

## Treball final de grau

# GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica Universitat de Barcelona

# Desenvolupament d'entorns interactius mitjançant tecnologia de realitat virtual no immersiva

Autor: Leying Wang

Directora: Dra. Mireia Ribera Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 6 de juny de 2024

### Abstract

The use of virtual reality (VR) technology has been one of the most relevant trends in recent years, significantly impacting various fields such as medicine, education, and tourism. This project aims to explore the practical value of virtual reality technology in the context of interior design through the development of an interactive virtual environment.

The primary objective was to observe and understand the practical applications of virtual reality by creating a realistic and interactive non-immersive environment. The project sought to illustrate how virtual reality can be utilized in interior design and to provide a modest example that could inspire future developments in similar fields.

Results from the development process highlight the efficiency and feasibility of using virtual reality tools like Unreal Engine, showcasing the detailed realism achievable within a reasonable development timeframe. Through this project, substantial technical skills were acquired, and a deeper understanding of virtual reality capabilities in creating interactive experiences was achieved. Additionally, this work serves as a basic model that others can build upon to develop their own virtual reality projects.

### Resumen

En los últimos años, el desarrollo de la realidad virtual ha incrementado su accesibilidad y ha proporcionado experiencias más inmersivas a los usuarios en distintos ámbitos, como la medicina, la educación y el turismo. La realidad virtual está demostrando su valor práctico en diversos sectores, ofreciendo soluciones innovadoras y mejorando las prácticas existentes.

Bajo este contexto, el objetivo del trabajo ha sido observar y comprender las aplicaciones prácticas de la tecnología de realidad virtual, así como crear un entorno de realidad virtual no inmersivo, realista e interactivo que sirva de ejemplo o modelo para futuros desarrollos.

Los resultados obtenidos en este proyecto reflejan varios aspectos, como la viabilidad de desarrollar un entorno virtual interactivo de alta calidad utilizando herramientas de realidad virtual como Unreal Engine. Además, a través de este proyecto, se adquirieron habilidades técnicas significativas y se logró una comprensión más profunda de las capacidades de la realidad virtual para crear experiencias interactivas. Estos logros no solo subrayan el potencial de la realidad virtual en el ámbito del diseño de interiores, sino que también establecen una base aplicable que otros desarrolladores puedan utilizar para facilitar la creación de proyectos similares.

### Resum

En els darrers anys, el desenvolupament de la realitat virtual ha incrementat la seva accessibilitat i ha proporcionat experiències més immersives als usuaris en diferents àmbits, com ara la medicina, l'educació i el turisme. La realitat virtual està demostrant el seu valor pràctic a diversos sectors, oferint solucions innovadores i millorant les pràctiques existents.

Sota aquest context, l'objectiu del treball ha estat observar i comprendre les aplicacions pràctiques de la tecnologia de realitat virtual, així com crear un entorn de realitat virtual no immersiu, realista i interactiu que serveixi d'exemple o model per a desenvolupaments futurs.

Els resultats obtinguts en aquest projecte reflecteixen diversos aspectes, com ara la viabilitat de desenvolupar un entorn virtual interactiu d'alta qualitat utilitzant eines de realitat virtual com Unreal Engine. A més, mitjançant aquest projecte, es van adquirir habilitats tècniques significatives i es va aconseguir una comprensió més profunda de les capacitats de la realitat virtual per crear experiències interactives. Aquests èxits no només subratllen el potencial de la realitat virtual en l'àmbit del disseny d'interiors, sinó que també estableixen una base aplicable que altres desenvolupadors puguin fer servir per facilitar la creació de projectes similars.

### Agraïments

Vull expressar el meu més sincer agraïment a les persones que han fet possible aquest treball de fi de grau:

En primer lloc, a la meva tutora Mireia Ribera, per la seva dedicació, comprensió i paciència. La seva disposició a ajudar-me, les seves revisions, propostes, les converses, han estat de gran ajuda durant tot el procés.

També vull agrair a les persones dels fòrums d'internet que, tot i no conèixer-nos personalment, em van ajudar durant el desenvolupament responent les meves consultes, oferint consells i suggeriments.

Un agraïment especial a la meva família, pel seu suport constant, la seva confiança i l'acompanyament en els moments d'estrès i d'alegria.

Aquest treball no hauria estat possible sense la col·laboració, suport i inspiració de totes aquestes persones, a les quals estic profundament agraït.

Finalment, vull agrair-me a mi mateixa per la constància i l'esforç dedicats durant aquest treball.

# Índex

1	Intr	oducció	5	1					
	1.1	Context i motivació							
		1.1.1	Panorama general de la tecnologia de realitat virtual	1					
		1.1.2	Realitat virtual al sector de disseny d'interiors	1					
	1.2	Objecti	us	1					
		1.2.1	Objectius en la fase d'exploració	2					
		1.2.2	Objectius de desenvolupament	2					
	1.3	Planific	ació	2					
		1.3.1	Objectius inicials i planificació	2					
		1.3.2	Replanificació i noves etapes	3					
	1.4	Estruct	ura	4					
<b>2</b>	$\mathbf{Est}$	at de l'a	art de la tecnologia de realitat virtual	6					
	2.1	Concept	tes fonamentals de la realitat virtual	6					
	2.2	Estat i	aplicacions actuals de realitat virtual	7					
	2.3	Desenvo	olupament de software en realitat virtual - Motors de videojoc	7					
		2.3.1	Unity	8					
		2.3.2	Unreal Engine	9					
	2.4	Inicis a	Unreal Engine	9					
		2.4.1	Unreal Engine i Unreal Editor	10					
		2.4.2	Sistema de Blueprints	11					
		2.4.3	Programació en C++	11					
3	$\mathbf{Dis}$	seny i p	lanificació tècnica	13					
	3.1	Identific	cació d'actors	13					
	3.2	Casos d	l'ús	13					
	3.3	Disseny	conceptual i planificació d'escena	15					
		3.3.1	Interfície d'usuari	15					
	3.4	.4 Aprenentatge d'Unreal Engine							
		3.4.1	Level editor	19					
		3.4.2	Static mesh editor	22					
		3.4.3	Material editor	22					
		3.4.4	Blueprint editor	22					
	3.5	Desenvo	olupament inicial i problemes trobats	24					
4	$\mathbf{Des}$	envolup	pament del sistema	26					

<b>4</b>	Desenvolupament	$\mathbf{del}$	sistema
----------	-----------------	----------------	---------

	4.1	Constr	rucció d'escenaris	26
		4.1.1	Il·luminació	26
		4.1.2	Importar paquet d'assets	27
		4.1.3	Importar models 3D	27
		4.1.4	Configuració d'HDRI	30
	4.2	Imple	mentació d'interaccions amb elements de l'escena	31
		4.2.1	Detecció de col·lisions i missatges	32
		4.2.2	Execució de les interaccions	34
	4.3	Implei	mentació de modes de visualització i captura	36
		4.3.1	Mode de foto $\ldots$	38
		4.3.2	Canviar perspectiva de la càmera	39
	4.4	Implei	mentació de la navegació amb mini mapa	40
	4.5	Imple	mentació de contorns ressaltats en objectes interactuables $\ldots$ $\ldots$	41
	4.6	Expor	tació i instal·lació del projecte	42
	4.7	Manua	al de l'aplicació <i>kitchen</i>	43
5	Rev	isió ge	eneral	46
6	Anà	disi ec	onòmica i de costos	47
7	Con	clusio	ns i treball futur	49
	7.1	Vies d	e recerca futura	49
	7.2	Conclu	usions	49
$\mathbf{A}$	Ann	nex		<b>54</b>
	A.1	Glossa	ui	54
	A.2	Bluep	rints	57

#### 1 Introducció

#### 1.1 Context i motivació

#### 1.1.1 Panorama general de la tecnologia de realitat virtual

La realitat virtual emergeix avui dia no només com una novetat tecnològica, sinó com una porta oberta a mons que fins ara eren inimaginables. Com han assenyalat Burdea Grigore i Philippe Coiffet en Virtual Reality Technology [6], descriuen aquesta evolució com la transició cap a un diàleg interactiu entre el sistema i l'usuari, on la interacció esdevé central, de forma que els entorns virtuals reaccionen de manera instantània i proporcionen un nou canal de comunicació entre l'individu i l'entorn digital.

D'acord amb les investigacions actuals de Hamad, A., i Jia, B [25], aquesta tecnologia de realitat virtual ha demostrat un potencial d'aplicació ampli en sectors tan variats com l'enginyeria, l'educació, la medicina, l'entreteniment, etc. I s'espera que la realitat virtual marqui un gir radical a la tecnologia, transformant de forma significativa el nostre dia a dia.

#### 1.1.2 Realitat virtual al sector de disseny d'interiors

En explorar les eines aplicades en diferents sectors, un camp del qual també se'n parla molt és el camp del disseny d'interiors i el sector immobiliari. Tal com indica Zhu [57], destaca que la tecnologia de realitat virtual no només millora visualment els dissenys, sinó que també construeix un pont real entre dissenyadors i usuaris i enriqueix molt l'art de l'espai interior, a més, aquesta tecnologia fa que el disseny d'interiors sigui més humanitzat i el rendiment del disseny d'interiors sigui més ràpid, científic i en temps real.

Per exemple, existeixen avui en dia aplicacions que permeten entrenar i simular escenaris realistes de disseny i ensenyar dissenyadors o arquitectes a navegar espais virtuals [39], fa possible la col·laboració remota, en què els equips poden treballar junts en un mateix espai virtual per prendre decisions de disseny i coordinar projectes [31]. Fins i tot es pot aplicar en el procés de compra i venda de propietats, ja que, a través de models 3D detallats, permet als clients fer visites virtuals sense haver d'assistir físicament [51].

Tots aquests exemples serveixen de testimoni de l'aplicabilitat i l'impacte d'aquestes innovacions en situacions reals. Gràcies a aquests casos pràctics, es pot veure com s'està aplicant la tecnologia i com pot revolucionar indústries tradicionals en millorar l'eficiència i l'accés. Aquests exemples m'han inspirat a explorar més a fons la tecnologia de realitat virtual i considerar-ne les possibilitats en el camp del disseny d'interiors.

#### 1.2 Objectius

Sota el context de la recerca actual i basant-se en la motivació, l'objectiu del treball és fer servir els coneixements adquirits durant els anys d'estudis en informàtica per explorar i desenvolupar un sistema de disseny d'interiors fent servir la tecnologia de realitat virtual. Aquest projecte té com a finalitat principal **veure** el valor pràctic d'aquesta tecnologia i **crear** un exemple o model que pot ser utilitzat en futurs per desenvolupar-ne altres.

A continuació, es detallen els objectius més concrets que s'han creat des d'una vista més tècnica, els quals són els **mitjans** per assolir l'objectiu més ampli que s'acaba de definir:

#### 1.2.1 Objectius en la fase d'exploració

Per tal d'establir una base teòrica i pràctica per al desenvolupament posterior del sistema. Seran necessaris els següents punts:

- Estudi de tecnologies i plataformes de realitat virtual existents: Revisar les tecnologies i plataformes de realitat virtual disponibles al mercat, com Unity i Unreal Engine i determinar quina s'usarà per a la implementació.
- Aprenentatge tècnic: Aprofundir en l'ús de la tecnologia escollida. Aprendre la interfície, saber programar i fer ús de l'editor del motor de joc.

#### 1.2.2 Objectius de desenvolupament

Crear un sistema de visualització d'habitatges en realitat virtual fent servint l'editor, mostrant amb detall les característiques de l'habitatge i implementar les funcionalitats interactives necessàries per a una experiència completa. Per assolir l'objectiu, és necessari desenvolupar els següents punts:

- **Disseny del sistema:** Disseny de la navegació, la interfície i interaccions de l'usuari, que permeti als usuaris moure's a través de l'espai virtual, accedir fàcilment a diferents zones de l'habitatge i interactuar amb objectes de l'escena.
- **Construcció de l'escena:** Crear l'escena virtual, incloent-hi tots els elements necessaris per representar de manera realista l'habitatge.
- **Funcionalitats interactives:** Desenvolupar funcionalitats com el control de la il·luminació, l'operació de portes i el moviment d'objectes, permetent als usuaris no només observar l'escena creada, sinó també simular l'experiència d'utilitzar l'espai.

#### 1.3 Planificació

#### 1.3.1 Objectius inicials i planificació

A la fase inicial del projecte, es van plantejar diversos objectius per integrar intel·ligència artificial en una escena de realitat virtual creada a Unreal Engine. L'objectiu principal era desenvolupar un sistema que pogués reconèixer i classificar objectes, específicament verdures, dins d'una escena interactiva.

Per assolir aquest objectiu, es va establir una planificació detallada que distribueix el treball en un període de 15 setmanes, dividides en 8 etapes principals, les quals són:

- Etapa 1- Aprenentatge d'Unreal Engine: Familiaritzar-se amb l'entorn de desenvolupament d'Unreal Engine, incloent-hi la interfície i els fonaments dels Blueprints i programació en C++.
- Etapa 2- Construcció de l'escenari a Unreal Engine: Crear una escena senzilla utilitzant l'editor d'Unreal Engine.

- Etapa 3- Creació de la interfície d'interacció amb l'usuari: Desenvolupar una interfície que permeti a l'usuari interactuar, en concret, que permeti fer captures de pantalla en l'escena.
- Etapa 4- Selecció i entrenament del model d'intel·ligència artificial: Crear i entrenar un model senzill de classificació d'imatges.
- Etapa 5- Integració del model a Unreal Engine: Establir la comunicació entre el model i l'escena de realitat virtual, desenvolupant lògiques específiques per gestionar la interacció.
- Etapa 6- Millora i extensió de l'escena de realitat virtual: Realitzar millores i extensions de l'escena, afegint nous elements i funcionalitats per enriquir l'experiència de l'usuari.
- Etapa 7- Entrenament d'un model més complex: Millorar el model entrenantne un de més complex, combinant dades i models preentrenats per augmentar-ne la precisió i l'eficàcia.
- Etapa 8- Revisions finals: Revisar la interacció entre la IA i l'entorn de realitat virtual, ajustant qualsevol problema i revisar la documentació del projecte.

Mentre que el desenvolupament de la memòria del projecte es durà a terme de manera continuada al llarg de totes les etapes del projecte.

Durant l'Etapa 5- Integració del model d'intel·ligència artificial a Unreal Engine, es van trobar amb diversos obstacles que van dificultar la implementació dels objectius inicials. Aquests obstacles van obligar a redefinir i ajustar els objectius del projecte. A la secció 3.5. Desenvolupament inicial i problemes trobats, es detallaran els problemes trobats.

#### 1.3.2 Replanificació i noves etapes

Com que les primeres quatre etapes es van completar amb èxit, es va decidir aprofitar algunes d'elles i continuar des d'aquí amb noves tasques alineades als nous objectius. Començant des de l'etapa 3, es redefineixen:

- **Etapa 4- Redefinició de l'objectiu:** Explorar les possibilitats de desenvolupament del sistema de disseny d'interiors utilitzant la tecnologia de realitat virtual.
- Etapa 5- Extensió de l'escena de realitat virtual: Ampliar i millorar l'escena inicial creada a l'etapa 2.
- Etapa 6- Implementació de funcionalitats interactives: Desenvolupar funcionalitats que permetin a l'usuari interactuar de manera dinàmica amb l'entorn creat.
- **Etapa 7- Revisions finals:** Revisió del projecte desenvolupat, i ajustar la documentació del projecte.



Figura 1.3.1: Diagrama de Gantt de la planificació (Font: elaboració pròpia)

A la Figura 1.3.1 es mostra la planificació del projecte distribuïda en un període de 15 setmanes. La línia vermella discontínua indica el moment en què es va decidir replantejar els objectius del projecte. Les barres amb línies contínues representen les etapes que s'han desenvolupat i les marques de verificació dins d'aquestes barres indiquen si l'etapa va ser útil per al desenvolupament del projecte. Els tics ( $\checkmark$ ) indiquen que l'etapa es va aprofitar per al desenvolupament del projecte. Les barres amb les creus ( $\times$ ), com l'etapa 4 i l'etapa 5, indiquen que encara que es van desenvolupar, no es van aprofitar en el procés de desenvolupament del projecte. Les barres amb línies discontínues representen les etapes que no es van desenvolupar després de la revisió dels objectius. Aquesta planificació permet una clara visualització de les diferents fases del projecte i el seu progrés al llarg del temps. Aquesta figura proporciona un seguiment del progrés real en comparació amb la planificació inicial, que permet avaluar el rendiment, identificar les àrees de millora en la planificació i execució de projectes futurs.

