

Introducere

Titlul temei este “Proiectarea și implementarea unei aplicații interactive, în Realitate Virtuală, pentru explorarea corpului B al Universității. Modelarea 3D pe bază imaginilor”. Tema a fost abordată pentru viitorii studenți care doresc să se înscrie la Universitate și doresc să facă un tur virtual al instituției și de asemenea pentru persoanele care doresc să ia la cunoștință câteva din obiectele de lucru de care dispune facultatea.

Am dorit să abordez această temă pentru pasiunea modelării 3D a obiectelor, dar și pasiunea pentru realitatea virtuală care este una din tehnologiile de ultimă generație. Am ales corpul B al Universității datorită faptului că în acest corp am avut cele mai multe ore pe parcursul celor patru ani de învățământ universitar.

În primul capitol am descris câteva lucruri de bază despre realitatea virtuală, mediile posibile pentru implementarea unor aplicații generale și resursele hardware utilizate.

În capitolul doi este descrisă proiectarea și implementarea aplicației, rezultatele cercetării teoretice, aplicative și experimentale, produsele informatice îmbunătățite și de asemenea și pașii de realizare a aplicației.

În capitolul trei se regăsește partea de testare a aplicației.

În al patrulea capitol sunt prezentate performanțele și concluziile elaborate pe baza cercetării efectuate în domeniul temei abordate.

Odată cu evoluția tehnologiei, producătorii de gadget-uri cum ar fi ochelarii de Realitate Virtuală încearcă să pătrundă în toate sectoarele. Astfel cei interesați pot să “viziteze” o stradă, să parcurgă un traseu, să exploreze lucruri noi și chiar să călătorească în timp sau în cazul nostru să facă un tur al Universității. Realitatea virtuală poate oferi o dorință a persoanelor de a explora mai mult.

Realitatea virtuală este un mediu în curs de dezvoltare care are potențialul să schimbe și să improvizeze modul în care studenții sunt instruiți în domenii diverse. Astfel, Realitatea Virtuală este o nouă tehnică în educația de nivel înalt, care are o istorie destul de lungă în aplicațiile de formare. De exemplu, VR-ul este folosit în sectorul militar de foarte mulți ani [1, 2]. Astronauții și piloții de avioane folosesc dispozitivele VR, cum ar fi ochelarii VR pentru a naviga, a zbura și chiar pentru a reacționa la situații neașteptate înainte de a pilota un avion real [3]. Luptele sunt simulate și ținute pentru a explora diverse manevre și posibile rezultate cum ar fi pregătirea pentru confruntări reale [4-6]. Tehnicile și echipamentele

dezvoltate de NASA și de organizațiile militare sunt introduse în aplicații care nu au legătură cu spațiul sau cu organizațiile militare.

În același timp, universitățile din întreaga lume își dezvoltă laboratoarele bazate pe Realitatea Virtuală, graficele pe calculator și vizualizarea științifică. Creșterea rapidă a internetului ca un mediu de comunicare și schimb de informație accelerează diseminarea informațiilor despre aplicațiile educative ale Realității Virtuale.

Studierea realității a reprezentat un obiectiv încă de la începuturile umanității, astfel au apărut o serie de teorii ale adevărului ajungându-se până la teorii ale realității în funcție de subiectul cunoscător.

Printre aplicațiile importante în inteligența artificială se numără și Realitatea Virtuală care este doar o oglindire a realității așa cum este percepută de organele noastre de simț, o copie a realității înconjurătoare, o lume a viselor.

Scopul acestei lucrări este de a studia modalitățile de proiectare și implementare a modelării pe baza imaginilor a unei încăperi în Realitate Virtuală care este o modalitate naturală de a interacționa cu tehnologia.

Realitatea virtuală are foarte multe de oferit în inginerie așa cum vom vedea în continuare.

Capitolul 1. Instrumente specifice Realității Virtuale

1.1 Realitatea virtuală – generalități

Realitatea virtuală se referă la un cadru artificial creat pe calculator care oferă o simulare a realității care îi da utilizatorului impresia de prezență fizică, aproape reală în locuri atât reale cât și imaginare. Aceasta datează încă de la sfârșitul anilor '40 – începutul anilor '50, nefiind o tehnologie nouă, ea progresând foarte mult pe parcursul anilor.

Realitatea Virtuală este un mediu extrem de interactiv creat pe calculator care poate fi bazat pe grafice sau text. În cea mai dezvoltată parte a Realității Virtuale, o simulare computațională a realității 3D afundă utilizatorul atât de adânc în această lume creată pe calculator încât dă impresia de lume reală. În cea mai simplă formă, utilizatorul se poate deplasa într-un ecran ce redă impresia lumii virtuale.

1.1.1. Tipuri de Realitate Virtuală

Brill [7] a propus o taxonomie a Realității Virtuale astfel:

- **Imersiune la persoana I** – un sistem pe deplin imersiv care folosește echipamente cum ar fi ochelari cu display stereoscopic, mănuși și costume cu senzori și un sistem stereo.
- **Simulator al mediului înconjurător** – în care utilizatorii stau într-un vehicul care simulează faptul că cei care îl folosesc sunt în mijlocul unei activități.
- **În spatele ecranului** – unde utilizatorul ia contactul cu lumea 3D prin spatele unui ecran și folosește un joystick (sau un mouse obișnuit, 3D controller sau alte astfel de echipamente) pentru a se deplasa prin această lume 3D.
- **O lume oglindită** – utilizatorii văd dubluri electronice sau video ale lor pe care le pot controla folosind imaginea lor.
- **Lumea Waldo (științifică)** – în care operatorul este conectat din lumea reală în lumea VR cu ajutorul unei telecomenzi printr-o pereche de ochelari sau alte astfel de tipuri de echipamente.
- **CAVE** – un teatru cu un proiector real 3D realizat din trei pereți și o podea, proiectat într-un sistem stereo și vizualizat prin intermediul unor ochelari stereo 3D.

În general echipamentele care interacționează cât mai puțin cu corpul sunt cele mai ieftine.

- Bazate pe text
- Grafice
- Tridimensionale

[illegible]

4

Realitatea virtuală bazată pe text este mult mai imaginativă decât cea bazată pe senzori. Utilizatorii citesc textul și discutând pe baza vizualizării obiectelor creează o realitate în mintea lor. Fiecare obiect într-o MOO bazată pe text este descrisă textual, chiar și caracterul utilizatorului creat chiar de acesta.

Mediile bazate pe grafică sunt de asemenea numite MOO, dar sunt referite ca niște grafice. Diferența principală dintre ele și MOO-urile bazate pe text este aceea că ele încorporează grafice pentru a înfățișa aria în care sunt utilizatorii, de asemenea și utilizatorii în sine. Textul este limitat pentru interacțiunea între utilizatori, el fiind de obicei scris în casete de text. Un exemplu de grafică MOO este The Palace (<http://www.thepalace.com/>), după cum se poate observa în figura 1.2. Acest mediu nu a fost neapărat destinat învățării unei limbi, dar este deschis publicului și este foarte popular printre tineri.

Chiar dacă mediul Realității Virtuale este deschis publicului și permite studenților să întâlnească o multitudine de oameni, are și câteva probleme. Limba folosită în The Palace este plină de expresii jargonice ale studenților, iar aceasta îi poate ofensa pe alții.



Figura 1.2. Screenshot din The Palace.[25]

Mediul tridimensional folosește grafica tridimensională să portretizeze decorul și utilizatorii. În mediile tridimensionale, o persoană poate observa schimbările care se întâmplă în jurului ei atunci când avatarul respectivei persoane este în mișcare. O persoană poate de asemenea să miște avatarul în jur pentru a vedea o vizualizare a scenariului la 360 grade. Avatarele pot de asemenea să danseze și să efectueze o multitudine de mișcări. Un exemplu este mediul Active Worlds (<https://www.activeworlds.com/index.php>). Acest site nu utilizează atât de multă învățare a unei limbi din cauza cerințelor pe care le plasează pe partea hardware a calculatorului și a vitezei și a lungimii de bandă a conexiunii cu internetul.

1.1.2. Cum funcționează tehnologia VR?

Principiul care stă la baza tehnologiilor VR este tehnica stereoscopică, adică o imagine 2D, simplă, este dublată – una pentru fiecare ochi, fiecare imagine fiind apoi distorsionată creând efectul 3D, ca în figura 1.3. Această imagine distorsionată ajunge la lentilele ochelarilor VR, unde este mărită, ajungând după aceea la ochi.

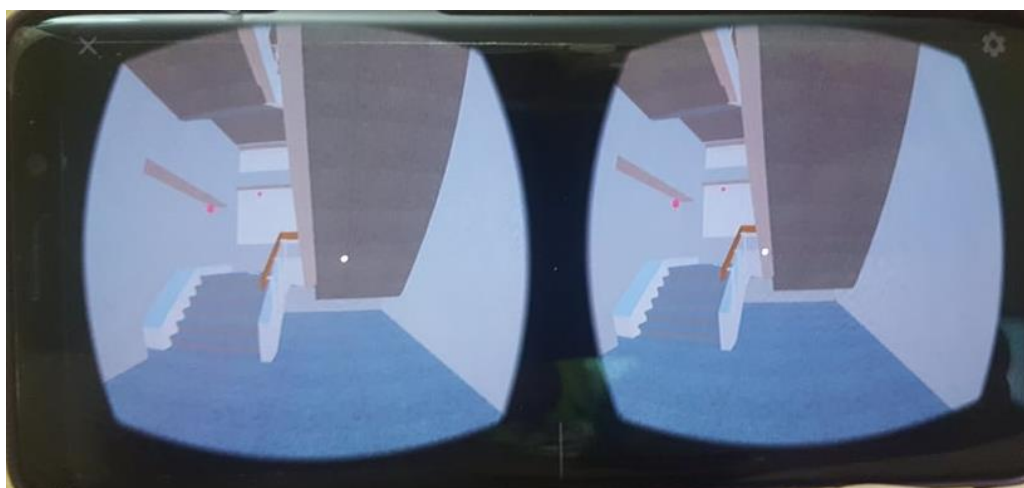


Figura 1.3 Imagine distorsionată realizată de pe telefon.

1.1.3. Cum se folosește tehnologia VR?

Pentru realizarea efectului de Realitate Virtuală, avem nevoie de o pereche de ochelari VR și un telefon compatibil care folosește ecranul său pentru afișare sau de un calculator care se conectează printr-un cablu de tip USB care conține și un ecran stereoscopic.

1.1.4. Când se utilizează Realitatea Virtuală?

Realitatea Virtuală are trei roluri importante în educația în inginerie:

- De subiect de dezbătut (învățat);

- De profesor care să învețe despre realitatea virtuală în sine;
- De profesor care să îi învețe pe alții despre acest subiect.

Realitatea Virtuală ca subiect de învățat poate fi susținută cu ajutorul unui laborator VR. Realitatea Virtuală poate fi folosită pentru a preda despre ea însăși tot cu ajutorul unui laborator VR. Rolul major al realității virtuale este utilizarea sa pentru a preda un subiect.

Deoarece Realitatea Virtuală este o simulare, ea poate fi considerată pentru utilizarea în studiul ingineriei oricând se poate utiliza o simulare. Poate fi utilizată ca o simulare care modelează lumea reală cât mai obiectiv posibil sau poate fi folosită ca o simulare a unei lumi imaginare.

Realitatea Virtuală poate fi folosită mai ales atunci când învățarea folosind lucruri, echipamente reale poate fi:

- periculoasă (ex. accidentarea unui student sau profesor);
- imposibilă (ex. mediul necesar nu poate fi experimentat în lumea reală);
- inconvenientă;
- consumă prea mult timp;
- prea costisitoare.

Realitatea Virtuală poate fi folosită când greșelile realizate de către utilizator pot fi:

- devastatoare sau demoralizatoare pentru cursant;
- dăunătoare pentru mediu;
- capabile să cauzeze distrugerea neintenționată a unui bun;
- capabile să realizeze distrugerea unui echipament;
- costisitoare;
- cauzatoare a unei risipe de materiale;

Realitatea Virtuală poate fi de asemenea utilizată atunci când:

- un model al unui mediu poate să simuleze și să învețe utilizatorul ca și când ar fi echipament real;
- interacțiunea cu un model poate să fie mult mai captivantă și motivantă decât atunci când se folosește un obiect real;

- costul, călătoria și timpul de planificare al unei clase este mult mai mare decât dacă s-ar folosi un model VR interactiv;
- experiența împărtășită de către un grup într-un anumit mediu este importantă;
- experiența creării unui mediu sau model simulat este importantă pentru învățarea obiectivă;
- vizualizarea informației este necesară de mai multe persoane astfel încât manipularea și rearanjarea informației să fie mult mai ușor de realizat;
- situațiile de instruire trebuie realizate “foarte real” (ex. experiența practică sub condiții reale) și trebuie realizat perceptibil imperceptibilul;
- se dezvoltă medii participative și activități care pot exista numai ca lumi generate pe calculator;
- se învață activități care implică dexteritate manuală sau mișcare fizică;
- realizarea unui mod de învățare interesant e esențial;
- lumea imaginară poate fi folosită pentru a testa terenuri și platforme pentru noi abordări;

În plus, P. Jackson [8] a listat următoarele situații ca fiind “domenii de instruire adecvate” în care Realitatea Virtuală poate fi folosită, unde oamenii utilizează echipamentele în locuri izolate de alte persoane sau de comunicația prin intermediul vocii, unde echipamentul restrictiv, limitat este purtat în lumea reală, unde mediul lumii reale este deja bazat pe VR, unde sesiunea de antrenament este prea scurtă și unde lumea reală poate fi distorsionată și scalată.

De asemenea există și situații în care învățarea prin intermediul Realității Virtuale nu este potrivită. Aceasta nu ar trebui folosit în scopuri educaționale sau sesiuni de antrenament care includ echipamente de protecție și operarea cu acestea, contactul direct cu diverse fluide, interacțiunea cu corpul unei persoane, mișcări rapide și pe scară largă, contactul foarte apropiat cu o altă persoană și operațiunile într-o echipă unde interacțiunea apropiată este necesară.

1.1.5. Modelul pentru determinarea utilizării VR

Educația în inginerie ocupă foarte multe arii și este imposibil și impracticabil să încercăm să schițăm o utilizare exactă pentru o disciplină sau domeniu. Pentru a trece peste

această problemă, se va prezenta în continuare un model pentru a înțelege când se va folosi VR-ul într-un curs oarecare.

În modelul următor se definesc obiectivele specifice cursului. Obiectivele care pot să folosească o simulare ca o măsură sau însemnătate pentru o realizare sunt marcate. Fiecare obiectiv marcat este examinat, primul pentru a determina care obiectiv poate folosi o simulare realizată pe calculator, apoi determină care din aceste obiective poate folosi o simulare interactivă 3D.

Nivelul de realism cerut este determinat pentru obiectivele selectate pe o scară de la cele mai simbolice la cele mai reale obiective. Tipul de interacțiune necesar este determinat pe o scară de la cel mai puțin imersiv până la cel mai imersiv tip. Astfel este determinat tipul de senzori din lumea virtuală sau mediul dorit.

În conformitate cu aceste alegeri, software-ul sau echipamentul hardware este ales. Mediul Virtual (VE) este realizat și construit. Rezultatele VE sunt evaluate folosind un grup pilot. Rezultatele evaluării sunt utilizate pentru a modifica VE. Evaluarea și modificarea continuă până când mediul virtual se arată a fi un succes din punct de vedere al măsurătorilor sau până când se ating obiectivele impuse.

Acest model este reprezentat în figura 1.4.

1.1.6. Avantajele tehnologiei VR

Realitatea Virtuală este folosită de Samsung într-o campanie numită “Be Fearless” pentru a ajuta oamenii să combată și să își învige o frică cum ar fi frica de înălțime, vorbitul în public, tulburări de alimentație, pregătirea și execuția unei operații etc.

Școlile de medicină au adoptat Realitatea Virtuală ca un mod pentru a pregăti generațiile următoare de chirurghi.

Sectorul de sănătate este un sector care utilizează Realitatea Virtuală pe scară largă, dar există și alte sectoare care au adoptat această tehnologie în scopuri de formare cum ar fi: educație, armată, construcție, telecomunicații și business.

Beneficii ale Realității Virtuale:

- Risc redus
- Scenarii realiste
- Potrivită pentru diverse forme de învățământ

- Arii controlate
- Zone sigure
- Procesele pot fi controlate de la distanță, economisind timp și bani
- Simplifică problemele/situațiile complexe
- Sunt inovative și agreabile

Instruirea este mult mai ușoară dacă experiența este plăcută și agreabilă ceea ce înseamnă un nivel înalt al implicării și înțelegerii.

Timpul și banii sunt un aspect important. Instruirea este un aspect necesar pentru a-i ajuta pe oameni să-și îndeplinească sarcinile la locul de muncă sau să învețe un subiect pentru a fi pe deplin productivi, însă costurile pot fi ridicate. Un exemplu ar fi dezvoltarea unei serii de prototipuri. Realitatea Virtuală elimină nevoile de realizări repetitive ale prototipurilor și implementării care sunt foarte costisitoare și le înlocuiește cu un singur model care poate fi folosit de mai multe ori și poate fi accesat din diferite locații ceea ce poate salva foarte mult timp, bani și poate fi accesat de către toți angajații unei firme fără a fi nevoie de produsul real care poate fi avariât.

