



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

GRAU D'ENGINYERIA INFORMÀTICA

ESPECIALITAT DE COMPUTACIÓ

Eina per a la generació d'imatges físicament acurada de models 3D de patrimoni cultural

Tutor de GEP:

David Pardos Benito
Departament d'Organització d'Empreses

Autora:

Míriam Méndez Serrano

Director:

Imanol Muñoz Pandiella
Departament de Ciències de la Computació

Codirector:

Carlos Andujar Gran
Departament de Ciències de la Computació

9 d'octubre del 2022

Resum

EINA PER A LA GENERACIÓ D'IMATGES FÍSICAMENT ACURADES DE MODELS 3D DE PATRIMONI CULTURAL

Actualment les eines de visualització de models i de renderització, es centren en obtenir imatges fotorrealistes en l'àmbit artístic, però tot i que visualment ho poden semblar, moltes vegades aquestes imatges no segueixen les lleis de la física.

En aquest projecte es desenvolupa una eina científica que genera imatges basades en el comportament físic de la llum, simulant diferents condicions d'il·luminació, combinant fonts de llum natural i artificial. Aquesta eina estén el popular software de modelatge Blender permetent crear imatges físicament acurades, de manera senzilla mitjançant el programari Radiance.

L'aplicació principal de l'eina és ajudar a validar les hipòtesis d'historiadors de l'art en esglésies preromàniques. Utilitzarem com a cas d'estudi l'església de Sant Quirze de Pedret, que es troba a Berga.

Per a realitzar una renderització realista, la nostra eina haurà de tenir en compte diversos factors, com la posició del sol segons el dia i l'hora, el comportament de la llum en funció dels materials a les finestres (vidre, cuir...) i els tipus d'il·luminacions artificials emprats.

PARAULES CLAU: renderització, fotorrealisme, Radiance, Blender, Sant Quirze de Pedret.

Resumen

HERRAMIENTA PARA LA GNERACIÓN DE IMAGENES FÍSICAMENTE PRECISAS DE MODELOS 3D DE PATRIMONIO CULTURAL

Actualmente, las herramientas de visualización de modelos y de renderización se centran en obtener imágenes fotorrealistas en el ámbito artístico, pero aunque visualmente pueden parecerlo, muchas veces estas imágenes no siguen las leyes de la física.

En este proyecto se quiere desarrollar una herramienta científica que genere imágenes basadas en el comportamiento físico de la luz, simulando diferentes condiciones de iluminación, combinando fuentes de luz natural y artificial. Esta herramienta extiende el popular software de moldeado Blender permitiendo crear imágenes físicamente precisas de manera sencilla mediante el software Radiance.

La aplicación principal de la herramienta es ayudar a validar las hipótesis de historiadores de l'arte en iglesias prerrománicas. Utilizaremos como caso de estudio la iglesia de San Quirico de Pedret, que se encuentra en Berga.

Para realizar una renderización realista, nuestra herramienta deberá de tener en cuenta diversos factores, como la posición del Sol según el día y la hora, el comportamiento de la luz en función de los materiales en las ventanas (vidrio, cuero) y los tipos de iluminación artificial empleados.

PALABRAS CLAVE: renderización, fotorrealismo, Radiance, Blender, San Quirco de Pedret.

Abstract

TOOL FOR THE GENERATION OF PHYSICALLY ACCURATE 3D MODEL IMAGES OF CULTURAL HERITAGE

Nowadays model visualization and rendering tools are focused on obtaining photorealistic images in the artistic field, but even though they may look like visually correct, many times these images do not follow the laws of physics.

This project aims to develop a scientific tool that generates images based on the physical behavior of light, simulating different lighting conditions, combining natural and artificial light sources. This tool extends the popular Blender modeling software by allowing you to easily create physically accurate images using Radiance software.

The main application of the tool is to help validate the hypotheses of art historians regarding pre-Romanesque churches. We will use the church of Sant Quirze de Pedret in Berga as a case study.

To perform a realistic rendering, our tool must take into account various factors such as the position of the sun according to the day and time, the behavior of light depending on the materials in the windows (glass, leather) and the types of artificial lighting used.

KEYWORDS: rendering, photorealism, Radiance, Blender, Sant Quirze de Pedret.

AGRAÏMENTS

Gràcies a tota aquella gent que m'ha acompanyat personalment, i m'ha ajudat a seguir i acabar aquest Treball de Fi de Grau.

Especialment als meus tutors: Imanol Muñoz i Carlos Andujar, per haver-me donat ànims i forces de cara al treball. A més a més, d'aportar-me, en tot moment, observacions i comentaris que m'han permès reformular diferents apartats.

Per altra banda, també a la meva família, en particular a la meva mare, per donar-me suport en tot moment, i ajudar-me amb aquesta tasca titànica, llegint-se aquest projecte i revisant-lo amb mi. A més a més, d'assistir-me en tot el possible.

Als meus amics i companys, que m'han escoltat i acompanyat en aquest viatge, respectant que tenia un treball a realitzar, i acceptant la meva vida monàstica, sobretot en les últimes setmanes del TFG. Però vull mencionar a la Sandra Garcia ja que també m'ha proporcionat moltes opinions honestes sobre la claredat del treball.

Finalment, donar les gràcies a l'Òscar Argudo, a l'Immaculada Lorés Otzet, al Joan Font Comas, al Genís Àvila Casademont, al Xavier Pueyo, al Museu Comarcal i Diocesà de Solsona, al grup de recerca ViRVIG i a totes les persones que han treballat per l'elaboració del model digital de Sant Quirze de Pedret.

ÍNDIX DE CONTINGUTS

1	Contextualització	7
1.1	Introducció i context	7
1.2	Definició de conceptes	8
1.3	Descripció del problema a resoldre	10
1.4	Actors implicats	11
2	Justificació	12
2.1	Estat de l'art	12
2.2	Solució presa	13
3	Abast	15
3.1	Objectius i sub-objectius	15
3.2	Requeriments (funcionals i no funcionals)	15
3.3	Obstacles i riscos	16
4	Metodologia	17
5	Planificació temporal	19
5.1	Descripció de les tasques	19
5.1.1	Tasques de gestió del projecte	20
5.1.2	Tasques de desenvolupament	20
5.1.3	Recursos	23
5.2	Estimacions i Gantt	24
5.3	Gestió del risc: Plans alternatius i obstacles	28
6	Gestió econòmica	29
6.1	Costos de personal per activitat	29
6.2	Costos materials	31
6.3	Costos genèrics	31
6.4	Contingències	31
6.5	Costos dels imprevistos	32
6.6	Pressupost final	33
6.7	Control de gestió	33
6.8	Desviació del pressupost total	33

7	Sostenibilitat	36
7.1	Dimensió econòmica	36
7.2	Dimensió ambiental	36
7.3	Dimensió social	37
7.4	Matriu de sostenibilitat	37
7.5	Autoavaluació	38
8	Renderitzant amb Radiance	39
8.1	Escena Sant Quirze de Pedret	40
8.2	Materials	42
8.2.1	Definició del plàstic	44
8.2.2	Definició de les textures	45
8.2.3	Definició del vidre	46
8.2.4	Definició de translúcid	47
8.2.5	Definició dels materials	48
8.3	Geometria	49
8.3.1	Definició de la geometria	50
8.3.2	Previsualització	50
8.4	Llum natural	52
8.4.1	Definició del cel	54
8.4.2	Redefinició de les finestres	55
8.5	Llum artificial	56
8.5.1	Lampcolor	56
8.5.2	Fitxers IES	57
8.6	Renderització	58
8.6.1	Anàlisi de la il·luminació	60
8.6.2	Anàlisi fotomètric	64
8.6.3	Temps d'execució	65
8.7	Jerarquia d'arxius final	67
9	Conversió de Blender a Radiance	69
9.1	Conversió de l'escena	69
9.2	Conversió de la càmera	71
9.3	Renderització	73
10	Addon	74

10.1 Mockups	74
10.1.1 Escena	74
10.1.2 Materials	76
10.1.3 Llum	78
10.2 Implementació	79
10.2.1 Declaració de propietats	80
10.2.2 Programació d'interfícies	81
10.2.3 Creació d'operadors	84
10.2.4 Creació de l'Addon	87
10.3 Validació	88
11 Adversitats	91
12 Conclusions	93
12.1 Competències tècniques	93
12.2 Treball futur	94
12.3 Valoració personal	94
Referències bibliogràfiques	96
Glossari	99
A Annexos	101
A Instal·lació de Radiance	101
B Generar les renderitzacions obtingudes amb Radiance	102
C Codi de la conversió	102

ÍNDIX DE FIGURES

1	Sant Quirze de Pedret. Font: [1]	7
2	Procés de renderització d'OpenGL 2.0. Font: [3]	8
3	Algoritme de Ray Tracing. Font:[3]	8
4	Diferència entre els renderings i la realitat. Fonts: (a)[4], (b)[5], (c)[6]	9
5	Diferència amb i sense il·luminació global. Font: [7]	9
6	Magnituds fotomètriques. Font:[8]	10
7	Renderitzacions contemporànies. Fonts: (a)[9], (b)[10], (c)[11] i (d)[12]	12
8	Comparació de la sala d'estar renderitzada amb Radiance i fotografiada. Font: [15] . .	13
9	Diagrama de Gantt inicial. Font: elaboració pròpia	26
10	Diagrama de Gantt final. Font: elaboració pròpia	27
11	Procés de renderització amb Radiance. Font: elaboració pròpia	39
12	Escena de Sant Quirze de Pedret[26] ja configurada en Blender. Font: elaboració pròpia	40
13	Sistema de coordenades de Radiance. Font: elaboració pròpia	40
14	Diferències entre el material plàstic i el metall. Font: [29]	42
15	Fotografia de Sant Quirze de Pedret. Font: Repositori de fotografies de Pedret	43
16	Reflexió difusa ideal. Superfície Lambertiana. Font: [30]	43
17	Colour picker. Font: elaboració pròpia	44
18	Bola de plàstic amb diferents especularitats i rugositats. Font: [29]	44
19	Resultat del baking. Font: elaboració pròpia	45
20	Panell Blender, mostra els objectes i els materials. Font: elaboració pròpia	46
21	Unificació de les dues escenes Blender en una única escena. Font: elaboració pròpia . .	47
22	Exportació de l'objecte Blender a OBJ. Font: elaboració pròpia	49
23	Visualització de Pedret amb el programa objview. Font: elaboració pròpia	51
24	Exemples de condicions reals de cel amb els models CIE. Font: [33].	52
25	Renderització d'espelmes a Radiance. Font: elaboració pròpia	57
26	Renderització amb il·luminació IES. Font: elaboració pròpia	58
27	Comparació de les renderitzacions de Pedret1 i Pedret2. Font: elaboració pròpia	61
28	Comparació de Pedret1 i Pedret2 amb finestres de vidre. Font: elaboració pròpia . . .	62
29	Comparació de Pedret1 i Pedret2 amb finestres de cuir. Font: elapròpia	63
30	Pedret1, renderització amb llum natural de 12:00 a 15:00. Font: elaboració pròpia . . .	64

31	Anàlisi de la il·luminància. Font: elaboració pròpia	65
32	Càmera Blender amb els paràmetres de Radiance. Font: elaboració pròpia	72
33	Mockup de l'escena. Font: elaboració pròpia	74
34	Mockups de diferents materials. Font: elaboració pròpia	77
35	Mockups de les fonts de llum. Font: elaboració pròpia	78
36	Interfície de la il·luminació i els materials. Font: elaboració pròpia	82
37	Interfície del material antimatter. Font: elaboració pròpia	84
38	Interfície de l'escena. Font: elaboració pròpia	85
39	Nodes del material de construcció inorgànic de Blender. Font: elaboració pròpia	86
40	Configuració dels diferents materials. Font: elaboració pròpia	89
41	Paràmetres del cel. Font: elaboració pròpia	89
42	Paràmetres de renderització. Font: elaboració pròpia	89
43	Previsualització de l'escena amb una exposició de 12. Font: elaboració pròpia	90
44	Previsualització de l'escena amb una exposició de 70. Font: elaboració pròpia	90
45	Primera renderització de Sant Quirze de Pedret	91
46	Tercer i quart problemes del projecte. Font: elaboració pròpia	92

ÍNDIX DE TAULES

1	Magnituds fotomètriques i radiomètriques. Font: elaboració pròpia	10
2	Comparativa de simuladors d'il·luminació predictius. Font: Elaboració pròpia	12
3	Resum de les tasques. Font: Elaboració pròpia	24
4	Riscos i plans alternatius. Font: Elaboració pròpia	28
5	Sous per hora en funció del rol. Font: Elaboració pròpia	29
6	Cost estimat en funció de les tasques. Font: Elaboració pròpia	30
7	Cost del personal. Font: Elaboració pròpia	31
8	Cost del maquinari per personal. Font: Elaboració pròpia	31
9	Cost de contingència del 15%. Font: Elaboració pròpia	32
10	Cost d'imprevistos. Font: Elaboració pròpia	32
11	Pressupost final del projecte. Font: Elaboració pròpia	33
12	Cost final en funció de les tasques. Font: Elaboració pròpia	34
13	Emissions contaminats. Font: Elaboració pròpia	36
14	Matriu de sostenibilitat. Font: Elaboració pròpia	37

15	Comandes utilitzades en el programa rvu. Font: elaboració pròpia	52
16	Temps d'execució de les diferents escenes a les 11:00. Font: elaboració pròpia	65
17	Temps d'execució de gwindow1 en les diferents franges horàries. Font: elaboració pròpia	66
18	Especificacions tècniques del MSI PS42 8RB. Font: elaboració pròpia	66
19	Propietats d'interfície de l'escena. Font: elaboració pròpia	75
20	Materials d'interfície. Font: elaboració pròpia	76
21	Propietats d'interfície dels materials. Font: elaboració pròpia	77
22	Propietats d'interfície de la llum. Font: elaboració pròpia	79

ÍNDIX DE CODIS

1	Radiance - Definició del material de pedret monomaterial. Font: elaboració pròpia . . .	48
2	Radiance - Definició del material de Pedret multimaterial. Font: elaboració pròpia . . .	48
3	Radiance - Definició del material de la finestra de cuir. Font: elaboració pròpia	49
4	Radiance - Definició del material de la finestra de vidre. Font: elaboració pròpia	49
5	Definició de la geometria de Pedret monomaterial. Font: elaboració pròpia	50
6	Definició de la geometria de Pedret multimaterial. Font: elaboració pròpia	50
7	Definició de la geometria de la finestra de cuir. Font: elaboració pròpia	50
8	Definició de la geometria de la finestra de vidre. Font: elaboració pròpia	50
9	Definició del cel en Radiance. Font: elaboració pròpia	54
10	Definició millorada de la finestra de cuir en Radiance. Font: elaboració pròpia	55
11	Definició millorada de la finestra de vidre en Radiance. Font: elaboració pròpia	55
12	Definició del material de l'espelma en Radiance. Font: elaboració pròpia	57
13	Definició de la geometria de l'espelma en Radiance. Font: elaboració pròpia	57
14	Coddi de l'escena gwindow1. Font: elaboració pròpia	58

1 CONTEXTUALITZACIÓ

Aquest TFG d'Enginyeria Informàtica, està emmarcat en l'especialitat de Computació i es centra en la renderització, un dels grans dominis pertanyents a la branca de computació gràfica. El projecte s'ha fet en la modalitat A, és a dir, en la Facultat d'Informàtica de Barcelona i sota la direcció i co-direcció d'Imanol Muñoz i Carlos Andujar, respectivament.

Dos doctors i investigadors del grup de recerca de ViRVIG que em van proposar aquest tema, afí al projecte EHEM en el qual es trobaven treballant. La proposta va ser acceptada amb molta il·lusió, ja que des d'un principi tenia molt clar que volia fer el TFG amb el doctor Imanol.

1.1 INTRODUCCIÓ I CONTEXT

El cas d'estudi del projecte està centrat en l'església de Sant Quirze de Pedret, Figura 1, una obra preromànica de Catalunya, que ha estat restaurada i conservada durant el segle XX i recentment digitalitzada.

Aquesta digitalització ens ha estat proporcionada amb l'objectiu de poder renderitzar aquest patrimoni cultural generant imatges físicament acurades. La generació d'imatges físicament realistes en entorns de patrimoni cultural és encara un tema obert que interessa a les comunitats d'informàtica gràfica i d'història de l'art.



Figura 1: Sant Quirze de Pedret. Font: [1]

De fet, està relacionat amb la tercera tasca del projecte europeu EHEM[2] anomenada **Raise and propose the resolution of lighting problems** i que obté la següent descripció:

«To date, trials have been carried out for the restitution of these problems in digital models based on the analysis of natural lighting, which we intend to improve. We also propose to deal with artificial lighting by chandeliers or oil lamps, which produced effects of painting vibration at the moment when, for liturgical reasons, the images "acted".»

És a dir, la nostra tasca consistirà a intentar ajudar als historiadors de l'art en la comprensió de quin paper jugava la il·luminació, en els actes litúrgics. Jo m'encarragaré de realitzar una eina de renderització físicament acurada, per a poder crear escenes de patrimoni cultural, de manera interactiva amb il·luminacions artificials i natural (la il·luminació del Sol en un instant concret), amb la finalitat de poder generar una imatge que segueixi les lleis de la física. Aquesta eina no només serà útil per a l'església de Sant Quirze de Pedret, sinó també, per a altres monuments de patrimoni cultural.

1. Contextualització

1.2 DEFINICIÓ DE CONCEPTES

Abans d'entrar en matèria, hi ha uns coneixements bàsics que s'han de tenir en compte degut a l'ampli abast de les renderitzacions, començant per la definició d'aquesta mateixa paraula.

La renderització és el procés de generar imatges a partir de models 2D o 3D, mitjançant programes informàtics. En aquest procés es poden aplicar diverses tècniques de visualització, de les quals em centraré en dues bàsiques: Rasterització i Ray Tracing.

Rasterització: És una tècnica de renderització extremadament ràpida si la comparem amb el *Ray Tracing*, però menys precisa, ja que és un algoritme bàsic, que consisteix en projectar el model de vèrtexs i triangles a píxels en una imatge 2D, com mostra la Figura 2.

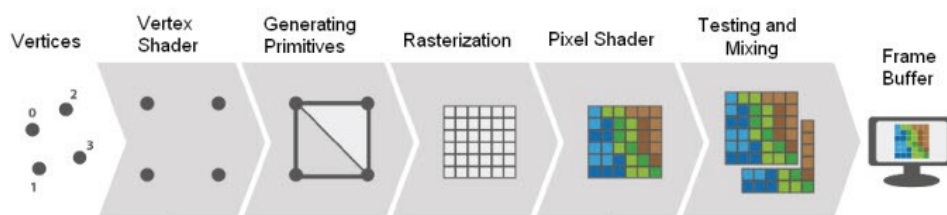


Figura 2: Procés de renderització d'OpenGL 2.0. Font: [3]

Ray Tracing: La diferència bàsica amb la *rasterització* és que la imatge 2D és generada a partir d'una sèrie de rajos que s'inicien des de l'observador (càmera) fins a l'escena a través del plànol de la imatge, tal com està representat en la Figura 3. Des del punt intersecat, es traça un nou raig en la direcció de les fonts de llum comprovant si aquestes són visibles. D'aquesta manera, el traçador de raigs calcula el color i l'ombregat exactes de cada punt de l'escena, tenint en compte el pla que estableix la posició de l'observador.

El *Ray Tracing* té la capacitat de generar un raig per a cada píxel de la imatge que es crea a la pantalla de l'ordinador. Aquesta tècnica és més lenta que la *rasterització*, però és superior en fotorrealisme.

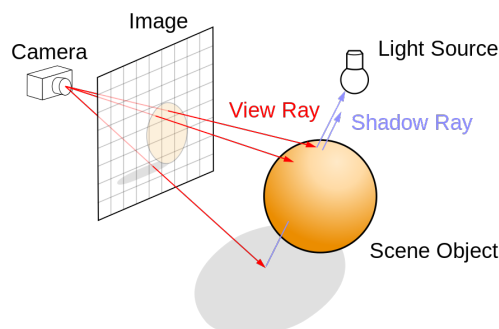


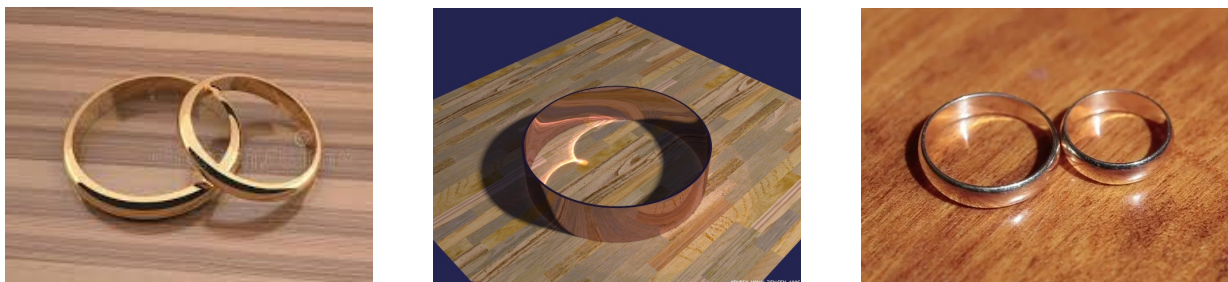
Figura 3: Algoritme de Ray Tracing. Font:[3]

No obstant una renderització anomenada fotorrealista, no té per què ser realista. Hi ha moltes renderitzacions que poden semblar tretes de la realitat, però en veritat no ho són. Per exemple, l'absència de *càustiques*, és una característica que mostra clarament la diferència d'una renderització físicament acurada i una fotorrealista, com mostra la Figura 4.

1. Contextualització

En la **renderització fotorrealista**(Figura 4a) tot i que la nostra percepció pot pensar que és realista, però com es pot contrastar, no s'han reproduït les càustiques que sí apareixerien en una fotografia (Figura 4c), que representa la realitat. Per tant, no s'ha realitzat volent imitar la realitat, sinó amb objectius artístics on es busca obtenir una imatge amb una estètica concreta que sembli visualment realista.

En canvi en la **renderització físicament acurada** (Figura 4b) les llibertats artístiques que permeten ometre efectes d'il·luminació indirecta, càustiques,... queden anul·lades, ja que aquest tipus de renderització segueix les lleis de la física. La renderització físicament acurada intenta representar la realitat, és a dir, ens permet obtenir imatges predictives que emulen la realitat.



(a) Rendering fotorrealista

(b) Rendering físic

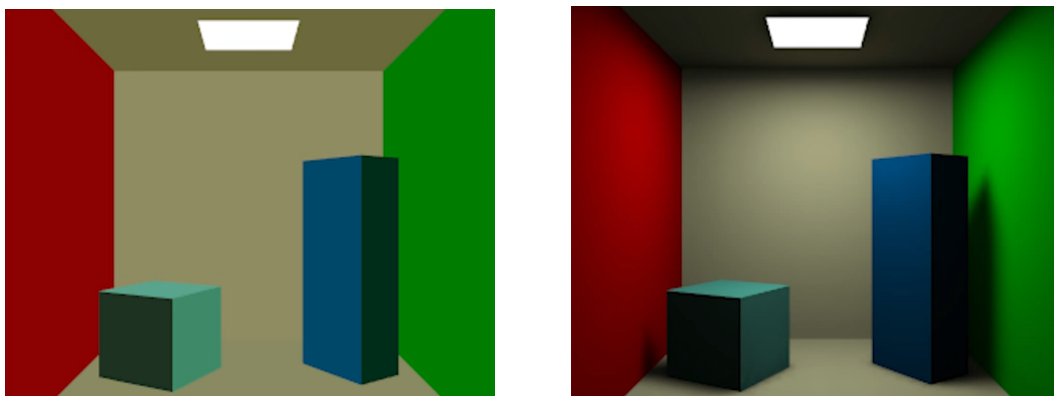
(c) Fotografia

Figura 4: Diferència entre els renderings i la realitat. Fonts: (a)[4], (b)[5], (c)[6]

Per altra banda, no podem oblidar la importància que té la il·luminació en aquest projecte. Per aquest motiu definiré alguns conceptes elementals sobre aquesta àrea:

Il·luminació global: no és només la il·luminació que prové directament de la font de llum, sinó també la indirecta, és a dir, la llum que prové de la mateixa font però que ha estat propagada a través de reflexions sobre les superfícies de l'escena.

Per tant, una escena sense il·luminació global, no implica una escena sense il·luminació (uniformement negra), sinó una com la Figura 5a; encara que en aquest cas sense la llum ambiental afegida al render, l'escena apareixeria totalment negra. Per altra banda, en la Figura 5b es pot veure un ombrejat natural a causa de la interacció de la il·luminació amb els objectes de l'escena. De fet, el *Ray Tracing* és un dels principals algorismes d'il·luminació global.



(a) Renderització sense il·luminació global

(b) Renderització amb il·luminació global

Figura 5: Diferència amb i sense il·luminació global. Font: [7]

1. Contextualització

Les magnituds fonamentals d'il·luminació, seran utilitzades en la secció d'anàlisi, quan les diferents renderitzacions de Sant Quirze de Pedret, es trobin definitives i s'hagi computat el càlcul d'irradiància.

La Taula 1 representa els aspectes més significatius de les magnituds fonamentals de luminotècnia amb les magnituds radiomètriques corresponents, i la Figura 6 projecta les definicions.

Magnitud fotomètrica	Magnitud radiomètrica	Símbol	Unitat
Flux lluminós	Flux radiant	ϕ	lumen (lm)
Intensitat lluminosa	Intensitat radiant	I	candela (cd)
Il·luminància	Irradiància	E	lumen/m ² (lux)
Luminància	Radiància	L	candela/m ² (cd/m ²)

Taula 1: Magnituds fotomètriques i radiomètriques. Font: elaboració pròpia

Flux lluminós: Quantitat de llum emesa per una font en totes direccions.

Intensitat lluminosa: Quantitat de llum emesa per una font en una direcció donada.

Il·luminància: Quantitat de llum que incideix sobre la unitat de superfície.

Luminància: Intensitat lluminosa per unitat de superfície aparent d'una font que emet llum o que la reflexa, captada per l'ull humà.

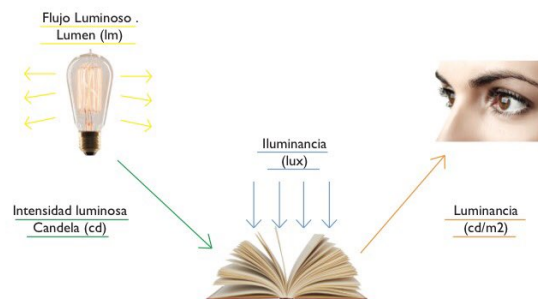


Figura 6: Magnituds fotomètriques. Font:[8]

1.3 DESCRIPCIÓ DEL PROBLEMA A RESOLDRE

L'obstacle que ens trobem és la renderització, hi ha moltes eines fotorrealistes i molt poques predictives. Nosaltres podríem haver invertit temps en crear una il·luminació realista que aconseguís una imatge fotorrealista, però com ens asseguràrem que aquesta imatge generada fos fidel a la realitat d'aquella època?

Per aquest motiu s'utilitza un motor de renderització físicament acurat, que ens permeti predir com era el passat, perquè la gent que estudia patrimonis culturals pugui simular les condicions del passat de manera acurada i científicament validada.

A diferència dels renders fotorrealistes que tenen una interfície molt completa i una gran comunitat com Blender, Maya, 3DStudio Max, AutoCAD, etc.; la majoria de motors no fotorrealistes tenen una comunitat pobre a causa de la poca popularitat que tenen. Per tant, la informació és escassa sobre aquests motors i el temps d'aprenentatge s'incrementa notablement, sobretot per a la gent no familiaritzada amb programes de renderització realista, o conceptes de fotografia i il·luminació.

En aquest projecte investigarem com integrar el motor de renderització extern que hem triat a una eina de modelatge. Això implica fer una transformació de les dades de l'escena utilitzades en les aplicacions de gràfics 3D, al format propi del motor de renderització. A banda d'això, necessitarem crear un sistema de comunicació entre l'eina de modelatge i el motor de renderització, i una interfície gràfica

1. Contextualització

per permetre a l'usuari ajustar els paràmetres necessaris.

D'aquesta manera crearem una eina totalment interactiva, que permeti a l'usuari simular diferents condicions d'il·luminació i fer renderitzacions físicament acurades, integrant a l'aplicació de gràfics 3D existent una extensió o plugin que actuï com a una interfície entre ells.

Per tal d'avaluar la connexió del plugin a l'aplicació de gràfics 3D, haurem d'analitzar i estudiar les imatges que haurem obtingut, demostrant la seva validesa a través de la física.

1.4 ACTORS IMPLICATS

Els *stakeholders* (parts interessades) del nostre projecte, es classifiquen en dos grups: interns i externs.

Els interns són els que tenen influència directa amb el TFG: el director Imanol, el codirector Carlos i l'autora Míriam.

- El director i el codirector són encarregats de supervisar i guiar el projecte procurant el seu correcte desenvolupament, i verificant que s'adeqüi als objectius del grup de recerca ViRVIG, al qual pertanyen els dos.
- L'autora serà la responsable de planificar, investigar, desenvolupar, experimentar i documentar el projecte.

Els externs, no actuen en el TFG però són beneficiaris dels seus resultats:

- El grup de recerca ViRVIG, en particular l'equip que es troba darrere el projecte EHEM, al qual va dirigit el producte. Desitgen aconseguir que l'eina formi part del desenvolupament del seu tercer objectiu [2].
- Gent de patrimoni cultural, els principals usuaris als quals va dirigit, especialment, els historiadors de l'art que duen a terme l'estudi de Sant Quirze de Pedret. Actualment, volen una eina científica que els permeti simular diferents escenes hipotètiques d'il·luminació, a fi de trobar explicacions i realitzar simulacions.
- Professionals interessats en la renderització predictiva: arquitectes, interioristes, *Art director lighting designer*, responsables de projectes lumínics, *lighting designer*, etc.
- Investigadors o desenvolupadors, es podrà accedir al codi de l'eina per a poder crear extensions o modificacions d'aquesta. Per tant, el codi hauria d'intentar ser de qualitat.
- Societat, gràcies a l'obtenció d'imatges acurades, es podrà aprofundir en els coneixements de la història i el món de l'art.

2 JUSTIFICACIÓ

2.1 ESTAT DE L'ART

Els gràfics 3D s'han enfocat molt en les renderitzacions fotorrealistes, ja que la seva utilització és molt profitosa en diferents camps: des de sectors artístics com la cinematografia, l'animació, els videojocs, VFX (Figura 7b); fins a sectors industrials com l'automobilístic, dispositius electrònics i productes materials (Figura 7a), com també en visualitzacions arquitectòniques i interiorisme (Figura 7c). Però no només han estat utilitzats per empreses per promocionar els seus productes, sinó també en sectors científics com ara la medicina, la bioinformàtica i l'arqueologia (Figura 7d).



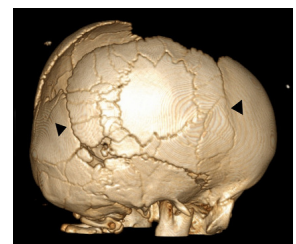
(a) Personalització del Tesla



(b) Star Wars - Yoda



(c) Disseny d'interior



(d) Crani - reconstrucció

Figura 7: Renderitzacions contemporànies. Fonts: (a)[9], (b)[10], (c)[11] i (d)[12]

La manca de sistemes de renderització predictius oberts, ha fet que moltes empreses i institucions s'hagin hagut de desenvolupar els seus propis sistemes de renderització predictius, i moltes d'aquestes han decidit mantenir el software de manera privada. No obstant, en la Taula 2 es mostra de manera visual l'estudi que es va fer per triar l'eina que simulés la il·luminació.

	Radiance	AGI 32	DiaLux	ReLux
Desenvolupadors	Greg Ward (Laboratoris de Berkeley Lawrence)	Lighting Analysts, Inc.	DIAL GmbH Lüdenscheid	RELUX Informatik AG
Llicència	Software lliure	Proprietary EULA	Software lliure	ReluxThirdParty
Gratuït	Sí	No	Sí	Sí
GUI	No (CLI)	Sí	Sí	Sí
Sistemes operatius	Windows, Unix, Linux, OS X	Windows	Windows	Windows
Corba de coneixement	Alta	Mitjana	Mitjana	Baixa
Algoritme	Ray Tracing	Radiosity	Ray Tracing	Ray Tracing
Gama de colors	amplia	limitada	amplia	limitada
Gama de textures	amplia	amplia	limitada	amplia
Temps de comput [13]				
Il·luminació natural	193s	24s	70s	10s
Il·luminació natural i artificial	225s	35s	90s	10s
Il·luminació artificial	53s	10s	8s	5s
Precisió del càlcul [13]				
Il·luminació natural	94%	91%	44%	87%
Il·luminació natural i artificial	93%	90%	49%	90%
Il·luminació artificial	86,3%	82,5%	92%	88%

Taula 2: Comparativa de simuladors d'il·luminació predictius. Font: Elaboració pròpia

2. Justificació

2.2 SOLUCIÓ PRESA

Radiance [14] va ser l'opció triada, ja que guanya la partida a la resta en els aspectes fonamentals per a nosaltres: la seva precisió en les simulacions, és a dir, la desviació mínima en la predicció de la realitat. A més a més, de l'extensa gama de colors i textures que ens permetrà definir materials complexos de l'època pre-romànica i la seva llicència oberta.

Per altra banda, utilitza *Ray Tracing* un aspecte important a tenir en compte, ja que permet simular reflexions especulars. En el cas de què l'algorisme hagués estat Radiosity, només podríem gestionar reflexions difuses, que qualitativament serien menys fidels a la realitat.

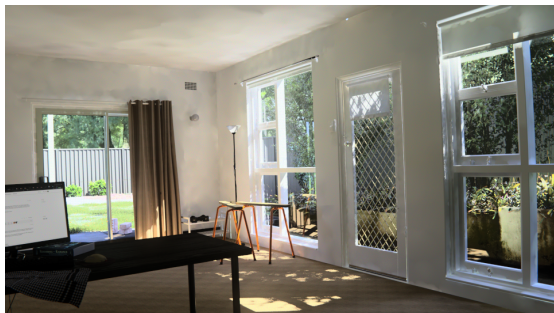
Radiance implementa un tipus de *Ray Tracing* anomenat *backward* que consisteix en què els raigs de llum són traçats des de l'observador fins a la font de llum tenint en compte totes les interaccions físiques (reflexió, refracció) entre els objectes que componen l'escena.

