

TREBALL DE FI DE GRAU

APLICACIÓ EN REALITAT VIRTUAL DE MUNTATGE D'ORDINADORS

Carlos Veny Carmona

Grau en Enginyeria Informàtica

Escola Politècnica Superior

Any acadèmic 2021-22

APLICACIÓ EN REALITAT VIRTUAL DE MUNTATGE D'ORDINADORS

Carlos Veny Carmona

Treball de Fi de Grau

Escola Politècnica Superior

Universitat de les Illes Balears

Any acadèmic 2021-22

Paraules clau del treball:

realitat virtual, simulador, ordinador, muntatge d'ordinadors

Nom del tutor / la tutora del treball: Antoni Oliver Tomàs

Nom del tutor / la tutora (si escau): Antonio Bibiloni Coll

Autoritz la Universitat a incloure aquest treball en el repositori	Aut	or/a	Tut	or/a
institucional per consultar-lo en accés obert i difondre'l en línia, amb	Sí	No	Sí	No
finalitats exclusivament acadèmiques i d'investigació	\boxtimes		\boxtimes	

A la meva família, per haver-me donat suport amb les meves metes i aspiracions, i per confiar sempre en mi.

Taula de continguts

Taula de contingutsi
Índex de figuresv
Índex de taulesvii
Nomenclaturaix
Resumxi
Capítol 1. Introducció 1
1.1. Contextualització1
1.2. Motivació 1
1.3. Objectius
1.4. Abast del projecte 2
1.5. Estructura del document
Capítol 2. Estat de l'art 5
2.1. Solucions existents a problemes semblants 5
2.1.1. Solucions de muntatge d'ordinadors7
2.1.2. Solucions en VR en entorns d'oficina8
2.2. Anàlisi i selecció de eines per resoldre el problema
2.2.1. IDEs
2.2.2. Tecnologies gràfiques
2.2.3. Compatibilitat i especificacions dels components
Capítol 3. Desenvolupament del projecte 11
3.1. Gestió del projecte 11
3.1.1. Metodologia de desenvolupament 11
3.1.2. Planificació temporal 12
3.1.3. Gestió dels interessats13
3.1.4. Gestió de riscs
3.1.5. Gestió econòmica

3.2. Analisi	15
3.2.1. Usuaris	15
3.2.2. Requisits	15
3.2.2.1. Requisits funcionals	16
3.2.2.1.1. Jugador	16
3.2.2.1.2. Dissenyador de peces	16
3.2.2.1.3. Dissenyador de proves	16
3.2.2.2. Requisits No Funcionals	17
3.2.2.2.1. Requisits No Funcionals d'interacció	17
3.3. Disseny	17
3.3.1. Disseny de les tasques	18
3.3.2. Model de dades	18
3.3.2.1. Model conceptual	18
3.3.2.2. Model relacional	19
3.3.3. Disseny arquitectònic	20
3.4. Implementació	21
3.4.1. Joc en VR	22
3.4.1. Joc en VR 3.4.1.1. Escenari exterior	22
3.4.1. Joc en VR 3.4.1.1. Escenari exterior 3.4.1.1.1. Textures	22 23 24
 3.4.1. Joc en VR 3.4.1.1. Escenari exterior 3.4.1.1.1. Textures 3.4.1.1.2. Component Instanced Mesh 	22 23 24 24
 3.4.1. Joc en VR 3.4.1.1. Escenari exterior 3.4.1.1.1. Textures 3.4.1.1.2. Component Instanced Mesh 3.4.1.1.3. Component cel i sol 	22 23 24 24 24
 3.4.1. Joc en VR 3.4.1.1. Escenari exterior 3.4.1.1.1. Textures 3.4.1.1.2. Component Instanced Mesh 3.4.1.1.3. Component cel i sol 3.4.1.1.4. Menú auxiliar 	22 23 24 24 25 25
 3.4.1. Joc en VR 3.4.1.1. Escenari exterior 3.4.1.1.1. Textures 3.4.1.1.2. Component Instanced Mesh 3.4.1.1.3. Component cel i sol 3.4.1.1.4. Menú auxiliar 3.4.1.2. Fonts 	22 23 24 24 25 25 25
 3.4.1. Joc en VR 3.4.1.1. Escenari exterior 3.4.1.1.1. Textures 3.4.1.1.2. Component Instanced Mesh 3.4.1.1.3. Component cel i sol 3.4.1.1.4. Menú auxiliar 3.4.1.2. Fonts 3.4.1.3. Menú de configuració 	22 23 24 24 25 25 25 26
 3.4.1. Joc en VR 3.4.1.1. Escenari exterior 3.4.1.1.1. Textures 3.4.1.1.2. Component Instanced Mesh 3.4.1.1.3. Component cel i sol 3.4.1.1.4. Menú auxiliar 3.4.1.2. Fonts 3.4.1.3. Menú de configuració 3.4.1.3.1. Opció "Registrarse" 	22 23 24 24 25 25 25 26 27
 3.4.1. Joc en VR	22 23 24 24 25 25 26 26 27 27
 3.4.1. Joc en VR 3.4.1.1. Escenari exterior. 3.4.1.1.1. Textures. 3.4.1.1.2. Component Instanced Mesh 3.4.1.1.3. Component cel i sol 3.4.1.1.4. Menú auxiliar. 3.4.1.2. Fonts. 3.4.1.3. Menú de configuració 3.4.1.3.1. Opció "Registrarse" 3.4.1.3.2. Opció "Iniciar Sesión" 3.4.1.3.3. Opció "Editar Perfil" 	22 23 24 24 25 25 25 26 27 27 27
 3.4.1. Joc en VR 3.4.1.1. Escenari exterior. 3.4.1.1.1. Textures. 3.4.1.1.2. Component Instanced Mesh 3.4.1.1.3. Component cel i sol 3.4.1.1.4. Menú auxiliar. 3.4.1.2. Fonts. 3.4.1.3. Menú de configuració 3.4.1.3.1. Opció "Registrarse" 3.4.1.3.2. Opció "Iniciar Sesión" 3.4.1.3.4. Opció "Información". 	22 23 24 24 25 25 25 26 27 27 27 27 28
 3.4.1. Joc en VR	22 23 24 24 25 25 25 25 25 26 27 27 27 27 27 28 28

3.4.1.4. PHP Session	. 29
3.4.1.5. Menú estadístiques	. 30
3.4.1.6. Escenari interior	. 30
3.4.1.7. Il·luminació	. 31
3.4.1.8. Desplaçament interior/exterior	. 31
3.4.1.9. Menú components	. 32
3.4.1.9.1. Generació de components	. 33
3.4.1.10. Menú tasques	. 33
3.4.1.11. Físiques	. 34
3.4.1.12. Interacció dels controladors	. 34
3.4.1.12.1. Controladors	. 35
3.4.1.12.2. Component super-hands	. 35
3.4.1.12.3. Intercanvi controladors i super-hands	. 36
3.4.1.13. Col·lisions	. 36
3.4.1.14. Interacció dels components	. 37
3.4.1.14.1. Agafar components	. 37
3.4.1.14.2. Col·locar components	. 38
3.4.1.14.3. Descol·locar components	. 39
3.4.1.15. Final de partida	. 40
3.4.1.15.1. Revisió de tasques	. 41
3.4.1.15.2. Revisió de col·locació dels components	. 42
3.4.1.15.3. Revisió de compatibilitat tècnica	. 42
3.4.1.15.4. Càlcul de puntuació	. 42
3.4.1.16. Tutorial	. 43
3.4.1.17. Centrar al jugador	. 44
3.4.1.18. Música i sons	. 45
3.4.1.18.1. Modificació del volum	. 46
3.4.1.19. Rendiment	. 47
3.4.1.19.1. Quest Browser	. 48
3.4.1.20. Seguretat	. 48

3.4.2. Interfície administradors	48
3.4.2.1. Login	49
3.4.2.2. Bootstrap	50
3.4.2.3. Icones	50
3.4.2.4. Targetes peces/proves	51
3.4.2.5. Visualització de peces/proves	52
3.4.2.6. Eliminació de peces/proves	53
3.4.2.7. Edició de peces	53
3.4.2.7.1. Control d'errors	54
3.4.2.8. Edició de proves	55
3.4.2.8.1. Compatibilitat de peces	56
3.4.2.9. Creació de peces/proves	57
3.4.3. Interfície estadístiques	57
3.4.3.1. Taula d'estadístiques	58
3.4.3.1.1. Filtres	59
3.4.3.1.2. Web responsive	59
3.4.3.2. Visites del joc	60
3.4.3.2.1. Emmagatzematge de les dades	60
3.5. Proves	61
3.5.1. Proves internes	62
3.5.2. Proves d'usuaris externs	63
3.6. Instal·lació	64
Capítol 4. Resultats	65
Capítol 5. Conclusions	67
Capítol 6. Referències bibliogràfiques	71

Índex de figures

Figura 2.1. PC Building Simulator	5
Figura 2.2. Joc VR de AMD	5
Figura 2.3. Job Simulator	6
Figura 2.4. edataconsulting VR Office	6
Figura 2.5. PC Repair Simulator	6
Figura 2.6. TestOut PC Pro	6
Figura 2.7. Captura de pantalla de PCPartPicker	10
Figura 3.1 Cronograma previst	12
Figura 3.2 Model de dades	19
Figura 3.3 Disseny arquitectònic	21
Figura 3.4 Submòduls de implementació del joc	23
Figura 3.5 Menú principal	24
Figura 3.6 Tanques amb textura	25
Figura 3.7 Font del títol	26
Figura 3.8 Menú de configuració	26
Figura 3.9 Teclat en VR	28
Figura 3.10 Interfície banner del perfil	29
Figura 3.11 Menú estadístiques de l'usuari "jaume"	30
Figura 3.12 Menú de components	32
Figura 3.13 Submenú CPU	32
Figura 3.14 Menú de tasques	34
Figura 3.15 Funcionament de col·lisions	37
Figura 3.16 Drop zones inicials	39
Figura 3.17 Drop zones placa base	39
Figura 3.18 Descol·locació de peces	40
Figura 3.19 Informació final de partida	40

Índex de figures

Figura 3.20 Funció per centrar al jugador	44
Figura 3.21 Interfície modificació de volum	46
Figura 3.22 Submòduls de la implementació de la interfície d'administradors	49
Figura 3.23 Login interfície peces	50
Figura 3.24 Targeta peça	51
Figura 3.25 Targeta prova	51
Figura 3.26 Interfície de visualització d'una peça	52
Figura 3.27 Interfície de visualització d'una prova	52
Figura 3.28 Interfície d'edició d'una peça	53
Figura 3.29 Edició tasca d'emmagatzematge	55
Figura 3.30 Edició tasca extra	56
Figura 3.31 Submòduls de la implementació de la interfície d'estadístiques	57
Figura 3.32 Interfície taula d'estadístiques	58
Figura 3.33 Menús desplegables interfície estadístiques	59
Figura 3.34 Interfície visites al joc	60

Índex de taules

Taula 2.1. Comparació de solucions similars	7
Taula 3.1 Rols definits dins el joc	15
Taula 3.2 Exemple simplificat taula 'tarea'	41
Taula 3.3 Exemple càlcul de puntuació	43
Taula 3.4 Correspondència menú desplegable/camp de text	54

Nomenclatura

- VR Virtual Reality
- IDE Integrated Development Environment
- LTIM Laboratori de Tecnologies de la Informació Multimèdia.
- **UIB** Universitat de les Illes Balears.
- **HTML** Hypertext Markup Language.
- JS JavaScript
- PHP Hypertext Preprocessor
- VSC Visual Studio Code

Resum

En la actualitat la realitat virtual és una tecnologia molt innovadora i avançada. Tot i això, sembla que el software educatiu que es troba dins aquest paradigma és molt escàs, encara que les possibilitats són molt àmplies. Per aquest motiu aquest treball de fi de grau té l'objectiu principal de desenvolupar un software relativament educatiu de muntatge d'ordinadors en realitat virtual.

Per dur-ho a terme s'utilitzarà A-Frame, un *framework* de codi obert que permet desenvolupar aplicacions en realitat virtual sobre entorns webs, perquè el software sigui accessible per a tothom, des de la majoria de navegadors, i sense haver de descarregar cap arxiu.

El resultat final serà el propi joc, disponible a través d'una adreça web i accessible a través de qualsevol dispositiu de realitat virtual amb 6 graus de llibertat. Com a part del resultat també es troben les interfícies web destinades als dissenyadors del joc, perquè es puguin modificar amb facilitat les peces i les proves disponibles dins el joc. A més, també es recopilen estadístiques bàsiques de tots els usuaris, per tal de mostrar-les a la interfície web corresponent de manera ordenada i amb la possibilitat d'aplicar filtres.

Les conclusions i les experiències dels usuaris han estat satisfactòries perquè la gran majoria d'usuaris han pogut, després d'haver jugat, identificar les peces de l'ordinador i el seu lloc de col·locació, així com entendre la majoria d'especificacions i la seva importància.

Capítol 1. Introducció

Dins aquest primer capítol introductori s'explicarà la motivació de realitzar aquest projecte i en quin context es troba. A més a més, es definiran els objectius bàsics que es cerquen assolir durant el desenvolupament del TFG, així com els límits del projecte i com està estructurat el document.

1.1. Contextualització

Actualment el mercat dels jocs de realitat virtual està creixent a un ritme elevat. Però, són relativament pocs els jocs que són educatius o que pretenen ser d'utilitat dins un àmbit professional. La majoria d'aplicacions o jocs simplement tenen la finalitat de passar-ho bé dins un ambient més immersiu de l'habitual.

No obstant, podem trobar una excepció dins una categoria de videojocs: els simuladors. Molts de simuladors en realitat virtual són usats per fer pràctiques o per provar coses d'una manera més econòmica i segura que si es fes a la realitat. Com exemples trobem els simuladors de conducció, de vol, i fins i tot mèdics. Tots aquests simuladors es poden usar amb finalitats únicament lúdiques o també per la formació o l'aprenentatge de persones dins diversos àmbits.

El joc a desenvolupar és un simulador de muntar un ordinador per peces, que de fet pretén ser educatiu dins l'àmbit de la informàtica, i també lúdic ja que cerca ser divertit a l'hora de jugar-lo.

1.2. Motivació

La necessitat de dur a terme aquest projecte és que no existeix cap simulador similar en realitat virtual i pot ser una bona eina per introduir, de manera divertida, a persones que no entenen de hardware, alguns coneixements bàsics dels components d'un ordinador i com es duu a terme el procés de muntatge i d'instal·lació d'un PC. No és només útil per persones amb poc coneixements de hardware, ja que hi ha molts de llicenciats en informàtica que no són capaços de muntar un ordinador per ells mateixos.

Part de la motivació d'aquest projecte és també personal ja que m'agrada molt saber muntar els ordinadors per mi mateix. A més, recentment he provat un joc similar (més complex però sense realitat virtual), del qual es parlarà més endavant, que m'ha ajudat a adquirir més coneixements i he pensat que seria una bona idea desenvolupar una cosa similar, fent-la encara més educativa i immersiva.

També cal dir que investigant la idea del TFG s'han trobat diversos fils de Reddit [1] [2] [3] on els usuaris parlen precisament de si existeix un joc com el que es pretén desenvolupar

1. Introducció

en aquest TFG. Arriben a la conclusió de que no n'hi ha cap però que seria una molt bona idea per desenvolupar. No obstant, a aquest altre fil [4] es diu que podria ser estressant degut a la gran quantitat de detalls com cargols o petits components. Per sort, tal com s'explicarà a l'abast del projecte, això no suposarà un problema dins el nostre joc, ja que no es pretén desenvolupar un simulador híper realista. Aquests fils m'han motivat encara més a dur a terme aquesta idea i poder satisfer les necessitats de més usuaris com jo.

1.3. Objectius

L'objectiu principal és desenvolupar un simulador de construcció d'ordinadors que permeti als usuaris seleccionar els components d'un ordinador per tal de realitzar el muntatge de manera adequada. El títol del joc és PC Builder VR. Durant aquest procés de muntatge s'han de col·locar tots els components als llocs corresponents (CPU, GPU, discs durs, placa base...), sempre tenint en compte l'ordre de col·locació, la compatibilitat entre components (*sockets*, potència de la font i consum dels components) i completant unes tasques les quals canvien a cada nivell del joc per fer-lo més interactiu i augmentar la dificultat.

Tot el procés descrit anteriorment es realitza en realitat virtual i l'usuari es pot moure per l'oficina i agafar els components que desitgi per col·locar-los on consideri. Això fa que un altre objectiu sigui aprendre a desenvolupar software en realitat virtual i millorar els meus coneixements actuals de desenvolupament web.

El joc tindrà un tutorial per poder ajudar i guiar als usuaris més inexperts, i altres nivells de dificultat que requereixen que es completin certes tasques i que es compleixin certs requisits. De tots els nivells es recopilarà informació dels jugadors (data, temps i puntuació) per tal de disposar d'estadístiques que es podran mostrar dins el propi joc. Aquestes dades ajuden a saber quin nivell de coneixements tenen els usuaris sobre el muntatge d'ordinadors i en quines proves es cometen més errors. Per tant, un altre objectiu és aconseguir que el software desenvolupat sigui educatiu i permeti millorar els coneixements sobre el muntatge d'ordinadors a tots aquells usuaris que l'utilitzin.

Per últim, el darrer objectiu és fer que el contingut del joc sigui dinàmic, de manera que tant les peces, com les proves i les seves respectives tasques, es puguin editar còmodament des d'una interfície web, sense haver d'accedir al codi font.

1.4. Abast del projecte

El joc no cerca proporcionar una gran quantitat de components ja que el propòsit és introduir coneixements bàsics a usuaris novells en la matèria i entretenir als usuaris que juguin.

D'aquesta manera, el nombre de components serà bastant reduït (no hi haurà 20 tipus de processadors o targetes gràfiques) però el suficient perquè el joc sigui entretingut i no massa senzill. També es simplificaran certs aspectes d'alguns components, per exemple, si existeixen 6 tipus de plaques base (ATX, Micro ATX, LPX, BTX...) al joc només es tindran

en compte els 2 tipus més comuns (ATX i Micro ATX) per simplificar la programació i millorar l'experiència dels usuaris.

Tampoc es cercar replicar exactament la gran quantitat de connexions i ports que hi ha dins els components d'un PC ja que en realitat virtual no es té tanta precisió com a la vida real amb components que són molt petits, com podrien ser cargols, connexions minúscules a ports molt petits...

Tot i això, s'haurà de muntar l'ordinador correctament, col·locant cada component al seu lloc i seguint un ordre de col·locació (per exemple, no es pot col·locar la CPU sense haver col·locat primer la placa base) per tal de que l'ordinador funcioni.

Pel que fa als components/peces o a les proves, ambdues es podran definir i modificar pels dissenyadors de peces i proves, respectivament, mitjançant una interfície agradable i intuïtiva, evitant així que hagin de tenir coneixements de bases de dades o de informàtica avançada. Dins cada prova també es podran crear, editar i modificar les tasques, exceptuant 3 tasques referents a la quantitat de memòria RAM, HDD i SDD, les quals seran obligatòries a cada prova, incloent el tutorial. Aquestes tasques obligatòries determinaran la quantitat de memòria mínima de cada tipus que s'ha de muntar a un determinat nivell. Tant les tasques obligatòries com les creades pels dissenyadors determinen els atributs que han de tenir les peces que es col·loquin a l'ordinador (per exemple, muntar una CPU amb una freqüència superior a 3 GHz).

Per últim, referent a les limitacions del joc podem veure com l'única possible manera de jugar és mitjançant dispositius de realitat virtual que permetin moviment i interacció amb l'entorn (Oculus, per exemple), és a dir, visors amb 6-DoF o graus de llibertat. No es donarà suport a la realitat virtual més simple (3-DoF) o des de dispositius mòbils degut a la necessitat d'agafar objectes i interactuar amb ells dins l'escenari. No obstant, la majoria de funcions es podran realitzar des de un ordinador (sense realitat virtual), usant teclat i ratolí, tot i que el joc no estarà pensat perquè es jugui d'aquesta manera, podent provocar certs errors o empitjorar l'experiència dels usuaris.

1.5. Estructura del document

Aquest document es troba estructurat de la següent manera: les primeres pàgines tenen una funció informativa relativa a aquest document de TFG, com són els agraïments, la nomenclatura, els índexs de de figures i taules, entre d'altres. A continuació arribem a aquest capítol introductori, seguit de l'estat de l'art, el qual cerca especificar com està la tecnologia en quant al nostre tema, la realitat virtual.

Les següents seccions del document contenen la major part del contingut del TFG ja que especifiquen el desenvolupament del projecte i els seus respectius resultats. El nivell de detall del desenvolupament del projecte és molt elevat ja que s'explica com és la gestió del projecte, l'anàlisi, el disseny, la implementació, les proves i la instal·lació del mateix.

1. Introducció

Pel que fa als darrers capítols tracten de tancar el document amb un capítol de resultats, unes conclusions finals de tot el projecte i la bibliografia usada durant la realització d'aquest treball.

Capítol 2. Estat de l'art

Aquest capítol descriu les tècniques i solucions existents que seran d'utilitat de cara al desenvolupament del projecte. Primer es començaran explicant les solucions similars al software que es vol crear i després es comentaran i s'organitzaran les possibles tècniques per dur a terme el projecte.

2.1. Solucions existents a problemes semblants

Existeixen algunes solucions que s'assemblen a la meva idea, tot i que no són exactament iguals al que es pretén desenvolupar en aquest TFG. Primer s'explicaran breument cada una de les solucions i després s'organitzaran en una taula comparativa per tal d'analitzar les funcions i què es pot extreure de cada una d'aquestes solucions. Finalment, es separaran les solucions segons si tenen VR o si són específicament de muntar ordinadors, per tal de treure conclusions finals de cada una de les temàtiques i veure com ens poden ser d'utilitat de cara al desenvolupament del problema.