Realitatea Virtuală este folosită pentru simularea modului în care anumite echipamente răspund la comenzi, îmbunătățirea modului în care funcționează anumite mașinării sau pentru a reproduce abilitățile software precum acțiunile și comportamentul uman. De asemenea, instruirea personalului poate fi riscantă datorită riscului ridicat de accidentarea al angajaților. Realitatea virtuală poate fi folosită în acest caz pentru a simula o situație potențial dăunătoare din viață reală sau să se reproducă un echipament care poate fi periculos pentru angajați.

În Realitatea Virtuală se pot crea de asemenea grafice uimitoare, scenarii interactive și fișiere audio.

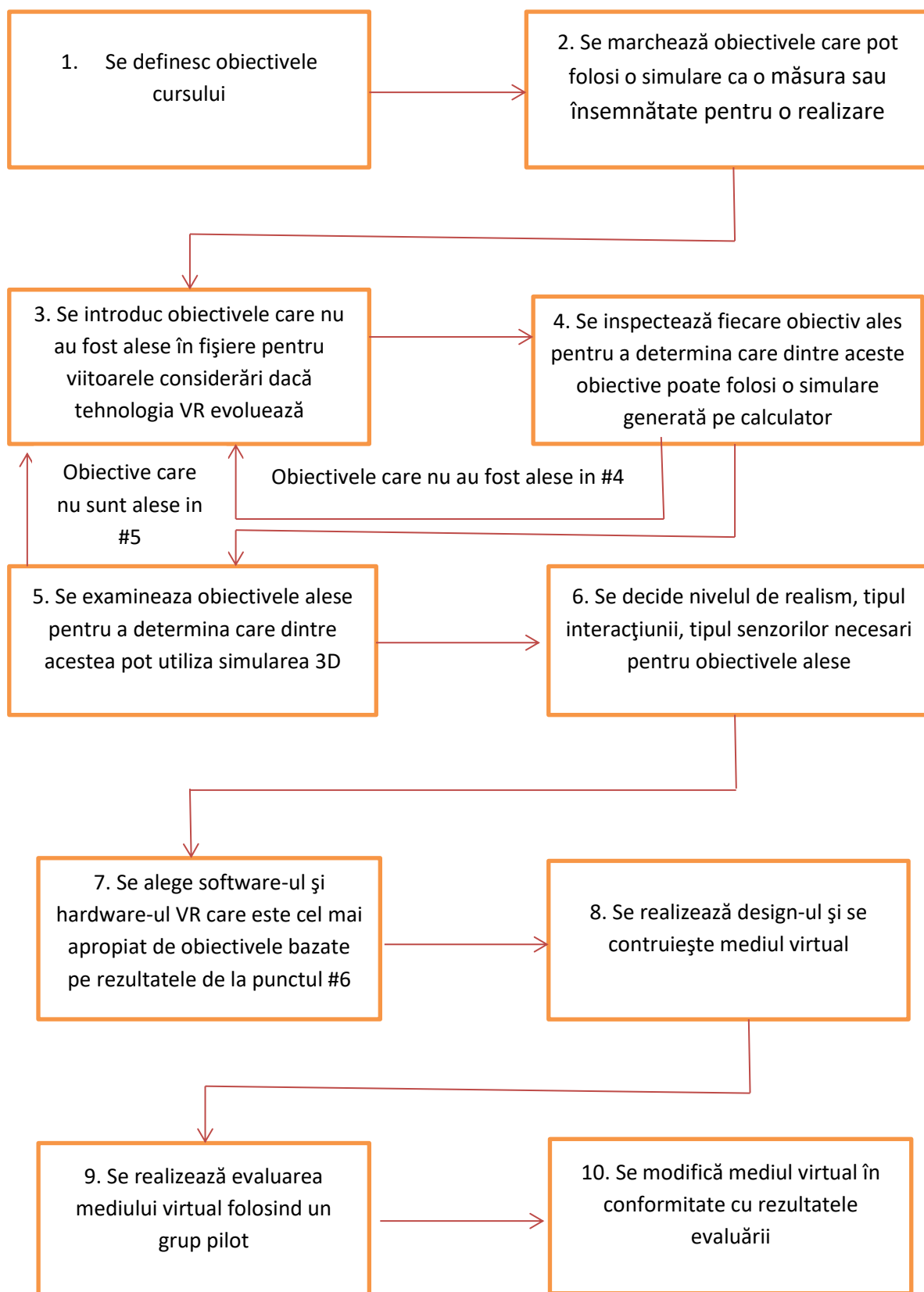


Figura 1.4 Modelul pentru determinarea utilizării VR.

1.1.7. Dezavantajele tehnologiei VR

Realitatea Virtuală, după cum am putut observa mai înainte, are foarte multe avantaje, însă acest tip de tehnologie, pe lângă avantajele care sunt evidente, are dezavantaje atât la nivel de grup, cât și la nivel de individ.

Unul din dezavantajele importante este acela că în viitor, pe măsură ce tehnologia avansează, diferență dintre realitate și virtual va fi foarte greu de diferențiat. Putem lua ca exemplu jocurile video în care există vieți nelimitate, însă în lumea reală există o singură viață care ar putea fi pierdută în spatele unor echipamente specifice Realității Virtuale, dacă persoana care le mănuieste nu poate face diferența dintre cele două lumi.

Un alt dezavantaj ar fi faptul că această lume imaginară, lumea realității virtuale, poate crea o dependență care se manifestă exact ca un drog, ceea ce nu constituie un mod de viață realist. De asemenea, dacă o parte foarte bună de populație ar folosi echipamente VR, nu va mai exista comunicare între persoane, astfel că o parte din locurile de muncă nu ar mai exista. De exemplu, o persoană va prefera să joace un meci de fotbal sau să participe la un astfel de meci în Realitatea Virtuală pentru că ar fi mult mai comod acest fapt decât să se deplaseze până la stadion.

1.1.8. Motivele utilizării Realității Virtuale în studiul ingineriei

Există multe motive pentru care Realitatea Virtuală ar trebui folosită în inginerie cum ar fi:

- Oferă motivație;
- Poate ilustra mult mai bine anumite caracteristici, procese;
- Permite o examinare foarte detaliată a unui obiect;
- Permite examinarea unui obiect de la distanță;
- Permite persoanelor cu dizabilități să participe la experimente sau să învețe despre un anumit domeniu;
- Oferă oportunitatea de a dezvolta intuiții despre anumite perspective;
- Permite unui student să proceseze informația acasă la el;
- Permite studenților să învețe pe perioadă lungă de timp, neprogramată de un orar școlar;

- Permite experiența unei tehnologii noi prin utilizarea reală a echipamentelor;
- Necesită interacțiunea, adică încurajează participarea activă a studenților la ore;

- Permite studentului să facă descoperiri pe care ulterior nu le cunoștea;

1.1.9. Aplicabilitatea tehnologiei VR

Pentru început ar trebui să observăm majoritatea utilizatorilor VR cum ar fi armata, industria de divertisment, antrenament spațial (ex. NASA) și pregătirea medicală cu VR pentru a vedea ce au realizat pentru educația în inginerie.

Câteva aplicații curente realizate în VR pentru educația în inginerie au fost publicate în literatură. Pe de altă parte, profesorii care predau ingineria dezvoltă modele VR pentru a le folosi la cursurile lor. P. Jackson a discutat câteva dintre aceste aplicații dintr-o perspectivă de învățare.

Mediile virtuale care au fost dezvoltate pentru învățarea folosind software-ul VR includ arii cum ar fi fizica și știința. Un laborator virtual de fizică care dezvoltă aplicații în inginerie a fost construit de Loftin și colegii săi la Universitatea Houston–Downtown, LinCom Corporation și Universitatea din California de Sud [9-11]. Un alt exemplu este laboratorul de Inginerie Virtuală proiectat și dezvoltat de Bourne and Brodersen la Universitatea Vanderbilt [12,13]. Un laborator virtual de știință pentru oamenii cu dizabilități care include cursuri de interes pentru profesorii din inginerie a fost dezvoltat de Nemire of Interface Technologies (Capitola, California) [14,15].

Utilizările VR-ului pentru învățarea cursurilor tehnice este un subiect de dezbătut și explorat. Câteva posibile aplicații includ: realizarea rapidă a prototipurilor [16], proiectarea asistată pe calculator a clădirilor [17], realizarea design-ului prin simularea fabricațiilor [18], chimia computațională, vizualizarea complexă a iluminării, simularea mediului de trai [19], telerobotica în industria nucleară [20].

Comitetul Dezvoltării și Cercetării Realității Virtuale înființat de Consiliul Cercetării Naționale au descoperit că Realitatea Virtuală are un potențial de aplicabilitate extrem de mare în vizualizarea științifică și a listat ca exemple proiecte în ingineria aeronautică, vizualizarea medicală, astrofizică [21].

Realitatea Virtuală este explorată și folosită în multe programe de training din industrie. De exemplu, Motorola utilizează Realitatea Virtuală pentru a învăța asociații să

opereze cu roboții[22]. Pentru un cost de 30,000-100,000 \$, compania poate trimite un ansamblu de laborator VR pentru instruire oriunde în lume.

1.1.10. Trendurile actuale și viitoare

În acest moment, laboratoarele VR se dezvoltă foarte rapid mai ales acum când lumea ia amploare foarte rapid din punct de vedere a tot ceea ce înseamnă aplicațiile nevoile de cercetare care se referă la Realitatea Virtuală, în special conexiunile de telecomunicații cum ar fi Internetul [23]. Prețurile majorităților tipuri de echipamente continuă să scadă, ducând la posibilități mult mai mari de laboratoare VR accesibile. Conversațiile cu personalul în mai multe companii care au utilizat înainte echipament VR scump indică un trend în folosirea echipamentelor hardware și software mult mai ieftine. Odată cu scăderea prețurilor, mult mai mulți profesori vor deveni familiari cu folosirea Realității Virtuale și dezvoltarea tehnicilor de învățare folosind VR-ul. Multe din acestea vor fi utilizate pentru educarea studenților din inginerie.

Viitoarele trenduri vor aduce laboratoare cu echipamente VR pentru a susține programul pentru educația în inginerie. Datorită faptului că Realitatea Virtuală este într-o continuă schimbare va încuraja utilizarea acesteia în clasele de studiu. Au fost predicții care spun că în mai puțin de 10 ani va exista o punte holografică precum se regăsește în seria de televiziune Star Trek realizată de U.S.S. Enterprise. Într-adevăr, Armata Statelor Unite dezvoltă programe holografice folosite pentru antrenament[24]. Un astfel de mediu de simulare poate oferi studentului senzația de o situație aproape reală. Soluțiile problemelor ingineriei pe care un student le poate învăța astăzi dintr-o carte, pot fi mâine învățate într-un mediu de simulare. Fie că se utilizează tehnologia Realității virtuale, interfață om-robot, sisteme de simulare care folosesc costume sau un calculator VR care folosește programe la un preț redus, Realitatea Virtuală va joaca un rol principal în educație în viitor. Realitatea Virtuală va schimba modul în care informațiile în timpul cursurilor este transmisă și educația va beneficia în urma acesteia.

1.2. Elemente de bază folosite în implementarea aplicațiilor VR

1.2.1. Resurse hardware

De preferat, pentru a rula o aplicație VR la un nivel calitativ ridicat este nevoie de calculatoare puternice. Pe partea de hardware este recomandat ca acestea să integreze componente precum procesoare multicore, memorie RAM cu capacitate cât mai mare, o placă video dedicată performantă (High End). Cheia principală pentru rularea aplicației este placa

video. Un exemplu de calculator care poate fi utilizat la rularea acestei aplicații ar putea avea în componența un procesor AMD RYZEN 5 1600 (6 nuclee, 12 file de execuție, 3,2 Ghz), memorie RAM HyperX 16 GB 3000 Mhz, placă video Nvidia GTX 1070 8GB.

Din punctul de vedere al resurselor utilizate pentru dezvoltarea aplicației, am folosit un calculator ale cărui componente sunt: procesor AMD RYZEN 5, 8 GB RAM, placa video Nvidia GTX 1070 8GB.

Pentru rularea aplicației ca și resurse am folosit o pereche de ochelari Google Daydream și un telefon Samsung Galaxy S9 dotat cu un procesor Exynos 9810 Octa Core, 4GB RAM, placa video Mali-G72 MP18, memorie internă 64 GB. Am folosit acest model de telefon deoarece este construit pentru Realitatea Virtuală cu o rezoluție mare a display-ului, nivelul de grafică foarte ridicat și senzori de înaltă fidelitate.

1.2.2. Resurse software

Câteva produse clasice cu ajutorul cărora se pot implementa aplicațiile VR:

- Unity 3D – este unul din cele mai populare instrumente folosite astăzi pentru dezvoltarea Realității Virtuale
- Unreal Engine – unul din cei mai mari competitori ai motorului grafic Unity 3D, Unreal Engine este de asemenea un motor de jocuri cu integrare VR și cu o documentație vastă.
- 3DS Max & Maya – sunt produse Autodesk pentru modelare, animație și VFX.
- Blender – a devenit rapid unul din programele favorite de modelare pentru dezvoltatorii VR
- SketchUp – este o aplicație cu o curbă de învățare foarte scăzută care poate să pregătească pe oricine într-o perioadă scurtă de timp
- WebVR Tools – sunt unelte pentru dezvoltarea WebVR-ului pentru diverse browsere
- Three.js – este o bibliotecă JavaScript care lucrează ca un layer peste WebGL (este un OpenGL care este implementat printre browsere moderne precum Chrome, Firefox și Safari)

Capitolul 2. Proiectarea și implementarea aplicației de explorare a corpului B al universității

2.1. O privire de ansamblu asupra aplicației de VR

Principiul care stă la baza aplicației este explorarea corpului B al Universității. Am ales această temă datorită pasiunii modelării 3D a obiectelor, dar și pasiunea pentru Realitatea Virtuală care este una din tehnologiile de ultimă generație. Am ales corpul B al Universității datorită faptului că în acest corp am avut cele mai multe ore pe parcursul celor patru ani de învățământ universitar. Tema a fost abordată pentru viitorii studenți care doresc să se înscrie la Universitate și doresc să facă un tur virtual în instituție și de asemenea pentru persoanele care doresc să ia la cunoștință câteva din obiectele de lucru de care dispune facultatea. Au fost urmați pașii indicați în figura 2.1.

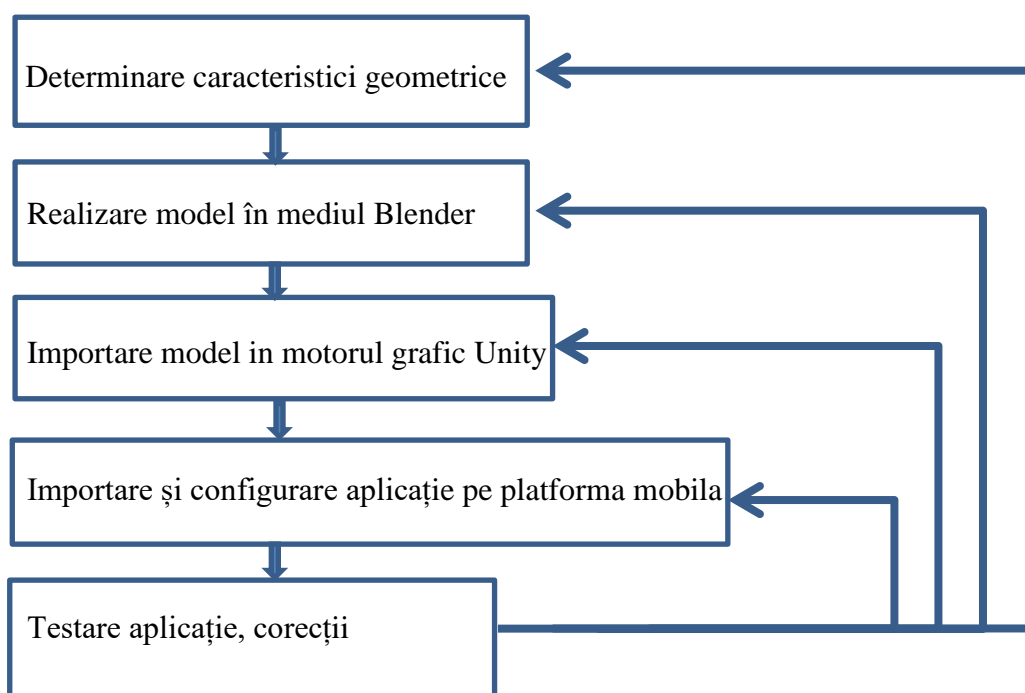


Fig. 2.1. Planul general al aplicației de VR construite.

Pașii de realizare a aplicației

1. Am determinat caracteristicile geometrice ale corpului B cu ajutorul unui telemetru și am realizat poze ale holurilor pentru începerea construirii corpului B.
2. Am pregătit programul de modelare 3D Blender.
3. Am început să construim corpul B în Blender.

4. Am texturat toate elementele din corpul B în Blender.
5. Am pregătit motorul grafic Unity pentru Realitatea Virtuală.
6. După realizarea construcției am importat modelul obținut în Unity.
7. Am codificat elementele construcției.
8. Am texturat elementele în Unity.
9. Am adăugat și codificat waypoint-uri pentru realizarea mișcării.
10. Am importat pe telefon aplicația și am conectat ochelarii la telefon pentru testarea acesteia.

2.2. Determinarea caracteristicilor geometrice ale corpului B

O aplicație interactivă este o aplicație care îl lăsa pe utilizator să preia controlul asupra a ceea ce se întâmplă în interiorul aplicației. Realitatea Virtuală se referă la ambianțe artificiale create pe calculator care oferă o simulare a realității atât de reușită, încât utilizatorul poate căpăta impresia de prezență fizică, atât în anumite locuri reale, cât și în locuri imaginare.

