://www.youtube.com/watch?v=oK_NzdVSxaQ

Gamesplusjames Tutorial

<https://www.youtube.com/watch?v=86Bgt--Ww7w>

Exporting to android device (Unity3D)

https://www.youtube.com/watch?v=i_hPha28LEc

Publishing to Google Play (Mobile .APK file) - Game Mechanics - Unity 3D

<https://www.youtube.com/watch?v=4eJ13SpxqeM>

7. Scripts

Protagonista

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

public class Protagonista : MonoBehaviour
{
    /**
    public static Protagonista player = null;
    private static Protagonista instance;
    public static Protagonista Instance

    {

        get
        {
            if (instance == null)
            {
                instance = GameObject.FindObjectOfType<Protagonista>();
            }
            return instance;
        }
    }

    bool estoyvivo = true;
    float tiempoactual;

    private int ingredients;
    private Animator Animacio;
    [SerializeField]
    private float Velocidad;
    private bool mirarDreta;
    [SerializeField]
    private Transform[] groundPoints;
    [SerializeField]
    private float groundRadius;
    [SerializeField]
    private LayerMask whatIsGround;
    [SerializeField]
    private bool controlSalt;
    [SerializeField]
    private float potenciaSalt;
    [SerializeField]
    private EdgeCollider2D SpoonCollider;
    private float direction;
    private bool move;
    private float btnhorizontal;
    // *****
```

```

// *****
public Rigidbody2D elRigidbody { get; set; }

public bool Atac { get; set; }

public bool Jump { get; set; }

public bool OnGround { get; set; }
// crido la vida per utilitzar en un futur si es crea una barra de vida pel personatge
public static float Health {

    get
    {
        return hhealth;
    }

    set
    {
        hhealth = value;
        if(hhealth <= 0) {
            Die();
        }
    }
}
// falling

public bool esFalling
{
    get
    {

        return elRigidbody.velocity.y < 0;

    }
}

[SerializeField]
private static float hhealth = 100.0f;
public AudioSource jumpSound;
public AudioSource AtacSound;
public AudioSource CoinSound;
public AudioSource DeadSound;

// inicialitzem
void Start()
{
    mirarDreta = true;
    elRigidbody = GetComponent<Rigidbody2D>();
    Animacio = GetComponent<Animator>();

}

// la mort
void Update()
{
    if (estoyvivo)
    {
        HandleInput();
    }
}

```

```

    }
    else
    {
        if (Time.time >= (tiempoactual + 1))
        {
            Application.LoadLevel("escena Game Over");
        }
    }
}
// Update is called once per frame

void FixedUpdate()
{
    float horizontal = Input.GetAxis("Horizontal");

    OnGround = IsGrounded();
    if (estoyvivo)
        Moviment(horizontal);
// botonssss
    if (move)
    {
        this.btnhorizontal = Mathf.Lerp(btnhorizontal, direction, Time.deltaTime * 4);
        Moviment(btnhorizontal);

        Flip(direction);
    }
    else
    {

        Flip(horizontal);
        MovimentSalt();
    }
// Flip(horizontal);
// MovimentSalt();
}
//moviment i estats
private void Moviment(float horizontal)
{
    if (esFalling)
    {
        gameObject.layer = 9;
        Animacio.SetBool("land", true);
    }
    if (!Atac && (OnGround) || controlSalt)
    {
        elRigidbody.velocity = new Vector2(horizontal * Velocidad, elRigidbody.velocity.y);
    }
    if (Jump && elRigidbody.velocity.y == 0)
    {
        //elRigidbody.AddForce(new Vector2(0, potenciaSalt));
        Vector2 elvector = elRigidbody.velocity;
        elvector.y = potenciaSalt;
        elRigidbody.velocity=elvector;
        Jump = false;
    }

    Animacio.SetFloat("velocity", Mathf.Abs(horizontal));
}
// acces teclat

```

```

private void HandleInput()

{
    if (Input.GetKeyDown(KeyCode.Space) && !isFalling)
    {
        Animacio.SetTrigger("jump");
        jumpSound.Play();
    }

    if (Input.GetKeyDown(KeyCode.LeftShift))
    {
        Animacio.SetTrigger("atac");
        AtacSound.Play();
    }
}
// orientacio flip
private void Flip(float horizontal)
{
    if (horizontal > 0 && !mirarDreta || horizontal < 0 && mirarDreta)
    {
        mirarDreta = !mirarDreta;

        Vector3 Scale = transform.localScale;
        Scale.x *= -1;
        transform.localScale = Scale;
    }
}
//audio

//matem al personatge
void OnDestroy()
{
    player = null;
}
static void Die() {
    Destroy(Protagonista.player.gameObject);
}

public static void Reset() {

    Health = 100.0f;
}
// sabem si esta a terra

private bool IsGrounded()
{
    if (elRigidbody.velocity.y <= 0)
    {
        foreach (Transform point in groundPoints)
        {
            Collider2D[] colliders = Physics2D.OverlapCircleAll(point.position, groundRadius, whatIsGround);
            for (int i = 0; i < colliders.Length; i++)
            {
                if (colliders[i].gameObject != gameObject)
                {
                    return true;
                }
            }
        }
    }
}

```

```

    }
    }
    }
    return false;
}
// sino salta

private void MovimentSalt()
{
    if (!OnGround)
    {
        Animacio.SetLayerWeight(1, 1);
    }

    else
    {
        Animacio.SetLayerWeight(1, 0);
    }
}

//atac espassa

public void SpoonAttack()
{
    SpoonCollider.enabled = !SpoonCollider.enabled;
}

public void OnCollisionEnter2D(Collision2D other)
{
    if (other.gameObject.CompareTag("Erizo"))
    {
        if (Vector2.Distance(GetComponentInChildren<EdgeCollider2D>().transform.position,
other.gameObject.transform.position) < 100)
        {
            if (SpoonCollider.enabled)
            {
                Debug.Log("Destruir");
                Destroy(other.gameObject.transform.parent.gameObject);
                DeadSound.Play();
                return;
            }
        }
    }

    if (other.gameObject.CompareTag("Pinchos") || other.gameObject.CompareTag("Erizo"))
    {
        Debug.Log("Hola");