#### 1.4 Estructura

Pel que fa a l'organització del contingut, la memòria està estructurada en set seccions principals, a més de les referències i l'annex. A continuació, es presenta una descripció detallada del contingut de cadascuna d'aquestes seccions.

Primer de tot, en la **secció 1** d'introducció, s'estableix el context i la motivació del projecte, presentant un panorama general de la tecnologia de realitat virtual i la seva aplicació al sector del disseny d'interiors. També es descriuen els objectius en la fase d'exploració i desenvolupament, la planificació inicial i els ajustaments realitzats durant el desenvolupament del projecte.

En la **secció 2**, s'introdueixen els conceptes fonamentals relacionats amb la realitat virtual i s'exploren les aplicacions d'aquesta tecnologia en diferents camps i sectors. Incloent la introducció dels motors de videojoc més rellevants actualment, com el Unity i Unreal Engine.

La **secció 3** està dedicada al disseny conceptual de l'aplicació, definint els actors i els casos d'ús per identificar les funcionalitats essencials. Es dissenya la interfície d'usuari i es planifiquen les interaccions dins de l'escena virtual. A més, es dedica una part significativa a l'aprenentatge d'Unreal Engine. També inclou una part en què s'explica

el que es va desenvolupar inicialment per a l'objectiu original del projecte, els problemes trobats durant aquest procés i els motius que van portar a la redefinició de l'objectiu.

A continuació, per la **secció 4**, es detalla el desenvolupament pràctic del sistema de realitat virtual. Explicant la construcció de l'escenari, i la implementació de les funcionalitats.

A la secció 5 es recapitula el treball fet, i s'avalua aquest treball al llarg del projecte.

A la **secció 6** es du a terme una anàlisi econòmica i de costos, detallant una estimació dels costos associats al desenvolupament del projecte.

Finalment, en la **secció 7** de conclusions i treball futur suggereix possibles direccions per a futures investigacions basades en els resultats del projecte i es resumeix els èxits obtinguts.

Per últim, s'hi inclouen les referències en l'apartat de **bibliografia**, que abasten totes les fonts bibliogràfiques i materials consultats durant el desenvolupament del projecte. I en l'**annex** es presenten materials addicionals que complementen el contingut principal, com el glossari i els blueprints.

### 2 Estat de l'art de la tecnologia de realitat virtual

Per tal de tenir una comprensió completa del potencial de la tecnologia de realitat virtual, a continuació es revisaran els conceptes bàsics de realitat virtual, les diferents classificacions i exemples d'aplicacions a diversos camps. Posteriorment, s'aprofundirà en les principals plataformes i motors d'aquesta tecnologia, analitzant les seves característiques, avantatges i aplicacions pràctiques a la indústria.

#### 2.1 Conceptes fonamentals de la realitat virtual

La realitat virtual (VR) és una tecnologia que simula un entorn virtual tridimensional generat per ordinador amb què els usuaris poden interactuar. Generalment, s'accedeix a aquest entorn virtual mitjançant un equip capaç de projectar informació 3D mitjançant una pantalla, que pot ser des de dispositius aïllats fins a cascos de realitat virtual (HMD, head-mounted display), juntament amb sensors d'identificació de l'usuari [47]. D'aquesta forma permet l'usuari explorar i manipular dins de l'escena proporcionada.



Figura 2.1.1: Cascos realitat virtual (Font: [49])

La tecnologia de realitat virtual es classifica principalment en dos tipus: immersiva i no immersiva. La **realitat virtual immersiva** és la que utilitza cascos HMD i sistemes de so multidimensional per proporcionar una experiència totalment immersiva que fa que l'usuari se senti completament submergit a l'entorn virtual. El nucli d'aquesta experiència és la immersió espacial segons com indica Charlene Jennett [28] —una percepció d'estar present físicament en un món no físic. Aquesta percepció es crea envoltant l'usuari amb imatges, sons i altres estímuls que constitueixen un entorn atractiu. La immersió espacial s'aconsegueix quan el món simulat és perceptivament convincent, és a dir que sembla "real" i "autèntic" i fa que l'usuari se senti, que realment "està allà".

Per l'altra banda, la **realitat virtual no immersiva** és la que generalment utilitza monitors tradicionals com a pantalla i dispositius d'entrada com teclats, ratolins, micròfons per interactuar amb el món virtual [48] [32]. Encara que aquest format no té la profunditat de la realitat virtual immersiva, ofereix una altra manera d'interactuar amb entorns virtuals i no requereix dispositius HMDs ni plataformes de computació d'alt rendiment, cosa que redueix significativament el cost, i la inversió tecnològica. A més, la facilitat d'ús és un altre punt a destacar, ja que els usuaris poden interactuar amb entorns virtuals sense necessitat de familiaritzar-se amb maquinari complex o especialitzat. D'altra banda, el fet de no submergir completament l'usuari en un entorn virtual, fa que hi hagi un risc menor de provocar desorientació o mareigs.

Explorant aquestes dues formes de realitat virtual, es pot comprendre millor el poten-

cial i les limitacions en diferents escenaris d'aplicació.

#### 2.2 Estat i aplicacions actuals de realitat virtual

En els darrers anys, el desenvolupament de la realitat virtual ha augmentat la seva accessibilitat i ha proporcionat experiències més profundes als usuaris, juntament amb els beneficis ja esmentats, la realitat virtual s'està assentant a diferents sectors demostrant el seu valor pràctic.

Com per exemple en l'àmbit mèdic, s'ha demostrat que la tecnologia de realitat virtual pot alleujar la sensació de dolor en els pacients. Els estudis mostren que durant tractaments de dolor utilitzant la realitat virtual, hi ha una disminució d'activitat a les regions del cervell associades amb el dolor, cosa que indica que la realitat virtual podria influir directament en la manera com el cervell processa els senyals de dolor [27]. A més, durant la pandèmia del COVID-19, la telemedicina s'ha utilitzat com un complement indispensable als serveis de salut, on la tecnologia de realitat virtual ha demostrat la seva contribució valuosa, especialment en el diagnòstic i tractament a distància [33].

No només en el camp de la medicina, el camp educatiu també ha estat transformat per les innovacions de la realitat virtual. S'ha trobat que la tecnologia de realitat virtual millora el rendiment acadèmic i l'interès dels estudiants en l'educació *STEM* (Science, Technology, Engineering and Mathematics) [3], especialment en la comprensió i la visualització de conceptes abstractes, proporcionant una eina d'ensenyament potent que facilita experiències d'aprenentatge més profundes i interactives [1] [26].

Pel que fa a la indústria turística, la realitat virtual ha impulsat el desenvolupament d'un nou model de turisme virtual (VT, Virtual Tourism), permetent als viatgers experimentar entorns digitals immersius basats en paisatges turístics reals abans, durant i després dels seus viatges [30]. I des de la perspectiva del consumidor, la realitat virtual ha permès que el turisme sigui accessible per a persones amb discapacitats físiques, restriccions econòmiques o temps [5].

A més d'aquests sectors, hi ha altres camps en què la realitat virtual també ha aconseguit un desenvolupament notable. Entre ells, l'entreteniment, on aquesta tecnologia ja és una experiència ben coneguda i àmpliament desenvolupada, i altres àrees com el màrqueting, l'enginyeria, el dissenv interior, entre d'altres.

#### 2.3 Desenvolupament de software en realitat virtual - Motors de videojoc

Després d'haver explorat una àmplia varietat d'exemples d'aplicacions de la realitat virtual, ens podem preguntar: Quines tecnologies fan possible aquestes implementacions impressionants?

Per al desenvolupament de la realitat virtual s'hi involucren diverses tecnologies, les quals es poden categoritzar-se en dos grans grups, que són el maquinari (hardware) i el programari (software). A l'àmbit del hardware, es troben elements com els cascos de realitat virtual, sistemes de seguiment, dispositius de sortida d'àudio i una varietat de controladors i perifèrics que proporcionen retroalimentació.

Per l'altra banda, el software també juga un paper molt important en el desenvolupament d'experiències de realitat virtual. Dins d'aquest espectre, els motors de videojoc són el nucli, els quals faciliten la integració d'eines de disseny artístic i software de disseny assistit per ordinador (CAD), a més proporcionen editors 3D potents que permeten als desenvolupadors construir i administrar entorns virtuals complexos [4].

Per començar a comprendre què és un motor de jocs, es poden definir com un programa o entorn de desenvolupament de software que inclou un conjunt d'eines de programari o interfície de programació d'aplicacions (*API*) dissenyades inicialment per desenvolupar videojocs. Actualment, aquests motors també s'utilitzen per crear aplicacions en diversos camps com la simulació, l'educació o la realitat virtual, oferint funcionalitats de renderització 2D i 3D [42] [54].

No obstant això, per obtenir una comprensió més profunda de l'arquitectura i la capacitat dels motors de jocs, segons la definició de Lewis, M., & Jacobson, J són plataformes de software per al desenvolupament de videojocs i aplicacions de realitat virtual, compostes per múltiples mòduls [29] els quals inclou:

- Gestió d'entrada i sortida: en què gestiona totes les operacions d'entrada i de sortida, incloent-hi la renderització en 3D, el dibuix 2D i el so, per tal d'assegurar la interacció fluida de l'usuari amb el món.
- **Simulació física i dinàmica**: ofereix s imulacions físiques bàsiques, com per exemple la detecció de col·lisions.
- Motor de renderitzat: és el nucli perquè és el responsable de renderitzar la perspectiva del jugador a partir de models 3D.
- Suport de xarxa: aquesta funcionalitat permet a múltiples usuaris interactuar en temps real dins del mateix entorn virtual, assegurant la sincronització entre les accions dels usuaris i l'estat de l'entorn.
- **Entorn de desenvolupament**: proporciona un entorn de desenvolupament que permet als desenvolupadors i creadors fer servir les eines que proporciona per dissenyar elements i escenes.
- **Controladors gràfics i interacció amb servidors**: els controladors gràfics tradueixen les sol·licituds de renderització a operacions específiques de la llibreria gràfica, mentre que el servidor administra la informació global del joc, assegurant la coherència del món del joc entre tots els jugadors.

#### 2.3.1 Unity

Quan parlem de motors de videojocs, és comú que el primer que se'ns acudeixi sigui Unity. Aquest motor, conegut també com a Unity 3D, va ser desenvolupat per Unity Technologies i publicat el 2005. Ha estat considerat com un dels millors motors de videojocs per l'accessibilitat que ofereix als desenvolupadors [35], segons la pàgina oficial de Unity, es consideren com a plataforma líder mundial per a la creació i operació de contingut interactiu en 3D en temps real (RT3D).

La plataforma expandeix el seu abast a més de 20 plataformes, i cobreix milers de milions de dispositius en l'àmbit mundial. Aquesta àmplia distribució és possible gràcies a la seva arquitectura i suport per al llenguatge de programació C#. A més, Unity a part de centrar-se en l'entreteniment també ofereix eines avançades que són essencials

per transformar dades CAD i 3D en aplicacions i experiències immersives [52]. Aquestes capacitats són valuoses en sectors com l'arquitectura, l'enginyeria i la construcció, on Unity facilita la integració de dades disperses i simplifica les complexitats operatives.

Pel que fa a les llicències, Unity proporciona diverses opcions adaptades a diferents necessitats [53]. La més accessible és *Unity Personal*, una versió gratuïta destinada a desenvolupadors individuals, petites empreses o startups que no superen als 100.000 dòlars d'ingressos anuals. Tot i que aquesta versió és completament funcional, presenta algunes restriccions quant a personalització i assistència tècnica. En canvi, *Unity Pro* està dissenyada per a empreses o desenvolupadors individuals els quals els seus ingressos anuals excedeixen els 100.000 dòlars. Aquesta versió no només elimina les limitacions de la versió personal, sinó que també enriqueix l'experiència de l'usuari amb característiques avançades, proporcionant un suport tècnic més complet i més capacitats de personalització i col·laboració.



Figura 2.3.1: Visual Studio Code amb Unity. (Font: [34])

#### 2.3.2 Unreal Engine

Un altre dels motors de videojocs més populars és Unreal Engine, desenvolupat per Epic Games i publicada per primer cop el 1998.

Unreal Engine es destaca per diverses característiques clau que contribueixen a la seva àmplia adopció a diferents àmbits. Com per exemple, el seu sistema de Blueprints, permet als desenvolupadors implementar lògiques de joc complexes sense escriure codi. A més, és reconegut pels seus gràfics d'alta gamma, cosa que el fa ideal per desenvolupar projectes que busquen construir un impacte visual. També ofereix un rendiment superior comparat amb altres motors, i gràcies al seu suport especialitzat i de la disponibilitat de certes eines avançades [41] és unes de les eleccions principals per desenvolupar aplicacions de realitat virtual.

A diferència d'Unity, que ofereix diferents llicències basades en el nivell d'ingressos del desenvolupador, Unreal Engine permet que qualsevol persona pugui descarregar i fer servir Unreal Engine sense cost inicial. I comença a cobrar una comissió del 5% sobre els ingressos un cop arriba o supera el llindar de \$1,000,000 en ingressos bruts del producte desenvolupat.

#### 2.4 Inicis a Unreal Engine

Per aquest projecte es farà servir Unreal Engine. Optar per un motor diferent del que s'usa comunament, com Unreal Engine en lloc d'Unity, proporciona una oportunitat per explorar noves possibilitats i provar alternatives al camp de la realitat virtual. L'assignatura de *Gràfics i Visualització de Dades* proporciona conceptes clau que estan relacionats amb l'ús d'Unreal Engine. Al curs, es van estudiar temes com l'aplicació de materials, els principis bàsics de la tecnologia de renderització i els fonaments computacionals darrere d'aquests processos.

Tot i que Unreal Engine no requereix implementar algorismes de renderitzat des de zero ni construir la infraestructura bàsica de models 3D, els coneixements obtinguts en gràfics faciliten la comprensió de la seva lògica subjacent. Per exemple, per utilitzar l'editor de *materials* d'Unreal Engine i crear efectes visuals, és fonamental conèixer les opcions d'ombreig i les configuracions de textures, conceptes que es van abordar a l'assignatura.



Figura 2.4.1: Jerarquia de programari gràfic (Font: [45])

A partir de la Figura 2.4.1, es pot veure com s'interconnecten els components del software gràfic, des d'aplicacions d'alt nivell fins al hardware de baix nivell. Unreal Engine es troba a nivell de motors gràfics, cosa que indica que està dissenyat per treballar en un nivell més alt d'abstracció, integrant i simplificant l'ús de llibreries més tècniques i complexes.

La formació en gràfics proporciona familiaritat amb moltes de les llibreries i principis que Unreal Engine utilitza de manera simplificada. Per exemple, en treballar amb *shaders* GLSL, s'interactua directament amb el hardware gràfic, cosa que permet comprendre les operacions de baix nivell que Unreal Engine gestiona internament. Això facilita reconèixer i entendre els conceptes que Unreal simplifica a través de la seva interfície i eines.

Tal com s'acaba de mencionar, Unreal Engine es distingeix pel seu sistema de Blueprints i el seu suport per a C++. Però abans d'entrar en detalls sobre com s'utilitzen Blueprints i C++, cal saber la diferència entre Unreal Engine i Unreal Editor.

#### 2.4.1 Unreal Engine i Unreal Editor

Unreal Engine és el motor que proporciona la base tecnològica per al desenvolupament de jocs i aplicacions interactives. És un *framework* de software que inclou un conjunt d'eines de desenvolupament i llibreries que permeten la simulació de físiques, renderització de gràfics, configuració de so, intel·ligència artificial, etc. És el nucli que s'encarrega de l'execució, gestiona la lògica, les dades i la presentació visual.

En canvi, l'Unreal Editor, és la interfície gràfica d'usuari que els desenvolupadors

utilitzen per interactuar amb Unreal Engine. És un entorn de desenvolupament integrat (IDE) que proporciona eines visuals als usuaris per dissenyar les escenes, els nivells, i els permeten configurar objectes, crear animacions, establir interaccions, etc.