Les solucions existents trobades són les següents: en primer lloc trobem el PC Building Simulator¹, un joc sense realitat virtual del qual es va obtenir parcialment la idea per realitzar aquest TFG. Tracta de reparar els ordinadors que els clients duen a la oficina, comprant els components romputs i muntant els ordinadors de nou. El nivell de detall és molt elevat degut a la gran quantitat de components i connexions. A la Figura 2.1 es pot veure el joc i els gràfics que té. L'estètica és similar al que es cerca desenvolupar en aquest TFG.

En segon lloc, al canal de YouTube de AMD, es mostra un vídeo d'un joc² en VR que reflexa pràcticament la idea que es vol desenvolupar. No obstant, pareix un software privat perquè no s'ha aconseguit trobar-lo per internet i només s'usa per fer màrqueting d'un ordinador nou de la pròpia empresa.



Figura 2.1. PC Building Simulator



Figura 2.2. Joc VR de AMD

¹ <u>https://store.steampowered.com/app/621060/PC Building Simulator/</u>

² <u>https://www.youtube.com/watch?v=J96W-VooWZE</u>

2. Estat de l'art

En tercer lloc podem situar tres jocs en VR que no tracten de muntar ordinadors: Job Simulator³, VR Office Experience⁴ i edataconsulting VR Office⁵. El primer dels tres és un dels primers jocs en VR que van sorgir i tracta, tal i com es pot veure a la Figura 2.3, de realitzar tasques pròpies d'una persona que està treballant a una oficina (apuntar a un paper, contestar el telèfon, moure's per l'oficina...). Els altres dos jocs són molt simples i permeten desplaçar-se dins una oficina i interactuar amb diversos objectes en VR com ordinadors o pissarres, entre d'altres. A la Figura 2.4 es pot veure un dels jocs que permeten interactuar amb objectes d'oficina.



Figura 2.3. Job Simulator

Figura 2.4. edataconsulting VR Office

En quart lloc cal destacar el PC Repair Simulator⁶, un prototipus poc funcional desenvolupat en Unity que tracta de reparar un ordinador. És similar al que es vol desenvolupar a aquest TFG però de manera molt simplificada i amb una estètica molt bàsica, com es mostra a la Figura 2.5.



Figura 2.5. PC Repair Simulator



Figura 2.6. TestOut PC Pro

En darrer lloc trobam un software educatiu anomenat TestOut PC Pro⁷. Aquest software permet als estudiants aprendre els components i les connexions d'un ordinador de

³ <u>https://store.steampowered.com/app/448280/Job Simulator/</u>

⁴ <u>https://store.steampowered.com/app/1657860/VR_Office_Experience/</u>

⁵ https://store.steampowered.com/app/1129550/edataconsulting VR Office/

⁶ <u>https://t3kmast3r.itch.io/pc-repair-simulator-2020</u>

⁷ <u>https://w3.testout.com/courses/pc-pro</u>

manera molt detallada. També ofereix certificacions si es superen els seus cursos. A la Figura 2.6 es pot veure un exemple del software TestOut PC Pro i del nivell de detall de les connexions.

Per tal d'ordenar totes aquestes solucions i extreure les conclusions adients s'han organitzat totes les solucions en la Taula 2.1 i s'han indicat, de cada una, una sèrie d'atributs clau (si la solució es troba en realitat virtual, si es pot interactuar amb objectes...) per facilitar la tasca organitzativa.

Solució	VR	Interacció objectes VR	Muntar PC	Motor gràfic	Tipus d'aplicació
PC Building Simulator	No	No	Sí	Unity	Joc/Simulador
Vídeo AMD	Sí	Sí	Sí	No trobat	Simulador
Job Simulator	Sí	Sí	No	Unity	Joc/Simulador
VR Office Experience	Sí	No	No	Unity	Simulador
Edataconsulting VR Office	Sí	Sí	No	Unity	Simulador
PC Repair Simulator	Sí	Sí	Sí	Unity	Prototipus de simulador
TestOut PC Pro	No	No	Sí	No trobat	Educativa

Taula 2.1. Comparació de solucions similars

Per treure conclusions d'aquesta taula anterior es separaran les solucions en 2 seccions. La primera tractarà sobre aquelles solucions que tenen a veure amb el muntatge d'ordinadors, siguin o no en VR. La segona, en canvi, tractarà totes les altres solucions que si que tenen VR però que no es basen en el muntatge d'ordinadors, ja que encara així ens poden resultar útils degut a que el jugador es troba immers dins una oficina en VR. És a dir, hem separat les dues seccions segons la columna "Muntar PC" de la Taula 1.1.

2.1.1. Solucions de muntatge d'ordinadors

Aquestes solucions ens poden resultar útils pel que fa a la part tècnica de la idea en qüestió ja que, per exemple, es poden agafar idees del PC Building Simulator i de com es gestionen les compatibilitats entre components o de quines tasques s'han de dur a terme.

2. Estat de l'art

El vídeo de AMD i el PC Repair Simulator segurament no ens resultin de gran ajuda perquè el primer dels dos no es pot provar de manera oberta i el segon és un prototipus molt poc funcional. Tot i així, el PC Repair Simulator presenta un avantatge enfront del PC Building Simulator perquè aquest primer permet realitat virtual i ens pot ajudar a veure com s'han modelat determinats components.

Per acabar, el TestOut PC Pro és una eina molt potent ja que pretén educar a estudiants d'enginyeria, però no es farà servir perquè requereix una llicència de pagament.

2.1.2. Solucions en VR en entorns d'oficina

Les 3 solucions restants són el Job Simulator, VR Office Experience i edataconsulting VR Office. Tot i que aquestes solucions no aporten res a l'apartat tècnic de muntatge d'ordinadors, compatibilitats, connexions... poden servir per agafar idees de cara al món virtual que s'ha de desenvolupar en VR. Les 3 solucions tenen VR i dues d'elles permeten interactuar amb objectes del joc, cosa que resulta de gran utilitat a l'hora de dur a terme el modelat de l'entorn d'oficina que es pretén desenvolupar.

2.2. Anàlisi i selecció de eines per resoldre el problema

Pel que fa a les eines utilitzades podem destacar tres tipus d'eines: IDEs, tecnologies gràfiques i eines per tractar les compatibilitats entre els components (CPUs, discs durs, targetes gràfiques...) que formaran part del joc.

Amb la finalitat d'explicar millor cada un dels tipus d'eines les hem dividit en tres seccions diferents. A cada una d'elles s'explicaran les opcions disponibles i quines s'usaran finalment pel desenvolupament del projecte.

2.2.1. IDEs

Dins l'àmbit de programació existeixen multitud de IDEs que faciliten la tasca d'escriptura de codi. L'elecció d'un IDE sobre un altre moltes vegades depèn del tipus de llenguatge de programació a usar, de les limitacions que presenten alguns IDEs en concret, o, fins i tot, preferències personals.

En aquest cas s'ha seleccionat un dels entorns de programació més populars, Visual Studio Code (VSC a partir d'ara). Aquest IDE permet programar en pràcticament qualsevol llenguatge de programació i té diverses extensions que faciliten la tasca de programació. A més a més, durant el desenvolupament del projecte es programarà en HTML usant el *framework* de codi obert A-Frame⁸ (explicat a la següent secció), del qual VSC té una extensió que permet una millor integració. No només s'ha seleccionat VSC degut a aquesta integració amb A-Frame sinó que personalment em trobo molt còmode usant aquest IDE ja que l'he usat per realitzar moltes tasques de programació.

⁸ <u>https://aframe.io/</u>

2.2.2. Tecnologies gràfiques

La majoria de jocs, ja siguin en VR o sense, estan desenvolupats usant pràcticament els mateixos motors gràfics. Aquests motors gràfics solen ser Unity⁹ o Unreal Engine¹⁰. Unity és el més antic dels dos i és usat en aplicacions com Blender o Adobe Photoshop. Unreal Engine en canvi, és un motor desenvolupat per Epic Games i usat majoritàriament en jocs de dispars tot i que també s'usa en MMORPG i RPG.

L'altre cara de la moneda pel que fa a tecnologies gràfiques és la programació en entorns web. Aquí podem introduir A-Frame, que és una biblioteca de JavaScript que permet, mitjançant codi obert, crear aplicacions en realitat virtual. La peculiaritat d'aquestes aplicacions és que funcionen sobre HTML per lo que són accessibles mitjançant qualsevol navegador modern.

La meva experiència amb aquests motors gràfics o A-Frame és nul·la però m'agrada molt el desenvolupament d'aplicacions web en HTML i la facilitat d'ús de tot el contingut web. A més, sabia de l'existència de Unity i Unreal Engine però no de A-Frame i, per aquesta raó, pens que és una bona oportunitat per començar a conèixer aquesta tecnologia i desenvolupar un projecte tan important com és aquest TFG.

S'ha plantejat la idea d'usar Blender¹¹ per millorar els gràfics del joc i realitzar textures personalitzades per als components, però com que la finalitat del joc no és l'aparença gràfica sinó les funcionalitats s'ha decidit que no s'usarà Blender i que els components tindran una aparença més simple.

2.2.3. Compatibilitat i especificacions dels components

Tot i que no es pretén implementar una gran quantitat de components dins el joc s'ha de gestionar el tema de compatibilitats entre ells, ja que hi ha components que no són compatibles entre si, i això s'haurà de veure reflectit en el joc. A més, les especificacions dels components que s'incloguin han de ser reals i això també afecta al joc. Per exemple, si una placa base té 4 ports HDD, al joc només s'han de poder posar 4 HDD damunt aquesta placa base.

Per tractar aquest tema es farà ús de PCPartPicker¹², una pàgina web que és com una gran base de dades amb informació de pràcticament tots els components que existeixen al mercat dels ordinadors. De cada component s'indiquen les seves especificacions i també la compatibilitat amb altres components (per exemple, segons el *socket* o el consum dels components). Aquesta web s'usarà per incloure els components desitjats a la base de

⁹ <u>https://unity.com</u>

¹⁰ <u>https://www.unrealengine.com</u>

¹¹ <u>https://www.blender.org/</u>

¹² <u>https://pcpartpicker.com</u>

2. Estat de l'art

dades pròpia i així poder determinar de manera exacta les especificacions i compatibilitats de cada component.

	System Builder											
		Over	rview Prices By Merchant									
& htt	ps://pcpartpicke	.com/list/LfN7Mc	Markup:	\$	ŧ	() ()	г	BB	🗗 History	스 Save As	+ Edit This P	art List
🕝 Comp	patibility: No issue	es or incompatibilities found.								🗲 Estin	ated Wattage:	309W
Component	Selection			E	lase	Pro	mo	Shipping	Tax F	Price	Where	
CPU		AMD Ryzen 7 2700X 3.7 GHz 8-Core Processor		5	\$189.99			FREE		\$189.99	BH	Buy
Motherboard		ASRock B450 Steel Legend ATX AM4 Motherboard		ę	\$108.99			\P rime	—	\$108.99	amazon.com	Buy
Memory		G.Skill Trident Z RGB 16 GB (2 x 8 GB) DDR4-3200 Me	emory	5	\$98.99			FREE	—	\$98.99	(constant)	Buy
Storage	2	Western Digital Caviar Blue 1 TB 3.5" 7200RPM Intern	al Hard Drive	1	\$39.99			FREE	—	\$39.99	Adorama	Buy
Video Card		EVGA GeForce GTX 1650 SUPER 4 GB SC ULTRA GAM	ING Video Card	1	\$184.48			\P rime	—	\$184.48	amazon.com	Buy
Case	4	Deepcool MATREXX 50 ADD-RGB 4F ATX Mid Tower Ca	ase	5	\$72.99			\$7.99		\$80.98	(energy	Buy
Power Supply	۲	EVGA SuperNOVA G3 650 W 80+ Gold Certified Fully N	Modular ATX Power Supply	5	\$109.99					\$109.99		Buy
Operating System		Microsoft Windows 10 Home OEM 64-bit		1	\$106.99			FREE	—	\$106.99		Buy

Figura 2.7. Captura de pantalla de PCPartPicker

A més, s'ha trobat una API no oficial d'aquesta web [5] (la documentació de la web és privada) desenvolupada per un tercer, la qual permet realitzar cridades demanant la relació de compatibilitat entre components. Si es volgués ampliar la quantitat de components disponibles dins el joc es podria fer ús d'aquesta API per facilitar la tasca de programació i evitar la creació d'una base de dades pròpia referent a les compatibilitats dels components. No obstant, segurament s'optarà per tenir una base de dades pròpia, així es poden crear, editar i eliminar components segons es vulgui i es pot tenir un control major sobre els components que hi ha dins el joc.

Capítol 3. Desenvolupament del projecte

Aquest capítol és el cor del treball ja que desenvolupa, per passes, tot el projecte. Es començarà detallant aspectes relacionats amb la gestió de projectes com la planificació temporal, els riscs, la metodologia... Després, es continuarà realitzant un anàlisi de requisits, el disseny de la solució, la seva respectiva implementació i les proves realitzades.

3.1. Gestió del projecte

Dins aquest apartat es descriu de quina manera es gestionarà el desenvolupament d'aquest TFG, és a dir, s'explicarà quina metodologia referent a projectes s'aplicarà, quina planificació es seguirà pel que fa a les dates i es comentaran alguns aspectes relacionats amb els riscs, els interessats i els costs.

3.1.1. Metodologia de desenvolupament

Hi ha diverses metodologies a l'hora de desenvolupar un projecte, des de les més conegudes com cascada i àgil, fins a les més específiques com prototipus, cadena crítica i híbrida. Degut a les circumstàncies tan variables que tindrem durant la realització d'aquest TFG (canvis d'idees, afegiment de funcionalitats, dates molt canviants degut a classes, exàmens i treballs, etc...) s'ha decidit que la metodologia més adequada és la àgil.

La metodologia àgil és molt flexible i es va adaptant a les circumstàncies del projecte ja que permet anar modificant el producte, en aquest cas el joc, durant tot el transcurs del TFG. D'aquesta manera es podrà anar desenvolupant el joc en qüestió i es generarà valor de manera contínua mitjançant prototipus incrementals. A més, en el cas de que sigui necessari, es poden modificar els requisits segons pertoqui a mesura que s'avanci en el desenvolupament del joc.

Durant el desenvolupament també es realitzaran proves les quals introduiran canvis en el joc i derivaran en la modificació o creació de nous requisits. Aquesta pràctica és clau dins la metodologia àgil perquè permet anar canviant els requisits a mesura que es troben errors o s'afegeixen funcionalitats.

Pel que fa a les eines utilitzades es farà ús d'un gestor de tasques en línia anomenat Todoist¹³, el qual servirà per anar definint tasques relacionades amb el desenvolupament del TFG.

¹³ https://todoist.com

3. Desenvolupament del projecte

3.1.2. Planificació temporal

La planificació temporal d'aquest projecte és un poc complicada de definir ja que, com s'ha comentat a l'apartat anterior, s'usarà la metodologia àgil perquè és difícil seguir una planificació concreta i inamovible. Tot i això, es té una idea aproximada de com es vol planificar la realització d'aquest TFG pel que fa a les dates.

Durant el primer semestre del curs 2021-2022 no es pretén realitzar massa feina degut a la gran càrrega de treball que hi ha en aquest darrer any de carrera. No obstant, es començaran a redactar els primers apartats generals d'aquest document i, cap al final de l'any 2021, es començaran a fer proves experimentals amb A-Frame, el qual és necessari dominar per dur a terme el treball.



Figura 3.1 Cronograma previst

Com es pot veure a la Figura 3.1 la major part de la feina es realitzarà durant la primera meitat del 2022 perquè s'ha previst que la càrrega de treball serà molt menor ja que només s'han de cursar les dues darreres assignatures del grau. Per això, es preveu que la disposició del cronograma serà la següent, tot i que s'anirà adaptant a les circumstàncies.

- Novembre Gener: Redactar els primers apartats del document i definició dels objectius.
- **Febrer:** Començar a redactar els primers requisits, a mode de pluja d'idees.
- Finals de febrer: Començar el desenvolupament software del joc (configuracions inicials, servidor, base de dades i menús) usant A-Frame i el servidor proporcionat per LTIM.
- Març: Desenvolupar la interacció en realitat virtual. És a dir, poder agafar objectes, afegir físiques al joc i poder amollar objectes a una determinada zona i que es col·loquin.

- **Finals de març**: Al acabar amb la primera part del desenvolupament es dissenyarà tot el tema dels components, la compatibilitat entre ells i els nivells de dificultat de cada un dels nivells.
- Abril: Acabar el desenvolupament de la interacció i programar el software (pàgines web) per poder gestionar els components, les proves i les estadístiques. A finals d'abril es pretén tenir el joc relativament acabat i que només falti augmentar el nivell de components i proves, i aplicar millores perquè el joc tingui més valor.
- **Maig i juny:** Aplicar les darreres millores referents a la programació del joc i continuar amb la redacció de tot el document, deixant-lo enllestit cap a principis de juny.

No obstant, s'ha de comentar que durant el desenvolupament software també s'anirà documentant, en la mesura de lo possible, el que es vagi fent perquè, quan s'hagi de redactar aquest document, la tasca sigui més senzilla.

3.1.3. Gestió dels interessats

Els únics interessats d'aquest treball som n'Antoni Oliver, el qual és el meu tutor del TFG, en Toni Bibiloni, el segon tutor del TFG, i jo. El nostre interès radica en dur a terme aquest projecte de la millor manera possible per tal d'aprendre, en el meu cas, la temàtica de la realitat virtual demostrar la capacitat de desenvolupar un projecte que reflecteixi tot el que he aprés a la carrera, i en el cas de n'Antoni i en Toni per poder ajudar-me al màxim i que els resultats finals obtinguts d'aquesta feina siguin millors als esperats.

3.1.4. Gestió de riscs

La realització d'un treball d'aquestes característiques pot comportar nombrosos riscs, els quals estan detallats a continuació. En cada cas s'indicarà el procés decidit per tal de prevenir, minimitzar o acceptar els riscs.

En primer lloc, no es disposa de cap dispositiu de realitat virtual per lo que a l'hora de programar no es poden veure els resultats (només es veuen a l'ordinador però no s'assembla massa al que es veu en VR). És cert que el laboratori de LTIM disposa d'unes Oculus Rift que es poden usar, però això implica haver d'estar físicament a la UIB programant. Es tractarà d'adquirir un visor de realitat virtual per tal de prevenir aquest risc però degut al preu elevat d'aquests dispositius potser no sigui possible. De totes maneres, per minimitzar aquest risc s'intentarà programar en blocs, de manera que tot el que requereixi visualitzar-se amb VR es programarà seguit aprofitant així les Oculus de la UIB.

Un altre risc destacable és que no es pugui trobar un sistema de físiques i de interacció que funcioni correctament. Aquest joc requereix que es puguin agafar i amollar objectes i que aquests objectes tinguin físiques associades. Sense això és pràcticament impossible desenvolupar el joc com s'ha pensat. Aquest risc no es pot prevenir però si que es pot

3. Desenvolupament del projecte

tractar de minimitzar, recorrent a fòrums com Stack Overflow¹⁴ o GitHub¹⁵, on hi ha molts de programadors que poden haver tingut problemes semblants. També es parlarà amb el tutor per si ell sap solucionar-ho o de quina manera es pot modificar el joc perquè es pugui jugar sense físiques ni interacció. Si no s'aconsegueix trobar una solució el risc s'hauria d'acceptar, cosa molt greu perquè, tal i com s'ha explicat, pot implicar que el joc no es pugui desenvolupar, suposant això haver de pensar una nova idea de TFG o almenys repensar-la.

Tots aquests riscs suposarien un retard en la planificació temporal ja comentada, però, almenys el primer risc, es podria minimitzar augmentant la càrrega de treball i programant més a la UIB, o també es podria evitar si s'aconsegueix adquirir un visor de VR. Tot i així, en el cas de no poder lliurar el treball a temps (abans del 8 de juliol de 2022) es lliurarà a la segona entrega programada pel setembre. No obstant, no és probable que això passi ja que la planificació temporal contempla petites desviacions per riscs.

3.1.5. Gestió econòmica

Aquest apartat s'ha inclòs al document perquè es preveu que hi ha haurà alguns costs durant el transcurs del treball. Aquests costs venen derivats de la problemàtica de no disposar d'un visor de realitat virtual propi. És cert que la UIB disposa d'un visor VR (Oculus Rift) al departament de LTIM per poder anar provant el software desenvolupat, però moltes vegades per poder progressar ràpidament es necessita usar un visor VR constantment perquè allò que es veu a la pantalla, no té res a veure amb el que es veu dins VR (mida, perspectiva, resolució, interaccions, desplaçament...). S'ha barallat la possibilitat de realitzar gran part del desenvolupament a la UIB, per així poder-ho provar directament en VR, però la comoditat i la potència de l'ordinador de casa no es pot comparar.

Per aquesta raó s'ha pensat que la millor solució és comprar un visor de VR relativament barat (pressupost màxim de 200€). Per triar un dispositiu el més important és que sigui compatible amb A-Frame, cosa que es pot consultar a l'apartat de dispositius¹⁶ de la documentació d'A-Frame. A més, el dispositiu ha de tenir 6 graus de llibertat perquè es permeti el desplaçament i la interacció. Això fa que les possibles opcions siguin les següents: HTC Vive, Oculus Rift, Oculus Quest 1 i Oculus Quest 2.

A partir d'aquest moment s'ha començat a fer una cerca per als portals web de segona mà més comuns (eBay, Wallapop i Milanuncios) per tractar de trobar alguna oferta. Finalment s'han trobat unes Oculus Quest 1 per 120€ a eBay i s'ha decidit que el preu era molt assequible i per tant s'han adquirit.

