

Universitat Politècnica de Catalunya

FACULTAT D'INFORMÀTICA DE BARCELONA

Grau en Enginyeria Informàtica

CONFECCIÓ D'UN JOC 3D:
MARIO ROLLING SKY

Memòria del projecte

Arnaud Arqué i Daniel Esquina

Professor: Jesus Alonso Alonso

19 de desembre de 2019

Índex

1	Introducció	3
2	Joc de referència	4
3	Descripció del joc	5
3.1	Objectius	5
3.2	Instruccions	5
3.3	Flux del joc	7
3.4	Tiles	8
3.5	Entitats del joc	11
4	Metodologia	25
4.1	Programari	25
4.2	Planificació	25
5	Conclusions i valoració personal	28
6	Bibliografia i webgrafia	30
6.1	Pàgines web	30
6.2	Vídeos	30

Índex de figures

3.1	<i>Flagpole</i> del joc situada al final del nivell 1.	5
3.2	Recompte de monedes als menús.	6
3.3	Pantalla de selecció dels controls.	6
3.4	Menú principal.	7
3.5	Menú d'inici del nivell 1.	7
3.6	<i>In-game</i> nivell 2.	8
3.7	Menú de pausa.	8
3.8	Menú de mort.	9
3.9	Tiles bàsics.	9
3.10	Jumping tiles.	10
3.11	Falling tiles.	10
3.12	Forward i Boost tiles.	11
3.13	Cubs bàsics bisellats.	12
3.14	Confecció del mystery block.	12
3.15	Mystery block in-game.	12
3.16	Bricks individuals.	13
3.17	Diferents situacions del brick block.	13
3.18	Objectes de Pipe.	14
3.19	Pipes aixecades a l'inici i, al fons, amagades.	14

3.20	Animació bouncing del hammer.	15
3.21	Sistema de partícules del bullet-bill.	15
3.22	Confecció del bullet-bill i bill-blaster mitjançant Blender.	16
3.23	Animació del bill-blaster expulsant un bullet-bill.	16
3.24	Animació de la piranha.	17
3.25	Estats del chomp.	18
3.26	Animacions del chomp a Blender.	18
3.27	Confecció del thowmp a Blender.	19
3.28	Esquelet del koopa-troopa i el blue shell.	19
3.29	Blue shell en partida.	20
3.30	Moviment i partícules de la moneda.	20
3.31	Esquelet del Bowser.	21
3.32	Calibració de pesos de l'esquelet.	21
3.33	Vista <i>top</i> de l' <i>inverse kinematics</i> del braç del Bowser.	22
3.34	<i>Side view</i> de l' <i>inverse kinematics</i> de les cames del Bowser.	23
3.35	Diagrama de flux de l'animació del boss.	23
3.36	Animacions del boss (Bowser).	24
4.1	Captura de pantalla del repositori de Github.	26
4.2	Captura de pantalla del <i>sourcetree</i> .	26
4.3	Tasques establertes.	27

Índex de taules

2.1	Taula-resum de les especificacions del joc Rolling Sky.	4
2.2	Taula-resum del joc Rolling Sky en diferents plataformes.	4

1 Introducció

El present projecte sorgeix com a part integrant de l'assignatura de Videojocs del Grau en Enginyeria Informàtica cursat a la Facultat d'Informàtica de Barcelona (Universitat Politècnica de Catalunya). Aquesta iniciativa exemplifica la convergència entre la innovació tecnològica i la creativitat artística, reflectint els avenços de la indústria dels videojocs i les capacitats dels estudiants moderns.

En aquest document, expliquem amb detall la concepció i execució d'un emocionant joc 3D, que serveix com a tribut al conegut joc Rolling Sky. Aquest projecte encaixa perfectament en el marc educatiu d'aquesta assignatura, proporcionant una plataforma per explorar les intricades connexions entre programació, disseny de jocs i experiència d'usuari.

Aprofundirem en les característiques centrals del nostre joc tribut, que s'inspira en els inoblidables personatges i enemics de la saga Mario de Nintendo. Mitjançant l'ús de Unity, hem creat tres nivells únics de Rolling Sky que incorporen aquests elements icònics, tot donant llibertat a la nostra creativitat per reinventar les mecàniques de joc i sorprendre els jugadors.

Analitzarem amb profunditat la dinàmica del gameplay, els desafiaments i els girs únics que hem incorporat per mantenir l'emoció i l'interès dels jugadors en cada nivell. A més, proporcionarem instruccions detallades sobre com gaudir plenament d'aquesta experiència de joc i superar els reptes que presentem.

Una part crucial d'aquest document és la descripció detallada del procés de desenvolupament. Explorarem les etapes que hem seguit per donar vida al nostre joc, des de la concepció inicial fins a la seva implementació i refinament. Això proporcionarà una visió completa de la dedicació i el treball en equip que hem invertit en aquest projecte.

Per últim, compartirem la metodologia que hem adoptat per abordar aquest desafiament i les conclusions que hem extret de la nostra experiència. Aquestes reflexions destilen les lliçons apreses i els èxits assolits, així com les oportunitats futures per a la millora i la continuació d'aquest emocionant viatge de creació de jocs.

2 Joc de referència

El joc de referència és Rolling Sky de la desenvolupadora Cheetah Mobile. La distribuidora és Cheetah Mobile juntament amb Rising Win Tech i TurboChili. Està disponible a les plataformes Android, iOS i Switch. Té una qualificació PEGI de 3/4+ i consta de més de cent milions de descàrregues i més d'un milió d'opinions amb una nota mitjana de 4.2/5. Podeu veure un resum de les especificacions enunciades a les taules 2.1 i 2.2.

Taula 2.1: Taula-resum de les especificacions del joc Rolling Sky.

Concepte	Especificació
Desenvolupador	Cheetah Mobile
Distribuidor	Cheetah Mobile, Rising Win Tech i TurboChili
Plataformes	Android, iOS i Switch
PEGI	3/4+
Descàrregues	+100 milions
Opinions	+1 milió
Nota mitja	4.2/5
Web oficial	https://cheetahgames.cmc.com/RollingSky
Tràiler	https://youtu.be/4LLTqh0HtzQ
Gameplay	https://youtu.be/fdgbYMX9wvA
Entrada wiki	https://rolling-sky.fandom.com/wiki/Rolling_Sky

Font: Elaboració pròpia

Taula 2.2: Taula-resum del joc Rolling Sky en diferents plataformes.

Concepte	Google PlayStore	AppleStore	Nintendo e-Shop
Data de llançament	26/01/2016	20/01/2016	12/04/2019
Última versió	3.3.8.1	2.3.7	–
Preu (Eur.)	Gratuït	Gratuït	9.99
Mida (MB)	73	208.7	288

Font: Elaboració pròpia

3 Descripció del joc

En aquest joc hem volgut homenatjar no només al Rolling Sky proposat, sinó a la mítica saga de Mario. Totes les entitats del joc i disseny de nivells estan fets pensant en la jugabilitat de jocs com *Mario Kart* o *New Super Mario Bros*. D'aquesta manera hem aconseguit adaptar dos jocs d'estils molt diferents en el resultat del nostre projecte.

En aquest joc es poden jugar a tres nivells diferents. La temàtica dels nivells està basada en l'estructura clàssica de jocs de *New Super Mario Bros*, on hi ha nivells de prat, de cel i al final un boss. Els nivells del joc tenen una dificultat incremental. Tots tres tenen noms de circuits de *Mario Kart* i els temes escollits per aquests són els que més s'assemblen al ritme actiu de la música de Rolling Sky. Altres funcionalitats com els menús també han estat pensats tenint en compte l'estil dels jocs de la saga. El menú principal té una clara estètica de *Mario Kart* i el menú de mort està basat en el de *Mario 64*. Els menús d'start i pausa s'assemblen molt més al joc de Rolling Sky original tot i que s'han adaptat per mantenir una coherència visual.

3.1 Objectius

L'objectiu principal dels nivells és arribar al final, on hi ha una asta de bandera, típica dels jocs de Mario (Figura 3.1).

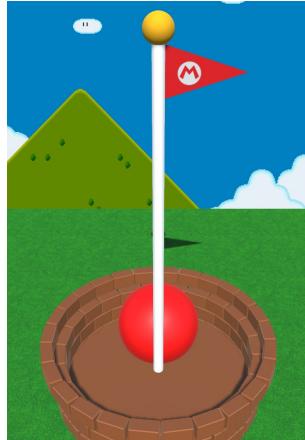


Figura 3.1: *Flagpole* del joc situada al final del nivell 1.

Igual que les gemmes del Rolling Sky original, hem utilitzat les monedes posicionades a llocs estratègics per incentivar camins no habituals i com un objectiu addicional del joc per afavorir la rejugabilitat de nivells. A continuació podem veure el comptador de les màximes monedes aconseguides al nivell, que es mostra tant al menú d'start (Figura 3.2a) de cada nivell com al menú principal (Figura 3.2b). Al menú de pausa, en canvi es mostren les monedes que has aconseguit actualment.

3.2 Instruccions

Hi ha dues maneres de controlar el moviment lateral de la pilota: amb les fletxes dreta/esquerra del teclat i arrossegant la pilota amb el ratolí. Aquestes dues possibilitats s'escullen a l'inici de



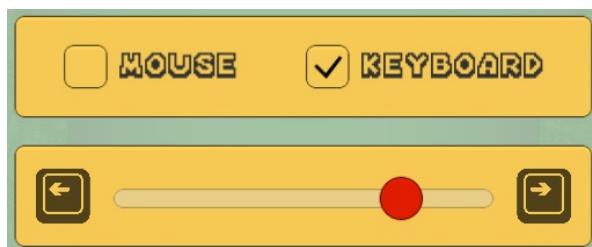
(a) Monedes al menú start.



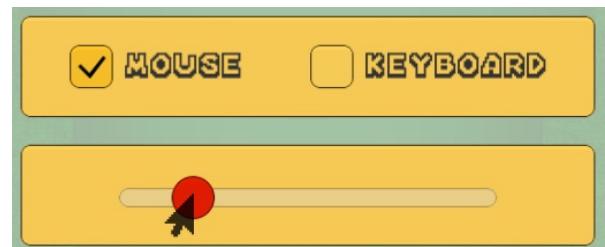
(b) Monedes del nivell 1 al menú principal.

Figura 3.2: Recompte de monedes als menús.

cada nivell i es mostra el respectiu funcionament com podem veure a la Figura 3.3.



(a) Opció de control amb *keyboard*.



(b) Opció de control amb *mouse*.

Figura 3.3: Pantalla de selecció dels controls.

Totes les interaccions del joc per transicionar entre menús es fan amb el ratolí. Aquestes transicions inclouen la selecció de nivells, començar, aturar i sortir del joc. Per fer més fàcil provar i testar el funcionament del joc s'han afegit dreceres que permeten fer diferents funcionalitats no destinades al públic del joc. Podeu veure un resum de les comandes a continuació:

- c Desactiva les col·lisions de la pilota.
- m Atura el moviment de la pilota, càmara i background.
- ↑ Puja la pilota una quantitat en l'eix y.
- ↓ Baixa la pilota a la seva altura predeterminada de l'eix y.
- 1 Posiciona la pilota, càmara i background en la posició 1/5 del mapa de l'eix z.[a]
- 2 Posiciona la pilota, càmara i background en la posició 2/5 del mapa de l'eix z.
- 3 Posiciona la pilota, càmara i background en la posició 3/5 del mapa de l'eix z.
- 4 Posiciona la pilota, càmara i background en la posició 4/5 del mapa de l'eix z.
- 5 Posiciona la pilota, càmara i background en la posició 5/5 del mapa de l'eix z.

3.3 Flux del joc

A la Figura 3.4 podeu veure la primera pantalla que és mostra en executar l'aplicació. Ens permet escollir un dels tres nivells o sortir. En seleccionar un d'aquests nivells se'n redirigeix al seu menú d'start del nivell corresponent (Figura 3.5).