No obstant, a causa de què els càlculs s'inicien des de la posició de l'observador, s'ha de fer un càlcul completament nou cada cop que es canvia la seva localització. Per tant, el fet de recórrer una escena incrementaria el cost computacional, que ja és bastant elevat a causa de la precisió i no estalviar temps de càlcul aplicant simplificacions.

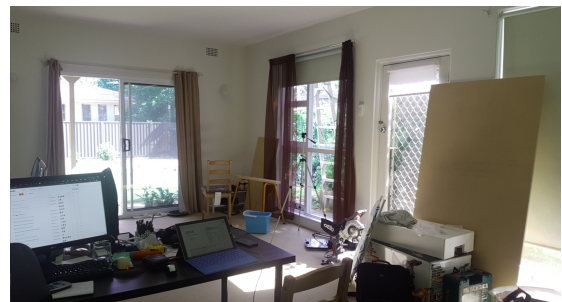
De fet, de tots els renderitzadors és el que té major temps de comput, podent arribar a superar l'hora. Tot i així és un temps que ens podem permetre, ja que no és res comparat amb alguns entrenaments de models de machine learning. Per aquest motiu, he decidit que Radiance serà l'element clau pel desenvolupament d'aquest projecte.

En resum, Radiance és únic en la descripció de la llum emesa, reflectida i refractada per a cada superfície, ja que no només considera la radiositat (radiància de sortida) com molts programes de visualització d'il·luminació, sinó inclou també en els mètodes reflexió especular i direccional-difusa. Ara bé, el més important és que el model d'il·luminació local tingui una simulació acurada en l'emissió de les fonts de llum, perquè si no es fa correctament de res servirà el resultat.

La llibreria Radiance ha estat validada numèricament i, com és lògic, la seva correcció s'observa en comparar els seus resultats amb fotografies equivalents. Existeix un article de BlenderNation [15], que documenta com crear renderitzacions a un nivell de detall qualitatiu alt, demostrant la capacitat de predicció de manera fiable simplement mesurant mostres d'objectes.



(a) Renderització amb Radiance



(b) Fotografia

Figura 8: Comparació de la sala d'estar renderitzada amb Radiance i fotografiada. Font: [15]

2. Justificació

Com mostra la Figura 8 de l'estudi [15], les diferències d'il·luminació són mínimes entre la fotografia i la renderització amb Radiance, tenint en compte que no es troben tots els models i la localització no és la mateixa que l'exposada en la fotografia. Amb tot i això, aquestes dues diferències tenen fàcil solució, ja que sempre es poden incorporar més models i la posició de la càmera es pot triar en Radiance.

En el nostre cas, per a poder utilitzar Radiance, no només haurem de configurar l'escena (geometria, materials i il·luminació), sinó també haurem d'incorporar el component temporal, per poder produir imatges en qualsevol època de l'any (data i hora). Així com les propietats tant històriques com físiques, de la il·luminació de làmpades d'oli i espelmes, per a oferir imatges plausibles amb una gran diversitat de possibles simulacions.

Tot i que pot ser considerada una llibreria, Radiance és una pura col·lecció d'executables especialment pensats per l'entorn UNIX. Degut a això, molts stakeholders no la utilitzarien per la seva poca usabilitat. Per tal de reduir el forat existent que Radiance té entre usabilitat i qualitat, utilitzarem una eina de modelatge existent que ens permeti exportar les dades de l'escena 3D, i transformar-les en el format propi de Radiance per a poder generar la renderització.

Blender [16] serà l'eina de modelatge triada, encara que existeixen moltes eines de modelatge que permeten l'exportació de dades de l'escena de modelatge (CAD, Maya, ZBrush, 3ds Max, ...) , vam decidir utilitzar Blender, perquè és un programa informàtic multiplataforma de codi obert i molt popular, tant en la docència, com en la recerca i en la producció. A més a més, de comptar amb una gran i activa comunitat al darrere.

A més a més, la seva extensió és senzilla, ja que utilitza **Python** com a llenguatge principal per a crear scripts, plugins i extreure informació de l'escena a través de la seva consola. Per últim, degut a què és una eina que s'utilitza en la docència dels cursos de gràfics de la FIB, ja m'hi trobo familiaritzada. Un fet important a tenir en compte, ja que Radiance té una corba d'aprenentatge molt pronunciada.

Aquest context m'obligarà a implementar una solució nova, ja que no conec cap eina que permeti realitzar renderitzacions físicament acurades de manera interactiva i amb una interfície que sigui usable, sobretot en entorns de patrimoni cultural.

3 ABAST

3.1 OBJECTIUS I SUB-OBJECTIUS

L'objectiu principal d'aquest projecte és desenvolupar una eina que combini la renderització físicament acurada (Radiance) i una eina de modelatge 3D (Blender) pel cas d'estudi, l'església preromànica Sant Quirze de Pedret.

Dit d'una altra manera, interactuar amb els diferents factors d'il·luminació com la posició del Sol (segons la data i l'hora), el comportament de la llum en funció dels materials a les finestres (vidre i cuir) i els tipus d'il·luminacions artificials emprats (làmpades d'oli i espelmes) .

Els dos programes, Blender i Radiance, seran combinats en la realització de l'eina d'aquest TFG, utilitzant scripts de Python com a mitjà d'unificació.

Per abordar aquestes fites es van establir els següents objectius i subobjectius:

1. Familiarització amb Radiance i els conceptes d'il·luminació.
 - (a) Definir tots els materials dels models en unitats físiques.
 - (b) Definir els tipus d'il·luminació artificial (làmpades d'oli i espelmes) en el IES
 - (c) Definir la il·luminació global segons la posició del Sol a través del format CIE Sky Model
 - (d) Simular la il·luminació en funció de les propietats específiques de les finestres,.
 - (e) Crear diferents renderitzacions validades de Sant Quirze de Pedret, des de diversos punts d'orientació, localitzacions i jugant amb les diferents funcionalitats.
2. Unificació de Blender i Radiance: Combinar els dos sistemes a través de l'API de Python incorporada a Blender.
 - (a) Conversió de les dades de Blender a Radiance: conversió manual dels formats Blender a Radiance.
 - (b) Definició automàtica dels materials i il·luminació a partir dels objectes de l'escena de Blender.
 - (c) Interacció amb la posició de l'observador.
 - (d) Crear una interfície (add-on) per a facilitar la interacció entre l'eina i l'usuari
 - (e) Crear diferents renderitzacions validades d'altres models també preromànics, de manera senzilla i intuïtiva.
3. Seguir la metodologia *Agile* correctament: utilitzar el programari per a la gestió del *backlog* i la actualització de l'estat de les tasques.

3.2 REQUERIMENTS (FUNCIONALS I NO FUNCIONALS)

D'altra banda, es tindran en compte els següents requisits per garantir la qualitat del projecte:

Requisits funcionals

3. Abast

- Obtenció d'imatges físicament acurades i qualitatives.
- Simular la il·luminació natural del Sol segons l'any, el dia i l'hora.
- Simular els diferents tipus d'il·luminació artificial: espelmes i làmpades d'oli.
- Simular el comportament de la llum en funció dels materials de les finestres: vidre i cuir.

Requisits no funcionals

- **Usabilitat:** L'eina serà intuïtiva i fàcil d'utilitzar per a qualsevol tipus d'usuari, i possiblement sense la necessitat d'haver tingut coneixements previs de Radiance.
- **Precisió i exactitud:** La simulació serà realitzada a partir de dades quantitatives que seran convertides en dades fotomètriques.
- **Adaptabilitat:** L'eina podrà ajustar-se a qualsevol model de patrimoni cultural.
- **Reusabilitat:** Flexible, modular i fàcil d'integrar en diferents projectes.

3.3 OBSTACLES I RISCOS

Tot projecte té diferents dificultats que han de ser analitzades anticipadament per minimitzar el seu impacte.

- **Termini del projecte:** Existeix una data límit per a l'entrega del projecte que condiona el desenvolupament, per la qual cosa és necessari disposar d'una bona planificació per acabar el projecte a temps.
- **Planificació temporal errònia:** Una mala planificació del projecte, podria provocar una mala eficiència en el seu desenvolupament i seria adient replanificar o modificar les tasques considerant les prioritats del projecte.
- **La inexperiència en el camp d'estudi:** El fet d'utilitzar tecnologies de les quals el desenvolupador no n'és expert fa que el temps d'aprenentatge sigui més alt i per tant pugui allargar-se el temps necessari per al desenvolupament.
- **Corba d'aprenentatge de Radiance:** L'escassa informació de la llibreria i la poca popularitat, comportarà dedicar molt de temps a Radiance, produint-se una reducció important en temps, al desenvolupament del projecte.
- **Pèrdua de dades:** Existeix el risc de perdre la feina realitzada, per la qual cosa, s'haurà d'anar guardant el treball fet en el núvol i un dispositiu d'emmagatzematge físic.
- **Bugs i altres dificultats de programació** Un bug en el codi podria afectar de manera inesperada al resultat. És per això, que es programarà codi llegible de la forma més simple possible.
- **Recursos computacionals:** La renderització i el seu cost computacional consumeixen gran part de la RAM, que creix amb el nombre de superfícies i el nombre de fonts de llum en l'escena.

4 METODOLOGIA

El desenvolupament del projecte es farà utilitzant la metodologia *Agile*, concretament *Scrum*[17], ja que enfoca el desenvolupament del projecte de forma iterativa i incremental. S'han realitzat algunes modificacions: els daily scrum i el rol de Scrum Master han estat eliminats, ja que en ser un equip de tres persones no té sentit i d'aquesta manera la desenvolupadora pot anar validant-se i verificant-se ella mateixa els requisits realitzats durant l'Sprint. El paper de Product Owner serà realitzat per l'Imanol Muñoz i el Carlos Andujar i el de Desenvolupadora per la Míriam Méndez.

Per a la planificació del projecte, s'ha realitzat un Sprint per setmana, és a dir, cada setmana es reunia l'estudiant, el director i el codirector, per a comprovar l'estat del projecte, a excepció del mes d'agost que es va continuar de manera autònoma i es va aprofitar per a redactar.

Es va fer una planificació inicial, però no es va poder seguir, ja que al ser un projecte amb un component de recerca hi ha hagut canvis en els requeriments, mentre s'anava avançant en el projecte i per aquest motiu hi ha hagut alguns canvis en les tasques (modificacions, addicions i subtraccions).

D'altra banda, també es van fer canvis temporals, ja que durant el quadrimestre em vaig sotmetre a una intervenció quirúrgica i sumat a la càrrega lectiva de la facultat, va significar una postergació en el treball.

Finalment aquestes tasques es van anar decidint a mesura que avançava el projecte. En les primeres etapes, es va dedicar temps a GEP i a estudiar Radiance, en les següents etapes es va començar a realitzar la integració de Radiance amb Blender, a través d'add-on (plugin), generant diferents prototips capaços de treure el màxim partit a Radiance i també ser una GUI amigable.

Durant el Procés de desenvolupament del projecte s'han utilitzat tres tipus d'eines: les de seguiment, les de desenvolupament i les de validació.

EINES DE SEGUIMENT (eines utilitzades tant per a la planificació com l'avaluació del projecte):

Google Meet, ha estat el software principal pel seguiment del projecte, ja que ens permetia reunir-nos de forma síncrona i resoldre els problemes que anaven sorgint. En cada reunió s'actualitzava l'estat de les tasques en el **Trello**, es creaven noves tasques i es verificava el trasllat de tasques de To Review a Done. (Hem estructurat 5 estats: forecast, to-do, in-progress, to-review i done). Abans de finalitzar la trucada es creava un esdeveniment en **Google Calendar**, per a posar data i hora a la pròxima reunió.

Altrament, també s'han utilitzat recursos com **Google Drive** per emmagatzemar i compartir arxius, com per exemple: models 3D, textures... **Gmail** per contactar de manera puntual tant amb el director com amb el codirector, **Overleaf** per documentar la memòria en format LaTeX en línia, permetent accedir al director i el codirector, perquè poguessin estar al corrent del contingut i **GanttProject** per a fer la planificació del projecte amb el diagrama de Gantt.

4. Metodologia

EINES DE DESENVOLUPAMENT (eines utilitzades durant el procés de construcció del software):

Blender v3.0.1, **Radiance v.5.4** i **Python 3.10.6**, seran les imprescindibles per desenvolupar el projecte, juntament amb l'editor de text **Visual Studio Code** i la consola de Linux.

Així mateix, també s'utilitzarà **Git** per controlar les versions del nostre codi i **GitHub** per a pujar el nostre projecte en el núvol, d'aquesta manera ens assegurem que en cas de voler recuperar una versió anterior de codi, es pugui tornar enrere en l'historial de `commits`. El codi estarà organitzat en dues branques: *feature* i *main*. La primera serà una branca amb el contingut en desenvolupament, mentre que *main* contindrà la versió estable del projecte. El progrés realitzat en la branca *feature* serà traslladat a la branca *main*, segons sigui necessari.

EINES DE VALIDACIÓ (eines que verifiquen que tot sigui correcte i proporcionen *feedback* en cas contrari):

Les reunions no només són de seguiment, sinó també de verificació, ja que es fa una validació del correcte desenvolupament de les tasques de l'Sprint, tot utilitzant **Google Meet**.

Respecte al marc avaluatiu oficial, s'ha utilitzat **Atenea** per entregar els lliuraments de GEP i **el Racó** per a trametre l'informe de seguiment i la memòria final. D'aquesta manera hem disposat de dos entorns validadors, ja que existeix una nota i un *feedback* posterior a l'enviament.

5 PLANIFICACIÓ TEMPORAL

El TFG va ser **iniciat el 2 de febrer del 2021**, on es va tenir el primer contacte amb el director, el codirector i es va parlar sobre el tema que anàvem a tractar. Segons la normativa[18], hauria d'haver estat finalitzat el **20 de juny**, una setmana abans que comencessin els torns de lectura, data que s'havia tingut en compte per a la planificació inicial. No obstant, es va sol·licitar una extensió, de manera que es va ampliar el temps i la data de finalització del TFG serà el **8 d'octubre**.

Entremig d'aquest període, a part de la intervenció quirúrgica, també es van cursar les últimes assignatures de la carrera d'informàtica i començar el màster. A més a més, vaig cursar l'avaluació continuada de GEP, on vaig obtenir *feedback* de tota la documentació inicial del TFG. De manera que vaig dedicar **4 setmanes a la gestió del projecte** (duració de GEP), en el cas de no tenir en compte l'extensió temporal haguessin estat **12 setmanes per al desenvolupament**, començant a comptar a partir del 21 de març, moment en el qual va finalitzar el curs de GEP. En canvi, si es té en compte l'extensió, passarien a ser **27 setmanes per al desenvolupament**.

Actualment, en el grau d'enginyeria informàtica el TFG té un pes de 18 ECTS (3 crèdits de GEP + 15 crèdits del mateix projecte). Segons fonts de la UPC [19], cada crèdit equival a 25h, però excepcionalment pel TFG pot arribar a igualar les 30h. En la planificació inicial es van considerar el mínim d'hores, ja que en ser una estimació preferia que faltessin hores en les tasques, i estendre el temps que necessités per finalitzar-les, així assegurar-me que sempre hi hauria alguna cosa a fer.

Per tant, el nombre d'hores dedicades havien d'haver estat:

$$\text{hores de GEP} = 3\text{ECTS} \cdot \frac{25h}{1\text{ECTS}} = 75h \quad (1)$$

$$\text{hores de desenvolupament} = 15\text{ECTS} \cdot \frac{25h}{1\text{ECTS}} = 375h \quad (2)$$

Aquestes hores calculades es van repartir al llarg de les setmanes en funció de les tasques, per tal de crear la planificació inicial del projecte, que havia de ser contrastada amb la final.

5.1 DESCRIPCIÓ DE LES TASQUES

A l'inici del projecte seguíem la metodologia Agile/Scrum, per tant, no tenim tasques definides, ja que el *Product Backlog* no es troba desgranat a nivell de tasca, sinó a nivell d'història d'usuari, és a dir, es feia difícil detallar les tasques, ja que l'*Sprint planning* es fa cada setmana i a més, el desconeixement i la falta d'experiència amb les principals eines que toca aquest projecte: Blender i Radiance, dificultava saber exactament quines eren les tasques i la seva estimació.

Per aquest motiu, era molt probable que les tasques referents al desenvolupament del projecte no fossin les definitives. És més, moltes tasques van patir canvis (modificacions, agregacions i/o eliminacions) mentre s'anaven realitzant Sprints. Ara bé, la planificació inicial em va servir per tenir una idea més clara i concreta del que havia de desenvolupar, ja que per fer les estimacions temporals m'havia de plantejar com ho havia de desenvolupar, per a poder fer una aproximació del temps que em requeriria completar la tasca.

5. Planificació temporal

Les tasques van ser dividides i subdividides segons el seu tipus: Gestió del Projecte (GP), Seguiment i Documentació (SD), Desenvolupament amb Radiance(DR), Desenvolupament amb Blender(DB), Desenvolupament amb Radiance i Blender(DRB) i Proves de Validació (PV).

5.1.1 TASQUES DE GESTIÓ DEL PROJECTE

- **GP1 ⇒ Contextualització i abast (26h):** Redacció de la primera entrega de GEP. Es documenta i es determina el context, la justificació, l'abast i la metodologia. No hi ha dependències
- **GP2 ⇒ Planificació temporal (12h):** Redacció de la segona entrega de GEP. Es documenta i es determinen les tasques, estimacions i Gantt, i Gestió de risc: Plans alternatius i obstacles. Dependències: GP1 < GP2
- **GP3 ⇒ Pressupost i sostenibilitat(13h):** Redacció de la tercera entrega de GEP. Es documenta i es determina la identificació i l'estimació de costos, el control de gestió i l'informe de sostenibilitat. Dependències: GP2 < GP3
- **GP4 ⇒ Documentació de la fase inicial (24h):** Integració dels tres lliuraments anteriors, fent les modificacions, extensions i rectificacions oportunes. Dependències: GP3 < GP4

Respecte la planificació final, cap d'aquestes tasques ha patit desviament i modificació, per tant, és va fer una bona estimació.

5.1.2 TASQUES DE DESENVOLUPAMENT

Seguiment i Documentació

- **SD1 ⇒ Sprint review (1h/setmana):** Cada dimecres l'equip es reuneix per comprovar l'estat de les tasques assignades a l'sprint. Aquestes tasques es generaven cada setmana a partir de les històries del Product Backlog. Un cop s'assignaven al Sprint Backlog, totes les tasques havien de trobar-se finalitzades en acabar la setmana, almenys de manera simple per a poder ser arxivades i començar un nou Sprint. No hi ha dependències.
- **SD2 ⇒ Documentació de l'Sprint (1h30min/setmana):** En finalitzar l'Sprint, es documenta en la memòria un esbós de l'avançament en el projecte, i s'afeixen les referències útils per a facilitar la posterior redacció. Dependències: SD1 < SD2 (paral·lela)
- **SD3 ⇒ Documentació fase final (56h):** Es va redactant de manera correcta, clara i concisa allò que ha estat redactat en SD2 i la resta del treball. Dependències: SD2 < SD3
- **SD4 ⇒ Preparació de la presentació final (14h):** Preparació de la presentació del treball. Dependències: SD3 < SD4

Respecte a la planificació final, la primera, la segona i la tercera, no són del tot certes:

5. Planificació temporal

SD1 ⇒ La duració de les reunions ha estat bastant variable, podent durar entre 1 hora o un quart d'hora. La idea central de les reunions era mostrar el que s'anava fent i aclarir dubtes si hi havia algun. En efecte, al principi les reunions eren més llargues en haver d'encaminar el projecte.

SD2 ⇒ A la documentació es va dedicar 30min, per apuntar els punts claus i inserir imatges. Però la major part ha estat eliminada per la memòria final, ja que vaig decidir documentar-la com si fos un tutorial, i prescindir d'alguns resultats erronis que únicament complicarien la lectura i desorientaríem al lector.

SD3 ⇒ En la redacció de la documentació final s'han dedicat 60h, ja que a part de documentar, també s'ha invertit bastant temps en la reparació de bugs i en la compill·lació del LaTeX.

Desenvolupament amb Radiance

- **DR1** ⇒ **Instal·lació de l'entorn Radiance (1h)**: S'instal·len correctament tots els executables en Linux Mint. No hi ha dependències
- **DR2** ⇒ **Cercar documentació i tutorials (1h)**: S'investiguen totes aquelles fonts que poden ser útils per al nostre treball. Dependències DR1 < DR2
- **DR3** ⇒ **Execució de tests i seguiment de tutorials (8h)**: S'explora Radiance i el seu funcionament seguint tutorials i executant testos. Dependències: DR2 < DR3
- **DR4** ⇒ **Reproducció d'un cas base amb Radiance (8h)**: Es reproduceix un cas d'estudi ja fet, en Radiance de forma manual; on tots els materials, llums i escena ja estan definits. Dependències: DR3 < DR4
- **DR5** ⇒ **Familiarització en la definició de materials, llums i escena (7h)**: S'aprèn a través d'exemples i pàgines a crear les nostres pròpies definicions, per a reproduir les nostres pròpies escenes. Dependències: DR4 < DR5
- **DR6** ⇒ **Definició de materials i textures (16h)**: Amb diferents models es fan proves per finalment definir Sant Quirze de Pedret. Dependències: DR5 < DR6
- **DR7** ⇒ **Calibratge de textures (4h)**: S'utilitza `macbethcal` per tenir en compte detalls com les variacions de la superfície i els colors dels objectes. Dependències: DR6 < DR7
- **DR8** ⇒ **Definició de la il·luminació natural(8h)**: Serà creada amb `genskye`, que generarà un CIE Sky Model físicament acurat, a partir de la geolocalització i una determinada hora. Dependències: DR7 < DR8
- **DR9** ⇒ **Definició de la il·luminació artificial utilitzant IES (8h)**: S'hauran de trobar fitxers `.ies` d'espelmes i làmpades d'oli o fer un escaneig fotomètric per a poder-los definir. Dependències: DR8 < DR9
- **DR10** ⇒ **Calibratge de l'*environment map*, combinat amb la il·luminació del cel(4h)**: S'ajusten els paràmetres del CIE Sky Model. Dependències: DR9 < DR10

Respecte a la planificació final, la setena i la desena van ser eliminades, ja que no van ser necessàries.

5. Planificació temporal

Pel que fa els temps estimats, van estar ben calculats. Malgrat això, en alguns casos jo he trigat més, perquè m'he trobat amb algunes dificultats externes de Radiance i algunes tasques les vaig complicar, quan en realitat eren més senzilles de realitzar. Però la tasca dos (cercar documnetació i tutorials) s'ha realitzat de manera paral·lela a la creació de l'escena en Radiance, per tant el temps ha estat molt major. Tanmateix, el comput d'una renderització en qualitat màxima pot tardar fins a sis hores. Però com es feien altres tasques de manera paral·lela a la generació d'imatges, no he comptat aquest temps.

Una tasca que falta aquí, és la definició dels paràmetres de renderització, per crear l'escena de Sant Quirze de Pedret en Radiance.

Desenvolupament amb Blender

- **DB1 ⇒ Familiarització amb l'entorn de Blender: materials, il·luminació i renderització (12h):** A través de video-tutorials i la documentació de Blender. No hi ha dependències
- **DB2 ⇒ Organització d'assets en l'escena (4h):** Estructurar l'escena i l'entorn per a poder-lo renderitzar físicament. Dependències: DB1 < DB2
- **DB3 ⇒ Modificació de material d'assets, si és pertinent (6h) :** Es comprovarà que tots els materials siguin vàlids per a crear la renderització amb Radiance, sinó serà modificat. Dependències: DB1 < DB3
- **DB4 ⇒ Introducció a l'API de Python (8h):** Aprendre a fer funcionar tant el Text Editor i la Python Console de Blender. Dependències: DB1 < DB4

Respecte a la planificació final, en la tercera tasca s'ha invertit més temps, ja que només el procés del baking va durar més de 6h (unes 12h). La resta van ser correctament estimades.

Desenvolupament amb Radiance i Blender

- **DRB1 ⇒ Desenvolupament bàsic de l'script d'automatització (64h):** S'especifiquen manualment les propietats dels models de Blender per a poder fer les següents conversions a Radiance. Dependències: DB4 < DRB1 i DR10 < DRB1
 - DRB1.1 ⇒ Conversió de models (16h).
 - DRB1.2 ⇒ Conversió de materials (16h).
 - DRB1.3 ⇒ Conversió de llums (16h).
 - DRB1.4 ⇒ Conversió de la càmera (16h).
- **DRB2 ⇒ Extensió avançada de l'script d'automatització (73h):** Les propietats seran introduïdes automàticament per l'script, a partir de l'escena creada en Blender. Per cada automatització s'ha creat una subtasca. Dependències: DRB1 < DRB2
 - DRB2.1 ⇒ Automatitzar materials (20h).
 - DRB2.2 ⇒ Automatitzar il·luminació artificial (24h).
 - DRB2.3 ⇒ Automatitzar la il·luminació natural (18h).
 - DRB2.4 ⇒ Incorporar trajectòries temporals del Sol (11h).

5. Planificació temporal

- **DRB3 ⇒ Conversió d'Script a Add-on (8h):** Transformació d'Script a Add-on, perquè sigui usable per l'usuari. Dependències: DRB2 < DRB3

Respecte la planificació final, cap d'aquestes tasques ha patit desviament ni modificació.

Proves de validació

- **PV ⇒ Proves de validació (24h):** Creació de testos de Radiance i Python per a validar el correcte desenvolupament de les tasques. Dependències: DR5 < PV

Respecte a la planificació final, no s'ha patit cap desviament ni modificació.

5.1.3 RECURSOS

Recursos humans: Els recursos humans implicats en les tasques són **l'estudiant ⇒ RH1** i tres experts: dos en el tema del treball de recerca (**el director ⇒ RH2** i **el codirector ⇒ RH3**) i un en la gestió de projectes (**el tutor de GEP ⇒ RH4**). Tots tres s'encarreguen de supervisar i guiar perquè el projecte de l'estudiant tingui èxit. A més a més, de les **persones externes ⇒ RH5** que han dedicat el seu temps a publicar informació en Internet que és útil per al nostre TFG.

Recursos materials: Els recursos materials utilitzats són:

- **RM1 ⇒ Overleaf:** Eina de redacció de LaTeX en línia, utilitzada per a la memòria.
- **RM2 ⇒ Google Meet i Google Calendar:** Eines utilitzades per a les reunions: Google Calendar per crear-la i Google Meet per a realitzar-la.
- **RM3 ⇒ Google Drive:** Eina per a compartir materials entre el director, codirector i l'estudiant.
- **RM4 ⇒ Atenea:** Eina de comunicació amb el tutor de GEP i utilitzada principalment per al lliurament d'entregues.
- **RM5 ⇒ Git amb GitHub:** Eines per gestionar el codi: Git per portar el control de les versions i GitHub per a emmagatzemar el codi en el núvol.
- **RM6 ⇒ Visual Studio Code:** IDE per a facilitar el desenvolupament del projecte.
- **RM7 ⇒ Python i llibreries diverses:** Llenguatge de programació principal per a la integració Radiance i Blender
- **RM8 ⇒ Radiance:** Llibreria per a crear renderitzacions físicament acurades.
- **RM9 ⇒ Blender:** Software utilitzat per a preparar l'escena.
- **RM10 ⇒ Trello:** Eina utilitzada per a regularitzar i gestionar les tasques.
- **RM11 ⇒ GanttProject:** Eina per produir el diagrama de Gantt.
- **RM12 ⇒ Ordinador:** Habilita l'ús de tots els recursos anteriors. Portàtil MSI PS42 amb 16GB DDR4 de RAM, processador i7 8550U, i targeta gràfica Nvidia GeForce MX150.

5. Planificació temporal

5.2 ESTIMACIONS I GANTT

La Taula 3 que es mostra a baix, mostra un resum de les tasques estimades de l'Apartat 5.1.

ID	Tasca	Temps estimat	Dependències estimades	Temps final	Dependències finals	Recursos
GP	Gestió de projecte	75h		75h		
GP.1	Contextualització i abast	26h		26h		RM1 ,RM2, RM4, RM12
GP.2	Planificació temporal	12h	GP.1	12h	GP.1	RM1 RM2, RM4, RM10, RM12
GP.3	Pressupost i sostenibilitat	13h	GP.2	13h	GP.2	RM1, RM2, RM4, RM12
GP.4	Documentació de la fase inicial	24h	GP.3	24h	GP.3	RM1, RM4, RM12
SD	Seguiment i documentació	110h		132h		
SD.1	Sprint Review	16h		24h		RM1, RM2, RM3, RM6, RM7, RM8, RM9, RM10, RM12
SD.2	Documentació de l'Sprint	24h	SD.1	20h	SD.1	RM1, RM12
SD.3	Documentació fase final	60h	SD.2	56h	SD.2	RM1, RM12
SD.4	Preparació de la presentació final	14h	SD.3	14h	SD.3	RM3, RM12
DR	Estudi i proves amb Radiance	65h		74h		
DR.1	Instal·lació de l'entorn Radiance	1h		1h		RM8, RM12
DR.2	Cercar documentació i tutorials	1h	DR.1	8h	DR.1	RM12
DR.3	Execució de tests i seguiment de tutorials	8h	DR.2	8h	DR.2	RM6, RM8, RM12
DR.4	Reproducció d'un cas base amb Radiance	8h	DR.3	8h	DR.3	RM6, RM8, RM12
DR.5	Familiarització en la definició de materials, llums i escena	7h	DR.4	7h	DR.4	RM6, RM8, RM9, RM12
DR.6	Definició de materials i textures	16h	DR.5	16h	DR.5	RM6, RM8, RM12
DR.7	Calibratge de textures	4h	DR.6	(eliminada)		RM6, RM8, RM12
DR.8	Definició de la il·luminació natural	8h	DR.8	8h	DR.7	RM6, RM8, RM12
DR.9	Definició de la il·luminació artificial utilitzant .ies	8h	DR.8	8h	DR.7	RM6, RM8, RM12
DR.10	Calibratge de l'environment map, combinat amb la il·luminació del cel	4h	DR.9	(eliminada)		RM6, RM8, RM12
DR.11	Definició dels paràmetres de renderització	(agregada)		8h	DR.5	RM6, RM8, RM12
DB	Estudi i proves amb Blender	30h		38h		
DB.1	Familiarització amb l'entorn de Blender	12h		12h		RM9, RM12
DB.2	Organització d'assets en l'escena	4h	DB.1	6h	DB.1	RM3, RM9, RM12
DB.3	Modificació de material d'assets, si és pertinent	6h	DB.1	12h	DB.1	RM9, RM12
DB.4	Introducció a l'API de Python	8h	DB.1	8h	DB.1	RM7, RM9, RM12
DBR	Implementació de la connexió de Blender i Radiance	146h		146h		
DBR.1	Desenvolupament bàsic del script d'automatització	64h	DB.4, DR.10	64h	DB.4, DR.10	RM7, RM8, RM9, RM12
DBR.1.1	Conversió de models	16h		16h		
DBR.1.2	Conversió de materials	16h		16h		
DBR.1.3	Conversió de llums	16h		16h		
DBR.1.4	Conversió de la càmera	16h		16h		
DBR.2	Extensió avançada de l'script d'automatització	74h	DBR.1	74h	DBR.1	RM7, RM8, RM9, RM12
DBR.2.1	Automatitzar materials	20h		20h		
DBR.2.2	Automatitzar il·luminació artificial	24h		24h		
DBR.2.3	Automatitzar il·luminació natural	18h		18h		
DBR.2.4	Incorporar trajectòries temporals del Sol	12h		12h		
DBR.3	Conversió d'Script a Add-on	8h	DRB.2	8h	DRB.2	RM7, RM9, RM12
PV	Proves de validació	24h	DR5	24h	DR5	RM7, RM8, RM9, RM12
Total		450h		489		

Taula 3: Resum de les tasques. Font: Elaboració pròpia

En la pàgina següent es mostra com queden distribuïdes les tasques estimades a través del diagrama de Gantt. Per després contrastar-la amb la distribució de les taques finals.