Pentru realizarea aplicației am achiziționat un telemetru cu ajutorul căruia am măsurat dimensiunile corpului B și o pereche de ochelari Google Daydream, una din componentele principale care ajută la utilizarea aplicației

Pasul 1. Măsurarea fizică a corpului B.

Cu ajutorul unui telemetru am măsurat corpul B al Universității după cum se poate vedea în figura 2.2.



Figura 2.2.a. Realizarea măsurătorilor etajului II al Universității cu ajutorul unui telemetru.



Figura 2.2.b. Realizarea măsurătorilor etajului II al Universității cu ajutorul unui telemetru.

Am întâmpinat dificultăți în momentul realizării corpului în Blender după măsurători. Chiar dacă am luat prima dată dimensiunile fiecărui perete și ușă, la construirea lor în Blender, ori rămânea spațiu prea mare la finalul construcției unui perete, ori nu rămânea spațiu pentru realizarea unei uși, astfel încât a trebuit să luăm din nou dimensiunile cu telemetrul în Universitate.

2.3. Realizare model in mediul Blender

Pasul 2. Pregătirea programului de modelare 3D Blender.

Fundația Blender este o corporație cu beneficii publice creată pentru a susține și a facilita proiectele pe “blender.org”.

Toate persoanele sunt libere să folosească Blender pentru orice scop, incluzând scopul comercial sau educațional. Această libertate este data de către Blender’s GNU General Public License (GPL).

Blender este un program gratuit și open source de creare 3D. Sprijină toate tipurile de creare 3D – modelare, montare, animație, simulare, randare. Compoziția și urmărirea mișcării, editarea video și crearea video. Utilizatorii avansați folosesc API-ul (Aplicacion Programming Interface – reprezintă un set de funcții și proceduri care permit crearea aplicațiilor care accesează caracteristicile și datele unui sistem operabil, unei aplicații sau unui alt serviciu) Blender-ului pentru scripturile scrise în Python pentru a personaliza aplicațiile și pentru a scrie instrumente de specialitate, care sunt adesea incluse în viitoarele versiuni ale Blender-ului. Blender este potrivit pentru persoane fizice și studiouri mici care beneficiază de pe urmă procesului de dezvoltare.

Blender este compus din mai multe platforme interconectate și rulează la fel de bine pe Linux, Windows și calculatoarele Macintosh (sunt o familie de calculatoare personale proiectate, fabricate și vândute de Apple Inc. din Ianuarie 1984). Interfețele lor folosesc OpenGL (Open Graphics Library – este compus din mai multe platforme interconectate de tip API în mai multe limbi pentru randarea 2D și 3D) pentru a oferi o experiență variată. Pentru a confirma o compatibilitate specifică, lista platformelor suportate indică acele platforme care sunt regular testate de echipă de dezvoltare.

Ca un proiect condus de comunitate în cadrul GNU General Public License (GPL), publicul este împuternicit să realizeze schimbări mici sau mari a bazei de coduri care duce la caracteristici noi, fixarea erorilor și o mai bună utilizare. Blender nu are un preț, însă oamenii pot investi, participa și ajuta la avansarea unui instrument puternic de colaborare. Blender este propriul software 3D al utilizatorului.

Primul element pentru modelarea aplicației a fost descărcarea și instalarea programului de modelare 3D denumit Blender, versiunea 2.79B (<https://www.blender.org/download/>). Pentru modelarea obiectelor 3D am avut nevoie de Add-on-uri, care ne permit să gestionăm scripturi secundare, care extind funcționalitatea Blender-ului. În figura 5 putem observa că putem cauta, instala, activa și dezactiva Add-on-uri.

Pentru a putea modifica Add-on-urile trebuie să tastăm Shift+Alt+U în Blender și va apărea căsuța din figura 2.3.

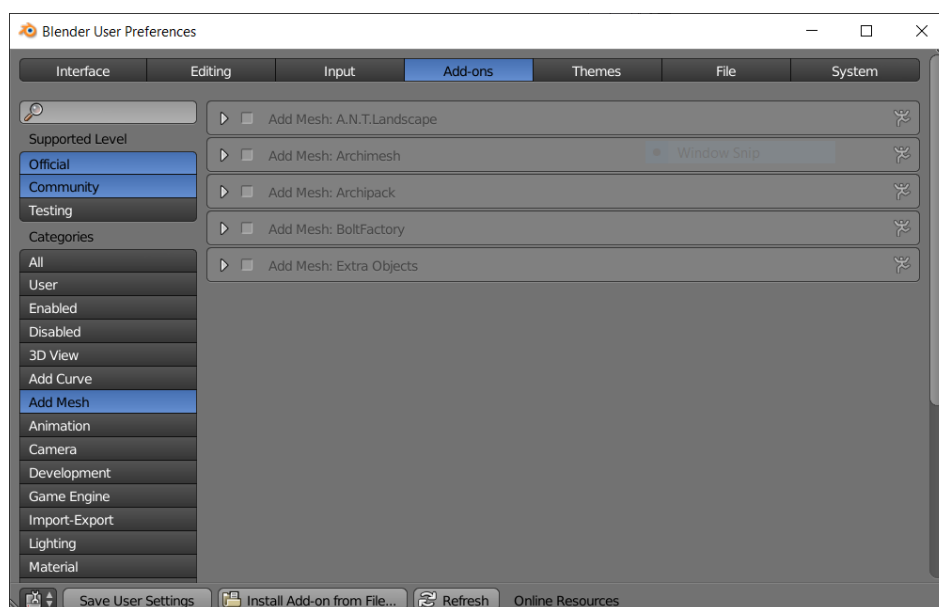


Figura 2.3. Utilizarea Add-on-urilor.

Blender vine cu o multitudine de Add-on-uri deja instalate, gata să fie activate. Dar de asemenea utilizatorul poate adăuga un add-on realizat de el sau un add-on găsit pe Internet.

Add-on-urile Blender-ului sunt împărțite în două grupuri în funcție de cine le scrie/le sprijină:

- Oficiale: Add-on-uri care sunt scrise de dezvoltatorii Blender
- Realizate de comunitate: Add-on-uri care sunt scrise de utilizatorii din comunitatea Blender.

Add-on-urile se împart în categorii în funcție de ce arie a Blender-ului folosesc .

Activarea și dezactivarea unui Add-on se realizează bifând sau debifând căsuța din dreapta Add-on-ului ales că în figura 2.4.

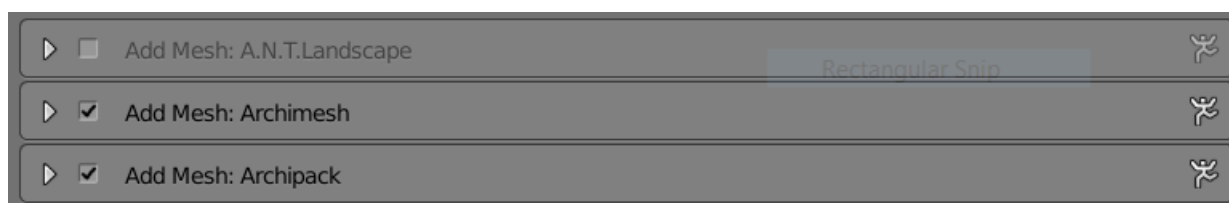


Figura 2.4. Activarea, dezactivarea Add-on-urilor.

Pentru Add-on-urile care se găsesc pe Internet sau realizate de utilizator, trebuie instalate realizând click pe “Install from File” așa cum se poate vedea în figura 5 și acestea trebuie să fie de tip “.zip” sau “.py”.

Acum Add-on-urile vor fi instalate dar nu vor fi automat activate. Se va da click pe Search pentru a caută Add-on-ul instalat și îl vom activa dând click pe căsuța din stânga denumirii Add-on-ului așa cum putem vedea în figura 2.4.

Add-on-urile utilizate în construirea corpului B al Universității sunt Add Mesh: Archimesh, Add Mesh: Archipack, Mesh: Bsurfaces GPL Edition, Mesh: F2, Mesh: Insert Polygon, Mesh: LoopTools, Mesh: Relax, Mesh: tinyCAD Mesh tools care se găsesc utilizând combinația de taste Shift+A.

Următorul element a fost să setăm unitățile de măsură în metri. Pentru aceasta vom da click în meniul “Scene” , “Units” de unde vom alege ca unități de măsură metri așa cum putem vedea în figura 2.5.

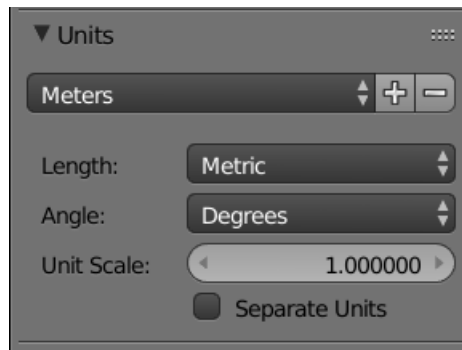


Figura 2.5. Setarea unităților de măsură.

Pasul 3. Construirea corpului B în Blender

Construirea scărilor a fost realizată cu ajutorul măsurătorilor luate cu telemetrul, conform figurii 2.6.



Figura 2.6. Măsurătoarea scărilor cu ajutorul telemetrului.

Apăsăm combinația de taste Shift+A, iar din meniul apărut vom selecta Cube. Vom apăsa pe Numpad_1 de la tastatură pentru a vedea cubul din față. Vom scala cubul apăsând pe tastă S. Vom da click pe Tab de la tastatură pentru a intra în Edit Mode, modul în care vom putea edita cubul și realiza scăările. Vom apăsa pe tasta A de două ori pentru a selecta tot cubul. Vom apăsa pe Ctrl+R de la tastatură, iar pe ecran va apărea o linie roz, verticală sau orizontală în funcție de plasare a mouse-ului pe cub. Vom alege o axă orizontală sau verticală pentru realizarea tăieturilor și vom da de la mouse Scroll Down pentru a apărea mai multe linii, astfel încât să împărțim cubul în mai multe părți. Așa se va proceda și pentru axă orizontală dacă mai devreme a fost selectată axă verticală sau invers. Astfel se vor apărea pe ecran mai multe cuburi care au fost realizate din cubul inițial așa cum putem observa în figura 2.7. Vom da click pe Face Select din meniul din bara de jos, după cum putem vedea în figura 2.8. După acest pas vom apăsa Click dreapta de la mouse și vom ține apăsat pe Ctrl+Click

Dreaptă pentru a putea selecta mai multe fețe așa cum putem vedea în figura 2.9. După ce vom selecta toate fețele pe care le dorim șterse, vom apăsa pe Delete de la tastatură și vom da Click pe Faces, pentru a șterge fețele selectate. După ce am șters fețele vom da click pe Edge Select, iconța din mijlocul figurii 2.8. Vom selecta muchiile opuse așa cum putem vedea în figura 2.9, după care vom apăsa tasta F pentru a putea umple spațiile goale. Vom apăsa A pentru a deselecta toate fețele, iar mai apoi vom apăsa Tab de la tastatură pentru a intra în Object Mode. În acest mod nu se mai poate edita nimic, doar vedea obiectul solid realizat în Edit Mode. În figura 2.10 sunt prezentate scările după editare.

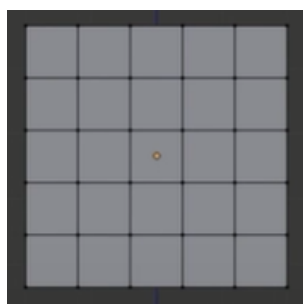


Figura 2.7. Împărțirea cubului în mai multe părți, pas ajutător la realizarea scărilor.



Figura 2.8. Face Select.

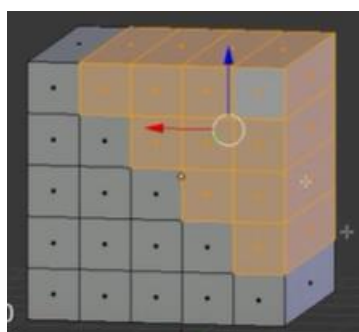


Figura 2.9. Fețele selectate ale cubului.

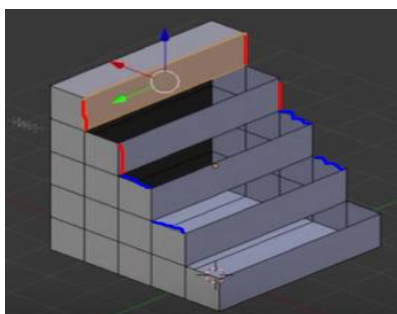


Figura 2.10. Realizarea umplerii golurilor cubului. Cu roșu sunt notate muchiile pe verticală, iar cu albastru muchiile pe orizontală.

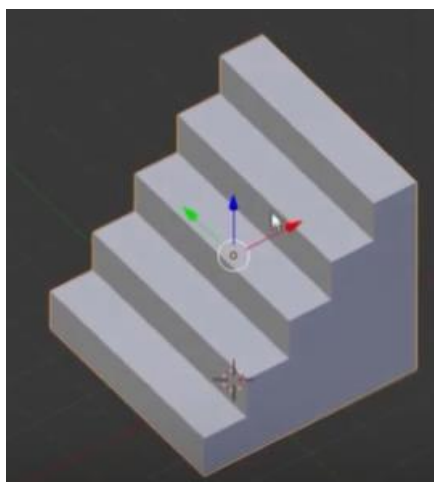
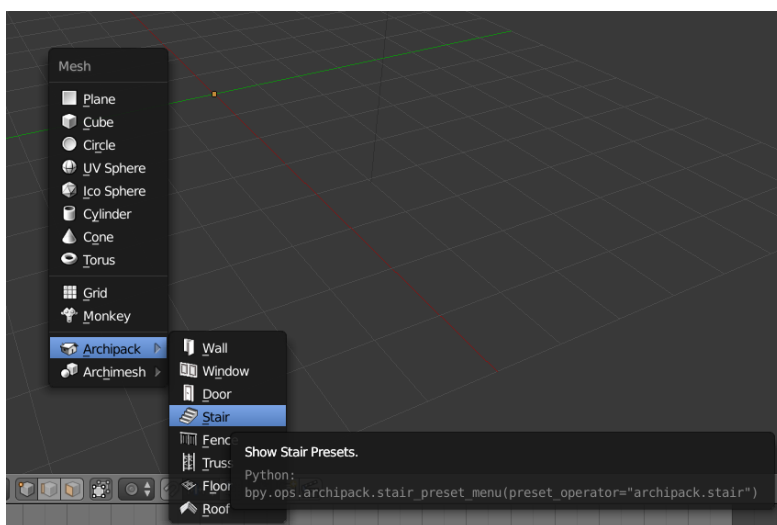


Figura 2.11. Scările realizate după editare

Un alt mod pe care l-am folosit pentru realizarea scărilor este să folosim un Add-on Mesh: Archipack prezentat la pasul 2. Vom da click pe Stair după cum putem observa în figura 2.12, după care pe ecran ne vor apărea mai multe tipuri de scări din care putem alege, conform figurii 2.13. Acest pachet pune la dispoziția utilizatorului un set de opțiuni de editare a scărilor, precum: înălțimea scărilor, lățimea scărilor, grosimea scărilor, numărul de trepte, balustrade, înclinarea scărilor.



Figură 2.12. Add-on-ul utilizat pentru crearea scărilor.

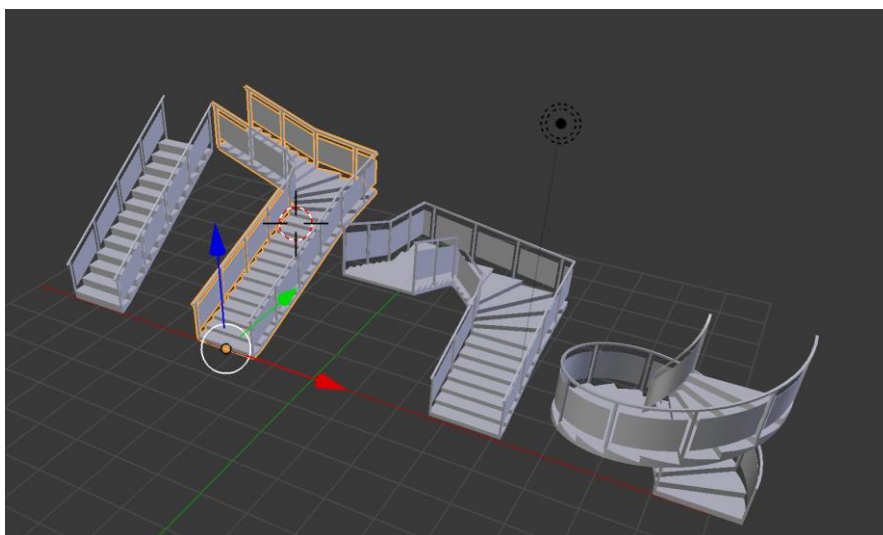


Figura 2.13. Tipurile de scări conținute în Add-on-ul Mesh: Archipack – Stair.

În Figura 2.14 putem regăsi un screenshot din aplicația realizată în Blender.