¹⁴ <u>https://stackoverflow.com/</u>

¹⁵ <u>https://github.com/</u>

¹⁶ <u>https://aframe.io/docs/1.3.0/introduction/vr-headsets-and-webxr-browsers.html</u>

Aquesta compra es podria considerar una inversió perquè, gràcies a poder provar el software constantment en VR, es preveu evolucionar i aprendre a un ritme molt elevat, podent així desenvolupar un software molt complet i funcional.

3.2. Anàlisi

És necessari realitzar un anàlisi abans de dur a terme el disseny de la solució, el qual tractarà sobre els possibles usuaris i els seus respectius rols dins el joc i, sobretot, es recopilaran els requisits de tot el sistema que es desenvoluparà en els següents apartats. Aquest darrer apartat és fonamental si es vol realitzar un projecte complet, sense manca de funcionalitats i que permet fer proves per demostrar-les.

3.2.1. Usuaris

Pel que fa als usuaris, s'han de diferenciar 3 rols diferents dins el joc, els quals s'expliquen a la taula següent.

Rol	Descripció
Jugador	Persona que juga de manera trivial. La majoria d'usuaris tenen aquest rol. Només té accés al joc en VR i a l'apartat d'estadístiques.
Dissenyador de peces	Administradors que poden afegir, eliminar o modificar peces dins el joc. De cada peça es defineixen totes les seves característiques i especificacions. No tenen accés a la zona de proves.
Dissenyador de proves	Administradors que poden definir, editar i eliminar proves del joc, assignant el nivell de dificultat i les tasques específiques. No poden accedir a la zona de peces.

Taula 3.1 Rols definits dins el joc

S'ha realitzat aquesta distinció de rols amb privilegis (dissenyador de peces i dissenyador de proves) perquè els dissenyadors de proves no puguin modificar les peces del joc i a l'inrevés. Això fa que, si es produeixen errades de configuració, estiguin limitades dins un únic camp (peces o proves) i que no es pugui produir una errada molt greu que afecti als dos camps.

3.2.2. Requisits

La recopilació de requisits s'ha realitzat mitjançant una pluja d'idees per tenir una primera aproximació de tots els requisits que es volen incloure i, posteriorment, s'han realitzat múltiples etapes iteratives per decidir, amb l'ajuda del tutor, quins són els requisits definitius. El resultat final dels requisits s'exposa a aquest apartat organitzat segons els tipus de requisits.

3. Desenvolupament del projecte

3.2.2.1. Requisits funcionals

Els requisits funcionals s'han agrupat segons els diferents rols (explicats a l'apartat anterior i exposats a la Taula 3.1) que es contemplen al joc.

3.2.2.1.1. Jugador

- **RF01.** Un jugador ha de poder realitzar el tutorial del joc.
- **RF02.** Un jugador ha de poder triar el nivell del joc al que vol jugar.
- **RF03.** Un jugador ha de poder registrar-se introduint un usuari i una contrasenya.
- **RF04.** Un jugador ha de poder iniciar sessió amb el seu usuari i la seva contrasenya.
- **RF05.** Un jugador ha de poder canviar el seu nom d'usuari.
- **RF06.** Un jugador ha de poder seleccionar un avatar, d'un conjunt de 5 avatars, per afegir al seu perfil.
- **RF07.** Un jugador ha de poder jugar de manera anònima o només amb un nom d'usuari genèric (sense compte registrat).
- **RF08.** Un jugador ha de poder visualitzar les estadístiques de tots els jugadors per a cada nivell de joc. Les estadístiques són les següents: nom, prova, data, temps empleat i puntuació.
- **RF09.** Un jugador ha de poder filtrar les estadístiques per prova i tipus d'usuari (registrat o no registrat).
- **RF10.** Un jugador ha de poder cercar les estadístiques d'un determinat usuari mitjançant un cercador.
- **RF11.** Un jugador ha de poder ordenar les estadístiques segons l'usuari, la prova, la data, el temps empleat o la puntuació.
- **RF12.** Un jugador ha de poder visualitzar el nombre de visites, per dia, al joc.
- **RF13.** Un jugador ha de poder visualitzar el nombre de visites, segons el país, al joc.
- **RF14.** Un jugador ha de poder centrar la càmera a la zona inicial.
- **RF15.** Un jugador ha de poder canviar el volum de la música.

3.2.2.1.2. Dissenyador de peces

- **RF16.** Un dissenyador de peces ha de poder definir noves peces indicant els següents camps: nom, descripció, tipus de peça, dimensions, textura, marca, model i les dades específiques segons el tipus de peça.
- **RF17.** Un dissenyador de peces ha de poder gestionar (editar i eliminar) les peces existents.

3.2.2.1.3. Dissenyador de proves

- **RF18.** Un dissenyador de proves ha de poder crear proves noves, especificant el nom, descripció, dificultat i les seves tasques.
- **RF19.** Un dissenyador de proves ha de poder gestionar (editar i eliminar) les proves existents.
- **RF20.** Un dissenyador de proves ha de poder crear, editar i eliminar, les tasques d'una determinada prova.

RF21. Un dissenyador de proves ha de poder visualitzar el nombre de peces compatibles amb cada tasca d'una determinada prova.

3.2.2.2. Requisits No Funcionals

- **RNF01.** Les contrasenyes dels jugadors i de les comptes amb privilegis s'encriptaran usant la funció de hash bcrypt¹⁷.
- **RNF02.** El joc funcionarà amb qualsevol dispositiu de realitat virtual que tingui 6-DoF o graus de llibertat.
- **RNF03.** El joc no serà compatible amb l'ús de teclat i ratolí, tot i que la majoria de funcions bàsiques funcionin.
- **RNF04.** El joc serà compatible amb tots els navegadors que suportin la API de WebXR.
- **RNF05.** El joc serà accessible des de qualsevol lloc del món, amb l'únic requisit de que la connexió a internet sigui estable.
- **RNF06.** Només es recopilaran les següents dades de cada partida que juguin els usuaris: nom, prova, data, temps empleat i puntuació obtinguda. Aquestes dades només s'usaran per mostrar les estadístiques a una pàgina web pública.
- RNF07. La puntuació global (0-10) es calcularà segons el temps empleat i el nombre d'errades. El temps empleat es defineix com un valor de 0 a 10, el qual es descompta de la màxima puntuació (10), i es calcula segons els valors mínims i màxims (2 min 15 min) decidits arbitràriament. A més, hi ha 3 tipus d'errades: tasca no completada, col·locació errònia del component i error de compatibilitat tècnica (*socket* no compatible o potència de la font insuficient). Les errades descompten 0.5, 1 i 1 punt de la puntuació global, respectivament.

3.2.2.2.1. Requisits No Funcionals d'interacció

- **RNF08.** Per seleccionar botons s'ha d'apuntar amb la mà que té el làser i fer clic amb el gatell corresponent.
- **RNF09.** Per agafar els components s'ha d'atracar la mà virtual del jugador al component i pressionar el botó d'agafar (botó lateral dels controladors).
- **RNF10.** Per amollar els components s'ha d'amollar el botó d'agafar.
- **RNF11.** Per col·locar els components s'han d'amollar a algun lloc de l'ordinador on es marqui d'un color vistós quan s'atraca el component.
- **RNF12.** Per centrar la càmera al punt inicial s'ha de prémer el botó X del controlador esquerre.
- **RNF13.** Per obrir el menú de música s'ha de prémer el botó Y del controlador esquerre.

3.3. Disseny

El projecte requereix d'una fase de disseny, la qual tracta de definir i exposar el model de dades i el disseny arquitectònic que s'usaran com a entrades de la fase d'implementació,

¹⁷ <u>https://en.wikipedia.org/wiki/Bcrypt</u>

3. Desenvolupament del projecte

on es desenvoluparà el joc en si. A més, també s'explicarà com s'han dissenyat les tasques de cada prova, perquè s'entengui millor el model de dades. És important que el disseny sigui robust perquè no s'hagi de modificar durant les fases posteriors.

3.3.1. Disseny de les tasques

Les tasques són una de les parts més importants del joc perquè són las que fan que cada prova del joc sigui completament diferent, podent augmentar o disminuir la dificultat de la mateixa. Sense les tasques el joc únicament es basaria en muntar l'ordinador sempre de la mateixa manera, per lo que seria molt monòton.

Per això, les tasques de cada prova funcionen de la següent manera. En primer lloc, el nombre màxim de tasques per prova és 10, per evitar que les proves siguin excessivament complexes. En segon lloc, les 3 primeres tasques de cada prova (tasques d'emmagatzematge) no es poden eliminar i fan referència a la quantitat de memòria RAM, HDD i SDD mínima requerida. La funció d'aquestes 3 tasques és permetre que els dissenyadors indiquin la capacitat mínima d'aquests tipus de peça perquè és una cosa molt comú a l'hora de muntar un ordinador, i perquè és l'única manera de definir aquest paràmetre així com estan dissenyades la resta de les tasques. En tercer lloc, la resta de les proves indiquen, mitjançant operadors lògics (=, !=, >, >=, <, <= i BETWEEN), el valor o valors que ha de complir un determinat atribut d'un tipus de peça.

Per aquesta raó, al model de dades descrit a l'apartat 3.3.2 cada tasca té un camp específic anomenat "almacenamiento" que indica si la tasca és d'emmagatzematge o no. En cas de que no ho sigui, es passarien a mirar la resta d'atributs ("atributo", "accion", "valor1" i "valor2") per mirar quin operador lògic s'ha d'aplicar a l'atribut per comparar amb el valor o els valors especificats. El procés de comprovació del compliment de les tasques es troba explicat a la implementació, concretament a l'apartat 3.4.1.10.

3.3.2. Model de dades

Aquest apartat pretén descriure el model de dades i el procediment seguit per arribar a ell, començant pel model conceptual i la transformació al model relacional. Tots aquests elements es descriuen en profunditat als següents apartats.

3.3.2.1. Model conceptual

El model conceptual s'ha dissenyat tenint en compte el procés de creació de proves i peces, perquè permet esbrinar quines eren les classes i les relacions entre elles.

El resultat, que es pot comprovar a la Figura 3.2, es compon de 6 classes ("Usuario", "Estadistica", "Prueba", "Tarea", "TipoPieza" i "Pieza"), a més de 6 classes que hereten de "Pieza" i que contenen diferents atributs segons el tipus de peça. Aquesta herència és del tipus {M, XOR} ja que és obligatori que una peça pertanyi a algunes de les 6 classes que hereten de "Pieza" i, a més, només poden pertànyer a 1 de les classes perquè no pot existir una peça que sigui, per exemple, una CPU i una GPU alhora.


Figura 3.2 Model de dades

3.3.2.2. Model relacional

Una vegada es tenen totes les classes amb els seus atributs s'ha de traduir del model conceptual al relacional. Per això, primer s'ha de definir una clau primària, marcada en negreta a la Figura 3.2 i subratllada al llistat següent. Després, cada una de les relacions del model conceptual esdevé una nova taula del model relacional, amb les claus de les de les dues classes com atributs d'aquesta nova taula.

USUARIO (<u>username</u>, pwHash, idAvatar, tipoUsuario) ESTADISTICA (<u>idEstadistica</u>, fecha, tiempo, puntuacion, usuarioRegistrado, userEstadistica) PRUEBA (<u>idPrueba</u>, nomPrueba, descrPrueba, dificultad) TAREA (<u>idTarea</u>, descrTarea, almacenamiento, atributo, accion, valor1, valor2) PIEZA (<u>idPieza</u>, nomPieza, descrPieza, width, height, depth, textura, marca, modelo) TIPOPIEZA (<u>idTipoPieza</u>, nomTipoPieza) CPU (<u>idCPU</u>, arquitectura, socketCPU, nucleos, frecCPU, frecTurbo, consumoCPU) GPU (<u>idGPU</u>, memoria, vram, consumoGPU) RAM (<u>idRAM</u>, estandar, velocidadRAM, capacidadRAM) DISCO (<u>idDisco</u>, tipoDisco, velocidadDisco, capacidadDisco) PLACABASE (<u>idPlacaBase</u>, tipoPlacaBase, socketMBCPU, slotsRam, slotsHDD, slotsSDD) FUENTE (<u>idFuente</u>, tipoFuente, potencia) VENTILADOR (<u>idVentilador</u>, velocidadVentilador)

R_USUARIO_ESTADISTICA (username, <u>idEstadistica</u>) R_ESTADISTICA_PRUEBA (<u>idEstadistica</u>, idPrueba) R_PRUEBA_TAREA (idPrueba, <u>idTarea</u>) R_TAREA_TIPOPIEZA (<u>idTarea</u>, idTipoPieza) R_PIEZA_TIPOPIEZA (<u>idPieza</u>, idTipoPieza)

Finalment, s'estudia la possibilitat de fusionar algunes de les taules anteriors. Només es poden fusionar aquelles taules que tinguin la mateixa clau primària, i s'han d'ajuntar tots els atributs a la taula fusionada resultant. En aquest cas es poden fusionar totes les taules.

Les taules resultants després de la fusió són les següents (s'indiquen en cursiva les claus foranes).

USUARIO (username, pwHash, idAvatar, tipoUsuario) ESTADISTICA (idEstadistica, fecha, tiempo, puntuacion, usuarioRegistrado, userEstadistica, username, idPrueba) PRUEBA (idPrueba, nomPrueba, descrPrueba, dificultad) TAREA (idTarea, descrTarea, almacenamiento, atributo, accion, valor1, valor2, idPrueba, idTipoPieza) PIEZA (idPieza, nomPieza, descrPieza, width, height, depth, textura, marca, modelo, idTipoPieza) TIPOPIEZA (idTipoPieza, nomTipoPieza) **CPU** (idCPU, arquitectura, socketCPU, nucleos, frecCPU, frecTurbo, consumoCPU) **GPU** (idGPU, memoria, vram, consumoGPU) RAM (idRAM, estandar, velocidadRAM, capacidadRAM) DISCO (idDisco, tipoDisco, velocidadDisco, capacidadDisco) PLACABASE (idPlacaBase, tipoPlacaBase, socketMBCPU, slotsRam, slotsHDD, slotsSDD) FUENTE (idFuente, tipoFuente, potencia) VENTILADOR (idVentilador, velocidadVentilador)

3.3.3. Disseny arquitectònic

Per tal d'explicar el conjunt de subsistemes que composen tot el sistema principal i com es relacionen entre si s'usarà un diagrama que reflecteix el disseny arquitectònic de tot el sistema. Aquest disseny es pot veure a la Figura 3.3 i s'explicarà amb detall a continuació.

Tot el sistema es troba sustentat pel servidor Apache, el qual emmagatzema tots els mòduls i subsistemes. Aquest servidor té configurada una base de dades de MySQL que s'encarrega de guardar totes les taules explicades a l'apartat 3.3.2 del Model de dades. Tot el back-end es controla mitjançant codi PHP que gestiona les peticions d'inserir, modificar o eliminar entrades i taules de la base de dades, entre d'altres.

Per una altra banda, trobem la interfície d'administradors, la qual mostra les peces i proves mitjançant una pàgina web (sense VR) només als usuaris amb privilegis (dissenyadors de peces i proves). Cal remarcar que la interfície de peces només és visible per als dissenyadors de peces i la de proves només per als dissenyadors de proves.

També es té la interfície d'estadístiques, la qual mostra totes les estadístiques de tots els usuaris de manera ordenada. Aquesta interfície és accessible per qualsevol jugador, sigui registrat o no registrat.

Per últim, i situat al centre de la Figura 3.3, es troba el joc en sí, el qual és la part principal de tot el disseny. El joc és accessible, al igual que les estadístiques, per qualsevol jugador però s'ha d'accedir mitjançant la realitat virtual, tot i que no és obligatori però si recomanat.



Figura 3.3 Disseny arquitectònic

3.4. Implementació

Aquest apartat pretén explicar el desenvolupament de tota la implementació del sistema explicat al disseny arquitectònic. Per facilitar la organització de l'apartat es dividirà segons els 3 mòduls principals: el joc en VR, la interfície d'administradors i la d'estadístiques.

Al tractar-se d'una implementació sobre entorns web els llenguatges utilitzats són HTML, CSS, JavaScript i PHP. A més, també s'han usat altres llibreries com JQuery¹⁸ o Bootstrap¹⁹ per tal de simplificar la tasca de programació. A cada uns dels apartats s'explicaran en detall les llibreries i eines externes usades per desenvolupar cada un dels mòduls.

¹⁸ <u>https://jquery.com/</u>

¹⁹ <u>https://getbootstrap.com/</u>

3.4.1. Joc en VR

Aquest apartat és el més important ja que engloba la interfície principal del joc i tota la gestió en VR. Tal i com s'ha explicat amb anterioritat el *framework* utilitzat és A-Frame, el qual simplifica bastant la feina de programar una escena que funcioni en VR sobre HTML.

Com que els meus coneixements sobre A-Frame al començament d'aquest treball eren nuls he hagut de realitzar tots els tutorials i llegir la documentació del *framework*²⁰ per entendre com funciona. Això no ha suposat cap problema perquè a la seva pàgina web hi ha una documentació tècnica prou extensa que tracta cada un dels aspectes i components de A-Frame.

Per tal de resumir A-Frame podríem dir que és un *framework* que permet desenvolupar programes en VR sobre HTML usant una sintaxis senzilla i intuïtiva. El programador no s'ha de preocupar del moviment ni la interacció del dispositiu de VR amb el programa, perquè el propi *framework* ho gestiona, i proporciona una documentació molt completa on s'expliquen els esdeveniments, les interaccions i els components, entre d'altres. Així una persona sense experiència en el món del desenvolupament VR és capaç de desenvolupar programes o jocs realment complexos.

També s'ha d'explicar que la base del *framework* són els components. Un component és un tros de codi que determina el que s'executa quan es crea, es modifica o es destrueix el component. Els components es poden associar a objectes per poder canviar el seu comportament reutilitzant codi. Per exemple, es pot tenir un component que reprodueixi un so determinat quan es clica l'objecte associat al component. Aquest component es pot associar a múltiples objectes i tots reproduiran el so quan es clica sobre ells. A més, els components són fàcils de compartir, i per això hi ha molts de components desenvolupats per altra gent, que permeten realitzar coses vertaderament interessants, com generar un escenari, crear un teclat, o inclús afegir físiques. La majoria de components són de codi obert i es poden trobar a GitHub, amb la seva respectiva documentació. Durant el procés d'implementació s'han usat nombrosos components externs, de codi obert i desenvolupats per usuaris externs, els quals es troben referenciats a peu de pàgina a mesura que s'anomenen dins el document.

Amb la finalitat d'explicar, pas per pas, tota la implementació del joc en VR es dividirà la secció segons els submòduls que s'han anat implementant i que es poden veure a la Figura 3.4. La llegenda també es pot consultar a la pròpia figura.

²⁰ <u>https://aframe.io/docs/1.3.0/introduction/</u>



Figura 3.4 Submòduls de implementació del joc

3.4.1.1. Escenari exterior

El primer pas dut a terme en la implementació ha sigut la creació d'un escenari relativament complex per tal de familiaritzar-me amb A-Frame i tenir una base sòlida sobre la qual poder fer algunes proves i començar a programar les parts més complexes.

El component usat per realitzar l'escenari que simula un camp verd amb arbres es diu *aframe-environment-component*²¹ i permet generar amb 1 sola línia de codi un escenari, d'un conjunt d'escenaris. En aquest cas s'ha usat l'escenari "forest" i s'ha personalitzat segons els atributs explicat al GitHub per posar la quantitat d'arbres desitjada, augmentar la mida de l'escenari i canviar els colors i la textura del sòl. En resum, s'ha adaptat aquest component per crear un escenari verd sobre el qual començar a treballar.

El següent pas ha estat crear un primer model del menú principal (veure Figura 3.5) i pensar la quantitat de botons que tindrà i on estaran situats. Aprofitant la immersió que proporciona la VR s'ha optat per fer que cada una de les opcions del menú estiguin "flotant" i que tinguin un regruix exterior per augmentar el realisme. S'han aplicat textures a cada una de les opcions i una animació al passar el ratolí (o el làser) per damunt. En aquest punt el menú encara és completament estàtic (no s'actualitza segons els nivells que hi ha) ja que només es cerca familiaritzar-se poc a poc amb l'ús de A-Frame.

²¹ <u>https://github.com/supermedium/aframe-environment-component</u>

Una vegada es té l'escenari bàsic i una primera aproximació del menú s'ha optat per acabar d'implementar tot l'escenari complet, amb un nivell de detall relativament alt. És veritat que un escenari complex no aporta cap funcionalitat al joc, però si aporta realisme al jugador i dona una imatge d'un projecte de més qualitat. La programació de tot l'escenari no ha estat una tasca complicada, però si molt llarga, perquè bàsicament s'han d'anar ajuntant elements bàsics (cubs, plans i esferes) perquè quedi el resultat desitjat.



Figura 3.5 Menú principal

3.4.1.1.1. Textures

Per augmentar el realisme s'han inclòs nombroses textures (asfalt, rajoles, passos de zebra, marbre, fusta, entre d'altres). Aquestes textures són senzilles d'incloure, però s'han de tenir en compte les dimensions de les figures sobre les quals s'apliquen, perquè no quedi deformada.

És important explicar que s'ha fet ús de l'Asset Management System²² d'A-Frame, un gestor de textures o sons que permet tenir-ho tot al mateix lloc i carregar-ho a l'inici de l'execució, per disminuir els temps de càrrega i millorar el rendiment.

3.4.1.1.2. Component Instanced Mesh

A l'escenari hi ha una tanca a ambdós costats del camí que està formada per varis rectangles verticals i 2 rectangles en horitzontal (veure Figura 3.6). Cada un d'aquests elements es renderitza per separat, cosa que penalitza el rendiment, tal i com es parlarà a l'apartat 3.4.1.19. Per això, investigant per internet, es va trobar un component que agrupa varies entitats en una sola entitat i així millora el rendiment. Aquest component

²² <u>https://aframe.io/docs/1.3.0/core/asset-management-system.html</u>

es diu *instanced-mesh*²³ i seguint la documentació de GitHub s'ha pogut fer que cada conjunt de tanques sigui una única entitat, en lloc de més de 15 entitats.