        Animacio.SetTrigger("mort");
        //Transicion de morir
        //Debug.Log("muriendo");
        estoyvivo = false;
    }
}

```

```

    tiempoactual = Time.time;
}

if (other.gameObject.tag == "ingredients")
{

    GameManager.Instance.CollecteIngredients++;
    MyPersistentData.IncrementarPuntos();
    CoinSound.Play();

    if (other.gameObject.GetComponent<QuestItem>().isQuestFinished())
    {
        Debug.Log("Quest Finalizado");
    }
    else
    {
        Debug.Log("Quest NO Finalizado");
    }

    Destroy(other.gameObject);
}
}

public void btJump()
{
    Animacio.SetTrigger("jump");
    Jump = true;

    // jumpSound.Play();
}

public void btAtac()
{
    Animacio.SetTrigger("atac");
    Atac = true;
    AtacSound.Play();
}

public void btMove(float direction)
{
    this.direction = direction;
    this.move = true;
}

public void btStopMove()
{
    this.direction = 0;
    this.btnhorizontal = 0;
    this.move = false;
}
}

```

Mov Plataforma

```

public class movplataforma : MonoBehaviour {

    private Vector3 posA;

```

```

private Vector3 posB;

private Vector3 segPos;

[SerializeField]

private float velocitat;

[SerializeField]
private Transform minTransform;

[SerializeField]
private Transform transformB;

// Use this for initialization
void Start () {

    posA = minTransform.localPosition;
    posB = transformB.localPosition;
    segPos = posB;

    }

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

        Move();

    }

private void Move ()
{
    minTransform.localPosition = Vector3.MoveTowards(minTransform.localPosition, segPos, velocitat *
Time.deltaTime);

    if (Vector3.Distance(minTransform.localPosition, segPos) <= 0.1)
    {

        CanviDesti();
    }
}

private void CanviDesti()
{
    segPos = segPos != posA ? posA : posB;

}
private void OnCollisionEnter2D(Collision2D other)
{
    if(other.gameObject.tag== "Player")

```

```

    {
        other.gameObject.layer = 8;
        other.transform.SetParent(minTransform);
    }
}
private void OnCollisionExit2D(Collision2D other)
{
    other.transform.SetParent(null);
}
}
}

```

Col plataforma

```

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

public class colplataforma : MonoBehaviour {

    private BoxCollider2D characterCollider;

    [SerializeField]

    private BoxCollider2D plataformaCollider;

    [SerializeField]
    private BoxCollider2D plataformaTrigger;

    // Use this for initialization
    void Start () {

        characterCollider = GameObject.Find("maincharacter").GetComponent<BoxCollider2D>();
        Physics2D.IgnoreCollision(plataformaCollider, plataformaTrigger, true);
    }

    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.name == "maincharacter")
        {
            Physics2D.IgnoreCollision(plataformaCollider, characterCollider, true);
        }
    }

    void OnTriggerExit2D(Collider2D other)
    {
        if(other.gameObject.name == "maincharacter")
        {
            Physics2D.IgnoreCollision(plataformaCollider, characterCollider, false);
        }
    }
}

```

Puerta

```

using System.Collections;

```



```

using System.Collections.Generic;
using UnityEngine;
//using UnityEditor.SceneManagement;

public class Puerta : MonoBehaviour {

    public int levelID = 1;

    [SerializeField]
    bool pasardenivel=false;

    void OnTriggerEnter2D(Collider2D otherCollider) {

        if (!otherCollider.CompareTag("Player")) {
            return;
        }
        //Application.LoadLevel(levelID);
        //GameObject.FindObjectOfType<QuestManager>()
        if ((QuestManager.GetQuestStatus("collectingredients") == Quest.QUEST_STATUS.ASSIGNED))
        {
            Application.LoadLevel(levelID);
            // EditorSceneManager.LoadScene(levelID);
        }

        if ((QuestManager.GetQuestStatus("collectingredients") == Quest.QUEST_STATUS.COMPLETED))
        {
            MyPersistentData.nivelesPasados++;
            Application.LoadLevel(levelID);
            // EditorSceneManager.LoadScene(levelID);
        }

        //}
    }
}

```

Cameramov

```

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

public class carmeramov : MonoBehaviour {

    [SerializeField]

    private float maximX;

    [SerializeField]

    private float maximY;

    [SerializeField]

    private float minimX;

    [SerializeField]

    private float minimY;

```

```

private Transform target;

    // Use this for initialization
    void Start () {

        target = GameObject.Find("maincharacter").transform;
    }

    // Update is called once per frame
    void LateUpdate ()
    {

        transform.position = new Vector3(Mathf.Clamp(target.position.x,minimX,maximX),
Mathf.Clamp(target.position.y,minimY,maximY),transform.position.z);
    }
}

```

MortSpoon

```

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

public class mortspoon : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

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

    }

    void OnCollisionHit(Collider collision)
    {
        Debug.Log(collision.gameObject.name);
        if (collision.CompareTag("Pinchos"))
        {

            DestroyObject(collision.gameObject);
        }
    }
}

```

Erisomov

```

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

public class erisomov : MonoBehaviour
{

```

```

private Vector3 posA;

private Vector3 posB;

private Vector3 segPos;

[SerializeField]

private float velocitat;

[SerializeField]
private Transform minTransform;

[SerializeField]
private Transform transformB;

// Use this for initialization
void Start()
{

    posA = minTransform.localPosition;
    posB = transformB.localPosition;
    segPos = posB;

}

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

    Move();

}

private void Move()
{
    minTransform.localPosition = Vector3.MoveTowards(minTransform.localPosition, segPos, velocitat *
Time.deltaTime);

    if (Vector3.Distance(minTransform.localPosition, segPos) <= 0.1)
    {

        CanviDesti();
    }

}

private void CanviDesti()
{
    segPos = segPos != posA ? posA : posB;