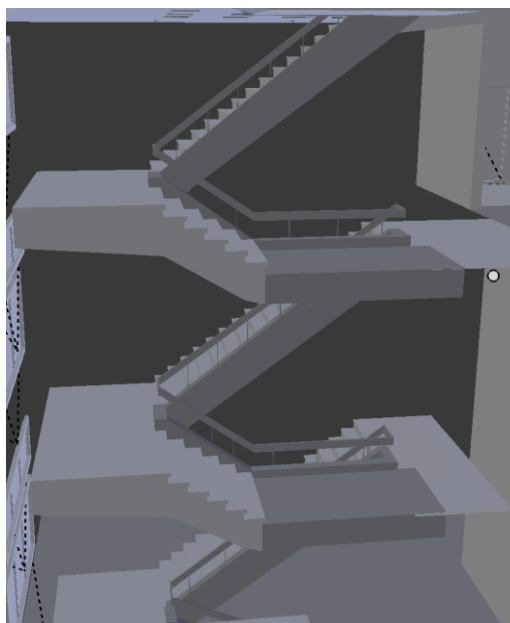


Figura 2.14. Screenshot al scărilor realizat din aplicație

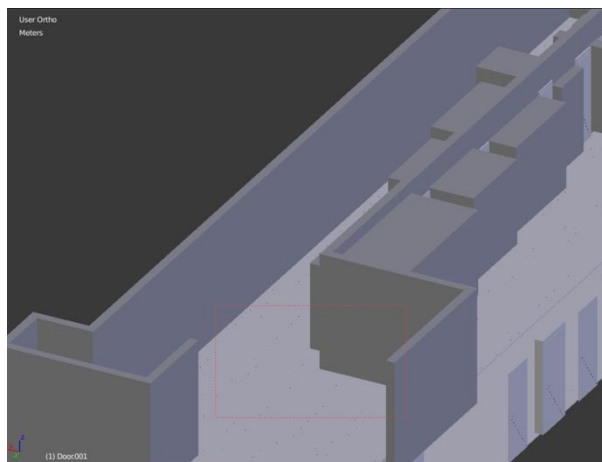
Construirea Corpului

Pentru realizarea corpului am modelat pereții pe baza măsurătorilor luate cu ajutorul telemetrului, conform figurii 2.2, și am plasat ușile în corp cu ajutorul unor cuburi ca și distanțiere între uși, conform figurii 2.15.

Realizarea pereților se poate face în două moduri:

În primul mod se poate folosi un cub dimensionat după măsurătorile realizate cu telemetrul.

Al doilea mod îl reprezintă utilizarea Add-on-ului Mesh: Archipack – Wall după cum putem vedea în figura 2.16. În acest mod putem genera în continuare pereți din peretele inițial, putându-i edita la ce unghi dorim noi să fie.



Figură 2.15. Distanțierea ușilor.

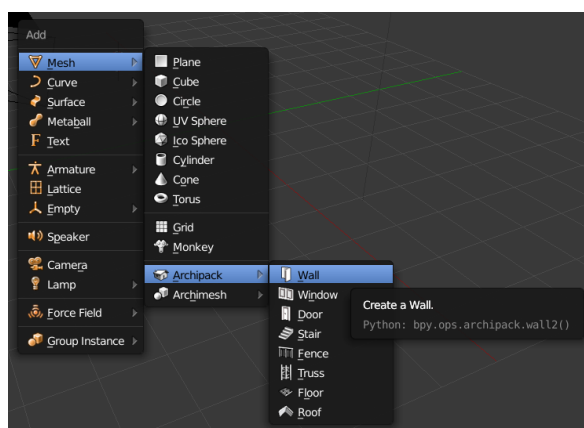


Figura 2.16. Add-on-ul utilizat pentru crearea pereților

Realizarea și plasarea ușilor în corpurile facultății

Primul pas a fost să plasăm ușa în perete după măsurătorile luate cu telemetrul, iar al doilea pas a constat în eliminarea porțiunii din perete care se suprapunea cu ușa conform figurii 2.17. Ultimul pas a constat în realizarea deschiderii ușilor.

Pentru realizarea ușilor am utilizat un Add-on denumit Mesh: Archimesh – Door așa cum putem observa în figura 2.18. Acest Add-on ne pune la dispoziție mai multe tipuri de uși așa cum putem observa în figura 2.19.

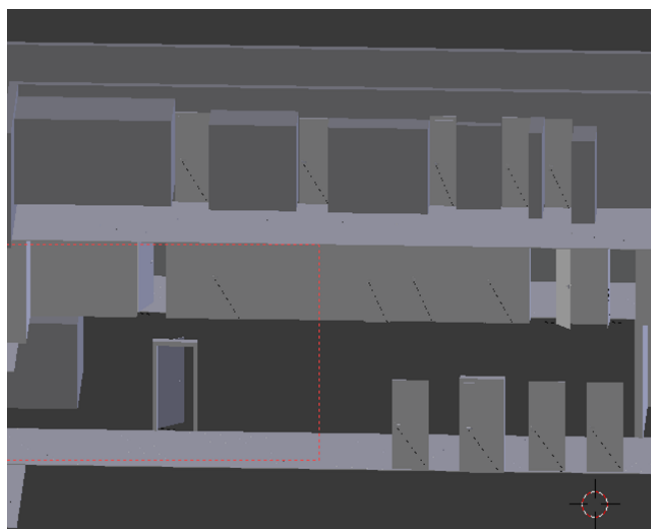


Figura 2.17. Plasarea ușilor în perete.

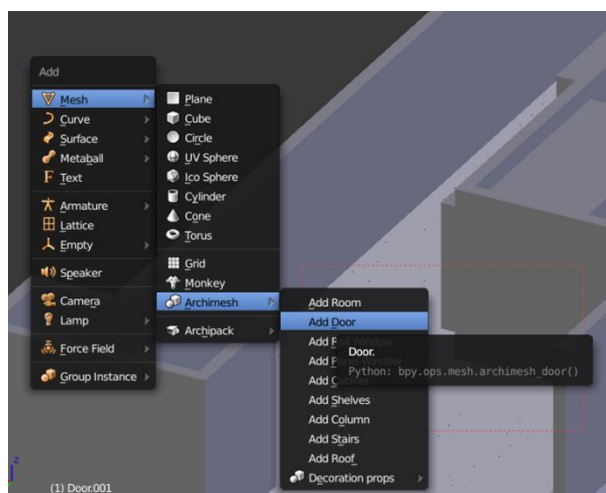


Figura 2.18. Add-on-ul utilizat pentru crearea ușilor.

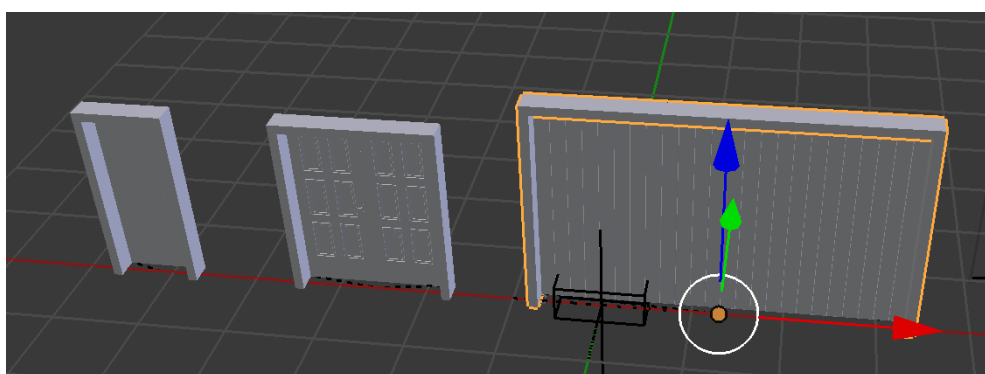


Figura 2.19. Tipurile de uși regăsite în Add-on-ul Mesh: Archimesh – Add Door.

După plasarea ușii în perete, a urmat tăierea peretelui în jurul ușii. Vom apăsa pe Tab de la tastatură pentru a intra în Edit Mode, apoi vom apăsa Z pentru a putea vedea exact unde tăiem ușa, iar ecranul va apărea exact ca în figură 2.20, vom apăsa A de la tastatură pentru a

deselecta toate fețele și selectam fața dorită pentru decupare, vom apăsa Ctrl+R de la tastatură pentru a putea tăia peretele după forma ușii. Vom realiza 3 tăieturi, una sus, una în stânga și una în dreapta ușii pentru a realiza deschiderea ușii fără să se vadă peretele ca în figura 2.21. După acest pas se va apăsa Ctrl+Tab și se va da click pe Face pentru a tăia bucata de perete care acoperă ușa. Se apăsă tasta C de la tastatură, click dreaptă de la mouse apoi X și Faces pentru a șterge peretele rămas peste ușă. Tab și Z pentru a ieși din Edit Mode. Pentru deschiderea ușii vom da click pe panoul ușii și apoi vom apăsa pe R de la tastatură și ușă se va deschide la în figura 2.22.

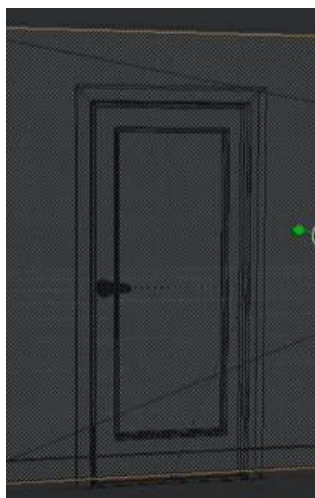


Figura 2.20. Screenshot al ecranului după ce am apăsă tastă Z.

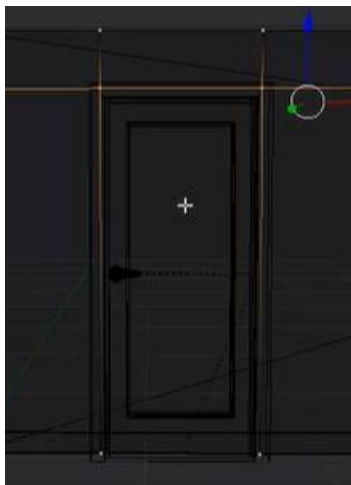


Figura 2.21. Plasarea tăieturilor ușii.

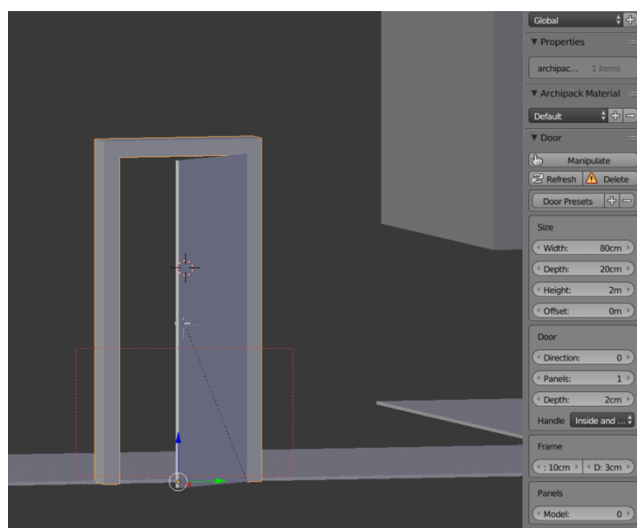


Figura 2.22. Realizarea deschiderii ușilor.

Realizarea încăperilor după o imagine de referință

Primul pas pentru realizarea încăperilor după o imagine de referință este să descărcăm Add-on-ul Blam folosind link-ul <https://github.com/stuffmatic/blam>, Add-on folosit pentru calibrarea camerei și proiecțiilor video, scris în Python care facilitează modelarea bazată pe fotografie.

Se va descărca ultima versiune folosind link-ul de mai sus, se va dezarhiva fișierul și se va localiza fișierul Add-on src/blam.py. Se va instala realizând click pe “Install from File” așa cum se poate vedea în figura 2.3.

Acum Add-on-urile vor fi instalate dar nu vor fi automat activate. Se va da click pe search pentru a căuta Add-on-ul instalat și vom activa Add-on-ul dând click pe căsuță din stânga denumirii Add-on-ului așa cum putem vedea în Figura 2.23.

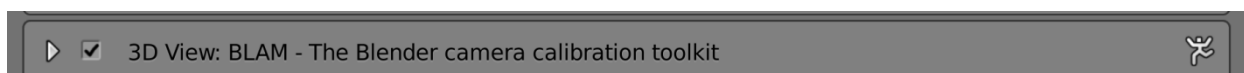


Figura 2.23. Add-on-ul Blam.

Blam este un Add-on specific pentru calibrarea camerei în Blender și este realizat pentru a facilita modelarea bazată pe o fotografie sau o imagine sintetică, realizată cu ajutorul unui program prestat cu o proiecție a camerei realizată din perspectivă.

Cu ajutorul Add-on-ului Blam putem să calculăm și să evaluăm orientarea și lungimea focală a camerei Blender bazată pe o fotografie astfel încât sursul 3D văzut prin cameră Blender-ului se potrivește cu cea din fotografie. Acest lucru face mult mă ușoară compunerea obiectelor 3D în imagini.

De asemenea, cu ajutorul Add-on-ului Blam putem să reconstruim geometria 3D cu fețe rectangulare din fotografii, proiecta imagini în rețea cu un singur click de mouse, lucru care este util de exemplu când realizăm cartografierea(maparea) proiecției camerei.

Calibrarea camerei statice

Calibrarea camerei statice este calculul lungimi focale și al orientării camerei pe baza unei singure fotografii sau unui cadru de film (notă: calibrarea camerei poate fi mișcată liber cât timp nu este rotită). Această funcție este localizată în Tools Panel a Movie Clip Editor-ului. Blam are nevoie de câteva intrări de utilizator(user input) pentru a realiza calibrarea. Această intrare este dată de mișcarea segmentului “grease pencil line” (Ctrl + D+ se trage mouse-ul ținând apăsat butonul din stânga), desenat în partea de sus a clipului video. Fiecare din aceste segmente de linie indică direcția data axelor de coordonate(x, y sau z). Toate segmentele de linie într-un layer al unui “grease pencil” trebuie să corespundă aceleiași axe. Această colecție de două sau mai multe segmente corespondente unei axe date va fi referită ca un pachet de linii, de segmente. De exemplu, segmentele de linie verde din figura 2.24 formează un pachet de linii.

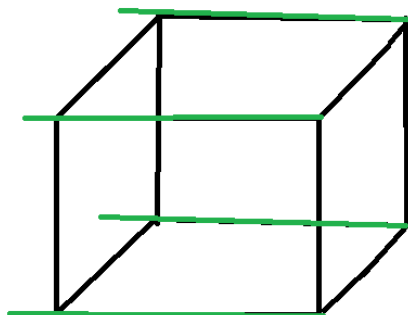


Figura 2.24. Exemplu de calibrare a camerei statice.

Calibrarea folosind două puncte orizont (two vanishing points)

Pentru a activa calibrarea folosit două puncte de orizont, se selectează această opțiune în meniul “Method” în panoul “Static Camera Calibration”, după cum se poate vedea în figura 2.25. Se selectează “Image midpoint” ca un centru optic (celelalte două opțiuni sunt descrise mai jos).

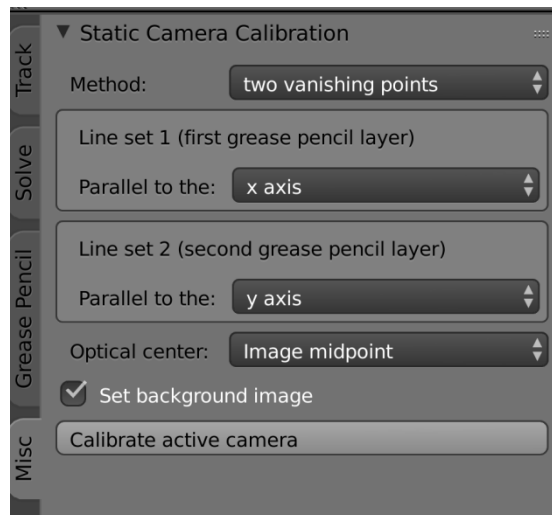


Figura 2.25. Calibrarea folosind două puncte de orizont.

Acasta metodă de calibrare calculează lungimea focală și orientarea camerei active bazate pe două pachete de linii de segment (un pachet pentru fiecare layer al grase pencil-ului) corespunzând direcțiilor perpendiculare ale axelor după cum putem observa în figura 2.26.

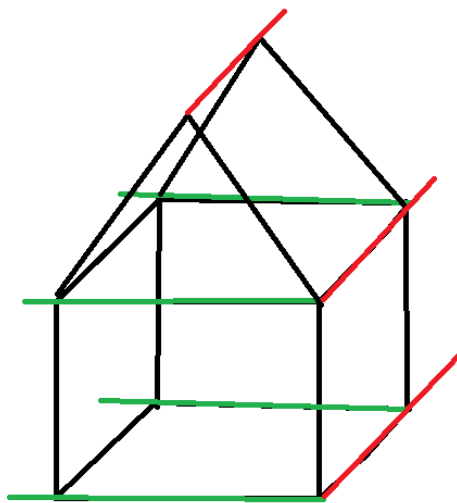


Figura 2.26. Calibrarea camerei folosind două puncte de segment.

Odată ce pachetele de linii de segment au fost desenate, putem alege care pachet ar trebui să corespundă unei axe folosind cele două căsuțe “Parallel to the”, care se regăsesc în figura 2.25. Pentru a realiza calibrarea, trebuie să fim siguri că există o cameră activă și apoi să apăsăm butonul “Calibrate active camera”, regăsit în figura 2.25. Dacă căsuța “Set background image” este bifată, clipul video va fi setat ca background pentru vizualizarea

camerei 3D când se realizează calibrarea. În figura 2.27 se regăsește podeaua cu pătratele din Blender văzută cu calibrarea camerei folosind două pachete de linii de segment.