Figura 3.6 Tanques amb textura

Ha estat un poc complicat poder implementar aquest component perquè la majoria d'exemples no permetien usar textures i la tanca té 2 tipus de textures diferents. No obstant, s'ha trobat un exemple a la documentació que explicava com incloure textures correctament.

3.4.1.1.3. Component cel i sol

Amb la finalitat d'augmentar el realisme s'ha inclòs a l'escenari un sol, el qual no il·lumina per si mateix ja que és purament decoratiu, i un cel que simula el cel real. Per això s'ha usat un component anomenat *aframe-simple-sun-sky*²⁴ i que permet, amb una única instrucció, afegir el cel i el sol a la nostra escena.

Aquest component s'ha combinat amb il·luminació (explicada a l'apartat 3.4.1.7) que simula que prové del sol i que il·lumina tot l'escenari.

3.4.1.1.4. Menú auxiliar

A la Figura 3.6 es pot apreciar un menú auxiliar, el qual s'ha decidit implementar perquè els jugadors puguin apreciar i modificar l'escenari del joc. Aquest menú permet mostrar/ocultar el menú principal i canviar la il·luminació. Aquests canvis no són definitius ja que quan es juga un nivell la il·luminació es torna activar, però almenys permeten que l'usuari apreciï l'escenari que l'envolta.

3.4.1.2. Fonts

Al començar a incloure text es va veure que només es renderitzaven els caràcters comuns, i tant els accents com certes lletres (\tilde{n} , ς ...) no es mostraven. Investigant a l'apartat de A-Frame relacionat amb el text [6] s'explica com canviar el tipus de lletra i com utilitzar caràcters que no siguin ASCII. Això requereix que es generi una font personalitzada que

²³ <u>https://github.com/diarmidmackenzie/instanced-mesh</u>

²⁴ <u>https://github.com/DougReeder/aframe-simple-sun-sky</u>

inclogui els caràcters desitjats. Per realitzar aquesta tasca s'ha usat *MSDF font generator*²⁵, un generador de fonts que permet seleccionar els caràcters a generar. S'han inclòs tots els caràcters que es volen usar i que no hi eren (accents, símbols...) i s'ha usat la font *Open Sans*.

Una vegada generada la font s'ha seguit la documentació de A-Frame per usar la font dins el joc. S'ha decidit que la millor idea era crear un component amb la font perquè així cada vegada que es crea una entitat de text se li assigna aquest component i es reutilitza codi.



Figura 3.7 Font del títol

Pel que fa al títol, es volia usar una font diferent a la resta, que ressalti i que sigui vistosa. Les fonts proposades per A-Frame són bastant limitades però s'ha trobat un repositori de GitHub anomenat *aframe-fonts*²⁶ el qual inclou més de 2000 fonts de Google Fonts llestes per usar dins A-Frame. Aquestes fonts tampoc admeten caràcters especials però com que el títol del joc està en anglès això no suposa cap problema. La font usada ha estat *Shadows Into Light* i el resultat es pot veure a la Figura 3.7.

3.4.1.3. Menú de configuració

El proper pas de la implementació ha estat desenvolupar el menú de configuració ("Ajustes" en el joc), el qual es pot veure a la Figura 3.8.



Figura 3.8 Menú de configuració

²⁵ <u>https://msdf-bmfont.donmccurdy.com/</u>

²⁶ <u>https://github.com/etiennepinchon/aframe-fonts</u>

Segons els requisits, els usuaris han de poder registrar-se, iniciar sessió i editar el seu nom d'usuari i l'avatar. Per complir amb aquests requisits s'ha d'implementar un sistema d'usuaris amb credencials segures. Tota la informació dels usuaris (nom, contrasenya i avatar) es guarda a la base de dades MySQL seguint el model de dades explicat a l'apartat 3.3.2. Als següents apartats s'explicarà amb detall cada una de les opcions del menú de configuració. A més, també es parlarà del teclat i del banner del perfil.

3.4.1.3.1. Opció "Registrarse"

En aquesta opció s'ha d'introduir un nom d'usuari vàlid (menys de 13 caràcters) i una contrasenya (que sigui igual als dos camps de text). Ni l'usuari ni la contrasenya poden tenir espais. Totes aquestes comprovacions es realitzen a la banda del client ja que l'únic perjudicat de saltar-se aqueixes restriccions és el propi usuari i tampoc afecten a la base de dades.

Posteriorment, les dades s'envien al servidor i s'introdueixen a la base de dades. No obstant, abans d'això es calcula el hash de la contrasenya usant la funció *password_hash()* pròpia de PHP i llavors s'introdueix a la base de dades. Aquest pas és per no guardar la contrasenya en text pla i augmentar la seguretat.

És important destacar que el servidor revisa si el nom d'usuari a introduir ja existeix perquè no hi pot haver varis usuaris amb el mateix nom. Això es notifica a l'usuari, indicant que si aquell és el seu nom d'usuari ha d'iniciar sessió.

Una vegada s'ha completat el registre automàticament s'inicia la sessió d'aquell usuari perquè es puguin guardar les seves estadístiques.

3.4.1.3.2. Opció "Iniciar Sesión"

Aquí s'introdueixen les credencials dels usuaris ja registrats i el servidor comprova si coincideixen amb alguna entrada de la base de dades. En el cas de la contrasenya, s'usa la funció *password_verify()* per comprovar si la contrasenya introduïda per l'usuari i el hash de l'entrada de la base de dades són correctes.

Una vegada iniciada la sessió es carrega l'avatar de l'usuari i les seves estadístiques.

3.4.1.3.3. Opció "Editar Perfil"

Aquesta opció permet modificar el nom d'usuari i l'avatar tant d'usuaris registrats com no registrats. En el cas dels usuaris no registrats, els canvis es perdran al tancar el joc però almenys el seu nom d'usuari es veurà reflectit en les estadístiques. En canvi, en el cas dels usuaris registrats s'ha d'actualitzar la informació a la base de dades. Abans de fer la modificació, el servidor comprova que el nou nom d'usuari no coincideix amb un altre usuari i llavors actualitza l'entrada corresponent.

S'ha de dir que els rols de dissenyador de peces i dissenyador de proves no poden modificar el seu avatar, perquè en tenen un per defecte.

3.4.1.3.4. Opció "Información"

La finalitat d'aquesta opció es purament informativa, indicant qui és el desenvolupador i el context del joc. També inclou un enllaç al Linkedin del desenvolupador.

3.4.1.3.5. Teclat

El joc està pensat per jugar en VR per lo que l'ús del teclat físic és inviable. Per aquest motiu, s'ha cercat un teclat que es pugui incloure dins VR i es pugui utilitzar amb els controladors. El teclat usat es diu *aframe-super-keyboard*²⁷ i permet generar un teclat a qualsevol lloc de l'escena i personalitzar-lo (colors, lletres o nombres, filtres...) al gust. En aquest cas no s'ha personalitzat massa ja que el teclat per defecte ja ens serveix per introduir text.

El teclat es crea quan es fa clic sobre un camp per introduir text. Una vegada s'ha introduït el text desitjat i es pitja la tecla "Enter" el teclat desapareix i el text escrit es mostra al camp de text corresponent. No s'han tingut massa problemes amb el teclat ja que la documentació de GitHub està prou clara.



Figura 3.9 Teclat en VR

Durant la implementació del teclat s'ha detectat que només es pot fer clic sobre les lletres del teclat amb un dels dos controladors (el que s'ha definit durant la configuració del teclat). S'ha intentat trobar la manera de que funcioni amb els dos controladors alhora però no ha estat possible. Cercant informació s'ha trobat un fil de GitHub [7] que parla amb el desenvolupador del propi component del teclat precisament d'aquest tema i afirma que no és possible que funcioni amb ambdós controladors. De totes formes, s'ha optat per implementar una solució parcial, la qual consisteix en que cada vegada que un usuari obre el teclat, es configura el teclat segons el controlador (dreta o esquerre) que l'ha obert. D'aquesta manera els usuaris esquerrans podran utilitzar el teclat amb el

²⁷ <u>https://github.com/supermedium/aframe-super-keyboard</u>

controlador esquerre ja que es suposa que han obert el teclat utilitzant aquest mateix controlador.

3.4.1.3.6. Banner del perfil

Al cantó superior dret del menú principal es troba un banner amb el nom i l'avatar de l'usuari. Si es fa clic sobre aquest banner s'indica el tipus de rol que té l'usuari (jugador, dissenyador de peces o dissenyador de proves) i si és un usuari registrat. En cas de que sigui un usuari registrat es dona la opció de tancar la sessió, i en cas contrari, si l'usuari no està registrat es permet iniciar sessió.



Figura 3.10 Interfície banner del perfil

3.4.1.4. PHP Session

Introduir les credencials cada vegada que es recarrega el joc és una tasca tediosa. Per això, s'ha implementat una funcionalitat que ofereix PHP i que es diu PHP Session. Això permet que les credencials dels usuaris es recordin encara que és tanqui el propi joc, però es perden quan es tanca el navegador. A diferència de les cookies, les credencials que es guarden usant PHP Session no es queden dins el navegador de l'usuari, sinó que és el servidor el que les recorda, augmentant així la seguretat ja que sinó la contrasenya estaria guardada en text pla al navegador dels usuaris.

El funcionament de PHP Session és el següent: quan un usuari entra amb les seves credencials s'inicia una sessió, la qual es manté fins que es tanca el navegador. Cada vegada que es torna entrar al joc es revisa si aquell usuari ja té una sessió oberta i, en cas afirmatiu, entra directament al seu compte sense haver d'introduir res. Quan un usuari tanca la seva sessió es destrueix la sessió de PHP perquè no es recordin les credencials.

Això no només s'usa en el joc, sinó que a l'apartat de proves i peces (es parlarà més endavant) també es requereixen credencials, les quals es recorden de la mateixa manera usant PHP Session. S'ha de dir que la sessió es guarda entre totes les pàgines (joc, proves i peces), per lo que s'han de guardar les credencials en 3 variables distintes perquè a un mateix ordinador s'han de poder introduir credencials diferents per a l'apartat del joc i pels apartats de proves i peces.

3.4.1.5. Menú estadístiques

Al menú principal també hi ha una opció d'estadístiques. Aquesta opció mostra les 5 estadístiques més recents de l'usuari que ha iniciat sessió. Si es tracta d'un usuari no registrat no es mostren estadístiques.

Atrás Estadísticas							
Usuario	Prueba	Fecha	Tiempo	Punt.			
jaume	Tutorial	27/04/2022	00:00	0.00			
Para visualizar todas las estadísticas y poder usar filtros puedes usar el							
	siguiente lini	k: Estadísticas (Sin VR	()				

Figura 3.11 Menú estadístiques de l'usuari "jaume"

Aquest menú d'estadístiques és relativament limitat però, com que per veure totes les estadístiques completes s'ha de redirigir a l'usuari a una altra plana web (sense VR), és una forma de veure les darreres estadístiques de cada propi usuari sense haver de sortir de VR.

Per implementar aquest menú simplement es recorren totes les estadístiques del usuari registrat, les quals ja s'han carregat des de la base de dades quan l'usuari ha iniciat sessió, i es mostren les darreres 5 estadístiques, tal i com es pot veure a la Figura 3.11.

No obstant, la implementació de la interfície completa d'estadístiques (sense VR) s'explicarà més endavant, a l'apartat 3.4.3.

3.4.1.6. Escenari interior

Una vegada explicat i implementat tot l'escenari exterior i les funcionalitats dels menús exteriors es continuarà explicant la implementació de l'escenari interior, zona on es desenvolupa la major part del joc.

L'escenari interior és relativament senzill però suficient per aportar realisme al joc. Consta de 2 taules amb ordinadors a sobre, les quals s'han programat creant 1 únic component. És a dir, s'ha programat un component que renderitza una taula amb ordinadors i llavors s'ha inclòs el component a l'escena 2 vegades per tal de generar 2 taules, però canviant la posició. Cal destacar que els ordinadors estan formats per 2 paral·lelepípedes

3.4. Implementació

rectangulars amb textures de teclat i de pantalla. Al afegir una textura a un objecte A-Frame aplica la mateixa textura a totes les seves cares. Això fa que els ordinadors sembla que tinguin 2 pantalles. Per evitar-ho s'ha usat un component anomenat *aframe-multisrccomponent*²⁸ el qual permet afegir diferents textures a cada cara d'un objecte. Aquest component també s'ha usat en altres objectes del joc per solucionar problemes semblants amb les textures. Sobre aquest component s'han tingut problemes perquè quan s'eliminava una entitat que tenia assignat aquest component sorgia un error. Investigant i parlant amb el desenvolupador del component s'ha pogut modificar el codi font, com es pot veure en aquesta conversació de GitHub [8], per eliminar l'error.

A l'escenari també hi ha una taula que envolta al jugador, simulant que és el dependent d'una tenda de reparació d'ordinadors. Aquesta taula té uns rètols que indiquen informació pròpia del joc i també té la capsa de l'ordinador que s'ha de muntar. Aquesta capsa té textures simulant una capsa d'ordinador real les quals s'han modificat amb Adobe Photoshop per tal de que coincideixin exactament amb la mida de la capsa.

3.4.1.7. Il·luminació

Per a la il·luminació del joc s'ha optat per usar els següents element:

- 1 llum ambient perquè hi hagi un poc de lluminositat a tota l'escena.
- 1 llum direccional baix terra el qual il·lumina el sòl i fa que es vegi més vistós.
- 2 llums direccionals simulant la llum del sol. S'ha decidit que siguin 2 perquè sinó l'escena no quedava suficientment il·luminada.
- 2 llums puntuals al sòtil de l'oficina perquè l'interior quedi ben il·luminat.

Tots els llums emetien ombres sobre els objectes, però degut als problemes de rendiment que s'explicaran a l'apartat 3.4.1.19 es varen haver de desactivar.

3.4.1.8. Desplaçament interior/exterior

El menú principal es troba a la part exterior de l'escenari i la zona de muntatge d'ordinadors es troba a l'interior de l'oficina. Per desplaçar al jugador entre les dues zones (interior i exterior) s'ha decidit crear una animació que posa la pantalla en negre de manera progressiva. Una vegada finalitzada l'animació es desplaça al jugador, cosa que s'aconsegueix incloent la càmera dins una altre entitat HTML anomenada "rig" i modificant la posició d'aquesta entitat, perquè A-Frame no permet modificar directament la posició ni la rotació de la càmera. Després, es tornar a realitzar l'animació de manera invertida per eliminar la pantalla negra.

Per aconseguir realitzar l'animació de la pantalla en negre es crea una esfera de color negre just sobre la posició de la càmera, amb una opacitat de 0. Aquesta opacitat es va

²⁸ <u>https://github.com/elbobo/aframe-multisrc-component</u>

incrementant fins que arriba a 1. Per invertir l'animació es fa el mateix però a la inversa, canviant la opacitat de 1 fins a 0 i llavors s'elimina l'esfera completament.

3.4.1.9. Menú components

Tots els components es carreguen des de la base de dades a l'inici del joc. Però, no és fins que es comença a jugar un nivell, que es mostren a la pantalla i es poden generar. El menú de components mostra els 7 tipus de components (placa base, CPU, ventilador, ram, disc, font i GPU) i, al seleccionar-ne un, mostra tots els components d'aquell tipus. Només hi caben 3 components per pàgina, però es generen les pàgines necessàries segons la quantitat de components que hi hagi a la base de dades. S'ha implementat la navegació entre pàgines mitjançant dues fletxes situades a la part superior del menú. Al mostrar els components el joc sap quina quantitat de components hi ha de cada tipus per lo que calcula quantes pàgines s'han de generar.

Cada tipus de component té els seus camps d'especificacions, els quals no coincideixen amb els altres tipus de components. Per això, cada vegada que s'han de generar les targetes dels components primer es mira de quin tipus és i, segons el tipus, es mostren les seves respectives especificacions ordenades segons l'ordre que s'ha determinat per aquell tipus. D'aquesta manera s'aconsegueix que cada tipus de component mostri les especificacions segons l'ordre desitjat, el qual ha estat pensat per facilitar la visualització.



Figura 3.12 Menú de components

Figura 3.13 Submenú CPU

A la Figura 3.12 i a la Figura 3.13 es pot veure el menú de components i concretament com es mostren les especificacions de les CPU i la possibilitat de canviar de pàgina, cosa que significa que hi ha més de 3 CPU diferents a la base de dades.

3.4.1.9.1. Generació de components

Des del menú de components es permet la generació dels mateixos dins l'escena. Simplement fent clic sobre la targeta d'un component (per exemple, a la Figura 3.13 al fer clic sobre una de les 3 targetes de CPU) es genera damunt la taula el component seleccionat.

Això s'aconsegueix creant un objecte de tipus "a-box" des del JavaScript assignant-li l'identificador, les dimensions, la textura i la rotació corresponent. La posició s'assigna de manera pseudoaleatòria dins una determinada zona de l'escena perquè no tots els components es generin exactament al mateix lloc i hi hagi un poc de dispersió. També se li assignen físiques, col·lisions i esdeveniments ("events") que contenen totes les especificacions el propi component. Les físiques i col·lisions s'explicaran en detall als apartats 3.4.1.11 i 3.4.1.13.

Si s'intenta generar un component que ja s'ha generat primer s'elimina el que ja existeix abans de generar-ne un altre, per evitar que hi hagi components duplicats. Però, en el cas de que es tracti de tornar a generar un component que ja està col·locat, es mostra un avís al jugador en lloc d'eliminar el component que ja s'ha col·locat. No obstant, en el cas dels discs i la RAM si es permeten generar múltiples components idèntics perquè s'han de poder col·locar varis discs o RAM exactament iguals a l'ordinador. Per gestionar l'identificador d'aquests components idèntics es modifica la terminació del propi identificador, afegint una 'X' al final per cada nova generació. Per exemple, si es generen 3 memòries RAM idèntiques tindran els següents identificadors, per ordre de generació: "cmp-ram-0", "cmp-ram-0X" i "cmp-ram-0XX".

3.4.1.10. Menú tasques

Al seleccionar una prova del menú principal i mentre l'usuari està sent desplaçat a l'interior de l'oficina es carreguen les tasques específiques d'aquella prova. Això s'aconsegueix fent una petició a la base de dades la qual retorna totes les tasques de la prova. Les tasques es mostren al menú de tasques (a la dreta del jugador), juntament amb el nom de la prova, el temps transcorregut i 2 botons per guardar o sortir de la prova (veure Figura 3.14).

Aquestes tasques serveixen per indicar al jugador certs requeriments que s'han de complir per tal d'obtenir bona puntuació. Els requeriments estan relacionats amb les especificacions dels components de manera que s'han de tenir en compte aquestes especificacions, les quals són diferents per a cada un dels components, a l'hora de complir amb les tasques.

Les 3 primeres tasques són d'emmagatzematge i segueixen el mateix patró a tots els nivells, ja que indiquen la quantitat de memòria mínima (RAM, HDD y SSD) que s'ha de muntar a l'ordinador. Per indicar que no hi ha restricció de memòria mínima aquestes tasques mostren 0 GB. La resta de les tasques simplement indiquen els valors que han de prendre certs atributs d'un tipus de component.



Figura 3.14 Menú de tasques

3.4.1.11. Físiques

Les físiques són necessàries per fer que els components es vegin afectats per la gravetat i els jugadors els puguin moure de lloc i col·locar-los al lloc desitjat. Per afegir físiques al joc s'ha utilitzat un component anomenat *aframe-physics-system*²⁹, el qual permet, de manera relativament senzilla i seguint la documentació, dotar als objectes de físiques.

Una vegada es tenen les físiques s'han de configurar correctament, aplicant l'etiqueta *dynamic-body* als objectes que s'han de veure afectats per les físiques, i l'etiqueta *static-body* a tots aquells objectes que han d'actuar com a objectes fixos quan un *dynamic-body* impacti sobre ells. Aquests objectes són parets, el sòl, el sòtil, les taules i els menús, ja que els objectes afectats per la gravetat han de xocar amb els *static-body*, en lloc de travessar-los.

Seguint la documentació de GitHub del component de físiques no s'han tingut massa problemes per implementar-ho dins el joc. A més, s'ha usat la opció de *debug* que proporciona el component per veure si les *hitbox* eren les adequades i s'ajustaven a la forma dels objectes. Per motius de *debugging* es configurava la gravetat a -0.5 m/s perquè es feia més senzill provar les físiques amb el teclat i ratolí. Tot i això, al joc final la gravetat s'ha deixat a -9.8 m/s, tal i com pertoca.

3.4.1.12. Interacció dels controladors

És evident que el joc no només ha de permetre el moviment i el desplaçament del jugador pel món virtual, sinó que el jugador ha de ser capaç d'interactuar (fer clic sobre botons, agafar i amollar components...) dins VR. Això fa que la interacció dels controladors es

²⁹ <u>https://github.com/n5ro/aframe-physics-system</u>

divideixi en dos apartats, els propis controladors per fer clic, i un component que permet interactuar amb els objectes.

3.4.1.12.1. Controladors

A-Frame proporciona per defecte una entitat per a cada controlador (dret i esquerre) els quals es renderitzen sobre el propi joc, tenint en compte el tipus de controlador que l'usuari està usant en la realitat. Per exemple, si s'està usant unes Oculus Quest, els controladors que apareixeran al joc seran els de Oculus Quest.

No obstant, també s'ha de configurar el raig que surt de cada controlador i que permet fer clic sobre els botons. Això és una tasca trivial perquè està ben explicada a la documentació d'A-Frame, a l'apartat dels controladors [9]. Bàsicament s'afegeix un atribut a l'entitat dels controladors, indicant la mà (dreta o esquerra), els objectes sobre els quals ha de poder interactuar el raig (sol ser una classe de CSS) i el color i la opacitat del raig.

