**TREBALL DE FI DE GRAU**

**ESCRIVIU EL TÍTOL DEL TREBALL EN MAJÚSCULES I NEGRETA**

**Alexandre Joan Morro Santandreu**

**Grau d’Enginyeria Informàtica**

**Escola Politècnica Superior**

**Any acadèmic 2024-25**

**ESCRIVIU EL TÍTOL DEL TREBALL EN MAJÚSCULES I NEGRETA**

**Alexandre Joan Morro Santandreu**

**Treball de Fi de Grau**

**Escola Politècnica Superior**

**Universitat de les Illes Balears**

**Any acadèmic 2024-25**

Paraules clau del treball:

paraula1, paraula2, paraula3...

*Nom del tutor / la tutora del treball*

*Nom del tutor / la tutora del treball*

# Agraïments

Aquesta secció és opcional. Si l’eliminau, feis que el nombre de pàgina de la taula de continguts comenci a “i” (si tampoc posau dedicatòria) o a “iii” (si posau dedicatòria).

*Això és una dedicatòria, que és opcional.*

# Taula de continguts

[1 Agraïments i](#_Toc198998601)

[2 Taula de continguts v](#_Toc198998602)

[3 Índex de figures ix](#_Toc198998603)

[4 Índex de taules xi](#_Toc198998604)

[5 Nomenclatura xiii](#_Toc198998605)

[6 Resum xv](#_Toc198998606)

[Capítol 1. Introducció 1](#_Toc198998607)

[1.1. Contextualització 1](#_Toc198998608)

[1.2. Adaptació 1](#_Toc198998609)

[1.3. Objectius 2](#_Toc198998610)

[1.4. Abast del projecte 2](#_Toc198998611)

[1.5. Seccions 3](#_Toc198998612)

[Capítol 2. Estat de l’art 5](#_Toc198998613)

[2.1. Jocs similars existents 5](#_Toc198998614)

[2.1.1. Royal Revolt! 5](#_Toc198998615)

[2.1.2. Bloons TD 6 6](#_Toc198998616)

[2.1.3. Temple run 6](#_Toc198998617)

[2.1.4. The Lab – Longbow 7](#_Toc198998618)

[2.2. Característiques interessants 8](#_Toc198998619)

[2.2.1. Característiques escollides 9](#_Toc198998620)

[Capítol 3. Desenvolupament del projecte 11](#_Toc198998621)

[3.1. Gestió del projecte 11](#_Toc198998622)

[3.1.1. Metodologia de desenvolupament 11](#_Toc198998623)

[3.1.2. Planificació temporal 11](#_Toc198998624)

[3.2. Anàlisi 11](#_Toc198998625)

[3.2.1. Usuaris 12](#_Toc198998626)

[3.2.2. Requisits funcionals 12](#_Toc198998627)

[3.2.2.1. Requisits funcionals d’usuari 12](#_Toc198998628)

[3.2.2.2. Requisits funcionals de sistema 12](#_Toc198998629)

[3.2.3. Requisits no funcionals 13](#_Toc198998630)

[3.3. Disseny 13](#_Toc198998631)

[3.3.1. Eines 13](#_Toc198998632)

[3.3.1.1. A-Frame 13](#_Toc198998633)

[3.3.1.2. Three.js 14](#_Toc198998634)

[3.3.1.3. Objectes i models 3D o animacions 14](#_Toc198998635)

[3.3.1.4. Blender [5] 14](#_Toc198998636)

[3.3.1.5. GitHub [6] 14](#_Toc198998637)

[3.3.2. Disseny gràfic 14](#_Toc198998638)

[3.3.2.1. Ambientació i il·luminació 14](#_Toc198998639)

[3.3.2.2. Estil gràfic 15](#_Toc198998640)

[3.3.2.3. Interfície d’usuari 15](#_Toc198998641)

[3.3.2.4. Enemics 15](#_Toc198998642)

[3.3.3. Cicle jugable 16](#_Toc198998643)

[3.3.4. Model de dades 17](#_Toc198998644)

[3.3.5. Disseny arquitectònic 18](#_Toc198998645)

[3.4. Implementació 19](#_Toc198998646)

[3.4.1. Servidor 19](#_Toc198998647)

[3.4.1.1. Codi del servidor 19](#_Toc198998648)

[3.4.1.2. Variables d’entorn 20](#_Toc198998649)

[3.4.1.3. Accés a la base de dades 21](#_Toc198998650)

[3.4.2. Base de dades 22](#_Toc198998651)

[3.4.3. Client 23](#_Toc198998652)

[3.4.3.1. A-Frame 23](#_Toc198998653)

[3.4.3.2. Components 23](#_Toc198998654)

[3.4.3.2.1. Arc 24](#_Toc198998655)

[3.4.3.2.2. Corda 25](#_Toc198998656)

[3.4.3.2.3. Fletxa 26](#_Toc198998657)

[3.4.3.2.4. Planta 29](#_Toc198998658)

[3.4.3.2.5. Vagoneta 29](#_Toc198998659)

[3.4.3.2.6. Terra 29](#_Toc198998660)

[3.4.3.2.7. 29](#_Toc198998661)

[3.4.3.3. Diana 29](#_Toc198998662)

[Capítol 4. 32](#_Toc198998663)

[7 Referències 33](#_Toc198998664)

[Apèndix A. 35](#_Toc198998665)

[A.1. Fonts de l’especificació de l’arc 35](#_Toc198998666)

[Apèndix B. 37](#_Toc198998667)

[B.1. Arc 37](#_Toc198998668)

[B.2. Fletxa 37](#_Toc198998669)

[B.3. Planta 37](#_Toc198998670)

[B.4. Vagoneta 37](#_Toc198998671)

[B.5. Arbres 37](#_Toc198998672)

[B.6. Diana 37](#_Toc198998673)

[B.7. Cel 38](#_Toc198998674)

[B.8. So partida perduda 38](#_Toc198998675)

[B.9. So arrancar fruita 38](#_Toc198998676)

[B.10. So perdre vida 38](#_Toc198998677)

[B.11. So botó pressionat 38](#_Toc198998678)

[B.12. So fletxa disparada 38](#_Toc198998679)

# Índex de figures

[Figura 2.1 Royal Revolt! 19](#_Toc197188806)

[Figura 2.2 Bloons TD 6 20](#_Toc197188807)

[Figura 2.3 Temple run 21](#_Toc197188808)

[Figura 2.4 The Lab - Longbow 22](#_Toc197188809)

[Figura 3.1 Planta alienígena 28](#_Toc197188810)

[Figura 3.2 Cicle de jugabilitat 29](#_Toc197188811)

# Índex de taules

[Taula 2.1 Característiques dels diferents jocs 23](#_Toc179648250)

[Taula B.1. Format dels encapçalaments i peus de pàgina de les seccions del document. 36](#_Toc179648251)

[Taula B.2. Format dels encapçalaments i peus de pàgina de les seccions del document. 37](#_Toc179648252)

[Taula B.3. Colors dels títols segons els estudis. 38](#_Toc179648253)

# Nomenclatura

|  |  |
| --- | --- |
| **LTIM** | Laboratori de Tecnologies de la Informació Multimèdia. |
| **UIB** | Universitat de les Illes Balears. |
| **HTML** | Hypertext Markup Language. |
| **Xatbot** | Programa que simula mantenir una conversa amb un ésser humà. |
| **VR** | Realitat virtual (Virtual Reality) |

# Resum

Aquest treball de fi de grau té com a objectiu el desenvolupament d’un joc en realitat virtual se centra en una experiència immersiva on els jugadors utilitzen un arc per a disparar a diversos enemics mentre segueixen un camí. Aquest enfocament combina la precisió i l’habilitat del tir amb arc amb l’emoció d’enfrontar-se a enemics en un entorn virtual. A mesura que els jugadors avancen pel camí, es troben amb diferents tipus d’enemics, cadascun amb les seves pròpies característiques i nivells de dificultat, la qual cosa afegeix varietat i desafiament al joc.  
La utilització de diferents enemics assegura que els jugadors es mantinguin en moviment constant, la qual cosa augmenta la intensitat i la dinàmica del joc. La combinació de la tecnologia de realitat virtual amb el tir amb arc ofereix una experiència única i atractiva, que pot atreure tant als entusiastes dels videojocs com a aquells interessats en l’esport del tir amb arc.