Explicat de forma més senzilla, Unreal Engine és el motor darrere de tot, el "nucli" que fa funcionar els jocs i les aplicacions, i l'Unreal Editor és la "interfície" amb què s'interactua per construir-los.

#### 2.4.2 Sistema de Blueprints

El Blueprint Visual Scripting system d'Unreal Engine és un entorn de programació visual dissenyat per facilitar la creació de lògiques de joc i comportaments interactius sense la necessitat d'escriure codi [14], ja que es basa en l'ús d'una interfície nodal per crear elements de joc directament des de l'Unreal Editor, on cada node representa una funció o un valor, i els desenvolupadors poden connectar aquests nodes per construir scripts complexos.

En concret, cada node té entrades i sortides, que es connecten mitjançant enllaços. Tal com es mostra a la Figura 2.4.2 Aquests enllaços defineixen el flux d'informació i la seqüència d'execució entre nodes. Aquests fluxos o seqüències poden ser activats per accions dins del joc, com per exemple la d'iniciar un nivell, pressionar un botó, o col·lisionar amb un objecte, etc. D'aquesta forma en detectar alguna de les accions es pot invocar alguna funció que executi una tasca específica.



Figura 2.4.2: Editor de blueprint (Font: elaboració pròpia)

#### 2.4.3 Programació en C++

Encara que el sistema de Blueprints és poderós per si mateix, Unreal Engine també ofereix l'opció de programar amb codi C++. És a dir, ofereix la possibilitat als desenvolupadors de començar primer amb Blueprints per a construir prototips ràpids i després passar a C++ per a casos més complexos o que requereixin major optimització. Ambdós mètodes es poden utilitzar de manera independent o en conjunt, depenent de les necessitats del projecte i de les habilitats de l'equip de desenvolupament.



Figura 2.4.3: Codi C++ en Unreal Engine. (Font: [38])

Per programar en C++ amb Unreal Engine, generalment es requereix un IDE compatible. El que se sol utilitzar per Windows és el Microsoft Visual Studio per la seva profunda integració amb Unreal Engine, i per als usuaris de macOS, XCode és l'IDE recomanat [17].

Per poder seleccionar aquesta opció, Unreal Editor demanarà triar entre Blueprint i C++ en crear un nou projecte. Si s'escull C++, Unreal Engine configura un projecte de Visual Studio automàticament amb els fitxers de codi font necessaris.

Pel que fa a la programació en C++ dins d'Unreal Engine, comparteix moltes similituds amb la programació C++ estàndard, però també presenta algunes característiques úniques, diferències i particularitats. Per exemple l'ús de les *macros* per definir propietats, classes, funcions i altres elements que interactuen directament amb el sistema d'Unreal. Com poden ser les macros UCLASS(), UPROPERTY(), UFUNCTION()[20] són utilitzades sovint per declarar classes d'Unreal, propietats i mètodes que s'exposen a l'editor i a Blueprints. A més, Unreal Engine proporciona un sistema anomenat *Garbage Collection* en què els objectes que hereten d'UObject són automàticament rastrejats i netejats pel motor, reduint els riscos de fuites de memòria [22]. En resum, encara que la base del llenguatge és la mateixa, la programació en C++ per a Unreal Engine incorpora diverses capes addicionals de complexitat i funcionalitat molt útils que no es veuen en programació en C++ estàndard.

### 3 Disseny i planificació tècnica

Basant-se en les aplicacions pràctiques de la realitat virtual explorades anteriorment, el projecte que es desenvoluparà és una **aplicació de realitat virtual no immersiva** a l'àmbit del disseny interior, dissenyada per simular l'experiència de visites virtuals d'habitatges. Aquesta aplicació busca resoldre les limitacions de les visites físiques, millorant-ne l'eficiència i l'accessibilitat, com s'havia explicat en l'apartat 1.1.2. Realitat virtual al sector de disseny d'interiors. Per tant, les funcionalitats estan dissenyades per permetre als usuaris explorar de manera interactiva una escena virtual, interactuant amb diferents elements de l'entorn.

#### 3.1 Identificació d'actors

Usuari final: El principal actor és l'usuari que interactua amb l'escena de realitat virtual.

**Sistema/Motor**: El motor (o sistema) es pot considerar un actor secundari que respon a les accions de l'usuari.

#### 3.2 Casos d'ús

A continuació es descriuen una sèrie de casos d'ús que defineixen com aquestes funcionalitats permeten als usuaris interactuar amb l'entorn:

USE CASE 1 Ob		r/tancar una porta	
Objectiu	Permetre a l'usuari obrir o tancar una porta a l'escena de realitat virtual.		
Precondicions	L'usu	ari és a prop d'una porta i la porta està oberta/tancada.	
Actors	Usua	Usuari, Sistema	
Trigger	L'usu	uari s'acosta a la porta.	
Descripció	Pas	Acció	
	1	L'usuari s'acosta a la porta.	
	2	El sistema detecta la proximitat i mostra l'opció d'obrir o tancar porta.	
	3	L'usuari pressiona el botó per obrir o tancar la porta.	
4 El sistema executa l'animació d'obrir o tancar la porta.		El sistema executa l'animació d'obrir o tancar la porta.	

Taula 3.2.1: Taula UC1 (Font: elaboració pròpia)

USE CASE 2 Encendre/apagar llum		
Objectiu	Perm	etre a l'usuari encendre o apagar una llum a l'escena de realitat virtual.
Precondicions	L'usu	ari és dins del rang d'interacció de l'interruptor.
Actors	Usua	ri, Sistema
Trigger	L'usu	ari s'acosta a l'interruptor de llum.
Descripció	Pas	Acció
	1	L'usuari s'acosta a l'interruptor.
	2	El sistema detecta la proximitat i mostra l'opció per encendre o apagar la
		llum.
	3	L'usuari prem el botó per activar l'interruptor.
	4	El sistema encén o apaga la llum.

Taula 3.2.2: Taula UC2 (Font: elaboració pròpia)

USE CASE 3	Moure un objecte		
Objectiu	Perm	etre a l'usuari moure un objecte a l'escena de realitat virtual.	
Precondicions	L'usu	ari és dins del rang d'interacció de l'objecte.	
Actors	Usua	Usuari, Sistema	
Trigger	L'usu	L'usuari s'acosta a aquest objecte.	
Descripció	Pas	Acció	
	1	L'usuari s'acosta a aquest objecte.	
	2	El sistema mostra l'opció d'agafar o deixar l'objecte.	
	3	L'usuari prem el botó per agafar o deixar l'objecte.	
4 L'usuari mou l'objecte al lloc que vol.			

Taula 3.2.3: Taula UC3 (Font: elaboració pròpia)

USE CASE 4 Encendre/apagar televisió			
Objectiu Permetre que l'usuari controli l'estat de la televisió, encenent-la o apagant			
Precondicions	L'usu	ari és a prop de la televisió i la televisió està encesa/apagada.	
Actors	Usua	ri, Sistema	
Trigger	L'usu	uari s'acosta a la televisió.	
Descripció	Pas	Acció	
	1	L'usuari s'acosta a la televisió.	
	2	El sistema mostra l'opció d'encendre o apagar la televisió.	
	3	L'usuari prem el botó per encendre o apagar la televisió.	
4 El motor del joc canvia l'estat de la televisió.			

Taula 3.2.4: Taula UC4 (Font: elaboració pròpia)

Totes aquestes funcionalitats permeten a l'usuari interactuar directament amb els objectes de l'entorn i cada cas implica una acció binària com poden ser obrir/tancar, apagar/encendre, etc.

A continuació, també s'inclouran funcionalitats que ofereixen a l'usuari un control més gran sobre la perspectiva des de la qual veu i interactua amb l'entorn, permetent alternar entre vistes i capturar moments destacats:

USE CASE 5 En		ntrar a Mode Foto		
Objectiu	Perm	Permetre a l'usuari entrar en mode de foto per prendre captures de l'escena.		
Precondicions	L'usu	iari és a l'escena i el mode foto no està activat.		
Actors	Usua	ri, Sistema		
Trigger	L'usu	ari prem el botó per canviar al mode foto.		
Descripció Pas Acció		Acció		
1 L'usuari prem el botó per activar el mode foto.		L'usuari prem el botó per activar el mode foto.		
	2	El sistema separa la càmera del personatge i permet a l'usuari moure la		
		càmera lliurement.		
	3	L'usuari ajusta l'angle i la posició per fer la foto.		
4		L'usuari clica la pantalla per fer la captura.		
	5 El sistema captura la imatge i ofereix opcions per desar o tancar.			

Taula 3.2.5: Taula UC5 (Font: elaboració pròpia)

USE CASE 6 Canv		iar perspectiva de la càmera	
Objectiu	Perm	etre a l'usuari alternar la vista de la càmera entre primera persona i tercera	
	persona.		
Precondicions	L'usu	L'usuari pressiona el botó per canviar la perspectiva de la càmera.	
Actors Usua		ari, Sistema	
Trigger	L'usu	ari és a l'escena.	
Descripció Pas A		Acció	
1 L'usuari prem el botó per canviar la perspectiva.		L'usuari prem el botó per canviar la perspectiva.	
	2 El sistema alterna entre la vista de primera i tercera persona.		

Taula 3.2.6: Taula UC6 (Font: elaboració pròpia)

Respecte a la navegació, el següent cas d'ús mostra la funcionalitat que ajuda l'usuari a orientar-se a l'entorn virtual i moure's més fàcilment:

USE CASE 7		Veure posició en el mini mapa		
Objectiu	Perm	Permetre a l'usuari veure la seva posició en un petit mapa dins de la interfície.		
Precondicions	L'usu	ari és a l'escena.		
Actors	Usua	Usuari, Sistema		
Trigger	L'usu	L'usuari és a l'escena.		
Descripció	Pas	Acció		
	1	L'usuari es desplaça dins de l'escena.		
	2	El sistema mostra la posició actual de l'usuari al minimapa.		
	3	A mesura que l'usuari es mou, el marcador de posició s'actualitza al mini-		
		mapa.		

Taula 3.2.7: Taula UC7 (Font: elaboració pròpia)

#### 3.3 Disseny conceptual i planificació d'escena

Com que la creació d'un entorn 3D des de zero requereix una quantitat significativa de temps i recursos, i no és l'enfocament del projecte, s'ha optat per fer servir un paquet predefinit que ja conté els elements necessaris, com els models de 3D, les *textures* i els materials, ja que disposar d'aquests recursos pot agilitzar el procés de desenvolupament. Tot i que el paquet pot proporcionar la base, aquest permet ser personalitzat i ampliat per satisfer els objectius específics del projecte.

Per a la selecció del paquet, la decisió s'ha basat en els casos d'ús definits anteriorment, que descriuen les funcionalitats a implementar en el projecte. Per tant, es buscarà preferentment un paquet que contingui elements com portes, llums, objectes i una televisió, ja que aquestes són els elements amb què s'espera que l'usuari pugui interactuar. Tot i que es podria crear una escena només amb aquests elements, es buscarà un paquet que contingui un entorn més complet perquè l'experiència de la visita virtual sigui tan realista com sigui possible. Si manca algun dels elements necessaris, es buscaran en recursos externs els models i materials adequats per afegir-los.

Encara que no són estrictament necessaris, es realitzaran ajustaments a les textures, materials, i configuració de llums i entorns per tal d'aprendre la interfície i la implementació tècnica d'Unreal Engine.

#### 3.3.1 Interfície d'usuari

Encara que l'estructura base de l'escena està definida pel paquet predefinit, sí que és important dissenyar i determinar com es veu i s'organitza la interfície del projecte. En primer lloc, hi ha d'haver una interfície que funcioni com a manual d'usuari per als usuaris que encara no coneguin les operacions disponibles de l'aplicació. Aquesta interfície no es mostrarà automàticament a l'inici del joc per no obligar els usuaris que ja coneixen les regles a veure-les. En canvi, sempre hi haurà una interfície visible com es mostra en la Figura 3.3.1, que indica que en pressionar la tecla [M] es pot accedir al menú, on allà estaran les instruccions.

SCENE	
Menu	

Figura 3.3.1: Interfície per accedir al menú (Font: elaboració pròpia)

En accedir a aquest menú, el joc es pausa i els usuaris poden optar per sortir de l'aplicació o continuar-hi. A més, aquest menú mostrarà tots els controls de l'aplicació, com el de canviar la vista del joc prement la tecla [V] o entrar en mode foto amb la tecla [H], també es pot utilitzar el ratolí per canviar l'angle de visió, entre d'altres. És una guia completa de totes les accions disponibles al joc.



Figura 3.3.2: Interfície menú (Font: elaboració pròpia)

La tercera interfície es mostra només en situacions específiques quan l'usuari s'acosta a un objecte interactiu dins de l'aplicació. En aquests casos, apareixeran petites targetes a la pantalla amb indicacions.



Figura 3.3.3: Interfície d'interaccions (Font: elaboració pròpia)

Finalment, la quarta interfície està dissenyada per mostrar un mini mapa que estarà present en tot moment durant el joc. Dins d'aquest mini mapa s'inclourà també un indicador que mostra la ubicació actual del personatge. A diferència de la segona i tercera interfície, aquest mini mapa no ha de ser activat per una acció específica de l'usuari. Igual que la primera interfície, el mini mapa estarà sempre visible.

SCENE	

Figura 3.3.4: Interfície mini mapa (Font: elaboració pròpia)

Per tant, a la pantalla principal de l'aplicació, sempre hi haurà dos elements visibles: el mini mapa i l'indicador de menú. Quan l'usuari premi la tecla [M], el joc farà pausa i es mostrarà la interfície del menú. En aquesta interfície de menú, l'usuari podrà veure tots els controls del joc, i també podrà triar continuar amb el joc o sortir de l'aplicació. I durant el joc, les targetes informatives apareixeran i desapareixeran segons la proximitat de l'usuari als objectes interactius.

#### 3.4 Aprenentatge d'Unreal Engine

Per començar a treballar amb Unreal Engine, a la seva pàgina web oficial s'indiquen algunes configuracions mínimes recomanades [16]. A continuació, es resumeixen els requisits específics per al sistema operatiu Windows, que és amb el que es treballarà:

 Pel sistema operatiu, en cas de ser Windows, ha de ser com a mínim Windows 10, versió 1703 (Creators Update) o superior.

- Cal **DirectX End-User** *Runtimes* (juny de 2010), que proporciona un conjunt de llibreries necessàries per a la renderització gràfica, la qual cosa permet que el sistema operatiu executi i mostri correctament gràfics en 3D, efectes especials i altres elements visuals d'alta qualitat.
- Es recomana un **processador** de quatre nuclis, sigui d'Intel o AMD, amb una velocitat mínima de 2,5 GHz.
- **Memòria RAM**: És recomanable disposar d'almenys 8 GB per a un rendiment òptim en el desenvolupament de projectes i l'execució de l'aplicació.
- Pel que fa a la **targeta gràfica** ha de ser compatible amb DirectX 11 o DirectX 12, juntament amb els controladors més recents.
- En cas de desenvolupar amb C++, l'**IDE** ha de ser Visual Studio 2022.

Assegurant que es compleixen tots els requisits demanats per poder treballar amb Unreal Engine, toca crear un projecte.



Figura 3.4.1: Navegador de projectes d'Unreal Engine (Font: elaboració pròpia)

En iniciar Unreal Engine, s'obre automàticament el navegador de projectes com es mostra en la Figura 3.4.1, des d'on es poden fer les accions com la de crear un nou projecte, obrir un projecte existent, gestionar els projectes, o crear-ne un de nou.

Per crear un nou projecte, cal tenir en compte els següents passos: la selecció de la categoria de desenvolupament (1), l'elecció de la plantilla (2), la configuració d'alguns paràmetres inicials del projecte (3), i finalment, assegurar-se de configurar correctament el lloc on es vol emmagatzemar el projecte i el nom que se li vol donar (4).

En aquest cas, es va seleccionar la plantilla *Third Person Character* en la categoria de *Games* per començar el desenvolupament de l'aplicació. Pel que fa als paràmetres de configuració inicial del projecte, s'han deixat la majoria dels camps per defecte escollint el projecte de tipus *Blueprint*.

