

Marako's Trip, creació d'un videojoc

Xevi Costa Brugué
Grau en Multimèdia
Videojocs

Helio Tejedor Navarro
Jordi Duch Gavaldà
Javier Luis Cánovas Izquierdo

5 de gener de 2020





Xevi Costa Brugué

Aquesta obra està subjecta a una llicència de [Reconeixement-No-Comercial-SenseObraDerivada 3.0](https://creativecommons.org/licenses/by-nc-nd/3.0/es/) Espanya de Creative Commons

Fitxa del treball final

Títol del treball	
Nom de l'autor	Xevi Costa Brugué
Nom dels consultors	Helio Tejedor Navarro Jordi Duch Gavaldà
Nom del PRA	Javier Luis Cánovas Izquierdo
Data de lliurament (mm/aaaa)	01/2020
Titulació o programa:	Grau de Multimèdia
Àrea del Treball Final	06.587 TFG - Videojocs
Idioma del treball	Català
Paraules clau	runner, game, platformer
Resum del treball	<p>El present Treball de Fi de Grau s'emmarca dins de l'especialització en videojocs del Grau de Multimèdia cursat a la UOC (Universitat Oberta de Catalunya), i detalla el procés de creació del videojoc Marako's Trip, producte destinat a plataformes mòbils com iOS i Android.</p> <p>Marako's Trip és un joc de plataformes de desplaçament lateral en el que la nostra protagonista avança automàticament per l'escenari i l'usuari realitza diferents accions com saltar, nedar o volar tocant la pantalla del dispositiu. Dins del gènere de les plataformes en 2D, s'emmarca concretament al subgènere conegut com a 'runner'.</p> <p>En aquesta memòria podrem veure des de la gestació del projecte en els seus inicis, fins al seu desenvolupament mitjançant el motor gràfic Unity3D, així com el procediment detallat de creació de tots els assets i recursos que el componen utilitzant software de tercers.</p> <p>L'enfoc a l'hora de desenvolupar aquest videojoc és que sigui un producte 'free to play', i partint d'aquesta premissa detallarem les decisions que s'han pres a nivell de monetització del producte, ja sigui integrant micropagaments i publicitat.</p> <p>La conclusió que podem extreure del present treball és que actualment l'evolució de les eines de desenvolupament de videojocs i les plataformes de distribució fan possible i factible que una sola persona o un equip reduït, pugui dur a terme amb èxit el desenvolupament d'un videojoc amb un pressupost reduït.</p>

Abstract

This Final Degree's project focuses on video game creation as part of the Multimedia Bachelor's degree studied at the UOC (Open University of Catalonia) and looks into the processes involved in the creation of Marako's Trip, a game for mobile platforms using IOS or Android.

Marako's Trip is a side-scrolling platform game in which our protagonist automatically advances along the the background scenery and the player has to perform various actions such as jumping, swimming or flying by tapping the screen's device. Hence, within the genre of 2D platforms, the game would fit into the sub-genre known as "runner".

The written report will deal with how this project came to be, how it was initially conceived and how its development though the Unity 3D graphic engine, as well as the detailed procedure tasks involved in the the creation of all assets and resources that has been employed by using third-party software.

The premise adopted when developing this video game was that it would fall into the "free to play" category and therefore, decisions regarding how to fund/monetise the product -be these integrating micropayments or advertisement- will be looked at and explained.

The conclusion that we can draw from this project is that the evolution of video game tools and and the platforms that currently exist make it possible as well as feasible for a single person (or a small team), to successfully create and develop a video game with a limited budget.

Índex

1. Introducció,	7
1.1 Context i justificació del treball,	7
1.2 Objectius del treball,	7
1.3 Enfocament i mètode seguit,	8
1.4 Planificació del treball,	9
1.5 Breu sumari de productes obtinguts,	9
1.6 Breu descripció d'altres títols de la memòria,	10
2. Estat de l'art,	11
3. Definició del joc,	12
3.1 Breu descripció del joc,	12
3.2 Subgènere/referències jugables de videojocs ja existents,	13
3.3 Apartat artístic i referents visuals,	14
3.4 Tipus d'interacció juc-jugador,	15
4. Conceptualització,	16
4.1 Història, ambientació i trama,	16
4.2 Definició de personatges i elements,	16
5. Monetització del producte,	19
6. Disseny tècnic,	20
6.1 Entorn de desenvolupament escollit,	20
6.2 Requeriments tècnics de l'entorn de desenvolupament,	21
6.3 Inventari d'eines emprades,	21
6.4 Integració d'APIs de tercers,	22
6.5 Inventari i descripció d'assets i recursos del joc,	23

6.6 Inventari complet,	26
6.7 Esquema d'arquitectura del joc/components,	28
7. Valoració econòmica del treball,	34
8. Disseny de nivells,	35
9. Manual d'usuari,	37
10. Conclusions,	39
11. Glossari,	40
12. Bibliografia,	41

1. Introducció

1.1 Context i justificació del treball

Des dels seus inicis fins als temps actuals, els videojocs s'han convertit en una de les formes d'entreteniment més rellevants en el món digital, superant en facturació a altres productes audiovisuals, com per exemple el cinema. Com a jugador des de la meua infància amb un MSX que em va regalar el meu pare quan tenia 4 anys, em sento satisfet d'haver pogut gaudir d'aquest art des dels seus inicis i haver pogut viure la seva evolució al llarg dels anys fins arribar als temps actuals.

A nivell personal, sempre m'he sentit atret per el món de la informàtica i el disseny gràfic, podent centrar i especialitzar-me en el disseny i el modelat en 3D orientat a videojocs. El 2012 vam fundar Bitology Studios, un petit estudi de desenvolupament de jocs per a mòbils amb un company que s'encarregava de la programació mentre m'encarregava de la part gràfica, la seva integració i el disseny de nivells.

Dins d'aquest context i a l'hora de desenvolupar un projecte, he sentit de nou la necessitat de crear un videojoc, per ser un producte que per dur-lo a terme és necessari treballar de forma multidisciplinària. Com a Treball de Fi de Grau, la creació d'un videojoc ho he afrontat com un repte degut a que és un tipus de projecte que s'ha de dur a terme en un interval temporal molt reduït i amb uns recursos limitats. Davant d'això, és necessari marcar-se un objectiu realista i assolible i poder treballar sense pressa però sense pausa, tenint en compte que els compromisos laborals i familiars no permeten dedicar el 100% d'una jornada en el desenvolupament. Finalment, i a nivell personal, necessitava afrontar un projecte que em permetés fer-me sentir autosuficient i polivalent.

1.2 Objectius del treball

L'objectiu principal d'aquest treball és el de realitzar un videojoc per a plataformes mòbils, principalment iOS i Android, afrontant totes les seves fases de desenvolupament. D'aquest objectiu sorgeixen múltiples tasques associades a complir per poder dur a terme el projecte de manera exitosa:

- » Aprenentatge i aprofundiment en les eines de desenvolupament de videojocs, com per exemple el motor gràfic Unity3D, i investigació d'APIs per integrar serveis externs
- » Creació d'assets gràfics utilitzant el mínim de recursos aliens possibles per poder dotar de personalitat pròpia al producte. En aquest projecte tots els recursos utilitzats són propis exceptuant l'apartat sonor, tant per la música com per els efectes de so.
- » Monetització del producte, en un mercat on s'ha acostumat al consumidor amb productes que inicialment són gratuïts, però que en ocasions apliquen pràctiques de monetització que sobrepassen límits que podrien considerar-se ètics.
- » Desenvolupament de la promoció del producte en forma de tràiler, captures de pantalla i descripcions per a complimentar les fitxes de producte de les respectives plataformes de distribució d'aplicacions.

1.3 Enfocament i mètode seguit

El procediment estàndar per desenvolupar un videojoc des de zero és avaluar les diferents eines de desenvolupament que es troben a disposició dels usuaris i valorar la més convenient en funció del tipus de projecte a realitzar. Un dels plantejaments podria ser desenvolupar un motor des de zero, possibilitat que es descarta d'entrada per falta de coneixement i de temps, i perquè l'objectiu és acabar publicant el joc en dues plataformes diferents, iOS i Android, i s'està cercant maximitzar els resultats tenint en compte el temps disponible per dur a terme el projecte. Així doncs, el més convenient és utilitzar un software de desenvolupament multiplataforma, d'on podem veure una taula comparativa a continuació, d'alguns motors disponibles al mercat adequats per realitzar un joc en 2D, tot i que alguns també permeten l'ús de 3D.

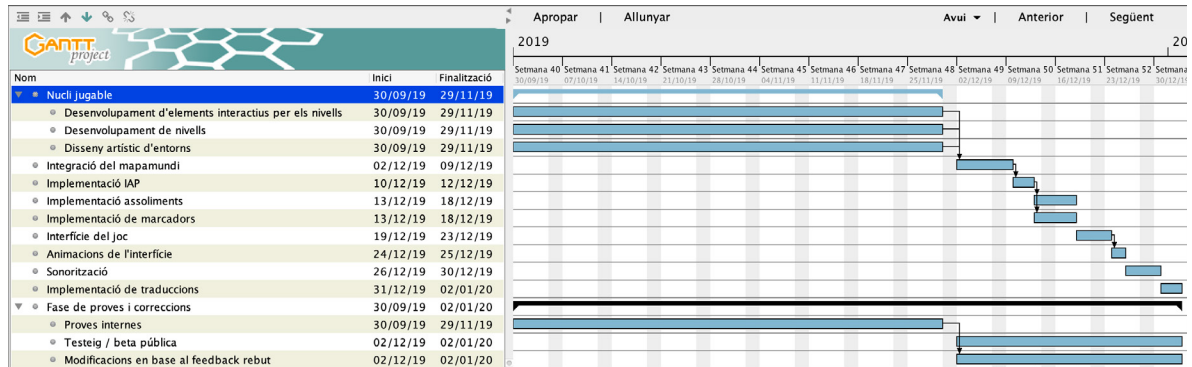
Nom del software	Llenguatge	Plataformes de publicació	Cost
Cocos2D	C++, Lua, JavaScript	iOS, Android, HTML5, Windows, Mac	Gratuït (Open Source)
Gamemaker	GML (Game Maker Language), propi	iOS, Fire, Android, UWP, Windows, Mac, Ubuntu, PS4, Xbox One, Switch	159\$/licència permanent amb exportació a iOS, Fire i Android
Unity3D	C#, JavaScript	iOS, Android, tvOS, PS4, Xbox One, WebGL, Facebook, Windows, Mac, Linux	Gratuït (per projectes que no superin 100.000\$ anuals)
Unreal	C++	Windows PC, PS4, Xbox One, Switch, Google Stadia, MacOS, iOS, Android, AR, VR, Linux, SteamOS, HTML5	Gratuït (amb pagament de comissió sobre els guanys obtinguts)
Godot	C++, C#, GDScript, Rust, Nim, D	Linux, macOS, Windows, FreeBSD, OpenBSD / DragonFly BSD, Android, iOS, BlackBerry 10, HTML5	Gratuït (Open Source)

Per el desenvolupament de Marako's Trip s'ha optat directament per utilitzar Unity per els factors següents:

- » Familiaritat prèvia amb l'ús del motor.
- » Gratuït en projectes que no superin els 100.000\$ anuals en guanys.
- » Relació temps invertit/qualitat del resultat molt equilibrada.
- » No s'està cercant un apartat tècnic que exprimeixi les plataformes a les que es destina.
- » Àmpliament extès en el desenvolupament mòbil i amb una gran comunitat al darrere.

1.4 Planificació del treball

Per dur a terme el projecte s'han identificat múltiples tasques i subtasques a realitzar, algunes amb dependències entre elles, i d'altres que s'han anat realitzant de forma paral·lela. Podem veure la planificació en el següent diagrama de Gantt.