El joc permet als jugadors interactuar amb un arc i unes fletxes per practicar el tir amb arc i entretenir-se disparant diferents enemics utilitzant les oportunitats que ofereix la realitat virtual.

El desenvolupament del projecte s’ha basat en una metodologia iterativa, amb una planificació detallada i una anàlisi exhaustiva dels usuaris i els seus requisits. La implementació del joc ha implicat l’ús de tecnologies com HTML, JavaScript, MySQL i Node.js, així com la implementació de models 3D i entorns interactius amb A-Frame.

El projecte s’ha realitzat amb èxit, i s’ha presentat de manera accessible per a tothom al ciència per a tothom de la UIB. S’ha aplicat un qüestionari SUS per avaluar la usabilitat del joc, i els resultats obtinguts indiquen una bona acceptació pel que fa a la seva usabilitat, amb diverses àrees de millora identificades per a futures versions.

# Introducció

En aquest capítol es dona una visió general del projecte explicant el context d’aquest Treball de Fi de Grau, els motius personals que han dut a realitzar-lo, els objectius que es volen aconseguir, l’abast del projecte i l’estructura del document.

## Contextualització

La Realitat Virtual (RV) ha generat molt d’interès recentment gràcies a l’aparició de multitud de dispositius per reproduir-la, com poden ser les Oculus Rift, Meta Quest, HTC Vive, Samsung Gear, Google Daydream, etc. Fins i tot han aparegut dispositius per a ser fets servir amb mòbils, arribant a les econòmiques Google Cardboard, que estan fetes de cartró.

El joc que es desenvoluparà en aquest TFG serà un joc d’atac a la torre invers, és una variant dels jocs del tipus *tower defense* [1], és a dir un joc a on enemics van avançant per destruir la torre que s’ha de defensar, en el qual en lloc de col·locar i millorar torres o estructures per evitar que les diferents ones d’enemics arribin a la base, el jugador assumeix el rol de l’atacant.

En resum, aquest TFG oferirà una visió sobre com la realitat virtual amb combinació a les tecnologies web, poden crear noves formes d’entreteniment en l’àmbit digital.

## Adaptació

~~El gènere de jocs de~~ *~~tower defense~~* ~~m’ha agradat des de que era petit. Em pareixen jocs molt entretinguts i a la vegada el suficientment pausats per a poder desconnectar, junt amb desafiament que suposa que a mesura que van passant les rondes es fa de cada vegada més difícil.~~

El gènere de jocs de *tower defense* solen ser una bona proposta com a joc, ja que són ambdós senzills d’aprendre i desafiants, però a l’hora de traslladar aquest tipus de jocs a un entorn de realitat virtual resulta un aprofitament molt pobre de les possibilitats que habiliten les ulleres de VR, degut a que funcionen molt bé en un tauler 2D.

En vers de fer un *tower defense* habitual es pot fer a la inversa, és a dir, el protagonista és el que ha d’anar superant les defenses. Aquesta proposta pot aprofitar millor les possibilitats de les ulleres VR, ja que es pot fer un joc molt més interactiu perquè pot ser en primera persona i la utilització d’armes de projectils, com un arc fa que s’aprofitin bé les possibilitats dels controls de les mans.

## Objectius

L’objectiu principal és desenvolupar un joc en primera persona a on el jugador avança de manera automàtica per un camí i s’enfronta a diferents tipus d’enemics els quals haurà d’eliminar. A mesura que el jugador progressi, més enemics apareixeran i més difícils d’eliminar seran. El joc no té final com a tal sinó que el jugador ha de sobreviure el màxim temps possible. El joc s’acaba quan al jugador no li quedin vides.

El joc està pensat per ser jugat amb equipament de VR, però també hauria de ser accessible a través d’una interfície per a ordinador convencional amb teclat i ratolí.

De cada partida es recopilarà informació per poder elaborar una taula de puntuacions que mostri el top 10 millors jugadors. Aquesta taula ajudarà a crear una competició entre els jugadors per estar-hi de la mateixa manera que en els marcadors de millors jugadors en jocs competitius o el menor temps assolit per pilots en un circuit.

Finalment, el darrer objectiu és tenir una versió jugable per exposar a la fira Ciència per a tothom que organitza la UIB. Per complir aquest objectiu haurem de dur una gestió del calendari adequada.

L’objectiu principal del joc consisteix en avançar per un camí infinit generat aleatòriament i anar superant les defenses enemigues, a l’estil *endless runner* [2], és a dir un joc de córrer sense fi, però amb l’afegit d’anar disparant als enemics. A mesura que el jugador avanci i elimini més enemics, la dificultat del joc augmentarà progressivament, fent que cada secció sigui més desafiant que l’anterior.

La puntuació principal del joc es mesurarà amb el màxim nombre de metres sobreviscuts, demostrant habilitat i estratègia en un entorn virtual cada vegada més hostil.

## Abast del projecte

El joc consistirà sobretot en la demostració del que es podria fer amb un equip de realitat virtual més que un joc amb característiques especials, sobretot per les limitacions de personal i coneixements en la creació de videojocs, sinó més aviat una prova del que es pot fer amb la realitat virtual junt amb diferents tecnologies web. Per tant, moltes característiques presents en jocs similars, no estaran presents o se tractaran de manera més superficial en aquest projecte. Especialment en l’àmbit del modelatge i animacions 3D.

Tampoc s’espera una gran varietat de models 3D. Ni en l’àmbit d’armament ni en el dels enemics i defenses. Una altra vegada, això és deu al desconeixement de tècniques de modelatge i a què aquest projecte tampoc té com a objectiu oferir una gran varietat. Així com tampoc hi haurà destrucció de l’entorn quan el jugador dispari a les parets de l’escenari.

La trajectòria del tir es veurà afectada per la gravetat, ja que com es dispara amb un arc té sentit i és natural que la fletxa vagi caient segons va recorrent metres. Encara que no hi haurà dispersió, perquè no se cerca una simulació completa de les físiques i molts jugadors ho trobarien frustrant.

Els enemics quedaran estàtics a la posició a on apareguin o tinguin assignada, ja que implementar un algorisme de *pathfinding* o similar no és l’objectiu del projecte, i la dificultat del joc resideix sobretot en l’ús d’un arc i no s’afegirà dificultat fent que els enemics es moguin.

S’elaboraran dues versions. Una per PC i un altre per equipament de RV. Ara bé, dispositius mòbils, tauletes, etc. no tindran suport.

No es té intenció d’implementar un mode multijugador en línia que permeti a diversos jugadors jugar una mateixa partida de forma cooperativa.

## Seccions

Aquest document es troba estructurat per capítols. El segon capítol és *Estat de l’art* on s’exposen i descriuen tècniques i solucions ja existents a problemes similars que poden resultar útils. El tercer capítol és *Desenvolupament del projecte* format per tots els apartats de gestió, anàlisi, disseny, implementació i instal·lació. El quart capítol és *Resultats* on es comenten els productes obtinguts. El cinquè i darrer capítol *Conclusions* conté les diferents conclusions que s’han obtingut i possibles millores per al futur.

# Estat de l’art

Aquest capítol exposa i descriu les tècniques i solucions existents que poden resultar d’utilitat a l’hora de desenvolupar aquest projecte.

Abans d’entrar en el procés de desenvolupament, podem observar què han fet altres amb l’objectiu d’aprendre d’ells. Què fan bé? Quines coses podem evitar?

## Jocs similars existents

A continuació es descriuen un conjunt de jocs amb característiques similars al joc que volem desenvolupar.

### Royal Revolt!

Royal Revolt![[1]](#footnote-2) és un joc de *tower defense* invers en el qual el protagonista és un príncep que torna al seu regne, però descobreix que el tron del seu pare ha estat usurpat.

La missió principal del joc és liderar soldats per recuperar el regne. Per a aconseguir això, s’han de liderar a les diferents tropes a través d’una sèrie de nivells, cadascun amb un castell enemic que s’ha de conquistar.

A mesura que s’avança s’aconsegueixen diners per millorar tant el príncep com les diferents tropes i màgies que es poden invocar.

Imatge que conté Joc d’ordinador, Software de videojocs, Videojoc d’estratègia, captura de pantalla

Descripció generada automàticament

Figura 2.1 Royal Revolt!

En la Figura 2.1 podem veure al príncep junt amb les seves tropes avançant pel territori hostil. Al cantó inferior esquerre es veuen els icones de les diferents tropes que es poden cridar i al cantó inferior dret les 2 màgies que es poden invocar.