3.4.1.12.2. Component super-hands

Els controladors amb els rajos no són suficients per complir els requisits del joc perquè hi ha d'haver una manera d' agafar els components i col·locar-los al lloc que pertoqui. Per complir aquest requisit s'ha decidit implementar un component molt usat dins el món de A-Frame i que es diu *aframe-super-hands-component*³⁰. Aquest component proporciona unes mans que es renderitzen dins el joc i permeten realitzar els següents gests: agafar, arrossegar, amollar i redimensionar. A més, també funciona juntament amb les físiques explicades a l'apartat 3.4.1.11 i teòricament permet realitzar funcions de *drag-and-drop*.

Aconseguir mostrar les mans dins VR i agafar, arrossegar i redimensionar objectes no ha suposat cap problema. Seguint la documentació del GitHub ha resultat fàcil que tot lo anomenat funcioni. Però, segons la pròpia documentació, les físiques haurien de funcionar juntament amb les mans i això no funcionava. El primer error detectat és que s'havia d'incloure un component anomenat *aframe-physics-extras* ³¹ que permet la interacció de *super-hands* amb físiques. Una vegada fet això les físiques seguien sense funcionar i es va decidir obrir un fil al GitHub [10], mentre es seguia cercant informació per Internet. Gràcies a una pregunta d'un usuari de Stack Overflow [11] es va trobar l'error, el qual radicava en que s'havia d'afegir l'etiqueta *static-body* a les entitats de les mans perquè funcionessin juntament amb les físiques. Una vegada solucionat l'error es va explicar al fil de GitHub i es va tancar el fil i va ser quan el creador del component va adjuntar un enllaç a un tutorial [12] de físiques i *super-hands*, el qual també ha resultat de gran ajuda.

³⁰ <u>https://github.com/wmurphyrd/aframe-super-hands-component</u>

³¹ <u>https://github.com/wmurphyrd/aframe-physics-extras</u>

Com ja s'ha explicat el propi component ofereix una funció de *drag-and-drop* però, després de fer nombroses proves, els resultats obtinguts no eren del tot útils pel que es volia implementar al joc. A més, aquesta funció no funcionava gaire bé la major part de les vegades. Per això, s'ha decidit realitzar la implementació d'aquesta funció utilitzant un altre sistema de detecció de col·lisions, el qual s'explicarà a l'apartat 3.4.1.13.

3.4.1.12.3. Intercanvi controladors i super-hands

A la zona del menú principal s'han de poder usar els dos controladors amb els rajos i a la zona de muntatge s'ha de poder fer clic a botons i agafar components. Per solucionar aquesta problemàtica s'ha decidit que a la zona de muntatge la mà dreta tindrà l'entitat de *super-hands* (per agafar objectes) i la mà esquerra tindrà l'entitat del controlador amb raig de A-Frame. D'aquesta manera els usuaris no han de canviar de tipus de controlador, sinó que el canvi es realitza automàticament segons la zona en que es trobin. Per gestionar aquest canvi simplement s'han dissenyat unes funcions per modificar el tipus de controlador de cada mà, les quals eliminen les entitats que pertoquen i en creen de noves segons el tipus de controlador especificat. Aquestes funcions s'executen quan l'usuari entra o surt d'un nivell, que és quan es canvia la posició del jugador segons el que s'ha explicat a l'apartat 3.4.1.8. El resultat final dels controladors i de *super-hands* a la zona de muntatge està reflectit a la Figura 3.18**jError! No se encuentra el origen de la referencia.**, on es pot veure el controlador a la mà esquerra i l'entitat de *super-hands* a la mà dreta.

3.4.1.13. Col·lisions

A l'apartat 3.4.1.12.2 s'ha comentat que el component *super-hands* en teoria té una funció de *drag-and-drop*, però ja s'ha dit que no serveix per la implementació que es vol realitzar dins el joc. La idea és que a la zona de l'ordinador on es poden amollar els components hi hagi un cub o un rectangle d'un color vistós. Aquest rectangle s'ha de remarcar quan s'atraqui un component, indicant així que ja es pot amollar el component i que quedarà col·locat. Per implementar això és necessari detectar d'alguna manera que el component agafat està xocant amb la zona on es vol col·locar (*drop zone*).

Després d'investigar per Internet s'ha trobat un component anomenat *aabb-collider*³² que detecta col·lisions entre objectes i té *listeners* que s'executen quan comença i acaba una col·lisió. Aquest component és perfecte i compleix totes les necessitats descrites.

La configuració del component no ha suposat massa problemes ja que la documentació es prou clara. Simplement s'ha d'afegir el component als objectes que intervindran durant la col·lisió (peces i *drop zones*) i se'ls hi ha d'assignar una classe de CSS, en aquest cas ".colisionable". A més, també s'ha registrat un nou component anomenat *colision*, que el tenen assignat totes les peces que s'han de col·locar a l'ordinador, el qual gestiona els *listeners* de quan comença i acaba la col·lisió (*hitstart* y *hitend*).

³² <u>https://github.com/supermedium/superframe/tree/master/components/aabb-collider/</u>

Quan comença la col·lisió s'agafa la *drop zone* amb la qual està xocant la peça i s'incrementa la seva opacitat (1.0), indicant que allà es pot amollar la peça. El component *aabb-collider* pot detectar col·lisions amb més d'una entitat alhora, cosa que no ens interessa ja que només s'ha de poder amollar la peça a una sola *drop zone*. Per això, només es té en compte la primera *drop zone* amb la qual hi ha col·lisió (la més propera) i és la única que canvia la seva opacitat. També es desmarquen totes les altres *drop zones* per forçar que només n'hi hagi una marcada. A la Figura 3.15 es poden veure dues *drop zones*, una amb més opacitat que l'altre perquè hi ha una peça travessant-la.

Quan acaba la col·lisió es desmarca la *drop zone* que estava xocant amb la peça i es retorna la opacitat al seu valor inicial (0.3).



Figura 3.15 Funcionament de col·lisions

3.4.1.14. Interacció dels components

En aquests moments ja es poden generar peces, moure-les i detectar col·lisions amb les zones corresponents. No obstant, encara s'han d'implementar certes funcions quan s'agafa un component, s'amolla sobre una *drop zone*, o es descol·loca del propi ordinador.

3.4.1.14.1. Agafar components

A l'apartat 3.4.1.12.2 ja s'ha explicat com, mitjançant el component *super-hands*, es poden agafar i amollar objectes, els quals també tenen físiques. El propi component gestiona aquestes interaccions i també proporciona uns esdeveniments (*ondragstart* i *ondragend*) per executar codi al agafar o amollar objectes.

Aquests esdeveniments s'usen perquè, al agafar un component, surten les seves especificacions a la zona indicada amb el nom "Componente actual" enfront de l'ordinador. Això serveix perquè els usuaris puguin saber les especificacions d'un determinat component una vegada ja s'ha generat. Mostrar les especificacions no ha requerit massa feina perquè s'ha reutilitzat el codi que crea les targetes del menú de components, les quals es poden veure a la Figura 3.13.

Al agafar un component també es mostren a l'ordinador les *drop zones* que hi ha disponibles en aquell moment. És a dir, si ja s'ha col·locat un component a una *drop zone*, aquella *drop zone* no és mostra. Això és possible perquè es té guardat dins uns *arrays* les *drop zones* lliures i així quan s'agafa un component es sap quines s'han de mostrar.

Cal explicar que al amollar el component es deixen de mostrar les seves especificacions i les *drop zones* de l'ordinador.

3.4.1.14.2. Col·locar components

Quan s'amolla un component sobre una *drop zone* ressaltada (és a dir, que ha detectat una col·lisió amb el component), el component s'ha de col·locar allà on està la *drop zone*. Per aconseguir això simplement s'agafa la posició de la *drop zone*, se li assigna al component i es converteix el component en un *static-body* (fins ara era *dynamic-body*) per tal de que no es pugui agafar amb *super-hands*. Al col·locar-se el component també se li afegeix, mitjançant l'atribut *title*, el nom de la *drop zone* on s'acaba de col·locar (per exemple: *title="drop-zone-0"*). Posteriorment es marca la *drop zone* com a ocupada perquè al agafar un altre component no es mostri aquesta *drop zone*.

Inicialment només hi ha 2 *drop zones* (color vermell) a l'ordinador, la de la placa base i la de la font, tal i com es mostra a la Figura 3.16. Això és perquè la resta de components van damunt la placa base, per lo que no té sentit que es mostrin aquestes *drop zones* si encara no s'ha col·locat la placa base.

Al col·locar la placa base es creen totes les *drop zones* (color groc) relatives a ella (CPU, RAM, HDD, SSD i GPU). Totes les *drop zones* es creen just damunt la placa base a excepció de les *drop zones* dels HDD, ja que la seva posició no és relativa a la placa base, sinó que sempre van a la banda dreta de l'ordinador. A la Figura 3.17 es poden veure les *drop zones* que s'han creat una vegada s'ha col·locat la placa base.

Les plaques bases tenen especificacions diferents, i hi ha 3 camps relacionats amb el nombre de *slots* de RAM, HDD i SSD que té cada placa base. Aquests 3 camps es tenen en compte a l'hora de crear les *drop zones* de la placa base ja que, per exemple, si la placa base té 4 *slots* per discs HDD, només es creen 4 *drop zones* a la banda dels HDD. Això es realitza comprovant cada un d'aquests 3 camps i creant només la quantitat de *drop zones* necessària. Al codi es tenen definides posicions pel màxim nombre de *slots* disponibles de cada tipus (RAM: 4, HDD: 8, SSD: 4), i es van creant les *drop zones* a les posicions que pertoquen fins que s'arriba a la quantitat que indiquen aquests camps de *slots*.

El cas del ventilador també es peculiar ja que s'ha de col·locar damunt la CPU. Per això, es realitza el mateix que s'ha explicat amb la placa base, al col·locar la CPU es crea una nova *drop zone* (color verd) just damunt ella perquè es pugui col·locar el ventilador.

Dit això, és important destacar que els jugadors poden col·locar els components allà on desitgin, encara que la col·locació sigui errònia (això es veurà reflectit a la puntuació final explicada a l'apartat 3.4.1.15.4). La única restricció és que no es pot col·locar res damunt

la placa base ni la CPU sense que s'hagin col·locat aquests components (per motius obvis, ja que és físicament impossible col·locar un component damunt un altre que encara no s'ha col·locat).



Figura 3.16 Drop zones inicials

Figura 3.17 Drop zones placa base

3.4.1.14.3. Descol·locar components

Una vegada s'ha col·locat un component també hi ha d'haver l'opció de descol·locar-lo, per si el jugador ha comés algun error o si simplement es vol llevar el component. Tots els components col·locats són *static-body* per lo que no es poden agafar mitjançant *super-hands*.

S'ha decidit que la millor manera de descol·locar un component és usant el controlador de la mà esquerra. Aquest controlador té un raig que, al apuntar cap a un component ja col·locat, mostra les seves especificacions a la banda de "Componente actual". Això és molt útil per conèixer les especificacions d'un component que ja s'ha col·locat. A part de mostrar les especificacions també es disminueix la seva opacitat perquè el jugador sigui conscient del component al que està apuntant, ja que hi ha components molt petits. Quan es deixa d'apuntar amb el raig de llum es deixen de mostrar les especificacions i es restaura la opacitat inicial del component. A la Figura 3.18 es mostra com s'apunta a la placa base amb el controlador, la seva opacitat és menor i es mostren les seves especificacions.

Per descol·locar el component el que s'ha de fer es apuntar amb el raig de llum i fer clic sobre ell. El codi el que fa és convertir el component en *dynamic-body* (perquè es pugui moure), assignar-li una posició pseudoaleatòria dins els límits de la zona de recol·locació i eliminar el contingut de l'atribut *title*, el qual contenia el nom de la *drop zone* sobre la qual s'havia col·locat, tal i com s'ha explicat a l'apartat anterior. A més, també marca la *drop zone* on es situava aquest component com a buida perquè un altre component es pugui col·locar al seu lloc.

En el cas de que el component a descol·locar tingui altres components associats damunt, com una placa base o una CPU, tots els components associats també s'han de descol·locar, ja que no té sentit descol·locar la placa base i deixar els components que hi ha damunt flotant. Per això, el que es fa és revisar si el component té altres components sobre ell, mitjançant uns *arrays* que indiquen les *drop zones* ocupades, i descol·locar tots aquests components associats, seguint el mateix procés descrit al paràgraf anterior.



Figura 3.18 Descol·locació de peces

3.4.1.15. Final de partida

Quan un jugador decideix finalitzar la partida utilitzant el botó situat al menú de tasques (veure Figura 3.14), es mostra un resum indicant el resultat obtingut durant la realització del nivell.

Nivel: Tutorial	Fin de Partida			Tiempo: 03:24	
Tareas		Colocación de componentes			
# Memoria RAM mínima: 0 GB (Montada: 0 GB)		Р	laca Base	Disco	
# Memoria HDD mínima: 0 GB (Mo	and the second	CPU	Fuente		
# Memoria SSD mínima: 0 GB (Mo	ntada: 0 GB)	V	entilador RAM	GPU	
			Compatibili	dad técnica	
		Socket:	Placa Base (/	AM4) - CPU (LGA1700)	
		Consumo:	Máximo (200) W) - Fuente (450 W)	
Puntuación tota	l: 1.46			Menú principal	

Figura 3.19 Informació final de partida

El resum es pot veure a la Figura 3.19 i inclou el nom de la prova, el temps utilitzat, i la correcta realització de les tasques, de la col·locació dels components i de la compatibilitat tècnica (indicat en verd o vermell). Tot això deriva en una puntuació final. En els següents apartats es desglossarà com es calcula cada cosa.

3.4.1.15.1. Revisió de tasques

Comprovar les tasques que s'han realitzat correctament no ha estat una cosa senzilla ja que no es tenen guardats a la base de dades els components compatibles amb cada tasca, sinó que a la pròpia taula de tasques es guarden les especificacions que han de tenir els components per complir una tasca. A continuació es mostren 2 entrades simplificades de la taula de tasques per entendre millor la mecànica.

Descripció	Atribut	Acció	Valor	Tipus Peça
Montar una placa base ATX	tipoPlacaBase	=	ATX	1
Montar una CPU amb més de 8 nuclis	nucleos	>	8	2

Taula 3.2 Exemple sir	nplificat taula 'tarea'
-----------------------	-------------------------

En el cas de la primera entrada es pot veure com fa referència a l'atribut "tipoPlacaBase" de la taula "placa base" (ja que la columna de tipus peça indica un 1) i ens indica que el valor de l'atribut ha de ser igual a "ATX". A la segona entrada es segueix la mateixa mecànica però canviant l'acció (major en lloc d'igual).

Una vegada explicat això ja s'entén de quina manera es poden calcular els components compatibles amb cada tasca. Aquest càlcul el realitza el servidor quan es carreguen les tasques d'una determinada prova i retorna, dins un *array*, els components compatibles amb cada tasca de la prova. Al finalitzar la prova s'ha de determinar si cada una de les tasques s'ha dut a terme correctament i aquí s'ha de diferenciar entre 3 tipus diferents de tasques.

Si la tasca és d'emmagatzematge (les 3 primeres tasques de totes les proves sempre ho són) simplement s'ha de sumar la quantitat de memòria RAM, HDD i SSD que s'ha muntat a l'ordinador i veure si és superior a l'especificada a aquestes 3 tasques. En aquest tipus de tasques no es fa ús de l'*array* que ha retornat el servidor amb els components compatibles.

Si la tasca no és d'emmagatzematge però fa referència a la RAM o els discs s'ha de revisar que tots els mòduls de RAM o discs (ja que es pot col·locar més d'un mòdul d'aquests components) es troben dins l'*array* de compatibilitats.

Si la tasca no és d'emmagatzematge ni de RAM o discs únicament basta mirar si hi ha alguna peça de l'ordinador que es trobi dins l'*array* de compatibilitats d'aquesta tasca.

D'aquesta manera es recorren totes les tasques de la prova i es va indicant, de color verd o vermell, si la tasca s'ha complit o no.

3.4.1.15.2. Revisió de col·locació dels components

Aquest apartat, juntament amb el de compatibilitat tècnica, és més senzill de calcular ja que no és necessari realitzar cap consulta al servidor per validar si es compleix o no.

Per comprovar la col·locació dels components es revisa, per cada *drop zone*, el component que té associat i si és del tipus que toca (per exemple, a la *drop zone* de la placa base hi ha d'haver una placa base). Si la *drop zone* està buida o no conté un component del tipus adequat és marca en vermell. D'aquesta manera, encara que el jugador hagi col·locat una placa base a l'ordinador, però a un lloc incorrecte, apareixerà en vermell l'etiqueta de la placa base a l'apartat de col·locació dels components.

En aquest apartat s'ha d'anar amb compte amb la RAM i els discs, ja que poden tenir nombroses *drop zones*, les quals no tenen perquè estar totes ocupades. Per això, s'ha de revisar que aquestes *drop zones* no estan completament buides i, si no ho estan, que tot lo que hi ha són components del tipus que pertoca (RAM, HDD o SSD, segons la *drop zone*).

3.4.1.15.3. Revisió de compatibilitat tècnica

El càlcul de la compatibilitat tècnica és el més senzill de tots perquè únicament s'han de comprovar 2 aspectes, els quals no requereixen ni fer consultes al servidor, ni recórrer totes les *drop zones*.

El primer aspecte que es comprova és si el *socket* de la placa base i el de la CPU coincideixen. Això s'aconsegueix mirant les especificacions de la placa base i la CPU col·locades.

El segon aspecte a revisar és el consum, ja que la suma dels consums de la CPU i GPU ha de ser inferior a la potència proporcionada per la font d'alimentació. Igual que amb la compatibilitat del *socket*, aquí únicament es miren les especificacions de la CPU, GPU i font col·locades i es realitza el càlcul per veure si es compleix o no.

3.4.1.15.4. Càlcul de puntuació

La puntuació final és el reflex del temps transcorregut durant la realització de la prova i els errors comesos a cada un dels apartats explicats anteriorment. A continuació es descriurà el procés seguit per calcular la puntuació, així com els descomptes aplicats segons els tipus d'errors. Cal recordar que la puntuació va de 0 a 10 (2 decimals).

En primer lloc, si hi ha algun error de col·locació la puntuació màxima que es pot obtenir passa de 10 a 5, ja que a la vida real si hi ha components mal col·locats l'ordinador no es posaria en marxa, per lo que s'ha pensat que aquesta penalització es prou realista. A més, per cada error de col·locació es resta 1 punt de la puntuació. En segon lloc, els errors de les tasques descompten 0.5 punts cada un. Aquests errors no són tan importants com els anteriors ja que l'ordinador funcionaria, però no s'haurien complit els requisits proposats.

En tercer lloc, els errors de compatibilitat tècnica descompten 1 punt cada un. Això és així perquè aquests errors són pràcticament igual de greus que els errors de col·locació.

Per últim, per al temps s'ha definit un màxim i un mínim per determinar quanta puntuació s'ha de descomptar. A més, el temps només pot arribar a descomptar 5 punts de la puntuació. Els valors definits són 2 minuts per al mínim i 15 minuts per al màxim. Després d'haver realitzat varies proves amb usuaris s'ha comprovat que en 2 minuts és possible muntar l'ordinador i que no massa usuaris passaven de 15 minuts. Una vegada es sap el temps utilitzat, s'aplica una formula per obtenir un valor entre 0 (2 minuts o menys) i 5 (15 minuts o més) i aquest valor es resta a la puntuació final.

Tipus	Quantitat	Descompte	Aclaracions
Errors col·locació	1	1*1	Cada error -1 punt (punt. max: 5)
Errors tasques	2	0.5*2	Cada error -0.5 punts
Errors compatibilitat	1	1*1	Cada error -1 punt
Temps	8.5 min	2.5	Oscil·la entre -0 (2 min) i -5 (15 min)

Taula 3.3 Exemple càlcul de puntuació

La Taula 3.3 mostra un exemple del càlcul de puntuació donat uns determinats valors. En aquesta taula la puntuació final seria (10 - 5 - 1 - 1 - 1 - 2.5) 0.5 sobre 10. El problema principal ha estat l'error de col·locació, ja que encara que només n'hi hagi un ja disminueix la màxima puntuació a 5 en lloc de 10.

3.4.1.16. Tutorial

S'ha decidit implementar una prova anomenada tutorial que serveix per introduir als nous usuaris els controls i informació útil sobre el joc. Aquesta prova presenta una sèrie de diferències respecte a les altres.

En primer lloc, aquesta prova no inclou cap tasca, excepte les 3 tasques obligatòries d'emmagatzematge explicades a l'apartat 3.4.1.10, però tenen valors mínims de 0.

En segon lloc, al entrar a la prova apareix un nou menú enfront del jugador que el va guiant durant tot el transcurs de la prova. Aquest menú comença explicant els controls bàsics i les mecàniques per agafar, amollar, col·locar i descol·locar components. A més, també s'explica perquè serveix cada component de l'ordinador, amb la intenció de que el joc sigui educatiu.

En tercer lloc, durant el procés de muntatge guiat pel tutorial s'explica on s'ha de col·locar cada component i no s'avança el tutorial fins que el jugador l'ha col·locat al lloc

corresponent, assegurant així que el jugador ho ha entès bé. Per implementar això cada vegada que es col·loca un component es revisa si l'usuari està jugant al tutorial i si la peça col·locada era la que es demanava al tutorial i si està al seu lloc. Si tot això es compleix, s'avança a la següent instrucció del tutorial.

Per acabar, quan es finalitza el tutorial i es mostra la puntuació final, apareix una nova pantalla, que no surt a cap de les altres proves, i que explica què significa cada apartat de la pantalla de fi de partida i com es calcula la puntuació final.

Per implementar el tutorial amb les diferències esmentades únicament es comprova al entrar a cada prova si es tracta del tutorial, ja que aquest té un identificador de prova únic i específic.