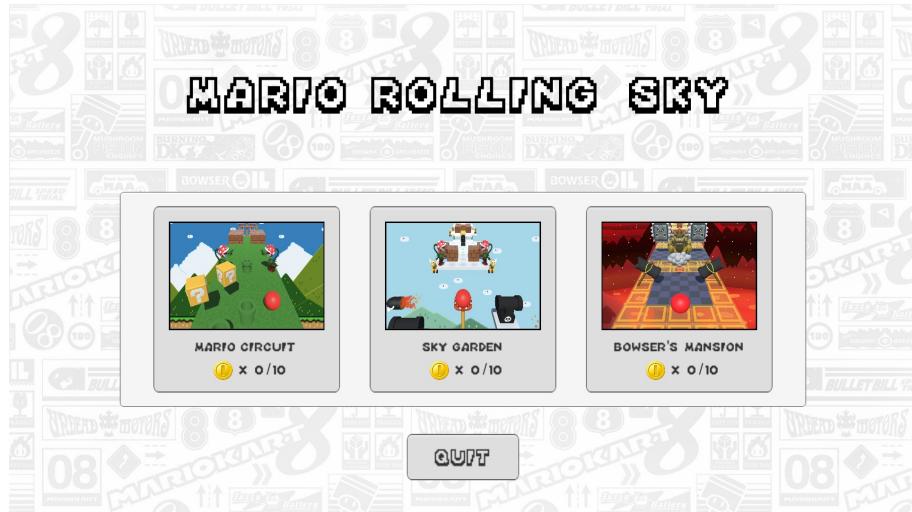


Figura 3.4: Menú principal.

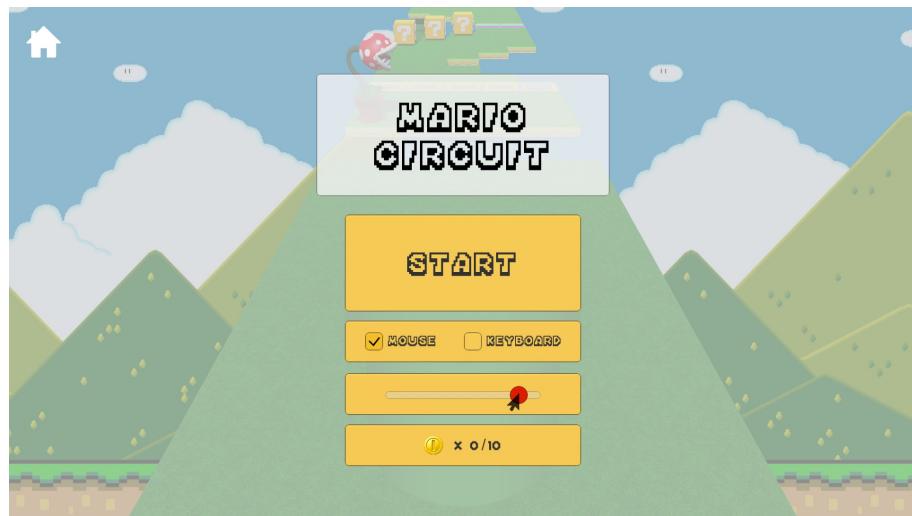


Figura 3.5: Menú d'inici del nivell 1.

Al menú que es mostra a la Figura 3.5 podem veure el nom del circuit que estem jugant i les monedes màximes que hem aconseguit. A la part superior esquerra podem trobar un botó en forma de casa que ens permet tornar al menú principal. El *toggle* de mouse i keyboard ens permet seleccionar el control que volem utilitzar al nivell. En premer el botó d'start si ens trobem als dos primers nivells, veurem el compte enrere i començarem a jugar (Figura 3.6).

Mentre un nivell està en joc podem trobar un botó de pausa a la part superior esquerra que ens permetrà accedir al menú de pausa del joc (Figura 3.7). Si col·lisionem amb alguna entitat o caiem al buit es mostrarà una transició que ens acaba mostrant el menú de mort (Figura 3.8). Si arribem al final del nivell, el joc ens retornarà al menú principal.

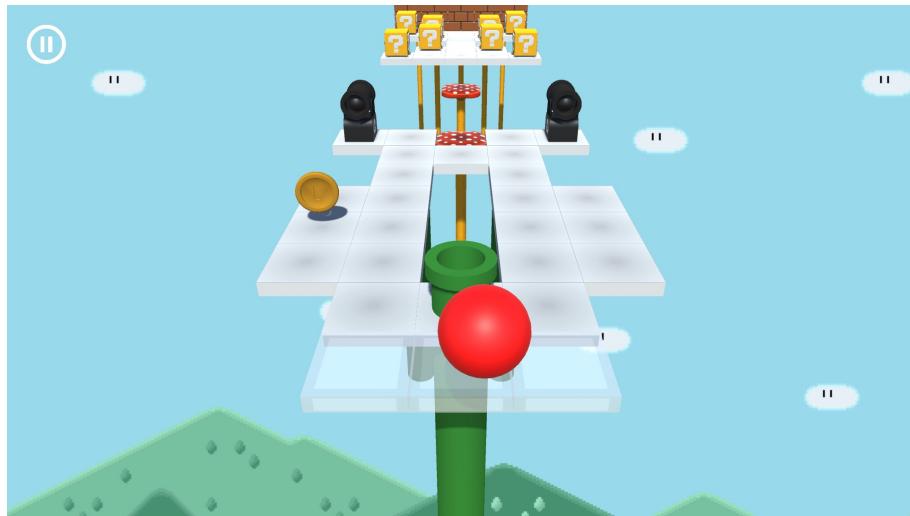


Figura 3.6: *In-game* nivell 2.



Figura 3.7: Menú de pausa.

Quan morim en qualsevol dels nivells accedirem a un menú de mort com el de la Figura 3.8. El botó situat a la part superior esquerra ens permet retornar al menú principal, mentre que el botó de *restart* ens porta al menú d'inici per tal de seleccionar els controls i reintentar el nivell.

3.4 Tiles

En aquest apartat ens centrarem en fer una descripció detallada dels diferents tipus de tiles que hem emprat durant la creació i confecció del joc.

3.4.1 Tile bàsic

Aquest és el tile per defecte en tots els nivells. És un tile bàsic que es troba immòvil i fa de terra durant tot el joc.

Pel nivell 1 hem emprat una textura amb un color verd bàsic. Per sobre hem fet una mescla amb una textura de gespa extreta d'un joc original de Mario (Figura 3.9a). Pel nivell 2, donat que es tracta d'un nivell aeri, hem decidit emprar una textura degradada de tonalitats grises que simulen els colors d'un núvol (Figura 3.9b).

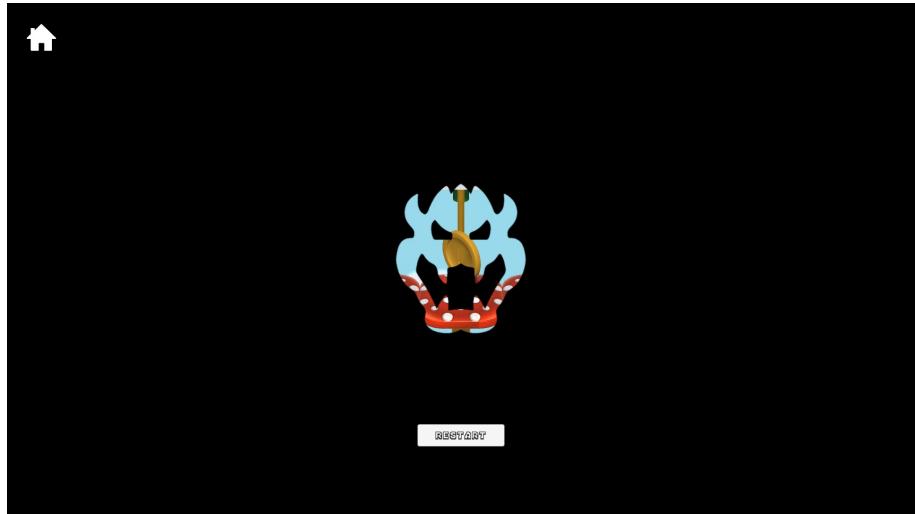
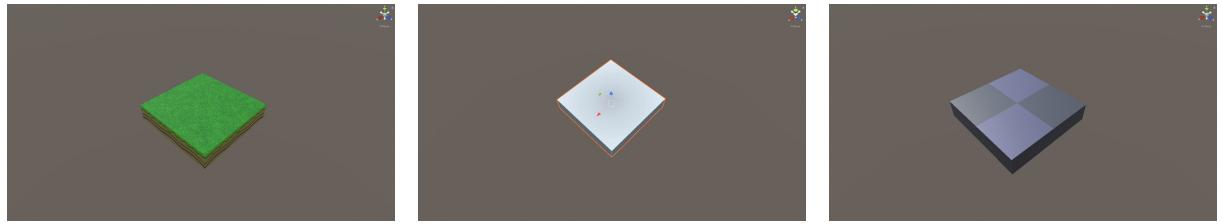


Figura 3.8: Menú de mort.

Finalment, pel nivell 3, hem emprat una textura que simula un tauler d'escacs, ara però, amb tonalitats liles fosques i clares. Ens hem inspirat en paletes de colors extrems de nivells de boss del Super Mario (Figura 3.9c).



(a) Tile bàsic del nivell 1.

(b) Tile bàsic del nivell 2.

(c) Tile bàsic del nivell 3.

Figura 3.9: Tiles bàsics.

3.4.2 Falling tile bàsic

Aquest tile té exactament les mateixes textures aplicades que un Basic tile. El que varia és la funcionalitat. Aquests tiles tenen la característica que no apareixen a nivell de terra. Quan el jugador s'aproxima a la posició en la qual hi ha un basic falling tile veu un forat i fa l'efecte que caurà. De sobte, el tile cau ràpidament del cel, tot emplenant el camí. Aquests tiles es fan servir per als nivells 2 i 3.

3.4.3 Jumping tile

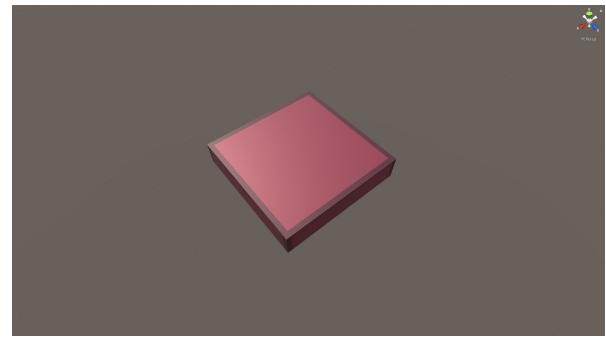
El jumping tile impulsa la bola i fa que executi un salt de 4 tiles de distància (respecte l'eix de les Z). Com que per programar el joc no hem fet servir cap tipus de força, se simula el salt seguint la següent equació:

$$y = f(z) = -\frac{1}{2}(z - 2)^2 + \frac{5}{2}$$

Per als jumping tile dels nivells 1 i 2 hem fet servir un tile especial generat amb Blender (Figura 3.10a) que simula els xampinyons que et fan saltar en algunes pistes del Mario Kart. Per al nivell 3 hem emprat un tile d'un color granat amb uns marges foscos (Figura 3.10b).



(a) Jumping tile de xampinyó.



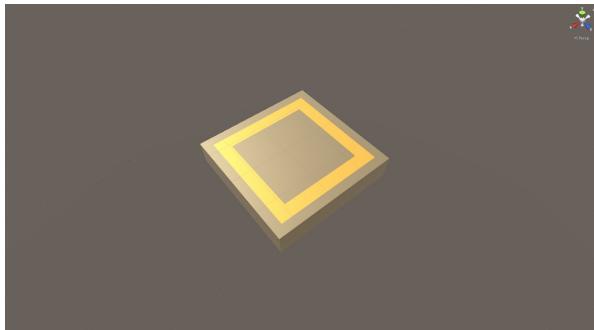
(b) Jumping tile del nivell 3.