En resum, el joc combina estratègia i acció, requerint l’ús tant de la força bruta com de la planificació tàctica per a avançar.

### Bloons TD 6

Bloons TD 6[[2]](#footnote-3) és un *tower defense* el qual l’objectiu és evitar que els globus arribin al final del camí. Per a això, s’han de col·locar estratègicament torres de micos i herois perquè destrueixin els globus abans que arribin al seu destí.

Imatge que conté text, captura de pantalla, Programari de gràfics, Dibuixos animats

Descripció generada automàticament

Figura 2.2 Bloons TD 6

Cada mico té una arma diferent i funciona especialment bé sobre un tipus de globus. La idea és cobrir tots els tipus de globus amb el menor nombre de micos i anar-los millorant perquè siguin més eficaços. A la Figura 2.2 es poden veure els diferents tipus de micos i globus, com es pot apreciar, els globus tenen diferents colors i formes, el color indica la quantitat de capes que li queden abans d’explotar i la forma indica el tipus.

### Temple run

Temple run[[3]](#footnote-4) és un *endless runner*, que ràpidament es va convertir en un èxit degut a la seva jugabilitat addictiva i senzilla.

L’objectiu és recórrer el màxim nombre de metres evitant obstacles, a mesura que es recorren metres, de manera progressiva, el personatge corr de cada vegada més ràpid i s’afegeixen més diferències en els obstacles, necessitant de cada vegada una reacció més ràpida. Si comets una errada el personatge corr més lent i permet que se li apropin els enemics, si comets una altra errada en poc temps aconsegueixen atrapar-te.



Figura 2.3 Temple run

A la Figura 2.3 es poden veure les criatures malignes que segueixen al personatge principal, qualque obstacle i diferents biomes que es recorren.

### The Lab – Longbow

The Lab[[4]](#footnote-5) és un videojoc de realitat virtual desenvolupat per Valve. El joc ofereix vuit tipus de minijocs diferents que impliquen experiències de demostració curtes que utilitzen diferents aspectes de les capacitats de la realitat virtual. La varietat també s’ofereix més enllà de les mateixes experiències per la quantitat d’interacció amb objectes de l’entorn que s’inclou.

Més específicament la demostració anomenada *Longbow* un joc basat en tir amb arc. Els dos controladors de moviment de VR s’utilitzen per apuntar i disparar un arc, amb un controlador que actua com l’eix d’arc i l’altre com la fletxa. El jugador té la tasca de defensar una porta del castell contra una força invasora mentre està sobre la murada del castell.

La característica especial d’aquest joc és la interacció amb l’arc i les fletxes a través dels comandaments de VR.



Figura 2.4 The Lab - Longbow

A la Figura 2.4 es poden veure els enemics a la part superior de la imatge i la porta del castell a l’esquerra a part de diferents elements distribuïts per l’entorn que donen punts extra o activen trampes.

## Característiques interessants

Després d’haver vist diferents jocs que assoleixen una part del que es vol fer, ara es valorarà que fan bé i encaixaria amb el joc que es vol fer.

La Taula 2.1 mostra el joc d’origen les característiques i una valoració de si encaixaria en el nostre joc.

|  |  |  |
| --- | --- | --- |
| Joc o jocs | Característica | Valoració |
| Royal Revolt! | Anar millorant el personatge a mida que es puja de nivell. | No |
| Royal Revolt! | Anar invocant tropes mentre s’avança dins un nivell. | No |
| Royal Revolt!  i  Bloons TD 6 | Utilització de màgies o habilitats especials. | No |
| Bloons TD 6 | Un tipus de mico és especialment fort contra cert tipus de globus. | Si |
| Temple run | El jugador avança pel mapa fins que comet 2 errades. | Si |
| Temple run | El jugador és perseguit per un enemic que limita les errades que pot fer. | No |
| The Lab - Longbow | Utilització d’un arc amb els comandaments de VR. | Si |
| The Lab - Longbow | Elements distribuïts pel mapa que donen punts extra. | Si |
| Bloons TD 6  i  The Lab - Longbow | Utilització del mateix enemic, però amb modificacions que li donen més vida o armadura. | Si |

Taula 2.1 Característiques dels diferents jocs

### Característiques escollides

Finalment, tenim un conjunt de característiques que sabem que fan bé altres jocs, ara toca adaptar-les al nostre.

* Un tipus de mico és especialment fort contra cert tipus de globus: això es podria adaptar posant diferents tipus de fletxes com fletxes explosives o de foc.
* El jugador avança pel mapa fins que comet 2 errades: Es podria posar un sistema de vida en el qual si els enemics superen una certa distància, en perds una.
* Utilització d’un arc amb els comandaments de VR: es pot implementar de la mateixa manera o molt similar.
* Elements distribuïts pel mapa que donen punts extra: es pot implementar de la mateixa manera.
* Utilització del mateix enemic, però amb modificacions que li donen més vida o armadura: es pot implementar el fet que els enemics tinguin més vida a mida que es puja de ronda.

# Desenvolupament del projecte

Asd

## Gestió del projecte

Aquest apartat descriu la metodologia de desenvolupament que s’ha seguit junt amb la planificació temporal.

### Metodologia de desenvolupament

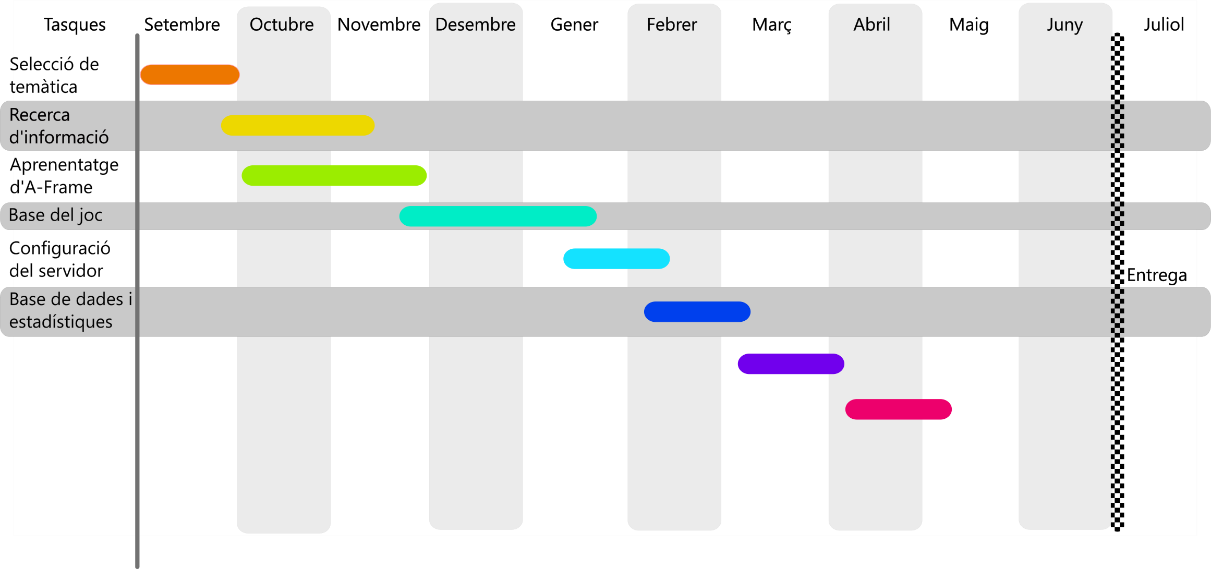
En el desenvolupament d’aquest projecte s’ha utilitzat la metodologia de prototip evolutiu. Aquest mètode consisteix en crear un prototipus inicial el qual va evolucionant de manera incremental afegint funcionalitats, el producte final és el prototipus inicial amb totes les funcionalitats requerides.

Aquesta metodologia és particularment efectiva en el desenvolupament àgil a on la retroalimentació i les iteracions són parts clau, a més de ser altament adaptable gràcies al desenvolupament incremental sobre el qual està basat.

### Planificació temporal

A la figura es pot veure la planificació del projecte que està distribuïda entre les diferents funcionalitats a grans trets.

Afegir figura



## Anàlisi

En aquest apartat es detallaran els requisits tant funcionals com no funcionals i l’usuari que interactuarà amb el joc.