3.4.1.17. Centrar al jugador

Pràcticament tots els jocs de VR tenen un botó que s'usa per centrar al jugador en la zona de joc. Això és molt útil perquè, al jugar en VR, s'ha de definir una zona segura delimitada al món real. Moltes vegades, al obrir un joc, el jugador no es troba al centre d'aquesta zona segura i per tant llavors no es pot moure en certes direccions sense sortir de la zona segura o xocar amb altres objectes del món real. Per això, aquesta funció s'ha d'usar quan l'usuari es troba al centre de la seva zona segura (al món real).

```
// Funcion que re-centra al jugador a la posicion inicial
// x, y: coordenadas sobre las que centrar. rotation: rotación de la camara
function centerPlayer(x, z, rotation) {
    // Coordenadas mundo de la camara
    var cameraEl = document.querySelector('#camara');
    var worldPos1 = new THREE.Vector3();
   worldPos1.setFromMatrixPosition(cameraEl.object3D.matrixWorld);
    // Coordenadas mundo del rig
    var rigEl = document.querySelector('#rig');
    rigEl.object3D.position.set(x, 0.2, z);
    var worldPos2 = new THREE.Vector3();
    worldPos2.setFromMatrixPosition(rigEl.object3D.matrixWorld);
    worldPos2.x += x;
   worldPos2.z += z;
    // Actualizar posicion y rotacion del rig
    document.getElementById("rig").object3D.position.set(worldPos2.x -
worldPos1.x, 0.2, worldPos2.z - worldPos1.z);
    document.getElementById("rig").setAttribute("rotation", rotation);
}
```

Figura 3.20 Funció per centrar al jugador

Per implementar aquesta funció s'ha intentat actualitzar directament la posició de la càmera, però això no funciona durant l'execució del joc. La solució, tal i com s'explica a una resposta de Stack Overflow [13] és crear una entitat que conté a la càmera i llavors modificar aquesta entitat. En el nostre cas s'ha definit una nova entitat anomenada *rig* que és pare de l'entitat de la càmera.

No obstant, això no és suficient perquè quan el jugador es desplaça només modifica la posició de l'entitat càmera i no l'entitat *rig*, per lo que modificar la posició de l'entitat *rig* a la posició inicial (0, 0, 0) no canvia res. Per solucionar aquest problema s'ha pensat que la millor idea és canviar la posició del *rig*, però a una posició negativa en els eixos X i Z respecte de la posició de la càmera. És a dir, si el *rig* i la càmera tenen una posició inicial de 0, 0, 0 i el jugador es desplaça (canvia la posició de la càmera) a 2, 0, 3, llavors s'ha d'actualitzar la posició del *rig* a -2, 0, -3 perquè el jugador torni estar al punt inicial.

Aquesta solució funciona perfectament si la posició inicial del *rig* és 0, 0, 0, però al joc quan el jugador es troba dins l'oficina l'entitat del *rig* no es troba a la posició 0, 0, 0 i per tant aquesta funció falla. El problema és que la posició de la càmera és relativa al *rig* i no és absoluta en referència a tota l'escena. Gràcies a una altre resposta de Stack Overflow [14] s'han pogut convertir les coordenades locals d'una entitat a les coordenades absolutes de l'escena. Adaptant el codi amb les coordenades absolutes s'ha aconseguit implementar aquesta funció amb èxit. El codi es pot veure a la Figura 3.20.

Una vegada es té la funció que centra el jugador falta assignar-li un botó del controlador perquè el jugador la pugui usar. Per això, s'usa un component que ja s'ha registrat prèviament i s'ha explicat a l'apartat 3.4.1.3.5 (teclat). Aquest component està assignat a l'entitat del controlador i escolta els esdeveniments dels botons del controlador. En aquest cas ens interessava (per complir el requisit RNF12) que el botó X del controlador sigui el que centra el jugador, per lo que escoltem l'esdeveniment *xbuttondown* i revisem si l'usuari està dins l'oficina o defora, per centrar al jugador a la posició corresponent.

3.4.1.18. Música i sons

Per augmentar el realisme i l'experiència dels usuaris i proporcionar feedback s'ha decidit implementar música i sons dins el joc. Aquests elements no aporten funcionalitat al joc però si que el fan més atractiu i més professional.

En primer lloc, s'ha registrat un component que reprodueix un so quan es passa el ratolí o el raig del controlador per damunt (*mouseenter*) l'entitat que el té assignat. Aquest component també reprodueix un altre so quan es fa clic (*click*) sobre dita entitat. Tots els botons del joc duen assignat aquest component.

En segon lloc, s'ha afegit una petita melodia que es reprodueix quan l'usuari finalitza una prova i es mostra la puntuació final, indicant que ha finalitzat la prova amb èxit.

En tercer lloc, s'han aplicat sons al col·locar i descol·locar les peces de l'ordinador, proporcionant així feedback als usuaris de que la peça s'ha col·locat/descol·locat.

Per últim, s'ha inclòs una música que es reprodueix en bucle durant tot el joc. Aquesta música és relativament pausada i té un volum baix. No obstant, els jugadors poden canviar el volum de la música mitjançant una funció implementada i que s'explicarà al següent apartat.

3.4.1.18.1. Modificació del volum

Canviar el volum de la música des del codi no és massa complicat ja que hi ha un atribut (*volume*) a JavaScript que permet modificar el volum d'un so donant-li valors decimals entre 0 i 1. Lo vertaderament enginyós és implementar una interfície perquè els jugadors puguin canviar el volum des de qualsevol lloc del joc i no haver d'entrar a les opcions de configuració per fer-ho. A més, s'ha pensat que una bona idea seria implementar alguna cosa similar a un *slider* per poder canviar el volum sense haver d'introduir cap valor.

Per implementar l'*slider* s'ha utilitzat un component que es diu *aframe-gui*³³ i que té una gran quantitat d'elements (panels, botons, etiquetes, *radio buttons, sliders*...) que funcionen dins A-Frame. D'aquest component únicament ens interessa l'*slider*, el qual s'ha implementat sense problemes seguint la documentació del propi component. Cada vegada que es modifica la posició de l'*slider* s'executa una funció la qual rep el valor de l'*slider* (entre 0 i 1, justament el mateix rang de valors que maneja l'atribut del volum). Aquest valor s'utilitza per modificar el volum de la música i es guarda dins una variable global perquè quan es torni obrir l'*slider* estigui a la posició que s'ha deixat la darrera vegada. La interfície de volum amb l'*slider* es pot veure a la Figura 3.21.



Figura 3.21 Interfície modificació de volum

No obstant, encara queda implementar que aquesta interfície sigui accessible des de qualsevol part del joc. Per això, s'ha decidit que amb el botó Y del controlador (com s'ha especificat al requisit RNF13) s'obrirà i es tancarà aquesta interfície. Com ja s'ha explicat a altres apartats es té registrat un component que escolta esdeveniments dels controladors, per lo que basta afegir un nou escoltador quan es premi el botó Y (*ybuttondown*) que mostri aquesta interfície si no existeix, i si ja existeix, que l'elimini.

³³ <u>https://github.com/rdub80/aframe-gui</u>

Per fer que la interfície aparegui enfront de l'usuari, estigui on estigui, es té una entitat que és filla de l'entitat de la càmera i que està posicionada 1 metre enfora d'ella en l'eix Z. Així, quan s'ha de crear la interfície del volum s'agafen les coordenades absolutes d'aquesta entitat i es posiciona la interfície a aquella posició. Per fer que aquesta interfície sempre quedi de cara a l'usuari encara que aquest es mogui una vegada ja generada la interfície, s'usa el component *look-at*³⁴, el qual modifica la rotació d'una entitat (en aquest cas la interfície de volum) perquè sempre quedi mirant cap a una altre entitat (la càmera del jugador).

3.4.1.19. Rendiment

Durant el desenvolupament del joc s'han tingut alguns problemes de rendiment, sobre tot en dispositius no massa potents. Com que tota la implementació es duia a terme i es provava a un ordinador relativament potent (i9 9900K i GTX 1050 Ti) no sorgien problemes de rendiment (60 fps), però al provar-ho a altres dispositius amb especificacions menors el rendiment era inacceptable (10 fps). Per tractar de millorar el rendiment s'ha llegit la documentació de A-Frame i s'ha trobat un apartat que justament dona consells per millorar el rendiment [15]. Llavors, les millores aplicades són les següents.

En primer lloc, tal i com s'ha explicat a l'apartat 3.4.1.1.1 de textures, s'han carregat totes les textures i els sons a l'*Asset Management System* d'A-Frame perquè les textures es carreguin al principi de l'execució i així no afectin al rendiment.

En segon lloc, s'ha aplicat el component *Instanced Mesh*, explicat a l'apartat 3.4.1.1.2, per disminuir el nombre de cridades i entitats ja que totes les tanques es troben unides com si es tractés d'una sola entitat, per tant només s'usa una cridada per cada grup de tanques.

En tercer lloc, s'ha decidit crear part de l'escenari exterior i interior de manera dinàmica i eliminar totes aquelles entitats que no es veuen des de la posició on es troba el jugador. Per exemple, quan el jugador està a la zona exterior únicament es mostra la taula de l'interior de l'oficina però totes les altres entitats (taules de portàtils, menús...) no hi son. Al desplaçar-se a l'interior de l'oficina es creen, dinàmicament, totes aquestes entitats i també s'eliminen aquelles entitats situades a l'exterior de l'oficina que no es necessiten (menú principal, menú auxiliar, decorats del sòtil de la part exterior de l'oficina...).

En quart lloc, s'ha detectat que el problema fonamental del baix rendiment es devia a les ombres que reflectien les llums sobre cada un dels objectes, ja que s'havia de calcular de quina manera incidia la llum i sobre quins altres objectes s'havia de reflectir. Això s'ha solucionat desactivant per complet totes les ombres, sacrificant així un poc de realisme, però a canvi de que el joc es pugui jugar a pràcticament qualsevol dispositiu.

Per últim, i també parlant de la il·luminació, s'ha decidit limitar la il·luminació de l'escena, estant únicament il·luminada la zona on es troba el jugador. Per exemple, quan el jugador

³⁴ <u>https://github.com/supermedium/superframe/tree/master/components/look-at/</u>

es troba fora de la oficina la llum il·lumina aquesta part però, quan es troba a l'interior, aquesta mateixa llum es canvia de posició per il·luminar la zona interior. D'aquesta manera no s'ha de tenir tot il·luminat alhora, sinó que únicament s'il·lumina allò que es necessita, amb una il·luminació relativament simple.

3.4.1.19.1. Quest Browser

La principal motivació de millorar el rendiment del joc era que funcionés correctament al navegador Quest Browser³⁵. Aquest navegador ve instal·lat amb els dispositius Quest 1 i Quest 2 i bàsicament té totes les funcionalitats que pot tenir qualsevol altre navegador, però amb la peculiaritat de que funciona dins VR, sense necessitat d'un ordinador. És a dir, només amb unes Quest 1 o Quest 2 i una connexió Wi-Fi estable es pot navegar per Internet. Això significa que també es podria jugar al PC Builder VR únicament amb un dispositiu VR, sense un ordinador.

Gràcies a les millores de rendiment explicades a l'apartat anterior s'han aconseguit millorar molt els fps quan es juga des del navegador Quest Browser (15 fps abans i 70 fps ara). Tot i que el joc es veu millor si es juga amb un dispositiu VR connectat a l'ordinador, es pot jugar perfectament a través de les Quest 1. A les Quest 2 el rendiment hauria de ser notablement superior i similar al d'un ordinador ja que, tal i com es parla a aquest article [16], les Quest 2 tenen un processador molt superior a les Quest 1 i, a més, tenen 6 GB de RAM enfront dels 4 GB de les Quest 1.

3.4.1.20. Seguretat

Per tal d'augmentar la seguretat del codi i no guardar contrasenyes en text pla dins el codi s'ha fet ús d'un fitxer de configuració (*pcbuilder.config*) el qual emmagatzema les credencials per accedir a la base de dades. Abans d'aquesta implementació les credencials tampoc eren visibles pels usuaris perquè estaven al codi PHP de la part del servidor, però encara així, sempre és millor no escriure contrasenyes directament al codi.

Ara, el codi PHP en lloc d'accedir directament a la base de dades amb les credencials, primer llegeix el fitxer .config esmentat i obté les credencials corresponents. Això també suposa una millora de cara a la flexibilitat, ja que si s'ha de canviar la contrasenya de la base de dades únicament s'han de modificar les credencials del fitxer .config i no de cada fitxer PHP que hi ha al servidor.

3.4.2. Interfície administradors

Els dissenyadors de peces i proves han de poder dur a terme els requisits descrits als apartats 3.2.2.1.2 i 3.2.2.1.3. Per això, s'ha decidit implementar una interfície que serveixi com a editor de peces i proves. La interfície es troba dividida en dues parts, perquè els dissenyadors només puguin accedir a l'apartat que els pertoca. No obstant, les dues parts es troben implementades de manera molt similar, per lo que s'explicaran dins aquest

³⁵ <u>https://www.oculus.com/experiences/quest/1916519981771802/?locale=es_ES</u>



mateix apartat, realitzant les distincions necessàries quan es parli de la interfície de peces o la de proves.

Figura 3.22 Submòduls de la implementació de la interfície d'administradors

De la mateixa manera que al mòdul del joc en VR aquí també es poden veure, de manera gràfica i gràcies a la Figura 3.22, tots els mòduls de la interfície d'administradors i com es relacionen entre sí.

3.4.2.1. Login

Per accedir a la interfície són necessàries unes credencials de dissenyador de peces o proves, segons l'apartat al que es vulgui accedir. Aquestes credencials s'introdueixen a una finestra com la que es mostra a la Figura 3.23 i s'envien al servidor, de la mateixa manera que s'inicia sessió dins el joc, però amb la peculiaritat que el servidor revisa que efectivament aquelles credencials pertanyen a un dissenyador de peces o proves (segons la interfície). Una vegada s'ha iniciat sessió correctament es mostra un llistat de les peces/proves que hi ha dins la base de dades.

En aquesta interfície també es fa ús de PHP Session, i funciona de la mateixa manera que s'ha explicat a l'apartat 3.4.1.4. Això evita tornar a posar les credencials quan es recarrega la pàgina, però deixa de funcionar quan l'usuari tanca el navegador, requerint així que es tornin a introduir.

S'ha habilitat un botó al cantó superior dret per tancar la sessió actual, cosa que pot ser útil per canviar d'usuari o simplement per no deixar la sessió oberta. Aquest botó envia una petició al servidor per esborrar completament les dades de sessió de la pàgina actual. Posteriorment es recarrega la pàgina perquè torni a sortir la interfície d'introducció de les credencials.

Zona administradores de piezas
Credenciales: usuario
contraseña
Acceder

Figura 3.23 Login interfície peces

3.4.2.2. Bootstrap

Bootstrap és una llibreria per facilitar el disseny en entorns web. Proporciona multitud d'eines i plantilles sobre HTML, CSS y JS. En aquest cas s'ha usat Bootstrap pel navbar, les targetes de cada peça/prova i el sistema de grid.

En el cas del navbar³⁶ s'ha implementat el més senzill que ofereix Bootstrap i s'han modificat les etiquetes, la imatge i els colors, d'acord amb l'estètica desitjada.

Les targetes³⁷ s'han usat únicament com a contenidor ja que només tenen text, sense cap imatge. No obstant s'han respectat les etiquetes *card-title*, *card-body* i *card-text* per motius d'accessibilitat i per evitar canviar la mida de la font.

El sistema de grid³⁸ s'ha utilitzat a pràcticament tota la interfície, per posicionar els elements a la posició desitjada i que sigui web responsive. Per exemple, totes les especificacions de les peces i les proves, així com la introducció de dades de cada peça/prova estan posicionades usant la grid de Bootstrap.

3.4.2.3. Icones

Totes les icones de la interfície d'administradors s'han obtingut de la llibreria Font Awesome³⁹. Aquesta llibreria conté una gran quantitat d'icones per usar de manera gratuïta, únicament s'ha de crear un compte a la seva web. A aquestes icones se les hi ha aplicat CSS per canviar-li el color i convertir-les en botons, canviant el cursor per un punter i ficant-les dins una etiqueta de tipus *a*.

³⁶ <u>https://getbootstrap.com/docs/5.0/components/navbar/</u>

³⁷ https://getbootstrap.com/docs/5.0/components/card/

³⁸ <u>https://getbootstrap.com/docs/5.0/layout/grid/</u>

³⁹ <u>https://fontawesome.com/</u>

3.4.2.4. Targetes peces/proves

Les targetes de proves (veure Figura 3.25) contenen informació bàsica de cada prova com el nom, la descripció i la dificultat. En canvi, les targetes de les peces (veure Figura 3.24) son més complexes que les de les proves perquè cada tipus de peça té especificacions diferents, mentre que totes les targetes de les proves presenten els mateixos camps (nom, descripció i dificultat). La implementació realitzada per solucionar aquesta complexitat és la següent.

En primer lloc s'ha de detectar de quin tipus de peça es tracta. Això és molt senzill perquè l'*array* proporcionat per el PHP del servidor és multidimensional i està indexat pel nom del tipus de peça. D'aquesta manera ja es poden mostrar totes les peces ordenades segons el seu tipus.

En segon lloc, per mostrar els camps de les especificacions s'ha definit una variable de tipus JSON que conté, per a cada tipus de peça, el nom i la unitat de cada especificació. Per exemple, en el cas de la capacitat dels discs el tipus de peça és "disco", el nom de l'especificació és "Capacidad:" i la unitat és "GB". Aquest JSON té les especificacions ordenades seguint l'ordre de l'*array* que ha retornat el servidor, per lo que per mostrar les especificaciós, el valor (proporcionat per l'*array*) i la unitat.

És important comentar que també s'ha implementat l'opció de mostrar únicament les peces d'un sol tipus, cosa molt senzilla de programar ja que, com s'ha dit abans, l'array de les peces està indexat segons el tipus de peça, per lo que basta mostrar únicament totes les peces de l'array amb l'índex (tipus de peça) que es desitja.

Listado de Piezas Todos	r Sesión
+ Nueva Pieza	C→ Cerrar Sesión Listado de Pruebas
Placa Base	• Nueva Prueba
Asus TUF GAMING X570-PLUS	Tutorial
Socket CPU: AM4 Slots RAM: 4 Slots HDD: 8	Descripción: Tutorial básico Dificultad: Fácil
Slots SSD: 2 Detalles C Editar Eliminar	Detalles 🕑 Editar 🛇 Eliminar

Figura 3.24 Targeta peça

Figura 3.25 Targeta prova

3.4.2.5. Visualització de peces/proves

Al clicar sobre el botó de "Detalles" es mostra tota la informació de cada peça o prova. Aquesta informació és més extensa que la que es mostra a les targetes explicades anteriorment perquè, en el cas de les peces (veure Figura 3.26), es mostra també la descripció, les dimensions, la textura, la marca i el model. Aquestes dades estan contingudes dins l'*array* de peces i es mostren a una finestra creada en JavaScript. La plantilla de les dades és la mateixa per a totes les peces a excepció de les especificacions pròpies de cada tipus de peça les quals es mostren seguint la mateixa dinàmica del JSON explicada a l'apartat anterior.

Descripción:	Marca: Asus
blabla	Modelo: TUF GAMING X570-PLUS
Dimensiones (ancho X alto X fondo):	Tipo: Placa Base
240mm x 30mm x 300mm	Tipo: ATX
Textura:	Socket CPU: AM4
	Slots RAM: 4
	Slots HDD: 8
	Slots SSD: 2

Figura 3.26 Interfície de visualització d'una peça

En el cas de les proves i tal i com es pot comprovar a la Figura 3.27, primer es mostren les dades comuns (nom, descripció i dificultat) i després les tasques d'aquella prova específica. Les tasques es troben contingudes dins un *array* proporcionat pel servidor, el qual conté indexades totes les proves i totes les tasques. Per recopilar totes les tasques d'una prova simplement es revisa el camp corresponent de cada tasca, el qual indica a quina prova pertany. Tot això es mostra de la mateixa manera que les peces, dins una finestra que s'obri després de clicar sobre el botó "Detalles" de la targeta d'una prova específica.

Tutorial	×
Descripción: Tutorial básico	Dificultad: Fácil
Tarea 1 Descripción: Memoria RAM mínima: 0 GB	
Tarea 2 Descripción: Memoria HDD mínima: 0 GB	
Tarea 3 Descripción: Memoria SSD mínima: 0 GB	
Edi	tar Eliminar

Figura 3.27 Interfície de visualització d'una prova

3.4.2.6. Eliminació de peces/proves

L'eliminació de peces o proves es pot realitzar des de les targetes, des de la finestra de visualització o fins i tot des de la finestra d'edició, explicada al següents apartats. Eliminar una peça o prova és una tasca molt senzilla ja que únicament s'ha d'enviar una petició al servidor perquè realitzi l'eliminació de la base de dades. No obstant, aquesta operació és molt delicada perquè és irreversible.

Per assegurar-se que realment es vol dur a terme l'eliminació es mostra una finestra emergent avisant al dissenyador de que l'operació és irreversible. Només si aquest accepta, la peça o prova s'elimina. A més, també es revisa a la banda del servidor que les credencials són les correctes i que l'usuari té els permisos adequats (dissenyador de proves o peces, segons el que es vol eliminar) per augmentar encara més la seguretat i per prevenir atacs d'usuaris experts que puguin realitzar peticions al nostre servidor.

S'ha de recalcar que a l'hora d'eliminar una peça simplement s'elimina la peça en qüestió, però, al eliminar una prova, primer s'han d'eliminar les tasques associades, llavors les estadístiques i finalment la pròpia prova. Això és perquè sinó es produiria un error a la base de dades al intentar eliminar una prova que es clau forana de les taules "tarea" i "estadística".