La interfície d'Unreal Engine és el primer que es veu quan s'obre el projecte. Unreal Engine emfatitza la importància de comprendre bé la interfície d'usuari perquè molts dels seus elements, com els panells i les barres de menú, s'utilitzen repetidament en diferents parts del motor i durant diverses etapes del desenvolupament [19]. Llavors saber com funcionen els panells, barres d'eines i altres components de la interfície ajudarà a simplificar les tasques repetitives i accelerar el procés de desenvolupament.

En aquesta secció, es destacaran els elements de la interfície que van ser més rellevants durant el desenvolupament del projecte, incloses aquelles parts de la interfície que es van fer servir amb més freqüència i van tenir més importància en el flux de treball.

#### 3.4.1 Level editor

Un cop obert el projecte, generalment s'inicia l'**editor de nivells (Level Editor)**,que forma part de la interfície. Els nivells (levels) a Unreal Engine són bàsicament diferents mons o entorns dins del motor que contenen geometries (*mesh*), objectes, actors, il·luminació, àudio i altres configuracions que formen l'entorn [21]. Cada nivell és una unitat independent d'organització que es pot carregar, descarregar i manipular en temps d'execució o a l'editor. I l'editor de nivells és la part de la interfície on es construeix i s'organitza el món col·locant aquests objectes, llums, personatges i altres elements. Aquest editor mostrat en la Figura 3.4.2, està format per diversos components i panells, com el Viewport (1), que és per on es veu i es manipula el nivell, el navegador de continguts (Content Browser) (2) per accedir als recursos, l'Outliner (3) per veure els actors del nivell, i el panell de detalls (Details Panel) (4) per modificar les propietats dels actors seleccionats.



Figura 3.4.2: Interfície d'usuari - Level editor (Font: elaboració pròpia)

El Level Viewport és la finestra o àrea de la interfície a l'editor de nivells on per-

met veure i manipular visualment el nivell en 3D. És l'espai principal on es treballa per dissenyar l'entorn, ja que ofereix funcionalitats com canviar la perspectiva de la càmera per obtenir diferents angles del nivell, permet seleccionar, moure, girar i escalar actors o objectes, proporciona controls i eines per ajustar les posicions i propietats dels actors, es pot iniciar el mode de simulació o de joc per veure com es veu i com es comporta el nivell des de la perspectiva del jugador, etc. En resum, el Level Viewport és on es pot veure el nivell que s'està dissenyant i treballant.



Figura 3.4.3: Level editor - Level viewport (Font: elaboració pròpia)

El **Content Browser** és l'eina que actua com un navegador o explorador de tots els recursos que té el projecte. Serveix per organitzar, crear, accedir i importar els diferents recursos necessaris per al desenvolupament, que poden ser per exemple, models 3D, materials, sons i altres components visuals o auditius utilitzats per construir els nivells. Llavors el Content Browser és el que s'encarrega d'emmagatzemar tots aquests recursos del projecte, que podran ser organitzats en carpetes, facilitant la classificació i l'accés a cada recurs.



Figura 3.4.4: Level editor - Content Browser (Font: elaboració pròpia)

L'**Outliner** a Unreal Engine és un panell que proporciona una vista jeràrquica de tots els actors (objectes) que estan presents al nivell actual. Per exemple, permet veure les relacions entre pares i fills entre els actors, permet seleccionar actors directament des de la llista de l'Outliner, cosa que és útil quan certs actors estan ocults o difícils de trobar al Level Viewport. En aquest panell és possible agrupar actors en carpetes per poder-los manipular de forma conjunta, i també ofereix altres funcionalitats com el de, en seleccionar un actor a l'Outliner, poder duplicar, eliminar o moure-ho, i veure i modificar les seves propietats al Details Panel (Panel de Detalls).



Figura 3.4.5: Level editor - Outliner (Font: elaboració pròpia)

El **Details Panel** és un panell que mostra i permet editar les propietats i configuracions dels actors, components i altres objectes seleccionats. És una eina essencial per personalitzar i ajustar les característiques de forma precisa cada element al nivell.



Figura 3.4.6: Level editor - Details Panel (Font: elaboració pròpia)

A més de l'editor de nivell, Unreal Engine té altres editors que formen part de la interfície i que són importants per al desenvolupament.

#### 3.4.2 Static mesh editor

Per exemple, l'editor de malles estàtiques (Static Mesh Editor) no es troba directament a la pantalla inicial quan es crea el projecte, sinó que s'accedeix a través del Content Browser. I és una eina especialitzada dins del motor que es fa servir per visualitzar, modificar i optimitzar les propietats de les malles estàtiques [18]. A través del Viewport d'aquest editor, es mostra la malla en 3D, permetent rotar-la i fer zoom per veure'n els detalls. Mentre en el panel de detalls es mostra les propietats de la mesh seleccionada.

#### 3.4.3 Material editor

L'editor de material (Material Editor) també és un editor important, ja que és l'eina principal per crear i personalitzar materials, que defineixen com es veuen les superfícies dels objectes al joc. Aquest editor ofereix un panell de graf (Graph Panel) [9], que és la secció principal on es creen els materials mitjançant un sistema de Blueprints, i amb això es permet definir l'estructura del material, com ara colors, textures, efectes especials i ombreig.

#### 3.4.4 Blueprint editor

Per al desenvolupament d'aquest projecte, el sistema de Blueprint ha estat fonamental, i per poder treballar amb Blueprint, ha estat imprescindible utilitzar l'**editor de Blueprint (Blueprint Editor)**, que forma part de la interfície d'Unreal Engine. Aquesta eina facilita la creació de lògica de joc mitjançant nodes visuals com s'ha mencionat anteriorment, permetent implementar comportaments i esdeveniments. El Blueprint Editor pot aparèixer en diferents formes dins Unreal Engine, adaptantse a les necessitats específiques de cada aplicació [23]. L'editor de classes de Blueprints (Blueprint Class Editor) és una de les variacions del Blueprint Editor i està dissenyat específicament per editar classes de Blueprints. Aquestes classes defineixen comportaments per als actors en el joc, proporcionant una estructura que pot ser heretada per altres blueprints [10]. I aquesta és una de les interfícies que més es fan servir durant el desenvolupament d'un projecte tipus Blueprint, composta principalment per les seccions:

Observant la Figura 3.4.7, en el panel de components (Components Panell) (1) és on es mostren tots els components que formen part de la classe Blueprint actual, des de models 3D, càmeres, llums, fins a col·lisions o scripts addicionals [15]. En el panell de blueprint (Blueprint Panell) (2), es llisten tots els elements que formen part del Blueprint, incloses les variables definides, algunes funcions, macros, esdeveniments, gràfics, etc [13].



Figura 3.4.7: Blueprint class editor - Components i Blueprint Panel (Font: elaboració pròpia)

Pel que fa a l'editor de Graf (Graph Editor) (3), és l'àrea principal on es creen les xarxes de nodes visuals que defineixen la lògica del Blueprint, i aquí és on es dissenya visualment la lògica que controla el comportament de l'actor o de la classe [12]. I per últim, disposa també d'un panell de detalls (Details Panel) (4) [11] semblant al panell de detalls de l'editor de nivell, que serveix per mostrar les propietats del node, component o actor seleccionat i poder ajustar els seus valors o qualsevol altra propietat configurable.

File Edit Asset View Debug Window Tools Help				Pare	– CI X Int class: <u>Static Mesh Actor</u>
🖀 🏚 💥 Comple : 📲 Diff 🗸 🖗 Find 🐁 Hole Unrelated : 🔅 Class Settings 📈 Class Debudits ⊳ Simulation 🕟 🌗 📗 🛋 : 🗈 No debug object selected 🗸 👘					
El Components ×	III Viewport J Construction Scr	Event Graph × () Timeline_Templ		🖊 Details	×
+ Add Q Search	¶ ✓ ← → 🏗 SM_Door2_Blue		Zoom 1:1	Q Search	
( <b>Ø 34). Dove2. Duegent (Sel)</b> ⊗ State Mesh Component (StateMeshComponentti) <u>Edit in C++</u>	Caston Furt	() Timile	∲ Set Actor Rotation Larget in Actor		1.0 1.0 1.0 State State More SM_Do ~ © \$5 \$5
A My Blueprint ×		Delay Update D	D	₩ Materials	
+ Add Q Search		D Play from Start Finished D	• Target self Return Value •	Element 0	
w duuwis ⊘ * \$: EventGraph ♦ Operation	Close_door Custom Event	Stop Direction Reverse New Track 0 Reverse From End	New Rotation X (Roll) 0.0 New Rotation Y (Pitch) 0.0	Element 1	M_Blinds_wo G    B    B M_Chrome
		D Set New Time	New Rotation Z (Yaw)	Liement 2	ິ 🛯 🦉 🖳
T ConstructionScript		> New Time 0.0	• Teleport Physics	Advanced	
MACROS				🗢 Actor Tick	
variables				Start with Tick	2
- Components				Allow Tick Befo	
				Advanced	
	3			Component Tick Start of ek Tigenter (s Advanced	<b>0</b> 0
			BLUEPRINT	Physics Simulate Physics Mass (kg)	
	🔀 Compiler Results 🗙			Linear Damping	0.01
				Angular Damping Enable Gravity	ao •
TG Content Drawer S Output Log IS Omd ✓ Enter Console Command				🕄 1 Unsaved	P Revision Control 🗸

Figura 3.4.8: Blueprint class editor - Graph Editor i Details Panel (Font: elaboració pròpia)

En resum, és necessari entendre bé la interfície abans de començar a desenvolupar un projecte, ja que moltes de les eines necessàries per desenvolupar estan organitzades a diferents editors, panells i menús, i familiaritzar-se amb la interfície facilita l'accés als recursos, i simplifica la navegació dins de l'editor en general.

#### 3.5 Desenvolupament inicial i problemes trobats

Durant les primeres etapes del projecte, l'objectiu inicial va ser integrar un model de classificació de verdures dins d'una escena de realitat virtual creada amb Unreal Engine. A continuació, es detallen els passos seguits i els problemes que s'han trobat durant aquest procés, que eventualment van portar a la redefinició dels objectius del projecte.

Es va començar amb la construcció d'una escena en un projecte anomenat *kitchen* en Unreal Engine. Aquest procés va involucrar la creació de l'entorn des de zero, utilitzant l'enfocament de *disseny modular* que consisteix en, fer servir elements com les parets i els mobles, o altres mòduls prefabricats per a construir l'escena [43]. I posteriorment, s'afegeixen detalls com electrodomèstics i decoració per completar-la. Entre aquests detalls es van incloure les verdures, ja que serveixen per a la posterior classificació de verdures utilitzant el model de classificació.

Per permetre a l'usuari interactuar amb l'escena i fer captures de la verdura en la pantalla, es va desenvolupar la funcionalitat de poder prendre captures de pantalla de l'escena, les quals serien usades posteriorment com a entrades per al model de classificació de verdures. Aquesta funcionalitat es mantindrà pel projecte actual, permetent a l'usuari entrar al mode foto i capturar imatges de l'escena en qualsevol moment.

El pas següent va ser l'entrenament d'un model de classificació de verdures utilitzant PyTorch. Es va fer servir un model preentrenat, específicament el ResNet-18, per aprofitar les característiques ja apreses del conjunt de dades ImageNet. I es va configurar les darreres capes per adaptar-lo a la tasca específica, en concret, es va reemplaçar la capa final, l'optimitzador i la funció de pèrdua (loss function).

La integració del model de classificació a l'escena d'Unreal Engine utilitzant C++, va ser l'etapa que presentava el desafiament més gran. En aquesta etapa es va explorar l'ús d'ONNX (Open Neural Network Exchange), que és un format obert per representar models d'aprenentatge automàtic que permet la interoperabilitat entre diferents frameworks, i s'ha fet servir per convertir el model de PyTorch, ja que era el compatible amb el connector (plugin) NNE (Neural Network Engine) d'Unreal Engine [38]. On NNE és un framework que facilita l'execució de models de xarxes neuronals dins l'entorn del joc.

El flux de treball suggerit pel desenvolupador de NNE [50] incloïa els següents passos:

- Habilitar els **plugins** necessaris a Unreal Engine tant el **NNE** mateix com els **runtimes**.
- Importar el model en format .onnx a l'editor.
- Obtenir el **runtime**, carregar el model i crear-ne la instància.
- Obtenir la imatge a la **CPU** i alimentar-la al model per obtenir les classificacions.

Tot i seguir els passos esmentats, es van trobar múltiples problemes durant la integració del model. Per exemple, un dels problemes va ser la incompatibilitat en la detecció de les llibreries proporcionades pel plugin NNE a l'hora de crear una instància del model, i per la manca de documentació i recursos disponibles va fer que la resolució de problemes fos lenta i complicada. Per a la resta de problemes trobats, tot i que es van publicar consultes en el fòrum oficial d'Unreal Engine [55], les respostes van ser limitades i no sempre resolutives. A causa de l'acumulació d'aquests problemes tècnics i considerant el temps restant del projecte, es va decidir redefinir els objectius i les planificacions per alinear-se millor amb els recursos i el temps disponibles.

### 4 Desenvolupament del sistema

#### 4.1 Construcció d'escenaris

Com es va esmentar anteriorment a la secció de disseny, a la construcció de l'escena s'han utilitzat recursos de diverses fonts per assegurar la creació d'un entorn realista. Entre aquests recursos s'hi inclouen models 3D importats externament, recursos (assets) de *Quixel Bridge*, i un paquet d'assets disponible al *Marketplace* d'Unreal Engine.

#### 4.1.1 Il·luminació

El primer que s'ha configurat és la il·luminació per assolir un ambient més realista. Es van configurar diversos components per millorar tant la il·luminació natural com l'ambiental. Van incloure:



Figura 4.1.1: Escena amb il·luminació configurada (Font: elaboració pròpia)

- **Sky Light** és el component que captura la il·luminació ambiental de l'entorn basada en el cel configurat o en un entorn HDRI (high dynamic range images).
- Directional Light, que actua com una simulació del sol.
- Sky Atmosphere, que simula la dispersió atmosfèrica afectant la llum de l'entorn.
- **Exponential Height Fog**, que afegeix profunditat mitjançant boira que incrementa amb l'altitud.
- **Post Process Volume**, que ajusta l'exposició general i aplica efectes visuals globals, funcionant com un filtre que afecta l'aparença final de l'escena.
- **Volumetric Clouds**, per afegir núvols tridimensionals que interactuen amb la il·luminació principal, contribuint a la profunditat i realisme del cel.

Inicialment, la il·luminació de l'escena es configura segons les especificacions detallades anteriorment. Un cop s'integri un HDRI a l'escena, s'anticipa que alguns d'aquests components podrien tornar-se innecessaris.

#### 4.1.2 Importar paquet d'assets

S'ha seleccionat el paquet Archviz Interior vol.3 [37] del Marketplace d'Unreal Engine com a base per a la construcció de l'escena, ja que inclou tots els elements necessaris per al projecte, permetent així centrar els esforços en la personalització i l'adaptació de l'espai en lloc de crear cada component des de zero.

Per poder importar assets des del Marketplace d'Unreal Engine un cop són adquirits i descarregats, aquests es troben a la biblioteca dins d'Unreal, i es poden afegir al Content Browser del projecte.

Aquest paquet ofereix dos fitxers de nivell *.umap*, un que conté una escena preconstruïda dissenyats pels creadors i un altre fitxer que es tracta d'un entorn d'exhibició on cadascun dels assets inclosos al paquet es mostra de manera individual o en grups petits.



Figura 4.1.2: Overview.umap (Font: [37])

Per aprofitar l'alta qualitat i assegurar que el projecte tingui un aspecte atractiu sense necessitat de passar hores ajustant l'estètica o la il·luminació bàsica s'ha escollit fer servir el nivell *interior.umap* com a punt de partida per a un desenvolupament més detallat o personalitzat.

#### 4.1.3 Importar models 3D

Per Importar models 3D a Unreal Engine, es pot fer de dues maneres principals, o fent servint Quixel Bridge per accedir a *Quixel Megascans*, una llibreria de models 3D, o important models d'altres fonts externes.