### Usuaris

El joc compta amb un únic tipus d’usuari, el jugador, el qual interactuarà amb l’arc i els diferents elements del joc. Durant el joc, el jugador podrà agafar l’arc i anar disparant fletxes utilitzant els controls de realitat virtual. A més, tindrà la possibilitat de guardar la puntuació dins la base de dades.

### Requisits funcionals

Fsd

#### Requisits funcionals d’usuari

* RFU-01 · Agafar un arc
  + El jugador ha de poder agafar i utilitzar un arc.
* RFU-02 · Moviment
  + El jugador s’ha de poder moure lliurement per la plataforma però no pot sortir.
* RFU-03 · Disparar fletxa
  + El jugador ha de poder disparar fletxes en qualsevol direcció fent ús de l’arc.
* RFU-04 · Modes de joc
  + Abans de començar la partida, l’usuari ha de poder elegir entre el mode normal i un mode de pràctica.
* RFU-05 · Puntuació
  + El jugador ha de poder veure la vida restant i puntuació actual mentre juga.
* RFU-06 · Marcador
  + El jugador ha de poder veure un marcador amb les 10 millors puntuacions obtingudes pels jugadors.

#### Requisits funcionals de sistema

* RFS-01 · Recorregut del camí
  + El camí es recorrerà damunt una plataforma que es mourà de manera automàtica un cop engegada.
* RFS-02 · Sons
  + Han d’existir efectes de so que es reproduiran quan es realitzin certes accions com disparar, ferir un botó o un enemic...
* RFS-03 · Puntuacions
  + El joc ha de guardar la puntuació obtinguda si és major a l’anterior a cada punt de control i quan es quedi sense vides.
* RFS-04 · Generació del camí
  + El camí a recórrer es generarà dinàmicament mentre l’usuari vagi avançant.
* RFS-05 · Descans
  + El jugador ha de tenir un punt de control com a mínim cada 90 segons per poder descansar degut al cansament dels braços.

### Requisits no funcionals

* RNF-01 · Immersió
  + S’ha de poder simular el tensament de la corda amb comandaments de VR, podent modificar la força aplicada segons la separació de les mans. A més a més, el comportament de la fletxa ha de seguir la trajectòria d’un projectil com si fos a la Terra, és a dir tenint en compte la gravetat terrestre.
* RNF-02 · Compatibilitat
  + El joc ha de ser compatible amb la majoria de dispositius de realitat virtual, però com a mínim amb l’Oculus i el Meta Quest.
* RNF-03 · Fluïdesa
  + El joc ha d’anar com a mínim a 30 fotogrames per segon per tenir una sensació agradable i evitar marejos.
* RNF-04 · Integritat
  + La puntuació de cada usuari s’ha de guardar dins una base de dades i només pot canviar la seva una vegada ha iniciat sessió.
* RNF-05 · Compatibilitat
  + El joc serà compatible amb tots els navegadors que suportin la API de WebXR.
* RNF-06 · Ús de dades
  + Només es recopilaran les següents dades: nom, edat, sexe, experiència prèvia en realitat virtual, nombre de disparaments, nombre d’encerts, puntuació i ronda. Aquestes dades només s’usaran amb fins estadístics.

## Disseny

En aquest apartat, s’explica el disseny gràfic i l’atmosfera escollida, l’estil de la interfície d’usuari, el programari utilitzat, el disseny de la jugabilitat i el model de dades usat per a la realització del joc.

### Eines

En aquest apartat, s’exploren diverses eines que s’han emprat per a la realització del projecte.

#### A-Frame

A-Frame [3] és un *framework* web dissenyat per crear experiències de realitat virtual (VR). Està construït sobre HTML i JS, el que facilita la creació d’escenes virtuals. No obstant això, A-Frame va més enllà de ser simplement un gràfic d’escena 3D o un llenguatge de marques; en el seu nucli, presenta un sistema sòlid d’entitats-components que ofereix una arquitectura declarativa, flexible i modular per a *Three.js [4]*.

#### Three.js

*Three.js [4]* és una biblioteca de JavaScript que permet crear i renderitzar gràfics 3D en un navegador web utilitzant WebGL. Proporciona una sèrie d’eines i funcionalitats que faciliten el desenvolupament d’escenes 3D, incloent-hi la creació de geometries, materials, llums, ombres i animacions. A més a més, com A-Frame utilitza *Three.js [4]* internament, es pot utilitzar la biblioteca de manera directa per realitzar funcions a més baix nivell.

#### Objectes i models 3D o animacions

Com que no es tracta d’un projecte de disseny 3D i que tampoc es tenen els coneixements ni el temps per crear els diferents models i objectes i animar-los en cas que sigui necessari, es farà ús de contingut generat per tercers amb llicències que ens permetin utilitzar-los per a la realització d’aquest projecte.

#### Blender [5]

Blender és un programari de codi obert i gratuït utilitzat per a la creació de gràfics en 3D. Ofereix una àmplia gamma d’eines i funcionalitats que permeten als usuaris modelar, esculpir, texturar, il·luminar i renderitzar objectes i escenes en tres dimensions. Aquest programa es farà servir per a la modificació i creació d’animacions sobre els models 3D definits en el punt anterior.

#### GitHub [6]

Git és un sistema de control de versions distribuït que permet als desenvolupadors gestionar i realitzar un seguiment dels canvis en el codi font d’un projecte al llarg del temps. S’ha decidit utilitzar la plataforma de GitHub (que fa servir Git), ja que permet proporcionar un historial detallat de les modificacions realitzades, la possibilitat de revertir canvis i la utilització de branques, la qual cosa possibilita el desenvolupament de noves característiques o correcció d’errors de manera aïllada.

### Disseny gràfic

En aquest apartat, analitzarem com aquests elements visuals no sols contribueixen a l’estètica del joc, sinó que també influeixen en la jugabilitat i en l’experiència general de l’usuari.

#### Ambientació i il·luminació

L’ambientació i la il·luminació són elements crucials que determinen l’atmosfera i l’experiència del jugador. L’elecció d’una il·luminació brillant i natural és fonamental per a establir un ambient positiu, i per això s’ha optat per simular un dia clar amb molta llum solar.

Quant a la disposició dels elements en l’escena, un disseny net i organitzat, amb un enfocament en la simplicitat, ajuda a minimitzar distraccions i permet que el jugador es pugui concentrar fàcilment.

#### Estil gràfic

S’ha optat per l’estil *low poly* [7] en el disseny gràfic a causa de la seva notable eficiència de rendiment, especialment considerant l’alt consum de recursos que implica la realitat virtual. Aquest enfocament és particularment rellevant quan s’utilitzen ulleres de realitat virtual com les Meta Quest, les quals fan servir processadors ARM similars als d’un dispositiu mòbil. A més, l’ús de tecnologies web actuals impedeix aprofitar al màxim el potencial del maquinari disponible, la qual cosa fa que la reducció de la complexitat dels models i textures sigui encara més beneficiosa. Gràcies a adoptar aquest estil, s’aconsegueix una experiència més fluida i accessible, que permet que els usuaris gaudeixin d’entorns immersius sense comprometre la qualitat visual ni l’estabilitat del sistema.

#### Interfície d’usuari

En el disseny d’interfícies d’usuari per a experiències en realitat virtual, és fonamental evitar l’ús d’elements fixos pel fet que a moltes persones els hi provoca mareigs o qualque tipus de malestar. A més a més, poden limitar la immersió del jugador. En lloc d’això, és més convenient aprofitar l’entorn virtual per a integrar la interfície de manera dinàmica i contextual. En utilitzar objectes i espais de l’entorn com a punts d’interacció, es crea una experiència més natural.

Per tot això, s’ha optat per adoptar els mètodes d’interacció amb l’usuari directament utilitzant l’arc i les fletxes per disparar als botons. A més d’implementar la informació de la vida i la puntuació directament sobre la vagoneta sobre la qual existirà el jugador durant tota la sessió.

#### Enemics

Atès que el jugador ja es desplaça activament pel fet que està posicionat sobre una plataforma, és crucial que els enemics romanguin estàtics per a no complicar la mecànica de disparar amb l’arc. Això és especialment important en un joc dissenyat per a tots els públics, on l’accessibilitat i la diversió són primordials. Per aquesta raó, s’ha optat per introduir una espècie de planta alienígena que genera fruites en el seu extrem com es pot veure en la Figura 3.1. Els jugadors hauran d’apuntar i disparar a aquestes fruites.



Figura 3.1 Planta alienígena

### Cicle jugable

Una vegada el jugador iniciï el joc, seguirà una sèrie de passes per començar una partida fins a acabar-la.