Figura 3.10: Jumping tiles.

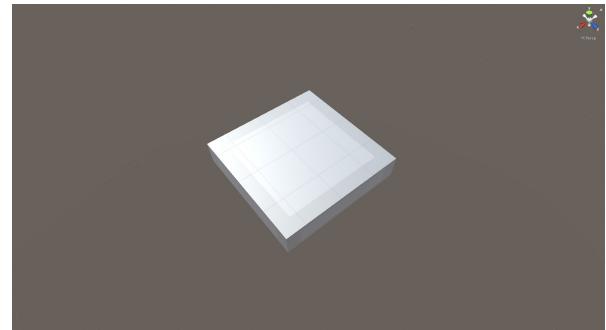
3.4.4 Falling tile

Els falling tile són un tipus de tile amb textura translúcida els quals, quan la bola passa per sobre entren en caiguda lliure. En cap cas provoquen la caiguda de la bola, però genera un efecte interessant.

Tant per als nivells 1 com 3 hem emprat un falling tile amb textura taronja translúcida amb uns marges taronges més opacs (Figura 3.11a). Pel nivell 2 hem emprat un tile similar, però amb tonalitats blanques. D'aquesta manera mantenim les tonalitats celestials del nivell (Figura 3.11b).



(a) Falling tile dels nivells 1 i 3.



(b) Falling tile del nivell 2.

Figura 3.11: Falling tiles.

3.4.5 Forward tile

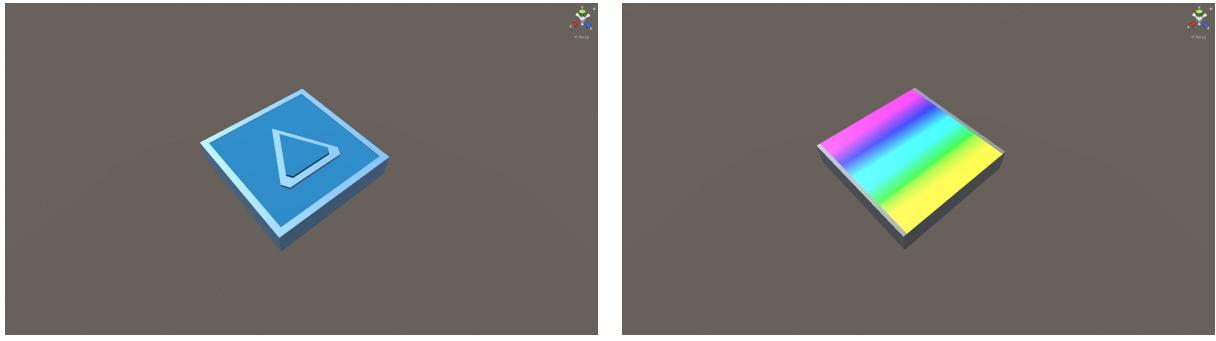
Els forward tile són tiles que tenen just al davant un forat. Quan la bola passa per sobre seu, es traslladen una unitat endavant de manera que el jugador no cau.

Tots 3 nivells tenen la mateixa textura aplicada per a aquests tiles. A més a més, quan la bola se situa sobre els tiles d'aquest tipus, la fletxa que indica el sentit del moviment canvia de color i passa a ser blanca (Figura 3.12a).

3.4.6 Boost tile

Els boost tile actuen com a propulsor de la bola quan aquesta passa per sobre. Es fan servir per al nivell 1 i tenen una textura d'arc de Sant Martí que es va movent amb el temps (Figura

3.12b). Pretén simular l'efecte de les vies propulsores que hi ha al joc Mario Kart. A més, hem afegit particules que simulen un efecte de propulsió a la càmara.



(a) Forward tile.

(b) Boost tile.

Figura 3.12: Forward i Boost tiles.

3.5 Entitats del joc

Com hem explicat en apartats anteriors, amb aquest joc hem volgut retre homenatge no només al RollingSky sinó també a diferents jocs de l'univers Mario com ara Super Mario Bros, Mario Kart, etc. Així doncs, la majoria de les entitats que apareixen al joc són de l'entorn del mític Mario.

Si bé és cert que en moltes pàgines d'internet hem trobat diferents objectes 3D que ens hagueren estat d'utilitat, una gran part dels nostres collisionables els hem creat nosaltres mateixos amb l'ajut del programa d'edició 3D Blender. Ha estat un gran repte perquè cap membre de l'equip tenia experiència amb Blender i hem hagut d'aprendre de pressa per tal d'obtenir resultats aparentment acceptables.

En aquest apartat explicarem les funcionalitats dels diferents objectes així com els diferents passos que hem seguit per a la creació i confecció d'aquests mitjançant Blender.

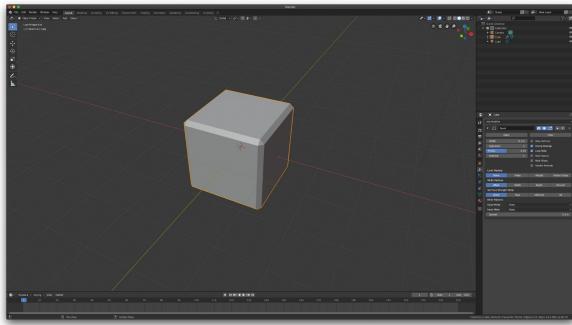
3.5.1 El primer objecte: Mystery Block

Aquest fou el primer objecte que vàrem crear mitjançant Blender. Tenia una forma senzilla i per tant, vàrem pensar que seria suficientment simple com per tenir una primera presa de contacte amb el programa d'edició.

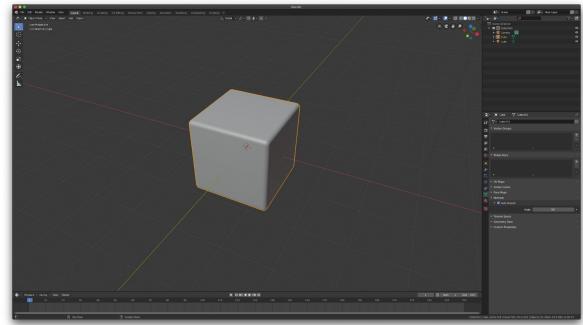
Vàrem partir d'un cub de mida 1x1 i li vàrem aplicar un bevel, que consisteix a aplanar les vores dels objectes per tal d'obtenir un bisellat de les arestes (Figura 3.13a). Seguidament vam aplicar-li una transformació de shade smooth per aconseguir l'efecte de cantonades arrodonides (Figura 3.13b). Arribats a aquest punt, ja teníem la base del mystery block.

A continuació, vam agafar una plantilla del que havia de ser la icona d'interrogant que havía de tenir en cadascuna de les cares (exceptuant la superior i l'inferior). Partint novament d'un cub, vàrem aplicar-li transformacions de retallat fins que vam obtenir l'objecte en forma d'interrogant que estàvem buscant (Figura 3.14a).

Finalment, vam duplicar el símbol d'interrogant i el vam adjuntar a cadascuna de les cares laterals del cub que havíem creat anteriorment. Només ens restà fer quatre forats a les cantonades de cada cara lateral i seleccionar els materials adients, per aconseguir el resultat que estàvem buscant (Figura 3.14b).

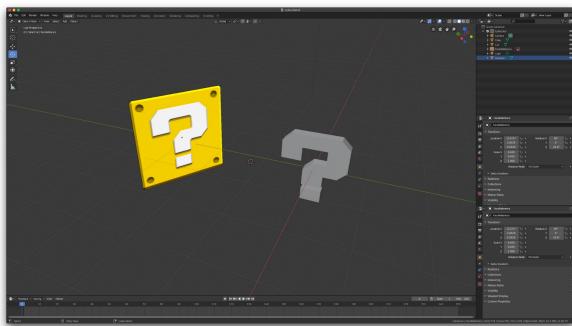


(a) Cub amb bisellat d'arestes.

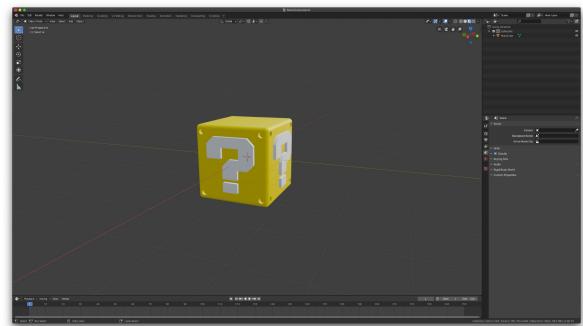


(b) Cub bisellat smooth.

Figura 3.13: Cubs bàsics bisellats.



(a) Plantilla i objecte interrogant.



(b) Mystery block final.

Figura 3.14: Confecció del mystery block.

La funcionalitat que vam pensar per aquest bloc fou purament de distracció. Vam decidir que el bloc estigués sobre el terra del mapa (Figura 3.15a) de manera que el jugador pensaria que és un obstacle. Aleshores, quan la bola s'apropés a l'objecte, aquest s'elevaria girant sobre el seu eix Y i deixaria pas al jugador (Figura 3.15b).



(a) Bloc a terra.



(b) Activació del mystery block.

Figura 3.15: Mystery block in-game.

3.5.2 Brick block

Aquest fou un dels objectes més fàcils que varem haver de crear. Començar generant un petit totxo (Figura 3.16). Tot seguit el vam duplicar i rotar per tal que ens quedés el brick block que tots coneixem de múltiples jocs de Mario. Vam triar unes tonalitats marrons perquè tingués l'aparença original del joc.

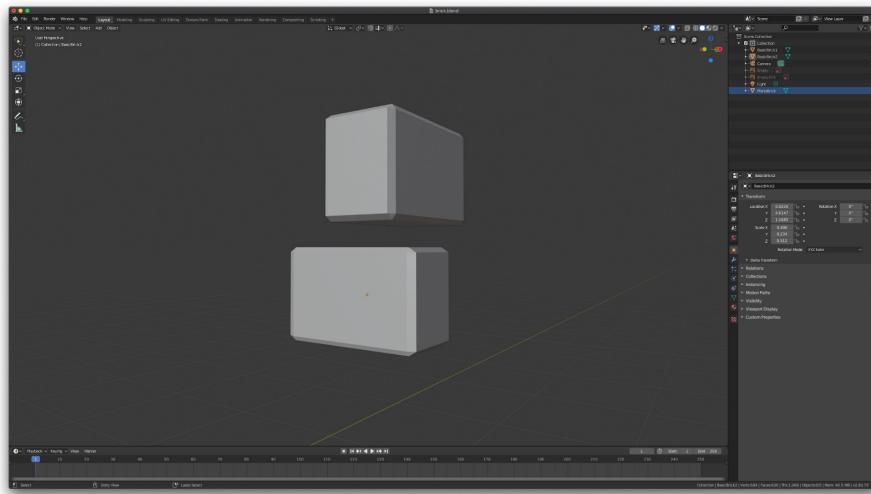
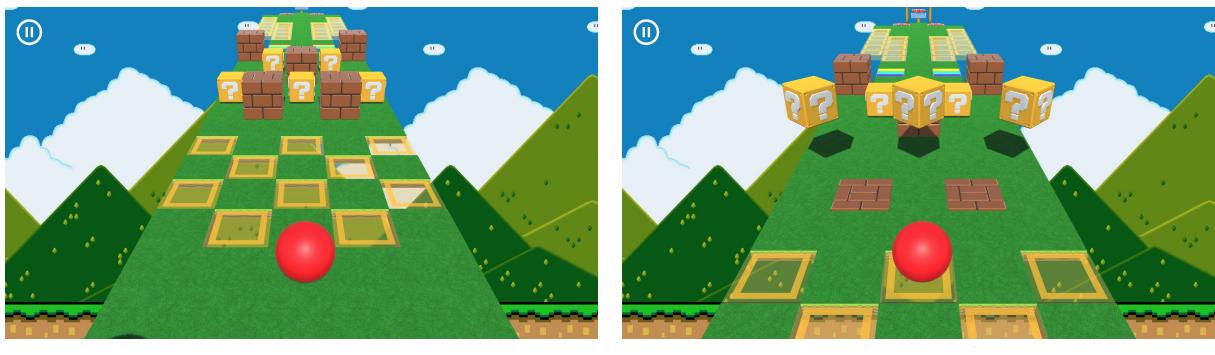


Figura 3.16: Bricks individuals.

