UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

DESERT HOPPER

JOAN VÁZQUEZ GONZÁLEZ

MARTÍ DURAN

JOC 3D VIDEOJOCS

ANTONI CHICA

ÍNDEX DE LA MEMÒRIA

1. ANÀLISI DEL JOC ORIGINAL: CLIFF HOPPER

Cliff Hopper és un joc per a dispositius mòbils que combina elements d'aventura i habilitat. El joc va ser desenvolupat per Mana Cube (una empresa de desenvolupament de videojocs amb seu a França) i llançat el 2017.

L'objectiu principal del joc és arribar al més lluny possible saltant de plataforma a plataforma, on cada plataforma té una longitud diferent i requereix un salt precís per aterrar a la següent. A mesura que avances en el joc, tot es torna més difícil.

Cliff Hopper no té un públic objectiu determinat, oer tant té usuaris de diverses edats i gustos.

No hem trobat informació sobre l'impacte, el nombre de descarregues i altres característiques. Això pot ser degut a que és un joc que ja no Està disponible per descarregar de forma gratuïta a dispositius iOS i Android a través de les respectives botigues d'aplicacions, i per tant, en limita la cerca. A més, IA's com ChatGPT entre d'altres, tampoc han sigut útils per a la cerca d'aquestes dades.

2. DESCRIPCIÓ DEL PROJECTE

Desert Hopper és un projecte basat en Cliff Hopper que s'ha desenvolupat durant unes 5 setmanes amb el motor de videojocs Unity.

2.1. OBJECTIUS

Els objectius d'aquest projectes són: familiaritzar-se amb Unity, el llenguatge de programació C# (C Sharp) així com aprendre a extreure informació contrastada de les diferents fonts. Pel que fa al resultat de Desert Hopper, cal tindre en compte que s'han de complir certs aspectes bàsics que comparteixen tots el videojocs així com aprofitar per crear una versió personalitzada fruit del nostre enginy.

2.2. ANÀLISI GENERAL DEL PROJECTE

Desert Hopper reflexa un problema que ha tingut un arqueòleg en una de les seves recerques. Després d'intentar resoldre el famós enigma de la tomba de Cleòpatra, una bèstia indomable, la guardiana del temple, comença a perseguir-lo amb l'objectiu de matar-lo. Per tant, Desert Hopper consisteix en la fugida pel desert d'un l'arqueòleg que haurà de lluitar i travessar diferents tempestes de sorra, trampes i obstacles mentre recull el tresor de l'antic imperi egipci.

2.2.1. INSTRUCCIONS

El gamePlay del joc consisteix en fugir de l'enemic mitjançant una ruta altament obstaculitzada. El jugador controlarà l'arqueòleg i mitjançant únicament l'ús de la tecla "space" intentarà aconseguir l'objectiu.

Per tant, per poder passar dels menús, has d'utilitzar el ratolí i clicar sobre els elements de pantalla que desitgis. Un cop començat el joc, qualsevol interacció que es vulgui realitzar, s'ha de fer amb la tecla "space" excepte el menú de pausa, en el que s'haurà d'interactuar amb el ratolí.

És a dir, qualsevol interacció de lògica feta en els menús es realitza amb el ratolí, mentre que les que són pròpiament del game play (saltar, girar, doble-salt, etc) es realitzen amb el teclat.

Per saltar i girar, només cal pitjar una vegada la tecla, mentre que per fer el doble salt, en calen dues.

2.2.2. FUNCIONALITATS

En aquesta secció explicarem les funcionalitats de cada script junt a una explicació del perquè i de la lògica darrere de cadascuna d'aquestes.

- AudioManager_Game i AudioManager_Menu

En aquestes classes trobem dues funcions:

private void Awake(): funció cridada per defecte al inicialitzar l'objecte. En aquesta funció es dóna valor al paràmetre AudioSource controlAudio.

public void SeleccioAudio(int index, float volume): en aquesta funció es reprodueix el clip referenciat per el valor de índex amb un volum marcat per volumen.

- script_button

public void changeImage(): aquesta funcionalitat ens permet canviar la textura o imatge del boto al clicar-lo, obtenint-ne així un comportament visual d'aquest fenomen.