}

}

```

QuestItem

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

public class QuestItem : MonoBehaviour {

    public string questName;
    private AudioSource theAudio = null;
    private SpriteRenderer theRenderer = null;
    private Collider2D theCollider = null;
    public AudioSource coinsound;
    // Use this for initialization

    void Awake() {
        theAudio = GetComponent<AudioSource>();
        theRenderer = GetComponent<SpriteRenderer>();
        theCollider = GetComponent<Collider2D>();
    }
    //
    void Start()
    {

        gameObject.SetActive(false);
        if (QuestManager.GetQuestStatus(questName) == Quest.QUEST_STATUS.ASSIGNED);
        gameObject.SetActive(true);

    }

    public bool isQuestFinished()
    {
        if (GameManager.collectedIngredients < 4)
        {
            return false;
        }
        else
        {
            QuestManager.SetQuestStatus(questName, Quest.QUEST_STATUS.COMPLETED);
            return true;
        }
    }

    void OnTriggerEnter2D(Collider2D collider)
    {

        Debug.Log(this.name);

        if (!collider.CompareTag("Player"))
            return;

        if (!gameObject.activeSelf == false)
            return;

        QuestManager.SetQuestStatus(questName, Quest.QUEST_STATUS.COMPLETED);

        theRenderer.enabled = theCollider.enabled = false;
    }
}
```

```

        if (theAudio != null)
            theAudio.Play();
        }

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

        }
    }
}

```

QuestManager

```

using System.Collections;
using UnityEngine;

[System.Serializable]
public class Quest
{
    public enum QUEST_STATUS
    {
        UNASSIGNED = 0,
        ASSIGNED = 1,
        COMPLETED = 2
    };

    public QUEST_STATUS status = QUEST_STATUS.UNASSIGNED;
    public string questName = string.Empty;
}

public class QuestManager : MonoBehaviour
{
    public Quest[] quests;
    private static QuestManager manager = null;

    public static QuestManager managerInstance
    {
        get
        {
            if (manager == null)
            {
                GameObject questObject = new GameObject("Default");
                manager = questObject.AddComponent<QuestManager>();
            }
            return manager;
        }
    }
}

```

```

private void Awake()
{
    if (manager)
    {
        DestroyImmediate(gameObject);
        return;
    }

    manager = this;
    DontDestroyOnLoad(manager);
}

public static Quest.QUEST_STATUS GetQuestStatus(string questName)
{
    foreach (Quest q in managerInstance.quests)
    {
        if (q.questName.Equals(questName))
        {
            return q.status;
        }
    }
    return Quest.QUEST_STATUS.UNASSIGNED;
}

public static void SetQuestStatus(string questName, Quest.QUEST_STATUS newQuestStatus)
{
    foreach (Quest q in managerInstance.quests)
    {
        if (q.questName.Equals(questName))
        {
            q.status = newQuestStatus;
            return;
        }
    }
}

public static void Reset()
{
    if (managerInstance == null)
        return;

    foreach (Quest q in managerInstance.quests)
    {
        q.status = Quest.QUEST_STATUS.UNASSIGNED;
    }
}
}

```

GameManager

```

using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class GameManager : MonoBehaviour
{
    private static GameManager instance;

```

```

[SerializeField]
private GameObject ingredientPrefab;

[SerializeField]
private Text ingredientText;

public static int collectedIngredients;

public static GameManager Instance
{
    get
    {
        if (instance == null)
        {
            instance = FindObjectOfType<GameManager>();
        }
        return instance;
    }
}

/* private void Awake()
{

    DontDestroyOnLoad(Instance);
}*/

public GameObject IngredientPrefab
{
    get
    {
        return ingredientPrefab;
    }
}

public int CollectedIngredients
{
    get
    {
        return collectedIngredients;
    }

    set
    {
        ingredientText.text = value.ToString();
        collectedIngredients = value;
    }
}
}

```

Killzone

```

using System;
using System.Collections;

```

```

using System.Collections.Generic;
using UnityEngine;

public class KillZone : MonoBehaviour {

    private static KillZone instance;

    public float damage = 100.0f;

    void OnTriggerEnter2D(Collider2D otherCollider)
    {

        /* if (!otherCollider.CompareTag("Player"))
            return;

        Destroy(instance.gameObject);

        if( Protagonista.player !=null)    {
            Protagonista.Health -= damage * Time.deltaTime;
        }*/
        Debug.Log(otherCollider.tag);
        if (otherCollider.CompareTag("Player"))
        {
            Debug.Log(otherCollider.tag);
            Application.LoadLevel("escena Game Over");
        }
        else
        {
            Destroy(otherCollider.gameObject);
        }

    }

}

```

8. Anexos

8.1 Entregues anterior

TFG

PAC 01

Disseny del videojoc

Introducció

Al llarg del grau Multimèdia he adquirit un conjunt de coneixements molt variats. Des de disseny gràfic a disseny d'interacció, passant per diversos llenguatges de programació, post-producció audiovisual, narrativa interactiva, animació 3D, empenedoria empresarial ,gestió de projectes...

En resum, he après a generar , manipular i gestionar continguts molt diversos en l'àmbit Multimèdia, un dels sectors empresarials més importants, i que més ha crescut en els últims anys.

Aquests últims mesos vaig descobrir Unity, un motor de producció de videojocs que engloba gran part del que he estudiat al llarg del grau, per aquest motiu , tot i no tenir gaires coneixements previs, vaig decidir realitzar el treball de fi de grau amb aquest software, produint un videojoc.

Context

Vaig començar el grau Multimedia al 2009, un any on el smartphone tot just començava a introduir-se en el mercat de masses i era difícil imaginar-se que acabaria convertint-se en un instrument vital en la nostre quotidianitat. Per aquell temps encara existien veus detractores que seguien defensant que el telèfon era un aparell per a realitzar trucades, i el fet de tenir càmera de fotos, escoltar música o tenir jocs no era necessari. Avui dia es inimaginable sortir de casa sense el smartphone. Utilitzem la aplicació de mapes per a trobar una adreça, ens entretenim al metro jugant a diversos jocs, realitzem fotos de 12 megapixels que no tenen res a envejar de les càmeres reflex...

Per aquest motiu la meva intenció es realitzar un videojoc per a IOS/ Android.