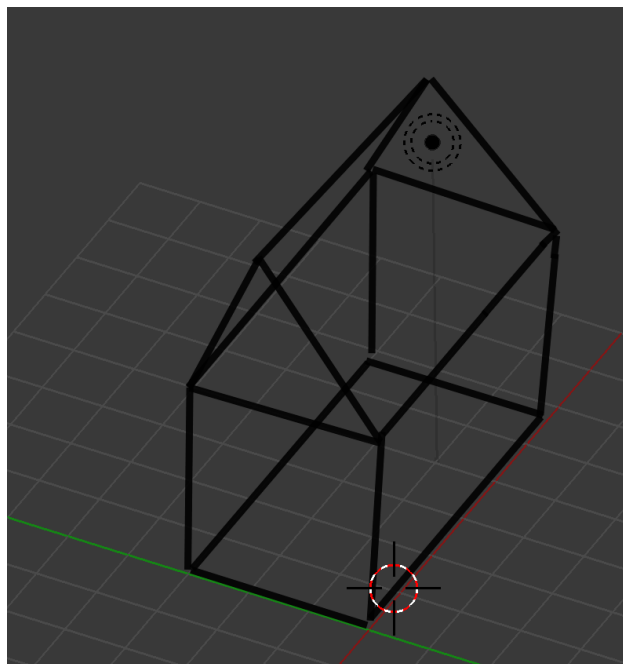


Figura 2.27. Calibrarea camerei în Blender folosind două pachete de linii de segment.

Definirea centrului optic

Centrul optic sau punctul principal este acel loc unde axele optice intersectează planul imaginii, de exemplu locul care trece prin mijlocul lentilei și întâlnește imaginea. În cele mai multe cazuri, centrul optic este în mijlocul imaginii, dar de asemenea există și cazuri în care centrul optic poate să nu fie în mijlocul imaginii dacă imaginea este tăiată asimetric de exemplu. Acest lucru poate fi greu de văzut cu ochiul liber.

Dacă calibrarea folosind punctul de mijloc ca centru optic nu dă rezultate precise, există șanse ca punctul optic să fie în alt loc. Blam suportă, în acest caz, două metode de definire a locației centrului optic: fie locația poate fi introdusă manual sau calculată pe baza unui pachet de segmente cu o a treia linie.

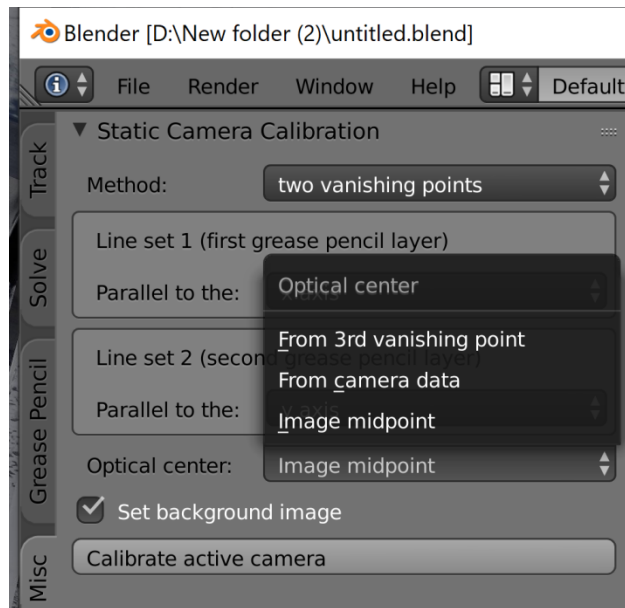


Figura 2.28. Tipurile de centru optic.

Pentru a folosi o locație a centrului optic cunoscută, se selectează “From camera data” din meniul “Optical center”, fapt reprezentat în figura 2.28, se setează locația folosind panoul de date al camerei din panoul de proprietăți al editorului de clipuri video și se apăsă “Calibrate active camera”.

Pentru a calcula centrul optic, prima data se selectează “From 3rd vanishing point” din meniul “Optical center”, apoi se desenează un pachet de linie de segment corespunzătoare celei de-a treia axă principală în layer-ul al treilea al “grease pencil” și se va da click pe “Calibrate active camera”. În figura 2.29 este reprezentat și al treilea segment de linie.

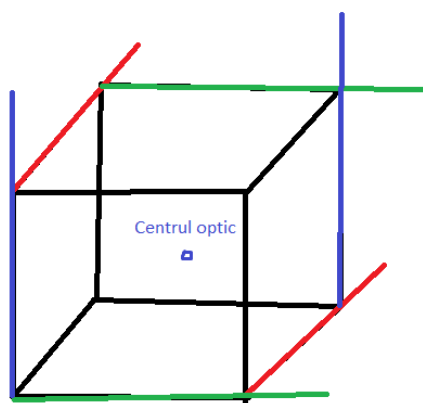


Figura 2.29. A treia linie de segment și centrul optic al imaginii.

Imagini de intrare problematice

Blam nu va funcționa pentru toate tipurile de imagini. Mai exact, o imagine de intrare este problematică dacă:

- Liniile de estompare de-a lungul oricărei axe sunt paralele sau aproape paralele
- Distorsiunea lentilei este mult prea severă
- Corecția din perspectivă sau alte tipuri de deformare au fost aplicate imaginii

Sfaturi de calibrare

Câteva lucruri de ținut minte pentru a îmbunătăți acuratețea calibrării:

- Folosind mai multe segmente de linii nu îmbunătățește neapărat acuratețea. Două linii de segment care se potrivesc unor linii de estompare sunt mult mai bune decât foarte multe segmente care nu sunt necesare. De asemenea, adăugând segmente de o acuratețe mai mică unei colecții de segmente de acuratețe va avea un impact negativ.

- Micșorând opacitatea și grosimea dărelor de creion de tip “grease pencil” face mult mai ușor să vedem dacă aceste dăre se aliniază corect.

- Dărele de “grease pencil” pot fi desenate cu acuratețe de sub-pixel dacă se mărește destul de mult imaginea

- Dacă imaginea noastră prezintă distorsiuni ale lentilelor vom trebuie să o dedistorsionăm (undistort) pentru a o putea utiliza cu Blam sau să acceptăm o acuratețe mai scăzută datorită faptului că liniile care trebuiau să fie drepte sunt curbe datorită distorsionării lentilelor

- Cu cât segmentele de linie sunt mai puțin paralele, rezultatele calibrării vor fi mult mai precise. Dacă putem face o alegere, trebuie să facem diferența în pantele segmentelor de linie cât mari posibile.

După descărcarea și activarea Add-on-ului Blam, ne vom duce la cubul din partea dreaptă și vom da click pe Movie Clip Editor pentru a putea realiza calibrarea, pas reprezentat în figura 2.30. După care vom da click pe Open în partea de jos pentru a deschide fotografia aleasă, o încărcăm, și va apărea pe ecran conform Figurii 2.31. Imaginea trebuie să fie de tip arhitectural pentru a realiza calibrarea. Vom alege din meniul din dreapta Grease Pencil pentru a putea realiza liniile de calibrare, după care vom alege din meniul din stânga de la “Static Camera Calibration”, metoda de calibrare, în cazul nostru vom folosi metoda “Two

vanishing points” prezentă în figura 2.28 deoarece, după cum putem vedea în figura 2.31, există două axe după care putem dezvolta fotografia, X și Y, liniile trasate fiind notate cu roșu și negru. Pentru trasarea acestor linii vom da click la “Grease Pencil Layers” pe New, apoi pe New Layer, conform figuri 2.32. Vom alege culoarea roșu pentru axa Y, conform figurii 2.33 și culoarea negru pentru axa X. După ce am ales culoarea pentru trasarea unei linii, vom apăsa pe Ctrl+D de la tastatură și apoi click dreapta pe mouse pentru a pune un punct pe ecran iar după vom da click stânga pe imagine în locul în care dorim să se formeze linia exact ca în figura 2.31. După ce se trasează prima linie, trebuie să trasăm a doua linie, tot de culoare roșie, care trebuie să fie la un unghi diferit de prima linie, dar pe aceeași axă. După ce vom trasa cele două linii, vom da click pe New layer pentru a forma liniile de culoare neagră, pe axă X și vom urma aceiași pași de mai sus. Înainte de a converti această imagine la o calibrare a camerei, va trebui să setam axele de coordonate ca în figura 2.28. Prima linie va fi setată pe axa X, iar a doua linie trasată cu creionul va fi setată la Y.

Următorul pas este să mai tragem un ecran care fie setat la 3D View pentru ne permite să modelăm imaginea și astfel să construim camera așa cum se poate vedea în figura 2.34, dar întâi, se click pe camera din figura 2.35 apoi pe “Calibrate active camera” din 2.28. Vom crea un plan după care vom începe să construim, însă acest plan va fi mult mare decât trebuie și va fi dimensionat. Iar acum putem construi toate obiectele din cameră, obiecte care pot fi vizualizate în figura 2.34. În figura 2.36 se află vizualizarea 3D a figurii 2.34.

În figura 2.37 se află reprezentat după o imagine de referință biroul domnului prof. dr. ing. Gabriel Rădulescu

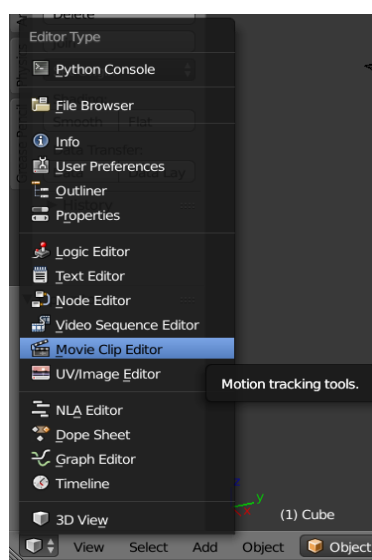


Figura 2.30. Deschiderea Movie clip editorului.

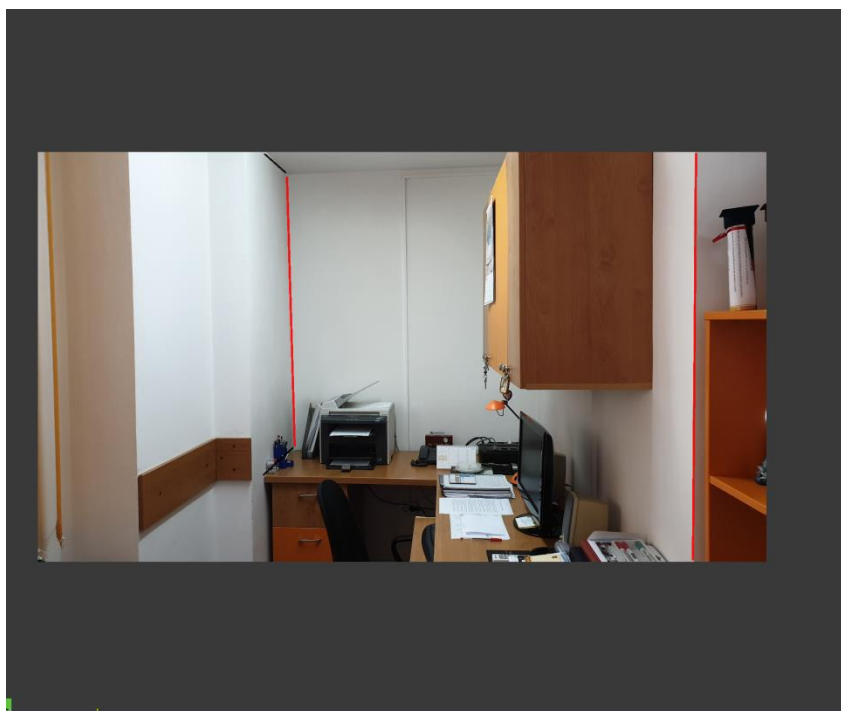


Figura 2.31. Încărcarea imaginii după care se va realiza camera.

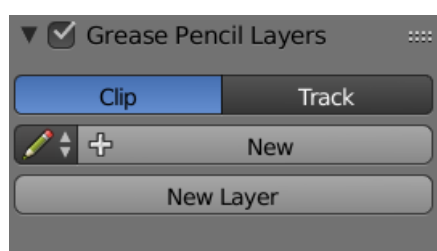


Figura 2.32. Grease Pencil Layers.

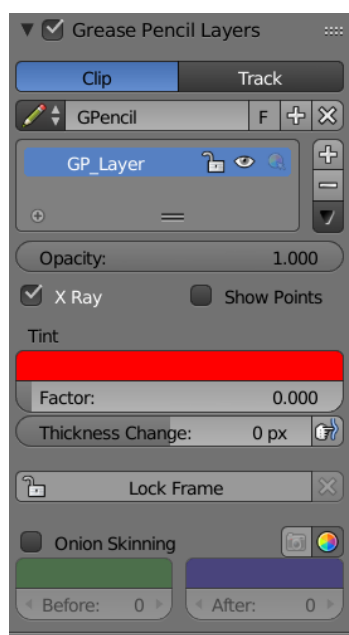


Figura 2.33. Alegerea culorii pentru trasarea liniilor.

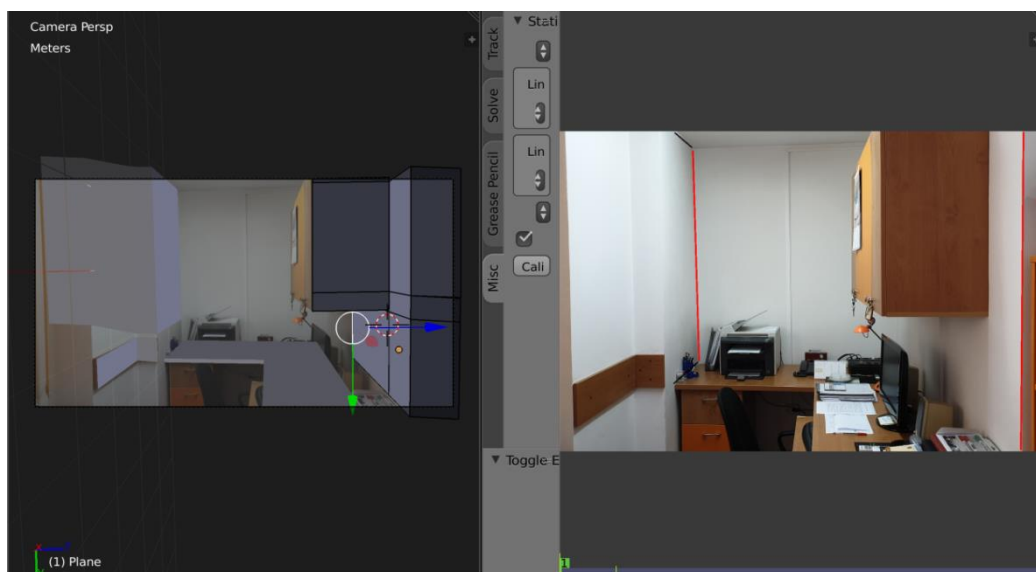


Figura 2.34. Camera calibrată.

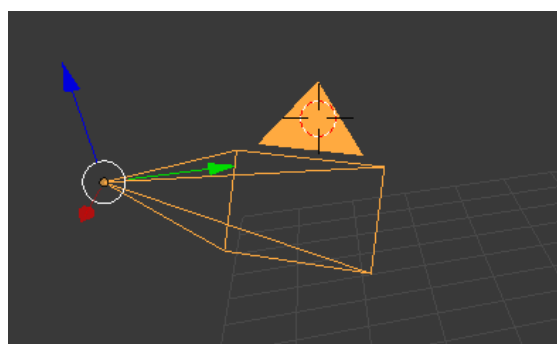


Figura 2.35. Camera specifică 3D View

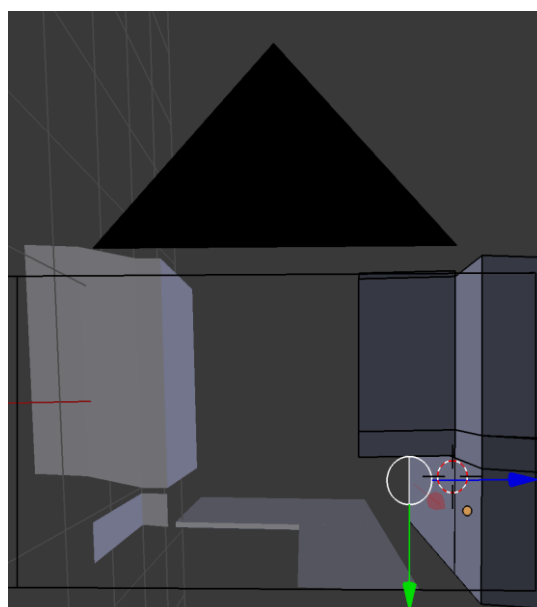


Figura 2.36. Reprezentarea Figurii 2.33 în 3D View.



Figura 2.37. Reprezentarea camerei domnului Prof. Dr. Ing. Gabriel Rădulescu.

Pasul 4. Texturarea elementelor din corpul B în Blender

Se deschide obiectul pe care dorim să îl texturăm. Se va seta “Blender Render” din meniul din bara de sus pentru o mai bună texturare ca în Figura 2.38. Se va da click pe Tab pentru a intra în modul de editare.

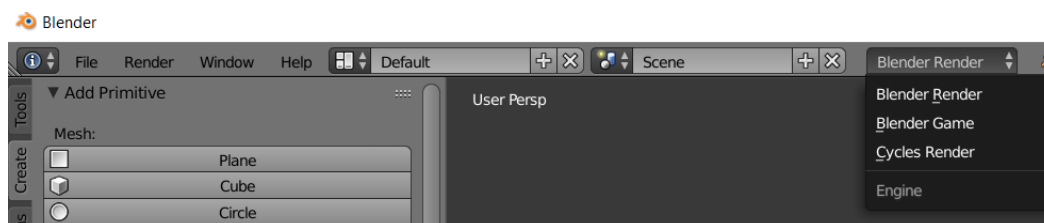


Figura 2.38. Blender Render.