En termes generals es pot considerar que s'han pogut realitzar totes les tasques planificades exceptuant la fase de proves i correccions. Si bé una desena d'usuaris l'han pogut provar en un entorn informal i el joc s'ha anat modificant en base al feedback rebut, serà necessari en el futur realitzar un qüestionari amb usuaris amb una versió que contingui estadístiques integrades amb Unity Analytics per poder veure de forma clara i precisa els punts febles del joc, que principalment i amb el feedback rebut tenen a veure amb la dificultat i poden provocar l'abandonament dels usuaris si el joc enlloc de generar un repte generés frustració.

Si bé la quantitat de contingut que s'ha creat és suficient per a mostrar totes les facetes jugables de Marako's Trip, de cara a un llançament formal a Google Play i l'App Store es fa necessari generar més nivells, o per el contrari ens trobariem davant d'un producte d'escassa duració i rejugabilitat.

1.5 Breu sumari de productes obtinguts

- » Paquet instal·lable per a Android en format APK. [Enllaç](#)
- » Executable en format AAB (Android App Bundle) publicat com a prova interna de Google Play, en format híbrid de 32 i 64 bits (podeu sol·licitar accés a la prova interna escrivint un correu a xevicb@uoc.edu).
- » Executable per a la versió d'iOS, disponible per a provar en dispositius Apple mitjançant TestFlight (podeu sol·licitar accés a les proves per iOS escrivint un correu a xevicb@uoc.edu).
- » Projecte de Unity amb tot el codi i els recursos audiovisuals que componen el joc.
- » Vídeo de presentació del joc i les seves funcionalitats. [Enllaç](#)
- » Tràiler de 30 segons per acompanyar les fitxes de producte de les respectives botigues (el tràiler simula que el joc ja està disponible a les diferents plataformes de distribució). [Enllaç](#)

1.6 Breu descripció dels altres títols de la memòria

Havent realitzat una introducció del projecte de desenvolupament de Marako's Trip, en els posteriors capítols d'aquesta memòria entrarem amb més detall a desgranar les diferents fases per les quals ha passat el joc.

- » **Capítol 2:** Revisió del gènere en el qual s'emmarca el joc i els dispositius en els quals està disponible, parlant també de les limitacions d'aquests últims.
- » **Capítol 3:** Revisió detallada del propi joc en sí i de les seves mecàniques, així com els referents en els quals s'inspira tant a nivell jugable com visual.
- » **Capítol 4:** Detall dels personatges que apareixen així com els elements interactius i no interactius que componen el joc a nivell conceptual.
- » **Capítol 5:** Tracta tots els aspectes relatius a la monetització del producte.
- » **Capítol 6:** Un dels capítols més extensos, tracta amb detall el programari utilitzat, la generació d'assets per el joc i el seu disseny a nivell tècnic.
- » **Capítol 7:** Valoració econòmica del cost de desenvolupament d'aquest projecte.
- » **Capítol 8:** Criteris de disseny de nivells.
- » **Capítol 9:** Veurem el manual d'ús del joc.
- » **Capítol 10:** Es tracten les conclusions extretes una vegada finalitzat el desenvolupament.
- » **Capítol 11:** Un glossari de termes utilitzats en aquesta memòria.
- » **Capítol 12:** Bibliografia.

2. Estat de l'art

Marako's Trip pertany al gènere de plataformes en dues dimensions, però per definir-lo de manera més acurada el podem emmarcar dins els subgènere dels 'runners'.

Si entenem com a joc tradicional un producte pensat per a ser gaudit mitjançant un teclat o un comandament, sigui en un PC o una videoconsola, veurem que intentar traslladar exactament la mateixa experiència a dispositius que com a mètode d'entrada utilitzen una pantalla tàctil no sempre resulta satisfactori. Molts jocs tradicionals son adaptats a aquests dispositius utilitzant un esquema de controls virtuals sobreimpresos a la pantalla, que comporten principalment dos hàndicaps importants: per una banda, els controls tapen bona part de la zona visible del joc, i per l'altra, es perd el factor precisió que ens ofereixen els comandaments físics, ja que moltes vegades l'usuari no toca el botó que voldria provocant accions indesitjades i com a resultat, la frustració del jugador. A conseqüència d'això, el subgènere dels runners es popularitza en els dispositius tàctils, davant de la necessitat d'oferir un sistema de joc i un esquema de control simplificat que encaixi perfectament amb aquest tipus de dispositius.

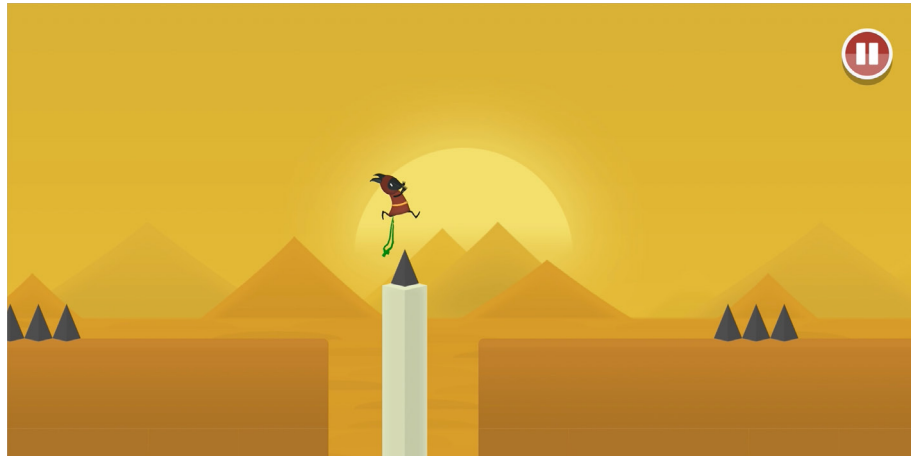
Per superar les limitacions en el control esmentades anteriorment, en un runner el personatge que controlem generalment avança de forma automàtica per l'escenari, i la tasca de l'usuari és la de realitzar diferents accions mitjançant la pantalla tàctil, generalment tocant o realitzant moviments lliscant el dit sobre qualsevol zona de la pantalla, evitant així l'ús de controls virtuals sobreimpresos en una regió concreta. Així doncs, es podria dir que en aquest tipus de jocs, el més primordial és el timing a la hora de realitzar les accions, ja que el jugador haurà d'estar concentrat per realitzar-les en el moment adequat.

El plantejament de Marako's Trip s'enfoca com un homenatge als videojocs clàssics de plataformes on disposem d'un únic objectiu, que és el de recórrer un nivell fins arribar a la meta per completar-lo.

3. Definició del joc

3.1 Breu descripció del joc

Marako's Trip es un joc de plataformes en 2d de desplaçament lateral en el qual haurem de completar nivells sorteiant obstacles saltant, nedant o volant per aconseguir-ho.



Marako saltant obstacles en els nivells que transcorren a peu



Marako nedant en un nivell aquàtic

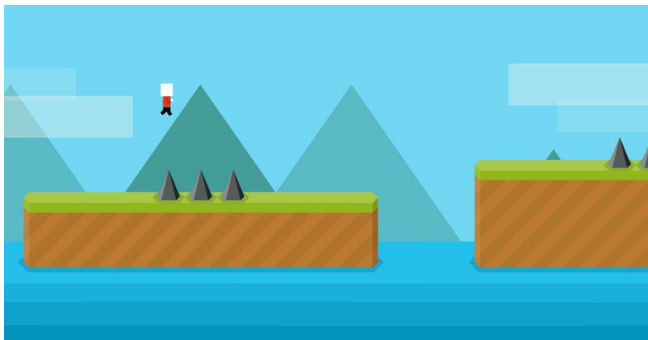


Marako volant després de recollir un jetpack

3.2 Subgènere i referències jugables a videojocs ja existents

Malgrat que el joc es pot emmarcar dins el gènere de plataformes, concretament en 2d, el podem definir com a runner. En el subgènere dels runners podem destacar els que son infinits, on els nivells es generen aleatòriament de forma procedimental, o els que no ho son, essent els nivells compostats de manera manual com la majoria de jocs de plataformes clàssics, i el projecte Marako's Trip pertany a aquest últim cas.

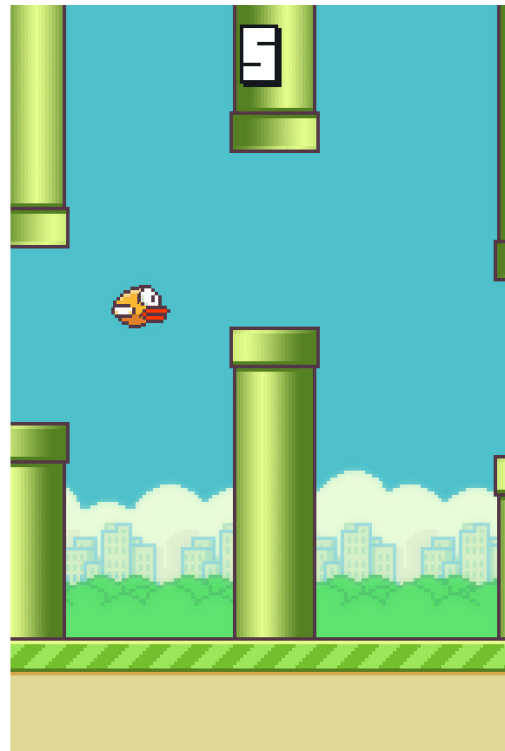
Com a referents principals, el joc s'inspira molt en la jugabilitat de Mr. Jump, i en menor mesura agafa elements d'altres runners com Run Sheldon, Flappy Bird, Jetpack Joyride i finalment algunes referències a nivell de jugabilitat de VVVVVV.



Mr. Jump (1-button SARL)



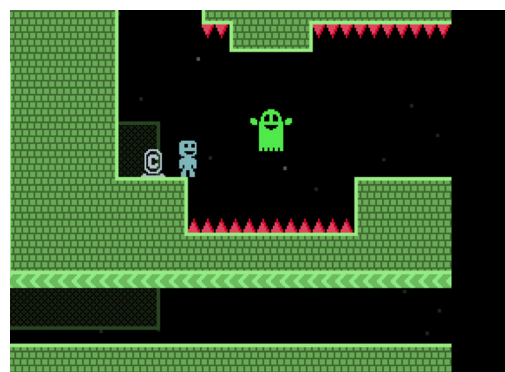
Run Sheldon (Bee Square)



Flappy Bird (Dong Nguyen)



Jetpack Joyride (Halfbrick)



VVVVVV (Terry Cavanagh)

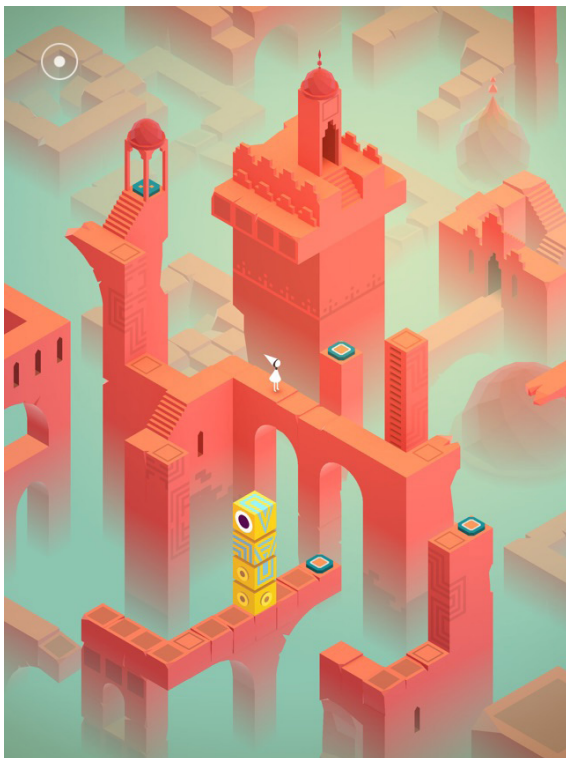
3.3 Apartat artístic i referents visuals

Donat que el temps de realització d'un Treball de Fi de Grau és limitat, per el projecte Marako's Trip s'ha optat per cercar un apartat visual molt minimalista, d'aparença "Flat Design". El joc s'inspira en referents per definir el seu aspecte visual com Journey, Monument Valley, i Alto's Odyssey.