Per començar, el jugador no podrà iniciar la partida fins que s’hagi identificat amb el seu nom i l’edat. Una vegada hagi obert sessió, podrà agafar l’arc i se li mostrarà el menú per començar la partida o jugar en el mode de pràctica.

El mode de pràctica és simplement un entorn a on van sortint dianes a diferents altures a les quals el jugador haurà d’encertar. Aquest mode serveix principalment per familiaritzar-se amb l’ús de l’arc i la caiguda de la fletxa segons la força aplicada.

El mode normal consisteix a disparar a la fruita de la planta mencionada anteriorment per aconseguir punts i evitar perdre vida si el jugador s’allunya molt d’aquesta. Aquest cicle és infinit fins que el jugador es queda sense vides, una vegada el jugador es quedi sense, es guardaran en la base de dades les dades obtingudes durant la partida, establertes en el punt 3.3.4 Model de dades, i se li donarà l’opció d’iniciar una nova partida. Si així ho decideix, el cicle tornarà a començar.

En la Figura 3.2 es mostra el diagrama de flux del cicle de jugabilitat.

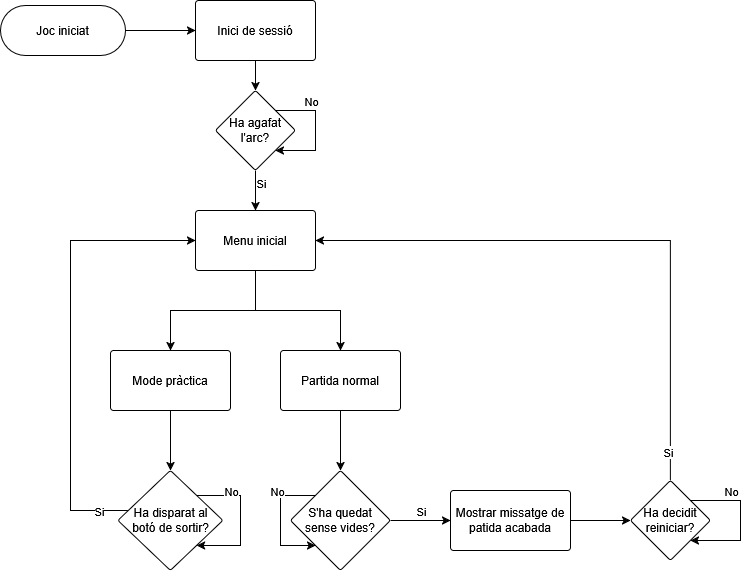


Figura 3.2 Cicle de jugabilitat

### Model de dades

Atès que la quantitat de dades a emmagatzemar és limitada, el model de dades es presenta de manera senzilla i eficient. Aquest model es compon d’una única taula que agrupa tota la informació rellevant, la qual cosa facilita la seva gestió i accés. En centralitzar les dades en una sola estructura, s’optimitzen tant el rendiment com la simplicitat en les consultes, cosa que permet un maneig àgil i directe de la informació sense la necessitat de complicades relacions entre múltiples taules.

|  |  |  |
| --- | --- | --- |
| Nom | Tipus | Atributs |
| nom | VARCHAR | PK |
| edat | SMALLINT | PK |
| experiencia | BOOLEAN |  |
| sexe | CHAR |  |
| tirs | MEDIUMINT | UNSIGNED |
| encerts | MEDIUMINT | UNSIGNED |
| puntuacio | INT | UNSIGNED |
| ronda | SMALLINT | UNSIGNED |

Taula 3.1 Model de dades

Com es pot veure a la Taula 3.1, les dades guardades són les següents:

* El nom i l’edat que introdueixi l’usuari servirà per identificar-lo, per això s’han definit com a claus primàries, això és perquè l’usuari pugui iniciar sessió i actualitzar la seva puntuació si aconsegueix una de més alta.
* Experiència serveix per saber si l’usuari tenia experiència prèvia en RV.
* Sexe és purament estadístic.
* Tirs i encerts són el nombre de tirs i encerts efectuats, respectivament, amb l’arc durant la partida.
* Puntuació és la puntuació total obtinguda al final de la partida.
* Ronda és la ronda en la qual ha perdut en aquesta partida.

### Disseny arquitectònic

En aquest apartat, es presenta l’arquitectura del sistema. S’ha implementat com un model client-servidor, aquest model divideix el sistema en dues parts principals:

* El client: és la interfície a través de la qual l’usuari accedeix al sistema.
* El servidor: és qui dona servei a les peticions dels clients.

L’esquema de funcionament d’aquest model seria el següent:

1. El client realitza una petició al servidor.
2. El servidor rep la petició, la processa i es comunica amb la base de dades si és necessari.
3. El client rep la resposta del servidor, la processa i mostra el resultat si és necessari.

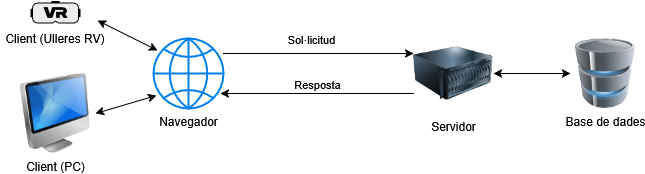


Figura 3.3 Diagrama arquitectònic

La Figura 3.3 mostra com s’ha implementat el model més específicament en aquest projecte. A on un client amb un ordinador o unes ulleres de realitat virtual independents utilitza el navegador per fer les sol·licituds al servidor, i aquest es pot comunicar amb la base de dades si fa falta, però el client no pot accedir a la base de dades directament tant per seguretat com per simplicitat.

## Implementació

En l'apartat d'implementació, s'abordarà l’explicació del disseny arquitectònic client-servidor definit en el punt 3.3.5. Aquest enfocament permet que els clients interactuïn amb el servidor a través de sol·licituds específiques, mentre que la base de dades s'encarrega de gestionar i emmagatzemar la informació de manera eficient. Al llarg d'aquesta secció, es detallaran els components clau de la implementació, incloent-hi la configuració del servidor, la integració de la base de dades i el desenvolupament de les aplicacions client, així com les tecnologies i eines utilitzades per a garantir un rendiment òptim i una experiència d'usuari fluida.

### Servidor

Per la creació del servidor s’ha utilitzat Node.js [8], junt amb el *framework* Express.js [9] per facilitar la creació del servidor web. Aquest proporciona el fitxer HTML del joc i també serveix com a intermediari entre la comunicació del joc i la base de dades.

Per a gestionar la instal·lació i manteniment d’aquest framework i altres mòduls necessaris s’ha utilitzat el gestor npm [10]. Aquest gestor et permet accedir a un gran repositori de paquets, facilitant la incorporació de funcionalitats als projectes sense necessitat de desenvolupar tot des de zero. A més, habilita la gestió de dependències, l’execució de scripts i la configuració de projectes a través d’un arxiu anomenat “package.json”.

#### Codi del servidor

Fent servir el *framework* d’Express.js, s’ha aconseguit gestionar les sol·licituds, respostes sobre el mateix i el *middleware* necessari perquè funcioni tot correctament. El Codi 3‑1 mostra la implementació.

const PORT = 80;

// Instanciar Express

const app = express();

// Middleware per llegir comandes JSON

app.use(express.json());  
app.use(express.*static*(path.join(\_\_dirname, 'src')));

// Servir els fitxers estàtics

app.get('/', function (req, res) {  
 res.sendFile(path.join(\_\_dirname, './index.html'));  
});

// Crear el servei HTTP

http.*createServer*(app).listen(PORT);

Codi 3‑1 Codi del servidor

#### Variables d’entorn

Les variables d’entorn són parells clau-valor que s’utilitzen per a emmagatzemar informació de configuració en un sistema operatiu o en un entorn d’execució d’aplicacions. Aquestes variables permeten als programes accedir a dades com a rutes d’arxius, configuracions de xarxa, credencials d’accés i altres paràmetres que poden variar entre diferents entorns (per exemple, desenvolupament, proves i producció). D’aquesta manera, es facilita la gestió de configuracions sense necessitat de modificar el codi font.

Per aconseguir aquesta funcionalitat, s’ha importat el mòdul dotenv [11] a Node.js perquè agafi aquestes variables des de l’arxiu “.env”. Aquest arxiu conté els parells clau-valor que es necessiten en el projecte. Com es pot veure en el Codi 3‑2 en el nostre cas només s’ha necessitat incloure la informació de connexió amb la base de dades.