1. Idea del joc

Nom provisional: Food Hunter

a. Descripció:

Serà un **joc de plataformes 2D** on el personatge protagonista que serà un aprenent de cuiner haurà de aconseguir una sèrie de ingredients que demani un personatge NPC (Cuiner) per a realitzar una recepta determinada. El personatge NPC es trobarà a la part inicial del escenari, i indicarà a partir d'una conversació textual quins ingredient necessita. A continuació el personatge principal iniciarà l'aventura movent-se per l'escenari on tindrà que superar una sèrie d'obstacles per aconseguir els ingredients i tornar a l'inici a entregar-los al cuiner . Al aconseguir els ingredients desitjats, el jugador obtindrà la recepta per escrit i podrà passar al següent nivell.

L'evolució lògica del joc serà un increment de la dificultat en els nivells, a través de diverses característiques encara per determinar. Algunes idees inicials podrien ser incorporar un cronòmetre que marqui un temps màxim, mostrar ingredients falsos, incrementar-la velocitat dels obstacles o plataformes...

b. Subgènere i referències

Els jocs de plataforma són un gènere que es caracteritza per tenir que caminar, córrer , saltar i escalar en una sèrie de plataformes i espadats, on apareixen enemics i on s'ha de recollir una sèrie de objectes per a completar el joc.

Es un gènere que es va popularitzar a la dècada dels anys 80 amb títols com Donkey Kong o Space Panic, i va ecllosionar amb el famós primer títol de la saga Super Mario Bros.



Donkey Kong 1981



Super Mario Bros 1985

Als anys 90 aquest gènere de joc es va estandarditzar tant en les consoles de 16 bits , com en les primeres consoles portàtils com la Game Boy o la Sega Game gear.

A finals de la dècada dels 90 es van introduir els primers jocs de plataformes 3D com Super Mario 64, tot i així mai superarà en popularitat als jocs 2D.

En els últims anys s'han recuperat moltes sagues, i el gènere va poder revitalitzar-se gràcies a consoles portàtils populars com Game Boy Advance o Ds.

En el context actual crec que és un gènere que es troba una mica paralitzat, ja que l'evolució del disseny del Smartphone actual, com en aparell tàctil , ha dificultat la interacció clàssica d'aquest gènere amb els seus usuaris.

c. Interacció

Com he explicat anteriorment un dels problemes actuals d'aquets gènere de videojocs és la interacció dels usuaris amb el principal dispositiu d'oci actual: el Smartphone.

Aquest problema d'interacció de joc de plataformes amb els actuals smartphones serà analitzat i estudiat al llarg del TFG, optant finalment pel disseny d'interacció del usuari que creiem més adient.

d. Plataforma de destí (mòbil, navegador, desktop,...)

La primera versió beta del joc serà per a utilitzar amb l'ordinador. Un cop el funcionament sigui comprovat , es procedirà una evolució per a utilitzar en la plataforma més accessible per a la majoria de usuaris, el telèfon mòbil. S'intentarà exportar una versió per **IOS** i una per **Android**.

2. Conceptualizació

a. Història, ambientació i/o trama

Un jove estudiant de cuina comença a treballar en un restaurant del seu poble. El seu objectiu es convertir-se en un gran cuiner per poder obrir un restaurant a SOUL, la capital del país.

Tot i que a priori no li fa gaire il·lusió, descobreix que el cuiner del restaurant del poble, era un dels cuiners més famosos a la dècada dels anys 60 a la ciutat de Soul, que un dia va tenir que fugir de la ciutat a causa d'amenaces de la màfia.

El joc estarà ambientat en un entorn gastronòmic amb estètica asiàtica.

L'objectiu del protagonista serà aconseguir els ingredients necessaris per a realitzar una determinada recepta que ens sol·licita el NPC.

b. Definició dels personatges/elements

PROTAGONISTA: Cuiner aprenent. Jove estudiant de cuina. Prové d'una família de pescadors i viu en un petit poble a les afores de la ciutat. El seu somni es convertir-se en el millor cuiner de la ciutat de Soul.

NPC: Cuiner expert. Cuiner/a molt vella i exigent. Era un cuiner molt famós en els anys 60, però va decidir marxar a un petit poble on viu sol en una cabanya.

Ingredients: Serà el element que el protagonista ha de recollir a través de contacte. Seran ingredients de diferents aliments.

c. Interacció entre els actors del joc

El protagonista es trobarà amb el NPC en el primer escenari, on al apropar-s'hi, s'iniciaria un text amb les explicacions corresponents dels ingredients necessaris per a la recepta.

d. Objectius plantejats al jugador

Recollir els diferents ingredients superant les dificultats dels escenari i retornar fins al NPC.

e. Disseny Gràfic:

Tots els elements gràfics del joc, com els icones, els personatges o el fons seran realitzats amb programari d'Adobe. Seguiran una temàtica gastronòmica, amb influències asiàtiques i japoneses.

S'utilitzarà una gamma cromàtica de colors de tipus pastel.

f. Inspiració



Ramen Master APP



Missy Messy's midnight ramen



Libreria d'Unity d'icones de menjar



Dibuix propi

g. Àudio:

Els efectes de so i la música seran produïts amb Ableton live.

3. Desenvolupament i Roadmap

a. Avaluació d'engines i kits de desenvolupament

Per a la producció del videojoc utilitzaré la eina multi plataforma de desenvolupament de Unity 5.

Inicialment en basaré en el kit basic de desenvolupament de Unity. Si utilitzo algun element de la asset Store al llarg del pla de producció, s'afegirà a la memòria final.

b. Planificació d'objectius

01/03/2017 – 08/03/2017 – Disseny inicial del projecte: Introducció, idea inicial, descripció, funcionament, jugabilitat, disseny gràfic.

PAC 01

09/03/2017 – 20/03/2017- Creació de tots els elements gràfics, narrativa i guió interactiu. **Corregir PAC1**

15/03/2017 – 01/04/2017 – Inici del projecte a Unity: resolució, càmera, llum, plans, fons, inserció elements 2d, personatge, col·lidors i cossos rígids, capes del joc i prefabs.

PAC 02

02/04/2017 – 15/04/2017 – Correccions, modificacions i millores de la primera entrega.

16/04/2017 – 01/05/2017-. Producció de la melodia principal i selecció d'efectes sonors.


01/05/2017 - 28/05/2017 - Exportar en diferents formats i realitzar proves amb usuaris. Corregir i millorar.