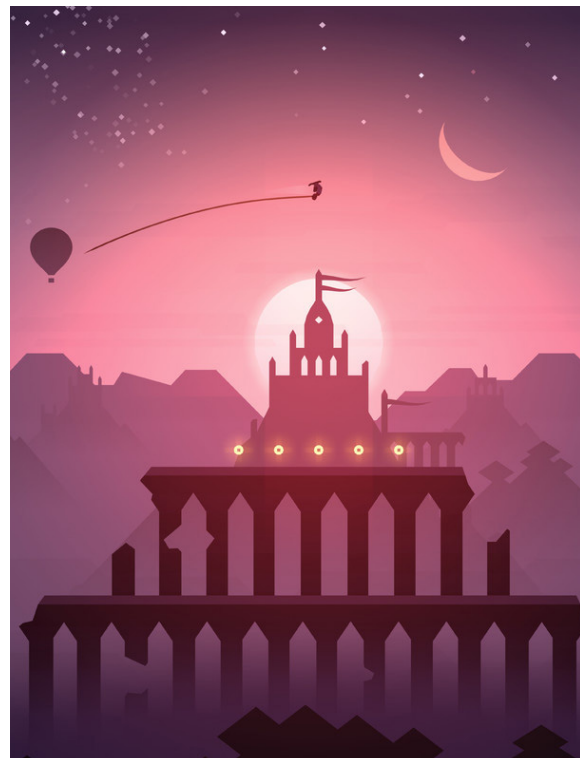
Partint d'aquestes referències visuals es decideix començar a generar els sprites que conformaran el joc. Es busca que l'apartat visual sigui de creació pròpia, evitant l'ús d'assets generats per tercers.



Journey (ThatGameCompany)



Monument Valley (ustwo games)



Alto's Odyssey (Noodlecake Studios)

3.4 Tipus d'interacció joc-jugador

En els runners tàctils, en la majoria de casos el personatge avança sol per el nivell, i el tipus d'interacció que el jugador pot fer és tocar qualsevol zona de la pantalla tàctil per realitzar accions com saltar, essent essencial el timing en el qual el jugador executa les accions. D'aquesta manera, s'aconsegueix evitar la imprecisió dels controls tàctils si ho comparem amb un joc de plataformes tradicional que faci ús d'un gamepad.

Podem distingir tres enfocaments diferents en quant a jugabilitat, que provocaran que al tocar la pantalla la Marako realitzi una acció o una altra en funció del context en el qual es trobi el jugador.

- » **Salt:** En la majoria de nivells, Marako avançarà per l'escenari corrent. En aquest cas, tocar la pantalla provocarà que aquesta faci un salt, més o menys elevat en funció del temps en el que el jugador mantingui el dit tocant la pantalla.
- » **Nedar:** Alguns nivells transcorren sota aigua, i Marako nedarà automàticament cap endavant. Un toc a la pantalla implica que aquesta nedi impulsant-se cap amunt. En aquest context jugable mantenir el dit més o menys temps prement la pantalla no implica cap canvi en el moviment. El referent més conegut semblant a aquest tipus de nivells seria Flappy Bird.
- » **Jetpack:** En alguns nivells Marako troba un Jetpack, que al recollir-lo es posa a mode de motxilla i li permet volar. En aquests casos, mantenir el dit a la pantalla implica que Marako s'elevi de forma indefinida, i deixar anar el dit desactiva el jetpack i fa que caigui per efecte de la gravetat. Els nivells de jetpack es combinaran amb les fases de runner, canviant la jugabilitat de forma dinàmica enmig del nivell en curs. Aquestes seccions es juguen de forma similar a Jetpack Joyride, un altre conegut runner per a mòbils.

4. Conceptualització

4.1 Història, ambientació o trama

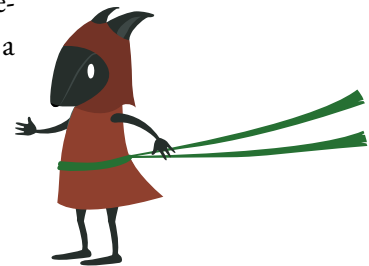
Per aquest projecte no s'ha cregut necessari dotar-lo de narrativa en homenatge als jocs de plataformes clàssics dels anys 90, per tant no es preveu dotar-lo d'una història. En quant a ambientació Marako visitarà una diversitat de localitzacions diferents que inclouen un desert, un bosc, una fàbrica, una illa tropical i caverne de lava, amb previsió de què es puguin ampliar els nivells disponibles en futures actualitzacions.

4.2 Definició de personatges i elements

Malgrat que inicialment s'havia decidit afegir enemics estàndards a les fases, algunes proves realitzades posaven en dubte de què el plantejament funcionés correctament degut a la velocitat a la que transcorren les fases. Per aquest motiu, s'ha decidit finalment implementar només enemics de final de món o 'Final Bosses'. A continuació passem a detallar-ne alguns:

Personatges/enemics

- » **Marako:** Protagonista del joc. Avança per l'escenari de manera automàtica i realitza diferents accions quan l'usuari toca a qualsevol zona de la pantalla segons el context on es trobi.
- » **Serp:** Boss del món del desert.
- » **Piranya:** Boss en una fase aquàtica de final de món.
- » **Robot:** Boss del món industrial.



Elements interactius

- » **Sprites d'entorn:** Generalment sprites que actuen a mode de terra per on la Marako pot avançar. És un dels elements més bàsics a l'hora de compondre l'escenari.
- » **Plataformes mòbils:** Plataformes flotants que es poden desplaçar fent un recorregut vertical, horitzontal o diagonal de manera cíclica.
- » **Punxes:** Element molt comú en el joc que elimina a la Marako amb el simple contacte.
- » **Jetpack:** Objecte recolectable que la Marako es posa a l'esquena i li permet volar.
- » **Monedes:** Elements que es troben presents als nivells i augmenten la puntuació obtinguda. Les monedes obtingudes s'acumulen i permeten desbloquejar els accessos a nous mons.
- » **Inversors de gravetat:** Elements que quan entren en contacte amb la Marako inverteixen la gravetat, permetent caminar per el sostre de l'escenari.
- » **Catapultes:** Presents a la fase final del desert, ens permet llençar un projectil contra el 'Final Boss' que ens està perseguint.
- » **Núvol:** El núvol funciona a mode de plataforma flotant que ens permet realitzar un salt ad-

dicional quan hi entrem en contacte durant un petit instant, abans de fondre's i desaparèixer.

- » **Portal de sortida del nivell:** Portal dimensional que al entrar-hi en contacte ens permet donar una pantalla com a finalitzada.
- » **Saltador/Jumper:** Plataforma amb una molla que al entrar en contacte amb la protagonista l'impulsa amb força cap amunt.
- » **Turbulència d'aigua:** Només a les fases aquàtiques, quan la Marako entra nedant dins d'una corrent aquàtica, aquesta l'empeny amunt o avall.
- » **Punt de control:** Quan la Marako passa per un punt de control, el progrés del nivell queda registrat. En cas de morir, s'ofereix la opció de ressuscitar a un punt de control a canvi de visualitzar un vídeo publicitari recompensat.
- » **Zoom in/out:** Element invisible consistent en un collider de tipus trigger que quan entra en contacte amb la protagonista realitza un zoom in o un zoom out de càmera, en funció de si interessa mostrar més o menys regió de l'escenari en un moment concret.
- » **Activadors/Desactivadors de portes:** Elements que quan entren en contacte amb la Marako obren o tanquen una porta (No implementats en aquesta versió del joc).

Aquest apartat queda obert de cara a la versió definitiva que es publicaria a les stores, ja que aquesta quantitat d'elements es preveu insuficient per mantenir la varietat jugable i serà necessari desenvolupar i afegir nous elements a fi de mantenir la jugabilitat fresca.

Elements no interactius

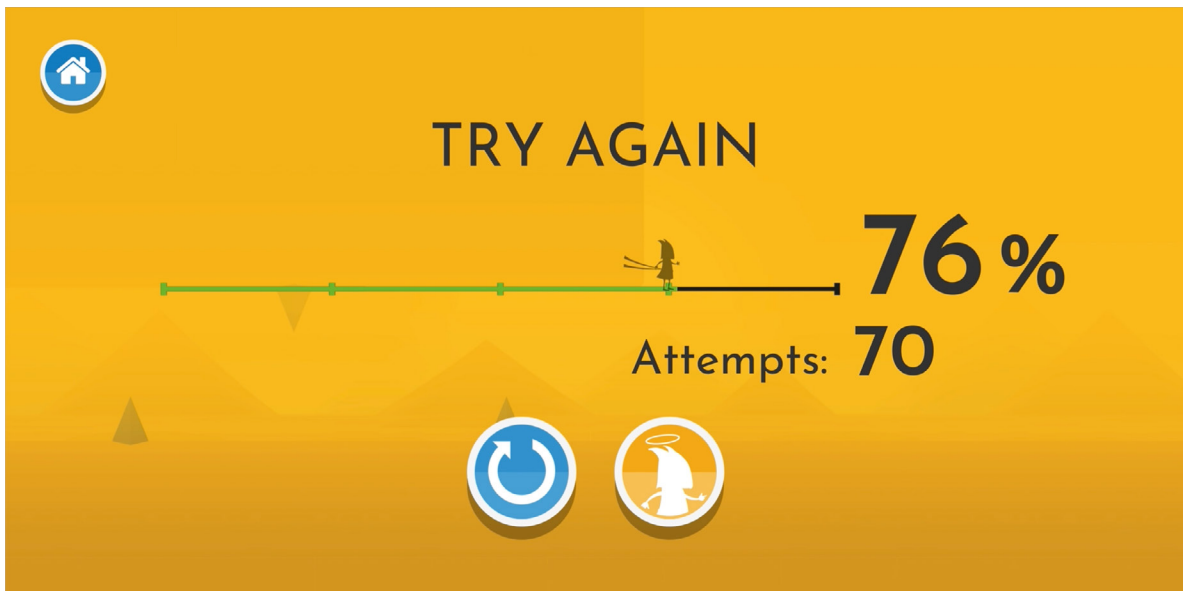
Quan parlem d'elements no interactius en gran part podem referir-nos a elements que afecten a l'aspecte visual del joc, com per exemple els sprites que conformen els fons de tots els nivells. S'han realitzat fons amb diferents capes de profunditat amb un scroll de velocitat variable a cadascuna de les capes, d'aquesta manera s'aconsegueix dotar d'un efecte parallax al conjunt.

Uns altres elements que podem considerar com a no interactius però no menys importants de destacar són els elements motivacionals, destinats a convidar a l'usuari a autosuperar-se, que a Marako's Trip expressem de tres maneres diferents:

- » **Línia de progrés:** És una línia discontinua que apareix a l'última posició on hem mort, perquè durant una partida l'usuari pugui veure que ha superat la seva última marca.
- » **Percentatge de progrés al morir:** Quan fallem en un nivell, a la interfície apareix una barra de progrés on es pot veure de forma percentual el punt concret de completació de nivell on ens trobem.
- » **Comptador d'intents:** Quan fallem en un nivell, a la interfície apareix un comptador d'intents, que ens mostra el número de vegades que s'ha jugat a una pantalla en concret.