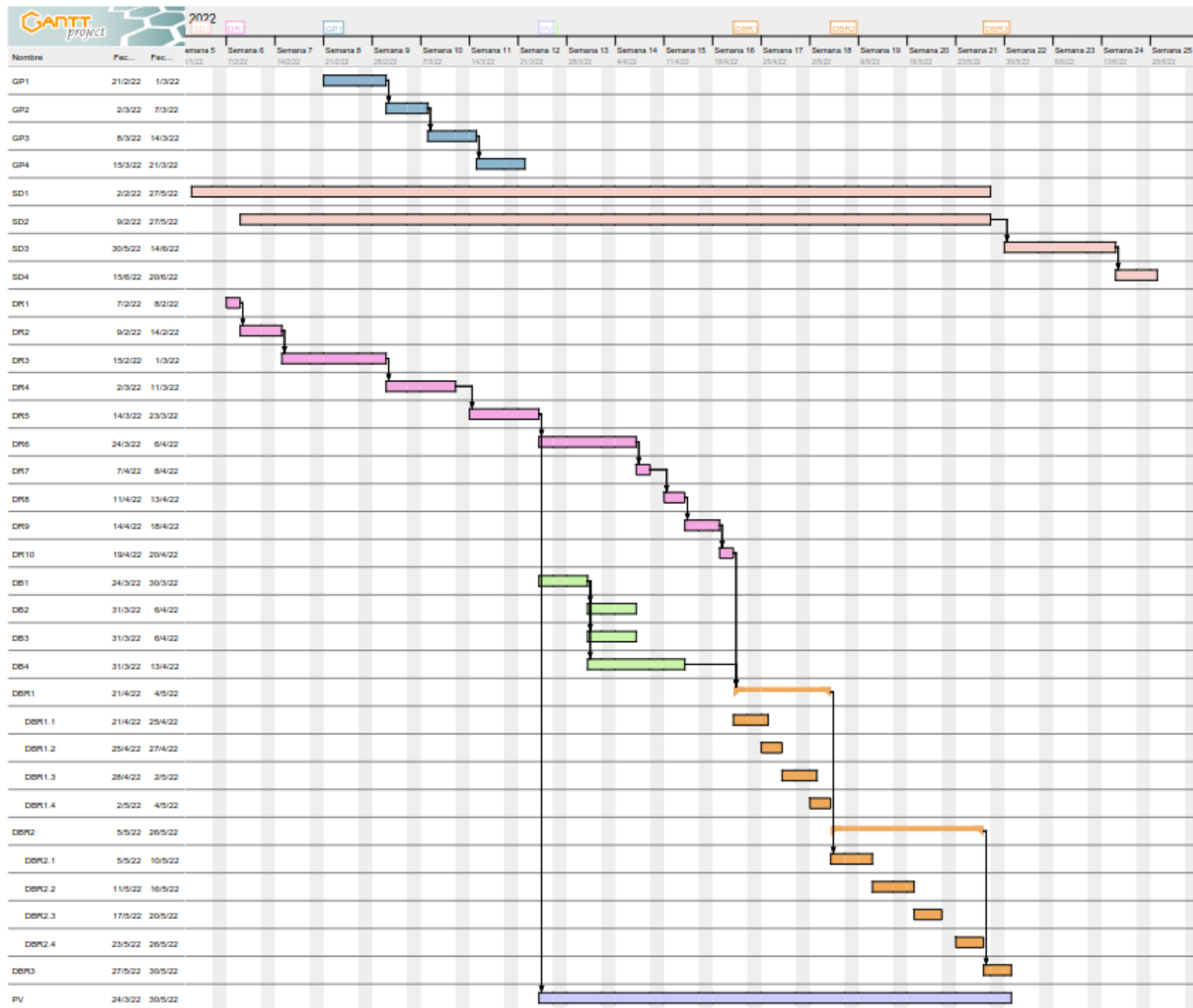


Figura 9: Diagrama de Gantt inicial. Font: elaboració pròpia

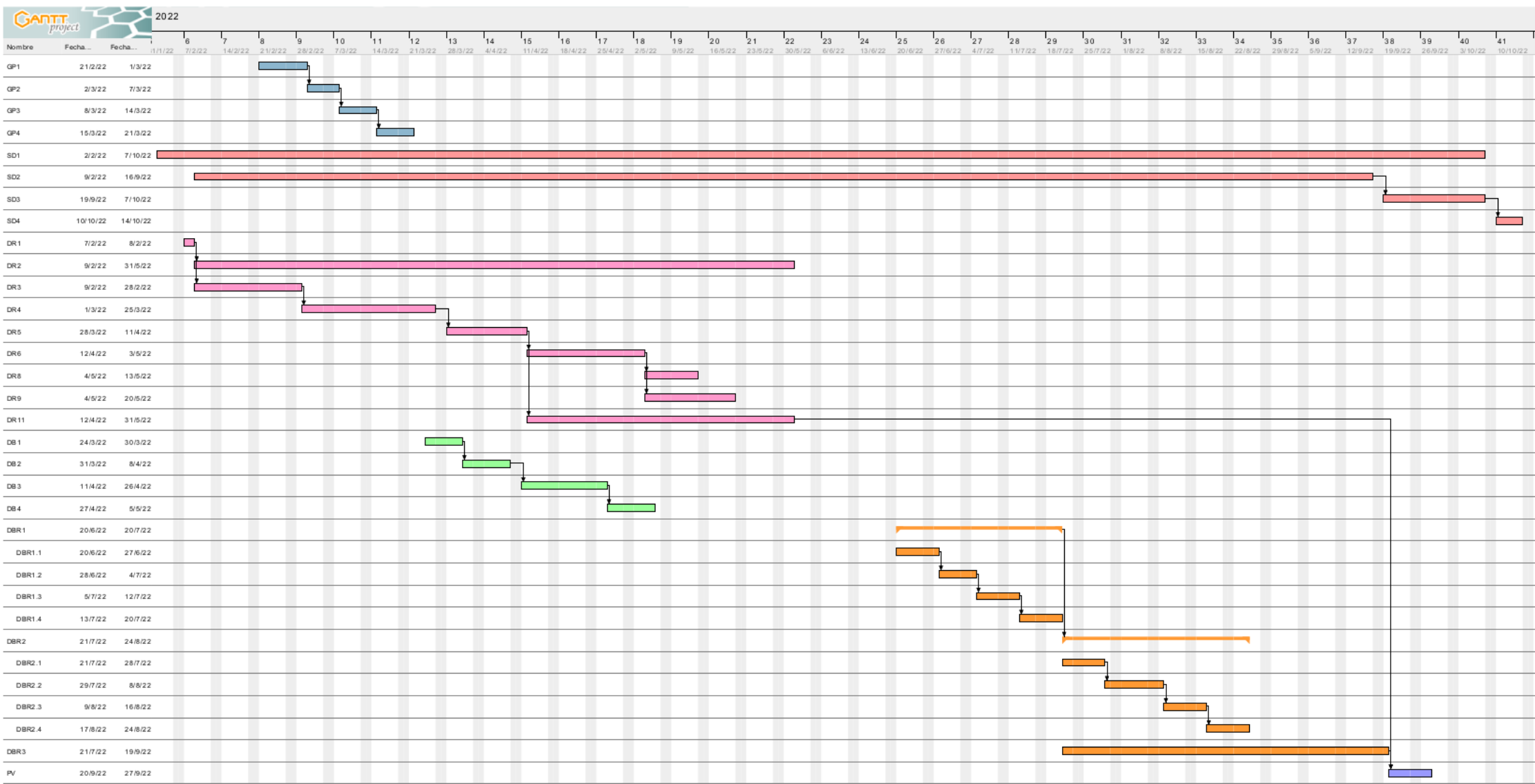


Figura 10: Diagrama de Gantt final. Font: elaboració pròpia

5. Planificació temporal

5.3 GESTIÓ DEL RISC: PLANS ALTERNATIUS I OBSTACLES

A continuació en la Taula 4 podem veure els diferents riscos ja introduïts en l'Apartat 3.3 (d'obstacles i riscos), amb la resolució i el nivell de risc experimentat durant la realització del projecte (impacte i probabilitat).

Risc	Impacte	Probabilitat	Pla alternatiu
Termini del projecte	Mitjà	Mitjana	Gestionar el temps abans de començar el projecte, pot generar infravaloracions en el temps. Per aquest motiu la planificació del desenvolupament acabarà un mes abans del període de finalització i es sobreestimarà el temps d'aquelles tasques que consideri que tenen risc mig o alt.
Planificació temporal errònia	Alt	Alta	Les tasques que no hagin estat validades durant l'Sprint assignat, seran agregades amb les de l'Sprint següent. També s'anirà revisant la planificació i ajustant les tasques futures a través d'experiències ja viscudes. En el cas de quedar-nos sense temps i una nova planificació no ajudés, s'incrementaran les hores de treball per setmana
La inexperiència en el camp d'estudi	Alt	Alta	Com mai s'ha tocat un tema similar i tampoc no s'ha tingut cap experiència amb cap de les eines, es compta amb una dificultat en l'estimació de temps. Per tant, en base a la duració dels tutorials i/o la quantitat d'informació que existeix sobre el tema, es sobrevaloraran el nombre d'hores.
Corba d'aprenentatge de Radiance	Alt	Alta	Al ser una llibreria que té documentació escassa i disposar de molt poca informació, el temps d'aprenentatge segurament serà gran, i això implica menor temps per ser productiva. Per tant, es procurarà acabar la fase de Radiance el més aviat possible, per poder finalitzar l'eina en el període d'entrega. I si tot va bé la UPC, comprarà el llibre digital de Radiance per agilitzar el seu aprenentatge.
Pèrdua de dades	Alt	Baixa	Anar guardant en el núvol el desenvolupament (Drive, GitHub)
Bugs i altres dificultats en la programació	Mitjà	Alta	Els imprevistos i els bloquejos deguts a la programació poden arribar a ser costos en temps, per tant s'ha estimat que la programació serà una de les fases amb més dedicació.
Recursos computacionals	Baix	Baixa	Treballar en entorns 3D té un cost en la RAM, i contra més gran sigui el cost, major serà el temps d'execució. Per tant, en el cas que el temps d'execució sigui molt gran o els recursos computacionals no suficients, es treballarà amb un nombre de models més limitat. Aquesta solució no hauria de repercutir en la planificació del treball.

Taula 4: Riscos i plans alternatius. Font: Elaboració pròpia

6 GESTIÓ ECONÒMICA

6.1 COSTOS DE PERSONAL PER ACTIVITAT

Encara que en el TFG tots els rols seran assumits pel director, codirector i l'estudiant. A partir de les tasques definides, podem trobar quatre rols diferents:

- **Cap de projecte (CP):** Principal responsable del projecte, s'encarrega d'encaminar-lo i assegurar el seu correcte desenvolupament.
- **Programador (P):** S'encarrega d'implementar l'eina.
- **Tester (T):** S'ocupa de realitzar les proves de validació del sistema. Ha de dissenyar proves, executar-les i presentar un informe per a poder arreglar els errors trobats.
- **Dissenyador (D):** S'encarrega de dissenyar l'Add-on.

Perquè sigui fiable l'estimació del salari, s'ha obtingut la informació de [Glassdoor](#), que proporciona la mitjana del salari brut anual en diferents rols que treballen en sectors propers al desenvolupament.

Per calcular el cost de cada rol en l'àmbit d'enginyeria, s'ha considerat una jornada laboral de 8h amb 220 dies (s'eliminen dissabtes, diumenges i els dies festius), per tant per a poder extreure la informació en hores s'ha efectuat el següent càlcul:

$$\text{Salari (€/hora)} = \frac{\text{Salari (€/any)}}{1760 \text{ hores}} \quad (3)$$

No obstant, per tenir el cost que pagaria l'empresa al treballador, se li ha de sumar el cost de la seguretat social que representa un 30 % d'increment del salari brut. Per tant, el càlcul que aplicarem per obtenir el salari brut amb SS serà el següent:

$$\text{Salari brut amb SS} = \text{Salari brut} \cdot 1,30 \quad (4)$$

El procés d'investigació de sous, s'ha resumit en la Taula 5. A cada rol se li ha adjuntat la seva referència de Glassdoor on s'ha obtingut l'estimació del seu salari brut anual mig, i a partir dels càlculs descrits anteriorment, s'ha calculat el salari brut amb SS en hores.

Rol	Referència	Sou brut (€/any)	Cost horari amb SS (€/hora)
Cap de projecte	[20]	48 067 €/any	35,50 €/h
Programador	[21]	25 372 €/any	18,74 €/h
Tester	[22]	25 000 €/any	18,47 €/h
Dissenyador UI	[23]	34 000 €/any	25,11 €/h

Taula 5: Sous per hora en funció del rol. Font: Elaboració pròpia

A partir d'aquestes estimacions, en la Taula 6, es determina de manera detallada el cost de cada tasca a partir del sou brut amb SS calculat, i en la Taula 7 el cost que s'hauria de pagar a cada membre de l'equip.

6. Gestió econòmica

ID	Tasca	Temps	Hores/rol				Cost brut amb SS
			CP	P	T	D	
GP	Gestió de projecte	75h	75h	-	-	-	2662,50 €
GP.1	Contextualització i abast	26h	26h	-	-	-	923,00 €
GP.2	Planificació temporal	12h	12h	-	-	-	426,00 €
GP.3	Pressupost i sostenibilitat	13h	13h	-	-	-	461,50 €
GP.4	Documentació de la fase inicial	24h	24h	-	-	-	852,00 €
SD	Seguiment i documentació	110h	47h	53h	7h	3h	2866,34 €
SD.1	Sprint Review	16h	8h	6h	1h	1h	440,02 €
SD.2	Documentació de l'Sprint	24h	-	22h	1h	1h	455,86 €
SD.3	Documentació fase final	56h	25h	25h	5h	1h	1473,46 €
SD.4	Preparació de la presentació final	14h	14h	-	-	-	497,00 €
DR	Estudi i proves amb Radiance	65h	-	65h	-	-	1218,10 €
DR.1	Instal·lació de l'entorn Radiance	1h	-	1h	-	-	18,74 €
DR.2	Buscar informació i tutorials	1h	-	1h	-	-	18,74 €
DR.3	Execució de tests i seguiment de tutorials	8h	-	8h	-	-	149,92 €
DR.4	Reproducció d'un cas base amb Radiance	8h	-	8h	-	-	149,92 €
DR.5	Familiarització en la definició de materials, llums i escena	7h	-	7h	-	-	131,18 €
DR.6	Definició de materials i textures	16h	-	16h	-	-	299,84 €
DR.7	Calibratge de textures	4h	-	4h	-	-	74,96 €
DR.8	Definició de la il·luminació natural	8h	-	8h	-	-	149,92 €
DR.9	Definició de la il·luminació artificial utilitzant .ies	8h	-	8h	-	-	149,92 €
DR.10	Calibratge de l'environment map, combinat amb la il·luminació del cel	4h	-	4h	-	-	74,96 €
DB	Estudi i proves amb Blender	30h	-	30h	-	-	562,20 €
DB.1	Familiarització amb l'entorn de Blender	12h	-	12h	-	-	224,88 €
DB.2	Organització d'assets en l'escena	4h	-	4h	-	-	74,96 €
DB.3	Modificació de material d'assets, si és pertinent	6h	-	6h	-	-	112,44 €
DB.4	Introducció a l'API de Python	8h	-	8h	-	-	149,92 €
DBR	Implementació de la connexió de Blender i Radiance	170h	-	146h	-	2h	2786,26 €
DBR.1	Desenvolupament bàsic de l'script d'automatització	64h	-	-	-	-	-
DBR.1.1	Conversió de models	16h	-	16h	-	-	299,84 €
DBR.1.2	Conversió de materials	16h	-	16h	-	-	299,84 €
DBR.1.3	Conversió de llums	16h	-	16h	-	-	299,84 €
DBR.1.4	Conversió de la càmera	16h	-	16h	-	-	299,84 €
DBR.2	Extensió avançada de l'script d'automatització	74h	-	-	-	-	-
DBR.2.1	Automatitzar materials	20h	-	20h	-	-	374,80 €
DBR.2.2	Automatitzar il·luminació artificial	24h	-	24h	-	-	449,76 €
DBR.2.3	Automatitzar il·luminació natural	18h	-	18h	-	-	337,32 €
DBR.2.4	Incorporar trajectòries temporals del Sol	12h	-	12h	-	-	224,88 €
DBR.3	Conversió d'Script a Add-on	8h	-	6h	-	2h	162,66 €
PV	Proves de validació	24h	-	-	24h	-	443,28 €
Total		450h	122h	292h	31h	5h	10 501,20 €

Taula 6: Cost estimat en funció de les tasques. Font: Elaboració pròpia

Per estimar el cost del personal, s'ha calculat el producte del total d'hores realitzades de cada rol pel seu cost en hores.

6. Gestió econòmica

Posició	Temps	Cost amb SS
Cap de projecte	122h	4331,00 €
Programador	292h	5472,08 €
Tester	31h	572,57 €
Dissenyador	5h	125,55 €

Taula 7: Cost del personal. Font: Elaboració pròpia

6.2 COSTOS MATERIALS

Pel que fa al **Hardware**, s'utilitzarà quatre ordinadors portàtils *MSI*, un per cada empleat. Aquest recurs té un preu d'adquisició de 1149 € i una vida útil de 4 anys, l'amortització d'aquest ordinador s'ha calculat a partir de l'Equació 5. En la Taula 8 es mostra el cost del maquinari per personal i el total que es correspon a les 450h d'ús.

$$\text{Cost d'amortització} = \frac{\text{Cost d'adquisició (€)}}{\text{Vida útil (anys)} \cdot 1760 \text{ hores}} \cdot \text{hores d'ús del recurs} \quad (5)$$

Rol	Hores d'ús del recurs	Cost maquinari
Cap de projecte	122h	19,91 €
Programador	292h	47,66 €
Tester	31h	5,06 €
Dissenyador	5h	0,82 €
Total	450h	73,45 €

Taula 8: Cost del maquinari per personal. Font: Elaboració pròpia

Per altra banda, tot el **Software** que s'utilitza (esmentats en l'Apartat de Descripció de tasques) són gratuïts, de llicència lliure, i per tant no s'aplicarà cap cost.

6.3 COSTOS GENÈRICS

S'ha decidit llogar una sala o una oficina de treball per facilitar els comptes i abastir a l'equip de treball un lloc on poder treballar i reunir-se còmodament. Per calcular el cost, s'ha localitzat un coworking-space en Barcelona [24] de 210 €/mes (iva no inclòs) amb Internet d'alta velocitat, electricitat, accés a impressora i escàner, aire condicionat, calefacció, sala de reunions i amb completa disponibilitat.

Per tant, el cost total del co-working serà:

$$\text{Cost del co-working} = 210 \text{ €/mes} \cdot 1.21 \text{ IVA} \cdot 4 \text{ mesos} = 1016,40 \text{ €} \quad (6)$$

6.4 CONTINGÈNCIES

Per a contingències, s'estableix un percentatge específic del 15 %. Així doncs, a la Taula 9 es pot veure els càlculs concrets per cadascun dels costos del projecte aplicant l'Equació 7.

$$\text{Contingència} = \text{Cost} \cdot 0,15 \quad (7)$$

Recursos	Cost	Contingència
Humans	10 501,20 €	1575,18 €
Materials	73,45 €	11,02 €
Generals	1016,40 €	152,46 €
Total	11 591,05 €	1738,66 €

Taula 9: Cost de contingència del 15%. Font: Elaboració pròpia

6.5 COSTOS DELS IMPREVISTOS

En l'Apartat 5.3 es van detallar els riscos que podien aparèixer en el projecte i els plans alternatius. En la Taula 10 es quantificarà el cost d'afrontar cada un d'aquests riscos, en el cas de causar alguna despesa econòmica. L'estimació es calcularà amb la següent Equació:

$$\text{Cost del risc} = \text{Preu del risc} \cdot \text{Probabilitat del risc} \quad (8)$$

- **Augment del temps de desenvolupament:** En el cas de necessitar més temps de desenvolupament, s'agregaràn 45h, que es corresponen al 10% de les hores totals del projecte, la distribució serà de 44h de desenvolupament i 1h de testing. El preu del risc seria $44h \cdot \text{Cost}_{\text{programador}} + 1h \cdot \text{Cost}_{\text{tester}} = 843,03 \text{ €}$. Probabilitat del risc: 20%
- **Modificacions en la planificació:** En el cas de què es (canviïn, eliminin o s'agreguin) tasques en la planificació, s'hauràn de replanificar totes les tasques a partir d'aquesta, per tant s'agregaràn 2h de planificació i 20h de desenvolupament (s'estima que hi hauran més agregacions que eliminacions), i seran adjudicades al programador, ja que és el rol més polivalent. El preu de risc seria $2h \cdot \text{Cost}_{\text{cap_de_projecte}} + 20h \cdot \text{Cost}_{\text{programador}} = 445,80 \text{ €}$. Probabilitat del risc: 20%
- **Averia a l'ordinador:** En el cas de que algun recurs computacional falli, s'haurà d'arreglar o comprar un de nou. El preu del risc seria **65,16 €** (Taula 8). Probabilitat del risc: 10%

Per altre banda els risc de pèrdua de dades no suposara un augment de cost, ja que les dos plataformes utilitzades (GitHub i Drive) són gratuïtes.

Imprevist	Preu	Probabilitat	Cost del risc
Augment del temps de desenvolupament	843,03 €	20%	168,61 €
Modificacions en la planificació	445,80 €	20%	89,16 €
Averia a l'ordinador	65,16 €	10%	6,52 €
Total	1353,99 €	-	264,29 €

Taula 10: Cost d'imprevistos. Font: Elaboració pròpia

6. Gestió econòmica

6.6 PRESSUPOST FINAL

En resum, el pressupost final de tot el projecte , té un valor de **13 594,00 €**. En la Taula 11 es presenta la suma de tots els costos, per obtenir el cost final.

Tipus	Cost
Costos del personal per activitat (CPA)	10 501,20 €
Costos materials (CM)	73,45 €
Costos generals (CG)	1016,40 €
Contingències	1738,66 €
Imprevistos	264,29 €
Total	13 594,00 €

Taula 11: Pressupost final del projecte. Font: Elaboració pròpia

6.7 CONTROL DE GESTIÓ

Per tenir un control de les desviacions en el pressupost, en el moment que es tanqui una tasca s'actualitzarà el pressupost en funció de les hores treballades i dels imprevistos. Mitjançant el seguit de fórmules explicades a continuació, es calcularan les desviacions dels valors reals teòrics previstos inicialment, aquests càlculs es realitzaran a través de programaris com l'Excel, que permeten automatitzar el control d'hores.

- **Desviació del cost:** $\text{Desviació de cost} = \text{Cost estimat} - \text{Cost real}$
- **Desviació d'hores:** $\text{Desviació d'hores} = \text{Hores estimades} - \text{Hores reals}$
- **Desviació d'hores per tasca:** $\text{Desviació d'hores/tasca} = (\text{Hores estimades} - \text{Hores reals}) \cdot \text{Cost real}$
- **Desviació del cost per tasca:** $\text{Desviació de cost/tasca} = (\text{Cost estimat} - \text{Cost real}) \cdot \text{Hores reals}$

Gràcies a aquests indicadors de desviació, podrem comprendre fàcilment on i per què hi ha hagut una desviació, així com el cost temporal i/o econòmic. Si les desviacions són degudes als imprevistos, es calcularà a partir dels costos ja estimats, i en el cas de què no fossin suficients o les desviacions provinguessin d'un altre factor no esmentat, s'utilitzarà el pressupost reservat a les contingències. Addicionalment si les desviacions són excessivament grans degut a la naturalesa dels problemes, s'ajustarà l'abast del projecte i es replanificarà tot plegat. Finalment, actualitzarem la llista dels costos.

6.8 DESVIACIÓ DEL PRESSUPOST TOTAL

En la Taula 12, s'han marcat de color verd, els pressupostos que s'han desviat positivament, és a dir que s'han reduït, respecte a l'inicial, i de color vermell s'han marcat els costos que s'han desviat negativament, és a dir, que s'han incrementat.

6. Gestió econòmica

ID	Tasca	Temps	Hores/rol				Cost brut amb SS
			CP	P	T	D	
GP	Gestió de projecte	75h	75h	-	-	-	2662,50 €
GP.1	Contextualització i abast	26h	26h	-	-	-	923,00 €
GP.2	Planificació temporal	12h	12h	-	-	-	426,00 €
GP.3	Pressupost i sostenibilitat	13h	13h	-	-	-	461,50 €
GP.4	Documentació de la fase inicial	24h	24h	-	-	-	852,00 €
SD	Seguiment i documentació	132h	51h	53h	7h	3h	3083,30 €
SD.1	Sprint Review	24h	12h	10h	1h	1h	656,98 €
SD.2	Documentació de l'Sprint	20h	-	18h	1h	1h	380,9 €
SD.3	Documentació fase final	60h	25h	29h	5h	1h	1548,42 €
SD.4	Preparació de la presentació final	14h	14h	-	-	-	497,00 €
DR	Estudi i proves amb Radiance	74h	-	74h	-	-	1386,76 €
DR.1	Instal·lació de l'entorn Radiance	1h	-	1h	-	-	18,74 €
DR.2	Buscar informació i tutorials	8h	-	8h	-	-	149,92 €
DR.3	Execució de tests i seguiment de tutorials	8h	-	8h	-	-	149,92 €
DR.4	Reproducció d'un cas base amb Radiance	8h	-	8h	-	-	149,92 €
DR.5	Familiarització en la definició de materials, llums i escena	7h	-	7h	-	-	131,18 €
DR.6	Definició de materials i textures	16h	-	16h	-	-	299,84 €
DR.8	Definició de la il·luminació natural	8h	-	8h	-	-	149,92 €
DR.9	Definició de la il·luminació artificial utilitzant .ies	8h	-	8h	-	-	149,92 €
DR.11	Definició dels paràmetres de renderització	10h	-	10h	-	-	187,40 €
DB	Estudi i proves amb Blender	38h	-	38h	-	-	712,12 €
DB.1	Familiarització amb l'entorn de Blender	12h	-	12h	-	-	224,88 €
DB.2	Organització d'assets en l'escena	6h	-	6h	-	-	112,43 €
DB.3	Modificació de material d'assets, si és pertinent	12h	-	12h	-	-	224,88 €
DB.4	Introducció a l'API de Python	8h	-	8h	-	-	149,92 €
DBR	Implementació de la connexió de Blender i Radiance	170h	-	146h	-	2h	2786,26 €
DBR.1	Desenvolupament bàsic de l'script d'automatització	64h					
DBR.1.1	Conversió de models	16h	-	16h	-	-	299,84 €
DBR.1.2	Conversió de materials	16h	-	16h	-	-	299,84 €
DBR.1.3	Conversió de llums	16h	-	16h	-	-	299,84 €
DBR.1.4	Conversió de la càmera	16h	-	16h	-	-	299,84 €
DBR.2	Extensió avançada de l'script d'automatització	74h					
DBR.2.1	Automatitzar materials	20h	-	20h	-	-	374,80 €
DBR.2.2	Automatitzar il·luminació artificial	24h	-	24h	-	-	449,76 €
DBR.2.3	Automatitzar il·luminació natural	18h	-	18h	-	-	337,32 €
DBR.2.4	Incorporar trajectòries temporals del Sol	12h	-	12h	-	-	224,88 €
DBR.3	Conversió d'script a Add-on	8h	-	6h	-	2h	162,66 €
PV	Proves de validació	24h	-	-	24h	-	443,28 €
Total		509h	122h	292h	31h	5h	11 074,22 €

Taula 12: Cost final en funció de les tasques. Font: Elaboració pròpia

La desviació total del cost del personal és de $11074 - 10501,20 = 573,20\text{€}$.

A més a més, s'ha d'aplicar una despesa del maquinari que es troba valorada en $83,07\text{€}$ (aplicació de l'equació 9):

$$\text{Cost d'amortització} = \frac{1149}{4 \cdot 1760} \cdot 509 = 83,07 \quad (9)$$

6. Gestió econòmica

Pel que fa al co-working, la despesa total serà de:

$$\text{Cost del co-working} = 210 \text{ €/mes} \cdot 1.21 \text{ IVA} \cdot 8 \text{ mesos} = 2032,80 \text{ €} \quad (10)$$

Per tant, la desviació del maquinari i el co-working ha estat: 9,62 i 1016,40 €, respectivament. Sumant totes les desviacions dels costos (personal, màquinaria i co-working), s'obté que **la desviació total és d'uns 1599,04€**, la qual es troba coberta pel cost total de contingència que es va calcular.

7 SOSTENIBILITAT

7.1 DIMENSIÓ ECONÒMICA

L'impacte del projecte en l'àmbit econòmic total ha estat 11 074,22 €, i el podem veure desglossat en l'Apartat 6.6. És un pressupost que inclou tots els recursos necessaris, a fi d'amortitzar-los i no desvalorar-ne cap. La partida gran de despeses són els recursos humans, que és un recurs inevitable per a la realització de qualsevol projecte.

De fet, aquest projecte és essencialment una eina d'automatització de tasques, per tant reduirà una gran quantitat d'hores en la producció d'imatges físicament acurades, a diferents professionals com arquitectes, responsables de la il·luminació... gràcies a la creació d'un add-on.

Reduint el número d'hores de qualsevol treballador, els costos totals disminuiran, ja que el càlcul de qualsevol projecte és:

$$\text{Cost total} = \sum (\text{valor hora de treball}_i \cdot \text{hores de treball}_i)$$

Respecte la vida útil, al ser un add-on de Blender, l'eina es troba lligada a aquest software, i si canvia quelcom en la llibreria bpy de Python, s'haurà d'afegir el cost de manteniment. No obstant, la probabilitat és baixa i s'estima que no hi haurà cap despesa econòmica en 5 anys.

Per tant, encara que tota l'estona s'hagi parlat d'un cost econòmic, es podria parlar d'inversió, perquè en un futur proper aquests costos podrien trobar-se coberts.

7.2 DIMENSIÓ AMBIENTAL

El projecte és purament software, per tant l'impacte ambiental és mínim, ja que no necessita de recursos naturals per desenvolupar-se. L'únic recurs necessari és l'ordinador portàtil com s'ha mencionat anteriorment, que en efecte té un impacte mediambiental tant en la seva producció com en el seu consum elèctric, tot i que depenent de la font d'obtenció d'aquesta electricitat, l'impacte mediambiental serà més gran o no.

A partir de la font del ministeri[25] i el consum elèctric del meu ordinador es va fer una aproximació de la seva petjada mediambiental. Aquestes dades han estat recollides en la Taula 13.

Per a una planta europea	Consum aproximat del meu ordiandor 200W per 509h de funcionament	Contaminants emesos a l'atmosfera o residus radioactius generats
0.166 kg/kwh CO2	101.8 kwh	16.9 kg CO2
0.254 g/kwh SO2	101.8 kwh	25.86 g SO2
0.217 g/kwh NOx	101.8 kwh	17.31 g NOx
Residus radioactius de baixa i mitjana activitat 0.00235cm ³ /kwh	101.8kwh	0.24 cm ³ residus radioactius de baixa i mitjana activitat
Residus radioactius d'alta activitat 0.287 mg/kwh	101.8kwh	29.22 mg residus radioactius d'alta activitat

Taula 13: Emissions contaminats. Font: Elaboració pròpia

7. Sostenibilitat

El projecte busca reduir i minimitzar el temps d'energia requerida efectuant previsualitzacions i fent que el codi sigui el més eficient possible, per tal de tenir un menor consum elèctric i una menor petjada ambiental. Pel que fa al risc ambiental aquest projecte no provoca cap, més aviat proporciona una millora ja que simula la il·luminació, per tant podria participar en projectes d'*smart cities* i decidir si col·locar sensors intel·ligents que determinin quan s'encenen i s'apaguen els llums, en funció de la irradiació del terra, d'aquesta manera el nostre projecte ajudaria a reduir l'impacte ambiental.

Un altre vessant en el qual podria ser útil aquest programa, tindria a veure en el posicionament òptim dels llums. Una bona col·locació de les fonts de llum artificials permetria ajustar el nombre de llums i la seva potència per aconseguir la il·luminació òptima i, a l'hora permetria reduir la contaminació lumínica i ambiental.

7.3 DIMENSIÓ SOCIAL

A nivell personal opino que la realització d'aquest projecte m'ha aportat diversos beneficis, en primer lloc en tenir la possibilitat de tenir experiència, en el desenvolupament d'un projecte d'un cas real, en segon lloc per saber com es gestiona un projecte de recerca, i finalment els aprenentatges de programaris nous i molts coneixements sobre il·luminació i el món 3D. Per tant tot això serà un enriquiment de cara a futurs projectes.

A nivell social, aquests coneixements, a partir de l'eina, obren la possibilitat de desenvolupar aplicacions en l'àmbit del confort ambiental i en l'estalvi de la despesa energètica.

L'impacte social més rellevant però, és conseqüència de què el projecte ha creat una eina útil per a historiadors de l'art i gent del patrimoni (sense coneixements informàtics), que els permetrà simular il·luminacions en entorns que ja no existeixen o han estat restaurats, facilitant-los la seva utilització i l'obtenció de resultats acurats en diferents moments de la història. Això els permetrà poder entendre millor diferents tipus d'actes cerimonials com els litúrgics. Aquesta aportació en el coneixement de la història i el món de l'art és una millora per a tota la societat.

7.4 MATRIU DE SOSTENIBILITAT

A mode de resum es presenta la Taula 14 que sintetitza els apartats anteriors.

	PPP	Vida útil	Riscs
Econòmica	11 074,22 €	5 anys	La no viabilitat econòmica
Ambiental	16,9 kg/CO2	Reducció energètica	La no millora energètica
Social	9/10	8/10	Pocs

Taula 14: Matriu de sostenibilitat. Font: Elaboració pròpia

7.5 AUTOAVALUACIÓ

En GEI-FIB sempre s'ha insistit molt en la sostenibilitat en l'àmbit TIC, sobretot en les assignatures de Hardware, on ens han fet xerrades i tallers, posant molt d'èmfasi en conceptes bàsics com l'obsolescència programada, les deixalles electròniques (remarcant l'impacte sobre la salut en les persones del tercer món), els costos de producció (disseny, validació i fabricació) que pot arribar a tenir crear un xip, entre d'altres; però no només ens informaven dels impactes, sinó també de possibles solucions per a reduir-los com les ecoetiquetes, i les eines d'avaluació com la Llei d'Amdahl, que indica si hi ha un guany significatiu en el disseny del maquinari.

En realitzar l'[enquesta](#) m'he adonat de la importància que tenen els informes de sostenibilitat en els projectes i que encara em queda molt per aprendre, sobretot en els aspectes econòmics i socials, on he descobert conceptes que desconeixia i hauré d'anar incorporant en la meva activitat.

En la meva opinió, la innovació mai s'ha de parar i menys si és sostenible. A més a més, crec que l'execució de diferents tècniques computacionals podria arribar a solucionar problemes de sostenibilitat a nivell energètic.

8 RENDERITZANT AMB RADIANCE

A l'hora de simular Sant Quirze de Pedret, s'han tingut en compte nombrosos factors per ajustar els resultats a la realitat (la fotografia).

De fet, les primeres imatges del patrimoni es van generar manualment, i de manera paral·lela es va anar aprenent aquest nou llenguatge: la definició de materials, la il·luminació, els executables, etc., per després generar la conversió de dades Blender a Radiance, que s'explicarà en la Secció 9.

El procés que es va seguir per crear renderitzacions de manera manual amb Radiance es mostra en la Figura 11.