La funcionalitat que varem atorgar-li fou, com en el cas del mystery block, purament de distracció. Ara, però la dinàmica seria una mica diferent. Igual que abans, el bloc es trobaria sobre el terra de manera que el jugador pensés en una primera instància que es tracta d'un obstacle (Figura 3.17a). En el moment en què la bola s'apropés, el brick block es traslladaria cap avall, tot obrint-li pas (Figura 3.17b).



(a) Brick block a terra.

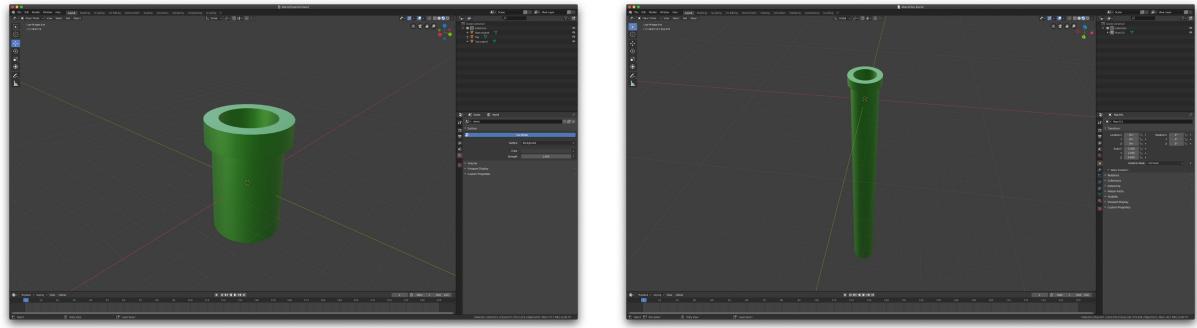
(b) Brick block activat.

Figura 3.17: Diferents situacions del brick block.

3.5.3 Pipe

En el nostre joc basat en RollingSky amb temàtica Mario, no podia faltar la mítica pipe. La seva confecció fou senzilla. Simplement varem haver de crear dos cilindres de diferents radis i situar-los un sobre l'altre de manera que se simulés la forma d'una canonada i tintar-los de color verd (Figura 3.18a). Vam crear dues versions: una pipe curta i una de més llarga (Figura 3.18b) per al segon nivell.

La seva funcionalitat és ser un objecte col·lisionable que apareix de sota terra. En un primer moment sembla que el camí està lliure, però quan la bola s'aproxima a certa distància, la pipe emergeix de sota terra entorpiint el camí. Si la bola xoca amb la canonada, s'acaba la partida. En la (Figura 3.19) es pot apreciar les pipes que ja han emergit al davant de tot de la imatge i al fons es poden veure les que estan amagades sota terra.



(a) Pipe curta.

(b) Pipe llarga.

Figura 3.18: Objectes de Pipe.

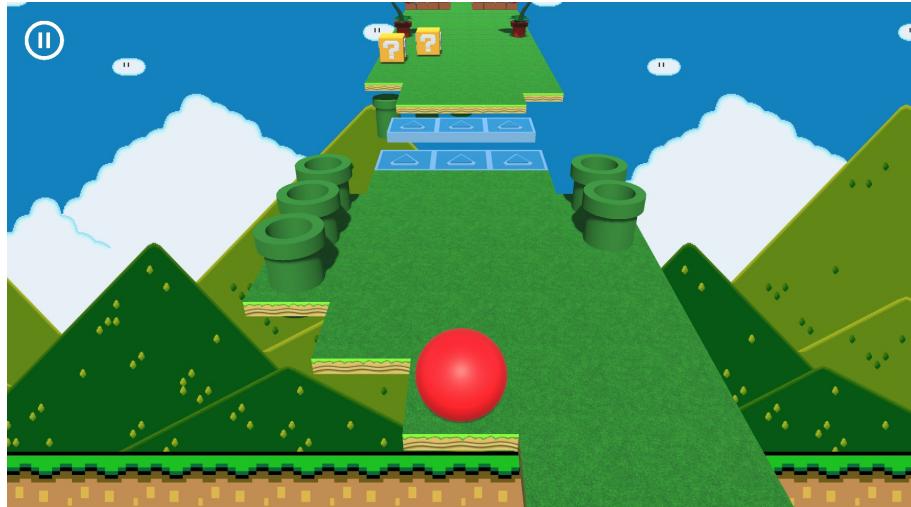


Figura 3.19: Pipes aixecades a l'inici i, al fons, amagades.

3.5.4 Hammer

Aquest és un dels pocs objectes que vàrem decidir agafar d'internet. El motiu és —bàsicament— que estava molt ben fet i replicar-lo ens seria molt més complicat.

L'objecte és un martell que apareix en els jocs de SuperMario Bros. És una adaptació dels martells que llancen els enemics Koopa-Troopa des d'un núvol. Malgrat que l'objecte no l'hem creat nosaltres, sí que vàrem editar la seva animació. D'aquesta manera també ens va servir per tenir una primera presa de contacte amb l'editor d'animacions de Blender (Figura 3.20). L'animació consisteix en deixar caure el martell des d'una posició vertical fins al terra. Un cop ha tocat el sòl, el martell rebota una mica per crear un efecte de caiguda lliure més natural.

La funcionalitat associada al hammer, com es pot intuir del paràgraf anterior, és mantenir-se en posició vertical fins que la bola s'apropi. Aleshores, el martell es deixa anar en caiguda lliure fins arribar a terra i fer un petit rebot. D'aquesta manera, el jugador a priori pensa que té un bloc disponible que, un cop ha caigut el martell, queda ocupat per l'objecte en posició horitzontal. El collisionable ocupa un tile quan es troba en posició vertical i quan cau, ocupa el tile anterior i un d'extra més a l'esquerra (o a la dreta, segons on estigui situat). Si la bola topa amb el martell, s'acaba la partida.

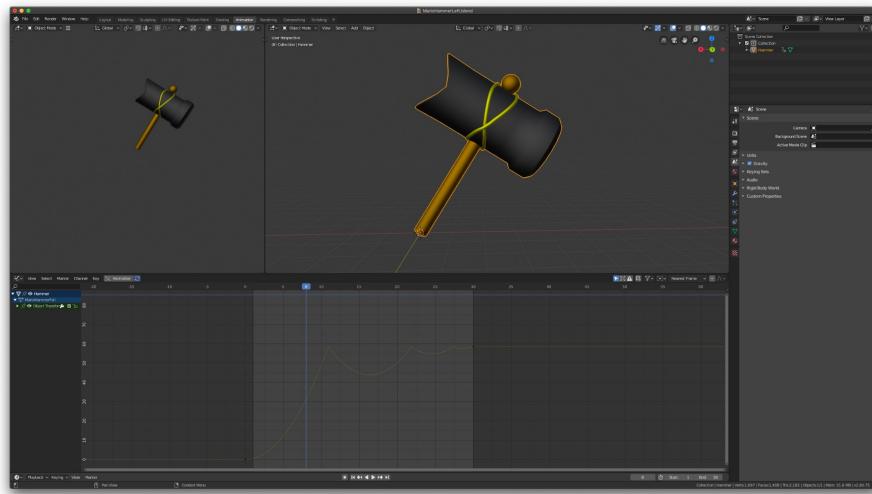


Figura 3.20: Animació bouncing del hammer.

3.5.5 Bullet-Bill i Bill-Blaster

La creació d'aquest objecte fou un dels primers grans reptes que ens varen sorgir durant el projecte. Hi havia diferents bullet-bill per internet, però tenien una qualitat molt baixa. El bill-blaster, per contra, ni tan sols el vam trobar.

Aquest objecte combina la creació dels objectes a partir de volums bàsics geomètrics 3D amb l'animació dels mateixos mitjançant Blender. A més a més, un cop els varem haver exportat a Unity, vam generar sistemes de partícules per simular el foc que surt del bullet-bill un cop ha estat disparat pel bill-blaster (Figura 3.21).

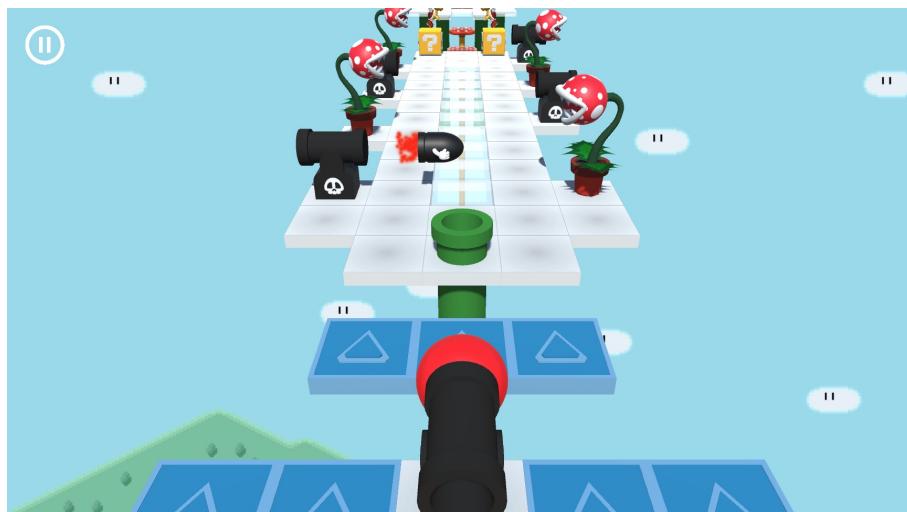


Figura 3.21: Sistema de partícules del bullet-bill.

Com hem comentat en l'anterior paràgraf, la creació d'ambdós objectes varem fer-la a partir de volums geomètrics 3D bàsics. No ho explicarem pas per pas perquè seria massa feixuc, però adjuntem algunes imatges (Figura 3.22) que mostren el procés. En addició a la creació dels objectes, també fou el primer cop que vam crear les textures per als mateixos (per al bill-blaster). Per fer-ho, vam haver de fer un unwrap (Figura 3.22c) del volum i crear la imatge de textura (es pot veure també a la Figura 3.22b) a part mitjançant el programa Gimp.



(a) Confecció del bullet-bill.

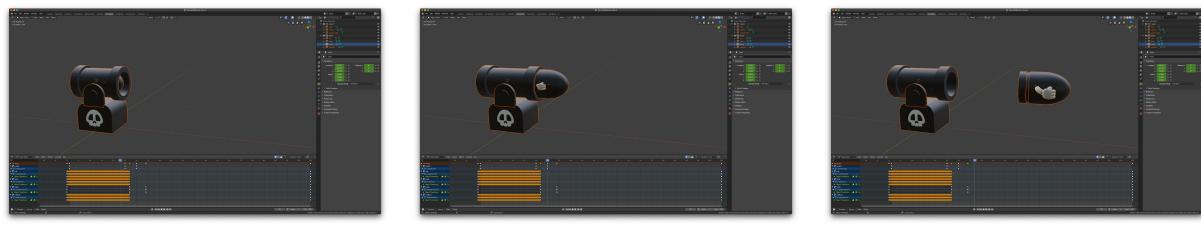
(b) Confecció del bill-blaster.

(c) Unwrap del bill-blaster.

Figura 3.22: Confecció del bullet-bill i bill-blaster mitjançant Blender.

L'animació que vàrem crear fou bastant més complexa que qualsevol altra que haguéssim fet fins aleshores. Consistia en el fet que el bill-blaster es contragués fins al punt en què es tornés a distendre, tot simulant el tret d'una bala. En el moment en què el bill-blaster fes l'animaçió de disparar, el bullet-bill havia de sortir disparat de dintre el canó i activar el seu sistema de partícules. La bala recorre certa distància fins que arriba un punt en què s'atura (Figura 3.23).