En base de l'escena del nivell interior.umap, es poden afegir models 3D externs en

formats *.fbx.* Específicament, alguns models han estat seleccionats de plataformes reconegudes per la seva àmplia oferta de recursos en 3D, que són d'accés gratuït com el *Free3D* [24] i *CGTrader* [7], que han estat les fonts principals d'aquests models.

Un cop descarregat el fitxer que conté el model, aquest hauria d'incloure la geometria el mesh, les textures i potser els materials associats.

En aquest cas en concret, el fitxer inclou només el mesh i les textures, però sense un material predefinit. Per aquests casos, s'han de crear i configurar el material manualment. Els passos són:

- 1. Importació del Mesh i les textures: S'arrossega i es deixa anar el fitxer .fbx dins del Content Browser. El mateix es fa amb les textures, que són fitxers .png o .jpg que acompanyen el mesh.
- 2. Creació del Material: Un cop importades les textures i el mesh, es crea un material des del Content Browser, i s'obre l'editor de materials fent-hi doble clic al material que s'acaba de crear.
- 3. Configuració del material: Normalment, un material es compon de diverses textures que s'assignen a diferents propietats per definir l'aspecte visual dels objectes 3D. Dins de l'editor de materials, les textures importades s'arrosseguen des del Content Browser al panell del graf de materials, on apareixen en nodes anomenats *Texture Samples*. Cadascuna d'aquestes textures es configura per aportar detalls específics al material. Per exemple, el color base defineix el color de la superfície del material; la textura de reflectivitat controla com la llum es reflecteix a la superfície, influint en la seva aparença brillant o mate; i la textura de rugositat ajusta l'aspra de la superfície. Aquestes textures es connecten arrossegant la sortida RGB a les entrades corresponents del material, com ara '*Normal*' a l'entrada de 'Normal', '*Roughness*' a 'Roughness', i així successivament. En la Figura 4.1.3 es mostra un exemple de configuració de material.



Figura 4.1.3: Material editor (Font: elaboració pròpia)

4. Aplicació del material al Mesh: Un cop connectat els nodes, es desa el material i es tanca l'editor de materials. Després, per aplicar el material a un mesh específic de l'escena, se selecciona el mesh en el viewport o al World Outliner, i a continuació, s'arrossega el material creat que està en el Content Browser fins al slot de material corresponent al panell de detalls del mesh seleccionat, aconseguint així que el material s'apliqui correctament al mesh.

Els models importats s'organitzen en carpetes específiques dins del Content Browser d'Unreal Engine. En aquest projecte, els models importats s'han guardat en una carpeta anomenada *Kitchen Assets*.



Figura 4.1.4: Carpeta Kitchen Assets (Font: elaboració pròpia)

Per importar models 3D utilitzant Quixel Bridge. Aquest procés és molt més fàcil, gràcies a la connexió nativa entre Quixel Bridge i Unreal Engine, que permet una exportació directa i fluida de models d'alta qualitat amb materials *PBR* (Physically Based Rendering) preconfigurats. Això s'aconsegueix en seleccionar els assets desitjats a Quixel Bridge, descarregar-los i un cop descarregats, exportar-los directament a Unreal Engine mitjançant el botó d'exportació. Mentre al Content Browser d'Unreal Engine, es crea una carpeta anomenada *Megascans* on des d'allà, els models són preparats per ser utilitzats a l'escena, i no es requereixen passos addicionals, com l'ajustament de materials o configuració.


Figura 4.1.5: Carpeta Megascans (Font: elaboració pròpia)

## 4.1.4 Configuració d'HDRI

Un cop configurada la il·luminació de l'escena i havent importat tots els models 3D necessaris, es pot procedir amb la configuració d'HDRI. Atès que l'escena inclou finestres a través de les quals es pot veure l'exterior, configurar un HDRI podria proporcionar un fons realista, millorant així la qualitat visual de l'entorn exterior.



Figura 4.1.6: Vista de l'escena (Font: elaboració pròpia)

High Dynamic Range Imaging (HDRI) inclou una varietat de tècniques per capturar, emmagatzemar, i mostrar imatges o vídeos amb un rang més ampli de luminàncies, estès de valors entre el negre i el blanc, [46] permetent així representar escenes reals proporcionant un realisme visual millorat i amb més qualitat d'imatge. Aquests fitxers es poden obtenir de diversos llocs web, com per exemple, l'HDRI que s'ha fet servir en el projecte s'ha obtingut des de *Poly Haven*[44], que és una llibreria pública que ofereix assets 3D d'alta qualitat, inclosos HDRIs disponibles en formats com *.hdr* i *.exr*, que són fàcilment integrables amb Unreal Engine.

Per configurar un HDRI a Unreal Engine, a part de descarregar el fitxer HDRI en format .hdr des de Poly Haven i importar-lo dins del Content Browser, per la part d'Unreal Editor, ha sigut necessari activar el plugin *HDRI Backdrop* des del menú de plugins. Ja que introdueix un actor (actor de fons HDRI) que fan servir imatges HDRI com a fons esfèrics que envolten tota l'escena.

A continuació, s'afegeix aquest actor HDRI Backdrop a l'escena i se li assigna el fitxer HDRI importat. Finalment, s'ajusten l'escala, posició i altres paràmetres d'il·luminació del HDRI Backdrop per optimitzar la il·luminació i aconseguir l'efecte desitjat.



Figura 4.1.7: Vista de l'escena amb HDRI configurat (Font: elaboració pròpia)

Després d'aplicar l'HDRI, es pot mantenir alguns elements d'il·luminació com el *directional light*, com que simula la llum solar directa, ajuda a millorar el realisme i la qualitat de les ombres; el *sky light* també és útil per capturar la il·luminació ambiental i els reflexos de l'HDRI; com que l'HDRI porta núvols, mantenir els núvols volumètrics poden afegir profunditat i realisme; i fins i tot es podria mantenir tot, ja que l'HDRI proporciona una il·luminació de matí amb llum solar similar a la configuració inicial. L'escena amb HDRI configurada queda com el que es mostra en la Figura 4.1.7.

### 4.2 Implementació d'interaccions amb elements de l'escena

Per implementar totes les funcionalitats descrites, es va optar per començar primer amb la implementació de les interaccions amb els objectes, per tal de garantir que els fonaments del sistema estiguin ben establerts abans d'afegir funcionalitats més complexes. Aquestes funcionalitats corresponen als casos d'ús UC1 Obrir/tancar una porta, UC2 Encendre/apagar llum, UC3 Moure un objecte i UC4 Encendre/apagar televisió.

Els objectes seleccionats per ser interactuables són, la televisió, la porta, la llum de

peu (floor lamp), un interruptor de llum, i una olla (pot). Cadascun d'aquests objectes ha de tenir les seves classes de blueprints perquè en algun moment hauran de ser manipulats o interactuats de manera específica durant l'execució.



Figura 4.2.1: Classes blueprints dels objectes interactuables (Font: elaboració pròpia)

#### 4.2.1 Detecció de col·lisions i missatges

Per controlar i determinar amb quin objecte el jugador està interactuant, es pot fer servir la tècnica de **Line Trace.** El Line Trace és una tècnica que serveix per detectar col·lisions mitjançant una línia imaginària projectada des d'un punt d'origen (per exemple, podria ser des de la càmera del jugador) fins a un altre punt específic. Això permet identificar quins objectes es troben al camí d'aquesta línia, és a dir, amb quins l'usuari està mirant o apuntant, i per tant, amb quins està "col·lisionant".

Després de detectar col·lisió amb l'objecte, és important proporcionar al jugador una indicació visual de l'acció que pot prendre. Tal com es va dissenyar en apartats anteriors, es mostra una targeta a la pantalla en la qual se li mostra un missatge indicant la tecla o el botó que ha de pressionar per fer la interacció.



Figura 4.2.2: Disseny del ui\_screen (Font: elaboració pròpia)

Per aconseguir això és necessari crear un Widget blueprint on es dissenya la interfície d'usuari que es mostrarà. Per aquest projecte, s'ha creat un Widget Blueprint anomenat  $ui\_screen$ , i s'han afegit els elements  $TV\_Tip$ ,  $light\_Tip$ ,  $door\_Tip$ ,  $pot\_Tip$ , que són imatges importades com a textures, on s'hi mostren els missatges com "Press F to open/close door", i així per a cada tipus d'objecte.

A part de mostrar el missatge en la pantalla després de detectar col·lisió, també es configura una variable anomenada Interaction. Aquesta variable serveix per identificar el tipus d'interacció que es vol realitzar. Quan el jugador prem la tecla d'interacció (en aquest cas, la tecla [F]), el sistema utilitza el valor de la variable Interaction per determinar quina acció específica s'ha d'executar. Cada valor de la variable Interaction està associat a un tipus d'interacció diferent:

Interaction  $\theta$  - Significa que cal invocar l'esdeveniment per encendre o apagar la televisió.

Interaction 1 - Indica que cal invocar l'esdeveniment per agafar o deixar anar l'olla.

Interaction 2 - Correspon a l'esdeveniment d'encendre o apagar el llum de peu.

Interaction 3 - Representa l'esdeveniment d'obrir o tancar la porta.

Interaction 4 - Indica l'esdeveniment per engegar o apagar l'interruptor de llum.

Cada valor de la variable Interaction serveix per decidir quina acció específica s'ha d'executar quan el jugador pressiona la tecla [F]. On dues d'elles, Interaction 2 i Interaction 4, estan destinades a encendre i apagar llums, però corresponen a dos tipus de llums diferents.

Per tant, el flux de detecció de col·lisions mitjançant Line Trace es pot descriure de la següent manera:

- **Inici del Line Trace**: Es projecta una línia des de la càmera del jugador cap endavant.
- **Col·lisió amb objectes**: Si la línia interseca amb algun objecte, es genera un *Hit Result* que conté detalls sobre la col·lisió.
- **Processament del Hit Result**: El Hit Result s'utilitza per identificar el tipus d'objecte impactat mitjançant nodes *Cast To* als Blueprints, que són per convertir una referència genèrica d'un objecte a un tipus específic d'objecte, d'aquesta manera permet accedir a les propietats i funcions específiques de l'objecte.
- Maneig de la col·lisió: Depenent del tipus d'objecte detectat, es realitzen accions específiques, com el de mostrar el missatge a la pantalla i configurar la variable Interaction.

A continuació, es presenta un pseudocodi que il·lustra la lògica de detecció de col·lisions fent servint Line Trace, on *Event Tick* indica que la lògica s'executarà per a cada *frame*:

Event Tick
cameraLocation $\leftarrow$ Get the player's current camera location
forwardVector $\leftarrow$ Get forward direction from player's camera
// Define the end point of the Line Trace
traceEnd $\leftarrow$ cameraLocation + (forwardVector * 1000)
// Make a Line Trace
$hitResult \leftarrow Line Trace (start = cameraLocation, end = traceEnd, traceChannel = visibility)$
if hitResult has Actor then
if Cast (hitResult.Actor) to SM_TV_Blueprint then
show TV tip
setInteraction $\leftarrow 0$
else
hide TV tip
if Cast (hitResult.Actor) to SM_Pot_Blueprint then
show Pot tip
setInteraction $\leftarrow 1$
else
hide Pot tip
if Cast (hitResult.Actor) to SM_FloorLamp_Blueprint then
show light tip
setInteraction $\leftarrow 2$
else
hide light tip
if Cast (httResult.Actor) to SM_Door_Blueprint then
show Door tip
setInteraction $\leftarrow 3$
else
hide Door tip
if Cast (httesult.Actor) to SM_Tumblr_1_Blueprint then
show light tip
setInteraction $\leftarrow 4$
else
hide light tip

Algorithm 4.2.1 Interaction detection and managements

4.2.1. Pseudocodi detecció de col·lisions (Font: elaboració pròpia)

La implementació d'aquesta lògica es durà a terme fent servint Blueprints en Unreal Engine. A causa de l'extensió i la complexitat dels Blueprints, les imatges detallades s'adjuntaran a l'annex de la memòria. Totes aquestes operacions s'implementen al Blueprint del personatge, ja que el personatge és l'element central que interactua amb l'entorn.

#### 4.2.2 Execució de les interaccions

Després d'implementar el mòdul de detecció de col·lisions, el pas següent és definir i desenvolupar les accions específiques que es desencadenen **quan l'usuari interactuï** seguint les indicacions del missatge.

S'ha dissenyat que la tecla [F] és la que permet interactuar amb l'objecte per a cada interacció. Per tant, un cop l'usuari prem [F], es comprova el valor de la variable Interaction, per tant, entre totes les interaccions tenim:

### Interaction 0,1,2

Quan l'usuari pressiona la tecla [F] i la variable Interaction està configurada a 0,1 o 2, s'activa la lògica mitjançant el node Flip-Flop, que permet canviar a l'estat B si el node Flip-Flop és a l'estat A i viceversa. Per simular l'encesa i apagat de la televisió, es canvia el material de la pantalla de la televisió; per moure un objecte simplement lliga l'olla a la càmera del jugador, és a dir, l'olla es mourà juntament amb la càmera del jugador, donant així la sensació que l'usuari l'està agafant. I per deixar anar l'olla es fa tot el contrari; mentre per la llum de peu, el que es fa és activar i desactivar la visibilitat de la llum segons l'estat.

Algorithm 4.2.2 Interaction 0: Toggle TV material			
<pre>if F pressed and Interaction == 0 then     // Toggle between two states using Flip-Flop     state ← FlipFlop()     if state == A then         SetMaterial(TV, StaticMeshComponent, M_TV_Display)     else         SetMaterial(TV, StaticMeshComponent, M_HiResScreen4)</pre>			

4.2.2. Pseudocodi Interacció 0 (Font: elaboració pròpia)

On  $M_TV_Display$  és una foto de color negre que simula l'efecte apagat, mentre  $M_HiResScreen4$  que pot ser qualsevol altra imatge per simular l'efecte d'encesa.

Algorithm 4.2.3 Interaction 1: Attach and detach pot				
if F pressed and Interaction $== 1$ then				
state $\leftarrow$ FlipFlop()				
$\mathbf{if} \text{ state} == \mathbf{A} \mathbf{then}$				
attach the Pot to the player's camera				
SetSimulatePhysics(Pot, False)				
else				
Detach the Pot from the player's camera				
SetSimulatePhysics(Pot, True)				
4.2.3. Pseudocodi Interacció 1 (Font: elaboració pròpia)				

Algorithm 4.2.4 Interaction 2: Toggle lamp spot light	
if F pressed and Interaction $== 2$ then	

state ← FlipFlop()
if state == A then
 SetVisibility(SpotLight, False)
else
 SetVisibility(SpotLight, True)

4.2.4. Pseudocodi Interacció 2 (Font: elaboració pròpia)

### Interaction 3, 4

Per a les interaccions 3 i 4, s'invoquen funcions que estan definides als blueprints dels objectes corresponents. Per exemple, per obrir i tancar la porta, s'invoquen els esdeveniments *Open\_door* i *Close\_door*, que estan definits al Blueprint de l'objecte porta, i en aquestes funcions s'inicien una línia de temps (*timeline*) que reprodueix l'animació d'obrir i tancar la porta segons l'eix Z (Yaw), que pot ser reproduït tant en mode normal com en mode revers. De manera similar, per encendre i apagar els llums, s'invoquen esdeveniments definits al blueprint de l'interruptor. Aquests esdeveniments configuren la visibilitat d'una llista de llums obtinguda de l'escena a l'inici del joc.