- script_control_GS

private void Awake(): funció cridada per defecte al inicialitzar l'objecte. En aquesta funció es dona valor al paràmetres de la classe.

private void LoadData(): ens permet emmagatzemar a memòria el valor del personatge seleccionat per tal d'usar-lo en altres escenes.

public int getSkinID(): retorna l'identificador del personatge

- script_fletxa_loading :

void Update(): amb aquest mètode anem rotant l'objecte fent-ne així un efecte de càrrega de pantalla.

- script_game_menu

private void Awake(): funció cridada per defecte al inicialitzar l'objecte. En aquesta funció es dona valor al paràmetres de la classe.

public void cambiarMenu(): ens permet canviar de escena

public void desplagarMenuPausa(); mètode per desplegar el menú pausa aplicant-ne transicions i efectes de so.

public void tancarMenuPausa(): mètode per tancar el menú pausa aplicant-ne transicions i efectes de so.

public void desplagarMenuSettings():mètode per desplegar el menú de configuració aplicant-ne transicions i efectes de so.

public void tancarMenuSettings(): mètode per tancar el menú de configuració aplicant-ne transicions i efectes de so.

public void desplagarMenuRecord(): mètode per desplegar el menú de rècord aplicant-ne transicions i efectes de so.

public void tancarMenuRecord(): mètode per tancar el menú rècord aplicant-ne transicions i efectes de so.

public void ReadStringInput(string str): mètode per llegir el valor de entrada en un UI component o component gràfic.

public void SaveData(): ens permet emmagatzemar a memòria certes dades seleccionat per tal d'usar-les en altres escenes.

public void RestartLevel(): mètode per reiniciar el nivell

public void ResumLevel(): mètode per continuar el nivell

public void StartLevel(): mètode que descriu la transició inicial

public void tancarMenuGame(): mètode que tanca el menú de joc

public void comprovaResultat(): mètode que ens permet detectar si estem davant d'un nou rècord.

- script_gestio_settings:

private void Awake(): funció cridada per defecte al inicialitzar l'objecte. En aquesta funció es dona valor al paràmetres de la classe.

public void changeAudioState(): mètode per canviar activació de l'àudio.

public void changeEffectState(): mètode per canviar activació efectes de so.

public void SaveData(): ens permet emmagatzemar a memòria certes dades seleccionat per tal d'usar-les en altres escenes.

private void LoadData(): ens permet carregar de memòria certes dades seleccionat per tal d'usar-les.

void Update(): mètode cridat cada frame que ens manté la coherència entre les dades i els components gràfics.

- script_girs

public void Re_Start(): mètode per reiniciar joc

void Start(): funció predefinida cridada en la inicialització de l'objecte. En aquest es defineix el valor inicial dels paràmetres de la classe.

public int getTurns(): retorna el nombre de girs efectuats.

string getTimeFormat(float temps); retorna el temps en el format estàndard.
void Update(): mètode cridat cada frame que ens manté la coherència entre les dades i els components gràfics.

public void SaveData(): ens permet emmagatzemar a memòria certes dades seleccionat per tal d'usar-les en altres escenes.

- script_inicial_message

void Update(): mètode que ens permet veure en cada frame, quin valor té el missatge així com la actualització d'aquest

- script_item

void Start(): funció predefinida cridada en la inicialització de l'objecte. En aquest es defineix el valor inicial dels paràmetres de la classe.

void Update(): es descriu el comportament de l'objecte al qual està associat el script.

- script_coin
-

void Start(): funció predefinida cridada en la inicialització de l'objecte. En aquest es defineix el valor inicial dels paràmetres de la classe.

void Update(): es descriu el comportament de l'objecte al qual està associat el script.

script_loading

void Start(): funció predefinida cridada en la inicialització de l'objecte. En aquest es defineix el valor inicial dels paràmetres de la classe.