El funcionament de l'objecte és que, un cop la bola s'apropa a l'objecte, el bill-blaster dispara un bullet-bill, de manera que si topa amb ell, s'acaba la partida. Vam decidir aplicar rotacions a l'objecte per tal que la bala pogués anar d'esquerra a dreta, de dreta a esquerra i en sentit contrari al moviment de la bola.



(a) Inici de l'animació.

(b) Punt intermig.

(c) Animació avançada.

Figura 3.23: Animació del bill-blaster expulsant un bullet-bill.

3.5.6 Piranha

Un altre clàssic dels jocs de l'univers Mario és la planta carnívora Piranha. Vàrem trobar aquest objecte per internet i vam veure que estava tan ben fet que no el podíem desaprofitar. Novament, per deixar-hi la nostra empremta, vam decidir animar-la.

L'animació de piranha va ser una primera presa de contacte amb la utilitat d'osso (bones) de Blender. Aquesta permet crear una estructura esquelètica en la qual cada os s'associa amb un conjunt de triangles de la malla que forma l'objecte. Aquesta associació la fa automàticament blender en funció de la proximitat dels ossos als diferents triangles i vèrtexs de la malla. Tot i així, un cop es va haver fet la distribució automàtica dels pesos, vam haver de retocar-ne alguns per tal que els moviments dels diferents ossos no afectés parts no desitjades de l'objecte.

L'esquelet de la piranha és relativament simple. Consta d'una columna vertebral i diferents ossos per permetre el moviment de la boca (Figura 3.24). En canvi, aprendre a usar Inverse Kinematics va ser considerablement complex. Aquesta funcionalitat permet generar un os que, quan se li apliquen moviments i/o rotacions, propaga el moviment natural d'una estructura òssia a través dels ossos amb els quals està connectat. Malgrat la dificultat afegida de la metodologia, aplicar-la ens va facilitar molt la posterior animació de l'objecte.

Hem creat dues animacions: En la primera l'objecte està en constant moviment simulant una respiració. La segona consisteix en que, quan la bola s'aproxima, la planta s'aboca sobre aquesta amb la boca oberta fent un gest per intentar mossegar-la. La funcionalitat és la descrita per l'explicació d'aquesta segona animació.

En primera instància vàrem decidir que la planta fos un objecte que en cas que la bola s'aproximés suficientment, la mossegaria i, en cas de col·lisió, s'acabaria la partida. Així i tot, un cop ho vàrem haver testejat en el joc, vam veure que l'animació era massa ràpida i el jugador no tenia massa temps de reacció per intentar evitar el xoc. Així doncs, vam decidir que l'animació d'atacar s'executaria com a maniobra de distracció. Tot i així, la bola pot col·lidir amb el test de la piranha i, si es dóna el cas, s'acaba la partida.

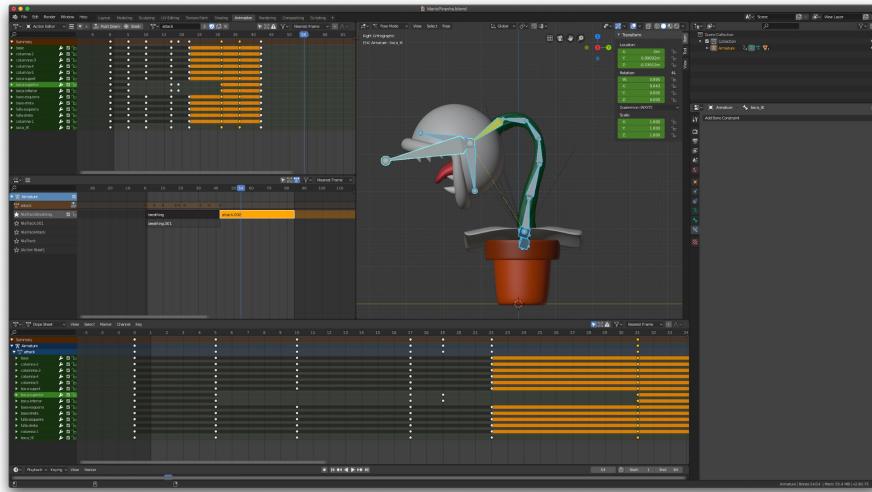
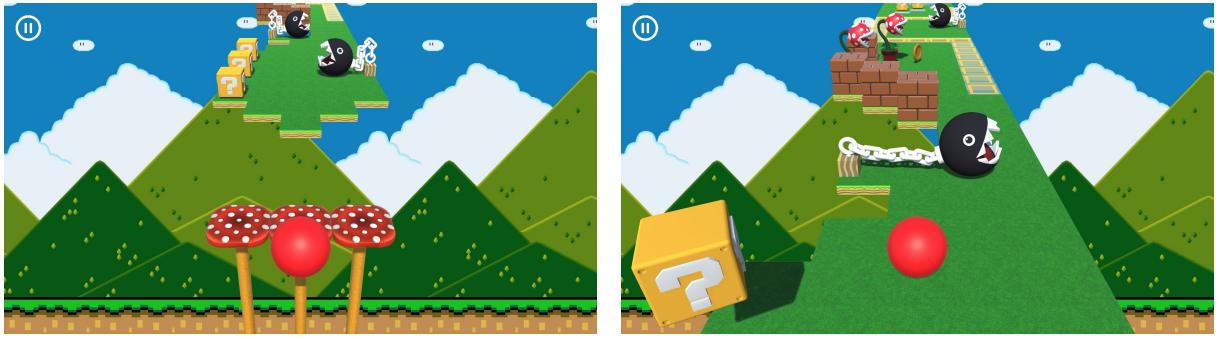


Figura 3.24: Animació de la piranha.

3.5.7 Chomp

El mític chomp! Quants cops ens ha clavat una mossegada sorpresa mentre jugàvem a Mario Kart? Passar de primer a últim lloc en qüestió de segons... Varem trobar l'objecte 3D de molt bona qualitat i no vam desaprofitar l'oportunitat. La principal dificultat va ser animar les cadenes. La funcionalitat que teníem al cap no era un altre que l'original: el chomp estaria quiet (Figura 3.25a) i quan la bola s'apropés, s'avancaria ràpidament per mossegar-la (Figura 3.25b). En cas de col·lisió, s'acabaria la partida.

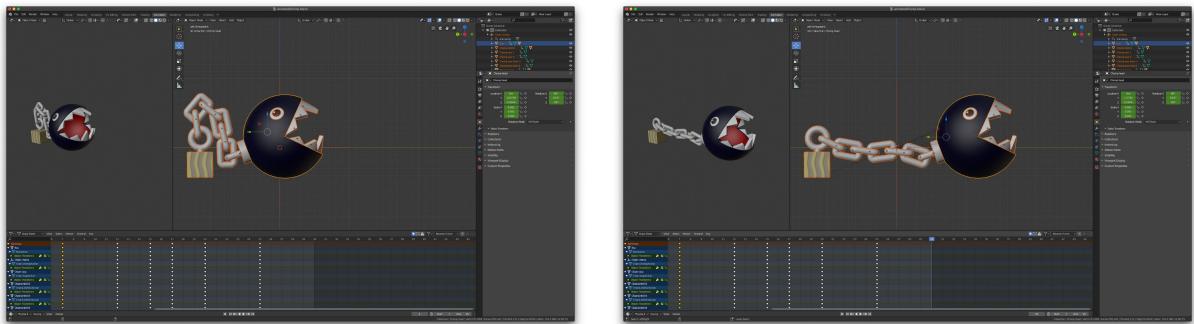
Tal com hem comentat anteriorment, la dificultat en aquest cas fou animar l'objecte (Figura 3.27) per tal que totes les cadenes es moguessin correctament i simulassin un "estira-i-arrostra" de les cadenes. Novament ens vàrem plantejar usar inverse kinematics, però per diferents motius vam acabar concloent que seria molt més complicat que no pas dur a terme l'animació cadena per cadena.



(a) Comp quiet quan el jugador està lluny.

(b) Chomp atacant quan s'aproxima el jugador.

Figura 3.25: Estats del chomp.



(a) Animació del chomp quiet.

(b) Animació del chomp atacant.

Figura 3.26: Animacions del chomp a Blender.

3.5.8 Thwomp

Aquesta entitat fou especialment difícil de confeccionar. En aquest cas no tant per l'animació, sinó per l'elevat nombre de parts que en formen part i per texturar les mateixes. Ens va ser molt complicat trobar els materials adequats, però el resultat final ha estat satisfactori.

El thwomp està format per un cub amb les vores bisellades (sense shade-smooth) que serveix com a suport de la resta de parts (Figura 3.27a). També consta d'un segon cub més petit que el primer que sobresurt per la davant i per darrere en la forma característica del thwomp original de Mario (Figura 3.27b). Finalment consta de dues fileres de punxes que envolten les cares superior, inferior i laterals del primer cub (Figura 3.27c). Per texturar l'objecte, varem fer un unwrap de tota la figura i vam aplicar-li parts estratègiques d'una textura original del thwomp d'un joc de Mario (Figura 3.27d).

El funcionament és l'original del Mario. El thwomp es troba elevat sobre l'eix Y i, quan la bola s'apropa es deixa anar en caiguda lliure fins que toca terra, tot obstruint el camí del jugador. En cas que hi hagi col·lisió, s'acaba la partida. No vam necessitar animar l'objecte perquè el moviment es podia generar fàcilment a través d'un script que modifiqués la posició en l'eix Y de l'entitat seguint la fórmula de la caiguda lliure.

3.5.9 Koopa-troopa i blue shell

Per al segon nivell –que, com hem explicat anteriorment, està ambientat en un nivell aeri de Mario–, varem decidir tenir algunes entitats i/o objectes amb ales que simulessin estar volant pel mapa i alhora servissin d'objectes col·lidibles. Un clàssic que reuneix aquestes característiques és la figura del koopa-troopa i els seus shells. Concretament aquell blue shell que arruïna totes

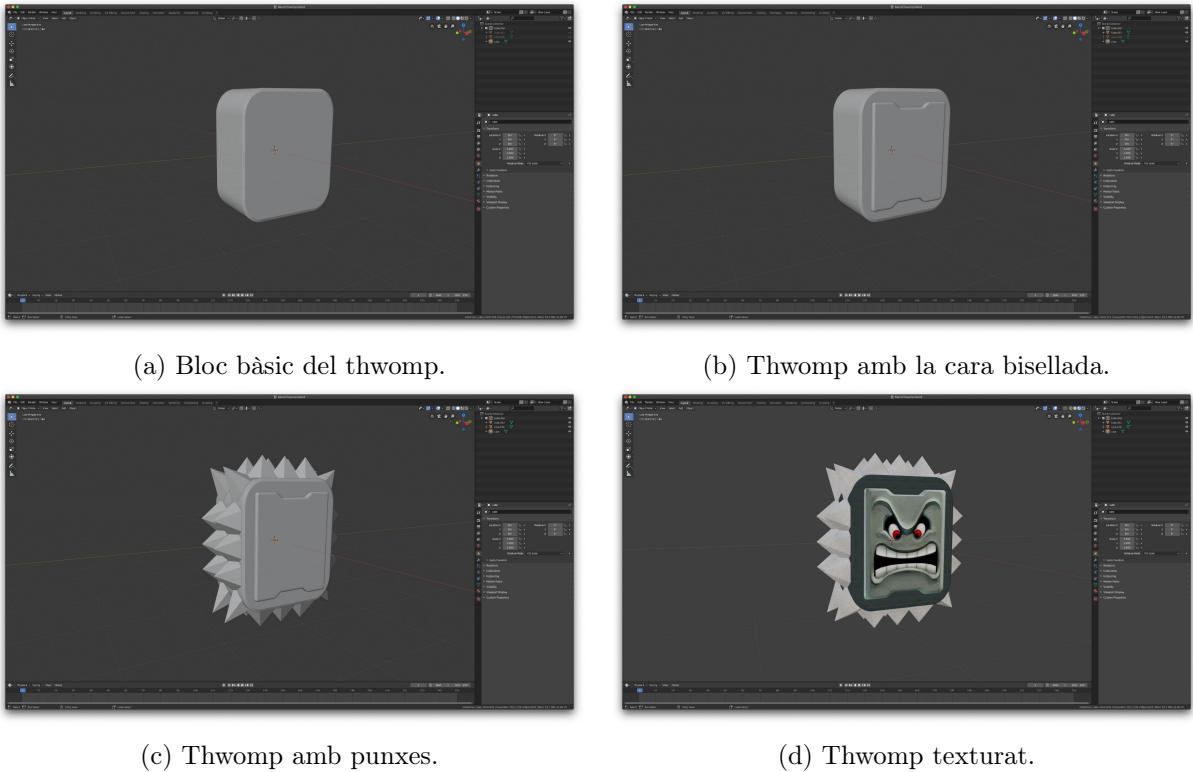


Figura 3.27: Confecció del thowmp a Blender.

les partides de Mario Kart... Ambdós objectes eren massa complicats per a generar-los. Així doncs, vàrem agafar dos models suficientment convincents i de bona qualitat d'internet.