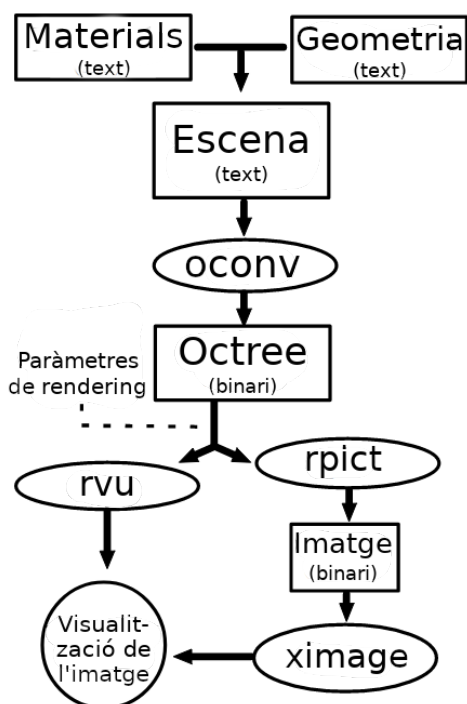


Figura 11: Procés de renderització amb Radiance. Les el·lipses són executables propis de Radiance. Els rectangles són fitxers, els parèntesis indiquen el tipus. Font: elaboració pròpia

Primer es va començar definint els materials de cada objecte (geometria) que es mostrava en l'escena Blender. Quan tots els materials i tota la part de geometria es van definir en un o més arxius d'escena, aquesta informació va ser compilada en un *octree* utilitzant el programa *oconv* de Radiance, d'aquesta manera:

```
$ oconv a1.rad ... aN.rad > file_name.oct
```

L'extensió *.oct* de *file_name.oct* és afegida com a convenció per identificar els *octree* produïts per *oconv*. Aquest és un fitxer binari essencial per a generar la renderització, la qual es pot visualitzar de dues maneres:

- *rvu*: Mostra l'escena interactivament. Això resulta útil per tenir una vista prèvia de l'escena, decidir la ubicació de la il·luminació i seleccionar vistes per la representació final d'alta qualitat amb *rpict*.
- *rpict*: Genera una imatge amb una qualitat més alta que *rvu* en format *.pic* o *.hdr*. Aquesta imatge és guardada i no es pot visualitzar fins a executar la comanda *ximage*.

Ambdues opcions s'han d'ajustar amb els paràmetres de renderització (focus de l'observador, exposició, il·luminació indirecta, etc.) a través de la consola, els quals es poden trobar llistats executant: *rvu -help* i *rpict -help*.

Així doncs, es va començar a renderitzar l'església de Sant Quirze de Pedret amb Radiance de manera manual.¹

¹Les instruccions per instal·lar Radiance es troben en l'Annex A

8. Renderitzant amb Radiance

8.1 ESCENA SANT QUIRZE DE PEDRET

D'entrada, l'escena de Sant Quirze de Pedret es va rebre amb un objecte en format .fbx conjuntament amb una *baked texture*, la qual definia la coloració de tot el model. Més endavant es va demanar si ens podien entregar el model de l'església de Sant Quirze de Pedret amb materials separats, ja que en Radiance no té sentit definir tota una escena amb un sol material.

Cal recordar que aquest no és un programa artístic, sinó científic i es caracteritza per definir els materials en unitats físiques que determinen la interacció de la il·luminació amb l'objecte. Per aquest motiu s'em va proporcionar l'escena en Blender, on cada objecte es trobava linkat als seus respectius materials.

Així i tot, es va començar a treballar amb el primer model rebut, el que tenia un únic material, i posteriorment amb el de múltiples materials. Per tant, en aquest projecte s'explicarà el procés amb les dues escenes, les quals fan servir el mateix model (digitalització de Sant Quirze de Pedret), però configurades de maneres diferents.

L'església tal com es mostra en la Figura 12 (escena ja configurada) té una amplada de 17,8m, 21m de profunditat, i una alçada de 9,87m. Cal fer esment específic sobre què les mesures han estat preses en metres, és més, en tot el projecte només s'utilitzaran les unitats del S.I..

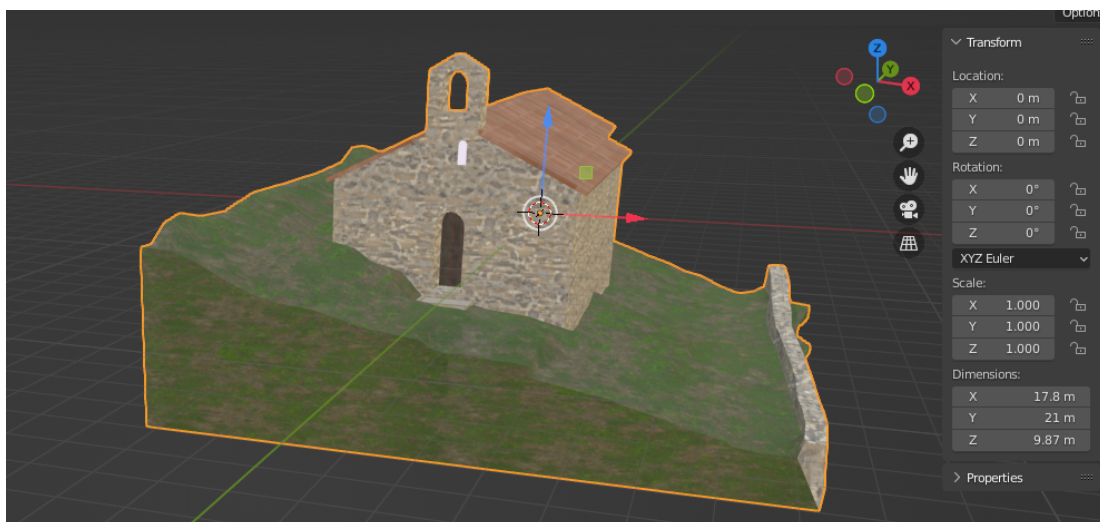
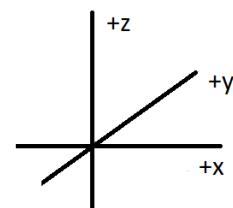


Figura 12: Escena de Sant Quirze de Pedret[26] ja configurada en Blender. Font: elaboració pròpia

Les dues escenes van ser configurades en Blender, ja que les operacions es simplificaven a l'hora d'utilitzar Radiance, amb les transformacions de rotació i translació.

La primera transformació va ser aconseguir el mateix sistema de coordenades (X,Y,Z) que hi ha al Radiance, el qual s'utilitzarà d'ara endavant, (Figura 13) amb l'eix x marcant l'est i el z positiu el nord. Per tant, es van rotar tots els objectes de l'escena amb el mateix nombre de graus per tal d'obtenir l'orientació de Radiance.



La segona transformació va ser la col·locar el model centrat al punt (0,0,0). Després d'haver fet diferents proves, es va concloure que aquesta era la millor opció, ja que a l'exportar el model a OBJ, les coordenades que es tenen en

Figura 13: Sistema de coordenades de Radiance. Font: elaboració pròpia

8. Renderitzant amb Radiance

compte són les globals, les de l'escena Blender.

Finalment, totes les transformacions fetes van ser aplicades amb (Ctr + A → All transforms) per a establir la rotació i la ubicació actual de l'objecte a 0.

D'altra banda a Radiance existeixen 5 classes de definicions, amb diferents propietats:

- **Geometria:** *source, sphere, polygon, cone, cylinder, ring, mesh, instance*
- **Materials:** *light, illum, glow, spotlight, mirror, prism1, prism2, mist, plastic, metal, trans, plastic2, metal2, trans2, dielectric, interface, glass, plasfunc, metfunc, transfunc, BRTDfunc, plasdata, metadata, transdata, antimatter*
- **Textures:** *texfunc, texdata*
- **Patrons:** *colorfunc, brightfunc, colordata, brightdata, colorpict, colortext, brighttext*
- **Barrejes:** *mixfunc, mixdata, mixpict, mixtext*

La sintaxi i la semàntica de cada propietat es troba en la documentació [27], tot i així les definicions segueixen la següent estructura general:

```
mod type id
nStrings   S1 S2 ... Sn
mIntegers  I1 I2 ... Im
kFloats    F1 F2 ... Fk
```

id: L'identificador de la descripció pot rebre qualsevol nom, però ha de ser únic dins del projecte, a fi de ser una possible referència pel camp de *mod*.

mod: El modificador pot tenir com a valor un void o un *id* prèviament definit. A través de l'*id* s'agrega una nova propietat a la descripció.

type: El tipus es correspon a una de les propietats de les 5 classes de definicions abans esmentades (*plastic, sphere, light,...*), i cadascuna d'elles té una sintaxi específica que indica quin és el nombre de Strings (*nStrings*), el nombre d'enters (*mIntegers*) i el nombre de paràmetres reals (*kFloats*) que es necessiten per a crear la definició.

La majoria de les definicions dels materials implementen la regla de la cadena, $(f \circ g)' = (f' \circ g) \cdot g'$. A tall d'exemple, es mostra el concepte de regla de la cadena definint una pota d'una cadira (Geometria) de plàstic (Material):

```
void plastic myChair
      myChair cylinder leg1
```

Els primers elements de la cadena mai tenen modificador, per tant sempre s'utilitzarà el void.

El primer element, `void plastic myChair`, defineix un material que s'anomena *myChair* que té les característiques del plàstic, i aquesta definició serà inclosa en `myChair cylinder leg1`, ja que l'identificador del primer element ha esdevingut el modificador del segon. De manera que es crea un cilindre de plàstic.

Aquesta regla és crucial perquè permet definir geometria i materials compostos.

En últim lloc, els comentaris es faran amb el coixinet (#) i el que vagi a continuació de la línia serà ignorat pel compilador, però si el primer caràcter de la línia és una exclamació (!), llavors la línia serà

8. Renderitzant amb Radiance

executada com si fos una ordre de comandes.

D'ara endavant, es començarà a definir l'escena de Pedret, però abans s'introduirà la jerarquia de directoris que es va utilitzar, per tal de tenir tots els fitxers organitzats i que el codi sigués fàcil de mantenir. Aquest sistema d'organització ha estat inspirat en les recomanacions de Radiance [28].

El directori principal va ser anomenat *pedret*, de Sant Quirze de Pedret, i es va crear amb una sèrie de subdirectoris. La finalitat de les carpetes es comenten a continuació:

```
$ mkdir pedret           # directori que conté tota l'escena
$ mkdir pedret/lib       # subdirectori d'objectes i materials definits
$ mkdir pedret/lib_ies   # subdirectori de fitxers IES
$ mkdir pedret/lib_view  # subdirectori de punts de vista per la renderització
$ mkdir pedret/scenes/xx # subdirectori dels fitxers de renderització de l'escena xx
$ mkdir pedret/scenes/xx/img # subdirectori d'imatges renderitzades per l'escena xx
```

Totes les ordres de comandes mencionades, s'han executat amb la CLI oberta en la carpeta principal (*pedret*).

8.2 MATERIALS

Els materials que es trobaven en Blender eren:

- **Escena 1:** *Pedret_IX-backed*
- **Escena 2:** *low_grass_soil, limeplaster_slate_wall_uv, patinated_wood_planks_dark_uv, limeplaster_slate_wall, limeplaster_wall, stone_wall_thick_grout, patinated_wood_planks_dark_coords, smoothed_limestone_rock_surface, patinated_wood_planks, stone_wall_thick_grout, cst_root_tile.*

En Radiance les definicions dels materials es poden correspondre a cap o diversos objectes, i un objecte pot tenir múltiples materials, igual que en Blender, però convé subratllar que cada objecte haurà de tenir separats els materials amb propietats físiques diferents.

Per tant, en teoria, una mateixa geometria que té propietats físiques diferents no pot ser definida com un únic material, és a dir, seria erroni definir un objecte que, per exemple, es compona de plàstic i metall amb només un material, ja que a efectes d'il·luminació juguen papers diferents. A més a més, com mostra la Figura 14 les seves propietats són diferents:

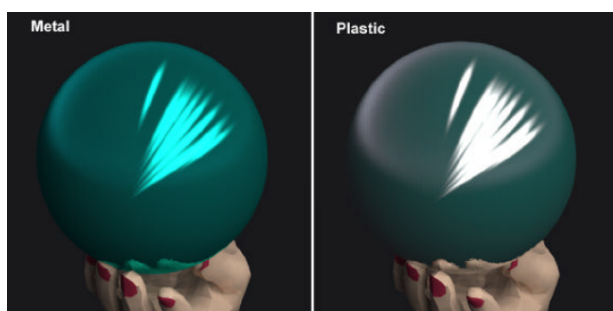


Figura 14: Les dues boles són il·luminades amb il·luminació blanca. La bola de plàstic mostra reflexos sense color mentre que la metàl·lica mostra un color turquesa, modificant el color dels reflexos. Font: [29]

Per tant, quan existeixin diferents propietats en l'objecte, s'haurien de crear tantes definicions diferents com tipus de propietats tingui.

8. Renderitzant amb Radiance

En total existeixen 25 tipus de materials i s'han de determinar quin encaixen millor per descriure els objectes, per tal d'obtenir la major aproximació a la realitat (Figura 15). Hi ha des de senzills d'utilitzar, com són el plàstic i el metall, fins a més complexos, com el dielèctric (material transparent que se li pot especificar el coeficient de transmitància i de refracció) i el BRTDfunc (el comodí dels materials, es pot especificar la reflectància, la transmitància i la direcció dels raigs) que permeten crear definicions més precises, configurant tots els aspectes direccionals de reflectància i transmitància.



Figura 15: Fotografia de Sant Quirze de Pedret. Font: Repositori de fotografies de Pedret

En aquest cas d'estudi, els elements que participen a l'escena són d'una banda **materials de construcció inorgànics**, com les roques utilitzades en les parets i l'argila amb què possiblement s'han construït les teules i, d'altra banda **fusta** i la vegetació modelada com a **gespa**.

No obstant la part exterior de Pedret és prescindible per al nostre projecte, ja que el nostre objectiu és saber l'impacte que tenia la il·luminació en l'interior de l'església. Per tant, tot allò que quedava a l'exterior com la gespa, les teules i el mur que hi ha a l'exterior s'han ignorat, perquè no s'anava a visualitzar, i es va decidir definir-los com a una **superfície Lambertiana**, és a dir, com a una superfície ideal amb reflexió difusa, que reflexa l'energia incident des d'una direcció igual a totes les direccions (veure Figura 16).

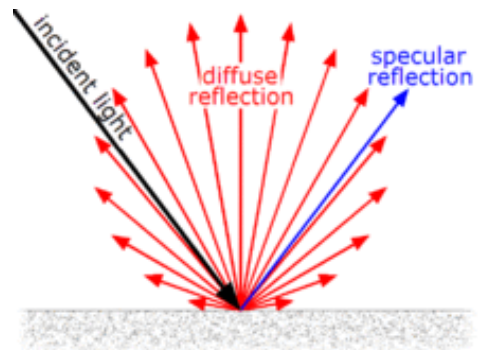


Figura 16: Reflexió difusa ideal. Superfície Lambertiana. Font: [30]

D'altra banda, tant **els materials de construcció inorgànics com la fusta** acostumen a seguir les següents propietats: reflecteixen difusament el color de la seva textura, absorbint tots els altres colors, i la reflexió especular que produeixen és incolora. Aquesta és la definició pròpia del tipus de material **plàstic** de Radiance.

No és gens estrany que el plàstic sigui el material més utilitzat en Radiance, ja que la seva definició es va pensar com un material de construcció bàsic opac que no està fet de metall. Per descomptat hi ha excepcions, però la fusta i els materials de construcció inorgànics no ho són.

En relació a les finestres, es van fer diferents definicions, ja que coexisteixen tres diferents hipòtesis, entorn a com eren en el passat. La primera és la **inexistència de finestra**, és a dir, es trobaven obertes a l'exterior, aquest va ser el cas més senzill, perquè no existeix geometria, ni materials a definir. El segon cas és l'existència de **finestres de vidre**, en aquesta situació es necessitaria d'una geometria que les defineixi i un material que seria del tipus `glass`. El tercer cas, són les **finestres de cuir**, les quals s'han definit amb un material de tipus translúcid, que és similar a un material de tipus plàstic,

8. Renderitzant amb Radiance

però deixa passar la llum a través de ell, és a dir, comparteix les mateixes propietats del plàstic més les propietats de transmissió. Aquestes propietats plàstiques van ser definides mentre es definien els plàstics.

8.2.1 DEFINICIÓ DEL PLÀSTIC

La sintaxi del material plàstic és la següent:

```
mod plastic id
0
0
5 red green blue spec rough
```

red green blue: Percentatge de vermell, verd i blau de reflectància. Tanmateix, no ens interessa definir un color sòlid, sinó una textura amb diferents colors.

spec: Reflexió especular. És mesurada amb l'espectròmetre. Rarament en les superfícies plàstiques les fraccions d'especularitat són superiors a 0.1.

rough: Rugositat de la superfície. És establerta amb RMS. Un valor de 0 correspon a una superfície perfectament llisa, i un valor d'1 a una de molt rugosa. Rarament una superfície plàstica és superior a 0.2.

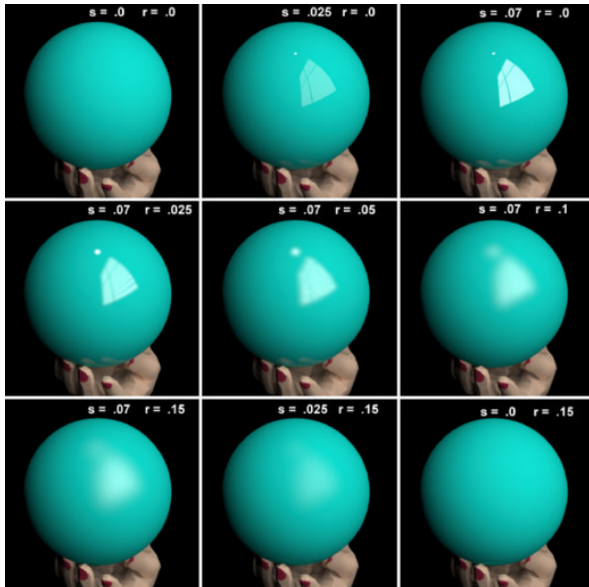


Figura 18: Bola de plàstic amb diferents especularitats i rugositats. Font: [29]

Ara bé, com aquests materials no són colors sòlids, sinó textures, a excepció del cuir que es va definir amb un color marró, a la resta de materials se'ls hi van assignar textures a través de l'herència (regla de la cadena), que s'ha formulat anteriorment, utilitzant la propietat `colorpict`, la qual s'explicarà en la següent secció:

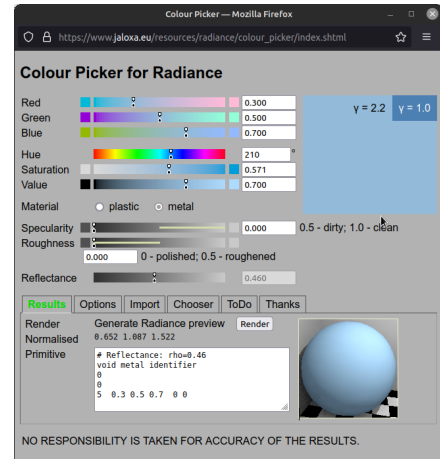


Figura 17: Colour picker. Font: elaboració pròpia

Malauradament, no disponia d'un espectròmetre ni tampoc d'un microscopi potent per a calcular la rugositat. Per tant l'`spec` i la `rough` van ser definides empíricament, i com a guia es va utilitzar la Figura 18 i el programa Radiance Colour Picker[31] (Figura 17) per previsualitzar el material i obtenir els resultats més semblants possibles a la realitat. Els valors que es van establir van ser els següents:

- Materials de construcció inorgànics: especularitat 0 i rugositat 0.2
- Cuir: especularitat 0.1 i rugositat 0.07
- Fusta: especularitat 0.07 i rugositat 0.15

Ara bé, com aquests materials no són colors sòlids, sinó textures, a excepció del cuir que es va

8. Renderitzant amb Radiance

8.2.2 DEFINICIÓ DE LES TEXTURES

La sintaxi del colorpict és la següent:

```
mod colorpict id
7+ rfunc gfunc bfunc pictfile funcfile u v transformations
0
m A1 A2 .. Am
```

rfunc gfunc bfunc: Tres paràmetres que seran passats a `funcfile` que modelen el color.

pictfile: Textura en format Radiance (`.pic` o `.hdr`).

funcfile: Fitxer que conté la funció de mapejar i acolorir la textura en la superfície. Aquest fitxer de funció rep els valors de color `red`, `green`, `blue`, amb els quals es defineix la imatge.

u v: Les coordenades de textura local. Aquestes coordenades seran interpolades i els arguments seran passats a les variables `Lu` i `Lv` durant la renderització.

transformations: Transformacions permeses amb XFORM (rotació, translació, escalat,...) que s'apliquen a `pictfile`.

Totes les textures es van obtenir a través de les dues escenes de Blender: **la primera escena (monomaterial)**, la textura de tot el model, la qual era un baking (tot l'objecte en una textura), va ser descarregada directament en format `.png`; i **la segona escena (multimaterial)**, va ser modificada per tal d'obtenir la imatge de Sant Quirze de Pedret amb només 3 textures (materials de construcció inorgànics, fusta i gespa) per tal d'arribar a la Figura 19. Els passos que es van seguir van ser els següents:

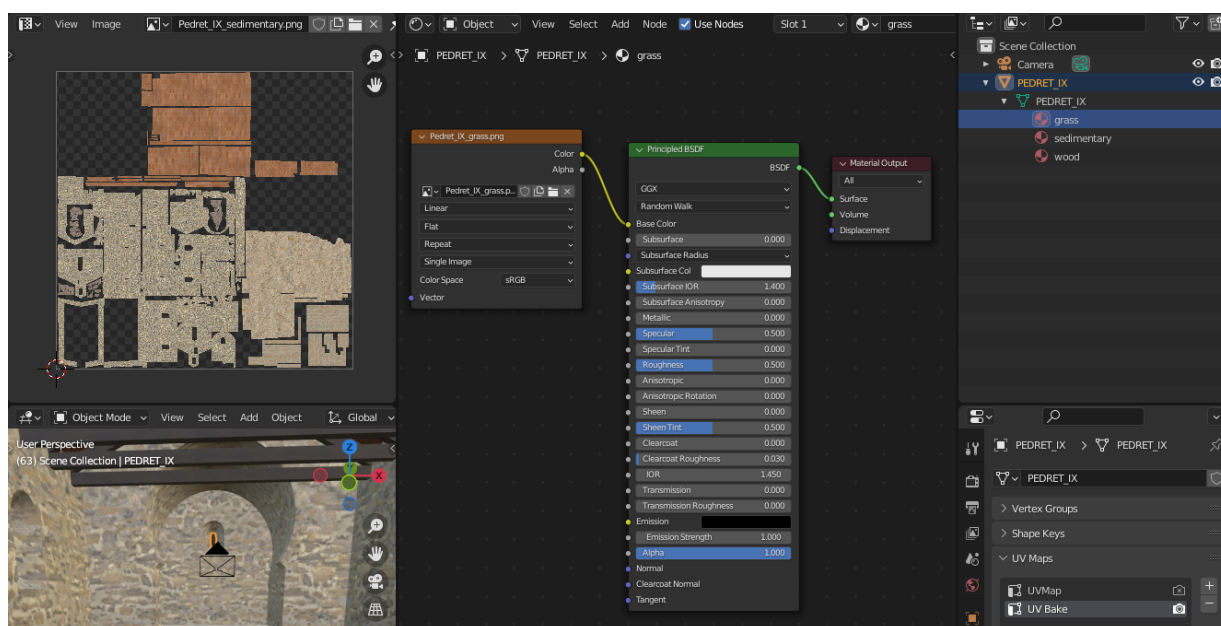


Figura 19: Resultat del baking. Font: elaboració pròpia

Primerament, tots els materials van ser agrupats segons les seves propietats, i a través de la unificació (`join`), els objectes i els seus materials es van combinar, creant **un objecte de múltiples materials**.

8. Renderitzant amb Radiance

La unificació va ser feta a través de la drecera Ctrl+J en el 3D viewport de Blender, en l'object mode. El resultat obtingut es mostra a la Figura 20 (3 objectes, en comptes dels 10 inicials).

En segon lloc, es va crear el **baking** per a cadascun dels 3 objectes, per convergir totes les textures en una. Cada una d'aquestes textures van ser descarregades en format .png per definir els tres materials en Radiance (wood, grass i building).

El procés de baking va trigar una mitja de 40 minuts per objecte, tot i que el unwrapping UV de Blender, es va poder reutilitzar, ja que ja n'existia un. Per generar el baking dels objectes, es va seguir el tutorial Bake Multiple Materials [32], el qual empra el motor de rendering Cycles, que es troba intern a Blender.

En tercer lloc, es van eliminar tots els materials de tots els objectes, per assignar a cadascun dels objectes el nou material Baked.

Per convertir la imatge .png a format Radiance (.hdr) es va utilitzar ImageMagick que es troba instal·lat per defecte en molts sistemes UNIX i Linux. La comanda mogrify utilitza l'extensió del fitxer per decidir el format al que s'exporta.

```
mogrify -format hdr image_to_convert.png
```

Executant la comanda d'amunt es van generar tres imatges amb extensió .hdr, que van ser guardades en *pedret/lib/tex*.

8.2.3 DEFINICIÓ DEL VIDRE

La sintaxi del vidre és la següent:

<pre>mod glass id</pre>	rtn gtn btn: Quantitat de transmissió en incidència normal, és a dir, la quantitat de llum que no s'absorbeix, quan l'angle amb el qual arriba la llum a la superfície és perpendicular. Per tant, tots els valors a 0 indiquen que és un cristall negre, i tots a 1 que és transparent. Per calcular aquest valor es necessita un termòmetre d'infrarojos especial per als materials.
<pre>0</pre>	
<pre>0</pre>	
<pre>3 rtn gtn btn</pre>	

Com que jo no tenia aquest tipus de termòmetre, vaig decidir definir la finestra amb la transmissivitat estàndard especificada en la referència [14], és a dir, 0.96 en tots els arguments.²

A Blender aquestes finestres de vidre van ser col·locades com un objecte independent a Pedret, Figura 21, ja que aquestes havien de ser simulades a marge de Pedret. A més a més, no sempre han d'existir, perquè es pot donar el cas que en les obertures de la paret no hi hagués cap material. No obstant

²Tots els valors que s'han predeterminat, es poden modificar de manera interactiva amb l'eina de renderització físicament acurada que s'ha creat amb Python, com per exemple els valor de transmitància que s'han pressuposat en el vidre, l'especularitat de les roques, etc.

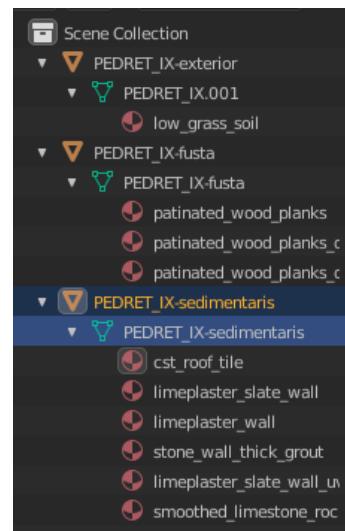


Figura 20: Panell Blender, mostra els objectes i els materials. Font: elaboració pròpia

8. Renderitzant amb Radiance

això, quan existissin les finestres de vidre o de cuir, sí que havia d'haver-hi un objecte i com a mínim un material en Blender, perquè pugui ser definit en Radiance.

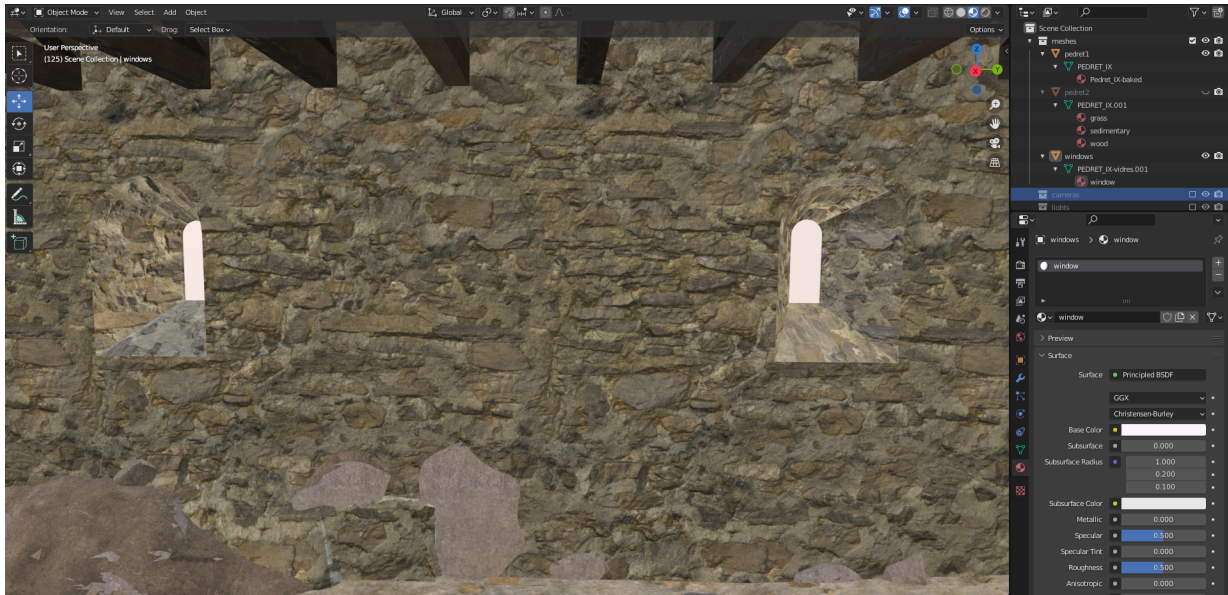


Figura 21: Unificació de les dues escenes Blender en una única escena. Font: elaboració pròpia

Com mostra el panell de la dreta de la Figura 21, les escenes de Blender es van unificar en una, és a dir, les dues escenes es van posar juntes en un mateix fitxer Blender, per fer més ràpida la interactivitat amb els objectes. Per tant les finestres, les càmeres i les fonts de llum no es troben repetides: una en cada escena. L'únic objecte que es troba repetit és *pedret* perquè està configurat de dues maneres diferents.

8.2.4 DEFINICIÓ DE TRANSLÚCID

El cuir va ser tractat com a un material translúcid, és a dir, amb les mateixes propietats que el plàstic, que permet el pas de la llum, però difuminant-la per difusió o refracció.

La sintaxi de translúcid és la següent:

```
mod trans id
0
0
7 red green blue spec rough trans tspec
```

red green blue spec rough: són les propietats plàstiques del material, les quals van ser definides mentre es definien els materials plàstics (Apartat 8.2.1).

trans: Percentatge de transmissivitat del material, és a dir, la fracció de llum penetrant que travessa el material.

tspec: Percentatge del component especular transmès que no es dispersa de manera difusa.

Durant el procés d'elecció dels valors utilitzats en les finestres de cuir es va decidir simular la transmissivitat amb un valor de 0.5 i el component especular transmès a 0, ja que el cuir no acostuma a transmetre especularitat. Aquests valors van ser definits en el material juntament amb els valors plàstics que ja es van establir, els quals eren: color marró, especularitat 0.07 i rugositat 0.15

8. Renderitzant amb Radiance

8.2.5 DEFINICIÓ DELS MATERIALS

Pedret1 (monomaterial), va ser definit com a una superfície Lambertiana al tenir un únic material. A causa d'això, la luminància no variarà encara que es canviï el punt de vista, ja que s'estarà renderitzant un model amb reflexió difusa ideal. Per aquest motiu s'haurien de trobar diferències entre la renderització de Pedret1 (monomaterial) i Pedret2 (multimaterial), que és la que s'ha anat definint al llarg dels capítols.

Respecte al codi, per a cada objecte de Blender es va crear un fitxer amb extensió `.mat` per definir els materials de l'objecte en Radiance. A més a més, els materials de Blender van ser enllaçats a Radiance, utilitzant el mateix nom en el material Blender, i l'últim id de la definició del material de Radiance.

Escena 1: Model de Pedret amb una textura `pedret/lib/mat/pedret1.mat`

```
void colorpict church
7 red green blue lib/tex/Pedret_IX_color.hdr . frac(Lu) frac(Lv)
0
0

church plastic Pedret_IX-baked
0
0
5 1 1 1 0 0
```

Codi 1: Radiance - Definició del material de pedret monomaterial. Font: elaboració pròpia

Escena 2: Model de Pedret amb múltiples textures `pedret/lib/mat/pedret2.mat`

```
void colorpict build_tex
7 red green blue Pedret_IX_sedimentary.hdr . frac(Lu) frac(Lv)
0
0

build_tex plastic sedimentary
0
0
5 1 1 1 0 0

void colorpict wood_tex
7 red green blue Pedret_IX_wood.hdr . frac(Lu) frac(Lv)
0
0

wood_tex plastic wood
0
0
5 1 1 1 0 0

void colorpict grass_tex
7 red green blue Pedret_IX_grass.hdr . frac(Lu) frac(Lv)
0
0
```

8. Renderitzant amb Radiance

```
grass_tex plastic grass
0
0
5 1 1 1 0 0
```

Codi 2: Radiance - Definició del material de Pedret multimaterial. Font: elaboració pròpia

Finestres: Seran utilitzades en les dues escenes.

Finestra de cuir: *pedret/lib/mat/lwindow.mat*

```
void trans window
0
0
7 0.294 0.21 0.13 0.1 0.07 0.0 0.5
```

Codi 3: Radiance - Definició del material de la finestra de cuir. Font: elaboració pròpia

Finestra de vidre: *pedret/lib/mat/gwindow.mat*

```
void glass window
0
0
3 .96 .96 .96
```

Codi 4: Radiance - Definició del material de la finestra de vidre. Font: elaboració pròpia

8.3 GEOMETRIA

Radiance no tan sols permet generar els teus propis models a mà a través de la classe Geometria, sinó que també permet treballar amb el format OBJ. Després d'experimentar les diferents opcions amb Radiance trobo que és més fàcil i ràpid utilitzar eines de modelatge que permeten exportar a OBJ.

Durant el procés d'exportació, Figura 22, per a cada objecte de l'escena Blender es marcava la casella d'exportar l'objecte seleccionat, especificant la Y com a Forward i la Z com a Up, com es va comentar anteriorment, ja que aquests són els eixos de Radiance (Figura 13).

Els fitxers resultants van ser anomenats: *pedret1.obj*, *pedret2.obj* i *window.obj*, i van ser guardats en el directori *pedret/lib/obj/*.

Els fitxers *.mtl generats van ser ignorats, ja que resultaven innecessaris, a causa de que a Radiance les definicions dels materials són amb unitats físiques.