En aquesta captura s'aprecia la línia discontinua que marca l'última posició on hem mort



Quan el jugador perd, la interfície mostra el progrés de forma percentual així com el nombre d'intents

5. Monetització del producte

En el mercat d'aplicacions orientades a dispositius mòbils, el plantejament d'oferir-les directament de pagament, com tradicionalment s'ha fet al mercat del software, és cada vegada menys freqüent en els jocs per a dispositius mòbils. El sistema que s'adopta des de fa uns anys en multitud de productes és poder-lo oferir amb modalitat 'Free to Play', és a dir, gratuït a l'hora de descarregar i de jugar, però amb un sistema de integrat que pot generar monetització. A continuació valorem els mètodes més emprats actualment:

- » **Publicitat integrada:** Una de les vies tradicionals de monetització és oferir publicitat integrada dins l'aplicació, que li apareix a l'usuari en els moments que el desenvolupador decideix. Diferents proveïdors de publicitat ofereixen les seves APIs per integrar, que generalment serveixen anuncis relacionats amb altres aplicacions gratuïtes que es poden descarregar des del mateix anunci. Es genera diner real en funció de les visualitzacions que té un anunci i/o en base als usuaris que descarreguen l'aplicació promocionada directament des de l'anunci.
- » **Micropagaments (inapp purchases):** S'ofereixen productes compres integrades dins de l'aplicació a canvi de diner real, com poden ser packs de vides, monedes virtuals, o eliminació de publicitat integrada, entre altres exemples. Es coneixen com a 'Pay2Win' (paga per guanyar) els videojocs que ofereixen un avantatge competitiu important al jugador que compra productes integrats respecte els que no ho fan. En molts casos aquestes pràctiques són criticades si s'implementen de forma abusiva, malgrat que si el joc genera un cert nivell d'addicció, s'ha demostrat que aquest plantejament pot funcionar millor que fent l'aplicació de pagament.
- » **Subscripció:** Un enfocament diferent a les compres integrades són els sistemes de subscripció en els quals l'usuari paga una quota periòdica per a fer ús d'un producte o servei de forma il·limitada mentre el pagament de la quota estigui vigent. Generalment s'apliquen aquests mètodes per desbloquejar funcions d'una aplicació o joc no disponibles a la versió gratuïta, o bé disponibles de forma parcial.

Evaluats aquests tres mètodes, per a Marako's Trip es decideix utilitzar una combinació de publicitat (UnityAds) i compres integrades (natiu iOS i Android), que es detallen a continuació:

- » **Publicitat:** En ocasions, quan l'usuari reinicïi un nivell o avanci una fase li apareixerà un anunci, que podrà saltar passats uns segons de visualització. Un segon cas és l'ús dels vídeos recompensats (rewarded ads). Alguns nivells disposen de punts de control i si Marako mor en un nivell després de passar un d'aquests punts, l'usuari té la opció de ressuscitar-la a l'últim punt de control a canvi de visualitzar un anunci que no es pot saltar, i per tant, té més capacitat de monetització que l'anunci tradicional. A la versió actual de Marako's Trip la publicitat està completament integrada però s'ha desactivat exceptuant a la opció de ressuscitar, a fi de facilitar l'evaluació del producte.
- » **Compres integrades:** Es contempla afegir la opció d'eliminar publicitat integrada, i en un futur oferir alguns items que permetin millorar la puntuació d'un nivell, però que en cap cas puguin donar un avantatge a l'hora de superar un nivell respecte els usuaris que no paguen.

6. Disseny tècnic

6.1 Entorn de desenvolupament escollit

Dispositius de desenvolupament

Per desenvolupar el joc en primer lloc és necessari disposar d'un ordinador amb tot el software necessari instal·lat (Veure apartat 6.3 per un inventari detallat de programari). Necessitem una plataforma que ens permeti compilar tant per iOS com per Android, i per aquest motiu la majoria del desenvolupament es farà utilitzant hardware d'Apple, concretament un Macbook Pro de 2018. No és necessari que aquest hardware sigui molt punter, donat que estem desenvolupant un producte destinat a plataformes mòbils i es cerca que el joc pugui funcionar de forma fluïda en terminals mòbils de gamma baixa-mitjana, per tant haurà de primar la optimització enlloc de la potència bruta.

Dispositius de testeig

Durant el desenvolupament serà necessari compilar constantment per les plataformes d'iOS i Android. Per aquest motiu, es disposen de dos dispositius de testeig que s'enumeren a continuació:

- » **iPad:** Per a provar la versió d'iOS s'utilitza una tauleta iPad enlloc d'un smartphone d'Apple, perquè és primordial poder provar el producte amb una pantalla amb un ràtio 4:3, totalment diferent al que estem acostumats a veure en smartphones.
- » **Xiaomi Redmi Note 6 Pro:** Per el testeig de la versió d'Android s'utilitza un terminal de gamma mitjana de la marca Xiaomi, el qual té un ràtio de pantalla de 19:9

Control de versions

Encara que aquest projecte sigui un treball individual, per tant no col·laboratiu entre diversos membres d'un equip, es fa imprescindible integrar un sistema de control de versions per els següents motius:

- » Poder guardar diferents punts de control per tenir còpies de seguretat, i poder retrocedir si és necessari.
- » Poder mantenir el repositori actualitzat si es treballa amb diferents equips.
- » Poder disposar de dos clons del projecte en el mateix equip, un configurat per compilar amb iOS i l'altre amb Android, i així evitar anar canviant de plataforma.
- » Facilitar als consultors del projecte la descàrrega i avaluació del joc.

S'ha optat per utilitzar un repositori gratuït de bitbucket.org, utilitzant Mercurial per el control de versions, per ser la metodologia coneguda i seguida en projectes anteriors a la realització del Grau de Multimèdia. L'elecció de Mercurial ha acabat convertint-se en una mala opció, ja que Bitbucket dona fi a aquest servei el proper 30 de maig de 2020.

6.2 Requeriments tècnics de l'entorn de desenvolupament

Essent Unity el principal motor escollit per al desenvolupament del joc, per determinar els requeriments de maquinari ens podem cenyir als mínims que demana aquest motor per a la versió 2019, amb dades extretes de la web oficial:

- » **Sistema Operatiu:** Windows 7 SP1+, 8, 10 (només versions de 64 bits), macOS 10.12+
- » **CPU:** Que suporti instruccions SSE2.
- » **GPU:** Targeta amb suport per DX10 (shader model 4.0).

Per altra banda, els dispositius que podran reproduir els jocs necessitaran complir els següents requisits mínims:

- » **iOS:** Qualsevol dispositiu amb iOS 9.0 o superior.
- » **Android:** Versió d'Android 4.1 o superior i suport per OpenGL ES 2.0 o superior.

Al marge de tot això, serà necessari disposar de connexió a Internet, no de forma permanent però sí per activar llicències de programari i pujar canvis al repositori.

En quant al software d'Adobe aquest no és molt precís sobre els requeriments, demanant processadors de múltiples nuclis amb suport per a 64 bits, i una quantitat de memòria RAM variable, 2GB mínim en cas de Photoshop i 8GB en el cas d'Illustrator.

En qualsevol cas, les característiques de l'equip que s'utilitza per el desenvolupament son les següents, i compleixen sobradament l'execució del programari necessari que es detalla en el punt 5.3 d'aquesta memòria:

Model: Apple Macbook Pro 2018

CPU: Intel Core i7 de sis nuclis a 2,6 Ghz.

RAM: 16GB de DDR4 a 2400Mhz

GPU: Radeon Pro 560X amb 4GB de VRAM

6.3 Inventari d'eines emprades

- » **Unity3D:** Motor gràfic multiplataforma utilitzat per el desenvolupament del joc en les seves versions per iOS i Android.
- » **Playmaker:** Plugin de Unity3D que permet desenvolupar en aquest motor utilitzant FSM (Finite State Machines) amb un conjunt d'accions predeterminades de forma visual, amb l'objectiu d'alleugerir al màxim la programació i d'aquesta manera poder centrar els recursos en el disseny.
- » **Gley Easy Achievements and Leaderboards:** Plugin de Unity que permet la integració de marcadors i assoliments amb Playmaker.
- » **Adobe Illustrator:** Programari d'il·lustració vectorial amb el qual s'han generat els sprites que componen el joc.

- » **Adobe InDesign:** Programari de maquetació utilitzat per compondre aquesta memòria.
- » **Adobe Photoshop:** Programari de tractament d'imatges basades en píxels, utilitzat per tractar posteriorment els sprites generats amb Illustrator.
- » **Adobe Audition:** Programari d'edició de so utilitzat per tractar i modificar els efectes sonors utilitzats en el joc.
- » **Adobe Premiere:** Programari d'edició no-lineal de vídeo per editar els vídeos relacionats amb el projecte.
- » **Adobe After Effects:** Programari de composició digital necessari per produir fragments concrets dels vídeos i el tràiler.
- » **Creature Animation:** Programari per a la generació d'animacions bidimensionals basat en malles poligonals, que entre altres funcionalitats permet l'aplicació de dinàmiques i física a algunes regions del model.
- » **Sourcetree:** Gestió de control de versions amb interfície d'usuari, permet utilitzar Git o Mercurial.
- » **Xcode:** Programari propietari d'Apple destinat a compilar executables per iOS i macOS.
- » **Android SDK:** Eines de Google necessàries per a poder compilar executables d'Android.

6.4 Integració d'APIs de tercers

Marako's Trip integra serveis de tercers, en aquest cas Google Play Services (Android) i Game Center (iOS) per a poder oferir un sistema integrat de marcadors i assoliments, que es detallen a continuació:

- » **Marcadors:** Els usuaris accedeixen a un marcador mundial on poden comparar la puntuació obtinguda amb la dels altres jugadors, i veure la seva posició en el rànquing.
- » **Assoliments:** Els assoliments són un element motivacional per als jugadors ja que es defineixen reptes, que una vegada s'assoleixen augmenten la puntuació de jugador, no del propi Marako's Trip, sinó en base a un còmput global en funció dels diferents jocs als que juga un usuari.

És necessari diferenciar en tot moment les APIs a utilitzar en funció de la plataforma amb la qual s'estigui jugant, i per aquest motiu, en funció de si el joc s'està executant a Android o iOS s'utilitzen unes APIs o altres.

6.5 Inventari i descripció d'assets i recursos al joc

Tots els assets relatius a l'apartat gràfic, tant per a l'in-game del joc com la seva interfície, s'han generat amb Adobe Illustrator i posteriorment s'han treballat amb Photoshop amb la finalitat de millorar el seu aspecte visual. Aquests es detallen a continuació:

Marako

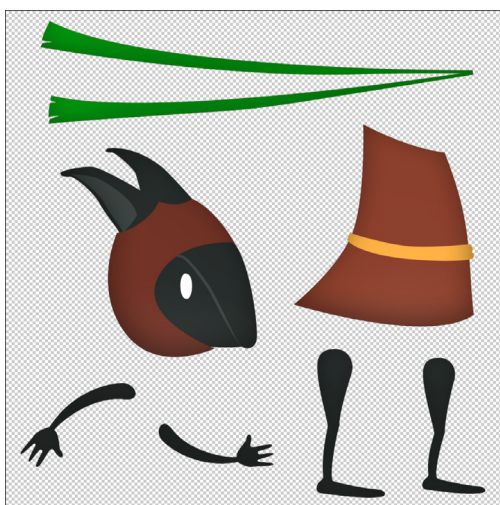
La protagonista del joc s'ha vectoritzat amb Adobe Illustrator utilitzant l'eina ploma, a partir d'un esbós dibuixat a mà alçada amb llapis.