Això sí, l'animació corre a compte nostre. Vam afegir una estructura òssia bàsica per les ales tant del koopa com del shell (Figura 3.28) mitjançant Blender i les vàrem animar de manera que les entitats anessin fent salts en loop i movent les ales. Amb aquest moviment, sembla que els objectes estiguin flotant en l'aire del nivell.

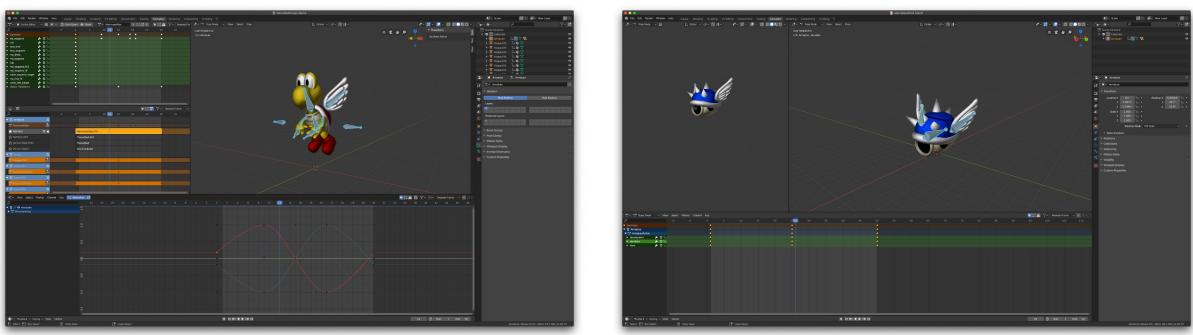


Figura 3.28: Esquelet del koopa-troopa i el blue shell.

El koopa-troopa té la funcionalitat de mantenir-se a l'aire i, quan la bola faci algun salt, hagi d'esquivar-lo. En cas que col·lideixi s'acabarà la partida. El blue shell simplement és de decoració. N'hi ha un flotant cap al final del segon nivell (Figura 3.29).

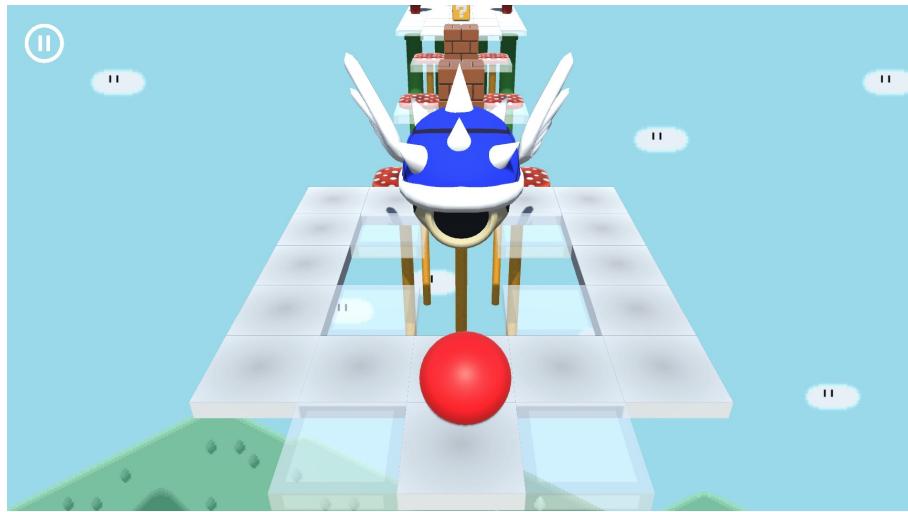
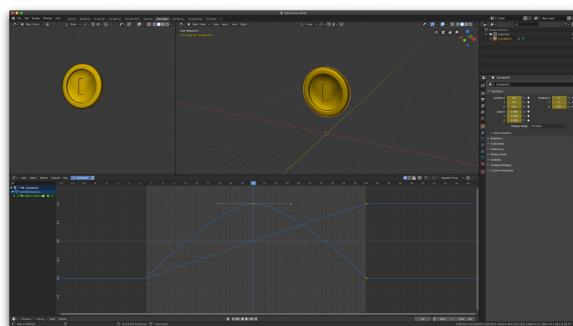


Figura 3.29: Blue shell en partida.

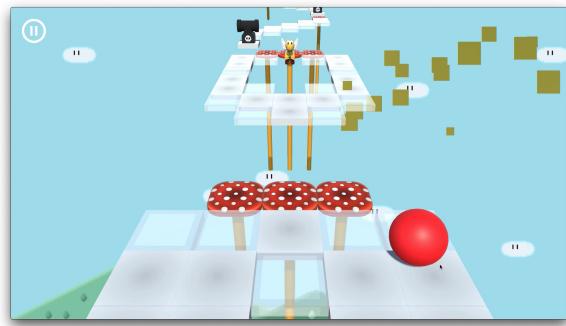
3.5.10 Moneda

La moneda fou dels objectes més simples de recopilar i d'animar. Vam agafar un model acceptable d'internet i varem editar mitjançant Blender la seva animació. Aquesta simplement consistia en un gir constant respecte l'eix Y i un moviment vertical que simulés un rebot constant. Per tal que el rebot fos convincent i no semblés que la moneda simplement pujava i baixava, vam convinar l'eina de DopeSheet de Blender (que facilita l'edició per frames) amb l'eina de Curves que et permet modificar les línies de moviment de l'animació del teu objecte (Figura 3.30a).

Aquest objecte ens serveix com a mètode de puntuació. Hi ha un total de 10 monedes repartides per tot el nivell. Quan la bola col·lideix amb una moneda, aquesta desapareix i s'activa el seu sistema de partícules, que consisteix en dispersar un conjunt de partícules daurades (Figura 3.30b).



(a) Corba del moviment de la moneda.



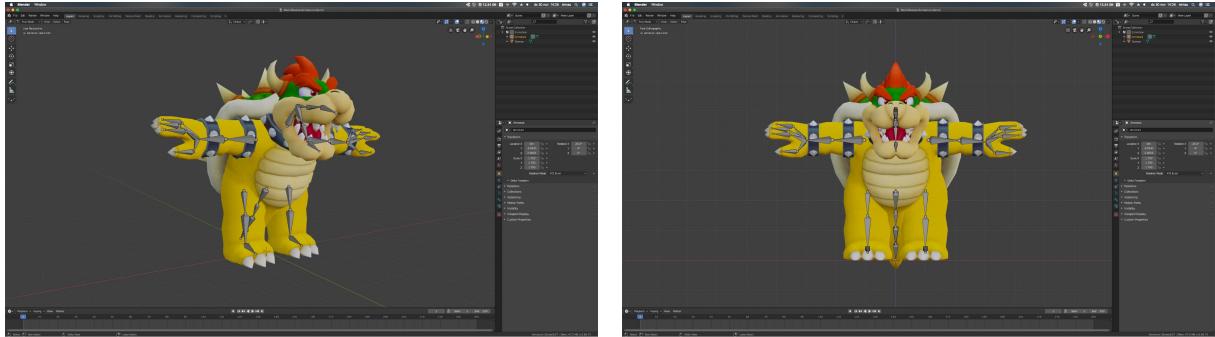
(b) Sistema de partícules de la moneda.

Figura 3.30: Moviment i partícules de la moneda.

3.5.11 El gran repte: Bowser

Aquest fou el model culminant de tot el projecte. Un cop vam tenir l'experiència d'edició amb Blender gràcies a la resta d'entitats, vam pensar en basar el tercer nivell en lluitar contra un boss. I qui és el més dolent de tots els dolents del món Mario? Sens dubte, Bowser!

El primer pas per a l'animació del model fou separar-lo per parts. Seguidament varem afegir-li una estructura òssia bàsica (Figura 3.31) per fer els primers moviments i testejar el seu funcionament. Un cop vam tenir els ossos lligats a la malla, vam haver de fer un retoc considerable dels pesos (Figura 3.32) que cada os tenia associat als diferents triangles de la malla. Ho varem necessitar perquè si no, a l'hora de moure, per exemple, ossos de la cama, també es modificava el moviment d'alguns triangles de la panxa.



(a) Esquelet bàsic del Bowser.

(b) Vista frontal de l'esquelet.

Figura 3.31: Esquelet del Bowser.

Seguidament, vam afegir alguns ossos més que, després d'haver provat a fer alguns moviments, varem adonar-nos que necessitaríem en addició als que ja teníem. Quan varem haver regulat els seus pesos (Figura 3.32), vam passar a afegir diferents inverse kinematics per tal de facilitar-nos l'edició de les animacions. Aquest model tenia una capacitat de personalització de moviments molt més elevada que la resta d'entitats que havíem usat fins aleshores. Així doncs, va ser necessari molts més IK que en els anteriors objectes. Vam afegir un IK per a cadascun dels dits (de totes dues mans) i un altre que controlés el moviment dels braços (articulacions del colze). Per tal que els colzes apuntéssin allà on volíem, varem generar ossos lliures que ens serviríem com a “apuntadors” dels IK (Figura 3.33).



Figura 3.32: Calibració de pesos de l'esquelet.

Després d'afegir les IK dels braços varem afegir-ne també pels peus i les cames (Figura 3.34). Novament, per tal que els genolls sempre apuntéssin allà on volíem, vam generar ossos lliures que tenien la funció d'apuntadors de les articulacions de les cames.

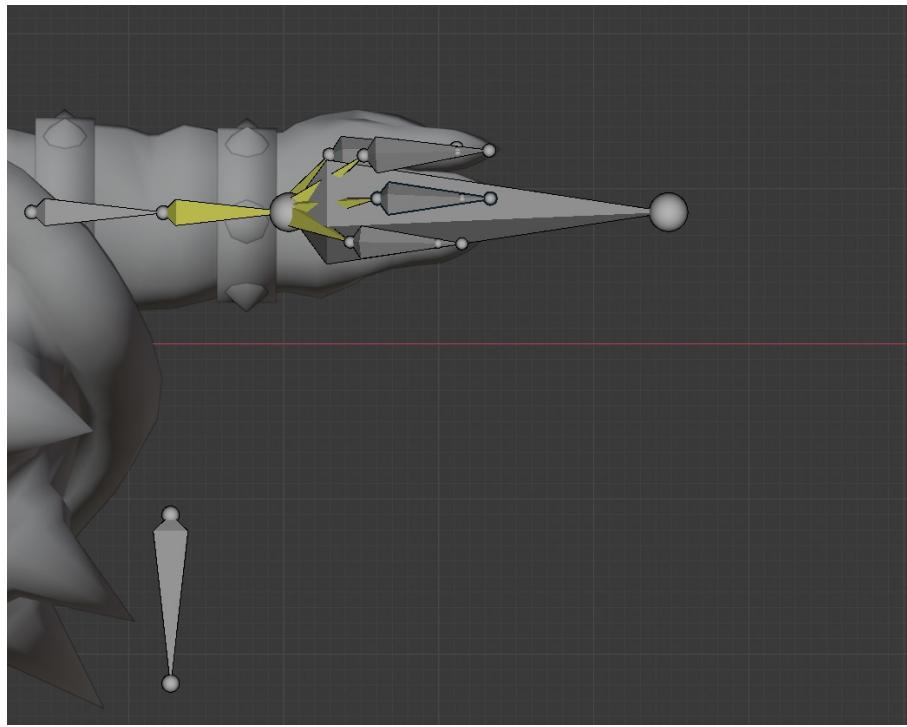


Figura 3.33: Vista *top* de l'*inverse kinematics* del braç del Bowser.