PAC 03

29/05 – 11/06/2017 - Correcció final, document final. Conclusions. + Pla de marketing. Pla empresarial.

PAC FINAL

c. Quantificació de temps i recursos per objectiu

		Nom	Durada	Inici	Fi
1		TFG	74d	01/03/2017	12/06/2017
2		Disseny inicial del projecte	5d	02/03/2017	08/03/2017
3		PAC1	1d	08/03/2017	08/03/2017
4		Creacio elements grafics: illustrator	7d	09/03/2017	17/03/2017
5		Narrativa i guió	5d	17/03/2017	23/03/2017
6		Inicialitzacio projecte Unity	10d	17/03/2017	30/03/2017
7		Crear escena, importar elements 2d, background,	3d	20/03/2017	22/03/2017
8		Llum i càmera. Incorporació del personatge	3d	23/03/2017	27/03/2017
9		Colliders i cossos rígids.	2d	27/03/2017	28/03/2017
10		Capes del joc i prefabs	4d	27/03/2017	30/03/2017
11		PAC 02	1d	31/03/2017	31/03/2017
12		Correccions, modificacions i millores de la primera entrega.	32d	05/04/2017	18/05/2017
13		Programar els elements. Moviment d'obstacles, i afegir dificultats	15d	10/04/2017	28/04/2017
14		Evolució moviment personatge	15d	18/04/2017	08/05/2017
15		Exportar a ios/android	6d	15/05/2017	22/05/2017
16		Disseny de nous nivells/dificultats	8d	03/05/2017	12/05/2017
17		Adequació/modificacions	8d	17/05/2017	26/05/2017
18		PAC 03	1d	26/05/2017	26/05/2017
19		Revisió/ Correcció	5d	05/06/2017	09/06/2017
20		Memoria Final	15d	18/05/2017	07/06/2017
21		PAC FINAL	1d	08/06/2017	08/06/2017

d. Planificació del projecte:

ARXIU ADJUNT

Anguera_Eric_TFG-GANT.pdf

PAC 2 – Versió parcial

Eric Anguera

Adjunto un vídeo amb el funcionament actual del videojoc. No ha estat possible aconseguir una versió jugable que explicar en aquestes setmanes. Tot i així estic força satisfet perquè ha estat la meva introducció a Unity i al seu funcionament, i crec que les pròximes setmanes el camí ha seguir es més clar i assequible.

09/03/2017 – 20/03/2017- **Corregir PAC1**

15/03/2017 – 25/03/2017- Creació de tots els elements gràfics, narrativa i guió interactiu

Narrativa

He realitzat una modificació a la ambientació i trama del joc.

Nova trama:

En un castell feudal, en l'època samurai als inicis de la restauració Meiji, un jove aprenent de cuina té la missió de preparar un banquet per l'emperador que està de visita. El xef oficial és un avi centenari, i envia al seu jove aprenent a recollir els ingredients per a realitzar cada recepta.

Personatge Principal: Shota

NPC: Koto

Han i Kai es troben en un escenari PREVI a la cuina del castell. Kai explica que per preparar una recepta pel banquet del senyor feudal necessita una sèrie de ingredients. Kai tindrà que superar una sèrie de obstacles per a recollir els ingredients.

Descripció del joc

Menu Inici: JUGAR – RECEPTES- CREDITS

Pantalla Inicial: Escenari cuina del castell. El NPC cuiner vell explica al jove que vol cuinar.

-Nivell 1: escenari amb plataformes flotants, obstacles, on s'han de recollir els ingredients

Pantalla Inicial

- Nivell 2: escenari amb plataformes flotants, obstacles, i enemics on s'han de recollir els ingredients

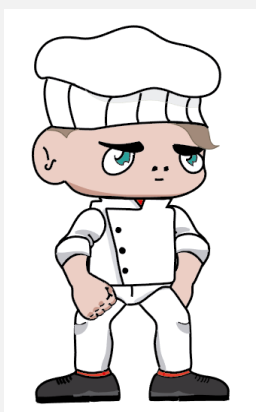
Pantalla Inicial

-Nivell 3:

Disseny Gràfic

Personatge Principal

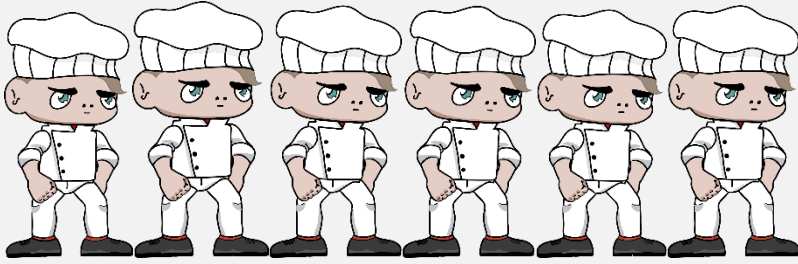
Vaig realitzar un dibuix amb vectors utilitzant la tableta gràfica Wacom Intuos i el software Illustrator. Vaig realitzar en diferents capes cada element del cos, per poder posteriorment realitzar les diverses versions.



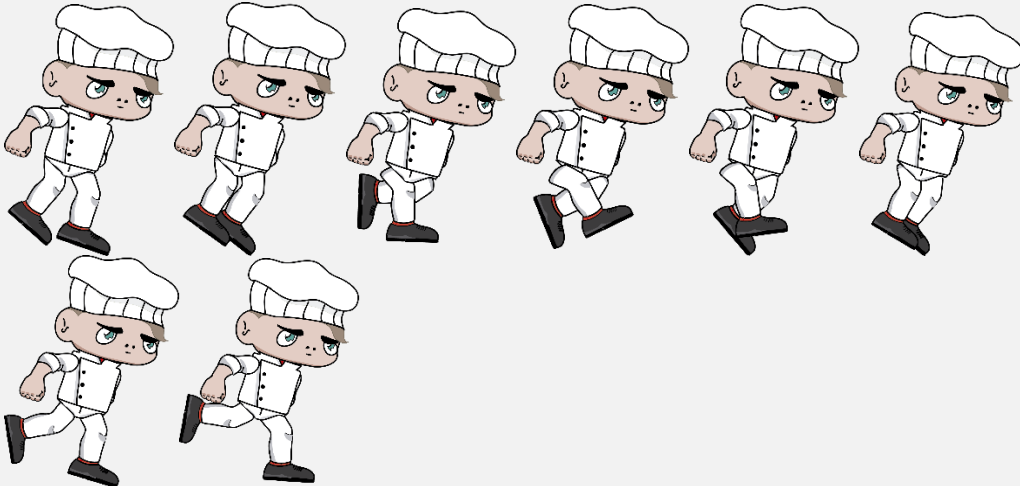
Shôta

Al incorporar el personatge a Unity van sorgir problemes de visualització, per això vaig tenir que augmentar el seu traç per estar millor definit.