DB\_HOST='localhost'  
DB\_USER='[usuari]'  
DB\_PWD='[contrasenya]'  
DB\_NAME='[nom de la BBDD]'

Codi 3‑2 Arxiu .env

Un cop tenim l’arxiu simplement necessitem carregar-lo tan aviat com es pugui dins l’aplicació Node.js, per tant, la primera línia de l’arxiu del servidor és la del Codi 3‑3 que importa les variables d’entorn definides anteriorment.

require('dotenv').*config*();

Codi 3‑3 Importar configuració .env

#### Accés a la base de dades

Com s’ha explicat abans, el servidor s’encarrega de la comunicació entre el joc i la base de dades. Com mostra el Codi 3‑4 Rutes AJAX, s’han afegit diverses rutes a les quals el client pot accedir mitjançant AJAX [12].

const DBFunctions = require('./DBFunctions.js');

app.post('/getUsuari', async (req, res) => {  
 DBFunctions.GetUser(req, res);  
});  
  
app.post('/preguntesUsuari', function (req, res) {  
 DBFunctions.InsertPreguntes(req, res);  
});  
  
app.post('/updatePuntuacio', function (req, res) {  
 DBFunctions.UpdatePuntuacio(req, res);  
});

app.get('/getTopPuntuacions', function (req, res) {  
 DBFunctions.*GetTopPuntuacions*(req, res);  
});

Codi 3‑4 Rutes AJAX

Aquestes rutes executen les funcions que conté l’arxiu “DBFunctions.js” amb el codi SQL que es comunica amb la base de dades.

Per la connexió amb la base de dades s’ha utilitzat el mòdul de MariaDB [13], el qual et permet crear connexions i fer consultes a bases de dades MySQL o MariaDB. El Codi 3‑5 mostra la connexió a la base de dades i les diferents funcions que executen les consultes o modificacions pertinents.

const mariadb = require('mariadb');

*// Connexió a la base de dades*  
const serverPool = mariadb.*createPool*({  
 host: process.env.DB\_HOST,  
 user: process.env.DB\_USER,  
 password: process.env.DB\_PWD,  
 database: process.env.DB\_NAME *// Nom de la base de dades a utilitzar*  
});

module.exports = {

*// Selecciona l’usuari (nom + edat) a la base de dades, si no existeix el crea*

*GetUser*(req, res) {  
 [...]  
},

*// Actualitza l’usuari afegint la informació de les preguntes (experiència i sexe)*

*InsertPreguntes*(req, res) {  
 [...]  
},

*// Actualitza la puntuació i altres dades estadístiques de l'usuari*

*UpdatePuntuacio*(req, res) {  
 [...]  
}

*// Selecciona les 10 majors puntuacions de la base de dades*

*GetTopPuntuacions*(req, res) {  
 [...]  
}

}

Codi 3‑5 Fitxer DBFunctions.js

Gràcies a aquestes funcions i la utilització d’AJAX, des del client podem crear l’usuari i saber si fer-li les preguntes o iniciar sessió directament i actualitzar les seves dades sobre la puntuació.

### Base de dades

Com ja s’ha explicat, les dades que hem de guardar són molt poques i es pot utilitzar només una taula per emmagatzemar tot el que necessitem.

CREATE TABLE `puntuacions` (  
 `nom` varchar(25) NOT NULL,  
 `edat` tinyint(4) NOT NULL,  
 `experiencia` tinyint(1) DEFAULT NULL,  
 `sexe` char(1) DEFAULT NULL,  
 `dispars` mediumint(11) DEFAULT NULL,  
 `encerts` mediumint(11) DEFAULT NULL,  
 `puntuacio` int(11) DEFAULT NULL,  
 `ronda` smallint(11) DEFAULT NULL  
);

ALTER TABLE `puntuacions` ADD PRIMARY KEY (`nom`,`edat`);

Codi 3‑6 Taula SQL

Com es pot veure en el Codi 3‑6, a part de crear la taula, també es genera el parell nom i edat com a claus primàries per identificar els diferents usuaris.

### Client

La implementació del joc es fa en el costat del client, per tant, aquest apartat abasta la major part de la implementació del projecte...

#### A-Frame

El *framework* d’A-Frame es basa en HTML i JavaScript, per tant, s’ha hagut de crear un document HTML afegint el *framework* dins l’element “<head>” de l’arxiu. Un cop fet es pot crear una escena d’A-Frame, aquesta escena és la que conté els components a mostrar del joc. Per crear i modificar la lògica dels components es fa a través de JavaScript.

Un component és una unitat modular que encapsula una funcionalitat específica i es pot reutilitzar en diferents entitats dins d’una escena, per exemple, el component fletxa es reutilitza per totes les fletxes generades. Existeixen components ja creats per A-Frame que ofereixen certes funcionalitats, però també et permet crear els teus propis, d’aquesta manera, els desenvolupadors poden agregar característiques i comportaments als elements 3D de manera senzilla i organitzada.

#### Components

Com s’acaba d’explicar en el punt anterior, els components necessaris i la modificació dels mateixos s’ha fet a través de JS.

AFRAME.registerComponent('nomComponent', {  
 schema: {  
 dada1: {type: 'number', default: 0.5},  
 [...]  
 },  
 init: function () {  
 [...]  
 },  
 update: function () {  
 [...]  
 },  
 tick: function () {  
 [...]  
 },  
 remove: function () {  
 [...]  
 },  
 funcioPersonalitzada: function () {  
 [...]  
 }  
});

Codi 3‑7 Component d’A-Frame

Al Codi 3‑7 es pot veure l’estructura típica d’un component, encara que pot contenir tots els següents apartats:

* Schema: És un objecte JSON[[5]](#footnote-6) que defineix les propietats del component com per exemple el color o la grandària.
* Init: És la funció que s’executa només una vegada quan es crea el component a l’escena.
* Update: És la funció que s’executa en qualsevol moment que les propietats del component canvien.
* Tick: És la funció que s’executa a cada fotograma que es genera.
* Tock: Idèntic a la funció tic però s’executa un cop ja s’ha renderitzat l’escena.
* Remove: És la funció que s’executa quan s’elimina el component.
* Pause: És la funció que s’executa quan l’entitat o l’escena es pausa.
* Play: És la funció que s’executa quan l’entitat o l’escena es reprèn.
* Funcions personalitzades: Es poden afegir diferents funcions al component encara que no s’executen automàticament per A-Frame, s’han de cridar explícitament.

Per cobrir les funcionalitats del joc s’han creat els següents components:

* Arc
* Corda
* Fletxa
* Planta
* Vagoneta
* Terra
* Diana

##### Arc

Aquest component s’encarrega d’assignar el model 3D de l’arc a l’entitat, el qual es pot veure a la i generar la *pool[[6]](#footnote-7)* de fletxes. Una *pool* en A-Frame és un conjunt de components de mida limitada els quals es poden anar agafant i retornant de manera dinàmica, utilitzant sempre les mateixes entitats ja generades. D’aquesta manera evitem haver de destruir l’entitat i generar-ne una de nova. En aquest cas, actua com un carcaix virtual, en el qual les fletxes ja disparades es retornen al carcaix i es reutilitzen.

Quan s’agafa l’arc, es canvia el model de la mà que l’ha agafat pel model de l’arc, també s’afegeix el component de la corda i internament s’assigna l’altra mà com la “mà de la corda”, que serà la que podrà agafar les fletxes.

Un cop fet això demana una fletxa de la *pool* i genera el menú principal.

Per igualar la rotació de la fletxa amb l’arc s’han fet ús dels quaternions[[7]](#footnote-8). Un quaternió és un nombre complex estès a quatre dimensions, que s'utilitza en matemàtiques i física per a representar rotacions en l'espai tridimensional. Els quaternions són especialment útils en gràfics per ordinador, robòtica i navegació, ja que permeten realitzar interpolacions suaus i evitar problemes com la *gimbal lock[[8]](#footnote-9)* que poden ocórrer amb altres mètodes de representació de rotacions, com els angles d'Euler. Es pot veure un exemple de *gimbal lock* en el moviment del braç de la Figura 3.4, a causa de la utilització d’angles d’Euler en vers de quaternions.



Figura 3.4 Exemple de *gimbal lock*

##### Corda

Aquest component simplement genera una corba utilitzant la funció de Three.js de *CatmullRomCurve3*. Aquesta corba simularà l’extensió i retracció de la corda.