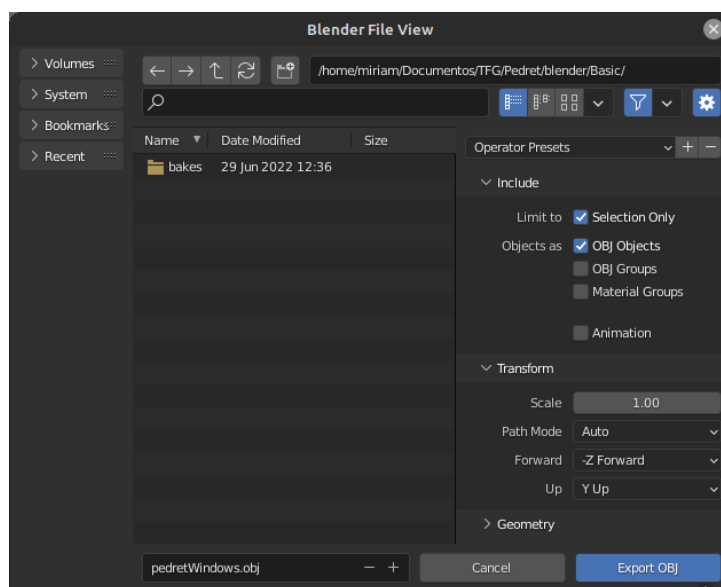


Figura 22: Exportació de l'objecte Blender a OBJ. Font: elaboració pròpia

8. Renderitzant amb Radiance

Amb el terminal obert, es va executar el convertidor de OBJ a rtm de Radiance amb la següent comanda:

```
obj2mesh -a lib/mat/pedret1.mat lib/obj/pedret1.obj lib/obj/pedret1.rtm # pedret scene 1
obj2mesh -a lib/mat/pedret2.mat lib/obj/pedret2.obj lib/obj/pedret2.rtm # pedret scene 2
obj2mesh -a lib/mat/lwindow.mat lib/obj/window.obj lib/obj/lwindow.rtm # leather window
obj2mesh -a lib/mat/gwindow.mat lib/obj/window.obj lib/obj/gwindow.rtm # galss window
```

Com tot va funcionar correctament, es van crear en el directori obj els 4 fitxers: *pedret1.rtm*, *pedret2.rtm*, *lwindow.rtm* i *gwindow.rtm*.

8.3.1 DEFINICIÓ DE LA GEOMETRIA

Els quatre fitxers anomenats anteriorment són essencials per a poder generar la geometria en Radiance. Cadascun d'aquests van ser definits en la subcarpeta rad, que és la contenidora de tots els objectes en format Radiance.

Pedret (monomaterial): *lib/rad/pedret1.rad*

```
void mesh pedret1
1 lib/obj/pedret1.rtm
0
0
```

Codi 5: Definició de la geometria de Pedret monomaterial. Font: elaboració pròpia

Pedret (multimaterial): *lib/rad/pedret2.rad*

```
void mesh pedret2
1 lib/obj/pedret2.rtm
0
0
```

Codi 6: Definició de la geometria de Pedret multimaterial. Font: elaboració pròpia

Finestra de cuir: *lib/rad/lwindow.rad*

```
void mesh window
1 lib/obj/lwindow.rtm
0
0
```

Codi 7: Definició de la geometria de la finestra de cuir. Font: elaboració pròpia

Finestra de vidre: *lib/rad/gwindow.rad*

```
void mesh window
1 lib/obj/gwindow.rtm
0
0
```

Codi 8: Definició de la geometria de la finestra de vidre. Font: elaboració pròpia

A continuació, es va fer una renderització bàsica dels objectes per a comprovar que tot es visualitzava correctament, i que no hi hagués cap anomalia en la previsualització.

8.3.2 PREVISUALITZACIÓ

Les previsualitzacions en Radiance es realitzen executant el programa rvu. No obstant, existeixen programes que l'utilitzen internament com l'objview.

L'objectiu del programa objview és visualitzar els objectes definits en format Radiance. Aquest programa és molt útil per a comprovar que no hi hagi incongruències en la visualització de models, abans de crear la definició d'il·luminació o d'ajustar paràmetres de renderització.

8. Renderitzant amb Radiance

Convé ressaltar que aquest programa agrega la il·luminació de manera automàtica, a través de paràmetres de renderització que també són ajustats automàticament pel programa, a fi de què l'escena³ no es visualitzi uniformement de color negre.

Amb les següents dos ordres es va comprovar que tot el procés fins aquell moment s'havia realitzat correctament, i que els objectes en Radiance no mostraven cap irregularitat:

```
$ objview lib/rad/pedret1.rad lib/rad/lwindow.rad  
$ objview lib/rad/pedret2.rad lib/rad/gwindow.rad
```

Executant la primera línia d'ordres es va visualitzar Pedret (monomaterial) amb la finestra de cuir (Figura 23a) i amb la segona ordre es va visualitzar Pedret multimaterial amb les finestres de vidre (Figura 23b).



(a) Pedret monomaterial amb finestres de cuir. Font: elaboració pròpia



(b) Pedret multimaterial amb finestres de vidre. Font: elaboració pròpia

Figura 23: Visualització de Pedret amb el programa objview. Font: elaboració pròpia

Com es pot observar en les imatges anteriors, les dues visualitzacions són molt similars. Les diferències principals són el color de les finestres: la de cuir és marró i la de vidre és transparent. A més a més, la sensibilitat dels materials a les condicions d'il·luminació ja són apreciables. De fet, en la Figura 23b, es pot apreciar l'efecte de la il·luminació que entra per les finestres.

Durant aquest procés de previsualització es van utilitzar diverses comandes per observar i inspeccionar l'escena. En la Taula 15 es fa un recull de les comandes, amb els paràmetres que s'han utilitzat i una descripció breu de la seva funcionalitat. Totes aquestes s'han fet servir durant tot el projecte, però existeixen més i poden ser consultades en la documentació de rvu⁴.

³L'escena és creada de manera temporal amb els objectes esmentats després de l'objview

⁴rvu: https://floyd.lbl.gov/radiance/man_html/rvu.1.html

8. Renderitzant amb Radiance

Comandes	Descripció
move <i>n</i> [<i>x y z</i>]	Mou la càmera <i>n</i> cops més a prop respecte al punt seleccionat o especificat <i>x y z</i> .
pivot <i>d</i> [<i>u</i>]	Gira la càmera sobre un punt seleccionat amb <i>d</i> graus cap a la dreta i <i>u</i> graus cap amunt.
rotate <i>d</i> [<i>u</i>]	Gira la càmera <i>d</i> graus cap a la dreta i <i>u</i> graus cap amunt.
exposure [<i>e</i>]	Ajusta l'exposició a <i>e</i> .
view [<i>filename.vf</i>]	Emmagatzema un fitxer anomenat <i>filename.vf</i> amb els paràmetres de visualització actuals.
last	Restaura el punt de vista anterior.
trace	Proporciona dades de la geometria i luminància sobre el punt seleccionat.
write [<i>filename.hdr</i>]	Captura l'escena i guarda la imatge amb el nom <i>filename</i> .
quit	Sortir del programa.

Taula 15: Comandes utilitzades en el programa rvu. Els paràmetres entre claudators ([]) són opcionals. Font: elaboració pròpia

Un cop es van obtenir aquestes dues escenes, on tots els objectes que es troben en elles eren visualment correctes, es va procedir a afegir la il·luminació.

8.4 LLUM NATURAL

La il·luminació emprada en el projecte és principalment llum natural, la qual actua com a il·luminació global, i és variable d'acord amb el CIE Standard General Sky, o també anomenat CIE. El CIE és bàsicament un model matemàtic que simula la llum diürna tenint en compte una sèrie de condicions meteorològiques diferents, com mostra la Figura 24.

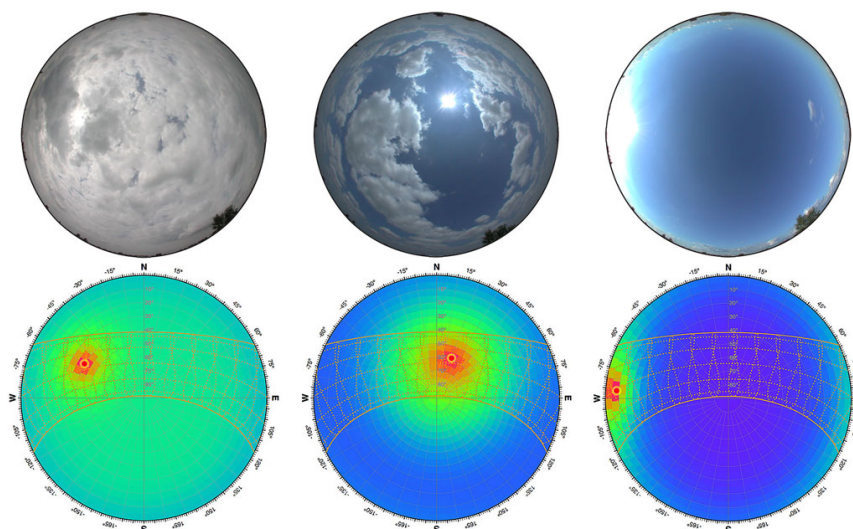


Figura 24: Exemples de condicions reals de cel amb els models CIE. Font: [33].

gensky⁵ produeix una descripció de l'escena Radiance pel CIE en el mes, el dia i l'hora especificats. Aquest programa pot produir quatre distribucions de luminància del cel:

sunny sky without sun: Cel assolellat. És correspon al model CIE clear day. S'executat amb el paràmetre *s*.

intermediate sky: Cel parcialment ennuvolat. Es correspon al model CIE intermediate day. S'executa amb el paràmetre *i*.

⁵gensky: https://floyd.lbl.gov/radiance/man_html/gensky.1.html

8. Renderitzant amb Radiance

cloudy sky: Cel ennuvolat. Es correspon al model CIE overcast day. S'executa amb el paràmetre *c*.

uniform cloudy sky: Cel ennuvolat uniforme. La distribució del cel és completament uniforme. S'executa amb el paràmetre *u*.

Aquests models de cel poden ser controlats amb els paràmetres de radiació zenital *-b* o la irradiància difusa horitzontal *-B*, que canviaran la distribució de luminància del cel.

Els models *sunny* i *intermediate*, permeten crear manualment la descripció del Sol en Radiance, executant el paràmetre *s* o *i* amb un *-* al davant. Però si hi ha un *+* al davant del paràmetre la descripció del Sol es crea automàticament per Radiance. En aquest cas la radiància solar és subministrada directament pel programa o es calcula a partir de la irradiància directa horitzontal, si ha estat especificada amb el paràmetre *-R*.

En la meua opinió, si no es tenen coneixements de meteorologia, és millor utilitzar un dels models CIE estàndards, que creen automàticament la descripció de Sol en l'escena Radiance, i ens permeten definir la posició del Sol, a partir de l'altitud i l'azimut o a partir del temps i les coordenades geogràfiques.

La renderització d'aquest projecte té un propòsit il·lustratiu, per tant, resulta indiferent la distribució. No obstant, es van fer diverses proves per decidir quina distribució utilitzar. Al començament es volia mostrar en la memòria la renderització de Sant Quirze de Pedret amb les pitjors condicions d'il·luminació, per tal de poder estudiar quins eren els valors mínims d'il·luminació a l'interior de Pedret; però més endavant, a partir dels resultats obtinguts, es va veure que no tenia massa sentit, perquè no eren resultats molt il·lustratius i, a més, eren difícils d'analitzar, ja que l'exposició (quantitat de llum que entrava a través de la càmera) influïa massa, aconseguint una sobreexposició o subexposició en la imatge⁶.

Per tant, en comptes d'estudiar una imatge, es va decidir mostrar tres renderitzacions diferents respecte el temps, en un determinat dia i mes, per visualitzar com és la línia temporal d'il·luminació en l'interval d'11:00 a 15:00, que és quan hi ha més intensitat lluminosa.

En Radiance aquest valor del temps es pot donar com a hores decimals o utilitzant dos punts per separar hores i minuts. En els dos casos és necessari especificar la zona horària o bé el meridià amb l'opció *-m*.

La simulació que es mostra en la memòria es va fer en la zona horària Central European Time (CET)⁷, el dia 1 d'agost de 1960, en les diferents hores que es mostren en el codi a continuació.

Sabent que les coordenades geogràfiques⁸ de Pedret són: 42.1073 graus de latitud i 1.8835 graus de longitud. Les ordres de *gensky* van ser les següents:

```
!gensky 1 8 11:00CEST -y 1990 -a 42.1073 -o 1.8835 +s
```

⁶L'exposició de les imatges renderitzades, van ser ajustades principalment amb GIMP i algunes amb el programa Luminance.hdr

⁷Llistat de zones horàries per països: https://en.wikipedia.org/wiki/List_of_time_zones_by_country

⁸Coordenades obtingudes amb Google Maps: <https://www.google.es/maps/place/42%C2%B006%2726.3%22N+1%C2%B053%2700.6%22E/@42.1073,1.881306,17z/data=!3m1!4b1!4m5!3m4!1s0x0:0xeb205b78bc09ab57!8m2!3d42.1073!4d1.8835?hl=es>

8. Renderitzant amb Radiance

```
!gensky 1 8 13:00CEST -y 1990 -a 42.1073 -o 1.8835 +s
!gensky 1 8 15:00CEST -y 1990 -a 42.1073 -o 1.8835 +s
```

8.4.1 DEFINICIÓ DEL CEL

Malauradament l'execució de les ordres `gensky` no són suficient, perquè només contenen la descripció del Sol i la distribució del cel. Per tant, s'ha de definir el cel i un terra genèric per a tota l'escena, el qual actuarà com a geometria.

Sky: *lib/rad/sky.rad*

```
!gensky 1 8 12:00CEST -y 1990 -a 42.1073 -o 1.8835 +s

# sky definition
skyfunc glow sky_glow
0
0
4 .9 .9 1.15 0

sky_glow source sky
0
0
4 0 0 1 180

# ground definition
skyfunc glow ground_glow
0
0
4 1.4 .9 .6 0

ground_glow source ground
0
0
4 0 0 -1 180
```

Codi 9: Definició del cel en Radiance. Font: elaboració pròpia

La primera línia del Codi 9 especifica la ordre de la distribució del cel, i la resta la geometria.

L'`skyfunc glow` va ser utilitzat per auto-il·luminar el cel, sense il·luminar els objectes pròxims. El del cel va ser anomenat `sky_glow`, el qual va ser acolorit blanc amb una saturació de blau, i l'últim paràmetre que té com a valor 0, indica que no participa en l'ombrejat. El del terra va ser anomenat `groud_glow`, es va definir de manera similar al anterior, però acolorit de color de marró amb una saturació de vermell.

El `source` va ser utilitzat per especificar que la font de llum es troba infinitament distant. Els tres primers paràmetres especifiquen la direcció (0,0,1) és UP, i (0,0,-1) és DOWN, l'altre paràmetre especifica l'angle de la font de llum.

8. Renderitzant amb Radiance

8.4.2 REDEFINICIÓ DE LES FINESTRES

Anteriorment es van definir les finestres (Apartat 8.2.5), però en aquestes definicions l'efecte de la llum que entra per la finestra no s'observa. En Radiance hi ha certes superfícies que actuen com a fonts de llum, com `light`, `illum`, `spotlight` i `glow`, mentre que altres com el `plastic` i el `glass` no actuen com a fonts de llum.

Perquè la finestra il·luminés directament l'església es va modificar `lwindow.rad` amb el Codi 10 i `gwindow.rad` amb el Codi 11, agregant el tipus `illum`, el qual està especialment dissenyat per a fonts de llum secundàries, com les finestres i altres objectes brillants, que no són només emissors, sinó que tenen altres propietats visuals importants.

Finestra de cuir: `lib/rad/lwindow.rad`

```
!cat ../mat/lwindow.mat

skyfunc brightfunc window_dist
2 winxmit winxmit.cal
0
0

window_dist illum window_illum
1 window
0
3 .459 .459 .459

window_illum mesh windows
1 lib/obj/lwindow.rtm
0
0
```

Codi 10: Definició millorada de la finestra de cuir en Radiance. Font: elaboració pròpia

Finestra de vidre: `lib/rad/gwindow.rad`

```
!cat ../mat/gwindow.mat

skyfunc brightfunc window_dist
2 winxmit winxmit.cal
0
0

window_dist illum window_illum
1 window
0
3 .88 .88 .88

window_illum mesh windows
1 lib/obj/gwindow.rtm
0
0
```

Codi 11: Definició millorada de la finestra de vidre en Radiance. Font: elaboració pròpia

Aplicant aquesta modificació les visualitzacions que es van aconseguir eren més realistes, ja que l'`illum` participa en el càlcul de la renderització actuant com a font de llum, però quan la càmera apunta directament a l'objecte, apareix amb el material que va ser definit anteriorment (abans d'aplicar la modificació).

En la primera línia es va definir el material de la finestra, que va ser reutilitzat dels fitxers `*.mat` de l'objecte. L'`skyfunc brightfunc` és la funció de distribució de les finestres, que és la distribució del cel modificada amb la funció de transmitància de finestra anomenada `winxmit.cal`.⁹ Aquesta definició es convertida a modificador en `window_illum`, la qual irradiarà la llum invisible a l'interior de l'església.

En el `window_illum` del vidre es va utilitzar una transmitància del 88% que és la estàndard dels vidres amb 96% de transmissivitat. En el cas del cuir es va utilitzar una aproximació matemàtica a partir de la Formula 11, definida en [29].

⁹El `winxmit.cal` és un fitxer que pertany a la llibreria Radiance instal·lada.

8. Renderitzant amb Radiance

$$T_s = \frac{(\sqrt{0.8402528435 + 0.0072522239 * T_n^2} - 0.9166530661) * T_n}{0.0036261119} \quad (11)$$

Aquesta fórmula calcula la transmissivitat (T_s) a partir de la transmitància (T_n), tanmateix va ser utilitzada per a calcular aproximadament el valor de transmitància del cuir a partir de la transmissivitat 0.5 definida, amb el qual es va obtenir un valor proper amb transmitància 0.459.

8.5 LLUM ARTIFICIAL

En Radiance hi ha diferents maneres de crear llum artificial, però jo explicaré dos mètodes que es basen en fitxers IES i lampcolor.

8.5.1 LAMP COLOR

El lampcolor és un programa interactiu de Radiance que serveix per a calcular el color amb la radiació adequada d'un emissor difús, a partir del flux lluminós (mesurat en lumens) i d'una descripció senzilla de la geometria de l'aparell.

En aquest càlcul s'utilitza una taula de colors de llum i el *depreciation factor* per a obtenir el color d'un tipus de làmpada específic.¹⁰

Tenint en compte la senzillesa de l'escena, s'han ubicat nou espelmes en l'absis, simulant una il·luminació semblant a la de l'espelma: llum suau, baixa potència i colors càlids.

Després d'explorar els diferents tipus d'il·luminació artificial que es mostraven a la taula (entre els quals no es trobava el foc). En el fòrum Radiance[34] es va trobar una persona que va derivar el color rgb de la flama a través de la temperatura de color¹¹. A l'estar validada per l'autor del llibre Rendering with Radiance[14], em vaig decantar per utilitzar-la en la simulació de les espelmes.

Per tant, la flama de la làmpada d'oli, va ser la definició que més em va convèncer per a representar la flama de l'espelma. Cada espelma emet una candela que, segons la referència [35], són 12.57 lumens.

Aquestes van ser totes les dades necessàries per a calcular la resplendor de la geometria, treta del catàleg d'IKEA[36], la qual ha estat reproduïda en Blender per poder-la exportar a Radiance. Les proporcions de l'espelma són de 10cm de llarg i 7cm de diàmetre. S'ha calculat la superfície de la flama a partir d'una estimació feta en Blender.

```
$ lampcolor
Enter lamp type [WHITE]: oil lamp
Enter length unit [meter]:
Enter lamp geometry [polygon]:
Polygon area [1]: 0.014696902
```

¹⁰La taula utilitzada per lampcolor es troba aquí: <https://github.com/NREL/Radiance/blob/3aff252e57e6d2ca9205cf7caf20aaa1a897aaf2/src/cv/lamp.tab>

¹¹La temperatura de color és la distribució espectral de la energia d'una font lluminosa i és expressada en Kelvin.

8. Renderitzant amb Radiance

```
Enter total lamp lumens [0]: 12.57
Lamp color (RGB) = 3.109069 0.926186 0.000000
```

Es va crear un fitxer nou anomenat `lights.mat`, i es va situar al directori `lib/mat/` on es disposarà el material de l'espelma.

```
void light flame                void plastic cylinder
0                                0
0                                0
3 0.045694 0.013612 0.000000    5 .9 .9 .9 0 0
```

Codi 12: Definició del material de l'espelma en Radiance. Font: elaboració pròpia

Després es va linkar amb `obj2mesh`:

```
obj2mesh -a lib/mat/light.mat lib/obj/candle.obj lib/obj/candle.rtm
```

Per a poder crear la definició de la geometria en Radiance que es va situar a `lib/rad`:

```
void mesh candle
1 lib/obj/candle.rtm
0
0
```

Codi 13: Definició de la geometria de l'espelma en Radiance. Font: elaboració pròpia

Aplicant el procés de renderització que s'explicarà més endavant (Apartat 8.6), es va obtenir el resultat mostrat en la Figura 25. Com es pot observar a la imatge, la il·luminació de l'espelma és gairebé imperceptible, perquè la potència aproximada en Watts de les espelmes no és notòria (només afecta les coses que estan molt a prop). Així que es va decidir fer una altra simulació de l'escena, però amb un fitxer IES d'il·luminació actual.



Figura 25: Renderització d'espelmes a Radiance. Font: elaboració pròpia

8.5.2 FITXERS IES

Els fitxers IES són el millor mètode per simular la il·luminació artificial, ja que utilitza les dades de distribució de llum, a diferència de `lampcolor`. De fet, la majoria d'objectes que difonen llum no són realment difosos.

Radiance inclou el programa `ies2rad` per convertir la fotometria de la font de llum feta per Illuminating Engineering Society of North America (IESNA) a Radiance.

El programa `ies2rad` crea dos fitxers que reben les extensions `.dat` i `.rad`. El fitxer `.dat` conté una taula de dades que especifiquen la distribució de la llum en format Radiance. Tots dos fitxers van ser guardats en el mateix directori `lib_ies`.

Per a mostrar com funciona el mètode que utilitza fitxers IES es va simular la fotometria d'un reflector PAR38, el qual es va convertir en un fitxer Radiance anomenat `type_s1` amb la següent comanda:

8. Renderitzant amb Radiance

```
$ ies2rad -df -t defaults -m .85 -c 1 1 1 -o type_S1 lib_ies/lamp892lm14w.ies
```

El nom del fitxer es va establir amb l'opció -o. La simulació del color i la intensitat de la llum, amb els paràmetres -c, que indica el color, en el qual pot participar -t (si el paràmetre és un tipus d'il·luminació descrit en la taula lampcolor, com l'halògena o incandescent) o no participar.

El paràmetre -m també influeix en la intensitat, ja que indica el *depreciation factor*, que com s'ha dit abans, és un factor molt important perquè la sortida de llum inicial, de la majoria de les làmpades, no és la mateixa que la sortida de llum cap al final de la seva vida útil.

Finalment, es va col·locar en l'escena la llum convertida a Radiance i en aplicar la renderització es va obtenir la següent imatge:



Figura 26: Renderització amb il·luminació IES. Font: elaboració pròpia

8.6 RENDERITZACIÓ

Per a la renderització d'una escena Radiance és necessari descriure l'escena. Aquesta reunirà tots els materials i la geometria que volem que apareguin en ella. Com a tall d'exemple es mostra el Codi 14, pertanyent a l'escena gwindow1, la qual incorpora el model de Pedret d'un únic material, la finestra de vidre i el cel.

Escena: *scenes/gwindow1/scene.rad*

```
!xform lib/rad/sky.rad
!xform lib/rad/pedret1.rad
!xform lib/rad/gwindow.rad
# !xform -n lamp1 -t 9 11 4 lib_ies/type_S1.rad
# !xform -n candle1 -t -2.8231 4.5072 -1.6 lib/rad/candle.rad
# !xform -n candle1 -t -2.8231 4.6167 -1.6 lib/rad/candle.rad
```

Codi 14: Codi de l'escena gwindow1. Font: elaboració pròpia

8. Renderitzant amb Radiance

Tots els objectes estan centrats en el punt (0,0,0) de l'escena. Per tant, a alguns objectes com les espelmes i el reflector, que es mostren comentats en el codi, se'ls van aplicar paràmetres de transformació, concretament de translació. A més a més, es va reanomenar *lib_ies/type_S1.rad* a *lamp1* (que actua com a nom identificador de l'escena), per tal de no generar conflictes en el fitxer que s'està definint (*scene.rad*), i poder tenir diferents *lib_ies/type_S1.rad*. La principal raó d'aquest canvi de noms es permetre posteriorment afegir més reflectors.

La col·locació d'objectes Radiance va ser realitzada de manera idèntica a totes les escenes:

- Pedret1: Conté Pedret monomaterial.
- Pedret2: Conté Pedret multimaterial.
- Gwindow1: Conté Pedret monomaterial i les finestres de vidre.
- Gwindow2: Conté Pedret multimaterial i les finestres de vidre.
- Lwindow1: Conté Pedret monomaterial i les finestres de cuir.
- Lwindow2: Conté Pedret multimaterial i les finestres de cuir.

La càmera de l'escena es va situar utilitzant Blender, i mitjançant l'execució de l'script de conversió de la càmera, el qual s'explicarà més endavant a l'Apartat 9.2, es van obtenir els paràmetres de la càmera en el format Radiance.

Aquests paràmetres eren el *-vu* (view up), el *-vd* (view direction) i el *-vp* (view point).

Les renderitzacions van ser generades amb *rad*, que és un programa versàtil que et permet crear la imatge o interactuar amb ella. Aquest programa automatitza el procés de rendering utilitzant fitxers de control que tenen l'extensió *.rif*.

El fitxer de control conté una llista de variables, generalment una per línia; les variables que poden tenir múltiples valors són escrites en minúscules i les que només poden tenir un valor en majúscules.

L'execució determina com es fa la renderització: *rad -o X11 file.rif* executa el fitxer interactivament, utilitzant internament el programa *rvu*; i *rad file.rif* emmagatzema la imatge on especifiqui el fitxer, utilitzant internament *rpict*.

La primera opció és útil per moure'ns per l'escena, a través de les comandes ja mostrades en la Taula 2, i també per a guardar les diferents posicions de la càmera amb *view nice.vf*.

La segona opció és útil per a representar l'escena de manera definitiva, és a dir, detalladament usant càlculs indirectes, precisos i acurats. La imatge generada es pot visualitzar utilitzant el programa *ximage*.

Els fitxers de control que es van crear per a cadascuna de les escenes seguien el següent esquema:

```
OCTREE=scenes/pedret1/scene.oct # fitxer binari de l'escena
AMB=scenes/pedret1/scene.amb # fitxer que conte el calcul ambient (memoria cau)
scene=scenes/pedret1/scene.rad # fitxers escena a convertir en Octree
```

8. Renderitzant amb Radiance

```
EXPOSURE = 1 # exposicio
VARIABILITY = High # variabilitat
DETAIL = High # detall
QUALITY = High # qualitat

INDIRECT = 8 # llum indirecta
PENUMBRAS = True # penombra
PICTURE = scenes/pedret1/img/lights20:00 # nom de la imatge
RESOLUTION = 512 512 # resolucio
# render=-i

# Cameres utilitzades per la renderitzacio
view= door -vf lib_view/door.vf
view= window -vf lib_view/window.vf
view= entrance -vf lib_view/entrance.vf

REPORT=1 # Informe del proces de renderitzacio
```

Com les imatges eren per a la memòria, es van crear amb qualitat òptima, independentment del temps de comput¹².

8.6.1 ANÀLISI DE LA IL·LUMINACIÓ

A efectes pràctics per al nostre context històric, aquest projecte s'ha enfocat principalment en la il·luminació natural, és per aquest motiu que les renderitzacions que es mostren a continuació només tindran en compte la llum natural i no l'artificial.

Una de les propietats de la propagació de la llum més evidents són les ombres, creades quan la llum és obstaculitzada. Com més petit és l'angle entre la direcció de la llum i un objecte allargat que l'obstaculitzi, més curta serà la seva ombra. D'altra banda, com més petit sigui l'angle entre la direcció de la llum i la superfície en la qual apareix l'ombra, més llarga serà aquesta. Si l'objecte està a prop de la font lluminosa, l'ombra serà major que si l'objecte es troba lluny. Si la superfície està corbada, hi haurà més distorsions.

Cal recordar, que la transmissió de la llum de l'objecte també és afectada per les propietats del material de l'objecte. La Figura 27 és un clar exemple. En la primera fila (Figura 27a, 27b, 27c), la transmissió difusa de la llum és dispersada en diverses direccions i, en conseqüència, aquesta és més suau i menys intensa, de manera que les ombres són menys nítides i les imatges més clares que les de la segona fila (Figura 27d, 27e, 27f). De fet en aquestes darreres imatges, en la fusta de la teulada i en l'esglaó de la porta, es pot veure una part de llum blanca que és transmesa, en canvi, en les de la primera fila, aquesta llum és absorbida pel material. Aquesta figura és un exemple de la importància d'una bona definició dels materials per aconseguir uns resultats de qualitat.

¹²Les instruccions per renderitzar Sant Quirze de Pedret es troben en l'Annex B.

8. Renderitzant amb Radiance



(a) Pedret1, visualització de la porta.
Font: elaboració pròpia



(b) Pedret1, visualització de les finestres.
Font: elaboració pròpia



(c) Pedret1, visualització de l'absis.
Font: elaboració pròpia



(d) Pedret2, visualització de la porta.
Font: elaboració pròpia



(e) Pedret2, visualització de les finestres.
Font: elaboració pròpia



(f) Pedret2, visualització de l'absis.
Font: elaboració pròpia

Figura 27: Comparació de les renderitzacions de Pedret1 i Pedret2. Font: elaboració pròpia

A la simulació feta incloent les finestres de vidre (Figura 28), la llum travessa un objecte transparent que transmet llum. Aquesta transmissió es pot considerar com una doble refracció: la llum pateix una primera refracció en passar de l'aire al vidre, segueix el seu camí i torna a refractar-se en tornar a passar a l'aire. Per aquest motiu, ara l'especularitat (part blanca) abans mencionada, es troba en punts diferents, respecte a les anteriors imatges. Per exemple, en la Figura 27d l'especularitat es troba en el centre de l'esglaó, mentre que en la Figura 28d es localitza en els laterals de l'esglaó. Pel que fa a Pedret1 i Pedret2, tenen les mateixes diferències que el comportament anterior, però el contrast blanc i negre de les ombres es troba molt més pronunciada que en la Figura 27, degut a que les finestres absorbeixen llum, segons les propietats que es van especificar en Radiance, el vidre hauria d'absorbir un 4% de la llum incident.

8. Renderitzant amb Radiance



(a) Gwindow1, visualització de la porta. Font: elaboració pròpia



(b) Gwindow1, visualització de les finestres. Font: elaboració pròpia



(c) Gwindow1, visualització de l'absis. Font: elaboració pròpia



(d) Gwindow2, visualització de la porta. Font: elaboració pròpia



(e) Gwindow2, visualització de les finestres. Font: elaboració pròpia



(f) Gwindow2, visualització de l'absis. Font: elaboració pròpia

Figura 28: Comparació de Pedret1 i Pedret2 amb finestres de vidre. Font: elaboració pròpia

A la Figura 29, es simula una finestra translúcida, per tant, no deixa passar molta llum (només la meitat de la llum que ja entrava) i l'exposició d'aquesta renderització ha estat molt incrementada. Per aquest motiu els resultats de la Figura 29, és veuen tan groguencs, perquè la il·luminació que incideix a l'escena és mínima. En aquestes condicions de tan poca il·luminació es fa difícil comprar o treure conclusions de la il·luminació.



(a) Lwindow1, visualització de la porta. Font: elaboració pròpia



(b) Lwindow1, visualització de les finestres. Font: elaboració pròpia



(c) Lwindow1, visualització de l'absis. Font: elaboració pròpia

8. Renderitzant amb Radiance



(d) Lwindow2, visualització de la porta. Font: elaboració pròpia

(e) Lwindow2, visualització de les finestres. Font: elaboració pròpia

(f) Lwindow2, visualització de l'absis. Font: elaboració pròpia

Figura 29: Comparació de Pedret1 i Pedret2 amb finestres de cuir. Font: elapròpia

Per a concloure l'estudi, es va realitzar una simulació d'il·luminació amb Pedret1, la qual va anar canviant respecte a les hores del dia, concretament de 12:00 a 15:00, de l'1 d'agost del 1990, amb una distribució de cel clar, sense núvols.

Observem la simulació en les diferents franges horàries (Figura 30):



(a) Simulació de la il·luminació a les 12:00. Font: elaboració pròpia



(b) Simulació de la il·luminació a les 13:00. Font: elaboració pròpia

8. Renderitzant amb Radiance



(c) Simulació de la il·luminació a les 14:00. Font: elaboració pròpia



(d) Simulació de la il·luminació a les 15:00. Font: elaboració pròpia

Figura 30: Pedret1, renderització amb llum natural de 12:00 a 15:00. Font: elaboració pròpia

En cadascuna de les franges horàries es pot veure com la llum que prové del Sol es va desplaçant de dreta a esquerra, a través de les reflexions i la distribució de la llum que es mostra en l'escena.

8.6.2 ANÀLISI FOTOMÈTRIC

Per a realitzar l'anàlisi físic de la il·luminació, es va utilitzar el programa `falsecolor`, el qual mapeja els valors de luminància i il·luminància de la imatge a colors. Aquesta eina és molt útil per a aquells que es dediquen a analitzar la il·luminació en interiors, per la facilitat de determinar si el nivell d'il·luminació és adequat.

L'anàlisi es va fer amb l'escena Pedret2, amb la il·luminació de les 11:00 (Figura 27d) quan entra llum. Per generar la imatge en `falsecolor`, és necessari renderitzar el model en termes d'irradiància, capturant el nivell de llum incident. Aquesta renderització ha estat creada afegint l'opció `-i` en el rendering.