IEnumerator MakeTheLoad(): descriu el temps I la execució del canvi d'escena desitjat

- script_menu

private void Awake(): funció cridada per defecte al inicialitzar l'objecte. En aquesta funció es dona valor al paràmetres de la classe.

public void desplegarMenuExtra() : mètode que ens permet desplegar menú mitjançant transicions i efectes de so.

public void desplegarMenuExit() : mètode que ens permet desplegar menú mitjançant transicions i efectes de so.

public void tancarMenuExit(): mètode que ens permet tancar el menú mitjançant transicions i efectes de so.
public void desplegarMenuScore(): mètode que ens permet desplegar menú mitjançant transicions i efectes de so.

public void tancarMenuScore(): mètode que ens permet tancar el menú mitjançant transicions i efectes de so.

public void desplegarMenuSkins(): mètode que ens permet desplegar menú mitjançant transicions i efectes de so.

public void tancarMenuSkins() mètode que ens permet tancar el menú mitjançant transicions i efectes de so.

public void desplegarMenuSettings(): mètode que ens permet desplegar menú mitjançant transicions i efectes de so.

public void tancarMenuSettings():mètode que ens permet tancar el menú mitjançant transicions i efectes de so.

public void desplegarMenuCredits(): mètode que ens permet desplegar menú mitjançant transicions i efectes de so.

public void tancarMenuCredits(): mètode que ens permet tancar el menú mitjançant transicions i efectes de so.

public void desplegarMenuIns(): mètode que ens permet desplegar menú mitjançant transicions i efectes de so.

public void tancarMenuIns(): mètode que ens permet tancar el menú mitjançant transicions i efectes de so.

public void startGame(): mètode definit per a donar delay i establir àudio a l'acció d'entrada al joc.

void escenaLoading(): mètode definit per carregar l'escena Loading

public void exitGame(): mètode definit per tancar aplicació.

- Script_pedra

public void Re_Start() : mètode per reiniciar el enemic.

void Start(): funció predefinida cridada en la inicialització de l'objecte. En aquest es defineix el valor inicial dels paràmetres de la classe.

void OnTriggerEnter(Collider collid): mètode activat quan l'enemic col·lisiona amb certs elements del joc.

void OnTriggerExit(Collider collid): mètode activat quan l'enemic surt de col·lisions amb certs elements del joc.

public bool getDireccio(): mètode que ens retorna la direcció de l'enemic

public void pauseGame()(): mètode que ens permet parar a l'enemic.

void Update(): mètode cridat a cada frame en el que s'actualitza la dirección, el moviment i l'estat de l'enemic.

- script_platform

void Start(): en aquest mètode s'inicialitzen els paràmetres de l'objecte associat.

void Update(): en aquest mètode s'actualitza la posició de l'objecte associat.

- script_roda

void Start(): en aquest mètode s'inicialitzen els paràmetres de l'objecte associat.

void Update(): en aquest mètode s'actualitza la posició de l'objecte associat.

- script_ranking_score

void Awake(): funció predefinida cridada en la inicialització on carreguem i recuperem certs elements.

public void reset(): mètode creat per resetejar el joc

public void newData(): mètode on actualitzem les dades

private void restoreData(): funció on recuperem les dades de memòria

private void saveData(): funció on guardem les dades a memòria.

public void resetData(): mètode creat per resetejar valors del joc

void OnDestroy(): mètode associat a la destrucció de l'objecte en el que es cridarà a altres mètodes.

void Update(): mètode definit per mantindre la coherència entre les dades de memòria i les de pantalla.

- script_shange_skin

void Start(): mètode incialitzador de les dades.

void Update(): mètode que ens manté la actualització i coherència de les dades. En aquest actualitzem el valor de l'objecte, activant-lo.

public void dreta() / void esquerra(): mètodes que ens permeten canviar de selección de personatge.

private void SaveData(): funció on guardem les dades a memòria.

private void LoadData(): funció on carreguem les dades de memòria.

private void OnDestroy(): mètode associat a la destrucció de l'objecte en el que es cridarà a altres mètodes.

- script_sprite_

void Start(): mètode incialitzador de les dades.

void Update(): mètode que ens manté la actualització i coherència de les dades. En aquest actualitzem la textura de l'objecte.

- script_cama_dre, script_cama_esquerra, script_brac_dret, script_brac_esq_

void Update(): mètode que ens manté la actualització i coherència de les dades. En aquest actualitzem la posició, rotació i direcció de l'objecte.