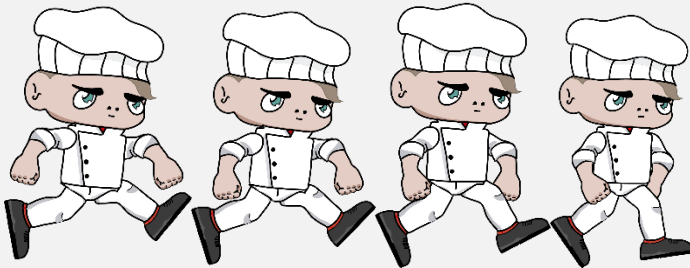
Versió Standard: simula un moviment de respiració



Versió caminant



Versió Saltant

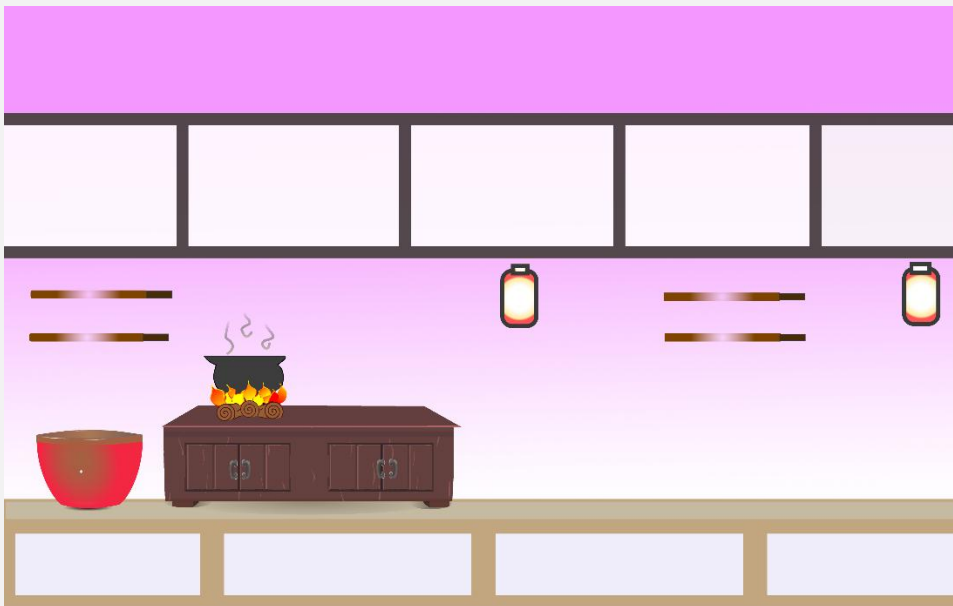


NPC



KOTO

Escenari inicial:

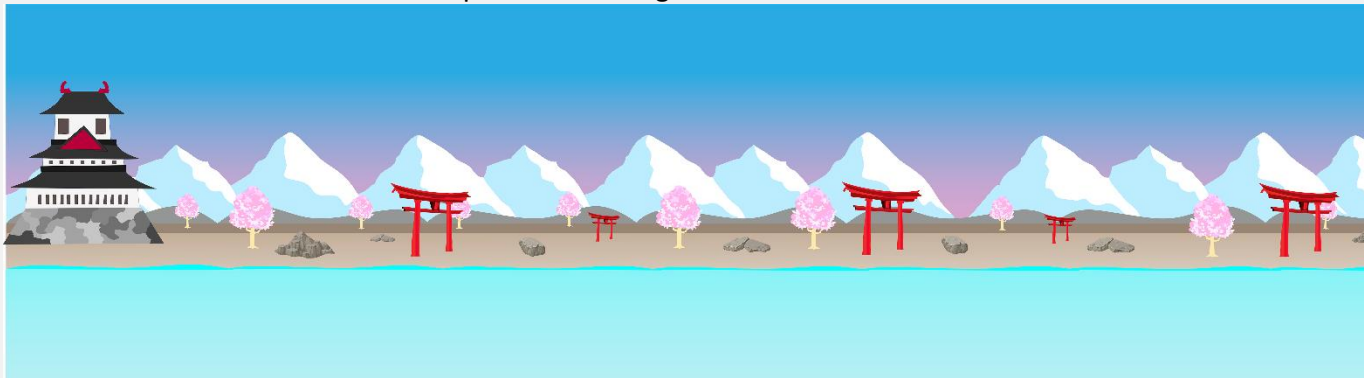


Escenari nivell 1:

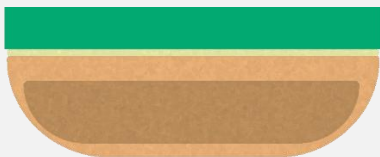
Seguint la estètica japonesa del argument introduït en la pac 01 vaig realitzar l'escenari.

Es un arxiu de Illustrator realitzat amb capes i modificable.

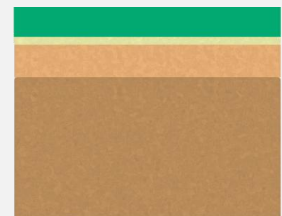
El escenari tindrà modificacions bàsiques de color segons el nivell.