A partir de la imatge d'irradiació s'utilitza el `falsecolor`, el qual es pot configurar amb les diverses opcions que ofereix aquest programa. Per exemple, l'ordre següent produeix un mapeig logarítmic en base 2 ($-\log 2$) entre 0 i 475 lux (`-s 475`) i una llegenda amb l'etiqueta Lux (`-l Lux`).

```
falsecolor -ip irradi11:00_door.hdr -s 475 -log 2 -l Lux > lux11:00_pedret2.hdr
```

La Figura 31 mostra els resultats d'aquestes operacions amb dos perspectives diferents:

8. Renderitzant amb Radiance

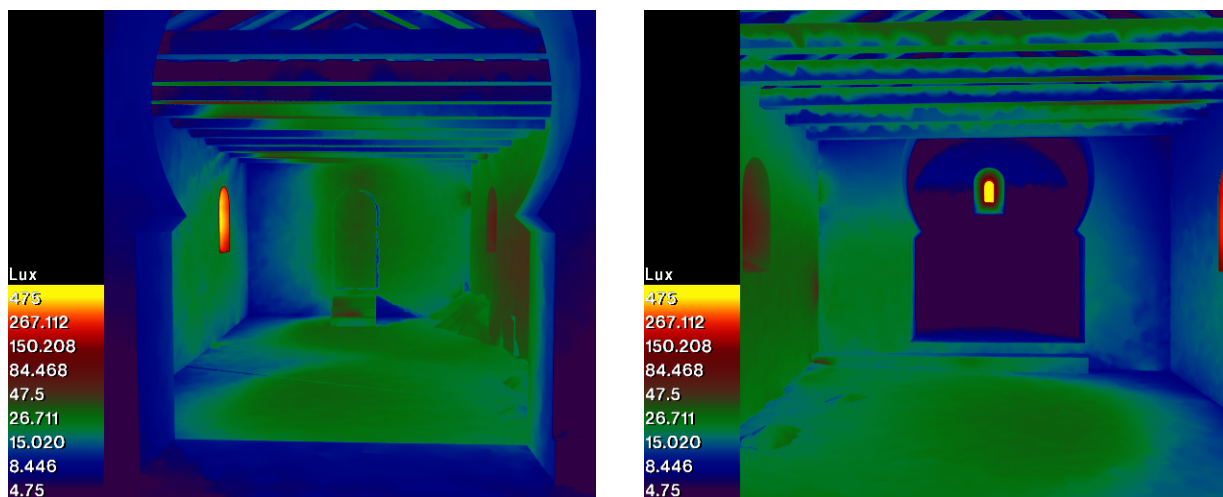


Figura 31: Anàlisi de la il·luminància. Font: elaboració pròpia

En cadascuna de les diferents perspectives es mostra quina és la quantitat de llum que incideix sobre la superfície. Les finestres són el punt de màxima incidència d'il·luminació, aproximadament 475 lux, mentre que les zones ombrades que hi havia en la renderització tenen la mínima incidència d'il·luminació, entre 4 i 10 lux.

Personalment, considero que el resultat obtingut d'il·luminació és baix, degut a la petita mida que tenen les finestres de les esglésies romàniques, tenint en compte que la majoria de la zona es troba entre els 20 i 40 lux, valors que es troben per sota dels 50 lux, que és el valor mínim permès pels vestuaris i els lavabos de centres docents. Com a zona de circulació i de pas, no està tan malament, però per a fer tasques considero que hi ha molt poca il·luminació, i que hauria de ser reforçada amb algun tipus d'il·luminació artificial.

8.6.3 TEMPS D'EXECUCIÓ

Per a cada escena es van fer moltes simulacions, i de totes aquestes, s'ha decidit mostrar taules amb els temps d'execució de les renderitzacions que es troben en aquesta memòria: la simulació de les diferents configuracions de Pedret el dia 1 d'agost del 1990 a les 11:00 (Taula 16); i la simulació temporal de gwindow1, és a dir, les diferents il·luminacions que va haver-hi durant l'1 d'agost del 1990, entre les 11:00 i les 15:00 a l'escena de Pedret monomaterial amb vidres (Taula 17).

Els temps escrits s'han extret del CPU time (temps que l'ordinador dedica exclusivament al nostre programa) dels informes de renderització del programa `rpict`, que són mostrats en la consola mentre es crea la renderització.

Tots els temps mostrats en les dues taules pertanyen a la primera execució de renderització de l'escena, amb el mateix punt focal `door.vf` (el qual visualitza la porta).

	Pedret1	Pedret2	Gwindow1	Gwindow2	Lwindow1	Lwindow2
11:00	9h 29min	7h 17min	11h 55min	11h 13min	18h 32min	12h 42min

Taula 16: Temps d'execució de les diferents escenes a les 11:00. Font: elaboració pròpia

8. Renderitzant amb Radiance

	11:00	12:00	13:00	14:00	15:00
Gwindow1	11h 55 min	3h 55 min	12h 26min	9h 33min	6h 43min

Taula 17: Temps d'execució de gwindow1 en les diferents franges horàries. Font: elaboració pròpia

Cal subratllar que aquests són els temps que va trigar a executar-se en el meu ordinador, i segurament molts factors van influenciar en l'elevat temps d'execució com, per exemple, la velocitat i el nombre de processadors, la quantitat de RAM, la GPU, les elevades temperatures de l'estiu, etc.

No és d'estranyar haver hagut d'esperar tant de temps, ja que en ser la primera execució s'ha de crear tant l'Octree (.oct), com el fitxer ambient (.amb), que és un arxiu que guarda la il·luminació ambient de l'escena i actua com a fitxer de memòria cau, i que pot ser reutilitzat quan es vulgui visualitzar l'escena amb una càmera diferent.

A més a més, la precisió de l'algorisme i la configuració dels paràmetres en òptima resolució, també han influenciat en l'elevat cost de computació.

Les especificacions tècniques del ordinador MSI PS42 8RB (equip utilitzat) queden resumides en la Taula 18.

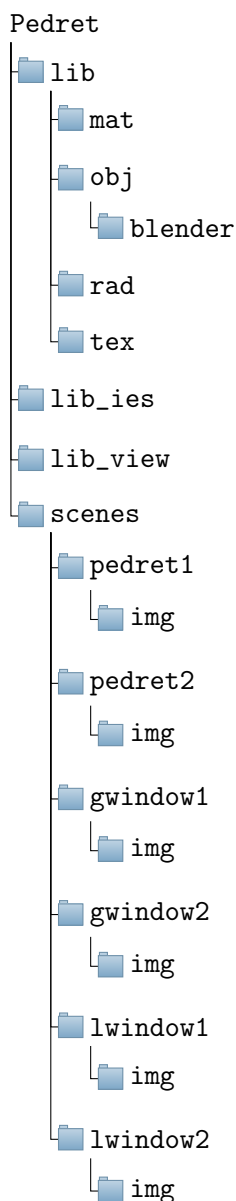
Especificacions tècniques	
Sistema operatiu	Linux Mint 21
CPU	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1 physical processor; 4 cores; 8 threads
RAM	16GB (15,9GB usable)
GPU	Intel Corporation UHD Graphics 620 (rev 07)
Proveïdor X11	The X.Org Foundation

Taula 18: Especificacions tècniques del MSI PS42 8RB. Font: elaboració pròpia

8. Renderitzant amb Radiance

8.7 JERARQUIA D'ARXIS FINAL

Sant Quirze de Pedret ha quedat construït, il·luminat, previsulitzat i analitzat en l'entorn de Radiance de manera manual. En el procés final es va dividir la carpeta principal, Pedret, amb les següents subcarpetes:



lib: és la llibreria de tots els possibles objectes que pot contenir l'escena. Dintre d'aquest directori es troba:

- **mat:** conté totes les definicions de materials de Radiance, identificades amb l'extensió `.mat`.
- **obj:** conté tots els fitxers `.obj` exportats des de l'escena Blender, que es troba guardada en la carpeta Blender, juntament amb les *backed textures* que es van generar. A més a més, en la carpeta `obj`, també es troben tots els fitxers `.rtm` (el `.obj` en format Radiance ja amb els materials associats de la carpeta `mat`).
- **rad:** conté totes les definicions de la geometria Radiance, ja preparades per unir-les de manera simple, en les descripcions de l'escena Radiance.
- **tex:** conté les textures dels materials en format Radiance, és a dir, en `.hdr`.

lib_view: és la llibreria de tots els punts focals definits, és a dir, aquí es guardaven les diferents col·locacions de les càmeres a l'hora de renderitzar l'escena.

lib_ies: és la llibreria de les fonts de llum IES, és a dir, aquí es guardaven les diferents fonts d'il·luminació artificial que es volien col·locar a l'escena de Pedret.

scenes: conte el recopilatori de totes les escenes, i les seves renderitzacions. Cadascuna conté una subcarpeta on es guarden les imatges que s'han generat a partir de les diferents renderitzacions realitzades.

- **pedret1:** escena de Pedret configurada amb un únic material i sense finestres.
- **pedret2:** escena de Pedret configurada amb múltiples materials i sense finestres.
- **gwindow1:** escena de Pedret configurada amb un únic material i finestres de vidre.
- **gwindow2:** escena de Pedret configurada amb múltiples materials i finestres de vidre.
- **lwindow1:** escena de Pedret configurada amb un únic material i finestres de cuir.
- **lwindow2:** escena de Pedret configurada amb múltiples materials i finestres de cuir.

8. Renderitzant amb Radiance

Tot el contingut dels directoris: codi Radiance, renderitzacions, fitxers IES, etc.; es pot trobar en [aquest](#) repositori de GitHub, a excepció de la carpeta `lib/obj/` i `lib/tex/`, les quals no han pogut ser publicades, perquè els models de Pedret només poden ser distribuïts en el context de docència i recerca, no al públic general¹³.

Les instruccions generals que es van usar per a compilar les escenes Radiance, de manera fàcil i senzilla, es troben en l'Annex B.

¹³El nom de la carpeta Pedret ha estat reemplaçat pel nom de `radiance`.

9 CONVERSIÓ DE BLENDER A RADIANCE

El nostre objectiu és renderitzar Blender a través de Radiance, és a dir, transformar les dades de Blender a Radiance, per tal que es puguin crear imatges físicament acurades en l'eina de modelatge.

En aquest apartat s'explica la tasca de conversió de Blender a Radiance, que consisteix essencialment a traduir l'escena 3D de Blender al format propi de Radiance. En síntesi, implementarem un convertidor Blender-Radiance.

L'escena de Radiance és bàsicament un conjunt de geometries i materials, on la il·luminació artificial i natural també hi participen.

La renderització és creada a partir dels fitxers d'escena, juntament amb la càmera. Per tant, la tasca inicial va ser dividida en diferents subtasques. Doncs, es van fer tres sub-tasques: convertir la càmera, convertir l'escena i generar la renderització.

Durant el procés de generació d'scripts, es va utilitzar paral·lelament la consola de Python de Blender, per a la seva gran utilitat, ja que permetia provar petits trossos de codi, i tenia l'autocompletat, de manera que facilitava navegar i explorar ràpidament l'API.

Tots els scripts utilitzen el mòdul bpy ¹⁴, el qual és un bàsic pel desenvolupament d'eines en Blender, ja que proporciona accés a tots els tipus de dades definides en el programa i a les dades existents en un fitxer .blend en particular.

A més a més del mòdul os ¹⁵, que és necessari per a accedir a les funcionalitats del sistema operatiu, com escriure i guardar arxius, o executar ordres de la línia d'ordres.

L'Script final, de totes les conversions que es mostren a continuació, està adjuntat en l'Annex C. Tot el codi i els comentaris van ser escrits en anglès, ja que és el llenguatge estàndard utilitzat en els entorns de programació. A més a més, tot el codi Python utilitza les convencions estilístiques de PEP8, per tal d'aconseguir la màxima llegibilitat en els scripts.

9.1 CONVERSIÓ DE L'ESCENA

La creació del fitxer *scene.rad*, consistirà bàsicament a convertir els objectes de tipus *mesh* i *light* de Blender. A més a més, de definir aquests materials en l'entorn Radiance.

Per a aquesta conversió es va crear una classe anomenada Scene en Python, que contindrà atributs i mètodes que convertiran la geometria en fitxers .rad i els materials en fitxers .mat.

L'operació d'instanciació de la classe Scene serà creada amb un estat inicial particular: el nom del fitxer i el contingut del mateix.

```
class Scene:
    """Holds the radiance scene definitions"""
```

¹⁴Documentació del mòdul bpy: <https://docs.blender.org/api/current/index.html>

¹⁵Documentació del mòdul os: <https://docs.python.org/es/3.10/library/os.html>

9. Conversió de Blender a Radiance

```
def __init__(self, filename):
    self.filename = filename
    self.content = ''
```

Els materials de Blender van ser convertits a través de la sintaxi pròpia de Radiance, ja que els fitxers .mtl queden descartats per no descriure el material en unitats físiques. Per tant, per a tota mena de material Radiance, ha d'existir un mètode equivalent en la classe Scene. En el cas de l'església, s'han de tenir mínim aquests quatre materials definits:

```
def addMaterialLight(self, id, r, g, b):
    self.content += f"void light {id} \n 0 \n 0 \n 3 {r} {g} {b}"
    self.content += '\n'
```

```
def addMaterialGlass(self, id, r, g, b):
    self.content += f"void glass {id} \n 0 \n 0 \n 3 {r} {g} {b}"
    self.content += '\n'
```

```
def addMaterialPlastic(self, id, r, g, b, specular=0., roughness=0.):
    self.content += f"void plastic {id} \n 0 \n 0 \n 5 {r} {g} {b} {specular}\
    {roughness}"
    self.content += '\n'
```

```
def addMaterialColorTexture(self, id, textureHdr):
    self.content += f"void colorpict {id+'_map'}\n 7 red green blue {textureHdr}\
    . frac(Lu) frac(Lv) \n 0 \n 0"
    self.content += '\n'
    self.content += f"{id+'_map'} plastic {id} \n 0 \n 0 \n 5 1 1 1 0 0"
    self.content += '\n'
```

La geometria de Blender, ja es troba pràcticament transformada a Radiance, amb l'execució del programa obj2mesh, que converteix el format OBJ a rtm (format propi de Radiance). Així doncs, la geometria de Blender va ser convertida a format OBJ.

En Blender els objectes poden ser des de llums utilitzades per il·luminar l'escena, models 2D o 3D, armadures per animar els models, i fins i tot càmeres. Però Radiance només considera objecte els models, que en Blender són identificats com a un objecte de tipus *mesh*. De manera que es va forçar una selecció per traslladar tots els objectes de tipus mesh, de l'escena Blender a Radiance, convertint cadascun d'aquests en OBJ.

La conversió a OBJ es va dur a terme invocant el mètode: `bpy.ops.export_scene.obj`, que exportarà la geometria Blender a format OBJ amb un nom, que és especificat en el camp `filepath`. Perquè l'objecte quedi orientat amb el sistema de coordenades Radiance, es van establir els paràmetres: `axis_forward='Y'` i `axis_up='Z'`.

La conversió a rtm es va dur a terme executant la comanda `obj2mesh` de Radiance.

9. Conversió de Blender a Radiance

```
def blend2mesh(materials, *argv):
    """Converts the blender scene meshes to radiance meshes"""
    bpy.ops.object.select_all(action='DESELECT')
    for arg in argv:
        ob = bpy.data.objects[arg]
        if ob.type == 'MESH':
            ob.select_set(True)
            bpy.ops.export_scene.obj(filepath= f'{ob.name}.obj', use_selection=True,\
                axis_forward='Y', axis_up='Z')
            command = f'obj2mesh -a {materials} {ob.name}.obj {ob.name}.rtm'
            os.system(command)
            ob.select_set(False)
        else:
            raise Exception(f'{ob} is not a Blender mesh')
```

Quan es va obtenir tota la geometria en format Radiance (*rtm*), es va agregar el següent mètode a la classe Scene, per afegir la geometria rtm (els *mesh* de Blender) a l'escena.

```
def addMeshRtm(self, id, mat, file_rtm, xform=""):
    s = f'{mat} mesh {id} \n1 {file_rtm} \n0 \n0 \n \n'
    f = open("tmp.rad", 'w')
    f.write(s)
    f.close()

    command = f'xform {xform} tmp.rad >> {self.filename} && rm tmp.rad'
    os.system(command)
```

9.2 CONVERSIÓ DE LA CÀMERA

La conversió de la càmera de Blender a Radiance, és principalment un càlcul de vectors, bastant tediós de calcular, tret que la càmera estigui en una ubicació molt senzilla i sigui ortogonal als eixos de coordenades.

Va ser dels primers scripts implementats, i també, utilitzat durant el procés d'execució manual de Radiance.

La càmera en el format Radiance necessita definir com a mínim els tres tipus de paràmetres essencials:

- vp x y z: Col·loca la càmera en el punt (x,y,z).
- vd x y z: Apunta la càmera a la direcció (x,y,z).
- vu x y z: Aplica el vector up (x,y,z) de la càmera.

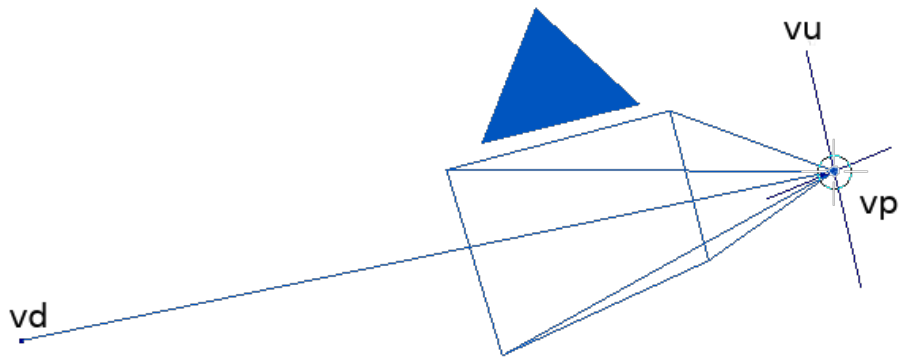


Figura 32: Càmera Blender amb els paràmetres de Radiance. Font: elaboració pròpia

La classe `Object` de Blender, permet accedir a les propietats d'un objecte arbitrari, com per exemple: la càmera, les fonts de llum, els mesh,... utilitzant el nom de l'objecte o l'índex com a clau, en el diccionari `bpy.context.scene.objects`.

Aquesta classe té diversos atributs definits. La localització de la càmera (`-vp`), es va obtenir accedint a l'atribut `location` de l'objecte. La direcció i el vector UP, es va obtenir utilitzant la matriu de rotació:

$$\begin{aligned}
 R &= \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \\
 &= \begin{bmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{bmatrix}
 \end{aligned}$$

Aquesta matriu representa una rotació tridimensional a l'espai euclidià. És utilitzada per a realitzar rotacions de vectors dels eixos x , y o z , en l'espai de tres dimensions.

En el sistema de coordenades Blender, l'eix y , és el vector Up i l'eix $-z$, és el vector Forward. De manera que el vector genèric up $(0,1,0)$ i el vector genèric de direcció $(0,0,-1)$, van ser rotats multiplicant-los per la matriu de rotació de l'objecte. Per tant, després de la rotació dels vectors, es va obtenir el vector up de l'objecte i el vector direcció de l'objecte.

Aplicat a codi Blender, les conversions dels paràmetres de la càmera, van ser represnetats de la següent manera:

```

cam = bpy.data.objects['Camera']
lx,ly,lz = cam.location # -vp
ux,uy,uz = cam.rotation_euler.to_matrix() @ Vector((0.0, 1.0, 0.0)) # -vu
dx,dy,dz = cam.rotation_euler.to_matrix() @ Vector((0.0, 0.0, -1.0)) # -vd

```

9. Conversió de Blender a Radiance

L'operador multiplicador @ està sobrecarregat de forma que rota el vector, amb les mateixes rotacions que la càmera ha estat sotmesa.

9.3 RENDERITZACIÓ

En la renderització, no es va fer cap conversió de Blender a Radiance, simplement es van cridar en Python les comandes que existeixen en Radiance: rvu, rpict i objview.

Als tres programes se'ls hi va passar, el mínim de paràmetres: la càmera i els fitxers a renderitzar. A excepció d'objview que només necessita la geometria per renderitzar. Els paràmetres per a ajustar la renderització, van ser establerts internament, dintre de la funció.

El programa que es va crear per generar la renderització va ser el següent:

```
def main():
    m = Scene("materials.mat")
    m.addMaterialColorTexture('grass', 'Pedret_IX_grass.hdr')
    m.addMaterialColorTexture('wood', 'Pedret_IX_wood.hdr', 0.07, 0.15)
    m.addMaterialColorTexture('sedimentary', 'Pedret_IX_sedimentary.hdr', 0, 0.2)
    m.addMaterialGlass('window', .96, .96, .96)
    m.save()

    blend2mesh(m.filename, 'pedret2', 'windows')

    s = Scene('geometry.rad')
    s.addMeshRtm('pedret1', 'void', 'pedret2.rtm')
    s.addMeshRtm('gwindow', 'void', 'windows.rtm')
    s.addSky(42.10745931228419, 1.8836540623509863, 1, 8, '16:00CEST', 1990)
    s.save()
    # objview(s.filename)

    view = cam2view('Door')
    rad_interact(view, s.filename)
```

10 ADDON

L'script que hem creat anteriorment, necessita unes textures prèvies, a més a més, limita la interactivitat de l'usuari a l'hora de configurar l'escena, a causa dels pocs materials definits en l'script i la necessitat de modificar paràmetres, els quals requereixen d'un usuari amb nocions bàsiques de programació. De manera, que es fa inaccessible per a molts dels nostres *stakeholders*.

El problema de les textures se soluciona amb l'automatització. No obstant, hi haurà un requisit: la textura ha de trobar-se linkada directament al BaseColor, sinó una excepció serà llançada. El segon serà solucionat definint tots els materials que es troben en la documentació de Radiance.

Per a l'últim problema, es va crear una interfície gràfica, que és integrada amb la interfície d'usuari de Blender.

Per poder portar a terme la creació de la interfície gràfica, per la nostra eina de generació d'imatges a partir de models 3D del patrimoni cultural, en primer lloc es van crear els mockups.

10.1 MOCKUPS

Els mockups es van dissenyar, ja pensant en la implementació i les diferents funcionalitats que ofereix Blender per crear la interfície. Però sempre amb la idea de què la interfície havia de ser usable i atractiva per a l'usuari.

Es van crear tres interfícies, cadascuna en diferents contextos: escena, materials i llums.

10.1.1 ESCENA

Aquest mockup (Figura 33) tenia com a objectiu renderitzar l'escena a través de preview (previsualització) o picture (imatge).

En el cas de la previsualització, els fitxers són de caire temporal, mentre que en el de la imatge és emmagatzemada, per tant, l'usuari podrà decidir com vol anomenar la renderització, a més a més, del tipus de renderització: fotomètrics o normal.

En els dos casos l'usuari podrà configurar els paràmetres de renderització: resolució, qualitat, variabilitat, detall, exposició, llum indirecta i punt focal.

Per a la generació de llum natural, des de la interfície es podrà jugar a simular diferents paràmetres del gensky, per a obtenir la llum del sol en una determinada localització, en una hora concreta d'un determinat dia de l'any

Amb aquesta definició d'interfície l'usuari podrà ajustar tots els

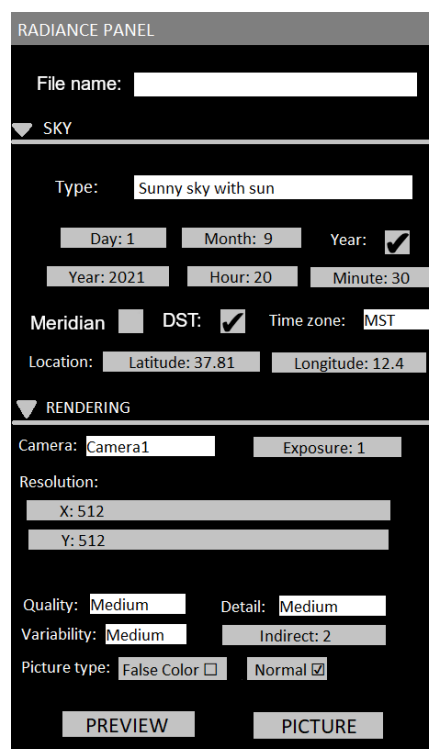


Figura 33: Mockup de l'escena. Font: elaboració pròpia

10. Addon

paràmetres de renderització i generar simulacions amb la il·luminació natural.

Totes aquestes propietats d'interfície van ser implementades en Python, definint els diferents paràmetres i els seus valors. En la Taula 19 queden resumits tots els paràmetres necessaris per a definir les funcionalitats del mockup.

Nom	Tipus	Valors	Valors per defecte	Descripció
file_name	String	-	scene	Output file name
sky_type	Enum	-	sunny sky with sun	CIE standard sky distribution
sky_day	Int	min = 1 max = 31	1	CIE standard sky distribution day
sky_month	Int	min = 1 max = 12	1	CIE standard sky distribution month
sky_time_h	Int	min = 0 max = 23	0	CIE standard sky distribution hour
sky_time_min	Int	min = 0 max = 59	0	CIE standard sky distribution minute
is_sky_time_zone	Bool	-	True	Indicates whether the time should be interpreted by the meridian or the time zone
is_sky_DST	Bool	-	False	Indicates whether the time zone is a local standard time or a daylight saving time
sky_DST	Enum	-	CEST (-2)	CIE standard sky distribution daylight saving time zone
sky_LST	Enum	-	CET (-1)	CIE standard sky distribution local standard time zone
sky_meridian	Float	min = -180 max = 180	0	CIE standard sky distribution meridian
is_sky_year	Bool	-	True	Indicates if the year is specified; if so, a more accurate solar position algorithm will be used
sky_year	Int	min = 1950 max = 2050	1950	Year to compute an accurate solar position
sky_latitude	Float	min = -90 max = 90	0	CIE standard sky distribution latitude
sky_longitude	Float	min = -180 max = 180	0	CIE standard sky distribution longitude
camera	Pointer	-	-	Camera used to render
exposure	Float	min = 0 max = inf	1	Exposure used for rendering
resolution	IntVector	min = 1 max = 1024	(512,512)	(X,Y) resolution, in pixels
quality	Enum	-	Low	The overall rendering quality
detail	Enum	-	Low	The overall level of visual detail
variability	Enum	-	Low	Indicates how much light varies over the surfaces
penumbras	Bool	-	False	Indicates if penumbras are desired or not
indirect	Int	min = 0 max = 8	0	Indicates the number of ambient bounces
indirect	Int	min = 0 max = 8	0	Indicates the number of ambient bounces
is_false_color	Bool	-	False	Indicates whether the image is rendered in false color or not
amb_file	String	-	-	Ambient file that acts as a cache

Taula 19: Propietats d'interfície de l'escena. Font: elaboració pròpia

10. Addon

10.1.2 MATERIALS

Com el material Blender no es pot convertir automàticament en materials Radiance, hauran de ser configurats una sèrie de paràmetres per a l'usuari. La interfície va ser pensada com una extensió del panell de materials de l'objecte Blender, de manera que tot objecte que tingui un material, podrà configurar també el seu material en Radiance.

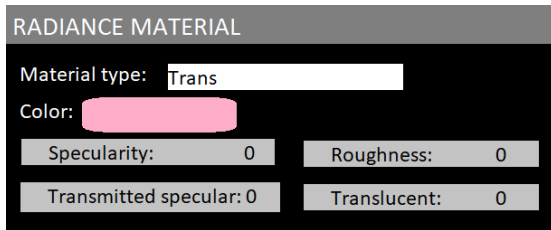
Aplicant la conversió explicada en l'Apartat 9.1, necessitarem per a cada tipus de material Radiance un mockup, que contingui els paràmetres propis de configuració de la sintaxi Radiance. Els materials convertits es mostren en la Taula 20, juntament amb la descripció que es mostrarà en la interfície i els paràmetres utilitzats.

Material	Paràmetres	Descripció
Light	color	The basic material for self-luminous surfaces (i.e., light sources).
Illum	material_type color	Used for secondary light sources with broad distributions. A secondary light source is treated like any other light source, except when viewed directly. It then acts like it is made of a different material, or becomes invisible. Secondary sources are useful when modeling windows or brightly illuminated surfaces.
Glow	color max_rad	Glow is used for surfaces that are self-luminous, but limited in their effect.
Spotlight	color angle direction	Spotlight is used for self-luminous surfaces having directed output.
Mirror	material_type color	Mirror is used for planar surfaces that produce secondary source reflections.
Plastic	color spec rough	Plastic is a material with uncolored highlights.
Metal	color spec rough	Metal is similar to plastic, but specular highlights are modified by the material color.
Trans	color spec rough trans tspec	Trans is a translucent material, whose reflection characteristics are similar to plastic. Light is also transmitted both diffusely and specularly. Transmitted and diffusely reflected light is modified by the material color.
Dielectric	color n hc	Dielectric material is transparent, and it refracts light as well as reflecting it. Its behavior is determined by the index of refraction and transmission coefficient in each wavelength band per unit length.
Glass	color	Glass is similar to dielectric, but it is optimized for thin glass surfaces ($n = 1.52$). One transmitted ray and one reflected ray is produced.
Antimatter	mod1 ... modN	Antimatter is a material that can subtract volumes from other volumes.

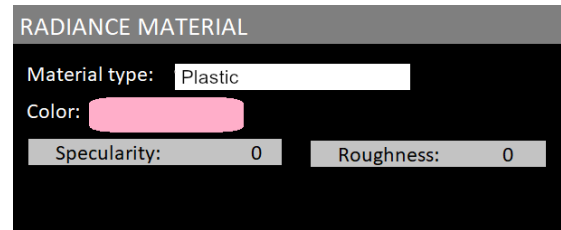
Taula 20: Materials d'interfície. Font: elaboració pròpia

La Figura 34 mostra tres mockups que es van dissenyar per als materials: *trans*, *plastic* i *glass*.

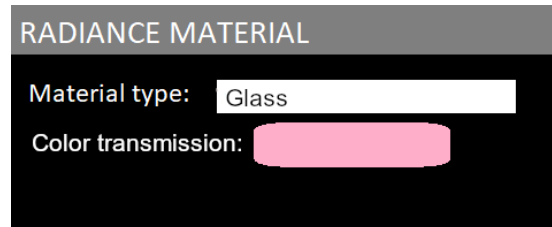
10. Addon



(a) Mockup del trans. Font: elaboració pròpia



(b) Mockup del plàstic. Font: elaboració pròpia



(c) Mockup del vidre. Font: elaboració pròpia

Figura 34: Mockups de diferents materials. Font: elaboració pròpia

En la Taula 21 queden recollides les propietats dels paràmetres utilitzats pels diferents mockups de materials.

Nom	Tipus	Valors	Valors per defecte	Descripció
material_type	Enum	-	plastic	Specifies a secondary material
is_texture	Bool	-	False	Indicates whether the color should be a texture or solid color
color	FloatVector	min = 0 max = 1	(0,0,0)	Material color
is_window	Bool	-	False	Indicates if it has window properties
transmissivity	FloatVector	min = 0 max = 1	(0,0,0)	Transmission color
spec	Float	min = 0 max = 1	0	Material specularity
rough	Float	min = 0 max = 1	0	Material roughness
trans	Float	min = 0 max = 1	0	The transmissivity is the fraction of penetrating light that travels all the way through the material
tspec	Float	min = 0 max = 1	0	The transmitted specular component is the fraction of transmitted light that is not diffusely scattered
maxrad	Float	min = -1 max = 100	0	Maximum radius for shadow testing
angle	Float	min = 0 max = 360	0	Angle in degrees (spotlight)
direction	FloatVector	min = -inf max = inf	(0,0,0)	Material (light) direction 1
n	Float		0	Index of refraction
hc	Float		0	Index of transmission
mod	Collection	-	-	Collection of material_type

Taula 21: Propietats d'interfície dels materials. Font: elaboració pròpia

10. Addon

10.1.3 LLUM

En Radiance existeixen dos programes per a definir la il·luminació: `lampcolor` i `ies2rad` (fitxers IES).

El primer de tots, no defineix una il·luminació en sí, sinó que defineix el color a partir de la radiància de l'objecte per poder-lo utilitzar en els materials d'emissió de llum de Radiance: `light`, `illum`, `glow` i `spotlight` o en els fitxers IES.

El mockup de la Figura 35 és una combinació de les dues tècniques, on el layout permet decidir a l'usuari si vol definir el color del fitxer IES amb `lampcolor` o no.

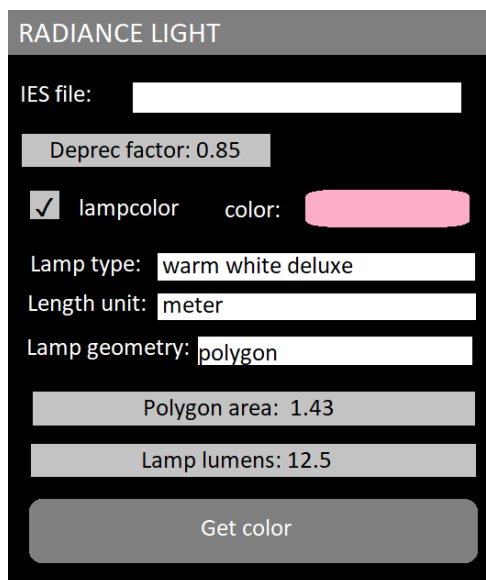


Figura 35: Mockups de les fonts de llum. Font: elaboració pròpia

Per a l'ús del `lampcolor` simplement es van extreure els paràmetres que són sol·licitats durant l'execució, que en el nostre cas tindrà lloc quan l'usuari premi el botó `Get color`, significat de què l'usuari ja ha definit de manera definitiva el valor dels paràmetres que es troben en la interfície, els quals aniran canviant depenent, de les opcions preses per l'usuari, és a dir, si l'usuari decideix canviar el tipus de geometria de polígon a cilindre, la interfície també canviarà mostrant l'altura i el radi en comptes de l'àrea.

Per altra banda, l'input del color que es troba en aquest panell, estarà sincronitzat amb l'escena Blender amb el motor de renderització Cycles, ja que és l'únic que permet utilitzar fitxers IES a Blender.

En la Taula 22, es descriuen les propietats d'interfície d'aquest mockup.