- script_ball

private void Awake(): funció cridada per defecte al inicialitzar l'objecte. En aquesta funció es dona valor al paràmetres de la classe.

void Start(): mètode incialitzador de les dades.

void Update(): mètode que ens manté la actualització i coherència de les dades. En aquest actualitzem la posició i direcció de l'objecte.

- Script_camera_personatge

void Start(): mètode incialitzador de les dades.

public void Re_Start(): mètode que ens permet reiniciar la càmera.

void Update(): mètode que ens manté la actualització i coherència de les dades. En aquest actualitzem la posició i direcció de l'objecte.

- script_cantonada

void Start(): mètode incialitzador de les dades.

public void changeBlock(): mètode que ens permet canviar l'aspecte de l'objecte associat, activant-ne un nou fill.

- script_mapa

public void Re_Start(); mètode que ens permet reiniciar la creació del camí o mapa.

public void pauseGame(): mètode que ens permet pausar la creació del camí o mapa.

void Start(): mètode incialitzador de les dades.

public bool getDireccio(): mètode que ens permet obtindre el valor del paràmetre direcció.

void Update():mètode que ens manté la actualització i coherència de les dades. En aquest actualitzem la posició i direcció de l'objecte, que mitjançant la generació de nombres aleatoris anirà creant certs objectes per tal de formar el mapa amb coherència.

- scriptMoviment1

private void Awake(): funció cridada per defecte al inicialitzar l'objecte. En aquesta funció es dona valor al paràmetres de la classe.

void Start(): mètode inicialitzador de les dades.

public void Re_Start(): mètode que ens permet reiniciar el personatge

void OnTriggerStay(Collider collid): mètode que es llença sempre que el personatge estigui col·lisionant amb algun element del joc.

void OnTriggerEnter(Collider collid): mètode que es llença sempre que el personatge col·lisiona, per primer cop, amb algun element del joc.

public bool isJumping(): mètode que ens retorna el valor del paràmetre jumping.

public int getCoins(): mètode que ens retorna el nombre de monedes recollides.

public int getGirs(): mètode que ens retorna el nombre de girs efectuats.

public float getTime(): mètode que ens retorna el temps de supervivència.

public bool getMode(): mètode que ens retorna el valor associat al paràmetre god_mode del personatge.

public bool getDireccio(): mètode que ens retorna el valor del paràmetre direcció.

public Vector3 getPosition(): mètode que retorna la posició del personatge.

public void pauseGame(): mètode que ens permet pausar al personatge.

void Update(): mètode que ens manté la actualització i coherència de les dades. En aquest actualitzem la posició i direcció de l'objecte, que sempre avançarà en línia recte respecte la seva posició.

3. METODOLOGIA

3.1. CRONOLOGIA DEL PROJECTE

Per poder dur a terme aquest projecte, s'han requerit certes indicacions i petites reunions entre integrants del grup i el propi professor per tal de fer petites valoracions sobre l'enfocament i el desenvolupament d'aquest. La metodologia seguida durant la creació del projecte, ha estat basada en la divisió del treball en diferents àmbits i al futura fusió d'aquests:

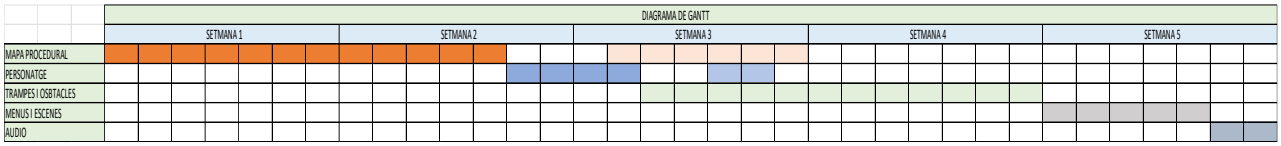

Figura 1 : DIAGRAMA DE GANTT

Les primeres sessions i setmanes de treball foren enfocades en la generació del mapa o camí per al que l'arqueòleg fuig. Inicialment, es van fer dues versions paral·leles: una basada en la generació de blocs totalment aleatoris i l'altre, basada en l'ús de prefabs.