Se va da click pe tasta U pentru a mapa obiectul 3D în 2 dimensiuni (unwrap) și se va deschide o fereastră nouă, ca în figura 2.39. Maparea UV a obiectului poate fi de mai multe tipuri în funcție de obiect: cubică, cilindrică, sferică etc.

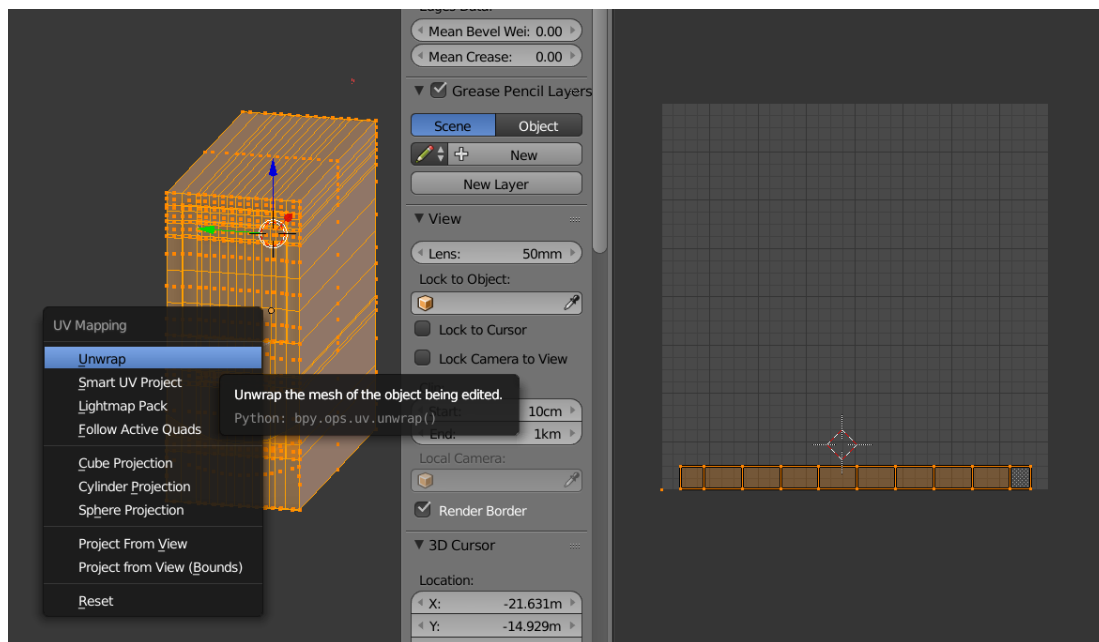


Figura 2.39. Maparea obiectului.

Se va selecta UV/Image editor pentru a vizualiza și edita imagini, pas reprezentat în figura 2.40.

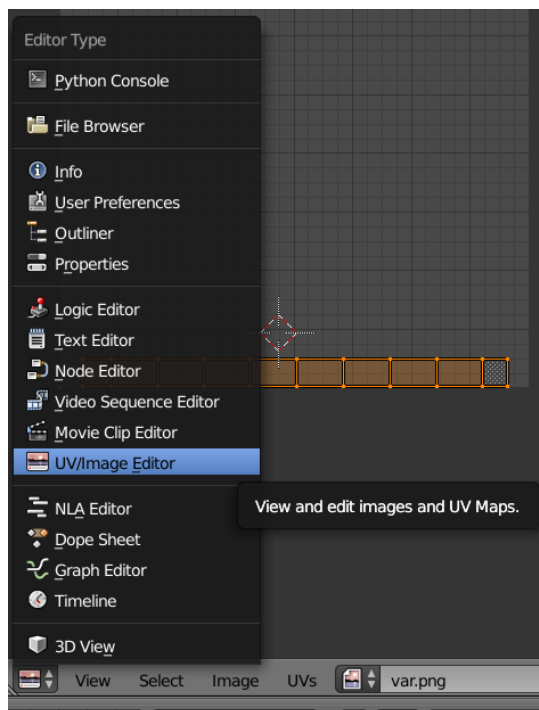


Figura 2.40. UV/Image Editor.

Pasul următor este căutăm și să descărcăm de pe internet o textură pe care dorim să o adăugăm obiectului. Îi vom da Open în fereastra cu UV/Image Editor așa cum putem vedea în

figura 2.41. Putem apăsa S pentru a scala imaginea, apoi G și X pentru a mișca imaginea pe axa X și apoi click stânga pentru a confirma poziția imaginii.

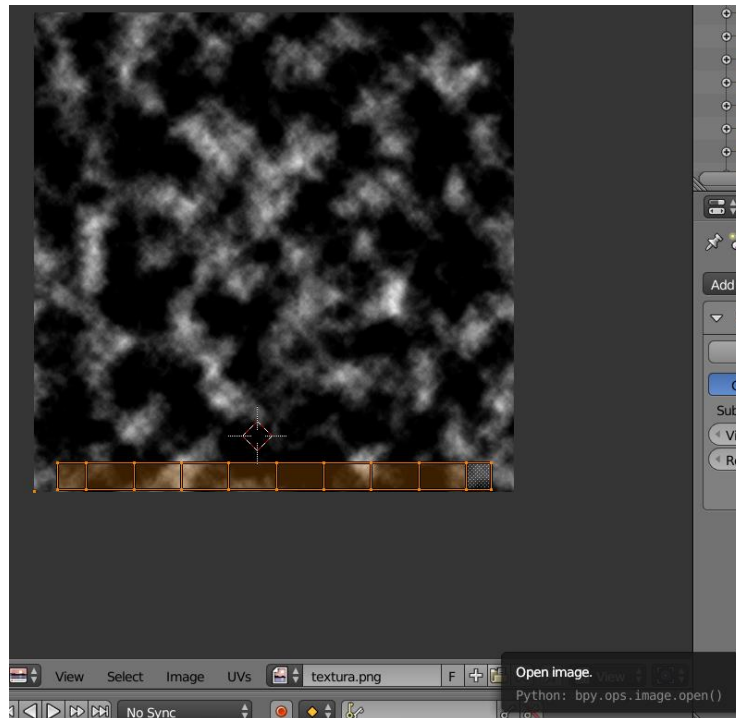


Figura 2.41. Deschiderea imaginii.

Se va da click pe Material din meniul din dreapta, apoi pe New pentru a adăuga un material nou, se va da click apoi pe Texture pentru a adăuga textură obiectului, se va da click pe New, unde se va încărca imaginea descărcată de pe internet ca în figura 2.42.

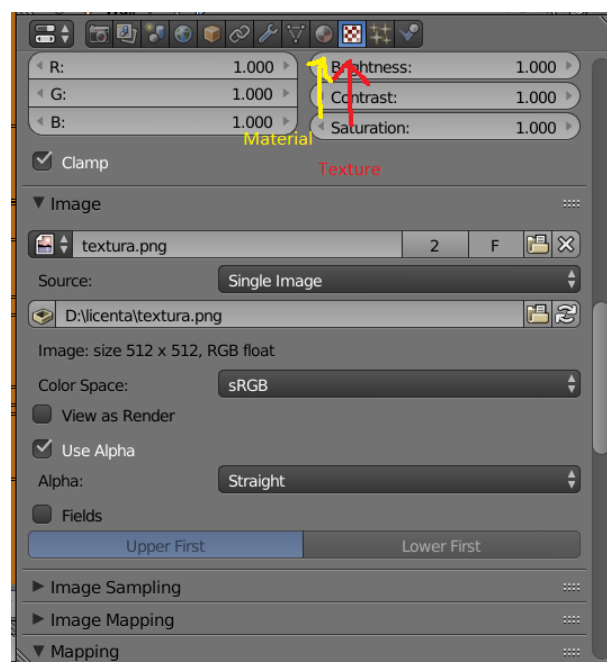


Figura 2.42. Adăugarea materialului și a texturii.

Se va da click pe Texture și apoi pe Object Mode pentru a vedea obiectul texturat, pas realizat în Figura 2.43.

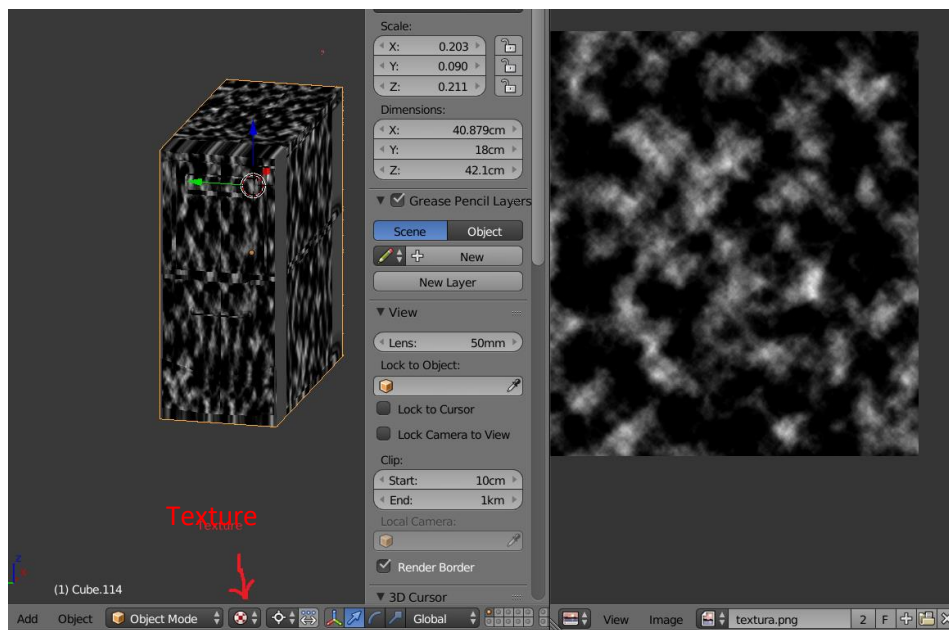


Figura 2.43. Obiectul texturat.

2.4. Importare model în motorul grafic Unity

Pasul 5. Pregătirea motorului grafic Unity pentru Realitatea Virtuală

Unity este folosit pentru a crea jumătate din jocurile realizate în lume. Unity este alegerea perfectă pentru studiourile mici și persoanele care doresc să realizeze jocuri sau structuri de arhitectură. Unity are o bază foarte mare de utilizatori activi, iar această comunitate foarte mare permite utilizatorilor noi să găsească și să primească răspunsuri la toate întrebările legate de acest mediu.

Unity oferă o balanță între caracteristici și funcționalitate. Versiunea gratuită Unity permite utilizatorilor să experimenteze, învețe, dezvolte și să vândă lucrările realizate.

Am descărcat și instalat Unity, versiunea 2018.3.0f2. Vom da click pe New după care vom scrie la “Project name” numele proiectului, îl vom denumi “DaydreamTest” . La “Template” vom da click pe 3D, vom seta locația unde vom dori să se salveze aplicația, după care vom da numele organizației, și vom da click pe “Create this project”.

Vom da click pe “File”, apoi pe “Build Settings” și mai apoi pe “Android” ca în figura 2.44. Vom da click pe “Switch Platform”. Vom da click pe “Player Settings” după care ne vom duce la “XR Settings”. Ne vom asigura de faptul că “Virtual Reality Supported” este

activat, se va da click pe semnul de plus după care vom adăuga proiectul “DaydreamTest” la acesta. Se va copia numele proiectului, în cazul nostru “DaydreamTest”. Se va da click pe “Other Settings” – “Identification” – “Package Name” și vom la paste la numele proiectului în forma următoare “com.DaydreamTest”. Acest pas este important dacă se dorește încărcarea proiectului pe App Store.

Se va da click pe “Edit” – “Project Settings” – “Editor” care se va deschide în partea dreaptă a ecranului iar la “Device” se va selecta “Any Android Device” ca în figura 2.45. După realizarea pașilor de mai înainte, Unity este pregătit pentru lucru.

Următorul lucru care trebuie realizat este să descărcăm pachetele pentru Google și Android pentru a putea seta mediul.

Vom descărca Android SDK. Pentru aceasta se va da click pe “Edit” – “Preferences” – “External Tools”, pas reprezentat în figura 2.46, iar în partea de jos se va găsi rubrica pentru Android cu SDK(Software Development Kit – este un software de dezvoltare care permite crearea de aplicații pentru un anumit pachet software, consolă pentru jocuri video, platformă hardware, sistem de operare sau orice altă platformă de dezvoltare) și NDK(Native Development Kit – este un set de instrumente care permite utilizatorului să folosească cod scris în C sau C++ cu Android și care oferă librării pe care utilizatorul le poate folosi pentru a manageria accesul la componentele dispozitivelor cum ar fi senzori) care inițial sunt goale.

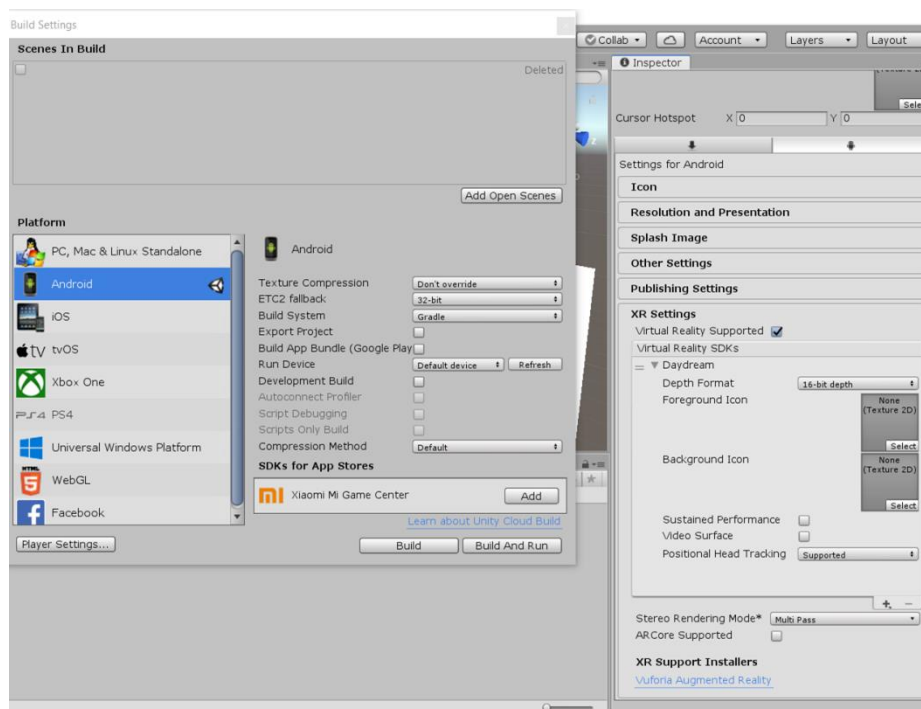


Figura 2.44. Build Settings.

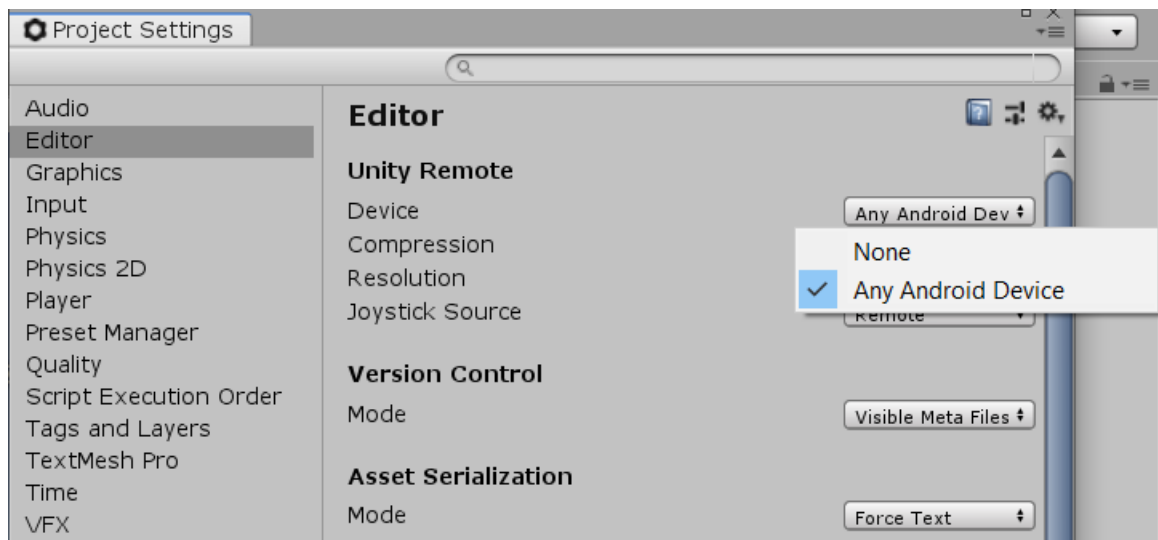


Figura 2.45. Project Settings.

Lângă SDK sunt două butoane, unul de tip Browse și unul de tip Download. Se va da click pe butonul de Download care ne va duce către link-ul <https://developer.android.com/studio/index.html/> care conține ultima versiune de SDK pentru dezvoltarea Android. Se va da click pe Download Android Studio și se va instala.