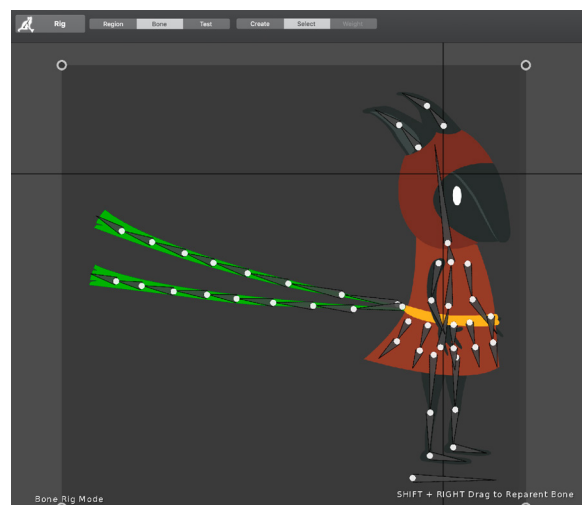
Fig. 9-

Posteriorment, s'han separat les diferents parts del cos de la protagonista per generar un atlas, necessari per a poder composar el personatge amb Creature Animator. L'atlas s'ha treballat amb Photoshop aplicant un efecte d'ombra interior (inner shadow), simulant, salvant les distàncies, oclusió ambiental, efecte àmpliament utilitzat en el món dels gràfics tridimensionals. També s'ha afegit efecte de bisellat (bevel and emboss) aplicat de manera molt subtil per mantenir l'estètica "flat design".

Finalment, s'ha compostat el personatge amb Creature Animator generant un rig per a poder animar el model, aplicant també dinàmiques i física a certs elements com el vestit, les orelles i les cintes.



Atlas de la Marako treballat amb Photoshop

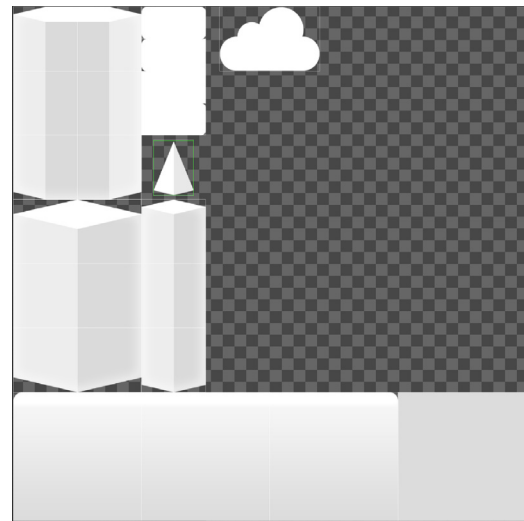


Rig de la Marako creat amb Creature Animator

Sprites d'entorn

Els sprites que conformen els diferents blocs i elements de l'entorn que Marako recorrerà també s'han generat en un sol atlas. Aquesta decisió s'ha pres per motius d'optimització, doncs si tots els elements es troben en una sola textura, malgrat que la fragmentem amb diferents sprites Unity les pot tractar com un sol objecte. D'aquesta manera reduïm la quantitat de draw-calls, és a dir les crides que la CPU fa a la GPU per renderitzar els gràfics.

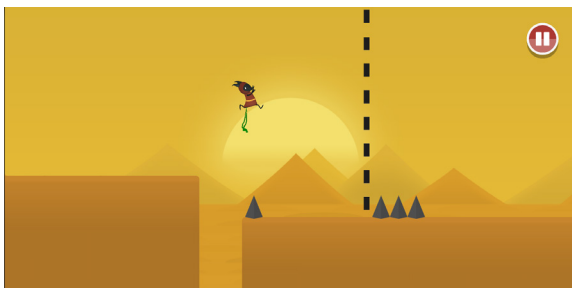
Com es pot veure en el següent atlas, s'ha deixat un espai lliure per a tenir cert marge per a poder introduir nous sprites en futures actualitzacions i a la vegada mantenir el mateix número de draw calls.



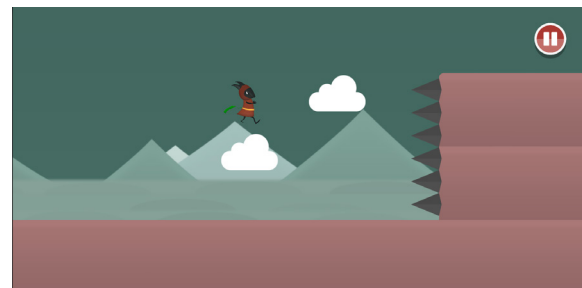
Sprites d'entorn corresponents al món del desert

És important mencionar el detall de què els sprites s'hagin generat amb escala de grisos. Ja que el material que utilitzen els sprites de Unity disposen del paràmetre color, és més ràpid i còmode a nivell de composició de nivells poder modificar el color des del propi motor. Això ens permetrà generar diferents variants cromàtiques dins d'un mateix món per poder oferir varietat visual sense comprometre el pes de l'aplicació final.

A les següents imatges podem observar una aplicació pràctica d'aquest concepte, generant dues variants cromàtiques diferents del mateix món utilitzant un únic sprite atlas.



Entorn del desert amb la gamma cromàtica original



Entorn del desert amb una diferent gamma cromàtica

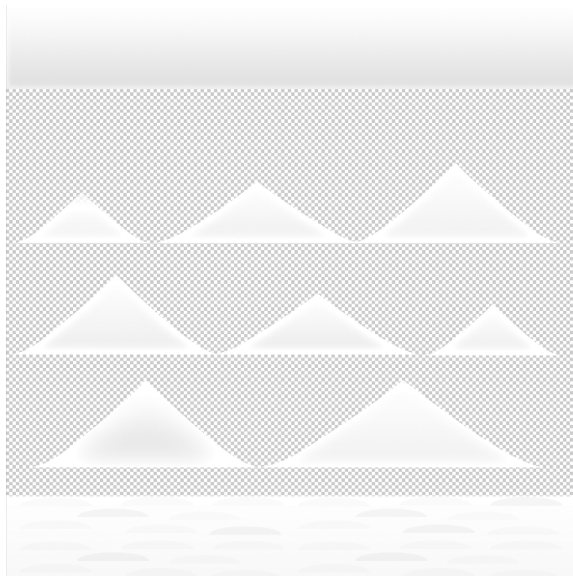
De la mateixa manera que amb l'sprite de la Marako, tots els atlas s'han compost inicialment amb Illustrator en escala de grisos i colors plans, i s'han tractat posteriorment amb Photoshop per aplicar diferents efectes i trencar de forma molt subtil l'aparença inicial completament plana. Els efectes són els següents:

- » **Gradient overlay**, amb opacitats variables en funció de l'sprite, per afegir una lleugera gradació de color vertical a cada sprite.
- » **Inner Glow**, per afegir un subtil halo intern realçant els contorns de les formes.

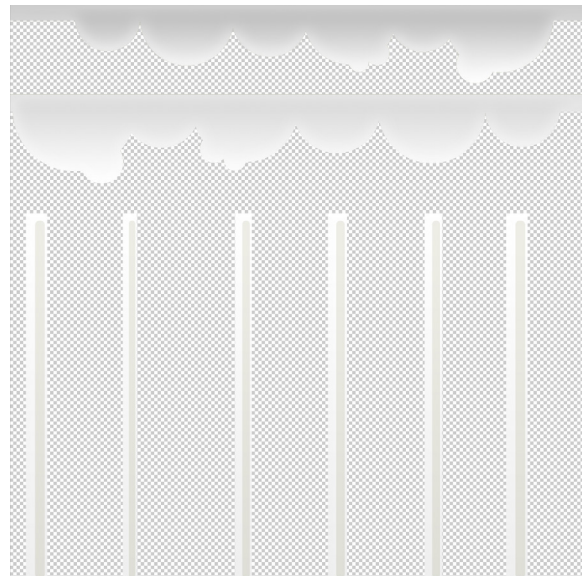
Tots els sprites s'han traslladat de Illustrator a Photoshop mantenint el seu format vectorial, per d'aquesta forma tenir un arxiu psd de pes reduït i escalable en quant a mides. Per comoditat i velocitat, es treballen els sprites en format psd fins i tot dins el projecte de Unity.

Sprites de fons (background)

Els sprites que conformen els diferents fons també s'han generat de la mateixa manera que els anteriors, amb el flux de treball d'Illustrator a Photoshop.



Background del món del desert



Background del món del bosc

Sprites d'interfície (UI)

En quant als sprites relatius a la interfície del joc, en aquest cas també s'han treballat amb Illustrator i el color s'ha definit en aquest programari. S'han copiat i enganxat a Photoshop però, simplement per tenir-los en un format ràpid d'editar i que el Unity el pugui llegir de forma nativa.



Sprites relatius a la interfície del joc

Efectes sonors

L'apartat sonor de Marako's Trip és l'única vessant del joc on els assets s'han hagut d'externalitzar per aconseguir un resultat òptim donat els temps de marge dels quals es disposa en un projecte de Final de Grau. En quant als efectes, aquests s'han obtingut del portal freesound.org utilitzant una cerca de materials amb llicència CC0, per tant es poden utilitzar per a qualsevol ús, incloent el comercial. En alguns casos s'han tractat amb Adobe Audition per realitzar processos com passar de mono a estèreo, modificar el pitch o eliminar espais en blanc a l'inici o al final de la pista de so.

Música

Totes les pistes que es poden escoltar al joc són del mateix autor, Kevin MacLeod, i s'han obtingut del portal incompetech.filmmusic.io. Es disposa de dues opcions diferents per utilitzar aquestes pistes, la primera, amb llicència CC BY permet l'ús sempre que es faci menció de l'autor en un apartat específic, encara que sigui per usos comercials. En el contexte d'un videojoc, aquest apartat

podria ser el de crèdits. La segona opció és pagar 20 € per cada pista, i obtenir una llicència de lliure ús. Per el cas de Marako's Trip s'ha considerat que el preu suposa un cost assumible, i quan s'acosti el moment de publicar el joc a les diferents stores s'adquiriran les llicències necessaries.