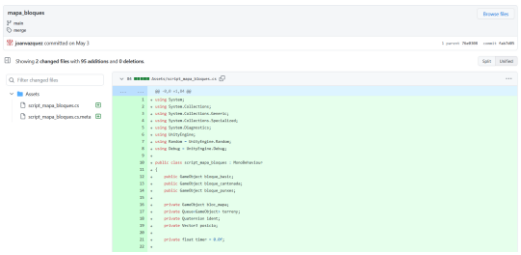

Figura 2: commit generació sense Prefabs

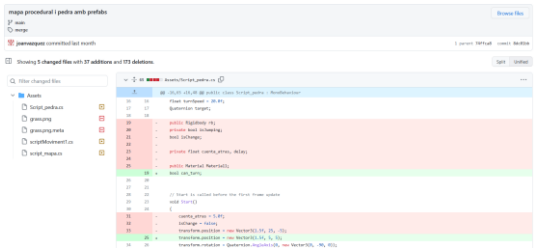

Figura 3: commit generació amb Prefabs

Finalment, desrpès d'una primera valoració per part del docent, vam decidir continuar amb aquesta darrere versió.

Després d'un parell de setmanes, vam acabar amb aquesta generació obtenint-ne com a resultat un mapa totalment procedural generat amb l'ús de diferents prefabs.

Seguidament, el següent aspecte en el que ens vam enfocar va ser en la creació i animació bàsica del personatge. L'ús de MagicanVoxel per a la creació, junt a les skins del joc original, ens van permetre aconseguir diferents models fàcilment programables.

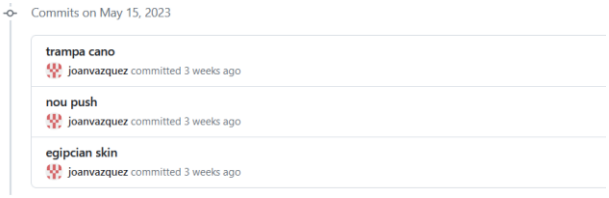

Figura 4: cronología Commits

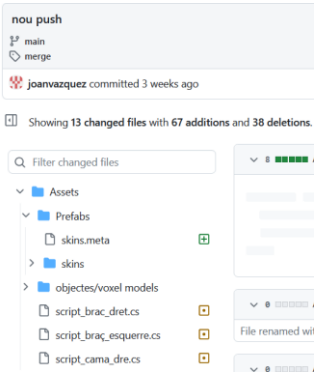

Figura 5: contingut push

Aquest segon item es va realitzar en un període d'entre dos – tres dies de treball.

Els problemes de coherència i la solució d'errors mapa – personatge ha sigut un dels procediments que més temps ens ha comportat. Inicialment el nostre personatge tenia un comportament molt bàsic millorat posteriorment que gràcies a la recerca massiva de informació.

Seguidament, es continua amb la creació amb la introducció de les trampes i/o obstacles del joc. Per fer-les vam utilitzar també el MagicVoxel. En l'enunciat ens demanava un cert nombre mínim de trampes, però nosaltres varem decidir introduir-ne les següents: les punxes, les "arenas movedizas", el canó, la plataforma, els forats i la roda.

Dins d'aquest bloc de treball, trobem la creació de l'enemic. Per obtenir-lo hem descarregat un model 3D totalment gratuït, que posteriorment amb Blender, acabaríem separant-lo en diferents parts per tal de poder exportar-lo a Unity i assignar diferents scripts a certes parts obtenint-ne així la animació desitjada. Cal remarcar la dificultat que ens ha comportat no només la divisió del model, sinó la seva cerca.