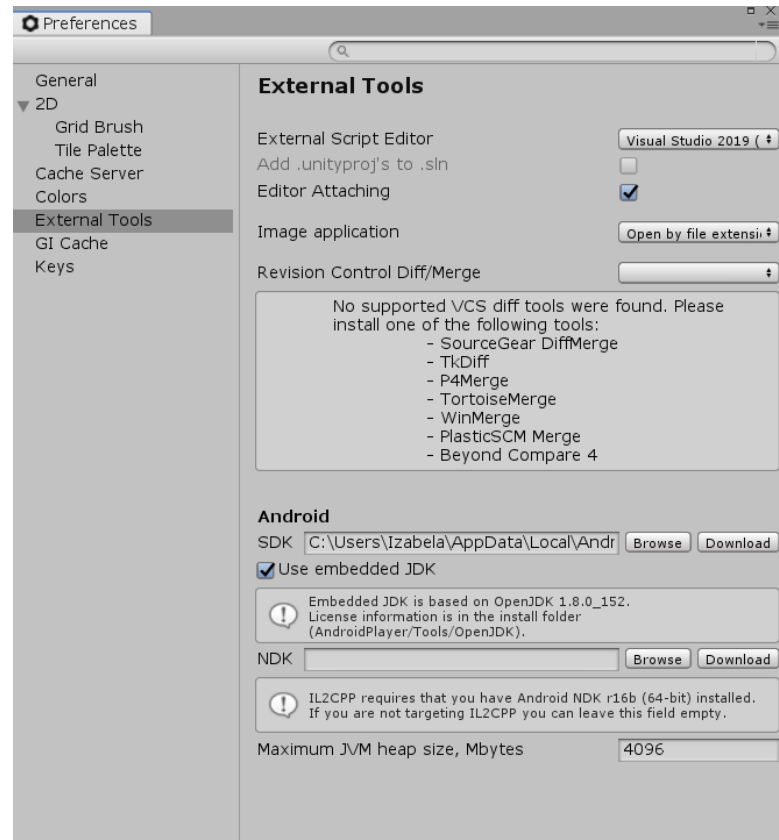


Figura 2.46. Unity Preferences.

Se va da click pe “Android Studio Setup” din descărcări și se vor urma pașii de instalare. După instalare se va da click pe “Edit” în Unity – “Preferences” – “External Tools”. Se va da click pe “Browse” în dreptul SDK ca în figura 2.46, iar în bara de căutare se va scrie calea locului unde a fost descărcat SDK-ul, în cazul nostru “C:\Users\Izabela\AppData\Local\Androidsdk”.

Se va bifa de asemenea și “Use embedded JDK”(Java Development Kit – este o componență cheie a platformei pentru construirea aplicațiilor Java).

Pasul următor este să descărcăm “Google Daydream SDK” folosind link-ul “<https://developers.google.com/vr/develop/unity/get-started-android>”. După ce îl vom descărca, îl vom importa în Unity dând click pe Assets și apoi click dreapta în căsuța Asset-urilor – “Import Package” – “Custom Package” pentru a încărca SDK-ul ca în figura 2.47. Se va deschide o fereastră în care selectăm fișierul din folder-ul în care a fost salvat, după care se va da Open. După ce se urmează acești pași, în “Assets” va apărea “GoogleVR”. Se va da click stânga pe “GoogleVR” și vor apărea mai multe foldere dintre care putem enumera “Demos”, “Scripts”, “Prefabs”.

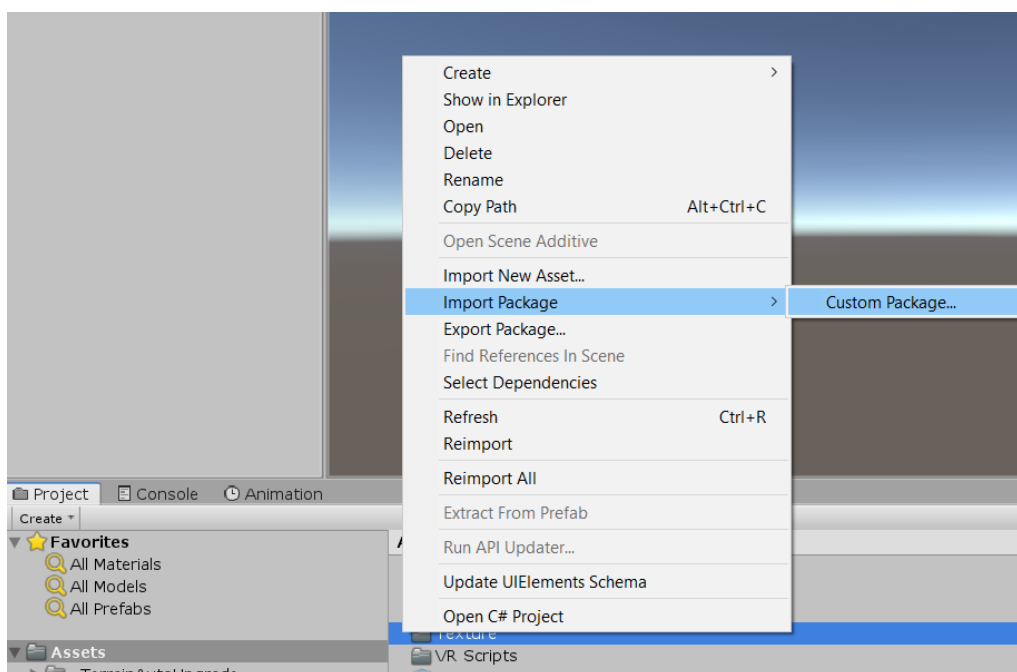


Figura 2.47. Importarea pachetului SDK.

Pasul 6. Importarea modelului obținut în Unity

Se va da click în Assets – Import New Assets, conform figurii 2.48 și se va importa fișierul realizat în Blender conform figurii 2.49.

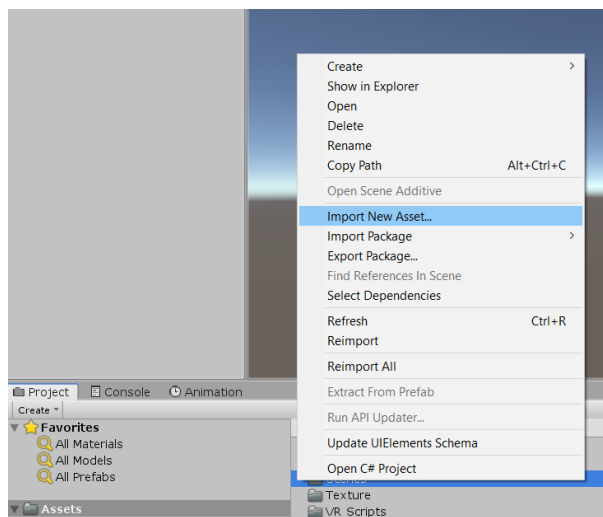


Figura 2.48. Importarea corpului realizat în Blender.

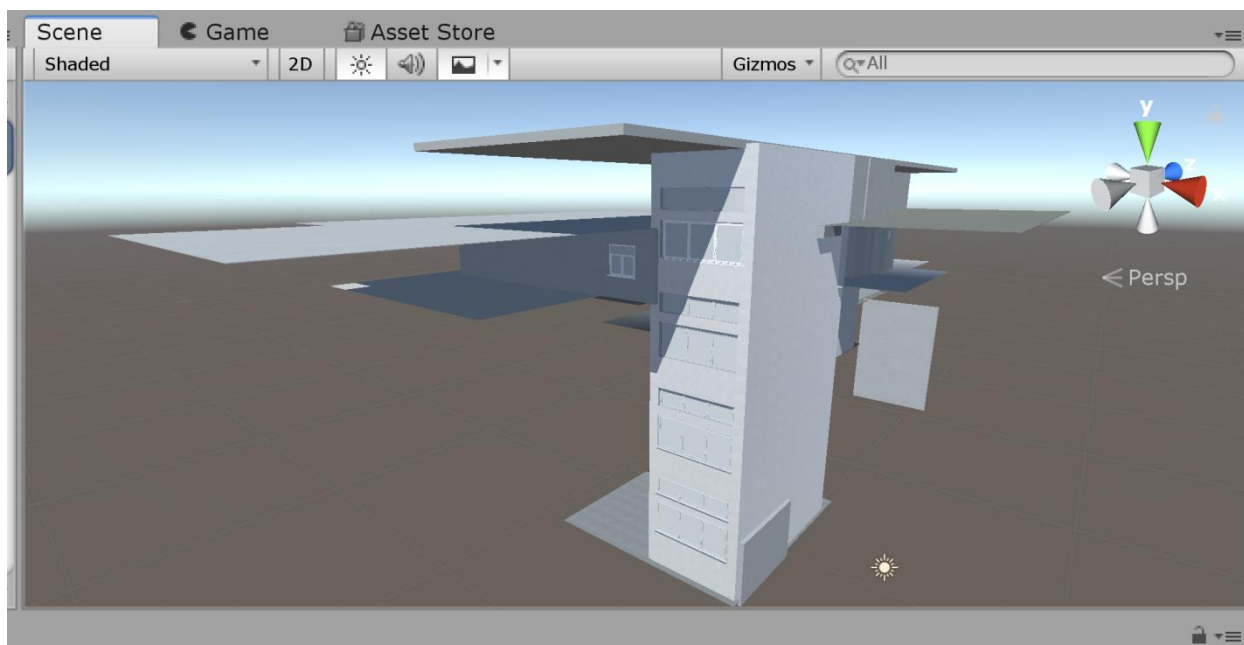


Figura 2.49. Corpul importat în Unity

Pasul 7. Codificarea elementelor construcției

Codificare ușilor - Pentru declanșarea animației am creat un script prin care efectul de deschidere se declanșează prin coliziunea măștii utilizatorului cu suprafața ușii.

Se deschide în Unity meniul “Window” – “Animation”, după care se va da click pe “Create Animation” pe care o vom denumi “DoorAnimation”. Pentru a crea animația avem nevoie de o singură proprietate, rotația, pentru realizarea senzației de deschidere a ușii. Se va apăsa pe butonul de “Record”. Din meniu “Animation” dăm click pe “Add Property”, după care selectăm meniul “Transform” din care alegem proprietatea “Rotation”, așa cum putem observa în figura 2.50. Automat se va crea un segment de 60 cadre. Cadrul 30 va fi marcat ca și Keyframe, unde se va seta rotația ușii pe axa Y la 90 de grade. După care se verifică animația apăsând pe butonul Play. În acest moment se deschide și se închide ușa. După acest pas se oprește înregistrarea, apăsând din nou pe butonul de record.

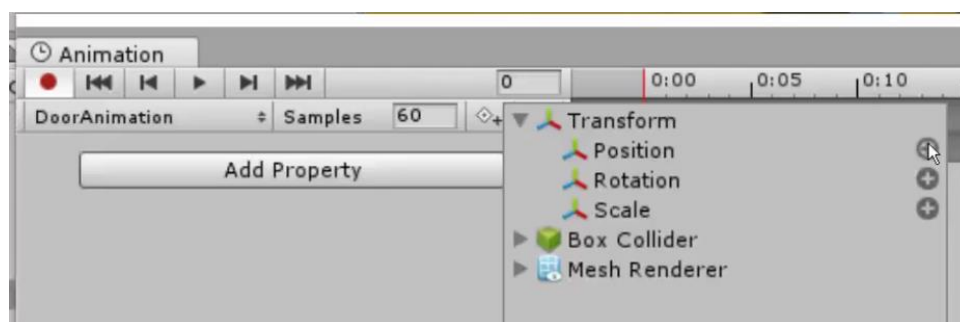


Figura 2.50. Proprietatea Rotation.

Vom da click pe meniul “Window” – “Animation” – “Animator”. Vom crea o nouă stare, dând click dreapta în spațiul cu stări, “Create State” – “Empty”. O vom denumi DoNothing(nu face nimic). Se va da click dreapta pe starea DoNothing care va fi setată ca stare de bază(Set as Layer Default State). De aici vom da click dreapta pe starea DoNothing și se va face o tranziție(Make Tranzition) către DoorAnimation, așa cum putem vedea în figura 2.51. Se va realiza de asemenea și o tranziție de la DoorAnimation către DoNothing. În acest moment avem nevoie de un “Trigger”(declanșator). Din meniul din stânga(Animator) se apăsă pe semnul plus din care se selectează Trigger, pe care îl vom denumi OpenDoor. Se va da click pe legătura de tranziție dintre stările DoNothing și Animation, iar în dreapta va apărea meniul “Inspector” și de aici se va adaugă condiția “OpenDoor”.

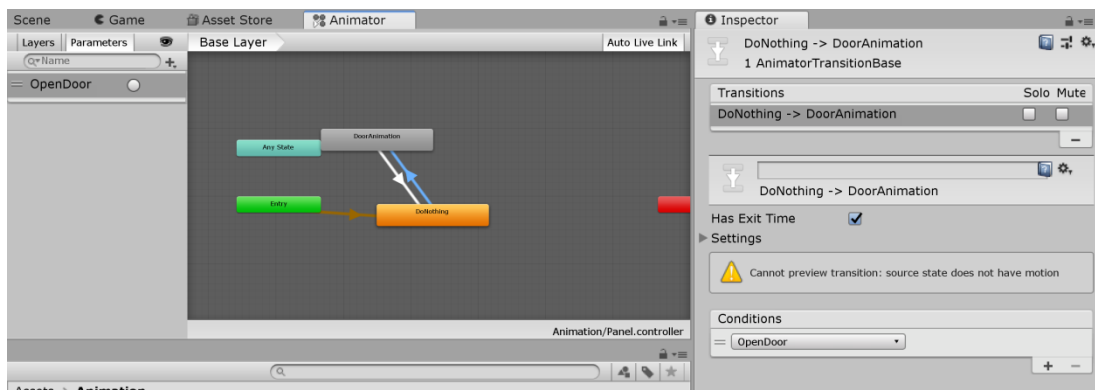


Figura 2.51. Realizarea tranzițiilor.

Pentru a declanșa animația avem nevoie de un script C#. Pentru aceasta se va da click dreapta în “Assets” – “Create” – “C# Script”. Pentru ca animația să funcționeze, trebuie să atașăm ca și componente obiectului ușă, componenta “Animator” – “Controller” iar prin metoda drag and drop, fișierul de animație creat anterior va fi pus la Controller; componenta Sphere Collider în care se va bifa căsuța Is Trigger, după cum se poate vedea în figura 2.52; componenta DoorScript în care este scris codul sursă pentru realizarea deschiderii ușii. La cadrul 30 al animației se va adăuga un eveniment(Event). Se selectează acest eveniment și îi vom seta ca funcție “pauseAnimationEvent()”, regăsit și în codul următor:

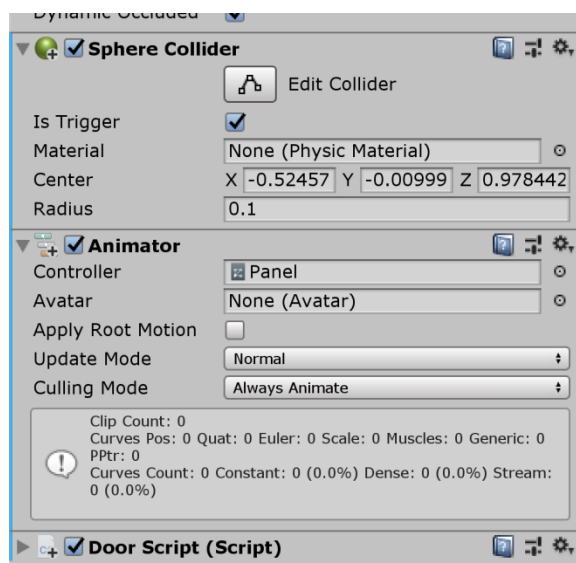


Figura 2.52. Sphere Collider.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class DoorScript : MonoBehaviour
{
    Animator anim; //variabila de tip animator
    // Use this for initialization
    void Start()
    {
        anim = GetComponent<Animator>(); //se acceseaza variabila
                                         //de tip animator
    }
    // Update is called once per frame
    void Update()
    {
    }
    void OnTriggerEnter(Collider other)
    {
        anim.SetTrigger("OpenDoor"); //declansarea animatiei
    }
    void OnTriggerExit(Collider other)
    {
        anim.enabled = true; //animatia este activata
    }
    void pauseAnimationEvent()
    {
        anim.enabled = false; //animatia este dezactivata
    }
}

```

Panoul și butonul de informații regăsite la fiecare sala realizată – Pentru a crea panoul de informație, se va da click dreapta în scena “licenta” și se va alege “UI”(User Interface) – “Canvas”(Canvas este zona în care toate elementele de tip UI ar trebui să se afle. Canvas este un GameObject cu o componentă Canvas și toate elementele de tip UI trebuie să fie de tip copil unui asemenea tip de Canvas. Zona Canvas este afișată ca un dreptunghi în aria de vizualizare. Canvas folosește obiectul EventSystem pentru a ajuta sistemul de mesagerie – Message System. Canvas are trei modalități de redare: Screen Space-Overlay, Screen Space – Camera, World Space. Dintre aceste trei tipuri am ales World Space pentru a se comporta Canvas ca orice alt obiect din scenă. Dimensiunea acestuia poate fi setată manual). Se va adăuga ca și copil obiectul” UI” – “Panel”. După acest pas se va seta Background-ul Panel-ului cu o imagine aleasă de noi. Obiectului Panel i s-a adăugat Obiectul UI Button. Pentru a scrie textul pe buton am folosit Add-on-ul “TextMeshPro”, descărcat și importat din Asset Store. După acest pas am adăugat încă un obiect de tip UI – “Image” căruia i-am setat fundalul(Background) cu o imagine aleasă de noi și am adăugat script-ul ShowHide. Pentru a scrie textul în imagine am folosit Add-on-ul TextMeshPro.

Butonului i-am adăugat script-ul “GazeInteraction” și trei evenimente declanșatoare.

Codul butonului:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;

public class GazeInteraction : MonoBehaviour
{
    public float gazeTime = 2f;
    private float timer;
    private bool gazedAt;

    // Use this for initialization
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        if (gazedAt)
        {
            timer += Time.deltaTime;
            if (timer >= gazeTime)
            {
                // execute pointerdown handler
                ExecuteEvents.Execute(gameObject,
                                        new PointerEventData(EventSystem.current),
                                        ExecuteEvents.pointerDownHandler);
                timer = 0f;
            }
        }
    }
    public void PointerEnter()
    {
        gazedAt = true;
        //Debug.Log("PointerEnter");
    }
    public void PointerExit()
    {
        gazedAt = false;
        //Debug.Log("PointerExit");
    }
    public void PointerDown()
    {
        //Debug.Log("PointerDown");
    }
}
```