6.6 Inventari complet

A continuació es llisten tots els assets utilitzats a la versió actual del joc al complet:

Sprites (es llisten els sprites de forma individual, sense especificar si son part d'un atlas):

Generals

- » **Marako:** Sprite del personatge principal del joc

Desert

- » **SnakeBoss:** Boss del món del desert
- » **MountainLayer1, 2, 3 i 4:** Elements de repetició que conformen els fons del desert
- » **Sun:** Sprite de sol que forma part del fons del desert
- » **BGStatic:** Fons en escala de grisos que s'utilitza a tots els móns canviant el paràmetre color
- » **Sun:** Sprite de sol que forma part del fons del desert
- » **BigBlock** (múltiples variants): Sprites connectables per on Marako pot caminar
- » **DesertPillar** (múltiples variants): Sprites connectables per on Marako pot caminar
- » **JumperCloud:** Núvol que permet a Marako realitzar un salt addicional
- » **MovingPlatform** (múltiples variants): Plataformes connectables mòbils
- » **Sand** (Múltiples variants): Terra connectable que forma part de la superfície del desert
- » **SmallBlock** (múltiples variants): Sprites connectables per on Marako pot caminar
- » **Spike:** Sprite en forma de punxa que mata a la Marako amb el contacte

Bosc

- » **Trees** (múltiples variants): Formen part dels elements de background del bosc
- » **Forest** (múltiples variants): Terra connectable que forma part de la superfície del bosc
- » **JumperBase i JumperSpring:** Junts conformen el "saltador" que llença a Marako cap amunt
- » **TreeBranch** (múltiples variants): Elements connectables que permeten compondre un arbre
- » **TreeTrunk** (múltiples variants): Elements connectables que permeten compondre un arbre
- » **TreeTop** (múltiples variants): Elements connectables que permeten compondre un arbre

Fases aquàtiques

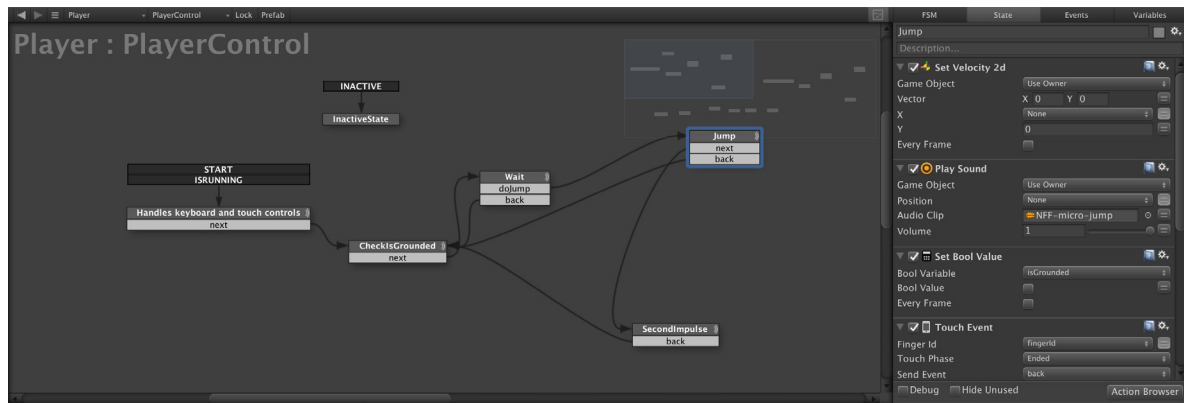
- » **Bubble:** Sprite per els efectes de partícules aquàtics
- » **BackgroundLayer** (múltiples variants): Elements que conformen el fons aquàtic
- » **BackgroundLightRays:** Rajos de llum exterior que conformen el fons aquàtic
- » **WaterPlants:** Plantes aquàtiques que conformen elements de fons
- » **Underwater:** Sprites connectables per on Marako pot nedar
- » **UnderwaterPillar:** Sprites connectables que formen part de l'escenari interactuable
- » **SmallBlock** (múltiples variants): Sprites connectables per on Marako pot nedar

Industrial

- » **IndustrialBlock:** Sprite per on la Marako pot caminar
- » **Spike:** Sprite en forma de punxa que mata a la Marako amb el contacte.
- » **Industrial Sprite:** Sprites connectables per on la Marako pot caminar.
- » **IndustrialBackground0, 1 i 2:** Elements de repetició que conformen el fons industrial.

6.7 Esquema d'arquitectura del joc/components

Per poder maximitzar la productivitat alleugerint les tasques de programació al màxim, i perquè el meu perfil és més de dissenyador que de programador, s'ha escollit l'ús del plugin Playmaker de Hutong Games. Aquesta eina, permet programar de forma visual utilitzant FSM (Finite State Machines), de forma similar al sistema de Blueprints que inclou de manera nativa Unreal Engine.



Malgrat que un programador no recomanaria mai l'ús exhaustiu d'aquests tipus de mètodes per falta d'optimització en alguns casos, i perquè en funció de la complexitat del joc es podria acabar amb un projecte impossible de mantenir per la gran quantitat de FSMs i nodes que es podrien arribar a generar, a nivell de dissenyador aquestes eines són de gran utilitat a l'hora de plasmar comportaments i interaccions amb objectes que només es podrien aconseguir mitjançant programació tradicional.

Havent valorat els avantatges i inconvenients que aporten aquests tipus de mètodes, per la simplicitat a nivell de mecàniques que són necessàries per desenvolupar Marako's Trip, s'ha decidit per el seu ús, ja que amb un interval de temps reduït l'ús del plugin permetria alleugerir i accelerar en gran mesura les tasques de programació, podent centrar els recursos en la creació d'assets gràfics i el disseny de nivells.

Feta aquesta apreciació al respecte de la programació, a continuació passarem a enumerar els aspectes més importants de l'arquitectura del joc i els components que el componen, agrupats en funció de en quina escena es troben.

Escena Main (Pantalla d'inici del joc)

L'escena Main és la primera que es carrega quan inicialitzem el joc, i conté els següents elements:

GameInitializer: Aquest prefab conté 3 FSM amb funcions que són de vital importància per el correcte funcionament del joc. Es detallen a continuació:

- » **SystemInitializer:** En funció de si detecta que el sistema és iOS o Android llança els events d'inici de sessió als serveis de marcadors i assoliments.
- » **LanguageInitializer:** Detecta l'idioma del sistema i l'allotja en una variable, d'aquesta manera els textos de la interfície es podran mostrar en l'idioma que pertorqui, o en el seu defecte l'anglès.

» **ScreenRatioInitializer:** Aquesta funció detecta el ràtio de pantalla per saber si és 4:3 o 16:9 o superior. En funció del que es detecti, durant les fases el paràmetre OrthographicSize de la càmera serà un o altre, a fi de poder mostrar més regió de la pantalla a les fases des d'un dispositiu 4:3.

Canvas: Conté la interfície bàsica d'inici, on podem trobar els següents elements:

» **PlayButton:** Botó de la UI que bàsicament carrega l'escena WorldMap per accedir al mapa del món.

» **LeaderboardButton:** Botó de la UI que carrega els marcadors que pertoquin, si el sistema detectat és Android obre la finestra de marcadors de Google Play Services, mentre que si detecta iOS llança els marcadors de GameCenter.

BackManager: Per poder complir les directives d'Android, aquesta funció controla si es prem el botó Back del dispositiu per poder sortir del joc.



Captura de l'escena Main, nucli principal del joc on s'inicialitzen una sèrie de processos relatius a l'idioma, el ràtio de pantalla i altres serveis

Escena Worldmap

L'escena Worldmap consisteix en un mapamundi des d'on es pot accedir a tots els nivells del joc. Inicialment tots estaran bloquejats excepte el primer, i segons es completa una fase es desbloquegen els nivells que estan connectats a aquesta. D'aquesta manera, el progrés en el joc no es torna completament lineal. A continuació s'exposen els prefabs més rellevants per el seu correcte funcionament:

WorldmapManager: Aquest objecte allotja la imatge del mapamundi i està compost del 3 FSM per el seu funcionament:

» **CameraPositionManager:** Procura que no ens poguem sortir dels limits de la pantalla quan llisquem amb el dit.

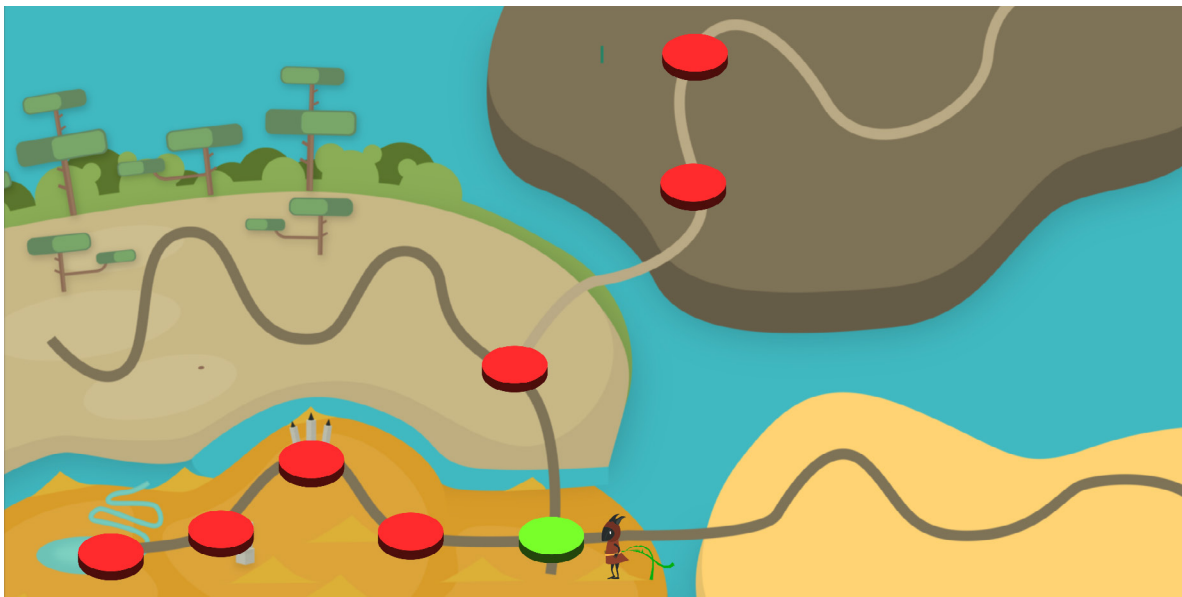
» **TouchControls:** Gestiona el moviment de càmera en funció de quan l'usuari llisca amb el dit sobre el mapa.

» **WorldmapInitializer:** Recupera la variable array amb el progrés del joc i inicialitza el personatge i envia una senyal als botons perquè es desbloquegin en funció de si han d'estar disponibles.

WorldmapButton: Botó que incorpora dos FSM per l'accés als nivells i la gestió del progrés. S'enumeren a continuació:

» **ButtonBehaviour:** Controla el funcionament del botó i que accedeixi al nivell que pertoca quan l'usuari el toca.

» **CheckLevelAvailability:** Controla que el botó es pugui desbloquejar en funció del progrés del joc consultant en un array desat a les PlayerPrefs.



Escena Worldmap, des d'on es pot accedir als nivells que estiguin desbloquejats

Escena de nivell

En una escena d'un nivell de Unity podrem distingir sempre els següents prefabs, indispensables perquè la pantalla i les mecàniques de joc funcionin correctament:

Player: Aquest objecte és el personatge Marako. És el jugador principal i consisteix en un objecte compost per un Rigidbody 2D, amb restricció aplicada a la rotació de l'eix Z. Porta inclòs un Collider 2D de tipus Càpsula per detectar les colisions amb l'entorn, i 6 FSMs de Playmaker que s'enumeren a continuació:

» **GroundCheck:** Aquest FSM s'encarrega de llençar un Linecast en 2D des de la posició de la Marako amb direcció cap al terra, per detectar si està en contacte amb el terra o no i actualitzar la variable de tipus booleana 'isGrounded'.

» **PlayerMovement:** S'encarrega d'executar un moviment de translació positiu a l'eix de la X determinat per la variable global 'Speed'.

» **SpeedController:** Determina constantment la velocitat vertical del personatge, per transferir aquests paràmetres a l'Animator Controller del personatge i així determinar si s'està el-

evant o caient a fi d'executar l'animació que pertoqui.

» **PlayerControl:** És el FSM més complex del personatge i gestiona tot l'esquema de control mitjançant teclat o pantalla tàctil del dispositiu. Hi han centralitzats els controls de córrer, nedar o volar, i s'alterna entre un mode de control o un altre mitjançant transicions globals de Playmaker.

» **CollisionControl:** S'encarrega de gestionar les col·lisions del personatge amb els objectes que el poden eliminar, activant la interfície que pertoqui. Per detectar aquesta col·lisió els objectes tenen assignat un tag 'Enemy'.