Algorithm 4.2.5 Interaction 3: Open and close Door	
if F pressed and Interaction $== 3$ then	
state $\leftarrow$ FlipFlop()	
$\mathbf{if} \text{ state} == \mathbf{A} \mathbf{then}$	
Invoke Event (Open_door)	$\triangleright$ Defined in the Door Blueprint
else	
Invoke Event (Close_door)	▷ Defined in the Door Blueprint

4.2.5. Pseudocodi Interacció 3 (Font: elaboració pròpia)

Event Open_door	
Play Timeline	$\triangleright$ Starts the door opening animation
Event Close_door	
Reverse Timeline	▷ Starts the door closing animation
Timeline Update	
Set Actor Rotation (self, New Rotation Z (Yaw))	

4.2.6. Pseudocodi del blueprint de porta (Font: elaboració pròpia)

Algorithm 4.2.7 Interaction 4: Open and close light list	
if F pressed and Interaction == 4 then state $\leftarrow$ FlipFlop() if state == A then	
Execute Event (Open_Light) else	$\triangleright$ Defined in the Light switch blueprint
Execute Event (Close_Light)	$\triangleright$ Defined in the Light switch blueprint

4.2.7. Pseudocodi Interacció 4 (Font: elaboració pròpia)

Algorithm 4.2.8 Light switch blueprint logic		
Event BeginPlay		
lightList $\leftarrow$ Get All Actors of Class (Tracklight)		
Store lightList in Light List array		
Event Open_Light	$\triangleright$ Open the light list	
for each element in Light List $\mathbf{do}$		
Set Visibility (element, True)		
Event Close_Light	$\triangleright$ Close the light list	
for each element in Light List do		
Set Visibility (element, False)		

4.2.8. Pseudocodi del blueprint d'interruptor de llum (Font: elaboració pròpia)

### 4.3 Implementació de modes de visualització i captura

A part de les funcionalitats de poder interactuar amb l'escena, també s'han definit casos d'ús que permeten a l'usuari canviar la manera com visualitza o captura l'escena, que són respectivament el *UC5 Entrar a Mode Foto* i el *UC6 Canviar perspectiva de la càmera*. Entrar en mode foto implica un canvi temporal a la funcionalitat de l'aplicació, permetent capturar imatges, mentre que canviar la perspectiva de la càmera altera la manera com l'usuari veu l'escena, sense interactuar directament amb els objectes.

Seguint amb el que es va proposar en l'apartat de disseny de la interfície d'usuari, s'ha implementat en la pantalla principal una indicació a l'usuari, de com accedir a la interfície de menú. Que es fa prement el teclat [M].



Figura 4.3.1: Disseny del ui\_hint (Font: elaboració pròpia)

Llavors un cop l'usuari hagi premut [M], el sistema invoca a l'esdeveniment que crea el widget de *ui\_menu*. Que és la interfície en la qual l'usuari pot veure tots els controls del joc, decidir si continuar jugant (Resume) i sortir de l'aplicació (Quit). És aquí on l'usuari podrà veure les instruccions com el de per poder canviar la perspectiva de la càmera o entrar en mode foto, que són, respectivament, prémer [V] per canviar la perspectiva de la càmera, i prémer [H] per entrar a mode foto i un cop és al mode foto, l'usuari pot fer fotos prement el botó esquerre del ratolí.

File Edit Asset V	ew Debug Window Tools Help *		– Parent class:	⊡ × <u>User Widget</u>
💾 👩 😥 Compile 🗄	- 🖞 Diff 🗸 🕨 📄 📥 📜 No debug object selected 🗸 👔 🖉 Widget Reflector	<b>- </b>	esigner	🕄 Graph
🕫 Palette 🗴 🛅 Library	<u>6 195 205 305 405 405 405 705 805 1050 1050 1050 1050 1050 1050 105</u>	🗶 Details		
× image	- 200 pc 1: 1	n bgColor		🗸 Is Varia
COMMON		Q Search		日命
	MENU		Anchors	~ ~
	michto		0,0	
			0,0	
E Hierar x E Biod Wi	Resume		0.0	- <u>• •</u>
O Search Widgets		Offset Bottom		
▼ [ui_menu]		P Augmment Size To Content		
	<u>Quit</u>	ZOrder	0	
■ bgColor ■ ©		w Appearance		
T [Text] "2. Press F to."			Bind	v 6
<ul> <li>✓ [] [Vertical Box]</li> <li>✓ ■ ResumeBtn</li> <li>✓ ● Ø</li> </ul>	1. Use WASD keys to move around the scene.		Nore C	
T[Hesume Ixt] "Hes∎" ∞ T ⊟ QuitBtn 🖬 ∞	2. Press F to interact with the outlined object when close enough.			•
T[QuitTxt]"Quit" மீ® ⊫ ABtn மீ®	3. Press H to enter photo mode. And then click the left mouse button to take a photo. Press H again to exit and return to normal gameplay.	Image size	555 m	and a second second
⊤≣SBtn ∎ੈ® ∏TISTxtI'S" ∎ੈ®			losana	
v ≣WBtn ∎ ® T[WTvt] W ∎ ®			No Tile	÷ I
v ■ DBtn 🖬 🔊			Bind	
T[Text] "MENU"	Change to photo mode			
T [Text] "Move" • • •				
T [Text] "Change viewir 🖬 👁				
T [Text] "Change came 🖬 👁	Move Change viewing Jump Interact Change camera view			
T [Text] "Click to take." • •	angle Click to take screensnot			
T[SpaceTxt] "SPACE	In Device Content Scale 1, 0	w Behavior		
⊨ BHBtn ∎ ⊕	1 1 20 x 720 (16:9) DPI Scale 0.6 9		Bind	
-> = FBtn = * ∞ = [FTxt] "F" = • •	Arimations X		Bind	
n*image_132	+ Animation Q. Search & A.	Visibility	U Sind	
		w Render Transform		
-				
🕼 Content Drawer 🖀 Animation	: 🔽 Outputlog 🔀 Cmd 🗸 Enter Console Command	📸 1 Unsave	d 🗜 Revisi	

Figura 4.3.2: Disseny del *ui\_menu* (Font: elaboració pròpia)

### 4.3.1 Mode de foto

La funcionalitat de mode de foto permet als usuaris capturar imatges de l'escena en temps real.

El procés s'inicia amb la configuració d'una acció d'entrada al projecte. Utilitzant el sistema *Enhanced Input*, s'ha establert una acció anomenada *IA.PhotoMode*, la qual s'activa mitjançant la tecla [H]. Aquest enllaç permet a l'usuari activar o desactivar el mode foto en qualsevol moment, transicionant a una càmera específica que ofereix una vista per a la captura d'imatges. Aquest canvi es fa mitjançant la generació dinàmica d'una instància de càmera en el joc, utilitzant el node *SpawnActor* i escollint com a classe la *BP\_PhotoCamera*. La nova càmera es posiciona al lloc i amb l'orientació desitjada, capturant així la perspectiva que l'usuari vol fotografiar i a més transfereix el control del joc al nou actor càmera. I per desactivar el mode foto, l'esdeveniment IA\_PhotoMode es controla des del blueprint de la càmera de fotos BP\_PhotoCamera. Quan es pressiona [H] de nou, s'obté la instància del personatge en tercera persona, es transfereix el control del jugador de tornada a aquest personatge i es destrueix la càmera de fotos.

Algorithm 4.3.1 Toggle photo mode	
<b>Event</b> IA_PhotoMode followCamera $\leftarrow$ Get World Transform (Follow Camera) photoCamera $\leftarrow$ Spawn Actor (BP_PhotoCamera) at followCa playerController $\leftarrow$ Get Player Controller (Player Index 0) Possess (playerController, photoCamera)	▷ Triggered by pressing H in normal mode
<b>Event</b> IA_PhotoMode thirdPersonCharacter $\leftarrow$ Get Actor of Class (BP_ThirdPerson) playerController $\leftarrow$ Get Player Controller (Player Index 0) Possess (playerController, thirdPersonCharacter)	▷ Triggered by pressing H again in photo mode )
Destroy Actor (self)	$\triangleright$ Destroy the photo camera

4.3.1. Pseudocodi per activar i desactivar mode foto (Font: elaboració pròpia)

Per capturar la imatge, s'ha implementat un component de captura d'escena (Scene Capture Component) que renderitza la vista de la càmera en un *Render Target*. Quan se li dona al clic amb el ratolí, activa aquest procés que captura l'escena. Un cop obtinguda la imatge, es procedeix a assignar aquesta textura capturada a un material dinàmic. Per aconseguir-ho, es configura un esdeveniment *PreConstruct* que obté el material dinàmic d'un component d'imatge, al qual se li assigna la textura de la captura mitjançant el node *Set Texture Parameter Value*.



Figura 4.3.3: Blueprint de l'assignació dinàmica del material (Font: elaboració pròpia)

Això permet visualitzar la imatge dins de la interfície del mode foto.

Aquesta interfície d'usuari està composta per un component d'imatge que mostra la foto i dos botons interactius. Aquests botons permeten a l'usuari desar la imatge capturada al sistema de fitxers o sortir del mode foto. El botó de desar utilitza la funcionalitat d'escriptura al disc per persistir la imatge en un format estàndard. En un projecte d'Unreal Engine empaquetat, el directori habitual on es guarda la imatge és diferent del que s'utilitza a l'Editor. Normalment, s'utilitza un directori específic dins la carpeta de l'usuari. Això depèn del sistema operatiu, però generalment es troba a la següent ruta:

Windows: C:\Users\[NombreUsuari]\AppData\Local\[NombreDelJoc]\Saved\

Mentre que el botó de tancar simplement desactiva la interfície d'usuari de la manera foto, permetent a l'usuari tornar al mode joc.



Figura 4.3.4: Interfície d'usuari del mode de foto (Font: elaboració pròpia)

#### 4.3.2 Canviar perspectiva de la càmera

Pel que fa al canvi de la perspectiva de la càmera, aquest s'activa mitjançant la tecla [V], i amb el node Flip-Flop, alterna entre les dues perspectives.

erson
erson

4.3.1. Pseudocodi canvi de perspectiva de càmera (Font: elaboració pròpia)

Per poder configurar la càmera per a la vista en primera persona. El que es fa primer és lligar la càmera del personatge (Follow Camera) al component Mesh del personatge en un socket anomenat *head*, és a dir, posiciona la càmera al cap del personatge, proporcionant una vista des de la seva perspectiva. A més, es configuren diversos paràmetres per ajustar el comportament del personatge, assegurant que la rotació i el moviment siguin adequats per a la vista en primera persona.

Per configurar la perspectiva de la càmera en tercera persona, la càmera del personatge (Follow Camera) s'adjunta al component *Camera Boom* del personatge, que actua com un braç que posiciona la càmera darrere del personatge, proporcionant una vista més àmplia del seu entorn. Igual que a la vista en primera persona, es configuren els paràmetres necessaris per ajustar el comportament del personatge, assegurant que la càmera segueixi el personatge i que l'orientació del moviment sigui la correcta.

#### 4.4 Implementació de la navegació amb mini mapa

La navegació amb mini mapa al joc permet al jugador visualitzar la seva posició en temps real. Aquesta funcionalitat correspon al cas d'ús *UC7 Veure posició al mini mapa*, permetent així al jugador navegar i orientar-se més fàcilment dins de l'entorn.

Per muntar la interfície del mini mapa, s'ha creat un widget anomenat *ui\_smallmap*, en el qual s'ha afegit un *scale box*, i dins d'aquest component s'ha afegit una imatge que representa el mini mapa, i una marca del jugador (smallmark).



Figura 4.4.1: Disseny de ui\_smallmap (Font: elaboració pròpia)

El procés comença amb la inicialització del jugador. Utilitzant l'esdeveniment *Event Construct*, s'obté la referència del jugador i s'emmagatzema en una variable anomenada Player, que servirà per accedir a les propietats del jugador durant l'execució.

Per cada frame del joc, s'activa l'esdeveniment Event Tick, que desencadena la lògica d'actualització tant de la ubicació actual del jugador com del *Smallmark* al mini mapa. Per actualitzar la ubicació, es fan servir les coordenades del jugador obtingudes mitjançant el node *Get Actor Location*, on aquestes coordenades es transformen a través d'uns càlculs específics per poder mapejar-les correctament al sistema de coordenades del mini mapa. A més de la ubicació, també s'actualitza la rotació del Smallmark, que es fa a través del node *Get World Rotation*.

Tot aquest procés d'actualització es fa utilitzant el node *Set Render Transform*, com es mostra en la Figura 4.4.2, que s'encarrega d'aplicar les noves transformacions de posició i rotació del Smallmark al mini mapa, assegurant que el jugador rebi una representació actualitzada de la seva posició a la pantalla.



Figura 4.4.2: Blueprint del mini mapa (Font: elaboració pròpia)

#### 4.5 Implementació de contorns ressaltats en objectes interactuables

Per poder identificar millor els objectes interactuables de l'escena, a part de les targetes que es mostren quan un usuari s'apropa, també s'implementaran uns contorns ressaltats (*outlines*) per aquests objectes.

Els outlines són efectes visuals que destaquen les vores dels objectes en una escena. La implementació dels outlines als objectes es basa en la manipulació d'informació de profunditat (*depth*) i l'ús del postprocessament. Per començar, és essencial entendre què és el depth. En un entorn 3D, el depth fa referència a la distància de cada píxel d'un objecte des de la càmera.

Aquesta informació ens permet determinar quins objectes són més a prop i quins estan més lluny. Aquells llocs on es presenta un canvi significatiu en el depth, és a dir, on la distància entre objectes varia significativament, solen indicar la presència d'una vora. Per tant, per poder separar-los de la resta d'objectes, s'activa l'atribut *Render Custom Depth Pass* en aquests objectes, que són com etiquetes per indicar a Unreal Engine que els processin de manera particular: un cop activada aquesta opció, els objectes seleccionats escriuen la informació de depth en un buffer especial anomenat Custom Depth Buffer.

Després, es crea un material de postprocessament que actua com un filtre aplicat a tots els objectes continguts dins del *Post Process Volume* definit. Seguint el tutorial [2] es configura aquest material perquè llegeixi la informació del Custom Depth Buffer, identifiqui les vores, i apliqui un color brillant, en aquest cas el groc, a aquestes àrees.

Per aplicar aquest material de postprocessament a l'escena, cal utilitzar un Post Process Volume, que serveix per definir una àrea dins de l'escena on s'aplicaran els efectes de postprocessament indicats. Aquest també pot ser configurat com a sense límits (*Unbound*), i així els efectes cobriran tota l'escena. A aquest Post Process Volume se li assigna el material de postprocessament creat, assegurant així que el volum aplicarà l'efecte d'outline a tots els objectes que tenen activat el Render Custom Depth Pass dins d'aquest volum. En la Figura 4.5.1 es mostra el Post Process Volume amb el material aplicat. D'aquesta forma, tots els objectes interactuables de l'escena es veuran amb les vores pintades amb color groc.



Figura 4.5.1: Post Process Volume amb el material de postprocessament aplicat (Font: elaboració pròpia)

## 4.6 Exportació i instal·lació del projecte

El procés d'exportació del projecte es fa des d'Unreal Engine. Per exportar l'aplicació, cal accedir al menú File i seleccionar *Package Project*, en aquesta finestra es tria la plataforma de destinació, en aquest cas Windows, i es configuren els paràmetres necessaris per a l'exportació, com la qualitat gràfica i la resolució. Una vegada finalitza el procés d'exportació, Unreal Engine genera una carpeta que conté tots els fitxers necessaris per executar l'aplicació, anomenada "Windows" si s'havia seleccionat Windows com a plataforma de destinació. Atès que el desenvolupament del projecte es va fer en una màquina amb sistema operatiu Windows, la generació d'un paquet per a macOS presenta diverses complicacions tècniques. Primer de tot es necessita un dispositiu Mac per dur a terme la compilació i empaquetat final, ja que les eines necessàries, i l'entorn de desenvolupament d'Apple, només està disponible en sistemes operatius macOS i no en Windows. Encara que hi ha solucions com l'ús d'una màquina remota amb macOS, això implica tenir accés a aquest maquinari addicional i configurar una connexió remota, cosa que no va ser factible en el context d'aquest projecte. A més, el procés també requereix una llicència de desenvolupador d'Apple per crear perfils de desenvolupament i signar les aplicacions [36], afegint una altra capa de complexitat i cost.



Per tal de poder accedir a l'aplicació sense tenir l'Unreal Engine, s'han de seguir els següents passos:

- 1. Descarregar la carpeta exportada Windows.
- 2. Extreure els fitxers si estan comprimits.
- 3. Executar el fitxer principal (per exemple, "kitchen.exe" a Windows).

En general, per poder executar l'exe i que funcioni correctament, es necessiten tenir unes dependències que inclouen [56]:

- $\circ~$  Visual C++ Redistributable
- $\circ~$  DirectX End-User Runtimes
- Altres llibreries externes/ de tercers

Si l'equip de l'usuari no té aquestes dependències instal·lades, l'instal·lador del joc generalment pregunta si vol descarregar i instal·lar-les automàticament.