3.4.2.7. Edició de peces

Al fer clic sobre el botó "Editar" de qualsevol targeta s'obre una finestra (veure Figura 3.28) amb els mateixos camps descrits a l'apartat 3.4.2.5 de visualització de peces/proves. La gran diferència és que aquests camps són editables i el dissenyador pot emplenar els camps segons consideri.

Asus TUF GAMING X570-PLUS			
Descripción:	Marca:	Asus	
blabla	Modelo:	TUF GAMING X57	70-PLUS
	Tipo de pieza:	Placa Base	`
	Тіро:	ATX	```
Jimensiones en mm (ancho X alto X ondo):	Socket CPU:	AM4	
240 X 30 X 300	Slots RAM:	4	`
extura (subir o seleccionar):	Slots HDD:	8	``
Examinar No se ha seleccionado ningún archivo.	Slots SSD:	2	`
Default Texture			

Figura 3.28 Interfície d'edició d'una peça

En el cas de la textura es mostra en el menú desplegable si la textura és la que ve per defecte (*default*) o si és una textura pujada per un dissenyador. En qualsevol cas es pot triar una de les textures del menú desplegable, o pujar-ne una de pròpia. Per gestionar això es guarda la textura pròpia (només n'hi pot haver una per peça) al servidor amb el mateix nom de la peça seguit de *.jpg* (per exemple, "cmp-cpu-2.jpg"). Al carregar la finestra d'edició d'aquesta peça es revisa si la textura assignada no és la *default* i es carrega al menú desplegable la textura pròpia ("cmp-cpu-2.jpg") i la *default*. Si es puja una textura pròpia i ja n'hi ha una altre, l'anterior es sobreescriurà amb el mateix nom que l'anterior.

3.4.2.7.1. Control d'errors

Per evitar errors a l'hora de introduir dades (d'una peça o prova) a la base de dades es realitza un fort control d'errors a la banda del client. S'ha pensat que no és necessari realitzar el control d'errors a la banda del servidor perquè els únics usuaris que poden accedir a aquesta interfície són els dissenyadors de peces/proves. Aquests usuaris no són maliciosos ja que els interessa que el joc funcioni bé, per lo que no executaran comandes de codi amb el fi de burlar el control d'errors del client. El control d'errors implementat és el següent.

En primer lloc, no es pot guardar una peça/prova sense haver emplenat per complet tots els camps. Això ho revisa una funció que s'executa cada vegada que s'escriu en un camp de dades, comprovant si hi ha algun camp buit. Si hi ha camps buits el botó de guardar està desactivat. Si no, s'activa per poder fer clic.

En segon lloc, no es permeten introduir més caràcters dels que accepten els atributs de la base de dades. És a dir, si el camp "nom" de la base de dades només accepta 30 caràcters, el nombre màxim de caràcters que es poden introduir al editar el camp "nom" d'una peça és 30.

Placa Base	CPU	Ventilador	RAM	Disco	Fuente	GPU
Тіро	Arquitectura	Velocidad	Estándar	Тіро	Potencia	Memoria
SocketCPU	Socket		Velocidad	Velocidad		Velocidad
Slots RAM	Núcleos		Capacidad	Capacidad		Consumo
Slots HDD	Frec. Base					
Slots SDD	Frec. Turbo					
	Consumo					

Taula 3.4 Correspondència menú desplegable/camp de text

Per últim, s'ha de tenir en compte que, al igual que en la visualització de les peces, cada peça té uns camps d'especificacions únics, i aquests camps també han de ser editables.
Per solucionar això s'ha tornat a fer un JSON com el descrit a l'apartat 3.4.2.4 però amb la diferència de que s'especifica quines especificacions han de tenir un camp per escriure text i quines han de tenir un menú desplegable. La intenció dels menús desplegables és reduir el nombre d'errors possibles. Per exemple, si l'arquitectura de la CPU pot ser Intel o AMD, el menú desplegable mostrarà només Intel i AMD, evitant així que s'introdueixi manualment i que hi pugui haver errors ortogràfics o de majúscules/minúscules.

El funcionament del JSON és el següent. El JSON conté el títol de cada un dels camps de cada tipus de peça seguit de *null* si es pot introduir qualsevol text, o seguit d'un *array* d'opcions si s'ha de crear un menú desplegable. El codi JavaScript llegeix aquest camp i crea un camp de text o un menú desplegable (amb les opcions de l'*array*) segons pertoqui. Per tenir clar quins camps han de tenir menú desplegable i quins han de tenir un camp de text s'ha creat la Taula 3.4 que indica en verd els camps sense menú desplegable, i en blau els que tenen menú desplegable.

Per guardar els canvis de la peça editada s'ha d'usar el botó "Guardar" situat a la part inferior de la finestra, el qual actualitza la base de dades amb la peça modificada.

3.4.2.8. Edició de proves

La finestra d'edició de proves primer mostra els camps comuns de cada prova en format camp de text en el cas del nom i la descripció, i un menú desplegable en el cas de la dificultat, per evitar errors. Després, es mostren cada una de les tasques d'aquella determinada prova.

Les tres primeres tasques són d'emmagatzematge, tal i com s'ha explicat a l'apartat 3.4.1.10, per lo que únicament s'ha de poder editar el valor de la capacitat mínima de RAM, HDD y SSD. Això es reflecteix a la interfície inhabilitant tots els camps menys el de valor, com es pot veure a la Figura 3.29. Al editar el camp del valor es modifica automàticament la descripció de la tasca amb el valor introduït.

Tarea 3					
Descripción:	Memoria SSD mínima	a: 0 GB			
Tipo de pieza:	Disco	~			
Atributo:	Capacidad (GB)	~	Acción:	es mayor/igual que	~
Valor 1:	0				

Figura 3.29 Edició tasca d'emmagatzematge

Les tasques que no són d'emmagatzematge (veure Figura 3.30) tenen tots els camps habilitats i permeten triar un atribut d'un tipus de peça i aplicar-li un operador (=, !=, >, >=, <, <= i BETWEEN) per dissenyar la tasca que es desitgi. Aquí sorgeix la problemàtica de que hi ha atributs que no són compatibles amb certs operadors (per exemple, el tipus d'una placa base només pot ser igual o desigual a alguna cosa, per lo que els demés operadors no es poden aplicar). Això es resol amb un JSON similar als explicats

anteriorment, que conté cada atribut de cada tipus de peça seguit de *null* o d'un *array* de valors, segons si l'atribut és compatible amb tots els operadors o només amb els d'igualtat i desigualtat, respectivament. Això soluciona dos problemes alhora perquè, per una banda els atributs que tenen un *array* de valors mostraran un menú desplegable (igual que a l'apartat 0 d'edició de proves), i per una altra banda, tots aquests atributs només permetran que s'usin amb els operadors d'igualtat i desigualtat, tal i com es mostra a la Figura 3.30.

És important comentar que en el cas del *socket* de la CPU i de la Placa Base no es deixa a disposició dels dissenyadors un camp de text, sinó que surt un menú desplegable amb els *sockets* disponibles segons les peces que hi ha a la base de dades. Això és possible perquè al realitzar la petició de les proves el servidor també proporciona els *sockets* existents tant per la CPU com per la placa base.

Les tasques, a part de modificar, també es poden crear i eliminar per complir així amb el requisit RF20. Per crear una nova tasca s'ha de clicar el botó "Nueva tarea" situat al final de la finestra d'edició d'una prova. S'ha de tenir en compte que només es poden afegir, com a màxim, 10 tasques a cada prova i si s'intenta afegir-ne més s'indica amb un missatge emergent que el màxim és 10. Per eliminar una tasca s'ha de clicar el botó vermell amb el text "Eliminar tarea", el qual esborra la tasca seleccionada de l'*array* de tasques. A la Figura 3.30 també es poden veure els botons de crear i eliminar tasques.

Tarea 7						O Eliminar tarea		
Descripción:	Montar una fuente SR	X						
Tipo de pieza:	Fuente	~						
Atributo:	Тіро	~	Acción:		es igual	~		
Valor 1:	SFX	~			es igual			
					no es igual			
Piezas compatibles	: 1 (Pulsa Validar para	es mayor que						
		es mayor/igual que						
🕒 Nueva tarea					es menor que			
		es menor/igual que						
	Valio	lar	Guardar	Eli	está entre			

Figura 3.30 Edició tasca extra

3.4.2.8.1. Compatibilitat de peces

Per complir amb el requisit RF21 i poder visualitzar la quantitat de peces compatibles amb cada tasca creada es fa ús de la mateixa funció explicada a l'apartat 3.4.1.15.1, la qual revisa quins components són compatibles amb cada tasca, per saber si la tasca s'ha complert o no. En aquest cas, es realitza una petita variació d'aquesta funció perquè únicament ens interessa saber el nombre total de peces compatibles, no l'identificador de cada peça.

El nombre de peces compatibles de cada tasca es pot veure a la part inferior de cada tasca i s'actualitza mitjançant el botó "Validar" situat al final de totes les tasques, com es pot

veure a la Figura 3.30. Aquest botó envia la petició al servidor, el qual retorna un *array* amb la quantitat de peces compatibles amb cada tasca.

Aquesta funció és molt útil perquè permet que els dissenyadors de proves vegin si la tasca que estan creant és possible de dur a terme, o si hi ha massa peces compatibles i per tant la tasca és massa senzilla. No obstant, una limitació trobada és que potser hi hagi dues tasques que siguin contradictòries entre si. Per exemple, una tasca que només permeti plaques base ATX i una altre que només permeti Micro ATX. A les dues tasques sortirà que hi ha peces compatibles, però realment serà impossible dur a terme les dues tasques alhora. Això s'ha de tenir en compte per part dels dissenyadors.

3.4.2.9. Creació de peces/proves

Per crear noves peces o proves s'ha de clicar el respectiu botó "Nueva pieza"/"Nueva prueba" situat a la part superior de les interfícies de peces i proves, tal i com es pot veure a la Figura 3.24 i a la Figura 3.25. Aquests botons obrin una finestra exactament igual que la d'edició de peces/proves, però amb la diferència de que tots els camps es troben buits, com és lògic perquè la peça/prova és nova. En el cas de la prova ja es creen les 3 tasques d'emmagatzematge perquè, com s'ha dit abans, són obligatòries.

Al clicar el botó "Guardar" es realitza la inserció a la base de dades amb la peça/prova creada. Per saber quin identificador li correspon es revisa l'identificador de la darrera peça/prova de la base de dades i es suma 1 a aquest identificador.

3.4.3. Interfície estadístiques

Aquest darrer mòdul de la implementació és el més senzill dels tres perquè únicament mostra les estadístiques de tots els jugadors i unes gràfiques relacionades amb les visites del joc. Per tal d'organitzar aquest apartat es dividirà segons els dos submòduls connectats al submòdul principal de la Figura 3.31, la taula i les visites.



Figura 3.31 Submòduls de la implementació de la interfície d'estadístiques

3.4.3.1. Taula d'estadístiques

S'ha pensat que la millor manera de mostrar totes les estadístiques del joc és mitjançant una taula que permeti aplicar filtres i veure els resultats de manera ordenada. Per això, s'ha usat un *plug-in* per la llibreria de JQuery anomenat DataTables⁴⁰. Aquest *plug-in* permet crear taules, tant en HTML com en JavaScript, definit el títol de les columnes i el contingut de cada una de les files. Un dels avantatges que presenten aquestes taules és que inclouen funcions d'ordenació de columnes, sense haver de configurar res, perquè l'ordenació és realitza a nivell de cadena de caràcters, cosa que ens serveix per a totes les columnes de la nostra taula (veure Figura 3.32) i fa que es compleixi el requisit RF11. A més, també permet canviar el nombre de files que surten a cada pàgina de la taula, navegar entre pàgines i realitzar cerques, les quals s'han configurat perquè es realitzin sobre la columna dels usuaris, complint així el requisit RF10.

La única configuració realitzada al *plug-in* ha estat la modificació de totes les etiquetes, perquè estiguin en l'idioma desitjat, i limitar a quines columnes es poden aplicar ordenacions (s'ha de poder a totes excepte la que indica el nombre d'estadística, marcada amb un '#') i cerques (únicament es permet a la dels usuaris, com s'ha dit abans). També s'ha configurat correctament la columna '#' perquè s'actualitzi cada vegada que es canvia l'ordre de les files.

Per tant, a l'hora de mostrar la taula es realitza una petició al servidor per obtenir totes les estadístiques i el nom de totes les proves, per poder aplicar els filtres explicats al següent apartat.

Estadisticas											
Pruebas	: Todas	~	Usuar	ios:	Todos	~	Columnas:	Tod	as 💌)
Mostrar	5 × esta	dísticas p	por página				Buscar usuario	o:			
#	Usuario	\$	Prueba	$\stackrel{\wedge}{\nabla}$	Fecha	$\stackrel{\mathbb{A}}{\nabla}$	Tiempo	\$	Pur	ntuació	in 🔻
6	Invitado2	1	Nivel 1		2022/04/27		00:01		0.00		
7	jaume	I	Futorial		2022/04/27		00:00		0.00		
8	Invitado	1	Futorial		2022/04/27		00:05		0.00		
9	carlos	1	Nivel 1		2022/04/28		00:05		0.00		
10	jaume	1	Nivel 1		2022/05/03		00:18		0.00		
Mostrano	do página 2 de	2					Anter	ior	1	2	Siguiente

Estadísticas

Figura 3.32 Interfície taula d'estadístiques

⁴⁰ <u>https://datatables.net/</u>

3.4.3.1.1. Filtres

Per complir amb el requisit RF09 que parla d'aplicar filtres a les estadístiques s'han implementat dos menús desplegables que contenen totes les proves del joc (obtingudes de la petició al servidor) i els tipus d'usuaris (registrats i no registrats), respectivament. Al aplicar qualsevol dels dos filtres es calculen les estadístiques que els compleixen, es guarden dins un *array* i es recarrega tota la taula perquè es mostrin únicament les estadístiques contingudes dins aquest *array*. Es poden aplicar els dos filtres alhora perquè el codi implementat revisa que les estadístiques compleixin ambdós filtres.

3.4.3.1.2. Web responsive

Aquesta taula creada amb el *plug-in* DataTables ja és web responsive, però quan la pantalla és massa petita hi ha certes columnes que surten de la pantalla i requereixen desplaçar la taula lateralment per visualitzar-la sencera. Això és completament normal perquè es tracta d'una taula amb vàries columnes però seria molt adequat que es poguessin ocultar columnes perquè la visualització, sobretot a dispositius mòbils, sigui més còmoda.

Per implementar aquesta funció s'ha utilitzat un altre *plug-in* de JQuery anomenat *multi-select*⁴¹, el qual permet crear un menú desplegable amb *checkboxes* per a cada opció. Amb aquesta interfície es poden mostrar quines columnes són visibles i permetre als usuaris marcar i desmarcar les columnes que desitgin.

Quan es marca o desmarca una opció s'executa una funció la qual mostra o oculta, mitjançant l'atribut "visible" del *plug-in* DataTables, totes aquelles columnes que estan marcades al menú desplegable. Cal destacar que s'ha hagut de configurar el *plug-in multi-select* perquè mostri les etiquetes "Seleccionar" i "Todas" quan no hi ha res seleccionat i quan està tot seleccionat, respectivament.

Tant els menús desplegables dels filtres com el de mostrar/ocultar columnes es poden veure a la Figura 3.33, on es troben desplegats perquè es visualitzin totes les opcions.





⁴¹ <u>https://plugins.jquery.com/multi-select/</u>

3.4.3.2. Visites del joc

Per poder visualitzar les visites, per dia i per país, del joc i complir així amb els requisits RF12 i RF13, s'ha implementat una gràfica i un mapa del món (veure Figura 3.34) mitjançant la llibreria Highcharts⁴². Aquesta llibreria de JavaScript serveix per mostrar gràfiques i mapes de manera senzilla, únicament havent de passar-li les dades amb el format adequat i configurar els colors, els títols i les etiquetes. La utilització de la llibreria Highcharts no ha suposat cap problema perquè la documentació és prou concisa i no és la primera vegada que faig ús d'aquesta llibreria.

En el cas de la gràfica de visites per dia s'ha usat la plantilla *Time series zoomable*⁴³ de Highcharts la qual permet mostrar una línia temporal amb valors diaris. Aquesta gràfica té la peculiaritat de que permet fer zoom, cosa que facilita la visualització i ofereix tant una visió global com una visió específica de les visites al joc.

Per al mapa s'ha usat la plantilla més simple de Highcharts Maps⁴⁴ que permet mostrar tots els països del món ressaltats segons el valor indicat. En aquest cas el valor és el nombre de visites per cada país. El mapa també permet fer zoom.



Visitas



3.4.3.2.1. Emmagatzematge de les dades

Les dades de les visites per dia i per país es troben emmagatzemades a dos fitxers JSON, respectivament, al propi servidor. En el cas de les visites per dia el JSON està format per un *array* que conté, a cada posició, el dia en mil·lisegons i el nombre de visites d'aquell

⁴² https://www.highcharts.com/

⁴³ <u>https://www.highcharts.com/demo/line-time-series</u>

⁴⁴ <u>https://www.highcharts.com/demo/maps/all-maps</u>

dia. Per altre banda, el JSON de les visites per país conté un *array* amb tots els codis dels països del món (segons la ISO 3166-1⁴⁵) i el nombre de visites de cada país.

L'actualització de les dades emmagatzemades es realitza quan s'accedeix al joc. En aquest moment es realitza una petició a la API de IPinfo⁴⁶ per obtenir el codi del país en que es troba l'usuari. Aquesta API permet obtenir una gran quantitat de dades dels usuaris com la IP, la regió, la ciutat, el país... però en aquest cas amb el codi del país ens és suficient perquè el mapa de Highcharts es genera mitjançant els codis dels països. Una vegada es té el codi del país es realitza una petició al propi servidor, la qual es gestiona amb PHP, i s'encarrega de sumar 1 al comptador de les visites per dia i al de les visites per país (amb el codi obtingut).

Es podria donar el cas de que, si es tracta del primer usuari del dia que accedeix al joc, no existeixi cap posició de l'array del JSON que contingui dades del dia actual. Per solucionar això s'ha implementat una funció que genera posicions d'aquest array des del darrer dia existent a l'array fins al dia actual. A totes aquestes posicions se'ls hi assignen 0 visites perquè si no existeixen a l'array significa que ningú ha visitat la pàgina, exceptuant el dia actual al qual se li assigna 1 visita, fent referència a la visita actual. Aquesta funció també s'executa al visualitzar les estadístiques perquè així es mostrin actualitzades i es vegin tots els dies, encara que no hi hagi hagut visites. En aquest cas no s'assigna 1 visita al dia actual perquè les visites només comptabilitzen quan s'entra al joc, no quan s'accedeix la interfície d'estadístiques.

3.5. Proves

Durant el transcurs de la implementació i una vegada finalitzat aquest procés s'han dut a terme una sèrie de proves per tal de testejar el software implementat i eliminar la major part dels errors detectats, ja que eliminar-los tots és completament impossible perquè els softwares estan en un procés d'evolució i millora constant.

Tal i com s'ha explicat a l'apartat 3.1.1 les proves es complementen amb la metodologia àgil per prototipus per tal de canviar els requisits segons els resultats de les proves. Per aquesta raó, les solucions i modificacions preses durant el procés de proves han afectat als requisits, dels quals es pot consultar la versió definitiva a l'apartat 3.2.2.

Cal remarcar que les proves s'han realitzat únicament utilitzant els següents dispositius: teclat i ratolí, Oculus Quest 1 i Oculus Rift. Seria interessant fer proves amb altres dispositius de VR però no és possible degut a que no es tenen més dispositius. Podria ser que a altres dispositius sorgeixin nous errors, però això escapa de l'abast d'aquest projecte.

⁴⁵ https://es.wikipedia.org/wiki/ISO 3166-1

⁴⁶ <u>https://ipinfo.io/</u>

Per organitzar aquest apartat es separaran les proves realitzades per mi mateix (internes) i les proves que han realitzat usuaris externs (professors, familiars...) mentre jugaven al joc. A cada subapartat també s'explicaran els resultats obtinguts i les solucions preses.

3.5.1. Proves internes

En primer lloc, i tal com s'ha parlat a l'apartat 3.4.1.19 al realitzar proves sobre varis dispositius (portàtils, ordinadors, mòbils...) de diferent potència es va detectar que el rendiment no era acceptable, per lo que s'han aplicat les mesures explicades l'apartat 3.4.1.19 de rendiment, les quals inclouen eliminar les ombres, reduir el nombre d'entitats i limitar la il·luminació.

En segon lloc, durant les proves realitzades sobre els menús de la zona exterior (principal, configuració i auxiliar) s'ha trobat que si es tracta de mostrar el menú principal quan aquest ja s'està mostrant, es torna a crear un altre menú principal damunt el que ja existeix. Per solucionar-ho, es comprova abans de crear un nou menú principal si ja existeix. En el menú de configuració també s'han trobat diversos errors, sobre tot al iniciar o tancar la sessió usant el botó del cantó superior dret, perquè si ja s'havia carregat un apartat de configuració (per exemple, el de "Editar perfil") i es sortia de la sessió actual es seguia permetent modificar el perfil perquè el contingut d'aquesta pantalla no s'actualitzava després d'iniciar o tancar sessió usant l'opció superior. Això s'ha solucionat actualitzant la part del menú de configuració en que es troba l'usuari després d'iniciar o tancar la sessió, assegurant-nos així de que el que es mostra està actualitzat.

En tercer lloc, i parlant de la interacció al agafar i amollar les peces, hi havia un problema molt important que produïa un error que feia que el joc deixés de funcionar. Aquest error consistia en que, al intentar col·locar una peça, si aquesta peça no es col·locava correctament (és a dir, no estava a sobre cap zona de col·locació) i quedava sobre una altre peça ja col·locada a l'ordinador, després al intentar agafar la peça no col·locada i amollar-la a una zona de col·locació sorgia un error perquè s'havien "agafat" dues peces alhora, la mal col·locada i la que ja s'havia col·locat. Això es deu a que el component *super-hands* permet agafar múltiples entitats alhora, guardant-les totes dins un *array*. Per arreglar això en primer lloc es comprova que l'entitat agafada no estigui ja col·locada (les peces col·locades no es poden agafar, s'han de descol·locar primer), i a més es limita el nombre d'entitats que es poden agafar, per permetre una sola entitat i evitar futurs errors.