Un cop vam tenir l'armadura completa amb els pesos adequats i els IK necessaris per al correcte moviment de totes les parts, vam començar amb l'animació. Per tal que la figura del boss no fos monòtona, varem pensar en tots i cadascun dels moviments que volíem que fes durant el transcurs del joc mitjançant un diagrama de flux (Figura 3.35) que exemplifiqués el comportament que el Bowser havia de tenir durant tot el nivell. El diagrama és una aproximació informal del que havia de fer el boss durant el transcurs del nivell 3. Les animacions han acabat essent les següents:

- Salt inicial. El boss es presenta apareixent d'un salt en l'escena.
- "Roar" inicial. Un cop ha aterrat, el Bowser fa un rugit terrorífic d'esquerra a dreta de l'escena (Figura 3.36a).
- Salt a sobre el núvol. El Bowser fa un salt i aterra sobre un núvol que apareix del no-res. Durant tot el nivell estarà situat sobre el núvol i davant la bola (Figura 3.36b).
- Atac. En alguns punts del nivell, el boss atacarà la bola amb una flamarada de foc (sistema de partícules) (Figura 3.36c). Si la bola toca la flama s'acaba la partida.
- Impacte. En alguns punts del nivell, la bola pot agafar l'objecte Flower que dispararà fragments punxeguts en direcció al boss i li inflingiran mal (Figura 3.36d).
- Get up. Si el boss encara té vida després d'un impacte, s'aixecarà d'un salt.
- Respiracions. Durant tota l'estona que el Bowser no duu a terme cap de les animacions anteriors, restarà respirant cíclicament de manera que fa l'efecte que el Bowser està constantment interaccionant.

El Bowser té un total de 6 punts de vida. Cada cop que rep un impacte, es pot apreciar com baixa la vida en la lifebar que hi ha durant tot el nivell a la part superior dreta (Figura 3.36d). Si el Bowser arriba a 0 punts de vida o s'acaba el nivell i la bola no ha impactat amb cap objecte, aquest morirà i executarà una última animació d'impacte, tot quedant-se en la posició final d'aquesta.

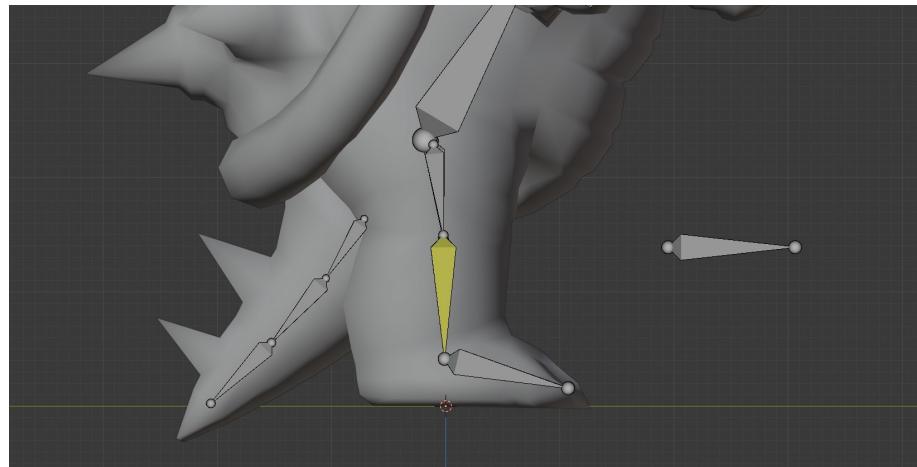


Figura 3.34: *Side view de l'Inverse kinematics de les cames del Bowser.*

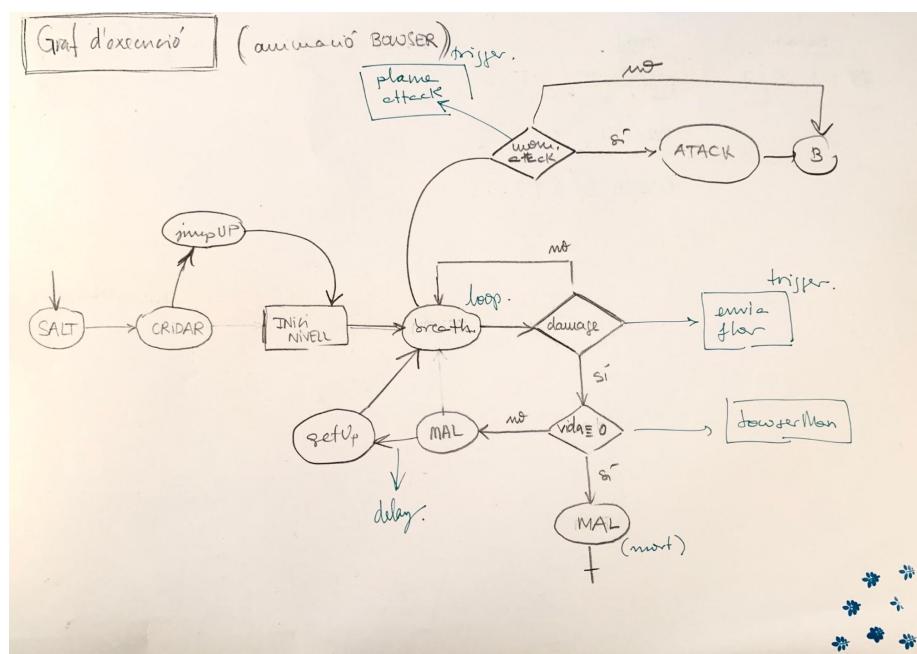


Figura 3.35: Diagrama de flux de l'animació del boss.

3.5.12 L'objecte final: la bandera

Després d'haver generat tots els objectes ens en vàrem adonar que teníem un final dels nivells massa simple. No passava res i retornava el jugador a l'inici. Aleshores, vam pensar, per què no fem un últim objecte que serveixi per culminar tots els nivells? I quin havia de ser si no ho era la clàssica bandera del castell dels jocs de Mario?

Vam aprofitar els bricks que havíem fet servir per al brick block i els vàrem reescalar i situar en cercle per acabar formant la forma d'un castell. Per acabar, vàrem situar un cilindre al mig de l'estructura amb una esfera al capdemunt i la bandera amb la "M" de Mario. La funcionalitat d'aquesta entitat o objecte és que quan el jugador superi el nivell, la bola faci un últim salt i acabi donant voltes al màstil fins arribar a terra (Figura 3.1).



(a) Rugit inicial.



(b) Salt al núvol.



(c) Atac del boss amb flama.



(d) Impacte de l'atac del jugador.

Figura 3.36: Animacions del boss (Bowser).

4 Metodologia

En aquest apartat descriurem el programari emprat durant el desenvolupament del projecte i la planificació establerta. A més a més, també enunciarem els mètodes i programes de comunicació i gestió de versions i tasques emprats.

4.1 Programari

4.1.1 Unity

Unity ha estat l'aplicatiu que més hem emprat durant el projecte. Unity és un motor de videojocs i alhora una plataforma de desenvolupament orientada als programadors de videojocs. Té un àmpli conjunt de funcionalitats associades que fan possible una etapa de programació molt ben integrada en la seva aplicació. Per exemple, els sistemes de partícules, editors d'interfícies d'usuari (UI), botiga d'assets, compatibilitat amb diferents programes externs (d'entre els quals hi ha Blender), etc.

4.1.2 Blender

Un altre dels programes que més hem usat durant l'etapa de desenvolupament del videojoc ha estat Blender en la seva versió 2.8. Blender és un programari lliure multiplataforma d'edició 3D. Inclou eines pel modelatge d'objectes, animació, render, edició de materials, entra d'altres. Nosaltres l'hem fet servir per generar la majoria de models d'entitats i animacions que hi ha al llarg dels mapes dels diferents nivells. La seva compatibilitat amb Unity ens ha facilitat en gran mesura la feina d'exportació des de Blender cap a Unity sense necessitat de passar per cap format intermig.

4.1.3 Github, Gitkraken i Sourcetree

Github és una plataforma de desenvolupament col·lectiu que es fa servir per allotjar projectes. Ens ha estat útil el seu sistema de control de versions per tal de mantenir en tot moment una versió actualitzada del nostre joc.

Els dos membres del grup hem emprat aplicatius diferents com a eina per fer els merge. Tant Gitkraken com Sourcetree són eines que ens han estat molt útils i que han demostrat dur a terme una feina de gran qualitat per facilitar el dia a dia dels programadors, en aquest cas, nosaltres.

4.2 Planificació

4.2.1 Reunions setmanals

Casualment els dos membres del grup no vivim a Barcelona, però vivim a dos minuts l'un de l'altre. Això ha facilitat que cada setmana ens poguem reunir en alguna de les nostres cases per tal de debatre quins eren els passos a seguir durant el desenvolupament del videojoc i mica en mica construir el que ha esdevingut la nostra versió final. En cada reunió, com a mínim, es tractaven els següents punts:

- Seguiment de disseny de cadascun dels nivells (1, 2 i 3).
- Seguiment de disseny dels menús (inici, inici de nivell, etc).
- Seguiment de la creació dels diferents models mitjançant Blender.
- Seguiment de l'animació dels models.

4.2.2 Github

Per mantenir un control de versions hem utilitzat un repositori de GitHub (Figura 4.1). Hem utilitzat les eines Sourcetree i GitKraken (Figura 4.2) per manegar-ho d'una manera més senzilla i eficaç. El repositori només consta d'una branca màster on es fan tots els commits. Aquests commits sempre tenen una descripció que indica quin canvi afegeix aquesta modificació.

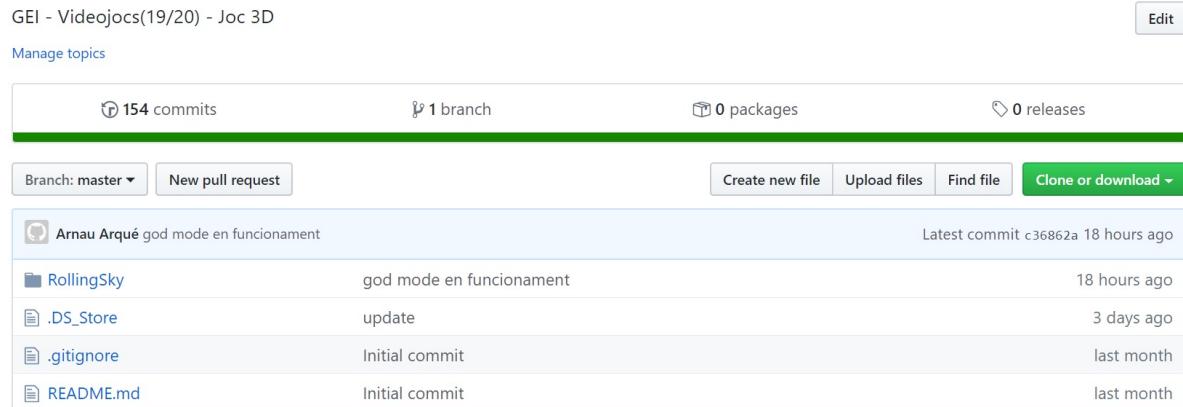


Figura 4.1: Captura de pantalla del repositorio de Github.