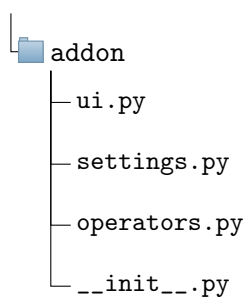
10. Addon

Nom	Tipus	Valors	Valors per defecte	Descripció
<code>is_lampcolor</code>	Bool	False	-	Indicates whether the light color should be defined with lampcolor or not
<code>IES_file</code>	String	-	-	The IES path file
<code>deprec_factor</code>	Float	min = 0 max = 1	1	Indicates the lamp depreciation factor
<code>lamp_color</code>	FloatVector	min = 0 max = 1	(1,1,1,1)	The IES lamp color
<code>lamp_type</code>	Enum	-	WHITE	The lamp type corresponds to one of the types registered in the lamp table file of Radiance. A value of WHITE means an uncolored source, which may be preferable because it results in a color balanced image
<code>lamp_unit</code>	Enum	-	meter	The length unit
<code>lamp_geom</code>	Enum	-	polygon	The lamp geometry
<code>area</code>	Float	min = 0 max = inf	0	The polygon area
<code>radius</code>	Float	min = 0 max = inf	0	The cylinder, sphere or ring radius
<code>length</code>	Float	min = 0 max = inf	0	The cylinder length
<code>lamp_lm</code>	Float	min = 0 max = inf	0	The output lamp lumens

Taula 22: Propietats d'interfície de la llum. Font: elaboració pròpia

10.2 IMPLEMENTACIÓ

La implementació ha estat dividida en 4 mòduls, configurats amb l'estructura *package* de Python, a fi de que el manteniment del codi fos més senzill i més entenedor ¹⁶.



ui.py: Implementa el disseny de la interfície d'usuari, és a dir, exposa a l'usuari els valor de les **propietats** (d'un objecte, de l'escena...) i també permet modificar aquests valors. A més a més, també poden exposar botons vinculats a accions que, en la majoria de casos, són **operadors**.

settings.py: Implementa les propietats d'interfície, és a dir, l'especificació de tots el paràmetres que s'han mostrat en les taules de la secció anterior (Apartat 10.1).

operators.py: Implementa totes les operacions, és a dir, aquí es troben totes les conversions explicades en l'Apartat 9 i les automatitzacions de l'escena Blender. Aquests operadors són simulats com a botons en la interfície d'usuari.

__init__.py: És l'arxiu principal que forma el *package* i és el que realment s'executarà en Blender. En aquest fitxer s'inclouen els tres mòduls citats anteriorment, a través de la importació.

¹⁶Veure tot el codi del package en el GitHub: <https://github.com/miriam-mendez/Pedret>

10. Addon

10.2.1 DECLARACIÓ DE PROPIETATS

En el context de l'API de Blender, els atributs de les classes se'ls anomena propietats. Aquestes pertanyen a tipus de dades existents a Blender definides a l'API sota el mòdul `bpy.props`.

A l'hora de crear una propietat s'ha d'indicar la classe que tindrà aquesta propietat. Per exemple, en el següent tros de codi, fem que totes les instàncies de tipus `bpy.types.Material` tinguin una propietat anomenada `radiance`, de tipus `Pointer` que apunta a la classe `MaterialSettingsRadiance`.

```
bpy.types.Material.radiance = bpy.props.PointerProperty(type= \
    MaterialSettingsRadiance)
```

També podem declarar diferents propietats a l'atribut `radiance` utilitzant `PropertyGroup`. Aquest paràmetre ens permet recopilar les propietats dins d'una classe. Un exemple pot ser `MaterialSettingsRadiance`, on es troben totes les propietats d'interfície dels materials, com ara:

```
class MaterialSettingsRadiance(bpy.types.PropertyGroup):
    spec: bpy.props.FloatProperty(
        name='Specularity',
        soft_min=0, soft_max=1, default = 0,
        description = 'Material specularity'
    )
    ...

    color: bpy.props.FloatVectorProperty(
        name='Color',
        soft_min= 0, soft_max=1, size=4, default = (0,0,0,1),
        subtype='COLOR', update=my_material_color # sync with Blender color
    )
```

Aquestes propietats, en estar declarades dins d'una classe, ja estan implícitament indicades a la classe que pertanyen.

Els diferents constructors que es troben en el mòdul `bpy.props`: `bpy.props.FloatProperty`, `bpy.props.FloatVectorProperty`,... permeten definir, entre d'altres paràmetres, un valor per defecte, un rang de valors permès o funcions callback.

Un cop s'executin aquests codis, tots els materials de la classe `Material` contindran el nou atribut anomenat `radiance`. Aquest codi contindrà alhora els diferents atributs de `MaterialSettingsRadiance`. A tall d'exemple, si executem aquesta ordre:

```
bpy.data.materials['grass'].radiance.color = (1,1,1,1)
```

S'assignarà el color blanc al material `grass`.

10. Addon

D'aquesta manera, es van declarar totes les propietats dels materials, de l'escena i de les llums. Les propietats relatives de l'escena es van afegir al tipus `bpy.types.Scene` i les de les llums a `bpy.types.Light`.

A més a més, es va crear una nova propietat en la classe `Material`, anomenada `modifier`, que en comptes d'actuar com a un agrupador de propietats, `PointerProperty`, actua com a un contenidor de propietats, `CollectionProperty`. És a dir, permet agregar a un mateix material nous ítems amb les propietats especificades en la classe que apunta `CollectionProperty`.

Per exemple, si necessitem un nou modificador en el material `antimatter`, podem executar el següent codi en la consola de Python o en un script:

```
mod = bpy.data.materials['wood'].modifier.add()
mod.material = 'plastic'
```

Les classes que agrupen les propietats de les interfícies es troben implementades en el mòdul `settings.py`, i les propietats *apuntadores* es troben en el mòdul `__init__.py`. Això és degut a què per a fer instal·lable l'eina, aquestes propietats s'han de registrar igual que els noms de les classes a les quals apunten. Quedant el codi `__init__.py` de la següent manera:

```
classes = [SceneSettingsRadiance, MaterialSettingsRadiance, \
           ModifiersSettingsRadiance, LightSettingsRadiance]

def register():
    for cls in classes:
        bpy.utils.register_class(cls)
    bpy.types.Scene.radiance = PointerProperty(type=SceneSettingsRadiance)
    bpy.types.Material.radiance = PointerProperty(type=MaterialSettingsRadiance)
    bpy.types.Material.modifier = CollectionProperty(type=ModifiersSettingsRadiance)
    bpy.types.Light.radiance = PointerProperty(type=LightSettingsRadiance)
```

Per a poder desfer els canvis introduïts pel codi anterior es va implementar la funció `unregister()`. Aquesta funció permet eliminar totes les propietats declarades i serà invocada per a desinstal·lar l'eina:



```
def unregister():
    for cls in reversed(classes):
        bpy.utils.unregister_class(cls)
    del bpy.types.Scene.radiance
    del bpy.types.Material.radiance
    del bpy.types.Material.modifier
    del bpy.types.Light.radiance
```

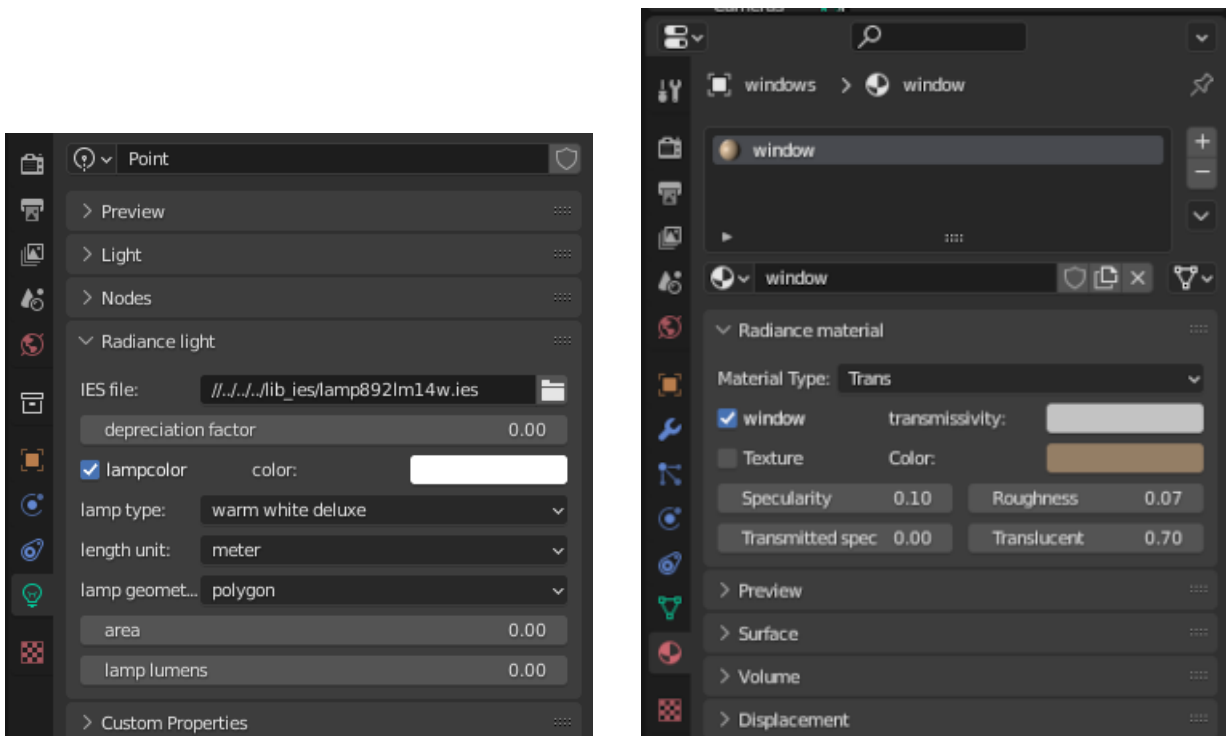
10.2.2 PROGRAMACIÓ D'INTERFÍCIES

La interacció de l'eina es realitzada a través de diferents panells que existeixen en Blender (`view3D`, `object`, `material`,...).

10. Addon

Aquestes classes s'hereten de `bpy.types.Operators`. Utilitzant els atributs `bl_region_type` i `bl_space_type` podem especificar la regió i el tipus de panell que es vol ampliar. Si el tipus de panell és general, és a dir, té diferents contextos, com és el cas del panell `PROPERTIES`, cal especificar-ho amb l'atribut `bl_context`.

`RAD_PT_Light` (panell de la il·luminació Radiance) i `RAD_PT_Material` (panell dels materials Radiance) es troben en el panell `PROPERTIES`. El primer ho està en el context `material`, representat amb la icona , i el segon, en canvi, en el context `data`, representat amb la icona , que és on es troben totes les propietats d'il·luminació. En la Figura 36 es mostra l'aspecte d'aquests dos panells de Blender en el context corresponent.



(a) Interfície de la il·luminació.

(b) Interfície dels materials.

Figura 36: Interfície de la il·luminació i els materials. Font: elaboració pròpia

Blender també permet crear subpanells dintre d'un panell. Això es pot especificar indicant l'*id* del panell *pare* amb l'atribut `bl_parent_id`.

El següent codi és un exemple que mostra aquest concepte. Va ser utilitzat per definir dos subpanells: `RAD_PT_Sky` (panell per a generar el gensky de Radiance) i `RAD_PT_Render` (panell per a generar la renderització); al panell d'escena, el qual va ser anomenat `RAD_PT_Scene`.

```
class RAD_PT_Scene(bpy.types.Panel):  
    """ Creates a panel to generate a  
        Radiance scene in the 3D viewport  
    """  
    bl_label = "Radiance Panel"  
    bl_idname = "RAD_PT_Panel"  
    bl_space_type = 'VIEW_3D'
```

```

bl_region_type = 'UI'
bl_category = "Radiance"

def draw(self, context):
    ...

class RAD_PT_Sky(bpy.types.Panel):
    """ Creates a subpanel to generate the
        Radiance gensky in the RAD_PT_PANEL
    """
    bl_label = "Sky"
    bl_idname = "SKY_PT_Panel"
    bl_space_type = 'VIEW_3D'
    bl_region_type = 'UI'
    bl_category = "Radiance"
    bl_parent_id = "RAD_PT_Panel"
    bl_options = {'DEFAULT_CLOSED'}

    def draw(self, context):
        ...

class RAD_PT_Render(bpy.types.Panel):
    """ Creates a subpanel to render the
        Radiance scene in the RAD_PT_PANEL
    """
    bl_label = "Render"
    bl_idname = "RENDER_PT_Panel"
    bl_space_type = 'VIEW_3D'
    bl_region_type = 'UI'
    bl_category = "Radiance"
    bl_parent_id = "RAD_PT_Panel"
    bl_options = {'DEFAULT_CLOSED'}

    def draw(self, context):
        ...

```

A la classe `RAD_PT_Scene`, els atributs de localització del panell (`bl_space_type` i `bl_region_type`) indiquen que el panell es troba en la vista 3D, dins el panell dret (UI) i en una pestanya anomenada Radiance (`bl_category = Radiance`).

L'aspecte del panell es configura en el mètode `draw()`, on els elements s'organitzen en files (`row`) on es poden afegir totes les propietats (`prop`) que han estat declarades en l'apartat anterior (Apartat 10.2.1). Amb aquest mètode també es poden agregar operadors (`operator`) que, tot i que es troben implementats en l'apartat següent (Apartat 10.2.3), es mostra com són declarats en la interfície, juntament amb les propietats, que es troben en el panell `RAD_PT_Render`.

```

def draw(self, context):
    radiance = context.scene.radiance
    layout = self.layout
    ...
    row = layout.row()
    row.prop(radiance, "variability")
    row.prop(radiance, "indirect")

    row = layout.row()
    row.operator('radiance.export', text="Export")
    row.operator('radiance.preview', text="Preview")

```

Totes les classes mencionades en aquest apartat, es troben sota el mòdul `ui.py`, on es troba implementada tota la interfície. No obstant, perquè el panell aparegui, es va registrar aquestes classes, en el mòdul `__init__.py`. És a dir, es van adjuntar els noms de les classes al vector `classes` d'aquest mòdul.

10. Addon

10.2.3 CREACIÓ D'OPERADORS

Els operadors són els que duen a terme les accions. Aquestes classes s'hereten de la classe `bpy.types.Operator` e invoquen el codi des del mètode `execute`. Aquest mètode contindrà les conversions de l'script explicades en l'Apartat 9, i operadors definits per a la usabilitat de la interfície. Aquests operadors seran executats quan es premi el botó de la interfície.

Per a la nostra eina crearem sis operadors: un de principal, `RAD_OT_Export`, i cinc de secundaris, definits a continuació:

- `MOD_OT_Remove`, `MOD_OT_Add` i `MOD_OT_Clear` són utilitzats en els materials que contenen materials modificadors, en la sintaxi Radiance.
- `LIGHT_OT_Color` assigna un color al fitxer IES.
- `RAD_OT_Preview` i `RAD_OT_Render` s'encarreguen de previsualitzar i renderitzar l'escena, respectivament.

Per a introduir el concepte d'Operator explicaré, a mode d'exemple, un dels bàsics: `MOD_OT_Remove`. Aquest operador representa els botons que es mostren amb un guionet de la Figura 37. A continuació, em centraré en el més important, el qual es troba en la interfície de l'escena.

L'execució de l'operador, en prémer el botó [-], donarà com a resultat l'eliminació d'un determinat element de la col·lecció de Modificadors dels materials. Per aquest motiu l'operador implementa un atribut, perquè necessita l'índex per tal d'eliminar l'element especificat. Per a realitzar aquesta acció utilitza l'atribut general `remove(index)` de les col·leccions:

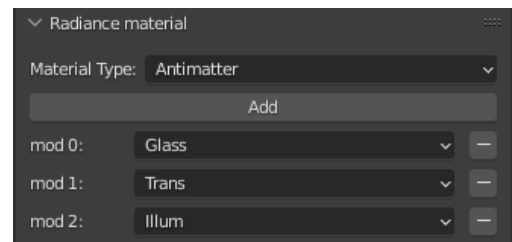


Figura 37: Interfície del material antimatter.
Font: elaboració pròpia

```
class MOD_OT_Remove(bpy.types.Operator):
    '''Remove a material type in active_material.modifiers'''
    bl_idname = "modifier.remove"
    bl_label = "Remove"

    index: bpy.props.IntProperty()

    def execute(self, context):
        bpy.context.object.active_material.modifier.remove(self.index)
        return {'FINISHED'}
```

L'operació `RAD_OT_Export`, serà l'encarregada d'exportar tota l'escena de Blender a Radiance. Caldrà que traslladi totes les dades de Blender, juntament amb les obtingudes a través de la interfície, a codi Radiance. Aquest codi va ser descrit en l'Apartat 8 (Renderitzant amb Radiance).

Aquesta operació d'exportació és realitzada quan s'executa el botó *Export* que es troba en la Figura 38. Per a poder dur a terme aquesta acció en Blender, implementada a través de l'script, es va organitzar el

10. Addon

codi en diferents funcions, reutilitzant el concepte de la classe Scene de *script.py*, definida en l'Apartat 9. Es van agregar nous materials a aquesta classe a fi de poder generar definicions més completes.

A més a més, també es van modificar els seus paràmetres. Es va realitzar una generalització per tal de què tots els mètodes d'aquesta classe utilitzin els mateixos paràmetres: els inputs de la interfície de materials de Radiance, `material.radiance` i `material.modifier`.

Aquesta modificació es va realitzar tenint en compte la seva implementació, que utilitza l'estructura de dades Hash map. Aquests canvis milloren l'eficiència, la llegibilitat i la senzillesa del codi. En la variable `addMaterials` es troben declarades les claus que corresponen als materials que, ara per ara, es poden convertir a Radiance i els valors que corresponen als mètodes de la classe.

A través del tipus de material obtingut a la interfície, s'executa un material o un altre, simulant la funcionalitat d'un switch, però executant mètodes d'una classe externa.

```
def def_material(context):
    radiance = m.material.radiance
    modifier = m.material.modifier
    addMaterials = {
        'light': s.addMaterialLight,
        'illum': s.addMaterialIllum,
        'glow': s.addMaterialGlow,
        'spotlight': s.addMaterialSpotlight,
        'mirror': s.addMaterialMirror,
        'prism1': s.addMaterialPrism1,
        'prism2': s.addMaterialPrism2,
        'mist': s.addMaterialMist,
        'plastic':s.addMaterialPlastic,
        'metal': s.addMaterialMetal,
        'trans': s.addMaterialTrans,
        'dielectric': s.addMaterialDielectric,
        'interface': s.addMaterialInterface,
        'glass': s.addMaterialGlass,
        'antimatter': s.addMaterialAntimatter
    }
    addMaterials[radiance.material](m.name,radiance,modifier)
```

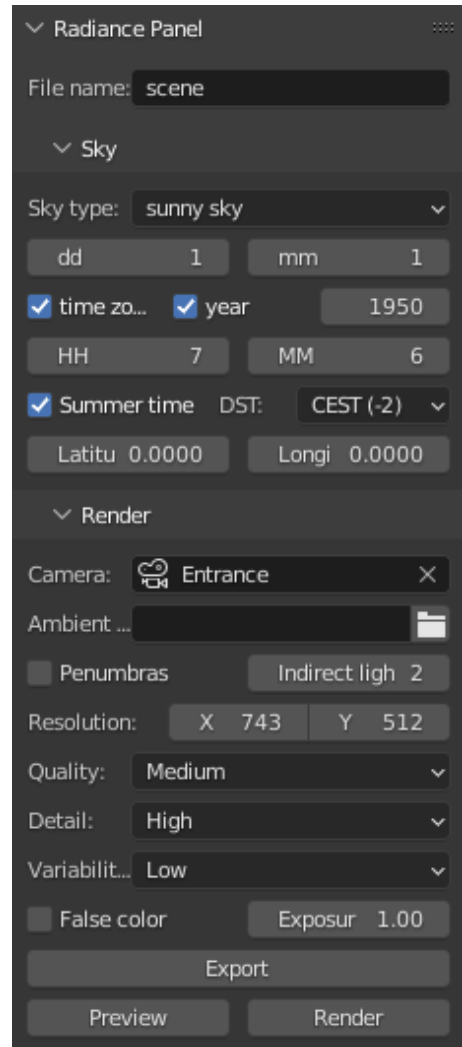


Figura 38: Interfície de l'escena. Font: elaboració pròpia

En utilitzar un map que fa ús d'un hash, el cost de consulta per a un valor (del qual es sap la clau) és $O(1)$.

El cas de l'automatització de la transformació de les textures de Blender a hdr va ser aconseguida, a través de les diferents operacions que proporcionen els nodes de Blender. Més específicament, es

va realitzar amb les operacions de l'apartat de shading, que es mostren en la Figura 39, la qual és important visualitzar-la per tal d'entendre el codi.

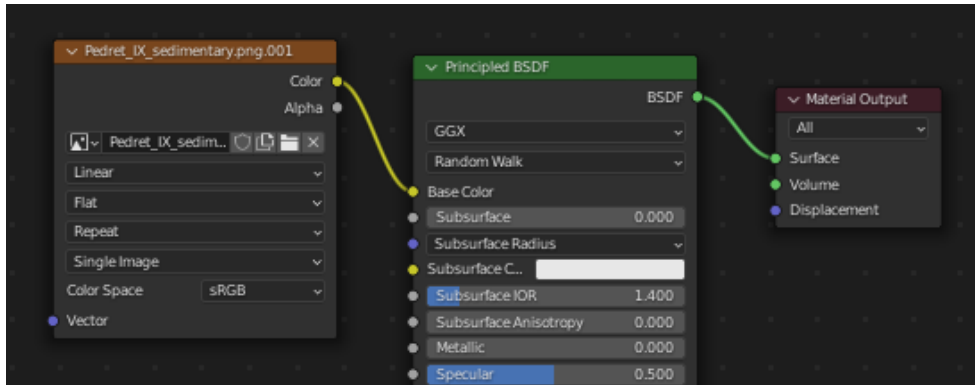


Figura 39: Nodes del material de construcció inorgànic de Blender. Font: elaboració pròpia

El nostre objectiu, és extreure la informació del node esquerra de la Figura 39, que és del tipus `TEX_IMAGE`, i es troba linkat amb el `BaseColor` del node Principled BSDF. El node de la dreta, `Material Output`, és el node general que es troba en tots el materials. El codi que ho aconsegueix es reproduïx a continuació:

```
def text2hdr(context):
    ...
    for m in ob.material_slots:
        if m.material:
            surface = m.node_tree.nodes.get('Material Output').inputs['Surface'].\
                links[0].from_node
            if surface.inputs['Base Color'].is_linked:
                texture = surface.inputs['Base Color'].links[0].from_node

                if texture.type == 'TEX_IMAGE'
                    texture.image.save_render(m.name)
                    command = f'mogrify -format hdr {m.name}'
                    os.system(command)
                    ob_mat.addMaterialColorTexture(m.name, f'{m.name}.hdr')
            else:
                raise Exception('Base Color is not directly linked to\
                    Texture Image Node')
        else:
            print('There is no texture image, so has been obtained by \
                the Blender Base Color')
            # Blender base color and Radiance color are sync
            # by callback properties
            def_material(context)
```

Per a cada material, de tots els materials de l'objecte, s'obté el node `Material Output`, amb `m.node_tree.nodes.get('Material Output')`. Per passar al node següent, `Principled BSDF`, es

10. Addon

demana el node al qual pertany el primer link de l'input Surface i es guarda aquest node en la variable `surface`.

Amb la variable `surface`, comprovem que el node es trobi connectat a un node de tipus `TEX_IMAGE`. Si és cert, s'emmagatzemarà la textura i serà convertida a `hdr`. En cas contrari, es llençarà una excepció a l'usuari, per a què connecti directament el node de textura amb el `BaseColor`. No obstant, si és a causa de què no existeix cap link en `BaseColor`, en aquest cas es crida la funció `def_material`, definida en el codi anterior, reproduint el mateix color de l'objecte que es troba a Blender.

La conversió d'objecte de Blender a Radiance, `obj2rad`, és gairebé idèntica a la de l' `script.py`, `blend2mesh`, però ara s'ha agregat la definició de l'objecte de Radiance a aquesta funció, a fi d'obtenir el codi més estructurat:

```
def obj2rad(context,ob,materials,mod):
    """Specifies the materials (mod) of the mesh"""
    ...
    with open(f'{ob.name}.rad', 'a') as f:
        f.write(f'{mod} mesh {ob.name}\n')
        f.write(f'1 {ob.name}.rtm\n0\n')
```

La funció per a exportar el cel també és molt similar a la versió del `script.py`, amb l'única diferència de què aquí, en tenir una interfície, és va crear l'opció de permetre a l'usuari decidir com definir l'hora: si a través de la zona horària o a través del meridià.

```
def generate_sky(context):
    s = bpy.context.scene.radiance
    with open('sky.rad', 'w') as f:
        f.write(f'!gensky {s.sky_month} {s.sky_day} ')
        if s.is_sky_time_zone:
            TZone = s.sky_DST if s.is_sky_DST else s.sky_LST
            f.write(f'{s.sky_time_h}:{s.sky_time_min}{TZone} ')
        else:
            f.write(f'{s.sky_meridian}')
    ...
```

Totes les classes mencionades en aquest apartat es troben sota el mòdul `operators.py`, on es troba tota la implementació dels operadors. De la mateixa manera que es van registrar els panells, també cal registrar els operadors, per a què es puguin utilitzar en el mòdul `__init__.py`, agregant els noms de les classes al vector `classes`, d'aquest mòdul.

10.2.4 CREACIÓ DE L'ADDON

Per a crear l'addon, com que ja s'havia dissenyat la seva estructura com a un package de Python, només va ser necessari comprimir tot el codi en un fitxer zip, que va ser anomenat `radiance`, designant el nom del l'addon.

10. Addon

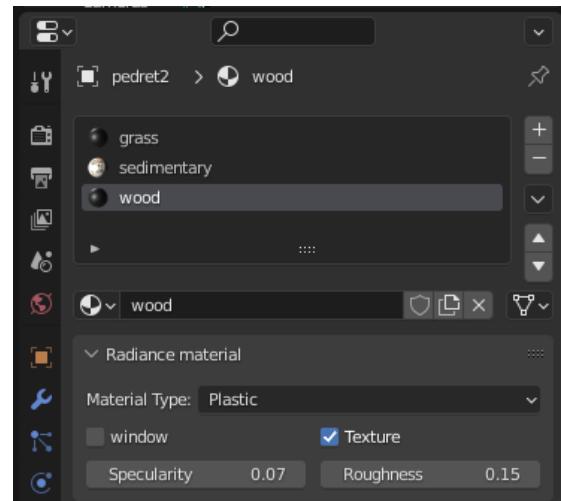
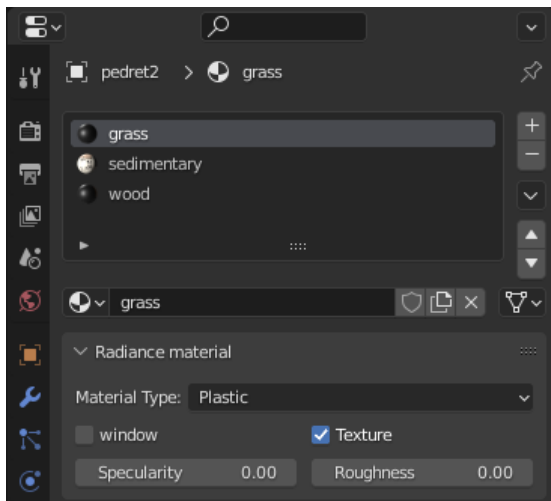
Per a crear un paquet instal·lable, es va procedir a canviar les línies d'import, de manera que es faci referència als fitxers python, a través d'una ruta relativa al directori del paquet, de la següent manera:

```
if __name__ == "radiance":
    from ./ui import *
    from ./settings import *
    from ./operators import *
else:
    from ui import *
    from settings import *
    from operators import *
```

Si el fitxer és executat com a un addon, la variable `__name__` prendria com a valor el nom del mòdul (el del directori on s'hagi emmagatzemat el codi). Per tant, comprovant el valor d'aquesta variable podem escollir la forma de carregar els mòduls associats a l'addon.

10.3 VALIDACIÓ

Per a validar el correcte funcionament de l'addon, s'ha realitzat una simulació d'un cas especial de cuir, que en ser més fi és més translúcid i deixa passar part de la llum. A continuació es va procedir a indicar les propietats dels materials en les interfícies de la Figura 40.



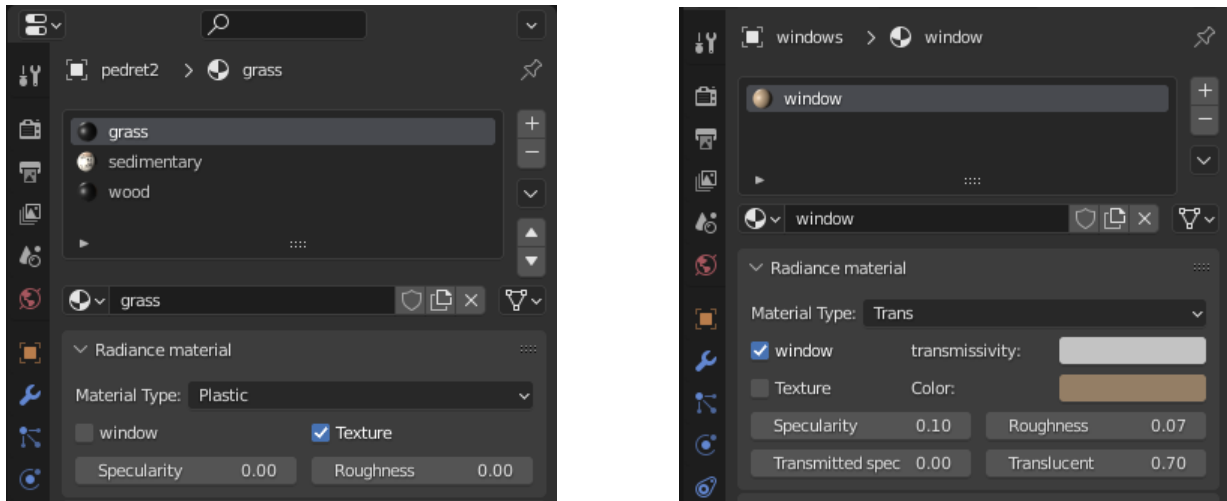


Figura 40: Configuració dels diferents materials. Font: elaboració pròpia

Aquestes propietats van ser les mateixes utilitzades quan es van definir els materials (Apartat 8.2.5), a excepció de la finestra, en la qual s'ha canviat la propietat de transmissió a un 70%. Per al valor de transmissivitat es va fer l'aproximació amb la Fórmula 11 abans mencionada, ja que no s'està simulant cap mostra exacta.

Un cop definits els materials, es va procedir a renderitzar l'escena, la qual va ser simulada també l'1 d'agost del 1950 a les 11:00. En les coordenades de Sant Quirze de Pedret, que utilitza les dades de la franja horària proporcionades per Central European Summer Time. En la Figura 41, es visualitza com s'introduirien aquests paràmetres.

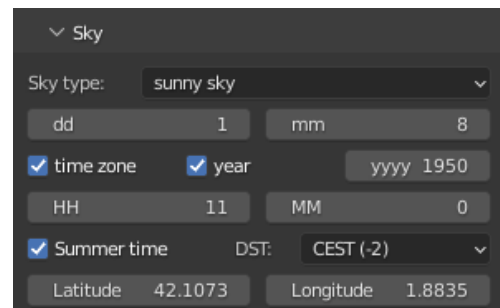


Figura 41: Paràmetres del cel. Font: elaboració pròpia

En el panell de la Figura 42 es va seleccionar de totes les càmeres que es troben en l'escena Blender la càmera Entrance, que és la que visualitza l'absis, també es va optar per crear un fitxer ambient (scenet3.amb), ja que partíem de zero i encara no hi havia la possibilitat de seleccionar un fitxer ambient que es trobés en el gestor d'arxius (arxius ambient creats prèviament).

La selecció d'un fitxer ambient és útil en el cas que es renderitzi la mateixa escena amb una càmera diferent, ja que emmagatzema totes les dades d'il·luminació indirecta realitzades durant el procés de renderització, i estalvia aquest futur càlcul en les pròximes renderitzacions, escurçant el temps de comput. No obstant si es canvia la geometria o les fonts de llum de l'escena, s'ha de crear un nou fitxer ambient.

La resolució de la previsualització es va especificar de 512px x 512px, amb una qualitat, variabilitat i detall mig; ja que era suficient per a fer una previsualització, i perquè fos ràpida la renderització de l'escena es va establir una llum indirecta amb 4 ambient bounces, que és la meitat de llum indirecta màxima que permet Radiance.

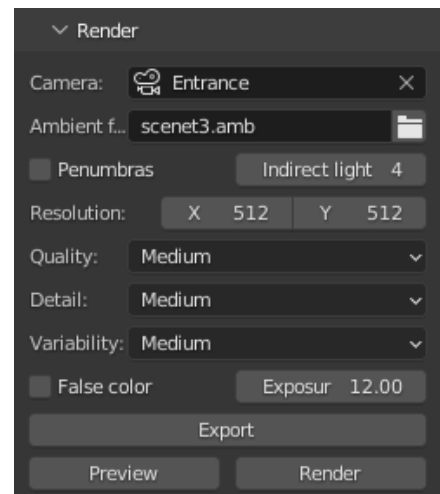


Figura 42: Paràmetres de renderització. Font: elaboració pròpia

10. Addon

A continuació es va prémer el botó export, per tal d'exportar tota l'escena Blender a Radiance, per a poder-la previsualitzar. Un cop s'indica que l'escena ha estat exportada, clicant el botó de Preview, es pot visualitzar la Figura 43

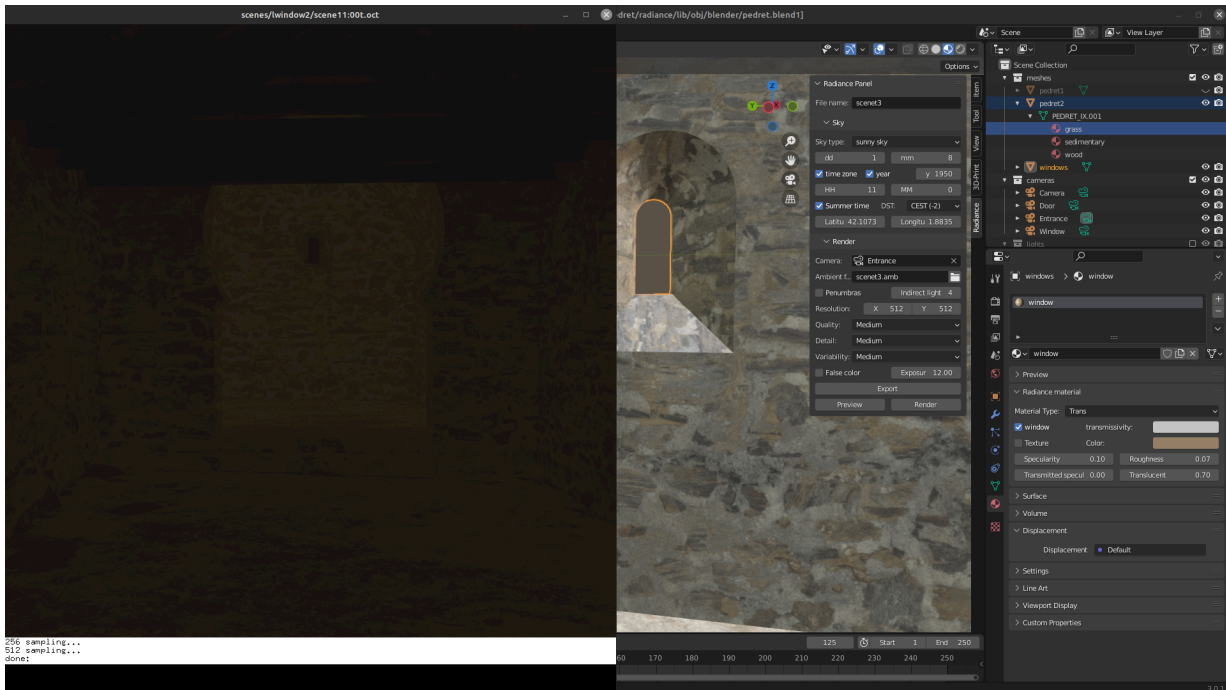


Figura 43: Previsualització de l'escena amb una exposició de 12. Font: elaboració pròpia

Com es veu en la darrera imatge, les primeres visualitzacions solen ser fosques, ja que l'exposició no es sol encertar a la primera, aquesta pot ser ajustada interactivament a la finestra emergent, per tal que quedi més il·luminada l'escena, amb la comanda *exposure*, o també tancant la finestra emergent i clicant el botó de Preview amb un nou exposure.