En quart lloc, a l'apartat 3.4.1.9.1 s'ha explicat que només es poden generar components duplicats en el cas de la RAM i els discos. Al principi de la implementació això no era així i no es permetia generar cap component duplicat perquè s'esborrava l'actual i se'n generava un de nou. Això no és correcte perquè s'haurien de poder col·locar varis mòduls de RAM i de discos que siguin exactament el mateix component duplicat. Es va modificar el codi per permetre generar únicament RAM i discos duplicats. Si bé això no és un error de programació, interfereix en el bon desenvolupament del joc per lo que s'ha canviat la mecànica per permetre únicament RAM i discos duplicats.

3.5. Proves

Per últim, en referència a la API de IPinfo explicada a l'apartat 3.4.3.2.1 s'han realitzat proves per comprovar que s'obtenen els codis dels països de manera correcta, i que llavors es visualitzen les dades al mapa. Aquestes proves s'han realitzant usant una VPN per poder canviar la IP i simular estar a un altre país. Les proves han estat satisfactòries però s'ha detectat que, si es té activat un bloquejador d'anuncis al navegador, la petició a la API no es realitza correctament perquè el bloquejador d'anuncis no la permet. Això fa que aquesta visita no es comptabilitzi a cap dels dos JSON (visites per dia i per país). Per aquest motiu, s'ha modificat el codi per actualitzar els fitxers JSON en dues peticions en lloc d'una, per assegurar-nos així que almenys es compatibilitza la visita al comptador de visites per dia. Tot i això, les visites per país no es comptabilitzen quan s'usa un bloquejador d'anuncis (o una VPN amb aquesta funció integrada) però és una restricció que s'escapa de l'abast d'aquest treball.

3.5.2. Proves d'usuaris externs

Aquest subconjunt de proves és crucial per a la solució d'errors perquè quan el propi desenvolupador prova el seu software ja sap com funciona i les passes realitzades solen ser sempre les mateixes. En canvi, quan un usuari extern prova el software es veuen diferents maneres d'usar el software i apareixen errors dels quals el propi desenvolupador no en tenia consciència. En aquest cas les proves han estat realitzades per familiars, amics i professors.

En primer lloc, el problema més comú entre els usuaris externs és que, al entrar dins una prova, el controlador dret deixa de ser un controlador i passa a ser una mà (*super-hands*). Això fa que pràcticament tots els usuaris tractin de fer clic sobre els botons usant la mà dreta de la mateixa manera que s'usa en el menú principal (amb el raig de llum) i clar això no està disponible i només es pot realitzar amb la mà esquerra. Per solucionar-ho s'ha decidit incloure un avís just al començar el tutorial (és la primera prova que solen realitzar els usuaris nous) que indica que en aquella zona del joc només es pot fer clic sobre botons usant el controlador esquerre ja que la mà dreta només serveix per agafar peces.

En segon lloc, un altre problema destacable és que una vegada col·locada una peça a l'ordinador no hi ha cap manera de veure les seves especificacions. Això fa que si un usuari vol comprovar, per exemple, quin *socket* té la placa base per col·locar una CPU amb el mateix *socket*, hagi de descol·locar la placa base, agafar-la per veure les especificacions i tornar-la a col·locar. Per això, s'ha implementat la funcionalitat explicada a l'apartat 3.4.1.14.3, la qual permet veure les especificacions d'una peça ja col·locada al apuntar-li amb el raig de llum del controlador esquerre.

En tercer lloc, s'ha comprovat que la majoria d'usuaris tenen problemes per saber si, al amollar una peça, aquesta s'ha col·locat a una zona de col·locació, o si pel contrari, simplement s'ha amollat damunt l'ordinador. Això es deu a que no hi ha cap tipus de feedback que indiqui que la col·locació s'ha dut a terme. Per tal de proporcionar feedback als jugadors i tal com s'ha explicat a l'apartat 3.4.1.18, s'han inclòs dos efectes de so, un

quan es col·loca una peça a una zona de col·locació i un altre quan es descol·loca. Així els jugadors són conscients de les col·locacions i descol·locacions.

Per últim, molts d'usuaris han demanat com es podia baixar o desactivar el volum de la música. Aquesta opció no estava implementada i s'ha pensat que és molt interessant perquè a tots els jocs es pot canviar el volum de la música. Per això s'ha implementat una petita interfície per canviar el volum de la música, tal i com s'explica a l'apartat 3.4.1.18.1.

3.6. Instal·lació

Aquest darrer apartat del capítol de desenvolupament tracta d'explicar, de manera ordenada, el procediment a seguir per tal d'instal·lar el joc i fer-lo públic. El joc en sí no requereix instal·lació perquè funciona sobre web, però encara així es necessari configurar un servidor que guardi el codi font i el mostri als usuaris.

El primer pas és muntar un servidor web. En aquest cas el servidor utilitzat és un contenidor *docker* que fa servir Ubuntu 16.04. Sobre aquest servidor s'ha instal·lat el servei d'Apache per poder servir contingut web. A part d'Apache també s'ha d'instal·lar MySQL⁴⁷perquè és el gestor de base de dades utilitzat, i PHP, el llenguatge de programació encarregat de gestionar el back-end i la interacció amb la base de dades MySQL. Per últim s'ha d'instal·lar phpMyAdmin⁴⁸ que és el software que permet gestionar la base de dades MySQL a través de la web. Això és útil per veure els contingut de la base de dades amb una interfície gràfica.

Una vegada ja s'ha instal·lat i configurat el servidor Apache amb els corresponents serveis s'ha d'introduir el codi del joc a l'arrel del directori Apache perquè estigui disponible a través d'una adreça web. El codi complet es troba al propi repositori de GitHub⁴⁹, incloent també el codi d'altres components que s'han hagut de modificar per satisfer les necessitats del joc. L'única configuració necessària és introduir al fitxer *pcbuilder_example.config* les credencials de la base de dades de MySQL configurada. A més, el nom d'aquest arxiu s'ha de modificar a *pcbuilder.config*.

Una vegada realitzat tot el que s'ha explicat en aquest apartat el joc hauria d'estar disponible a través de l'adreça web del propi servidor muntat. Cal recordar que hi ha elements del joc que només funcionen si es serveixen sobre *https* per lo que potser es produeixin errors si es fa servir el protocol *http*.

⁴⁷ https://www.mysql.com/

⁴⁸ https://www.phpmyadmin.net/

⁴⁹ <u>https://github.com/carlosveny/pc-builder-vr</u>

Capítol 4. Resultats

El resultat principal obtingut de la realització del projecte és una aplicació web que es divideix en varis mòduls. En primer lloc, es troba el mòdul del joc, accessible per a tots els usuaris a través d'un dispositiu de VR. En segon lloc, es té el mòdul d'administradors, el qual es divideix segons el tipus de dissenyador (peces o proves) i permet gestionar les peces i proves del joc. Per últim, trobem el mòdul de les estadístiques, que mostra les dades obtingudes de cada partida que juguen els usuaris, així com unes gràfiques que reflecteixen les visites al joc per dia i per país.

Cada un d'aquests mòduls resultants es pot consultar mitjançant les adreces web llistades a continuació. El joc complet en VR es troba a l'arrel del servidor, a l'adreça <u>https://alumnes-ltim.uib.es/pcbuilding/</u>. El mòdul d'administradors es pot consultar mitjançant les dues adreces següents, segons si es tracta de la interfície de peces o la de proves, respectivament: <u>https://alumnes-ltim.uib.es/pcbuilding/admin-piezas.html</u>, <u>https://alumnes-ltim.uib.es/pcbuilding/admin-pruebas.html</u>. Per últim, el mòdul d'estadístiques també té la seva pròpia interfície i es pot visualitzar mitjançant la següent adreça: <u>https://alumnes-ltim.uib.es/pcbuilding/estadisticas.html</u>.

Amb la finalitat de realitzar un recorregut per tota l'aplicació resultant, es dividirà aquest capítol en paràgrafs, segons els mòduls explicats anteriorment, i dels quals s'ha explicat la seva implementació a l'apartat 3.4.

Pel que fa al mòdul del joc el resultat final és completament funcional, tal i com s'ha explicat a l'apartat 3.5 de proves, i compleix amb les expectatives que s'esperaven del producte final. A més, després d'haver vist com certs usuaris (amics o familiars) juguen al joc, es pot afirmar que els seus coneixements de muntatge d'ordinadors han millorat notablement, fent que aquells que no sabien pràcticament res siguin capaços d'identificar la majoria dels components i saber la funció que realitzen dins l'ordinador. Tot això és possible gràcies al tutorial implementat perquè permet que els usuaris inexperts coneguin, no només els controls del joc, sinó per a què serveix cada peça i quin és el seu lloc de col·locació. A més, la resta de proves/nivells del joc han fet que els usuaris siguin conscients de les especificacions que tenen cada una de les peces i què representen. Per exemple, molts d'usuaris no coneixien el concepte del *socket* i, després de jugar als diferents nivells, són conscients de que el *socket* de la CPU ha de coincidir amb el de la placa base. Per últim, la majoria d'usuaris s'han sorprès al comprovar que s'ha de tenir molt present el consum de les peces a l'hora de muntar un ordinador, ja que la font d'alimentació ha de proporcionar la potència suficient per alimentar totes les peces.

4. Resultats

En el cas del mòdul dels administradors el resultat final són dues interfícies diferenciades, segons si s'accedeix a les peces o a les proves. Ambdues es troben protegides amb credencials perquè només els dissenyadors de peces o proves, respectivament, puguin accedir-hi. Aquestes interfícies permeten fer canvis sobre les peces i proves que hi ha a la base de dades, podent crear-ne de noves si es desitja. El funcionament d'aquest mòdul és l'esperat perquè permet gestionar les peces i proves del joc de manera satisfactòria.

Per últim, el mòdul d'estadístiques té com a resultat final una interfície que permet visualitzar la informació de cada partida (jugador, data, temps utilitzat i puntuació) que s'ha jugat. La visualització es realitza mitjançant una taula que permet aplicar filtres (segons la prova i el tipus d'usuari), mostrar/ocultar columnes, realitzar ordenacions segons les columnes i limitar el nombre de registres per pàgina. A més, a part de la taula d'estadístiques, en aquest mòdul també es troben un gràfic i un mapa, els quals mostren la quantitat de visites que ha tingut el joc segons la data i segons el país, respectivament. El funcionament d'aquest mòdul és adequat i molt convenient perquè proporciona informació sobre les partides jugades i sobre el tràfic que rep el joc, tant per data com per país.

Capítol 5. Conclusions

Amb la finalitat de concloure aquest treball es recorreran els objectius proposats a l'inici del document justificant i comprovant que s'han complert. També es farà una breu conclusió sobre l'estat de l'art una vegada dut a terme aquest treball. Per últim, es donaran opinions personals sobre el propi TFG, les dificultats i altres aspectes.

En primer lloc, l'objectiu principal era aconseguir desenvolupar un simulador de muntatge d'ordinadors que permeti triar els components i col·locar-los on es desitgi. Aquest objectiu s'ha complert amb èxit perquè el resultat final permet triar, d'un conjunt relativament ampli de components, el lloc de col·locació de cada component, sempre tenint en compte les tasques proposades a cada nivell.

Un altre objectiu feia referència a desenvolupar una interfície per poder actualitzar les peces, les proves i les seves respectives tasques, per tal de fer el joc dinàmic. Això s'ha aconseguit perquè les dues interfícies desenvolupades (peces i proves) permeten modificar els continguts del joc, tot de manera protegida mitjançant un sistema d'accés que requereix unes credencials pertanyents a un determinat rol d'usuari.

Es podria afirmar que l'objectiu de desenvolupar un software que sigui educatiu també s'ha dut a terme correctament perquè, després d'haver deixat provar el joc a amics i familiars amb nul·la experiència en hardware, molts d'ells han entès com funciona un ordinador i per a què serveix cada un dels seus components. Alguns d'ells han jugat vàries vegades i penso que serien capaços de muntar un ordinador a la vida real per si mateixos, òbviament sense tenir en compte aspectes com el cablejat, els cargols o la pasta tèrmica. No obstant, s'hauria de realitzar un estudi formal per comprovar i demostrar la capacitat que té el sistema per ensenyar aquestes tasques de muntatge.

Personalment, i en referència a l'objectiu d'adquirir coneixements sobre la programació en VR, estic gratament sorprès amb mi mateix perquè penso que he aconseguit desenvolupar, en relativament poc temps, un software bastant complet i funcional que funciona realment bé en realitat virtual, i tenint en compte que la meva experiència abans de començar era nul·la, puc afirmar que aquest objectiu s'ha superat de manera excel·lent.

Parlant un poc de l'estat de l'art, abans de dur a terme el treball les solucions existents eren poques i a més no eren exactament com la que s'ha desenvolupat. De fet, no hi havia cap joc en VR (que no fos un prototipus o que tingues alguna funcionalitat més enllà de col·locar peces una damunt l'altra) que permetés muntar ordinadors de la manera en que es pot fer al joc desenvolupat. A més, tots aquests jocs requereixen una instal·lació, no com el desenvolupat en aquest treball el qual funciona directament en el navegador,

5. Conclusions

sense descàrregues ni instal·lacions. Això fa que aquesta solució sigui molt més ràpida i accessible que les altres.

Una vegada dut a terme el treball penso que l'estat de l'art està en el mateix punt que abans de començar perquè no s'han introduït jocs nous similars, a excepció del desenvolupat en aquest treball, el qual es pràcticament únic en la seva categoria i ofereix prestacions similars als jocs de muntatge d'ordinadors existents, però amb la diferència de que és en realitat virtual, aportant així molt més realisme, immersió i percepció del que s'està fent.

No obstant, és important ser crític i parlar també dels aspectes millorables del joc. En primer lloc, es podria haver donat l'opció de modificar la mida de la capsa ja que en realitat existeixen varies mides, les quals condicionen els components que hi caben a dins (segons la mida de cada component). En segon lloc, es podria haver implementat una configuració software de l'ordinador a muntar, i que una vegada muntat s'hagi de posar en marxa, configurar la BIOS i instal·lar el sistema operatiu. En tercer lloc, també es podria haver implementat una tapa per poder tancar l'ordinador i una funció per poder permetre posar la capsa dreta. Aquestes darreres millores augmentarien molt la immersió del jugador i permetrien augmentar la component educativa del propi joc, instruint als usuaris sobre com es configura la BIOS o s'instal·la el sistema operatiu.

També és molt important parlar dels errors o les limitacions del software implementat. En aquest cas podem veure com, a la interfície de proves, es poden definir tasques contradictòries entre si perquè no es realitza cap tipus de control sobre el conjunt de les tasques, sinó que únicament es realitza sobre cada tasca en específic. També s'ha detectat algun problema al intentar agafar una peça quan està molt a prop o sobre altres peces perquè el joc es pensa que s'ha agafat una peça que en realitat no és la que el jugador té agafada, sinó la que estava a devora. Això fa que al col·locar-la es col·loqui la peça que realment no s'ha agafat. Per últim, si es col·loca una peça molt grossa al lloc incorrecte i es queda tocant vàries zones de col·locació (una vegada ja col·locada), totes les peces que s'amollin després es col·locaran automàticament a les zones que estan tocant la peça mal col·locada. Encara que això s'avisa al tutorial del joc, no dona una bona imatge i fa que sorgeixin nombrosos errors i l'experiència dels usuaris no sigui bona.

Durant el transcurs del treball s'han adquirit i après nombrosos coneixements, els quals s'explicaran a continuació, juntament amb la realització d'un breu repàs per les assignatures que més m'han servit per dur a terme el treball. En primer lloc, he de dir que l'àmbit de la informàtica que més m'agrada és la programació, i recentment la programació web. Això es deu a assignatures com Tecnologia Multimèdia (21755) o Gestió i Distribució de la Informació Empresarial (21753), les quals m'han introduït al món de la programació web sobretot en la capa de front-end, i un poc en la de back-end. Gràcies a aquestes dues assignatures he pogut progressar de manera molt ràpida en aquest treball perquè molts de coneixements de l'àmbit web ja els havia adquirit. No obstant, el tema del back-end no l'havia treballat massa i he hagut de millorar els meus coneixements de

PHP, així com aprendre a muntar i desplegar un servidor Apache. Pel que fa a les bases de dades, jo només he cursat l'assignatura de Base de Dades (21716) i trobo que a aquesta assignatura únicament es donen els coneixements més bàsics. Això va resultar un problema al principi del treball perquè es requeria pensar el disseny de la base de dades i després saber com interactuar amb ella a través de PHP. Gràcies a l'ajuda dels tutors i dels apunts de l'assignatura de Base de Dades s'ha pogut dur a terme la part més complexa de la base de dades i, a més, han millorat notablement els meus coneixements sobre aquest aspecte. Per últim, cal destacar la importància de l'assignatura de Gestió de Projectes (21723) per poder dur a terme la gestió d'aquest projecte, la qual es pot veure a l'apartat 3.1 d'aquest document.

Abans d'acabar trobo important dedicar un paràgraf a parlar de GitHub. A la carrera en cap moment se'ns ha ensenyat a usar eines com GitHub o Stack Overflow. Aquestes eines són molt potents perquè serveixen per trobar resposta a la gran majoria de problemes que sorgeixen mentre es programa. En concret, GitHub ha fet possible la realització d'aquest treball perquè he usat codi d'una gran quantitat de repositoris, he pogut solucionar els meus errors gràcies a les preguntes d'altres usuaris demanades a dits repositoris, i fins i tot he demanat ajuda i altres usuaris m'han ajudat. Un altre bon exemple és Discord, el qual m'ha servit per comunicar-me amb la comunitat d'A-Frame, rebre ajuda per parts de desenvolupadors com jo i també he ajudat als altres, en la mesura de lo possible. Avui en dia aquestes eines són pràcticament indispensables i penso que tots els enginyers informàtics haurien de saber treure-li el profit.

Per acabar, m'agradaria donar la meva opinió sobre tot el treball en sí i el propi joc desenvolupat. Trobo que la feina realitzada és molt completa i té molt de valor. És cert que s'han invertit moltes hores, moltes més de les invertides en qualsevol altre treball universitari que he realitzat, però ha valgut la pena. El resultat final penso que està molt encertat i reflexa perfectament els coneixements apresos durant el transcurs d'aquests 4 anys de carrera a la UIB. A més, és un projecte que segur que m'obre moltes portes al món laboral i em serveix de cara a les empreses per demostrar els meus coneixements i les meves aptituds informàtiques.

Capítol 6. Referències bibliogràfiques

- [1] «Reddit» [En línia]. Available: https://www.reddit.com/r/pcbuildingsimulator/comments/egsqtr/pc_building_i n_vr/. [Últim accés: 20 11 2021].
- [2] «Reddit» [En línia]. Available: https://www.reddit.com/r/pcmasterrace/comments/4ya9ix/could_someone_ma ke_a_pc_building_simulator_for_vr/. [Últim accés: 20 11 2021].
- [3] «Reddit» [En línia]. Available: https://www.reddit.com/r/virtualreality/comments/mgjqa4/is_there_a_vr_pc_b uilding_simulator_dying_to/. [Últim accés: 20 11 2021].
- [4] «Reddit» [En línia]. Available: https://www.reddit.com/r/virtualreality/comments/grw7ky/imagine_playing_pc _building_simulator_in_vr/fs2jqhh/?utm_source=share&utm_medium=web2x&c ontext=3. [Últim accés: 20 11 2021].
- [5] JonathanVusich, «GitHub» [En línia]. Available: https://github.com/JonathanVusich/pcpartpicker. [Últim accés: 13 01 2022].
- [6] A-Frame, «Text component» [En línia]. Available: https://aframe.io/docs/1.3.0/components/text.html. [Últim accés: 23 02 2022].
- [7] iangilman, «GitHub» [En línia]. Available: https://github.com/supermedium/aframe-super-keyboard/issues/17. [Últim accés: 27 04 2022].
- [8] carlosveny, «GitHub» [En línia]. Available: https://github.com/elbobo/aframemultisrc-component/issues/1. [Últim accés: 04 04 2022].
- [9] A-Frame, «Interactions and Controllers» [En línia]. Available: A-Frame. [Últim accés: 4 03 2022].

6. Referències bibliogràfiques

- [10] carlosveny, «GitHub» [En línia]. Available: https://github.com/wmurphyrd/aframe-super-hands-component/issues/219.
 [Últim accés: 11 03 2022].
- [11] Ava, «Stack Overflow» [En línia]. Available: https://stackoverflow.com/questions/64902347/aframe-problem-with-physicsdynamic-body-grabbable. [Últim accés: 12 03 2022].
- [12] kylebakerio, «GitHub» [En línia]. Available: https://github.com/wmurphyrd/aframe-super-handscomponent/blob/master/getting-started.md. [Últim accés: 12 03 2022].
- [13] ngokevin, «Stack Overflow» [En línia]. Available: https://stackoverflow.com/questions/36677671/aframe-set-camera-position-atruntime. [Últim accés: 02 04 2022].
- [14] ngokevin, «Stack Overflow» [En línia]. Available: https://stackoverflow.com/questions/38363963/is-there-a-way-to-know-thereal-position-of-an-a-frame-element. [Últim accés: 05 04 2022].
- [15] A-Frame, «Performance» [En línia]. Available: https://aframe.io/docs/1.3.0/introduction/best-practices.html#performance.
 [Últim accés: 17 03 2022].
- [16] «VR Expert» [En línia]. Available: https://vr-expert.com/the-most-importantdifferences-between-the-oculus-quest-1-and-2/. [Últim accés: 21 05 2022].