## 4.7 Manual de l'aplicació kitchen

En iniciar l'aplicació "kitchen.exe", l'usuari es trobarà amb una vista en primera persona del personatge. A la Figura 4.7.1 es mostra la ubicació dels objectes interactius, en qualsevol moment l'usuari pot prémer la tecla [V] per canviar a la vista en tercera persona.



Figura 4.7.1: Objectes interactuables de l'escena (Font: elaboració pròpia)

En aquesta vista, l'usuari pot explorar la cuina movent-se amb les tecles [W] per avançar, [A] per moure's a l'esquerra, [S] per retrocedir, [D] per moure's a la dreta, i saltant fent servint la barra d'espai. L'escena té una sala d'estar i un dormitori. Inicialment, la porta del dormitori està tancada. Per obrir-la, l'usuari ha d'acostar-se a la porta fins que aparegui una targeta d'interacció amb el missatge "press F to open/close door". En prémer [F], la porta s'obrirà i l'usuari podrà accedir a ell. En el mini mapa que està a la part superior dreta indicarà que l'usuari ha entrat al dormitori.



Figura 4.7.2: Interfície menú de l'escena (Font: elaboració pròpia)

Si l'usuari necessita consultar els controls o ha oblidat de com fer una acció específica, com per exemple el d'entrar a mode foto, pot seguir la indicació a la part inferior esquerra de la pantalla que suggereix pressionar [M] per accedir al menú. En entrar al menú, el joc es pausa i es mostra una llista de controls, com el d'entrar al mode foto amb [H] i clicar el botó esquerre del ratolí per fer captura de pantalla. Després de revisar els controls, l'usuari pot sortir del menú prement el botó [Resume].

En reprendre el joc, l'usuari pot entrar en el mode foto prement [H], tenint en compte que l'usuari no podrà accedir al menú mentre està en mode foto, per poder tornar al mode de joc normal s'ha de prémer [H] de nou.

A més de la porta, hi ha altres objectes interactius a la cuina que s'indiquen a la Figura 4.7.1 l'usuari pot seguir interactuant amb la resta d'objectes.

Per sortir de l'aplicació, haurà d'anar al menú i prémer el botó [Quit] o fer servir [Alt + F4] per tancar l'aplicació.

## 5 Revisió general

Tenint en compte els objectius generals del projecte, que inclouen la de veure el valor pràctic de la tecnologia de realitat virtual i la creació d'un sistema de visualització d'interiors en realitat virtual no immersiva que serveix com a exemple en el qual a partir d'aquest poder-ne explorar d'altres, es revisaran i s'avaluaran tots els objectius específics que es van planejar, per veure si s'han complert i, en conseqüència, determinar si s'han assolit els objectius generals del projecte.

Durant la fase d'exploració, es van revisar les tecnologies i plataformes de realitat virtual existents, com Unity i Unreal Engine. Després de l'anàlisi, es va decidir utilitzar Unreal Engine per les seves capacitats i recursos que ofereix. En aquesta fase, també s'ha completat la formació necessària, adquirint les habilitats per manipular l'editor d'Unreal Engine i fer servir Blueprints i codi per implementar diverses funcionalitats.

En la fase de desenvolupament, es va començar pel disseny del sistema. Es va crear una interfície d'usuari intuïtiva que permet als usuaris explorar l'entorn virtual de manera fluida, i es van definir els casos d'ús de les funcionalitats essencials de l'aplicació. Després de tenir el disseny establert, es va procedir a la construcció de l'escena fent servir els coneixements adquirits durant la fase d'exploració. Finalment, es van implementar les funcionalitats interactives complint els requisits dels casos d'ús, que permeten als usuaris interactuar amb diversos elements de l'entorn. Per tant, els objectius de la fase de desenvolupament també s'han assolit completament, permetent als usuaris no només observar l'escena creada, sinó també simular l'experiència d'utilitzar l'espai.

Fins aquí, s'ha recorregut el procés integral des de la planificació fins al desenvolupament del sistema de visualització d'interiors amb realitat virtual no immersiva, complint els objectius establerts inicialment. S'ha vist un dels valors pràctics d'aquesta tecnologia implementant una aplicació de visites virtuals en què les funcionalitats estan dissenyades per permetre als usuaris explorar de manera interactiva diferents elements de l'entorn. I a partir de les funcionalitats implementades i la documentació, serveixen com a exemples concrets i aplicables que altres desenvolupadors poden utilitzar per facilitar la creació de projectes similars en futurs desenvolupaments.

## 6 Anàlisi econòmica i de costos

Aquesta secció es dedicarà a examinar els diversos costos associats amb el desenvolupament i implementació del projecte. Es presentarà una estimació detallada que reflecteixi la inversió necessària tant en termes de recursos humans com materials.

#### • Recursos humans i costos laborals

Per al desenvolupament del projecte, es contractarà a un estudiant a temps parcial per un total de 450 hores de treball. Tenint en compte el salari mínim establert per la Universitat de Barcelona de 8 euros per hora [8], el cost laboral seria d'uns:

$$450 \,\mathrm{h} \times 8 \,\mathrm{eur/h} = 3600 \,\mathrm{C}$$

#### • Inversió en maquinari i programari

Encara que per aquest projecte no s'ha plantejat fer servir dispositius de realitat virtual, sí que es necessitarà un ordinador amb capacitat suficient per al desenvolupament en Unreal Engine i per a la implementació de models. S'estima aquest cost a 1000 euros [40].

Per calcular el cost dels equips en un projecte, generalment es fa servir el mètode de depreciació, basant-se en la vida útil esperada de l'equip per calcular el cost mensual:

Se suposa que la vida útil esperada de l'equip és d'aproximadament, 3 anys (36 mesos). Es divideix el cost inicial de l'equip entre el total de mesos del període de depreciació.

#### $1000 \div 36 = 27,78 \oplus$

Per calcular el cost durant el període del projecte es multiplica el cost mensual de depreciació pel nombre de mesos que l'equip s'ha utilitzat al projecte. Per exemple, si el projecte dura 3 mesos, el cost de l'equip al projecte seria:

$$27,78\times3=83,34\,\textcircled{\bullet}$$

I pel que fa a *software* necessari per al desenvolupament, incloent les llicències d'Unreal Engine, són de lliure accés o ofereixen plans educatius gratuïts. I respecte als recursos que es necessiten per a la creació de l'escena, suposem un cost entre:

$$0\sim 50\,{\rm C}$$

#### • Despeses operacionals i fixes

Les despeses mensuals operacionals com l'internet i utilitats són estimades a: Connexió a internet:

$$30 \,\mathrm{eur/mes} \times 3 \,\mathrm{mesos} = 90 \,\mathrm{ee}$$

Electricitat i aigua:

$$50 \,\mathrm{eur/mes} \times 3 \,\mathrm{mesos} = 150 \,\mathrm{ee}$$

## $\circ\,$ Costos de local

Encara que es treballa de manera remota, es necessitarà un espai de treball. Lloguer d'espai de treball:

$$500 \,\mathrm{eur/mes} \times 3 \,\mathrm{mesos} = 1.800 \,\mathrm{C}$$

### • Reserva per a Imponderable

Per a cobrir possibles imprevistos, s'afegeix un 10% sobre la suma total dels costos estimats.

Reserva per a imponderables:

 $10\% \times (3.600 + 83, 34 + 50 + 90 + 150 + 1.800) = 577, 334 \oplus$ 

#### • Resum de Costos

Fent la suma de tots els conceptes, tenim un cost total estimat per al projecte de:

 $3.600 + 83, 34 + 50 + 90 + 150 + 1.800 + 577, 334 = 6.350, 674 \\ \textcircled{\bullet}$ 

## 7 Conclusions i treball futur

#### 7.1 Vies de recerca futura

Els resultats obtinguts demostren el potencial d'Unreal Engine per desenvolupar entorns de realitat virtual interactius, amb diverses aplicacions potencials al món real. Per exemple es poden fer visites virtuals que permeten als compradors explorar habitatges de manera interactiva des de qualsevol lloc. Encara que en aquest projecte s'ha centrat en la creació d'entorns de realitat virtual interactius, hi ha moltes altres àrees que podrien beneficiar-se d'investigacions addicionals i millores futures, podria ser per exemple el que es va proposar inicialment, en avanços en intel·ligència artificial, d'investigar la incorporació dels models d'intel·ligència artificial més avançats i específics per a tasques com el reconeixement d'objecte; també es poden veure avenços a l'àmbit industrial, com per exemple en aplicacions com simuladors de conducció, simuladors de vol per a pilots i simuladors de combat per al personal militar, etc. En el camp de l'educació i la formació, la realitat virtual pot ser utilitzada per practicar procediments mèdics en un entorn sense riscos, i que obre oportunitats perquè les persones puguin practicar aquestes tasques de manera segura, mentre se senten prou immerses perquè l'experiència sigui realista i transferible al món real [25]. Tot i que la tecnologia de realitat virtual ha avançat significativament en els darrers anys, continua evolucionant, sent una àrea de recerca i desenvolupament activa amb un gran potencial per explorar.

## 7.2 Conclusions

Després de completar totes les fases del projecte, des de l'exploració fins al desenvolupament, s'han complet els objectius generals establerts tal com s'han mencionat la revisió general, s'ha vist el valor pràctic de la tecnologia de realitat virtual i s'ha creat un exemple senzill que pot ser millorat i desenvolupat per a futurs treballs.

Tot i això, els aprenentatges obtinguts amb aquest projecte van molt més enllà del que s'havia proposat inicialment. En primer lloc, començant amb la investigació del camp de la realitat virtual, s'ha descobert l'amplitud de les seves aplicacions en diferents àmbits. A mesura que s'avança en l'aprenentatge d'Unreal Engine, i el desenvolupament del sistema, es van poder veure les connexions amb les assignatures estudiades a la carrera. No obstant això, hi va haver parts en què va ser necessari aprendre coses noves des de zero.

En resum, aquest projecte no només ha proporcionat coneixements relacionats amb la tecnologia de realitat virtual, sinó que també ha ensenyat molt sobre el procés de recerca en general. S'ha après a gestionar les iteracions del projecte, a afrontar les pauses necessàries i a redefinir objectius, és a dir, a comprendre que investigar qualsevol tema és un camí ple de reptes.

Per tant, tant l'estudi de la realitat virtual com l'elaboració d'aquest projecte han estat experiències profundament formatives que han proporcionat un aprenentatge valuós. El projecte no només ha aconseguit els objectius establerts, sinó que també ha contribuït significativament al meu desenvolupament personal i acadèmic.

## Bibliografia

- F. Arici, P. Yildirim, Ş. Caliklar, and R. M. Yilmaz. Research trends in the use of augmented reality in science education: Content and bibliometric mapping analysis. *Computers & Education*, page 103647, 2019.
- M. Aspland. Creating an outline material in unreal engine 5. https://www.youtube. com/watch?v=INiUXnNlB1I, February 23 2023. Retrieved May 31, 2024.
- [3] S. E. August, M. L. Hammers, D. B. Murphy, A. Neyer, P. Gueye, and R. Q. Thames. Virtual engineering sciences learning lab: Giving stem education a second life. *IEEE Transactions on Learning Technologies*, 9(1):18–30, 2016.
- [4] T. Bahrynovska. Overview of vr development: History, tools, & best practices. Forbytes, https://forbytes.com/blog/vr-development-overview/, March 13 2023.
- [5] V. Bogicevic, S. Seo, J. A. Kandampully, S. Q. Liu, and N. A. Rudd. Virtual reality presence as a preamble of tourism experience: The role of mental imagery. *Tourism Management*, 74:55–64, 2019.
- [6] Grigore C Burdea and Philippe Coiffet. Virtual reality technology. John Wiley & Sons, 2003.
- [7] CGTrader. Cgtrader. cgtrader 3d models for vr / ar and cg projects. https: //www.cgtrader.com/, 2011. Retrieved May 11, 2024.
- [8] Comissió Acadèmica de la Facultat de Matemàtiques i Informàtica. Facultat de matemàtiques i informàtica. https://mat.ub.edu/empreses/informaciopractiques-en-empresa/#requisits, 2018.
- [9] Epic Games, Inc. Material editor use. https://dev.epicgames.com/ documentation/en-us/unreal-engine/unreal-engine-material-editoruser-guide. Retrieved May 8, 2024.
- [10] Epic Games, Inc. Blueprint editor blueprint class ui. https://dev.epicgames.com/ documentation/en-us/unreal-engine/blueprints-visual-scripting-userinterface-for-blueprint-classes-in-unreal-engine, 2024.
- [11] Epic Games, Inc. Blueprint editor details panel. https://dev.epicgames.com/ documentation/en-us/unreal-engine/details-panel-in-the-blueprintsvisual-scriting-editor-for-unreal-engine, 2024.
- [12] Epic Games, Inc. Blueprint editor graph editor. https://dev.epicgames.com/ documentation/en-us/unreal-engine/graph-editor-for-the-blueprintsvisual-scripting-editor-in-unreal-engine, 2024.
- [13] Epic Games, Inc. Blueprint editor my blueprint panel. https://dev.epicgames. com/documentation/en-us/unreal-engine/my-blueprint-panel-in-theblueprints-visual-scripting-editor-for-unreal-engine, 2024.
- [14] Epic Games, Inc. Blueprints visual scripting in unreal engine unreal engine 5.4 documentation — epic developer community. Epic Games DEV, https://dev.epicgames.com/documentation/en-us/unreal-engine/ blueprints-visual-scripting-in-unreal-engine, 2024.

- [15] Epic Games, Inc. Components window in unreal engine unreal engine 5.4 documentation — epic developer community. Epic Games DEV, https: //dev.epicgames.com/documentation/en-us/unreal-engine/componentswindow-in-unreal-engine, 2024. Retrieved May 8, 2024.
- [16] Epic Games, Inc. Documentation epic developer community. Epic Games DEV, https://dev.epicgames.com/documentation/en-us/unreal-engine/ hardware-and-software-specifications-for-unreal-engine, 2024.
- [17] Epic Games, Inc. Programming with c++ epic developer community. Epic Games DEV, https://dev.epicgames.com/documentation/en-us/unreal-engine/programming-with-cplusplus-in-unreal-engine, 2024.
- [18] Epic Games, Inc. Static mesh editor ui. https://dev.epicgames.com/ documentation/en-us/unreal-engine/static-mesh-editor-ui-in-unrealengine, 2024.
- [19] Epic Games, Inc. Unreal editor interface. Unreal Engine 5.4 Documentation, https://dev.epicgames.com/documentation/en-us/unreal-engine/ unreal-editor-interface, 2024.
- [20] Epic Games, Inc. Unreal engine reflection system. Wikipedia, https: //dev.epicgames.com/documentation/en-us/unreal-engine/reflectionsystem-in-unreal-engine, 2024.
- [21] Epic Games, Inc. Unreal engine terminology unreal engine 5.4 documentation — epic developer community. Epic Games DEV, https://dev.epicgames.com/ documentation/en-us/unreal-engine/unreal-engine-terminology, 2024.
- [22] Epic Games, Inc. Unreal object handling epic developer community. https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ ProgrammingWithCPP/UnrealArchitecture/Objects/Optimizations/, 2024.
- [23] Epic Games, Inc. User interface reference for the blueprints visual scripting editor in unreal engine — unreal engine 5.4 documentation — epic developer community. Epic Games DEV, https://dev.epicgames.com/documentation/en-us/unrealengine/user-interface-reference-for-the-blueprints-visual-scriptingeditor-in-unreal-engine, 2024.
- [24] Free3D. Free3d. 3d models for free free3d.com. https://free3d.com/, 2014. Retrieved May 11, 2024.
- [25] Ayah Hamad and Bochen Jia. How virtual reality technology has changed our lives: an overview of the current and potential applications and limitations. *International journal of environmental research and public health*, 19(18):11278, 2022.
- [26] R. L. Hite, M. G. Jones, G. M. Childers, M. Ennes, K. Chesnutt, M. Pereyra, and E. Cayton. Investigating potential relationships between adolescents' cognitive development and perceptions of presence in 3-d, haptic-enabled, virtual reality science instruction. *Journal of Science Education and Technology*, 2019.
- [27] Hunter G. Hoffman, Todd L. Richards, Arthur R. Bills, Theodore Van Oostrom, Jill Magula, Eric J. Seibel, and Sam R. Sharar. Using fmri to study the neural correlates of virtual reality analgesia. CNS Spectrums, 11(01):45–51, 2006.