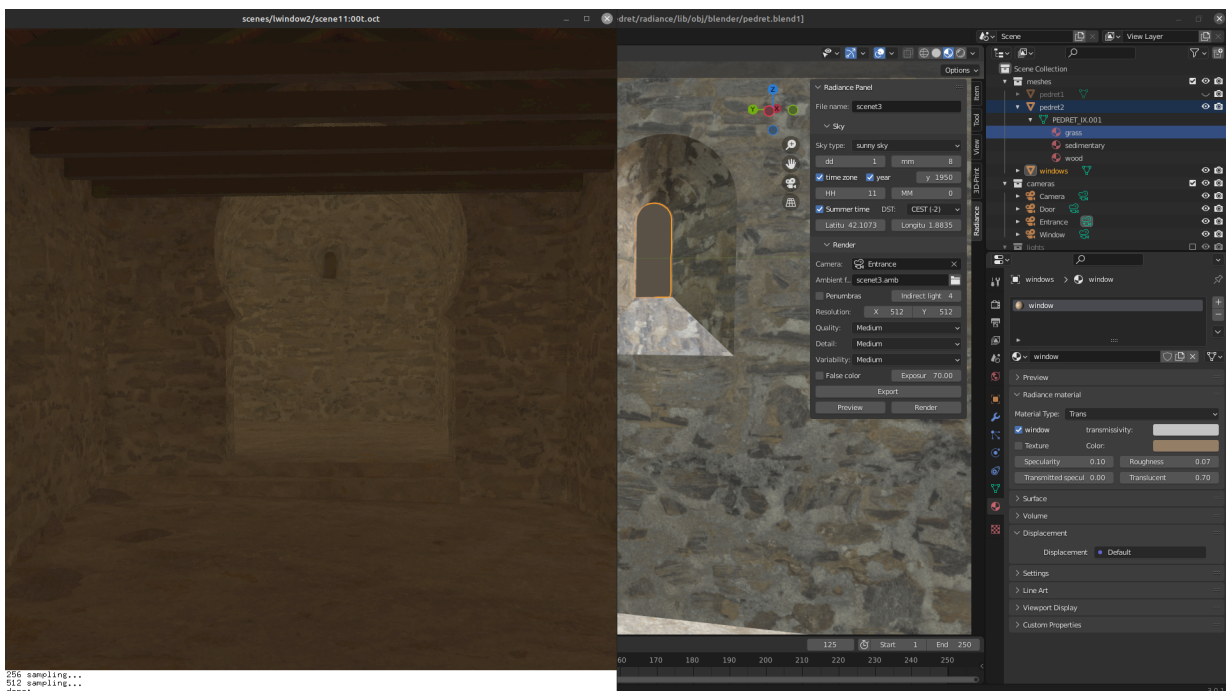


Figura 44: Previsualització de l'escena amb una exposició de 70. Font: elaboració pròpia

11 ADVERSITATS

La **primera dificultat** que em va sorgir és que no sabia per on començar, com organitzar-me (quina informació necessitava, ja que el vocabulari era molt tècnic i desconeixia molts conceptes relacionats amb la il·luminació, els materials, la fotometria ...). Vaig haver de dedicar molt de temps per a refrescar conceptes apresos durant la meva escolarització i adquirir nous aprenentatges en diferents camps de la física (molt específics).

El **segon problema** de consideració va estar obtenir una imatge en color.

scene.rad

```
void mesh church
3 church.rtm -rx 90
0
0
```

materials.rad

```
void plastic Pedret_IX-baked
0
0
5 1 1 1 0 0
```

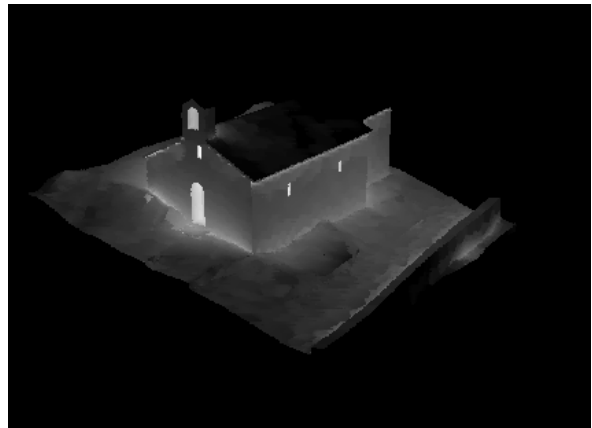


Figura 45: Primera renderització de Sant Quirze de Pedret

En un començament no sabia com aconseguir el color a partir de la textura monomaterial. De manera que el primer rendering que es va obtenir és el mostrat en la figura 45. Cap convertidor que vaig trobar en Internet d'imatges png (Textura Blender) a hdr (Textura Radiance) transformava la imatge correctament, de manera que obtenia una imatge uniformement negra. Jo crec que podria ser a causa de les compressions internes que es feien durant la conversió, perquè molts convertors que es troben en Internet no suportaven el pes de la imatge (256 MB). Això es va poder solucionar en trobar la comanda `mogrify` de Linux.

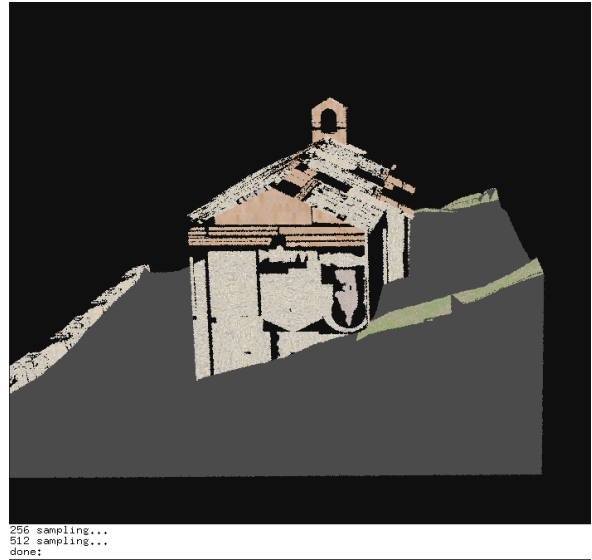
La **tercera adversitat** estava relacionada amb la il·luminació indirecta com mostra la Figura 46a. No entenia què fallava, cosa que va comportar que el grup de recerca ViRVIG comprés el llibre[14], per tal de trobar la solució. Al fòrum de preguntes de Radiance hi havia escassa informació al respecte i, quan existia, les explicacions eren complexes i confuses, molt tècniques i difícils de seguir.

Els entrebancs es van anar superant a mesura que anava entenent i incorporant els nous conceptes de renderització, degut a la progressiva comprensió de cada paràmetre.

El **quart problema** va estar el procés del baking. Vaig topar-me amb alguns obstacles a Radiance, com mostra la Figura 46b. El principal problema residia en l'exportació de l'objecte. Com es pot veure en la imatge, durant l'exportació es perdia la parametrització del model i les textures es visualitzaven incorrectament a Radiance. Aquest problema va ser solucionat exportant únicament l'UVMMap que ens interessava i descartant la resta.



(a) Problema de la renderització indirecta.



(b) Problema baking.

Figura 46: Tercer i quart problemes del projecte. Font: elaboració pròpia

D'altra banda també es van trobar **algunes limitacions**. Una d'elles és que els materials que s'han descrit en l'estudi s'han formalitzat mitjançant un procediment heurístic, ja que no es disposava de definicions precises dels materials originals que formaven part de Pedret en els diferents períodes històrics de l'església. Per solucionar-ho, s'hauria de treballar de forma més cooperativa amb historiadors de l'art perquè em poguessin facilitar aquestes definicions, per a poder aconseguir simulacions que tinguessin un valor científic més rigorós, en recrear amb imatges les diferents èpoques de la construcció. En tot cas, aquest problema es troba fora de l'abast dels nostres objectius i possibilitats, Treball de Fi de Grau.

A més a més, també es va intentar visitar l'església de Sant Quirze de Pedret, durant el mes de juny, però per problemes entre la Diputació de Barcelona i l'Ajuntament de Berga estava tancada al públic. No obstant, els meus directors em van proporcionar fotografies que van ser utilitzades com a material base.

12 CONCLUSIONS

En aquest Treball de Fi de Grau s'han presentat tots aquells coneixements necessaris per entendre, desenvolupar i implementar un addon que permet utilitzar el programari Radiance.

Per consolidar aquests coneixements, primerament es van fer diverses renderitzacions manuals a Radiance, però es va concretar amb el model de Sant Quirze de Pedret, el model que utilitzaríem com a base per a crear la nostra eina. Durant aquest procés es va il·luminar, previsualitzar, renderitzar i analitzar l'entorn de Sant Quirze de Pedret, aconseguint imatges fotorrealistes molt similars a la realitat.

Havent vist com es renderitzaven les escenes Radiance, es va tenir una idea de la gran quantitat de simulacions que es poden realitzar, a través de les diferents configuracions de paràmetres que hi ha en Radiance i la interactivitat que permet Blender per a configurar una escena: agregar objectes i modelar-los (espelmes), definir un objecte amb múltiples materials (join), col·locació interactiva dels objectes...

A continuació, es va procedir a implementar les conversions, per a transformar les dades Blender a Radiance, de manera que el codi fos eficient i no s'afegissin sobrecostos a l'eina, aportant sempre solucions senzilles, perquè el codi resultés fàcil de llegir i d'entendre.

Finalment, es va dissenyar i implementar la interfície gràfica de l'eina en Blender, juntament amb alguns retocs de conversions, per tal que tot el codi fos íntegre i aprofités la flexibilitat ofertada per la interfície. El resultat va ser una extensió de Blender modular, la qual permet a tot usuari de Blender poder fer ús del programari de renderització físicament validat (Radiance), sense haver d'invertir un temps elevat en la seva corba d'aprenentatge.

12.1 COMPETÈNCIES TÈCNIQUES

Aquest Treball de Fi de Grau ha estat dut a terme seguint les següents competències tècniques, les quals aquí s'expliquen i es justifiquen:

CCO2.2: Capacitat per a adquirir, obtenir, formalitzar i representar el coneixement humà d'una forma computable, per a la resolució de problemes.

L'eina desenvolupada i implementada és bàsicament una extensió d'una aplicació existent que crea renderitzacions físicament acurades. Permet que l'usuari no hagi de saber utilitzar Radiance, que és el programari complicat que crea les renderitzacions. A més a més, a partir d'aquesta eina l'usuari podrà representar l'escena de Blender, amb un requisit senzill: especificar els materials en unitats físiques.

CCO2.3: Desenvolupar i avaluar sistemes interactius i de presentació d'informació complexa, i la seva aplicació a la resolució de problemes de disseny d'interacció persona computador. [Bastant]

L'add-on implementat permet visualitzar de manera senzilla simulacions complexes d'interacció entre diferents fonts de llum i els materials d'una escena. L'eina ha estat dissenyada tenint en compte la seva usabilitat i, a més a més, permet la visualització del resultat de la simulació de dues maneres:

12. Conclusions

mostrant la imatge visual generada o mostrant els resultats numèrics de la visualització. D'aquesta manera es permet a l'usuari entendre millor la il·luminació de l'escena per a poder validar hipòtesis o prendre decisions relacionades amb la conservació.

CCO2.6: Dissenyar i implementar aplicacions gràfiques, de realitat virtual, de realitat augmentada i videojocs. [En profunditat]

En aquest TFG s'ha dissenyat i implementat un visualitzador de gràfics per a generar simulacions físiques de Sant Quirze de Pedret, de manera que permeti apreciar els comportaments lumínics, amb diferents propietats en els materials simulats, i introduir certa interacció a dins del visualitzador (l'usuari pot interactuar amb l'eina i visualitzar els resultats).

Aquest visualitzador implementa diverses tècniques vistes a les assignatures d'Interacció i Disseny d'Interfícies (IDI) i Gràfics (G), per a aconseguir crear la renderització físicament acurada.

12.2 TREBALL FUTUR

Tot i que es considera que la part principal del treball està acabada, hi ha alguns detalls de l'add-on que no s'han pogut arribar a implementar o estudiar, que ha comportat dur a terme una prioritització de tasques:

- Aprofundir en l'estudi de fitxers IES i en la cerca d'alternatives per tal de simular amb més profunditat fonts de llum com làmpades d'oli i espelmes.
- Simular la posició del Sol a partir de diferents franges horàries, en el programa 3D Blender prement un botó, perquè l'usuari pugui entendre fàcilment la il·luminació natural de la renderització en Radiance.
- Afegir una opció que permeti generar imatges en múltiples direccions, per a poder combinar-les i així, sintetitzar imatges 360^º que es puguin visualitzar en dispositius mòbils i de realitat virtual.
- Configurar la compatibilitat de l'eina en l'entorn Windows, sense cap dependència amb l'entorn Linux.

12.3 VALORACIÓ PERSONAL

Ara que el treball s'ha acabat i miro enrere en el temps, m'adono de tot l'esforç que ha suposat. A més a més, ha estat un procés d'aprenentatge interessant, ja que he après nombroses coses. Per exemple, a nivell d'il·luminació natural, m'han sorprès molts conceptes meteorològics que no hagués pensat mai que s'utilitzessin per a generar renderitzacions físiques com el zenit, i l'existència dels diferents simuladors de cels, que s'han trobat durant el procés de recerca (CIE, Perez, Matsuura,...).

En aquest punt em vaig adonar de la gran importància que té la il·luminació en l'escena i la precisió que em requeriria per a definir-la. També vaig entendre que no sempre és millor començar per les situacions més complexes, com les últimes hores del vespre, que dificulten trobar solucions degut a la manca de llum.

12. Conclusions

La part d'analitzar l'escena, va ser molt interessant i alhora estressant, ja que les renderitzacions no surten bé a la primera, i cal tenir paciència. Així i tot, després del gran nombre de renderitzacions que es van fer, finalment es van aconseguir resultats adequats i explicables. Penso que aquest treball a part d'aportar-me coneixements intel·lectuals, també m'ha servit per a interessar-me en el món de la recerca, i en un futur no descarto treballar en investigació.

A l'hora d'implementar l'addon, em vaig adonar de la difícil tasca que tenia entre mans: implementar quelcom d'ús general, però que pugui crear renderitzacions de manera acurada. Ha estat bastant motivador haver-ho pogut aconseguir.

D'altra banda, m'agradaria agrair, un cop més els meus tutors del TFG, el Carlos i a l'Imanol, per haver estat darrere donant-me suport i feedback, des dels inicis del projecte fins ara, i haver confiat en mi per concedir-me l'oportunitat de realitzar aquest treball.

REFERÈNCIES BIBLIOGRÀFIQUES

- [1] Turismo prerrománico. San Quirico de Pedret. <https://www.turismo-prerromanico.com/wp-content/uploads/2016/06/MSQUIRCEG2.jpg>. Consulta [en línia]: 27/02/2022.
- [2] EHEM. Enhancement of Heritage Experiences. <https://ehemproject.eu/page/what-we-do/>. Consulta [en línia]: 17/09/2022.
- [3] El Grupo Informatico. Ray Tracing. <https://www.elgrupoinformatico.com/que-ray-tracing-t73504.html>. Consulta [en línia]: 03/04/2022.
- [4] Encrypted-tbn0. Rendering fotorrealista. <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQn0tCnqQiICYx-fIXWEJIVBRISFRi2oZ7eXw&usqp=CAU>. Consulta [en línia]: 27/02/2022.
- [5] Henrik. Stanford graphics. https://graphics.stanford.edu/~henrik/images/metalring_small.jpg. Consulta [en línia]: 27/02/2022.
- [6] Picdn. Ring photos. <https://ak.picdn.net/shutterstock/videos/15604369/thumb/1.jpg>. Consulta [en línia]: 27/02/2022.
- [7] Blender. Radiosity. <https://download.blender.org/documentation/htmlI/PartR/radiosity/gfx/RR04.png>. Consulta [en línia]: 20/09/2022.
- [8] escuelaangg. Flujo luminoso, intensidad luminosa, iluminancia, luminancia. <https://pbs.twimg.com/media/CskPWTvVMAAW1uU.jpg>. Consulta [en línia]: 27/09/2022.
- [9] Tesla. Design your model Y. <https://www.tesla.com/modely/design#overview>. Consulta [en línia]: 20/03/2022.
- [10] Rendering Definition. Rendering is the process of generating a final digital. <https://en.vcenter.ir/technical/rendering-definition/>. Consulta [en línia]: 20/03/2022.
- [11] xpsnl on Envato Studi. Interior Design Concept & Photorealistic Renderings. <https://studio.envato.com/explore/3d-design-modeling/7778-interior-design-concept-photorealistic-renderings>. Consulta [en línia]: 20/03/2022.
- [12] G.A. Castrillón et al. A: Hallazgos imagenológicos de la displasia cleidocraneal 81 (maig de 2017). Consulta [en línia]: 20/03/2022. DOI: [10.1016/j.rard.2017.03.006](https://doi.org/10.1016/j.rard.2017.03.006).
- [13] Peter Byrne. ?Comparison Study of Four Popular Lighting Simulation Software Programs? A: MSc Dissertation 81 (febr. de 2015). Consulta [en línia]: 20/03/2022. DOI: https://issuu.com/peter.byrne1000/docs/dissertation_-_peter_byrne_-_publis.
- [14] Greg Ward Larson i Rob Shakespeare. Rendering with Radiance: The Art of Lighting Visualization. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. ISBN: 1558604995.
- [15] Dion Moul. Photorealistic rendering with radiance. <https://www.blendernation.com/2019/06/09/photorealistic-rendering-with-radiance/>. Consulta [en línia]: 27/02/2022.
- [16] Free i Open 3D Creation Software. Blender. <https://www.blender.org/>. Consulta [en línia]: 24/08/2022.
- [17] Wikipedia. Scrum (software development). [https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development)). Consulta [en línia]: 28/02/2022.

- [18] Facultat d'Informàtica de Barcelona. Treball de Fi de Grau. <https://www.fib.upc.edu/ca/estudis/graus/grau-en-enginyeria-informatica/treball-de-fi-de-grau>. Consulta [en línia]: 2/03/2022.
- [19] International Relations Bureau. Assessment system and sredits. <https://www.upc.edu/sri/en/students/students-mobility-office/incomings/studying-in-upc/assessment-system-and-credits/assessment-system-and-credits>. Consulta [en línia]: 2/03/2022.
- [20] Glassdoor. Sueldo: Jefe De Proyecto (Marzo, 2022). https://www.glassdoor.es/Sueldos/jefe-de-proyecto-sueldo-SRCH_K00,16.htm. Consulta [en línia]: 9/03/2022.
- [21] Glassdoor. Sueldo: Programador (Marzo, 2022). https://www.glassdoor.es/Sueldos/programador-sueldo-SRCH_K00,11.htm. Consulta [en línia]: 9/03/2022.
- [22] Glassdoor. Sueldo: Software Tester (Marzo, 2022). https://www.glassdoor.es/Sueldos/software-tester-sueldo-SRCH_K00,15.htm. Consulta [en línia]: 9/03/2022.
- [23] Glassdoor. Sueldo: Ui Ux Designer (Marzo, 2022). https://www.glassdoor.es/Sueldos/ui-ux-designer-sueldo-SRCH_K00,14.htm. Consulta [en línia]: 9/03/2022.
- [24] Aurea co-working. Precios Coworking Barcelona. <https://www.aureacoworking.com/ca/precios-coworking/>. Consulta [en línia]: 9/03/2022.
- [25] Ministerio de España. Calculo de emisiones - 2011. https://www.miteco.gob.es/es/ceneam/programas-de-educacion-ambiental/hogares-verdes/2011%20-%202016%20Calculo%20de%20emisiones%20HV2011_tcm30-171463.pdf. Consulta [en línia]: 9/10/2022.
- [26] Imanol Munoz-Pandiella et al. ¿Digital Reintegration of Distributed Mural Paintings at Different Architectural Phases: the Case of St. Quirze de Pedret? A: Eurographics Workshop on Graphics and Cultural Her
Ed. de Federico Ponchio i Ruggero Pintus. The Eurographics Association, 2022. ISBN: 978-3-03868-178-6. DOI: [10.2312/gch.20221227](https://doi.org/10.2312/gch.20221227).
- [27] Lawrence Berkeley National Laboratory. The RADIANCE 5.1 Synthetic Imaging System. <https://floyd.lbl.gov/radiance/refer/ray.html>. Consulta [en línia]: 03/04/2022.
- [28] Dion Moul. A proposed Radiance Filesystem Hierarchy Standard. <https://thinkmoul.com/proposed-radiance-filesystem-hierarchy-standard.html>. Consulta [en línia]: 2/09/2022.
- [29] ACM Siggraph 98 Course #33. Rendering with radiance notes - S98NOTES.PDF. <https://floyd.lbl.gov/radiance/refer/s98c33.pdf>. Consulta [en línia]: 3/09/2022.
- [30] Wikipedia. Reflexión difusa. https://es.wikipedia.org/wiki/Reflexi%C3%B3n_difusa. Consulta [en línia]: 27/09/2022.
- [31] JALOX. Radiance Colour Picker. https://www.jaloxa.eu/resources/radiance/colour_picker.shtml. Consulta [en línia]: 6/09/2022.
- [32] Ryan King Art. Backe Multiple Materials. <https://www.youtube.com/watch?v=wG60N8wZYLc>. Consulta [en línia]: 6/09/2022.
- [33] Andrew Marsh. CIE Sky Generator. <http://andrewmarsh.com/software/cie-sky-web/web-cie-std-gen-sky-simulation.jpg>. Consulta [en línia]: 12/09/2022.
- [34] Modeling an oil lamp source. Discourse Radiance. <https://discourse.radiance-online.org/t/modeling-an-oil-lamp-source/1507>. Consulta [en línia]: 27/09/2022.
- [35] Improve Depot. Candle to lumens. <https://improvedepot.com/pages/lumens-vs-candlepower-whats-the-difference>. Consulta [en línia]: 24/09/2022.

Referències bibliogràfiques

- [36] IKEA. ADLAD Vela perfumada grueta. <https://www.ikea.com/es/es/p/adlad-vela-perfumadagruesa-bosque-escandinavo-blanco-80502272/>. Consulta [en línia]: 4/10/2022.
- [37] GitHub. Releases - LBNL-ETA/Radiance. <https://github.com/LBNL-ETA/Radiance/releases>. Consulta [en línia]: 27/08/2022.

GLOSSARI

absis construcció adossada a la nau, normalment de forma semicircular, tot i que n'hi ha de quadrangulars (absis carrats) i poligonals. Acostuma a ser cobert amb una volta (sovint de quadrant d'esfera) i acostuma a trobar-se darrere de l'altar formant capelles.

baked texture És una aproximació dels efectes que tenen les superfícies complexes a mapes de bits 2D simples, que després es poden assignar al objecte.

CEST Central European Summer Time.

CIE Sky Model La Comissió Internacional d'Il·luminació ha publicat una sèrie de "tipus de cels" que proporcionen informació de distribució de la luminància que s'utilitza en els programaris de simulació d'il·luminació per a calcular els nivells de llum probables en models simulats durant aquestes condicions de "tipus cel".

CLI La interfície de línia d'ordres.

càustica Superfície tangent als raigs reflectits o refractats per un sistema.

depreciation factor La proporció d'il·luminació quan tot està net.

EHEM Enhancement of heritage experiences in the middle ages.

FIB Facultat d'Informàtica de Barcelona.

GEP Gestió de Projectes.

IES Format de fitxer per a la transferència electrònica de dades fotomètriques, creat per Illuminating Engineering Society of North America.

IESNA Illuminating Engineering Society of North America.

mockup És una mostra prèvia que realitza el dissenyador gràfic, mitjançant un fotomuntatge a escala per mostrar a un client com quedarà el disseny de la web, el servei o el producte que hagi contractat..

reflectància Propietat d'un cos de reflexar llum.

RMS És una escala per mesurar la rugositat, descrivint la variació de pendent a través d'una superfície..

superfície Lambertiana una superfície idealment difusa que dona lloc a que la llum es reflecteixi per igual en totes les direccions.

TFG Treball de Fi de Grau.

transmitància Quantitat d'energia que travessa un cos en una determinada quantitat de temps.

UPC Universitat Politècnica de Catalunya.

ViRVIG Visualització, Realitat Virtual i Interacció Gràfica.

zenit El zenit és el punt de la intersecció entre la vertical de l'observador i l'esfera celeste que queda per sobre de l'observador.

A ANNEXOS

A INSTAL·LACIÓ DE RADIANCE

Cada versió de Radaince s'ha anat recopilant al repositori de GitHub que es troba en [37], i totes elles han estat llançades per Linux, Windows i Mac.

Les definicions de les escenes, les renderitzacions i les execucions dels programes Radiance s'han generat utilitzant la versió més recent, Radiance 5.4a, en Linux. Per instal·lar-la es pot fer de la següent manera:

1. Descarreguem el fitxer Radiance_71eb6b30_Linux.zip que es troba [aquí](#) o bé executem la següent comanda en la consola:

```
:~$ wget \
https://github.com/LBNL-ETA/Radiance/releases/download/71eb6b30/Radiance_71eb6b30_Linux.zip
```

2. Descomprimim l'arxiu descarregat:

```
:~$ unzip Radiance_71eb6b30_Linux.zip
:~$ tar -xvzf radiance-5.4.71eb6b308d-Linux.tar.gz
```

3. Movem els arxius de la carpeta radiance-5.4.71eb6b308d-Linux/usr/local/radiance/lib a /usr/local/lib/radiance:

```
:~$ sudo mkdir /usr/local/lib/radiance
:~$ sudo mv radiance-5.4.71eb6b308d-Linux/usr/local/radiance/lib/* \
/usr/local/lib/radiance
```

4. Movem els arxius de la carpeta radiance-5.4.71eb6b308d-Linux/usr/local/radiance/bin a /usr/local/bin:

```
:~$ sudo mv radiance-5.4.71eb6b308d-Linux/usr/local/radiance/bin/* \
/usr/local/bin
```

5. Netegem els arxius innecessaris:

```
:~$ rm -rf radiance-5.4.71eb6b308d-Linux* Radiance_71eb6b30_Linux.zip
```

6. Agreguem la variable d'entorn RAYPATH i reiniciem l'ordinador:

(a) Obrim amb el nostre editor el fitxer *.profile* o *.bash_profile*

(b) Agreguem al final del fitxer:

```
export RAYPATH=$RAYPATH:/usr/local/lib:/usr/local/lib/radiance
```

(c) Guardem i reiniciem l'ordinador

Ara si tot ha anat bé, ja es pot executar i recrear totes les renderitzacions generades en aquesta memòria.

A. Annexos

B GENERAR LES RENDERITZACIONS OBTINGUDES AMB RADIANCE

Aquestes són les instruccions per generar les renderitzacions realitzades en la memòria amb Radiance.

1. Descarregar el repositori que es troba [aquí](#) o bé executar la següent comanda en la consola:

```
:~$ wget \  
https://github.com/miriam-mendez/Pedret/archive/refs/heads/main.zip
```

2. Anar al directori radiance i executar el makefile (crearà els fitxers.rtm)

```
:~radiance$ make lib
```

3. Anar al directori scene i al subdirector scene que es vulgui renderitzar. Finalment, executar el fitxer de control de l'escena:

- Mode interactiu: : radiance/scene/pedret1\$ rad -o x11 scene.rif
- Mode fotografia: : radiance/scene/pedret1\$ rad scene.rif

C CODI DE LA CONVERSIÓ

```
import bpy  
import os  
from mathutils import Vector  
  
class Scene:  
    """Holds the radiance scene definitions"""  
  
    def __init__(self, filename):  
        self.filename = filename  
        self.content = ""  
  
    def save(self):  
        f = open(self.filename, "w")  
        f.write(self.content)  
        f.close()  
  
    # Material defintions  
    def addMaterialLight(self, id, r, g, b):  
        self.content += f"void light {id} \n0 \n0 \n3 {r} {g} {b}\n"  
        self.content += "\n"  
  
    def addMaterialGlass(self, id, r, g, b):  
        self.content += f"void glass {id} \n0 \n0 \n3 {r} {g} {b}\n"  
        self.content += "\n"
```

```
def addMaterialPlastic(self, id, r, g, b, specular=0.0, roughness=0.0):
    self.content += f"void plastic {id} \n0 \n0 \n5 {r} {g} {b} \
        {specular} {roughness}\n"
    self.content += "\n"

def addMaterialColorTexture(self, id, textureHdr, specular=0, roughness=0):
    self.content += f"void colorpict {id + '_map'} \n7 red green blue \
        {textureHdr} . frac(Lu) frac(Lv) \n0 \n0"
    self.content += "\n"
    self.content += f"{id + '_map'} plastic {id} \n0 \n0 \n5 1 1 1 \
        {specular} {roughness}\n"
    self.content += "\n"

# Geometry definitions
def addMeshRtm(self, id, mat, file_rtm, xform=""):
    s = f"{mat} mesh {id} \n1 {file_rtm} \n0 \n0 \n"
    s += "\n"
    f = open("tmp.rad", "w")
    f.write(s)
    f.close()

    if xform:
        command = f"xform {xform} tmp.rad > tmp.rad"
        os.system(command)
    f = open("tmp.rad")
    self.content += f.read()
    f.close()

    command = f"rm tmp.rad"
    os.system(command)

def addSky(self, latitude, longitude, day, month, hour, year=""):
    self.content += f"!gensky {month} {day} {hour} -a {latitude} -o \
        {longitude}"
    self.content += f" -y {year}\n\n" if year != "" else "\n\n"

    self.content += f"skyfunc glow sky_glow \n0 \n0 \n4 .9 .9 1.15 0\n"
    self.content += f"sky_glow source sky \n0 \n0 \n4 0 0 1 180\n"
    self.content += f"skyfunc glow ground_glow \n0 \n0 \n4 1.4 .9 .6 0\n"
    self.content += f"ground_glow source ground \n0 \n0 \n4 0 0 -1 180\n"

def blend2mesh(materials, *argv):
    """Converts the blender scene meshes to radiance meshes"""
```



```
bpy.ops.object.select_all(action="DESELECT")
for arg in argv:
    ob = bpy.data.objects[arg]
    if ob.type == "MESH":
        ob.select_set(True)
        bpy.ops.export_scene.obj(
            filepath=f"{ob.name}.obj",
            use_selection=True,
            axis_forward="Y",
            axis_up="Z",
        )

        command = f"obj2mesh -a {materials} {ob.name}.obj {ob.name}.rtm"
        os.system(command)
        ob.select_set(False)
    else:
        raise Exception(f"{ob} is not a Blender mesh")

def objview(*argv):
    """Runs Radiance objview program"""
    command = f"objview"
    for arg in argv:
        command += f" {arg}"
    os.system(command)

def cam2view(camera):
    """Converts the Blender camera to Radiance view file"""
    ob = bpy.data.objects[camera]
    if ob.type == "CAMERA":
        lx, ly, lz = ob.location
        ux, uy, uz = ob.rotation_euler.to_matrix() @ Vector((0.0, 1.0, 0.0))
        dx, dy, dz = ob.rotation_euler.to_matrix() @ Vector((0.0, 0.0, -1.0))
        v = f"rvu -vtv -vp {lx} {ly} {lz} -vd {dx} {dy} {dz} -vu {ux} {uy} {uz}"

        f = open(f"{camera}.vf", "w")
        f.write(v)
        f.close()
        return f"{camera}.vf"
    else:
        raise Exception(f"{camera} is not a Blender camera")

def rad_interact(view, *argv):
```

A. Annexos

```
    """Runs Radiance rvu program"""
    command = f"oconv {' '.join(argv)} > scene.oct"
    os.system(command)
    command = f"rvu -vf {view} -ab 4 scene.oct"
    os.system(command)

def rad_image(view, *argv):
    """Runs Radiance rpict program"""
    command = f"oconv {' '.join(argv)} > scene.oct"
    os.system(command)
    command = f"rpict -vf {view} -ab 4 scene.oct > output.hdr"
    os.system(command)

def main():
    m = Scene("materials.mat")
    m.addMaterialColorTexture("grass", "Pedret_IX_grass.hdr")
    m.addMaterialColorTexture("wood", "Pedret_IX_wood.hdr", 0.07, 0.15)
    m.addMaterialColorTexture("sedimentary", "Pedret_IX_sedimentary.hdr", 0, 0.2)
    m.addMaterialGlass("window", 0.96, 0.96, 0.96)
    m.save()

    blend2mesh(m.filename, "pedret2", "windows")

    s = Scene("geometry.rad")
    s.addMeshRtm("pedret1", "void", "pedret2.rtm")
    s.addMeshRtm("gwindow", "void", "windows.rtm")
    s.addSky(42.10745931228419, 1.8836540623509863, 1, 8, "16:00CEST", 1990)
    s.save()
    # objview(s.filename)

    view = cam2view("Door")
    rad_interact(view, s.filename)

main()
```