La corba de Catmull–Rom[[9]](#footnote-10) utilitza l’algorisme de Catmull–Rom per generar una corba llisa segons els punts indicats. S’ha fet servir aquest algorisme, perquè la corba que genera és molt similar a l’extensió d’una corda, com es pot veure en la Figura 3.3. A més a més, també dibuixa una línia recta si es posen tots els punts d’un eix a la mateixa altura, per tant ens permet simular tots els possibles estats de la corda.

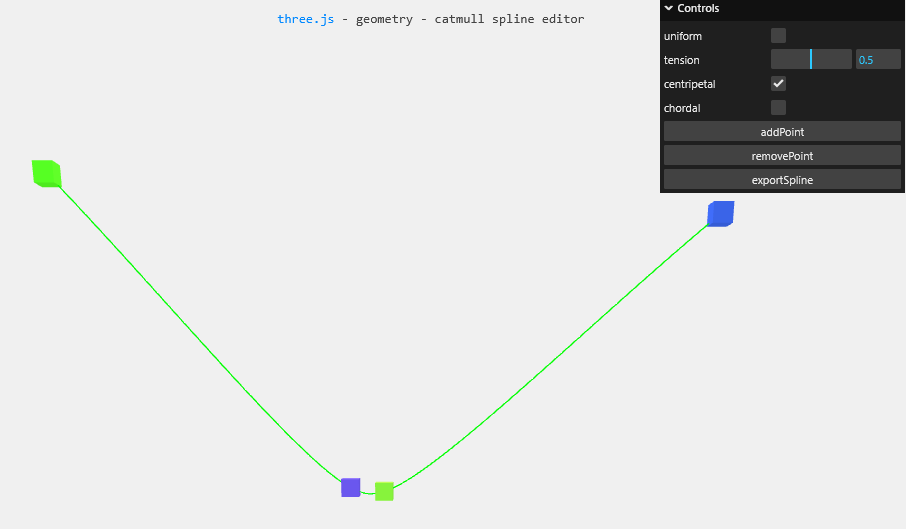


Figura 3.5 Corba Catmull-Rom three.js

##### Fletxa

El component de fletxa és el que més càlculs conté, defineix la fletxa, el seu moviment i les col·lisions.

Primerament, defineix el model 3D de la fletxa i li assigna el component *aabb-collider[[10]](#footnote-11)* el qual aprovisiona de detecció de col·lisions.

Un detector de col·lisions AABB (*Axis-Aligned Bounding Box*) és un mètode utilitzat en gràfics per ordinador i simulacions físiques per a determinar si dos objectes en un espai 2D o 3D incideixen. Aquest enfocament es basa a embolicar cada objecte en un rectangle (en 2D) o un paral·lelepípede (en 3D) alineat amb els eixos de coordenades, la qual cosa simplifica el càlcul de col·lisions. En verificar si les AABBs de dos objectes se superposen, es pot determinar ràpidament si hi ha una possible col·lisió entre ells, la qual cosa permet optimitzar el rendiment en aplicacions on es manegen múltiples objectes, com a videojocs o simulacions físiques.

També disposa de la funció *tick*, la qual actualitza el seu estat en cada fotograma. Aquesta funció s’encarrega de copiar la posició i rotació de la mà de la corda fins que s’apropa prou a l’arc i es pressiona el botó disparador. Quan ocorr això, en vers de copiar la posició i rotació de la mà de la corda, passa a copiar-ho de la mà de l’arc. D’aquesta manera se simula que s’ha posat la fletxa a l’arc.

Un cop la fletxa està acoblada a l’arc, segons la distància entre la mà de la corda i la mà de l’arc, es mourà la fletxa i la corda de forma proporcional fins a un límit. Amb això aconseguim modular la tensió de la corda.

A partir d’aquí, un cop s’amolla el botó disparador, s’agafa el mínim entre la distància entre les mans i el límit de la corda, establert anteriorment, per calcular la velocitat inicial de la fletxa seguint la fórmula:

A on

* = Eficiència de l’arc, pels arcs medievals és de *0.9*.
* = Força de l’arc, pels arcs medievals varia entre 200 i 400 N.
* = Distància entre la posició en repòs i la tensió actual, en el nostre cas la distància entre les mans.
* = La massa de la fletxa.
* = Factor el qual representa la suma de les energies cinètiques de les parts mòbils de l’arc. Pels arcs medievals varia entre 0.03 i 0.07.
* = La massa de l’arc (aproximadament 1 kg).

En el cas del joc, totes aquestes variables són constants durant tota la durada de l’execució, excepte la (la distància entre les mans). Per tant, es pot separar l’equació en dues parts:

D’aquesta manera el que aconseguim és tenir una constant multiplicada per la variable de la distància per estalviar recursos en temps d’execució.

Com a constants, per tenir una representació el més real possible, s’han definit les següents a partir de dades reals i ajustant alguns valors, per obtenir una sensació de dispar més agradable.



Un cop coneguda la velocitat inicial de la fletxa, ara ha de seguir la trajectòria d’un projectil típic amb la força de la gravetat actuant sobre ell. Per aconseguir això s’ha utilitzat la següent fórmula:

A on:

* = És la velocitat inicial obtinguda anteriorment.
* = És el temps transcorregut.
* = És l’angle en el que apunta la fletxa.
* = És la força de la gravetat.

L’angle de dispar s’obté a partir del quaternió de la fletxa com ja s’ha explicat anteriorment.

Al igual s’ha fet abans per reduir el nombre d’operacions en temps d’execució, s’ha dividit la fórmula en les següents passes.

Com a especificació del component es té que:



A l’hora de disparar es calculen les velocitats inicials sobre i , sense tenir en compte el temps.

Com el que varia és el temps transcorregut, al bucle de renderització (la funció tick() del component) es calcula la resta d’operacions que depenen d’ell.

Gràcies a aquesta optimització, el bucle *tick()*, que s'executa en cada fotograma, realitzarà un menor nombre d'operacions. Cosa que es tradueix en una major eficiència en el rendiment del sistema, es millora la fluïdesa i la resposta general de l'aplicació, la qual permet una experiència més fluida per al jugador.

A partir de la fórmula de velocitat inicial segons l'arc i la fórmula de la trajectòria d'un projectil, podem obtenir una representació realista del tir d'una fletxa amb un arc en la Terra. Aquestes fórmules ens permeten calcular amb precisió la velocitat i l'angle de llançament, i al mateix temps determinar la trajectòria que seguirà la fletxa. Gràcies a això, podem simular de manera efectiva com es comporta la fletxa en un entorn real, proporcionant així una major immersió.

Finalment, quan la fletxa supera el temps establert, té una posició en (altura) negativa o col·lisiona amb alguna entitat, es retorna a la *pool* per poder tornar a utilitzar-la més endavant sense haver de generar i renderitzar una fletxa nova cada vegada.

##### Planta

La planta és l’enemic principal, aquest component s’encarrega de assignar a la entitat una rotació aleatòria i la mida especificada, un cop fet, genera dos elements fills que contenen els models 3D, un és el de la planta amb les animacions de créixer i de morir i l’altre és el de la fruita i li assigna un color diferent segons la vida que tingui, i només amb l’animació de morir, degut a que el component de les col·lisions no s’actualitza a la posició real, sinó que assigna el col·lisionador al primer fotograma de l’animació. Degut a aquest mateix problema, no es pot crear una *pool* d’enemics, perquè el col·lisionador s’assigna a la primera posició que obté l’element i quan es retorna a la pool i es torna a assignar a l’escena amb una nova posició, aquest seguia tenint el detector de col·lisions en la primera posició assignada originalment.

La lògica programada del component és la següent:

* Quan es crea un nou element del tipus planta, aquest comença amb l’animació de la planta de créixer, quan finalitza l’animació, la fruita passa a ser visible.
* Quan el col·lisionador de la fruita detecta una col·lisió, i aquesta té menys de dos punts de vida, comença les animacions de morir, tant de la fruita com de la planta.
* Quan la vagoneta supera la distància assignada, s’elimina el component de l’escena i s’activa el so de perdre vida.

##### Vagoneta

...Ús de *navmesh* per limitar el moviment del jugador a només la vagoneta...

##### Terra

... es genera un model gran i es comprova la distància quan s’arriba al punt de control, per tant es tenen a la vegada com a molt dues entitats de terra en escena...