Plataformes



plataforma voladora



Base

Plataforma terra

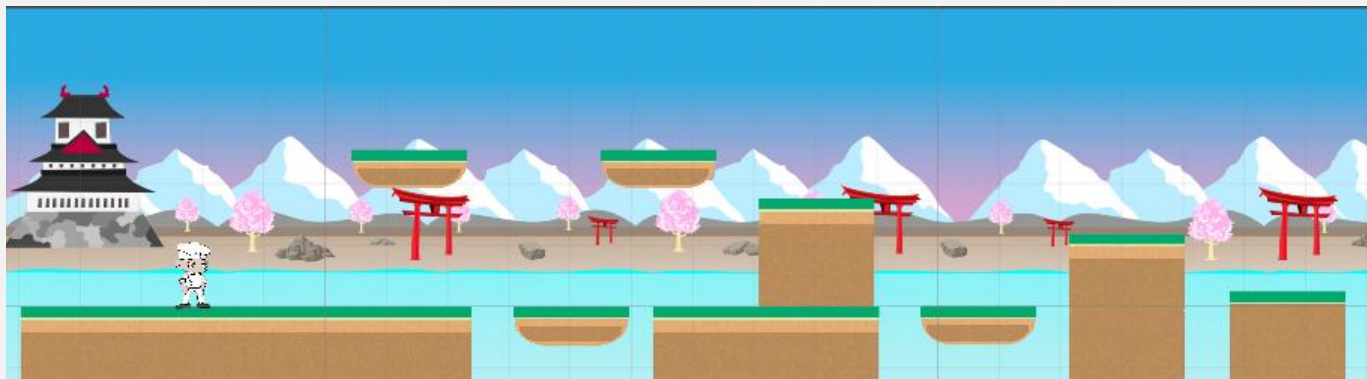
25/03/2017- 02/04/2017

Inicialització del projecte en UNITY

Escenari inicial



Escenari nivell 1



He col·locat el main character a partir dels 8 fotogrames mostrats anteriorment al escenari, en la posició de layer 1. El escenari en ordre de layer -1 i les plataformes en 0.

He escalat cada element tenint en compte la càmera.

Visió càmera (game).



Colliders

El Main character té un collider rectangular força estret, així quan s'apropa al precipici cau quan la major part del cos està fora de la plataforma.

En les plataformes he afegit diversos box colliders situats segons es requeria.

Control Personatge

He afegit un Rigidbody 2d , propietat bàsica en la que em basaré en el script.

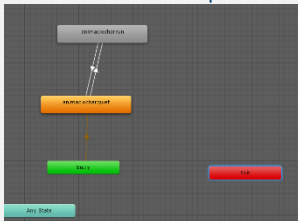
```
Rigidbody = GetComponent<Rigidbody2D>();
```

Per a realitzar el moviment creo un void moviment, on la velocitat i el moviment es regeixen a través de la estructura vector2 (molt útil en entorns 2d). Aquesta estructura es basarà en un moviment en els eixos horitzontals que crido en la funció "fixedupdate". S'utilitza fixed ja que estem controlant un rigidbody.

A continuació serialitzo la velocitat del moviment per controlar-la millor des de Unity. El següent pas ha estat el fer que el personatge giri verticalment segons la direcció (facing right/left), conegut com a "Flip". Segons la posició del vector sabrem o no si mira a la dreta a través d'una funció booleana. Per girar-lo tornem a utilitzar la estructura de vector 2 , en aquest cas per automatitzar el gir canviant la x de la escala del personatge a -1.

A continuació he iniciat el arbre del animator, activant l'animacio principal del character quiet, i creant una nova, quan corre. He cridat l'animator a través del script.

```
Animacio = GetComponent<Animator>();
```



De moment els valor de velocitat es de 20. I el de Gravatat es 3.

Objectius de cara la próxima entrega:

Escenari Inicial

- Moviment NPC
- Diàleg/ Text de QUEST recepta 1

Escenari Nivell 1

- Acció JUMP (saltar) i adequar-la al moviment.

- Disseny gràfic dels elements a recollir (ingredients).
- Seguiment de la càmera al personatge.
- Obstacles
- Acció atac*
- Crear enemic *
- Barra de vida
- Recol·lecció d'ingredients
- Moviment de plataformes
- Nivell 2,3..
- Transicions
- Cronòmetre (límit de temps)
- Menú inicial
- Efectes i Música.

PAC 3 – Versió jugable

INFORME



Eric Anguera

Context

Al ser el meu primer joc amb Unity i tenir una sèrie de problemes tècnics que no sabia solucionar, no va ser possible entregar una versió jugable en l'anterior entrega.

Tots els objectius en aquesta entrega s'han accelerat i s'han complert segons el calendari inicial.

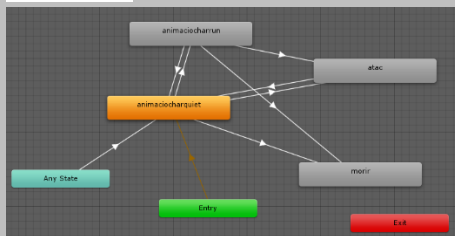
Unity

Com que no soc programador les primeres setmanes qualsevol error en el codi, o en comportament del animator, o qualsevol detall del inspector suposava la pèrdua de hores, i sobretot una desconfiança absoluta amb el que estava fent. En aquestes últimes setmanes he anat agafant seguretat amb el software, aprenent i entenent el funcionament del programa, i adquirint habilitats en `c#`. Reconec que el codi és una mica caòtic, i m'he deixat algunes parts que al final no utilitzo entremig per a futures evolucions del joc.

Main Character / Personatge Principal **Protagonista**

En l'anterior entrega vaig tenir problemes en aquest apartat, així que vaig decidir reiniciar el codi per a construir un script per al personatge més clar i funcional.

El personatge funciona a través de diferents animacions controlades a l'animator en dues capes, una capa de "ground" i una capa de "air". Cada estat té la seva pròpia animació.



Ground

Air

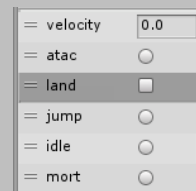
He creat una instància del personatge principal per a poder ser cridada per altres scripts que anomeno "behaviours" (comportaments). Utilitzo una sèrie de booleans i triggers per a cridar les accions.

El personatge es mou, gira i salta a través d'uns vectors horitzontals.

El salt també té una animació de caiguda. Finalment vaig afegir una animació de atac al clicar "shift" per utilitzar més endavant.