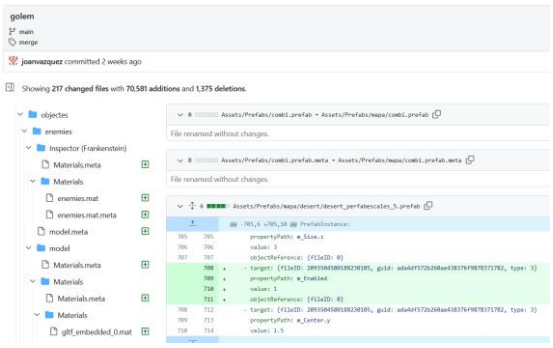

Figura 6: contingut Commit

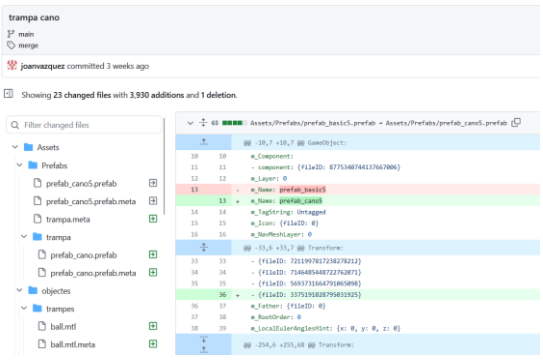

Figura 7: contingut Commit

Finalment, varem dedicar la darrere setmana per a la creació i coordinació dels menús, i la animació mitjançant sons de les diferents interaccions mon-jugador.

El joc està distribuït bàsicament en tres escenes: GameScene, Loading, Menus, que compartiran dades i s'invocaran mitjançant diversos mètodes.

GameScene està formada per tots aquells elements que formen part del gameplay bàsic (personatge, mapa, enemic, obstacles, puntuacions, temps, ...) i per tant, és l'escena on més estona passarà l'usuari.

Pel que fa a Loading, és una escena transitòria que ens permet generar una transició més realista entre els menús i el gameplay.

Respecte Menus, cal dir que és una escena formada bàsicament per un conjunt de panells que simulen les diferents pantalles del menú principal. En aquest podrem fer certes accions que tindran repercussió en el gameplay.

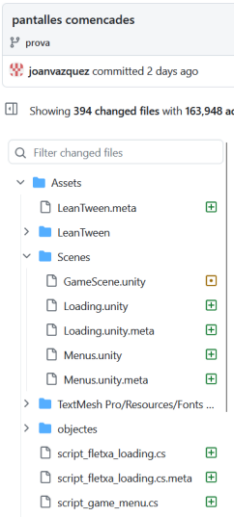

Figura 8: primer Commit

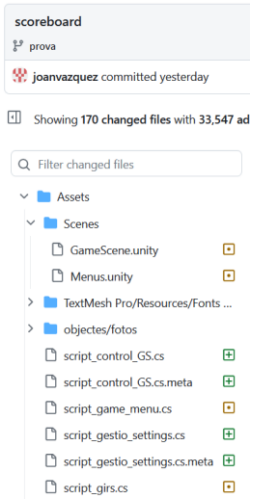

Figura 9 : Commit ScoreBoard

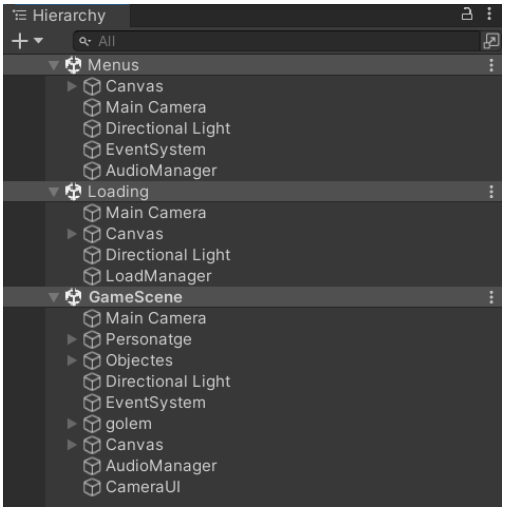

Figura 10: contingut escenes del projecte

La coordinació de dades, s'ha fet mitjançant l'ús de memòria (s'han guardat variables a memòria) i la càrrega de dades durant la càrrega i descàrrega de les escenes.

Pel que fa a l'audio, en cada escena s'ha creat un objecte buit que en serà l'administrador dels sons i de la música. La música ambient ha estat associada a la càmerea mitjançant la creació d'un compnent audio en aquest objecte, mentre que els efectes, seran gestionats per l'objecte buit AudioManager i les crides a les seves funcions.

3.2. REUNIONS SETMANALS

Un factor fonamental per a un millor desenvolupament, han sigut les reunions setmanals dutes a terme entre l'equip de treball i el docent.