- [28] Charlene Jennett, Anna L Cox, Paul Cairns, Samira Dhoparee, Andrew Epps, Tim Tijs, and Alison Walton. Measuring and defining the experience of immersion in games. *International journal of human-computer studies*, 66(9):641–661, 2008.
- [29] M. Lewis and J. Jacobson. Game engines. Communications of the ACM, 45(1):27, 2002.
- [30] L.-P. (Lynn) Lin, S.-C. (Lucy) Huang, and Y.-C. Ho. Could virtual reality effectively market slow travel in a heritage destination? *Tourism Management*, 78:104027, 2020.
- [31] Infurnia Technologies Pvt Ltd. Interior design software. https://www.infurnia.com/interior-design-software, 2024.
- [32] Francisco Marotta, Gastón Andrés Addati, and José Andrés Montes de Oca. Simulaciones con realidad inmersiva, semi inmersiva y no inmersiva. Technical report, Serie Documentos de Trabajo, 2020.
- [33] Matias Matamala-Gomez, Sara Bottiroli, Ombretta Realdon, Giuseppe Riva, Laura Galvagni, Thomas Platz, Cecilia Tassorelli, et al. Telemedicine and virtual reality at time of covid-19 pandemic: an overview for future perspectives in neurorehabilitation. Frontiers in Neurology, 12:646902, 2021.
- [34] Microsoft. Visual studio code and unity. Visual Studio Code, https://code. visualstudio.com/docs/other/unity, 2024.
- [35] R. Mitra. 14 best video game engines, ranked. Game Rant, https://gamerant. com/best-video-game-engines-ranked/, February 7 2024.
- [36] Naimaste. How to export and package for mac users? https://forums. unrealengine.com/t/hpw-to-export-and-package-for-mac-users/142541, April 1 2020.
- [37] Next Level 3D. Archviz interior vol.3 in architectural visualization ue marketplace. https://www.unrealengine.com/marketplace/en-US/product/archvizinterior-vol-3, August 10 2023.
- [38] Nico. Nne overview 5.3 neural network engine (nne). Epic Games DEV, https://dev.epicgames.com/community/learning/courses/e7w/unrealengine-neural-network-engine-nne/G3rx/unreal-engine-nne-overview-5-3, September 5 2023.
- [39] Meaghan O'Neill and Kelsey Mulvey. 26 interior design software programs to download in 2024. https://www.architecturaldigest.com/story/best-interiordesign-software-programs, January 2024.
- [40] PC Componentes y Multimedia SLU. Pccom ready intel core i5-12400f / 32gb / 1tb ssd / rtx 4060 ti + windows 11 home. https://www.pccomponentes.com/pccomready-intel-core-i5-12400f-32gb-1tb-ssd-rtx-4060-ti-windows-11-home, 2024.
- [41] Perforce Software, Inc. What are the best game engines? Perforce Software, https: //www.perforce.com/blog/vcs/most-popular-game-engines#unreal, 2023.
- [42] Perforce Software, Inc. What is a gaming engine? game engine overview. https: //www.perforce.com/resources/vcs/game-engine-overview, 2024.

- [43] L. Perry. Modular level and component design. *Game Developer*, pages 30–35, 2002.
- [44] PolyHaven. About / contact poly haven. https://polyhaven.com/aboutcontact, 2024. Retrieved May 14, 2024.
- [45] Anna Puig. Introducció a la visualització interactiva 3d. Grau d'Enginyeria Informàtica, Facultat de Matemàtiques i Informàtica (UB), PowerPoint presentation, 2023.
- [46] E. Reinhard. High dynamic range imaging. In Computer Vision: A Reference Guide, pages 1–6. Springer International Publishing, Cham, 2020.
- [47] Giuseppe Riva et al. Presence-inducing media for mental health applications. In Matthew Lombard, Frank Biocca, Jonathan Freeman, Wijnand IJsselsteijn, and Rachel Schaevitz, editors, *Immersed in Media*, pages 163–185. Springer, Cham, 2015.
- [48] George G Robertson, Stuart K Card, and Jock D Mackinlay. Three views of virtual reality: nonimmersive virtual reality. *Computer*, 26(2):81, 1993.
- [49] Mr. Debashish Roy. Head mounted display market 2021 by global key players. https://www.openpr.com/news/2492206/head-mounted-displaymarket-2021-by-global-key-players-types, September 21 2021.
- [50] Seozero00. I want to use yolov7 on unreal engine. Unreal Engine Forum, https://forums.unrealengine.com/t/i-want-to-use-yolov7-onunreal-engine/1623054, January 18 2024.
- [51] Ltd. Suzhou Yundu Network Technology Co. Business running system model of house online preview and sale. Google Patents, https://patents.google.com/patent/ CN103116850A/en, 2013. Patent application CN103116850A.
- [52] Unity Technologies. Motor de unity. https://unity.com/es/products/unityengine?\_rsc=2slht, November 9 2017.
- [53] Unity Technologies. Herramientas en tiempo real para el desarrollo en 3d, ra y rv productos. Unity, https://unity.com/es/products, 2024.
- [54] Universidad Europea. ¿qué es un motor de juegos? https://creativecampus. universidadeuropea.com/blog/motor-juegos/, November 9 2017.
- [55] xiii139. Build error in ue5: Lnk1181 cannot open libcurl.lib file. Unreal Engine Forums, https://forums.unrealengine.com/t/build-error-in-ue5lnk1181-cannot-open-libcurl-lib-file/1803203, April 14 2024.
- [56] zeblote. Required dependencies for ue4 games. https://forums.unrealengine. com/t/required-dependencies-for-ue4-games/4826?page=3, 2024.
- [57] Z. Zhu and Y. Du. Research on interior design optimization based on virtual reality technology. In *Journal of Physics: Conference Series*, volume 1746, page 012063. IOP Publishing, 2021.

## A Annex

En aquesta secció es presenten els Blueprints del sistema desenvolupats pel projecte, així com un glossari de termes utilitzats al llarg del document. Aquests elements proporcionen una visió detallada de la implementació tècnica i clarifiquen els conceptes clau per facilitar la comprensió del treball.

### A.1 Glossari

- **API** Application Programming Interface. Interfície de programació d'aplicacions. Un conjunt de regles i definicions que permeten que diferents aplicacions es comuniquin entre elles. 8
- **CAD** Computer Aided Design. Disseny assistit per ordinador. És una tecnologia que utilitza programari per dissenyar i documentar processos d'enginyeria. Facilita la creació de plans i models tridimensionals detallats. 8
- **Camera Boom** Component a Unreal Engine que controla la posició i el moviment d'una càmera respecte al seu actor principal. 40
- depth Profunditat, mesura de la distància des de la càmera fins als objectes en una escena 3D, utilitzada per determinar-ne la posició relativa. 41
- Enhanced Input Sistema millorat de gestió d'entrades a Unreal Engine que permet configurar i gestionar controls d'usuari. 38
- **Event Construct** Esdeveniment a Unreal Engine que s'executa quan un giny és construït i afegit a la pantalla. 40
- **Event Tick** Esdeveniment a Unreal Engine que s'executa en cada frame, permetent actualitzar lògicament els actors en temps real. 33
- frame Un frame en anglès és una única imatge estàtica en una seqüència d'imatges. 33
- **framework** Un conjunt d'eines i llibreries que proporcionen una estructura estàndard per desenvolupar aplicacions. 10
- Get Actor Location Funció a Unreal Engine que retorna la posició actual d'un actor al món. 40
- Get World Rotation Funció a Unreal Engine que retorna la rotació actual d'un actor al món. 40
- HMD Head-Mounted Display. Un dispositiu que es porta al cap i mostra contingut visual directament davant dels ulls de l'usuari, comunament usat en realitat virtual. 6
- **IDE** Integrated Development Environment. Entorn de desenvolupament integrat. Entorn de desenvolupament de software que proporciona eines completes per a la programació. 11

- macros Secuència d'instruccions automatitzades que s'executen per realitzar tasques repetitives de manera més eficient. 12
- Marketplace Plataforma a Unreal Engine on els desenvolupadors poden comprar i vendre assets, plugins i altres recursos per al desenvolupament de jocs i aplicacions. 27
- **materials** Configuracions que determinen com les superfícies dels models 3D interactuen amb la llum i es veuen a l'entorn. 10
- **mesh** Una estructura de dades que defineix la forma d'un objecte 3D mitjançant vèrtexs, arestes i cares. 19
- Normal Mapa de normals, una textura que simula detalls de superfície en un model 3D. 28
- outlines Efecte visual a Unreal Engine que destaca les vores dels objectes. 41
- **PBR** *Physically Based Rendering.* Renderitzat basat en física. És un mètode de renderització que simula de manera precisa com la llum interactua amb els materials, produint imatges més realistes i físicament correctes. 29
- **Post Process Volume** Una regió a Unreal Engine on s'apliquen efectes de postprocessament, com el desenfocament o correcció de color, a l'escena. 41
- **PreConstruct** Esdeveniment a Unreal Engine que s'executa abans que el giny sigui construït, permetent configuracions prèvies. 38
- **PyTorch** Una llibreria de codi obert per a aprenentatge automàtic i xarxes neuronals. 25
- **Quixel Bridge** Una eina que facilita la importació i la gestió d'assets de *Megascans* en aplicacions. 26
- Quixel Megascans Llibreria en línia que ofereix una àmplia col·lecció de recursos digitals d'alta qualitat, i que s'integren fàcilment amb diversos motors de joc i programari de disseny 3D. 27
- Render Custom Depth Pass Opció a Unreal Engine que permet renderitzar la informació de profunditat de certs objectes en un buffer especial. 41
- Render Target Superfície on es pot renderitzar una escena o part d'ella, per després ser utilitzada com a textura a Unreal Engine.. 38
- Roughness Propietat d'un material que defineix com de rugosa o suau és la superfície, afectant la dispersió de la llum. 28
- **RT3D** Real-Time 3D. Renderitzat en Temps Real 3D. Tecnologia que permet la creació i visualització de gràfics tridimensionals en temps real, utilitzada en videojocs, simulacions i aplicacions interactives. 8
- Runtimes Entorns d'execució que proporcionen els recursos i els serveis necessaris perquè els programes de software s'executin correctament en diferents sistemes operatius. 18

- Set Render Transform Funció que permet ajustar la posició, la rotació i l'escala d'un objecte renderitzat. 40
- Set Texture Parameter Value Node a Unreal Engine que permet canviar dinàmicament els paràmetres de textura d'un material. 38
- shaders GLSL Programes escrits al Llenguatge d'Ombrejat d'OpenGL (*OpenGL Shading Language*, GLSL) que controlen com es rendeixen els gràfics a la pantalla, aplicant efectes com il·luminació, ombres i textures en temps real. 10
- socket Punt d'ancoratge en un esquelet d'un model 3D a Unreal Engine, utilitzat per adjuntar altres objectes o efectes. 40
- SpawnActor Funció a Unreal Engine que crea i posiciona una nova instància d'un actor a l'escena. 38
- **STEM** Science, Technology, Engineering, and Mathematics. Ciència, Tecnologia, Enginyeria i Matemàtiques. Un enfocament educatiu que integra la ciència, tecnologia, enginyeria i matemàtiques. 7
- **Texture Samples** Operacions que es realitzen en el motor Unreal Engine per a obtenir valors de textura d'una imatge aplicats a una superfície en el procés de renderitzat.. 28
- **textures** Imatges aplicades a superfícies de models 3D per donar-los aparença detallada i realista. 15
- **timeline** Eina a Unreal Engine que permet crear i gestionar animacions i canvis de propietats al llarg del temps. 35
- **Unbound** Configuració a Unreal Engine que permet que un volum de postprocessament afecti tota l'escena, sense límits d'àrea. 41
- **VT** Virtual Tourism. Turisme Virtual, una manera d'explorar destinacions turístiques a través d'entorns virtuals immersius, sense necessitat de viatjar físicament. 7
- Widget blueprint Eina a Unreal Engine per crear interfícies d'usuari interactives mitjançant blueprints. 32

# A.2 Blueprints



Figura A.2.1: Inicialització del personatge a tercera persona



Figura A.2.2: Creació i visualització de l'ui\_hint



Figura A.2.3: Control de la interfície Menú I

J Add to Viewport       Target is User Widget		-fQ SET	f Set Game Paused
Target	Player Controller	💿 Show Mouse Cursor 🧹 🔍	🕒 Paused 🗸 🛛 Return Value 🔿
· ·	In Widget to Focus	- Target	
	In Mouse Lock Mode Do Not Lock Hide Cursor During Capture		
Menu Widget	• Flush Input		
	<ul> <li> <i>f</i> Get Player Controller         <ul> <li></li></ul></li></ul>		
🗲 Set Input Mode Game Only	f C		
Player Controller	Show Mouse Cursor		
Flush Input	Target		
<i>f</i> Get Player Controller	alue		

Figura A.2.4: Control de la interfície Menú II

Quit		
🔷 On Clicked (QuitBtn) 🗖	f Quit Game	
	-•	D
	Specific Player	
	Quit Preference	
	O Quit V	
	Ignore Platform Restrictions	

Figura A.2.5: Esdeveniment botó Quit

Resume					
On Clicked (ResumeBtn)     Set Game Paused	🗲 Set Input Mode Game Only		► → Cast To BP_ThirdPersonCharacter		J Remove from Parent Target is Widget
▶ <b>─</b> ▶ ▶	-• •-		- <b>b b</b> -		— → D
Paused Return Value	Player Controller	Show Mouse Cursor	🔷 Object Cast Failed D	💿 Is Menu Opened 📃 🔍	➡ Target Self
	Flush Input	Target	As BP Third Person Character 🔶	Target	
f Get Player Controller	€ Get Player Controller	£ Get Player Character			
Player Index 0 Return Value	Player Index 0 Return Value	Player Index 0 Return Value			

Figura A.2.6: Esdeveniment botó Resume



Figura A.2.7: Configuració de variable Interaction amb Line Trace I



Figura A.2.8: Configuració de variable Interaction amb Line Trace II



Figura A.2.9: Canvi de perspectiva de la càmera

Capture scene to render targ	get and display			
© EnhancedinputAction IA, TakePhoto Triggere () Starte )- Action Value ()	Create Render Target 20  Wath 1920  Wath 1920  Height 1980  Create And Control of the second	SET • Target Gamma 33 • Target Peturn Yalus •	SET Transet Transet Scere Capture Component 20	Create Photo Display Widget

Figura A.2.10: Creació i visualització de Photo Display Widget I



Figura A.2.11: Creació i visualització de Photo Display Widget II



Figura A.2.12: Habilitar la interacció del ratolí a la UI del mode foto



Figura A.2.13: Esdeveniment botó tancar del ui\_screen

Spawn photo mode camera instar	nce			
C EnhancedInputAction IA_PhotoMode			Control trai	nsfer
Triggered D		😒 SpawnActor BP Photo Camera		
Started		-•	•	7 Possess Target is Controller
Action Value 📀	J GetWorld Transform Target is Scene Component	Class or ritoto came v (i) (	Return Value 🔹	D
	Follow Camera 💿 🔷 🗣 Target Return Value 🔶 —	Spawn Transform		
		Collision Handling Override Try To Adjust Location, Dut Always Spawn V	J Ge	t Player Controller
		<ul> <li>Instigator</li> </ul>	O Pk	ayer Index 0 Return Value

Figura A.2.14: Generar una instància de la càmera al mode foto



Figura A.2.15: Esdeveniment IA\_photoMode



Figura A.2.16: Esdeveniment del botó save en ui $\$ screen



Figura A.2.17: Esdeveniment tecla F per interactuar amb objectes



Figura A.2.18: Control estat de la televisió



Figura A.2.19: Control estat de l'olla



Figura A.2.20: Control estat de la llum a través de l'interruptor I



Figura A.2.21: Control estat de la llum a través de l'interruptor II



Figura A.2.22: Control estat de la porta I



Figura A.2.23: Control estat de la porta II



Figura A.2.24: Control estat de la llum de peu



Figura A.2.25: Configuració del material outline I


Figura A.2.26: Configuració del material outline II