El script del personatge és el principal, que utilitzo posteriorment per crear la mort, la recollida d'ingredients i obtenir punts.

Escenari 1



He millorat el escenari 1, he afegit els objectes de la taula posteriorment per a que estiguin en una capa superior. He afegit un efecte de partícules de fum a la olla del cuiner.



NPC

He afegit una animació de moviment al NPC.

Plataformes en moviment **movplataforma colplataforma**

Un dels punts més simples del joc. He creat dues posicions vector 3, un moviment a la plataforma, i la funció destí. Cada plataforma seguint una velocitat definida, anirà fins a un punt seleccionat i tornarà al seu origen.

Les plataformes tenen un altre script que es diu colplataforma que utilitzo perquè el personatge col·lisió amb la plataforma i es mantingui a sobre, però que si salta des de baix pugui travessar-la.

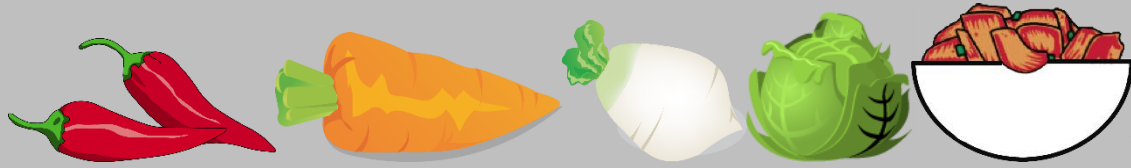
Seguiment de la càmera **carmeramov**

He definit dos punts màxims i mínims en les coordenades x,y fins on pot "moures" la càmera. També he afegit un target al personatge principal perquè el segueixi a través de l'escenari.

Escenari 2

He modificat i millorat el escenari principal, afegint plataformes i elements gràfics com arbres i roques. He creat un petit laberint que requereixi un nivell intermig per assolir els objectius.

Creació dels objectes a recol·lectar : “Ingredients”



Si el personatge col·lisiona amb els colliders dels ingredient els destrueix i obté punts. A més també els he convertit en prefabs per a ser utilitzats des de el “quest manager” “quest item”, i la seva obtenció el objectiu de la quest.

Quest Quest item Quest manager

L’objectiu es defineix al escenari 1 al apropar-se al circle collider del NPC, i la quest passa de “unassigned” a “assigned”. A la vegada es modifica el canvas de text creat on mostra les instruccions del joc. Si es retorna al NPC sense complir el objectiu, et diu que no ho has aconseguit. Per aconseguir el objectiu s’ha d’anar al següent escenari ,recollir els 4 ingredients, i retornar al NPC.

Comptador d’ingredients GameManager

Tot i que aquest script inicialment tenia un altre objectiu, l’utilitzo per a detectar els ingredients com a prefabs, i després realitzar un sumatori quan els personatge els destrueix.

Relacionar el comptador amb la quest em va suposar bastants problemes, per això vaig crear el script “my persistent Data”, i finalment un “dontdestroyonload”. El comptador és reinicia al morir, però es manté al tornar al escenari 1.

Porta Puerta

He creat un objecte buit amb un script que provoca que al interactuar amb el personatge (un objecte amb el tag “Player”) es carregui el nivell desitjat. He situat les portes en els llinars dels escenaris 1 i 2.

La mort KillZone

Inicialment volia crear una barra de vida i que es restés la vida poc a poc, però finalment vaig decidir una solució més simple. A través d’un objecte buit amb un collider “trigger” i un script que referència el tag “Player” amb el objecte. Així doncs quan el personatge cau pel precipici es carrega la escena de Game Over.

Escena Game Over

He creat una escena de game over, amb un botó de “restart”. He afegit un petit “efecte” dels Standard assets de Unity per simular la sang.



Escena Receita

L'objectiu del joc era aconseguir ingredients per a una recepta. Per tant al aconseguir els 4 ingredients, i tornar al NPC, ens felicita per l'objectiu aconseguït, i posteriorment al sortir del seu collider ens envia a una escena nova on trobem la recepta.

En aquesta escena he afegit un Standard asset de fum per al menjar, i un botó de retorn al menú inicial.

Menú Inicial botones

Es la primera escena del joc. Té tres botons (jugar , receptes i sortir).

Pinxos

He creat una trampa mortal, que provoca la mort al personatge quan cau en ella.

Enemic eriçó erisomov

Vaig intentar crear un enemic intel·ligent que ataqués al personatge però em va resultar massa complicat inicialment pels meus coneixements. Per això vaig tenir la idea de crear un enemic posant en pràctica altres funcions dels joc. Amb il·lustrador vaig crear 16 imatges d'un eriçó en moviment i que realitzava un gir. Aquesta animació li he aplicat el mateix script que a les plataformes, i així tenim un enemic que es mou pel camp, i que si coincideix amb el personatge el mata.

Audio

He produït una cançó amb ableton live, i la he carregat a totes les escenes. He utilitzat el mòdul de roland JV 1080 per obtenir els sons de tipus asiàtic.

Optimització

El joc ha arribat a pesar 800 mbs, degut sobretot a les textures. Per la meua inexperiència vaig carregar els pngs amb mides molt elevades que he tingut que comprimir RGBA DXT5 a 512. La mida actual és de 95 mbs.

També he ordenat els elements del joc en carpetes.

En els pròxims projectes tindrè molt més en compte l'ordre en els assets del projecte. Al iniciar el joc no sabia que era una animació, ni un controller, i ho tenia tot barrejat. Tenia molts elements duplicats , i objectes png que no utilitzava.

Logotip

He creta un logotip pel joc que utilitzo al exportar-l'ho.

Millores/ Correccions

- Aprofitar la acció atac per interactuar en el joc (per exemple per destruir el eriçó).
- Mantenir la cançó entre escenes i que no es reinici cada cop.
- Afegir alguns efectes d'àudio a les accions del personatge o elements del escenari.
- Recollir informació de companys que testegen el joc, i fer correccions.
- Si al final decideixo fer una versió per a dispositiu mòbil, he de fer una versió del joc molt més optimitzada i amb menys pes.
- Aplicar el controlador per a mòbil. Segurament utilitzaré un dels Standard assets d'Unity. L'altra opció seria crear botons per a cada acció del personatge.