En la primera reunió, no teníem el joc començat però per tal de comprovar el rumb que estàvem prenenet, varem explicar quin enfocament teníem pensat donar-l el nostre projecte. La primera resposta fou satisfactòria.

En la següent reunió, ja teníem fetes les dues versions en paral·lel. En aquesta, el docent va recomanar-nos ajuntar les versions i començar a treballar més en equip per tal de poder avançar el projecte i distribuir de la millor manera la càrrega de treball que aquest comporta.

La tercera reunió va ser la que més conclusions vam poder extreure. Bé és cert que en aquesta ja teníem el joc avançat, de fet més del que deia el guió de planificació, i això ens va permetre poder fer una simulació davant del docent i tindre'n una primer crítica. Els aspectes fonamentals extrets foren la millora de coherència entre mapa – personatge (ajustaré velocitats de creació, aparició de items, ...), la implementació millorable de la càmera així com la ja urgent implementació de l'enemic.

En les dues darreres reunions, es va parlar sobretot sobres les millores que calien implementar sobre la execució vista, així com la futura creació, i el futur plantejament de les diferents escenes/dades i de l'àudio, fonamental en tot videojoc.

4. CONCLUSIONS

Aquest projecte ha sigut un repte personal que podriem dir que hem assolit amb èxit. Bé es cert que avui dia, trobar informació de com desenvolupar i crear coses amb Unity és fàcil de trobar gràcies a la amplia documentació del pròpi motor, però això no ens treu mèrit en res.

Per a nosaltres, Desert Hopper és fruit del treball realitzat durant un període de temps d'unes 5 setmanes, on hem vetllat amb molts problemes, hem après com depurar i programar tant amb Unity com amb C#.

Ha sigut un projecte molt enriquidor, que ens ha permès veure una mica més el món dels videojocs, ja que ens hem sentit protagonistes tant en el disseny con en el desenvolupament, veient un resultat final molt gratificant. Per tant, podem concloure que Desert Hopper s'ha patit i molt, però ens ha servit i ha sigut un aprenentatge constant sobre Unity i el potencial que aquest ens brinda.

5. BIBLIOGRAFIA

Pixabay. (s. f.). *Free Death Sound Effects Download - Pixabay*. https://pixabay.com/es/sound-effects/search/death/

"Pàgina utilitzada per a la descàrrega de sons i audio"

YouTube. (s. f.). https://www.youtube.com/?hl=es
"Eina utilitzada per a la descàrrega de sons i audio"

How to disable a game component (audio) - Unity Answers. (s. f.). https://answers.unity.com/questions/1152459/how-to-disable-a-game-component-audio.html
"Fòrum utilitzat com a font de consulta d'errors"

DédaloLab. (2021, 24 junio). *AUDIO en UNITY* 🔊 ▶ *TUTORIAL en ESPAÑOL* ✔ [Vídeo]. YouTube. https://www.youtube.com/watch?v=z6rY9sYx7Oo
"Font de información per a la creació de l'audio"

Aqsa Nadeem. (2021, 29 marzo). *Confetti VFX Particle System| How to make a Confetti in unity using Particle System VFX* [Vídeo]. YouTube. https://www.youtube.com/watch?v=9SNgs7Xv6M8
"Font d'informació per a la creació del sistema de particules"

LeanTween - LeanTween. (s. f.). https://dentedpixel.com/LeanTweenDocumentation/classes/LeanTween.html
"Font d'informació per a les transicions de menus"

C# - Modify an array in place, without creating another array in memory. (s. f.). Stack Overflow. https://stackoverflow.com/questions/50033837/c-sharp-modify-an-array-in-place-without-creating-another-array-in-memory
"Forum consultor de informació"

GameDevTraum. (2020, 1 agosto). *Cómo pasar DATOS ENTRE ESCENAS en Unity usando PlayerPrefs* [Vídeo]. YouTube. https://www.youtube.com/watch?v=5D7rxEcaJQ4
"Font d'informació per a comunicación dades"

Technologies, U. (s. f.). *Unity - Scripting API:* https://docs.unity3d.com/2019.1/Documentation/ScriptReference/
"Manual official de Unity"

Ice Age Font | dafont.com. (s. f.). https://www.dafont.com/ice-age-font.font
"Font de extraccio de fonts"