» **PlayerScoreCalculator:** S'encarrega de calcular la puntuació obtinguda al finalitzar un nivell en curs, així com registrar-ho a les PlayerPrefs, tant per guardar la puntuació obtinguda com el progrés en el joc.

Main Camera: És la càmera principal que s'encarrega de seguir al personatge. Incorpora tres objectes amb collider i el tag 'Enemy' que eliminen la Marako si sobresurt per el límit superior de la pantalla (volant) el límit inferior (caient al buit) o a l'esquerra del personatge (xocant contra un obstacle). Només té el següent FSM aplicat:

» **CameraMovement:** S'encarrega d'executar un moviment de translació positiu a l'eix de la X determinat per la variable global 'Speed'.

Background: El següent prefab conté totes les capes que conformen els sprites de fons de l'escenari. Cadascuna d'aquestes capes porta el següent FSM:

» **Scroller:** Aquest FSM s'encarrega del desplaçament d'una capa de background cap a l'esquerra (-X). Per fer-ho, multipliquem el valor de la variable Speed per un número negatiu entre 0 i 1, i a cada capa li donem un valor diferent per aconseguir l'efecte Parallax. Una vegada la capa surt del límit esquerre de la pantalla reiniciem la seva posició al límit dret perquè es vagi reproduïnt de forma cíclica.

GameManager: Aquest prefab s'encarrega de la gestió general del nivell i realitza diferents tasques d'inicialització de variable, així com situar una marca visible en l'últim punt on hem mort de la pantalla en curs. Passem a detallar el seu comportament amb els FSM que el componen:

» **GameInitializer:** S'encarrega de resetejar totes les variables que puguin estar actives o amb valors alterats cada vegada que s'inicia un nivell. També detecta objectes rellevants en el nivell i els allotja en diferents variables per a la seva gestió, i perquè altres FSMs d'altres objectes puguin accedir-hi. S'encarrega també de detectar el prefab Background i anidar-lo dins del prefab MainCamera, per assegurar que el fons sigui sempre visible.

» **Backmanager:** Simplement controla si es toca el botó Back en els terminals Android per retrocedir i així complir amb les directives d'aplicacions Android.

» **DeathMarkLocate:** Aquest FSM detecta la última posició on hem mort en un nivell recuperant-la de les PlayerPrefs i situa on objecte consistent en una marca discontinua perquè el jugador sàpiga en quin punt s'ha quedat a l'última partida que ha fet.

MusicManager: Aquest prefab no conté cap FSM, simplement porta el component AudioSource

amb un asset d'àudio amb la música del nivell, amb els paràmetres Play on Awake i Loop.

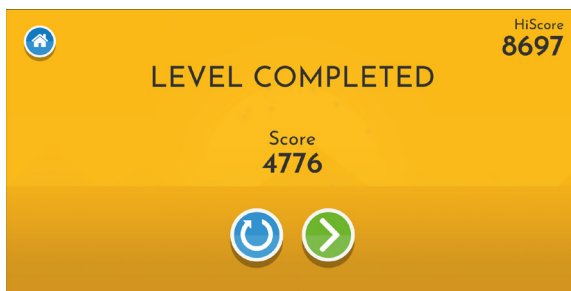
AdManager: Aquest prefab és necessari per el correcte funcionament de la gestió de la publicitat integrada. Ho fa gràcies al següent FSM:

- » **AdTimerManager:** Controla el temps transcorregut jugant a un nivell i activa una variable booleana per determinar si es pot reproduir un anunci.

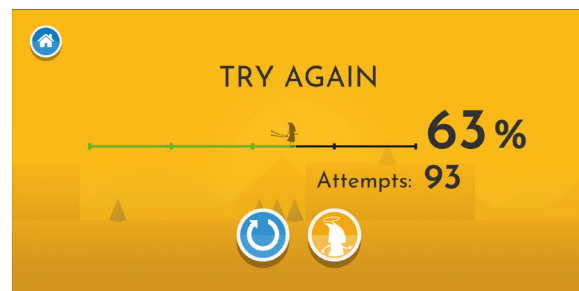
LevelEndPoint: És el portal de sortida d'un nivell, i permet finalitzar-lo quan la Marako hi entra en contacte. Està compost de un FSM que detecta la col·lisió amb Marako i canvia la seva posició "absorbint-la" i fent aparèixer la interfície de nivell completat.

ingameCanvas: Aquest prefab conté tots els elements relatius a la interfície, i està compost de varis objectes fills d'aquest, que enumerem a continuació:

- » **Timer:** És un comptador progressiu que s'activa després de sortir d'una pausa o de reviure, a fi de donar marge al jugador per situar-se abans de tornar a passar a l'acció.
- » **LevelCompletedPanel:** Aquest panell s'activa quan es completa un nivell, i conté informació com la puntuació actual, la màxima obtinguda, els botons de reiniciar el nivell o accedir al següent.
- » **LevelFailedPanel:** Aquest panell s'activa quan morim, i mostra informació com el número d'intents del nivell, i l'objecte LevelProgressionBar, que calcula el percentatge de progressió del nivell perquè el jugador pugui saber en quin punt de la pantalla s'ha quedat. També habilita el botó de reinici del nivell en curs.
- » **LevelFailedRevivePanel:** Similar a l'anterior, però s'activa en cas de no superar un nivell havent passat prèviament per un punt de control. D'aquesta manera, s'habilita el botó de reviure, per poder ressuscitar a canvi de què l'usuari visualitzi un vídeo recompensat.



Panell de nivell completat



Panell de nivell no completat amb opció de ressuscitar

- » **LevelPercentage:** S'activa conjuntament amb qualsevol dels dos panells anteriors i mostra el percentatge de progressió del nivell en curs amb una animació incremental.
- » **LevelPercentageSymbol:** Simplement mostra el símbol % al costat del percentatge numèric.
- » **BackButton:** És el botó per tornar al mapamundi, s'activa en determinats contextes, com per exemple quan pausem el joc.
- » **PauseButton:** Botó que activa el menú de pausa.

- » **PausePanel:** Conté el menú de pausa, amb botó per tornar al mapamundi, reiniciar el nivell o sortir de la pausa.
- » **DangerText:** S'activa només a les fases on hi ha un 'Final Boss' i mostra un text de perill per avisar que en breu apareixerà l'enemic, en homenatge a jocs arcade clàssics.
- » **BossHealthBar:** De la mateixa manera que l'objecte DangerText, només s'activa en els nivells on hi ha 'Final Boss' i mostra amb una animació d'escalat horitzontal la barra de vida de l'enemic.
- » **LoadingScreen:** Mostra l'animació d'entrada de la pantalla de càrrega.

7. Valoració econòmica del treball

Havent fet prèviament un inventari de tot el software utilitzat per el desenvolupament del treball a l'apartat 5.3, a continuació detallem el cost total del projecte, tant per el propi software de pagament, com per serveis addicionals com poden ser les llicències de desenvolupador per publicar a l'iOS App Store i a Google Play.

Tantmateix en el cas del programari Adobe, per el desenvolupament s'han utilitzat les llicències subministrades per la UOC. Per aquest motiu es calcularàn dos pressupostos, on un d'ells tindrà en compte el cost que tindrien en cas de que s'haguéssin d'adquirir.

Tenint en compte que podem considerar Marako's Trip com un producte complet en un context de Treball de Fi de Grau, per poder-lo publicar a l'App Store i a Google Play seria necessari desenvolupar més contingut perquè el producte tingui una duració acceptable. Feta aquesta apreciació, es contempla aproximadament un any de desenvolupament tenint en compte que no serà possible dedicar al desenvolupament el 100% d'una jornada laboral.

Software	Comentaris	Cost
Unity3D	Gratuit en cas de no superar 100.000\$ anuals en guanys	0 €
Playmaker	Pagament únic, llicència per un sol lloc de treball	58,07 €
Adobe Creative Cloud	60,49 € mensuals per el paquet complet. Es contempla una llicència anual	725,88 €
Creature Animation	Pagament únic, llicència per un sol lloc de treball	99 €
Llicència Google Play	Pagament únic vitalici de 25\$	22,37 €
Llicència Apple Developer	Cost anual	99 €

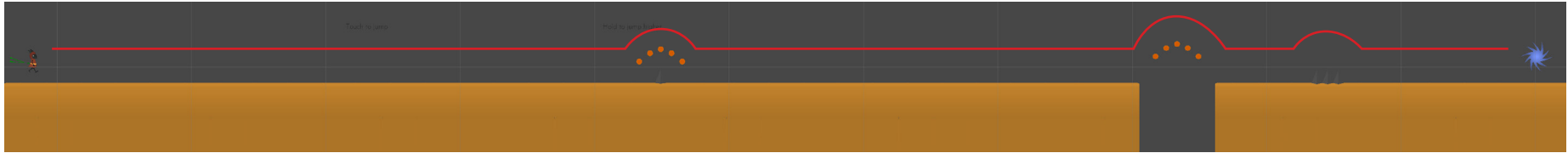
Total: 1004,32 € (Incloent el cost del programari Adobe).

Total real: 278,44 €

És important comentar que aquest pressupost només comprèn el cost de les llicències de programari i de distribució, i no s'està contemplant el cost per hora del rol de programador/dissenyador.

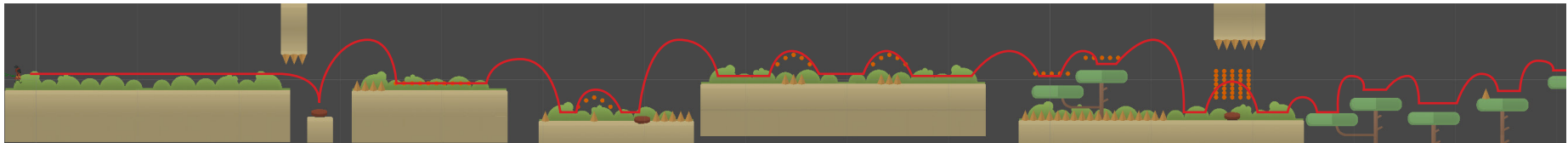
8. Disseny de nivells

A continuació s'exposen exemples d'alguns nivells per poder extreure'n les decisions a nivell de disseny més rellevants. Les fases analitzades son, en aquest ordre: tutorial inicial, nivell ambientat al bosc amb jetpack i nivell industrial on s'inverteix la gravetat.



Secció del tutorial inicial. Hem de saltar un primer obstacle consistent en una punxa.

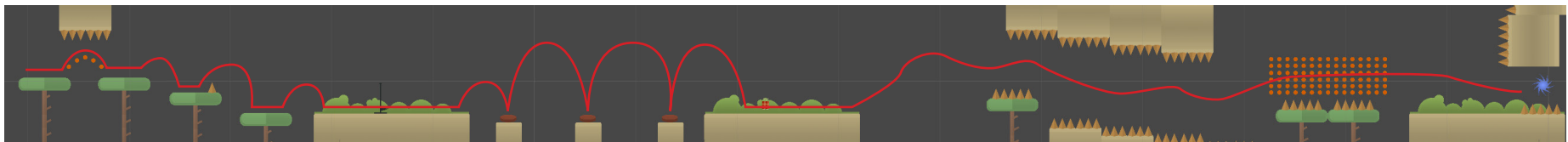
Hem de saltar el barranc impulsant-nos el màxim possible i seguidament saltar tres punxes.



Trobem una primera secció on l'usuari descobreix el 'jumper'. Es força a caure sobre el jumper degut a que les punxes no ens permeten saltar.

Secció de plataformes on utilitzem un altre jumper per saltar a la plataforma superior, seguint les monedes que ens marquen la posició de les punxes.