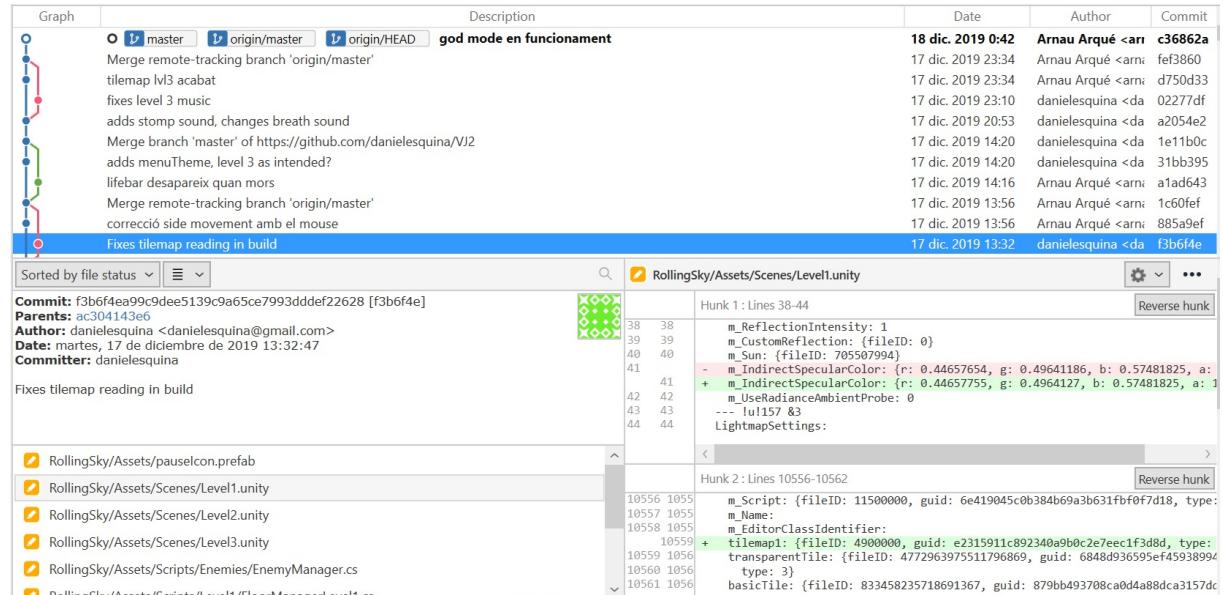


Figura 4.2: Captura de pantalla del sourcetree.

4.2.3 Trello

Per organitzar el projecte i portar un seguiment de les tasques a fer, hem utilitzat Trello. Aquesta eina permet moure tasques en diferents llistes, afegir descripcions (Figura 4.3b) o links d'interès. Hem utilitzat 5 llistes (Planned, InProgress, Tested, Completed, Info) (Figura 4.3a) que ens han permès classificar les tasques segons el seu estat.

(a) Captura de pantalla de les llistes *completed* i *info*.

(b) Captura de pantalla de la descripció d'una tasca.

Figura 4.3: Tasques establertes.

4.2.4 Discord

Per garantir la comunicació en temps real de membres de l'equip hem utilitzat Discord. Aquest programa permet fer trucades de manera que els participants poden compartir pantalla, arxius o apuntar qualsevol detall en un xat. D'aquesta manera hem pogut fer pair-programming en tasques on era indispensable.

4.2.5 WhatsApp

Hem utilitzat WhatsApp per comunicació instantània. Això ens ha permès resoldre dubtes d'implementació en molts casos i planificar temporalment les tasques a fer i la divisió d'aquestes entre els membres del grup.

5 Conclusions i valoració personal

Un cop acabat el projecte i realitzada la resta de la memòria ha arribat el moment de fer una valoració personal i fer alguns comentaris en relació amb l'organització del treball.

Per nosaltres ha estat una experiència totalment nova. Fins ara mai havíem treballat amb Unity ni havíem programat un videojoc d'aquestes característiques. Alhora, tampoc havíem treballat amb cap programari semblant a Blender. És cert que un dels components del grup havia cursat l'assignatura de Gràfics de la FIB i tenia certa experiència amb OpenGL i amb els models 3D, però tot i així ha estat una vivència molt diferent de tot el que havíem treballat fins ara.

La idea de programar un videojoc 3D ens entusiasmava. A més a més, el RollingSky ha estat un joc que tots dos hem jugat durant molts anys als nostres mòbils. Tot i així, no vàrem trigar gaire a notar que el temps se'n tirava al damunt. Creiem que vàrem "perdre" massa temps aprenent a usar Unity i els seus complements. D'alguna manera ens va passar que, a mesura que avançàvem en la programació del videojoc, també apreníem més a fer servir el programari (cosa que no és pas dolenta). Arribats a aquest punt, ens n'adonàvem d'errates que havíem comès temps enrere i que, per com havíem estructurat el joc, en aquell moment ens va ser molt costós de canviar. Per dir-ho fàcil i ràpid, ens dóna la sensació que hem après Unity a còpia de caure i caure. I això, quan tens un projecte amb un temps considerablement limitat, és un problema.

L'altre punt d'autocrítica està més relacionat amb Blender que no pas amb Unity. Voler abraçar no només aprendre a usar Unity sinó que també introduir-nos al món del Blender va ser una jugada molt arriscada pel temps que disposàvem. Blender, com hem dit en punts anteriors, és un programari gratuït d'edició 3D i, valorant-ho des d'un punt de vista objectiu, és extremadament complet i té una capacitat de personalització excepcional. Això és perfecte perquè et permet generar qualsevol cosa que et pugui passar pel cap, però també té un punt negatiu i és que el programa és tant complet com complex d'utilitzar, d'aprendre i d'entendre.

En relació amb el procés de pluja d'idees només podem dir coses favorables. En un principi volíem que cada nivell tingués una temàtica diferent (el primer de Mario, un altre ambientat en la selva...). Així i tot, després de veure el potencial que tenia l'univers Mario, vàrem decidir ambientar tot el joc en la mateixa temàtica. La veritat és que fou divertit començar a pensar en els objectes que faríem servir, les seves funcionalitats, animacions que durien a terme... És cert que potser vam ser més ambiciosos del que se'n demanava, però veient el resultat, estem molt contents i satisfets del resultat final i de com es veuen i interactuen els objectes i les entitats en cadascun dels nivells.

Com es pot apreciar en tota la memòria, no només hem centrat tots els esforços en Unity. Una gran part de la feina l'hem tingut també amb Blender i, en concret, amb l'animació del boss final; el Bowser. Ha estat un treball difícil i que ens ha ocupat setmanes de feina. Primer amb l'armadura (ossos), seguint pel procés de recalibrar els pesos de cada os i acabant per la generació de tots els moviments i les animacions. L'únic que podem dir és que estem molt contents amb el resultat final i que, malgrat que sabem que no és perfecte, creiem que per haver estat la nostra primer animació, ha quedat prou convincent.

Havent acabat el projecte i veient-ho des d'aquesta perspectiva, creiem que, independentment de la nota que acabem obtenint, hem fet una bona feina i hem estat constants amb el treball. Setmanalment ens reuníem i avançàvem i, malgrat que aquestes últimes setmanes han estat intenses (també per la feina acumulada d'altres assignatures), hem treballat contínuament en el projecte fins a arribar a un resultat final del qual estem contents i satisfets.

6 Bibliografia i webgrafia

Nota: Les següents referències presenten la següent estructura:

“Concepte”, Autor (si en té). URL. Descripció.

6.1 Pàgines web

“Trello”. www.trello.com. És una pàgina de taules de desenvolupament de projectes.

“Google Drive”. <https://drive.google.com/>. És la pàgina principal del sistema d’emmagatzematge de Google.

“Opacity of an object”. <https://bit.ly/36K02mv>. Pàgina de l’Stack Exchange de Blender on s’explica com modificar l’opacitat d’un objecte.

“Bullet Bill”, d’Anthony Yanez. <https://bit.ly/2S6TaLM>. Pàgina d’Sketchfab que conté un exemple de model de Bullet Bill de Mario.

“Mario Hammer”, d’Antony Yanez. <https://bit.ly/2EvTGLo>. Pàgina d’Sketchfab que conté un model de Hammer de Mario.

“Mario coin”, d’Antony Yanez. <https://bit.ly/2Qb8rZF>. Pàgina d’Sketchfab que conté un model de la moneda de Mario.

“Preserving aspect ratio when unwrapping”. <https://bit.ly/2YZuSEW>. Pàgina de d’un fòrum de Blender on s’explica com fer l’unwrap d’un model preservant l’aspecte.

“Calculadora gràfica”. <https://www.desmos.com/calculator>. Pàgina d’una calculadora online de funcions i gràfiques.

“NLA editor and actions”. <https://bit.ly/2tqVdQt>. Pàgina de RGB-LABS que explica com fer servir l’editor NLA i les accions de Blender 2.8. Està en alemany, però és molt útil.

6.2 Vídeos

“Best places to find 3D models”, de Max Novak. <https://youtu.be/XGjBOLY8utg>. És un vídeo de la plataforma YouTube que exposa les millors pàgines per trobar models 3D.

“Basic smoothing”, de SadowickDevelopment. <https://youtu.be/HfDBVjLyQtc>. És un vídeo de YouTube que exposa els elements bàsics de l’smooth shading de Blender 2.8.

“Modeling Super Mario Bomber”, de Redninja Multimedia Productions. <https://youtu.be/55wDgt2kuDk>. És un vídeo de YouTube en el qual s’exposa com modelar un super mario bomber bàsic.

“Alpha transparent materials”, de KatsBits. <https://youtu.be/z2cFVVotcs8>. Vídeo que exemplifica com generar materials translúcids amb Blender 2.8.

“How to animate in Blender 2.8”, de Thilakanathan Studios. <https://youtu.be/Huu07RK01RE>. Vídeo que explica els elements bàsics de l'animació amb Blender 2.8.

“Volumetric clouds”, de PuckLovesGames. <https://youtu.be/LLUUIAKFgWg>. Vídeo que mostra com es generen núvols volumètrics des de Unity.

“How to create fog”, de Jimmy Vegas. <https://youtu.be/UgJE3TgT3o8>. Vídeo que mostra com generar boira des de Unity.

“Tips using bevel”, de TopChannellon1. <https://youtu.be/2jTLbjbN1DY>. Vídeo que exemplifica com usar l'eina bevel de Blender 2.8.

“Armature tutorial”, d'Steven Scott. <https://youtu.be/l3IEwxHRs3g>. Vídeo que mostra com generar una armadura bàsica mitjançant Blender 2.8.

“How to create and bake normal maps”, de Darkfall. https://youtu.be/Hpdwo6F_WYQ. Vídeo que ensenya com generar un normal map a partir d'una textura mitjançant Blender 2.8.

“How to bake normal maps”, de Bestdani. <https://youtu.be/h24akA8K-40>. Vídeo curt i simple de com generar un normal map amb Blender 2.8.

“Procedural lava”, de Ducky 3D. <https://youtu.be/HOYYB50qbjk>. Vídeo que explica com generar lava procedural amb Blender 2.8.

“How to unwrap and bake textures”, de One Wheel Studio. <https://youtu.be/c2ut0Trcdi0>. Vídeo en el qual s'explica com fer l'unwrap i el bake de textures a Blender 2.8.

“How to animate scroll textures”, de Jimmy Vegas. <https://youtu.be/auVq3TSz20o>. Vídeo que ensenya com generar textures que facin scroll.

“Bones and armature”, de Grant Abbitt. <https://youtu.be/IAiTYaizmY0>. Vídeo molt útil sobre com generar armadures, lligarles a la malla i animar-les mitjançant Blender 2.8.

“Basic IK and foot roll”, de Grant Abbitt. <https://youtu.be/gH5uATTYB4>. Vídeo novament molt útil sobre com emprar l'eina d'inverse kinematics de Blender 2.8.

“Rolling sky all levels”, de Tugan Nguyen. <https://youtu.be/w51dzhrQPXc>. Vídeo amb tots els nivells actuals del joc RollingSky original.