##### 

##### Diana

Adsf

**Complicacions**

Arc: la mesh de l’arc amb armature three.js la detecta però és molt complicat modificar la mesh a partir dels ossos detectats. Per això s’ha optat per generar una corba Catmull–Rom per simular la deformació de la corda. ([Afegir enllaç](https://en.wikipedia.org/wiki/Centripetal_Catmull%E2%80%93Rom_spline))

Enemics: Primerament es varen afegir els enemics com una *pool*, però quan es demanava una entitat ja utilitzada a l’escena anteriorment i se li assignava una nova posició, el component de detecció de les col·lisions AABB no s’actualitzava a la nova posició, per tant per que detectés la col·lisió havies de disparar a la primera posició assignada a l’entitat, posició en la qual ja no hi havia res visualment a l’escena.

Terra: millor una sola entitat amb un model 3d gran, encara que s’utilitzin pools i el mateix model 3D per totes les entitats.

Fira ciència per a tothom:

Va servir per descobrir errades de programació, les quals es varen arreglar.

* Dos botons diferents per agafar la fletxa provoca una errada visual a on surt més d’una fletxa quan pressiones els dos botons. Solució només un botó per agafar la fletxa, com al mini joc Longbow del Steam Lab el botó per agafar la fletxa és el disparador i quan el vaig provar per primera vegada em va parèixer més intuïtiu només s’ha deixat aquest.
* Es permetia la rotació de la càmera amb la palanca del controlador, si la utilitzaves l’arc també rotava, però les fletxes no, per tant, només afectava visualment. Solució llevar la rotació amb el controlador, ja que no és necessari i és més intuïtiu que roti el jugador.
* Es podia agafar l’arc un cop haguessin iniciat sessió, però com la connexió no era molt bona a vegades es trigava fins a uns 5 segons a poder agafar l’arc per mor d’esperar la resposta del servidor. Solució, un cop verificat que el nom és correcte, poder agafar l’arc abans d’enviar-ho al servidor.
* La partida normal es feia un poc massa llarga i arribava a ser un poc avorrida. Solució augmentar la velocitat de la vagoneta, augmentar el nombre d’enemics a la vegada i reduir el temps de generació d’enemics.
* Les fletxes estaven constantment adherides a l’arc, el que feia que li llevés immersió. Solució fer que les fletxes estiguin a la mà contrària fins que es pressiona el botó disparador a prop de l’altra mà, en aquest moment passa a estar adherida a l’arc igual que anteriorment.
* A vegades, si es tiraven moltes fletxes seguides i quan es demanava una fletxa a la *pool*, aquesta estava buida i havies de reiniciar el joc complet. Solució si quan demanes una fletxa no en retorna cap, afegir un temps de “penalització” fins a tornar a demanar-ne una.

En general a la gent li va agradar el joc.

# 

# Referències

|  |  |
| --- | --- |
| [1] | «Vikipèdia tower defense,» [En línia]. Available: https://ca.wikipedia.org/wiki/Tower\_Defense. |
| [2] | «Wikipèdia endless runner,» [En línia]. Available: https://es.wikipedia.org/wiki/Corredor\_sin\_fin. |
| [3] | «A-Frame,» [En línia]. Available: https://aframe.io/. |
| [4] | «Three.js,» [En línia]. Available: https://threejs.org/. |
| [5] | «Blender,» [En línia]. Available: https://www.blender.org/. |
| [6] | «GitHub,» [En línia]. Available: https://github.com/. |
| [7] | «Low poly - Wikipedia,» [En línia]. Available: https://en.wikipedia.org/wiki/Low\_poly. |
| [8] | «Node.js,» [En línia]. Available: https://nodejs.org/en. |
| [9] | «Express,» [En línia]. Available: https://expressjs.com/. |
| [10] | «npm,» [En línia]. Available: https://www.npmjs.com/. |
| [11] | «dotenv - npm,» [En línia]. Available: https://www.npmjs.com/package/dotenv. |
| [12] | «AJAX - Wikipedia,» [En línia]. Available: https://en.wikipedia.org/wiki/Ajax\_(programming). |
| [13] | «Mariadb - npm,» [En línia]. Available: https://www.npmjs.com/package/mariadb. |

# 

Nota: el vostre document no té per què tenir apèndixs.

## Fonts de l’especificació de l’arc

* Massa de la fletxa: <https://exarc.net/issue-2019-3/mm/shooting-experiments-early-medieval-arrowheads>
* Força de l’arc: <https://en.wikipedia.org/wiki/English_longbow#Draw_weights>

# 

Apèndix de les obres creades per altres autors.

## Arc

Llicència: [CC BY 3.0](https://creativecommons.org/licenses/by/3.0/)

Autor: Zsky

Modificacions: No

Ref.: <https://poly.pizza/m/XkJVO6cACA>

## Fletxa

Llicència: [CC BY 3.0](https://creativecommons.org/licenses/by/3.0/)

Autor: Poly by Google

Modificacions: No

Ref.: <https://poly.pizza/m/6pMBOmBFxGt>

## Planta

Llicència: [CC0 1.0](https://creativecommons.org/publicdomain/zero/1.0/)

Modificacions: No

Ref.: <https://poly.pizza/m/0bVHWZZnNg>

## Vagoneta

Llicència: [CC BY 3.0](https://creativecommons.org/licenses/by/3.0/)

Autor: Hunter Paramore

Modificacions: Si

Ref.: <https://poly.pizza/m/fjvk6xVJ3u3>

## Arbres

Llicència: [CC0 1.0](https://creativecommons.org/publicdomain/zero/1.0/)

Modificacions: No

Ref.: <https://poly.pizza/m/oYtDty0fR6>

## Diana

Llicència: [CC0 1.0](https://creativecommons.org/publicdomain/zero/1.0/)

Modificacions: Si

Ref.: <https://poly.pizza/m/gKYbYR3z0M>

## Cel

Llicència: [CC0 1.0](https://creativecommons.org/publicdomain/zero/1.0/)

Modificacions: No

Ref.: <https://polyhaven.com/a/kloofendal_48d_partly_cloudy_puresky>

## So partida perduda

Llicència: [CC0 1.0](https://creativecommons.org/publicdomain/zero/1.0/)

Modificacions: No

Ref.: <https://freesound.org/people/Fupicat/sounds/538151/>

## So arrancar fruita

Llicència: [CC0 1.0](https://creativecommons.org/publicdomain/zero/1.0/)

Modificacions: No

Ref.: <https://openverse.org/audio/42d2a0db-e82c-4732-aacf-986d7acfbec1?q=squashing+fruit&p=9>

## So perdre vida

Llicència: [pixabay](https://pixabay.com/service/license-summary/)

Modificacions: Si

Ref.: <https://pixabay.com/sound-effects/search/effect/>

## So botó pressionat

Llicència: [CC BY-NC 4.0](https://creativecommons.org/licenses/by-nc/4.0/)

Autor: Mellau

Modificacions: No

Ref.: <https://freesound.org/people/Mellau/sounds/506054/>

## So fletxa disparada

Llicència: [CC0 1.0](https://creativecommons.org/publicdomain/zero/1.0/)

Modificacions: Si

Ref.: <https://freesound.org/people/bruno.auzet/sounds/527435/>

1. <https://play.google.com/store/apps/details?id=com.flaregames.royalrevolt&hl=en-US> [↑](#footnote-ref-2)
2. <https://store.steampowered.com/app/960090/Bloons_TD_6/> [↑](#footnote-ref-3)
3. <https://play.google.com/store/apps/details?id=com.imangi.templerun&hl=en-US> [↑](#footnote-ref-4)
4. <https://store.steampowered.com/app/450390/The_Lab/> [↑](#footnote-ref-5)
5. <https://en.wikipedia.org/wiki/JSON> [↑](#footnote-ref-6)
6. <https://aframe.io/docs/1.7.0/components/pool.html#main> [↑](#footnote-ref-7)
7. <https://ca.wikipedia.org/wiki/Quaterni%C3%B3> [↑](#footnote-ref-8)
8. <https://en.wikipedia.org/wiki/Gimbal_lock> [↑](#footnote-ref-9)
9. <https://en.wikipedia.org/wiki/Centripetal_Catmull%E2%80%93Rom_spline> [↑](#footnote-ref-10)
10. <https://en.wikipedia.org/wiki/Minimum_bounding_box> [↑](#footnote-ref-11)