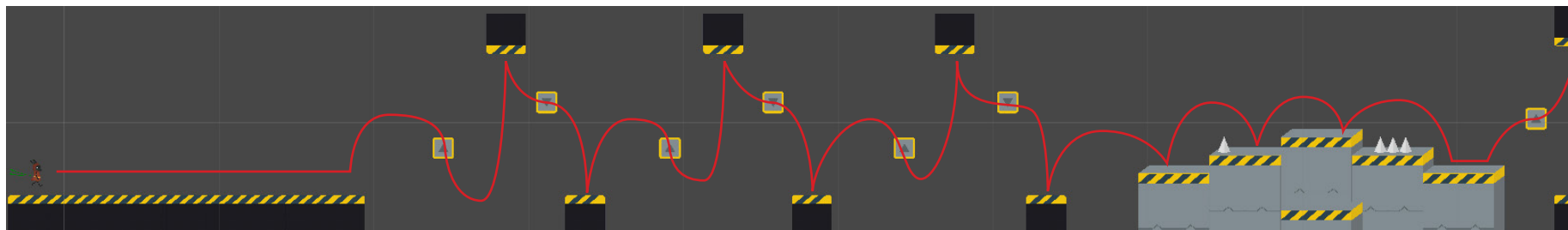
Tercera secció on el jumper actua a mode de trampa, impulsant-nos contra les punxes. Per tant, l'hem de saltar i continuar caminant sobre els arbres.



Hem d'evitar saltar al màxim en aquest tram per no tocar les punxes.

Continuem amb els jumpers fins arribar a la següent plataforma, on hem d'agafar el jetpack.

A partir d'aquest tram hem de volar sense xocar amb cap obstacle, recollint el màxim de monedes possible fins arribar al final de la fase.



Patró de repetició per invertir la gravetat i acostumar a l'usuari a les mecàniques jugables

Secció asimètrica de plataformes amb obstacles



Secció on caminem per el sostre. Ens hem de deixar caure a l'inversor de gravetat per continuar el nostre recorregut per el terra

Arribada al punt de control. Continuar a través del jumper és una trampa que ens llença a les punxes

Última secció del nivell, on hem de saltar des de sota la plataforma cap a l'inversor de gravetat

9. Manual d'usuari

9.1 Requisits tècnics:

iOS: Qualsevol dispositiu iOS amb la versió 9.0 o superior.

iPadOS: Qualsevol dispositiu amb iPadOS.

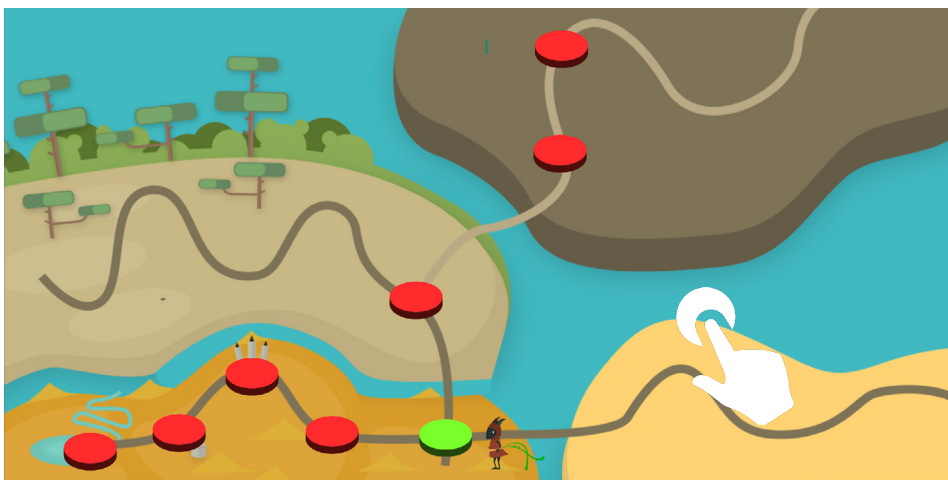
Android: Qualsevol dispositiu Android amb la versió 4.1 o superior.

9.2 Com jugar

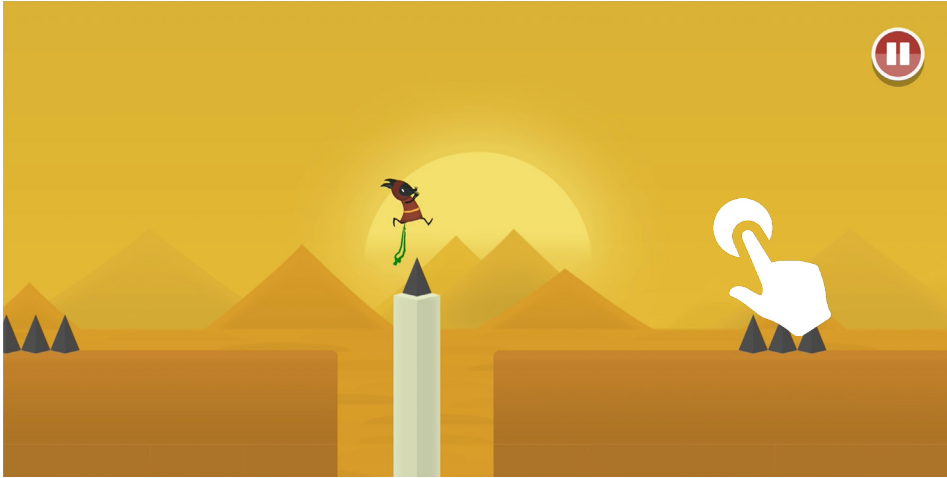
Des de la pantalla principal podem accedir al mapamundi o bé als marcadors per poder veure la puntuació global



Una vegada al mapamundi, ens podem moure lliscant amb el dit sobre la pantalla o bé tocar un botó per accedir a un nivell (en cas de que l'haguem desbloquejat prèviament).



Als nivells on la Marako corre podem tocar a qualsevol punt de la pantalla perquè aquesta realitzi un salt.



Als nivells on la Marako neda podem tocar a qualsevol punt de la pantalla per nedar cap a la superfície.



Als nivells on la Marako vola amb el jetpack podem mantenir el dit a qualsevol punt de la pantalla per volar més amunt.



10. Conclusions

Per a aquest treball de Fi de Grau, l'objectiu proposat ha estat en tot moment el de realitzar totes les fases principals de desenvolupament d'un videojoc, utilitzant algunes de les eines dins d'una gran varietat que tenim a la nostra disposició. Desenvolupar Marako's Trip ha estat un autèntic repte malgrat tenir experiència prèvia amb el desenvolupament de jocs amb Unity, principalment per treballar amb uns marges de temps molt ajustats. Aquest projecte també m'ha permès repassar i aprofundir els coneixements amb Unity, Playmaker i la Suite d'Adobe.

Des d'un primer moment ha estat clau marcar un objectiu realista, malgrat que no considero que tots els objectius s'hagin complert plenament. Quedaria pendent desenvolupar més nivells i integrar nous móns i mecàniques jugables, així com refinar la interfície i sobretot, la interacció amb el mapamundi. També s'ha quedat a mig camí l'apartat de la monetització, i malgrat estar la publicitat integrada, en un moment donat s'hauran de desenvolupar i integrar les compres dins de l'aplicació. Queda també a mitges les traduccions, i per ara el joc es troba disponible en Català, Castellà i Anglès.

En general la planificació i el resultat han estat bastant fidels, malgrat que durant el desenvolupament han anat sorgint noves idees que no s'han pogut implementar per falta de temps i poden millorar el resultat final de cara a un possible llançament real. Per altra banda, no s'ha pogut realitzar un test formal amb usuaris, malgrat que unes deu persones l'han pogut provar en un entorn informal donant el seu feedback sobre punts crítics relatius a la dificultat. Quedarà pendent doncs, poder realitzar un test combinat amb la implementació de Unity Analytics, per detectar si algun nivell té algun pic de dificultat que pugui alimentar la frustració.

Amb tot això, em sento completament satisfet amb el resultat obtingut per ser un producte propi que intenta fer un mínim ús d'assets generats per tercers, amb un aspecte visual molt correcte i minimalista, tant per les fases com per la interfície. Des del principi he tingut molt clar que Marako's Trip no es podia quedar com a Projecte de Fi de Grau, i en un moment donat s'haurà de publicar a Google Play i l'App Store, una vegada pugui desenvolupar més contingut amb paciència, dedicació, i donant-li al projecte el temps que es mereix.

11. Glossari

- » **AAB:** (Android App Bundle) Format per a la publicació d'aplicacions a Google Play, que conté diverses versions d'una aplicació o joc per a múltiples arquitectures, per exemple 32 i 64 bits.
- » **After Effects:** Programari d'Adobe per a postproducció de vídeo i composició digital.
- » **Android:** Sistema operatiu per a dispositius mòbils desenvolupat per Google Inc.
- » **API:** (Application Program Interface), consisteixen en una sèrie de funcions i rutines que permeten connectar aplicacions, generalment s'utilitzen per integrar serveis proveïts per externs a una aplicació.
- » **APK:** (Android Package) Format d'instal·lador executable per a dispositius Android.
- » **App Store:** Botiga d'aplicacions d'Apple per els sistemes iOS, iPadOS o MacOS.
- » **Array:** Matriu que pot emmagatzemar diversos tipus de dades.
- » **Collider:** Component que envolta un objecte i li ofereix diferents maneres de detectar una col·lisió amb altres objectes.
- » **Free to Play:** Concepte amb el que es coneixen els jocs en els quals la descàrrega és inicialment gratuïta, però que integren algun sistema de monetització, sigui a través de publicitat o compres integrades a l'aplicació.
- » **Google Play:** Botiga d'aplicacions de Google per als sistemes Android.
- » **Illustrator:** Programari d'Adobe per a la creació de gràfics vectorials.
- » **iOS/iPadOS:** Sistema operatiu desenvolupat per Apple per a la seva gamma de dispositius mòbils.
- » **Pay2Win:** Concepte amb el qual es defineix de forma col·loquial els jocs que ofereixen un avantatge competitiu important als usuaris que realitzen compres integrades respecte els que no ho fan.
- » **PlayerPrefs:** Arxiu on s'emmagatzemen diferents dades relatives d'un joc de Unity, principalment dades de progrés i configuracions.
- » **Photoshop:** Programari d'Adobe per a la gestió d'imatges rasteritzades.
- » **Premiere:** Programari d'Adobe per a l'edició no lineal de vídeo.
- » **Runner:** Subgènere amb el qual es coneixen els jocs on el protagonista es desplaça de forma automàtica i el jugador s'encarrega de realitzar certes accions concretes.
- » **TFG:** Abreviació de Treball de Fi de Grau.
- » **Unity3D:** Motor gràfic àmpliament extès en el desenvolupament de videojocs.
- » **Playmaker:** Plugin de Unity que permet utilitzar màquines d'estats de forma visual.

12. Bibliografia

<https://unity.com> - 1 de setembre de 2019

<https://hutonggames.com> - 1 d'agost de 2019

<https://creature.kestrelmoon.com/> - 5 de gener de 2020

<https://play.google.com/apps/publish/> - 4 de gener de 2020

<https://itunesconnect.apple.com> - 4 de gener de 2020

<https://freesound.org> - 4 de gener de 2020

<https://incompetech.filmmusic.io> - 4 de gener de 2020

<https://developer.android.com/studio> - 1 de setembre de 2019

<https://unrealengine.com> - 23 de desembre de 2019

<https://halfbrick.com/our-games/jetpack-joyride/> - 10 de setembre de 2019

<https://thatgamecompany.com/journey/> - 10 de setembre de 2019

<https://apps.apple.com/gb/app/vvvvvv/id880645949> - 10 de setembre de 2019

<https://play.google.com/store/apps> - 10 de setembre de 2019