Codul panoului:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ShowHide : MonoBehaviour
{
    public GameObject InfoObject;
    private bool Show = false;
    // Start is called before the first frame update
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
    }
    public void ShownHide()
    {
        if (!Show)
        {
            InfoObject.SetActive(true);
            Show = true;
        }
        else
        {
            InfoObject.SetActive(false);
            Show = false;
        }
    }
}
```

Pasul 8. Texturarea elementelor în Unity

Am creat materialul specific obiectelor dând click dreapta la meniul Assets așa cum am văzut anterior, apoi “Create” – “Material” conform figurii 2.53. În folder-ul Texture se va da click dreapta pe “Import New Asset”, conform Figurii 2.54, astfel se va crea textura pe care o vom aplica materialului creat anterior cu drag and drop în căsuța “Albedo” din secțiunea “Main Maps”. Aplicarea materialului obiectelor se va face tot cu drag and drop.

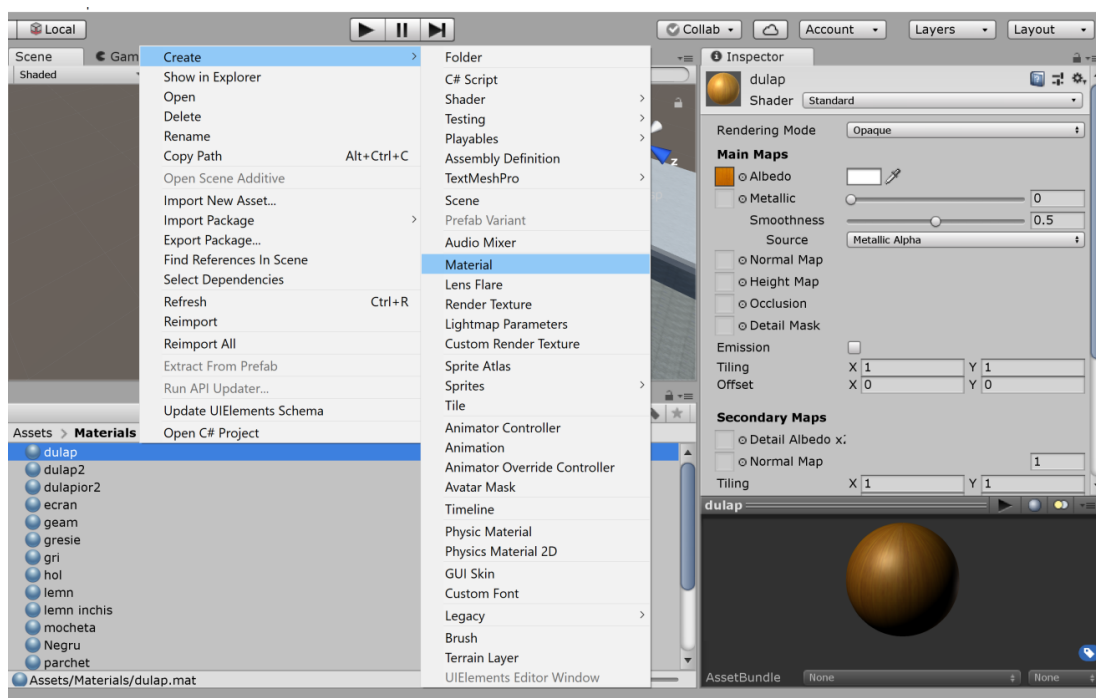


Figura 2.53. Crearea materialelor.

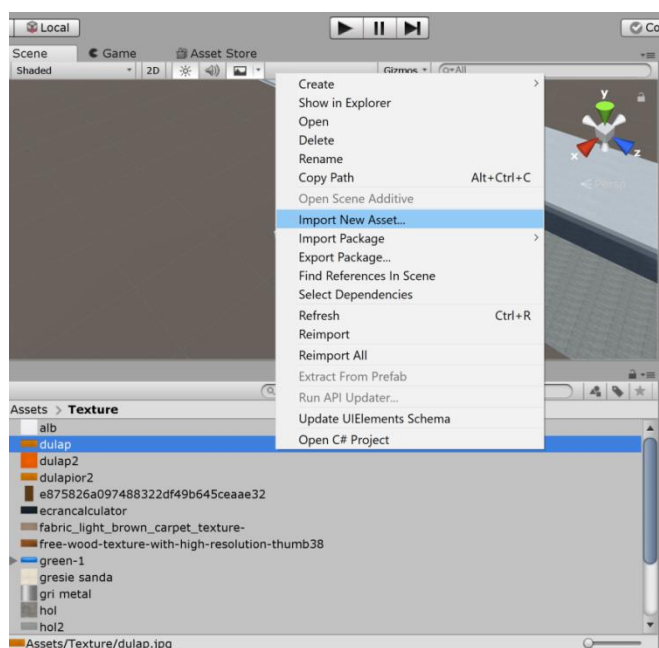


Figura 2.54. Importarea texturii.

Pasul 9. Realizarea și codificarea Waypoint-urilor în Unity

Un Waypoint este un punct intermediar, o linie/punct de parcurgere a unui traseu, un punct de stop sau un punct în care direcția de deplasare se schimbă, prima utilizare a acestui termen fiind introdusă în 1880. Cu alte cuvinte, un Waypoint se referă la coordonatele specifice unei poziții pe glob realizate pe calculator. Waypoint-urile sunt seturi de coordonate care identifică un punct în spațiul fizic.

Se va da click dreapta pe “Create Empty” în meniul din partea stânga denumit Licență, se va denumi Waypoints. Se va da click dreapta pe Waypoint-ul creat și va alege opțiunea “3D Object” – “Sphere”. Sfera o vom redenumi Waypoint. După care se va da click dreapta în “Assets”, se va crea un material nou așa cum am văzut în pasul anterior, denumit “WaypointMaterial” pe care îl vom atașa Waypoint-ului.

Pentru a ne putem mișca în spațiu când privim Waypoint-ul, trebuie să îi punem un tag care se află în partea dreaptă la “Inspector” – “Tag” – “Add Tag”, se va da click pe semnul de plus, și acest tag se va denumi Waypoint. Se va selecta Waypoint-ul din meniul din partea stânga, iar la Tag se va adăuga Tag-ul denumit de noi Waypoint. Numele de Waypoint din meniul din stânga trebui să fie același cu cel al Tag-ului, conform figurii 2.55. Am dat click dreapta în meniul din stânga și am creat un “Empty Game Object” folosind “Create Empty” denumit Player, ca în figura 2.56.

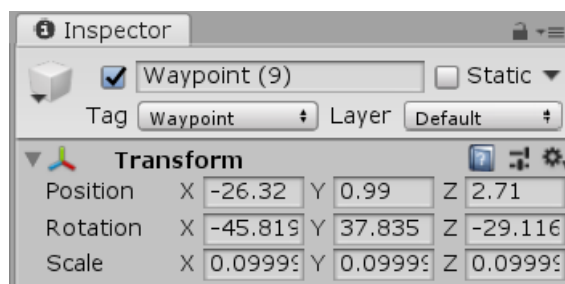


Figura 2.55. Tag-ul Waypoint.

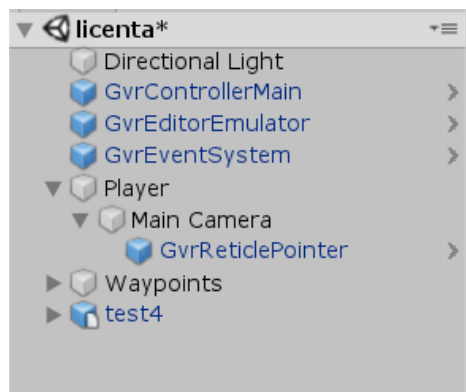


Figura 2.56. Player – Empty Game Object.

De asemenea, se va da click în meniul “Assets” – “Create” – “C# Script”, denumit Player. Se va da dublu click pe scriptul denumit Player se va deschide Visual Studio 2017, unde se va scrie următorul cod:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Player : MonoBehaviour
{
    public Vector3 targetPosition;

    public float timer;
    public float timeSpentGazingAt = 2f;

    bool isMovingToTarget = false;

    // Start is called before the first frame update
    void Start()
    {
        targetPosition = transform.parent.position;
    }

    // Update is called once per frame
    void Update()
    {
        RaycastHit hit;

        //verifică dacă raycast lovește o țintă
        if (Physics.Raycast(transform.position, transform.forward, out
hit))
        {
            //verifică dacă ținta este un tag
            if (hit.transform.tag == "Waypoint" && isMovingToTarget ==
false)
            {
                Debug.Log("WaypointHit");

                //preia poziția țintei
                targetPosition = new Vector3(hit.transform.position.x,
hit.transform.position.y, hit.transform.position.z);

                //pornește ceasul
```

```

        timer += Time.deltaTime;

        //verifica daca ceasul a atins 2f
        if (timer > timeSpentGazingAt)
        {
            Debug.Log("TimeToMove");
            //spune metodei move ca player-ul are voie să se
deplaseze
            isMovingToTarget = true;
        }
    }

    //daca raycast nu lovește nimic
    else if (hit.collider == null)
    {

        //porneste numărătoarea inversă
        Debug.Log("Not Hitting");
        while (timer > 0)
        {
            timer -= Time.deltaTime;
        }

    }

    //apelează metoda
    movePlayer();
}

void movePlayer()
{
    //verifică dacă player-ul are voie să se miște
    if (isMovingToTarget == true)
    {
        transform.parent.position =
Vector3.Lerp(transform.parent.position, targetPosition, Time.deltaTime);
    }

    //verifică dacă player-ul a ajuns la țintă
    if (Vector3.Distance(transform.parent.position, targetPosition) <
0.1)
    {
        Debug.Log("TimeToStop");
        //nu se mai deplasează
        isMovingToTarget = false;
        //resetează ceas
        timer = 0;
    }
}
}

```

După ce este scris codul în Visual Studio, ne vom întoarce în Unity, vom da click pe codul scris în C# și îl vom duce la Player în meniul din stânga exact ca în figura 2.57. În acest moment dacă vom da Play la ceea ce am realizat până acum player-ul se va duce către Waypoint dacă acesta fixează Waypoint-ul pentru doua secunde.

Pentru a adăuga mai multe Waypoint-uri tot ceea ce trebuie să facem este să dăm click dreapta pe “Waypoint” – “Duplicate”, ceea ce putem observa în figura 2.58.

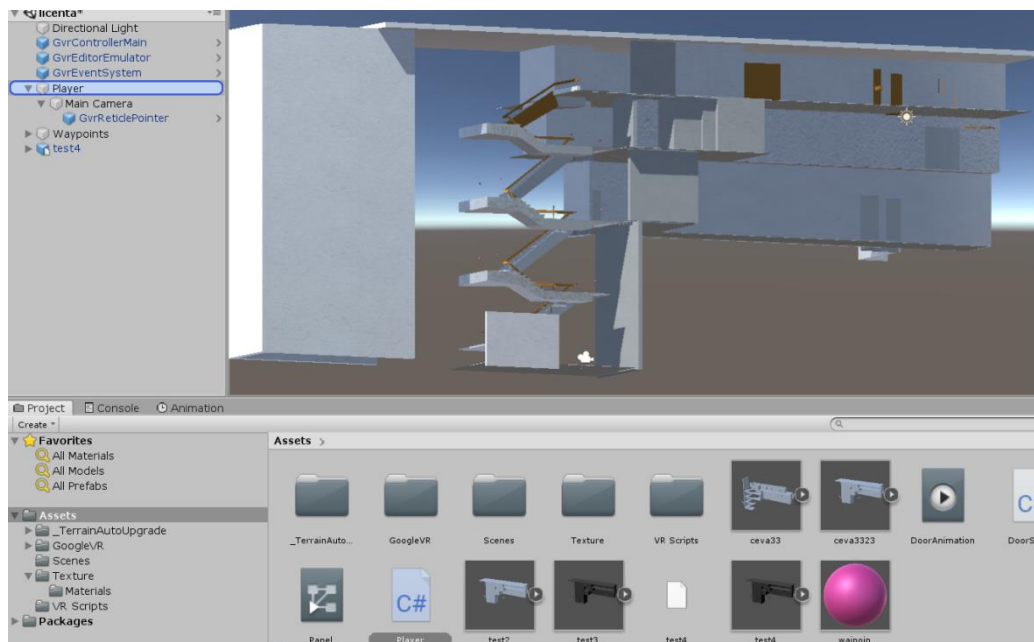


Figura 2.57. Scriptul Player introdus în Empty Game Object – Player.

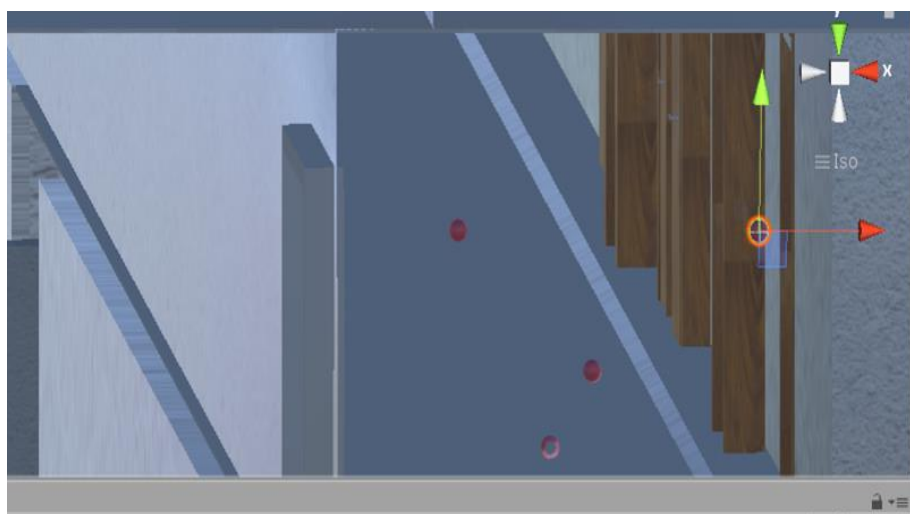


Figura 2.58. Waypoints.

2.5. Importare și configurare aplicație pe platforma mobilă

Pasul 10. Importarea pe telefon a aplicației și realizarea conexinii cu ochelarii Google Daydream

Pentru utilizarea aplicației sunt necesare următoarele echipamente, prezentate în figura 2.59:

-O pereche de ochelari Google Daydream

-Un telefon android care este compatibil cu ochelarii Google Daydream



Figura 2.59. Ochelarii Daydream și telefonul compatibil.

Se exportă aplicația pe telefon printr-un cablul USB, după care se pune telefonul în suportul ochelarilor, se apasă butonul de pornire de pe controller, iar acum aplicația este funcțională.

Capitolul 3. Testarea aplicației

Pentru testarea aplicației pe telefon am conectat telefonul la calculator și am exportat aplicația pe telefon sub formă de fișier APK din motorul grafic Unity, selectând din meniul “File”, opțiunea “Build and Run”.

Pentru a putea fi instalată și testată aplicația este necesar să activăm modul Dezvoltator pe telefon, iar pentru aceasta trebuie să apăsăm pe “Număr versiune” de 5 ori, după cum putem vedea în Figura 3.1.

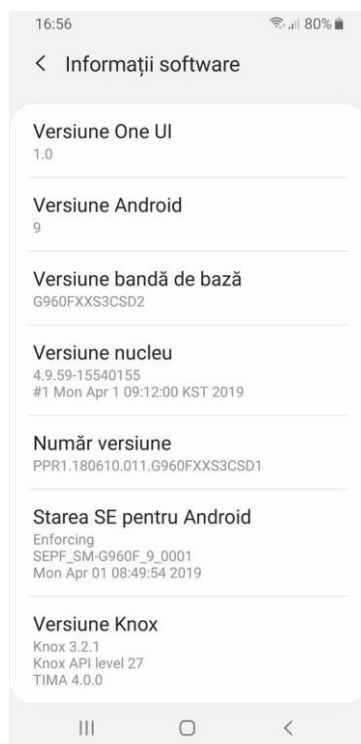


Figura 3.1. Număr versiune a telefonului.

Din telefon se va activa modul USB Debugging și se va activa setarea de permitere a aplicațiilor din alte surse decât cele oficiale. Acum aplicația este funcțională.

În continuare, vom prezenta câteva screenshot-uri ale aplicației în figurile 3.2, 3.3, 3.4.



Figura 3.2. Holul etajului 3 din corpul B al Universității.

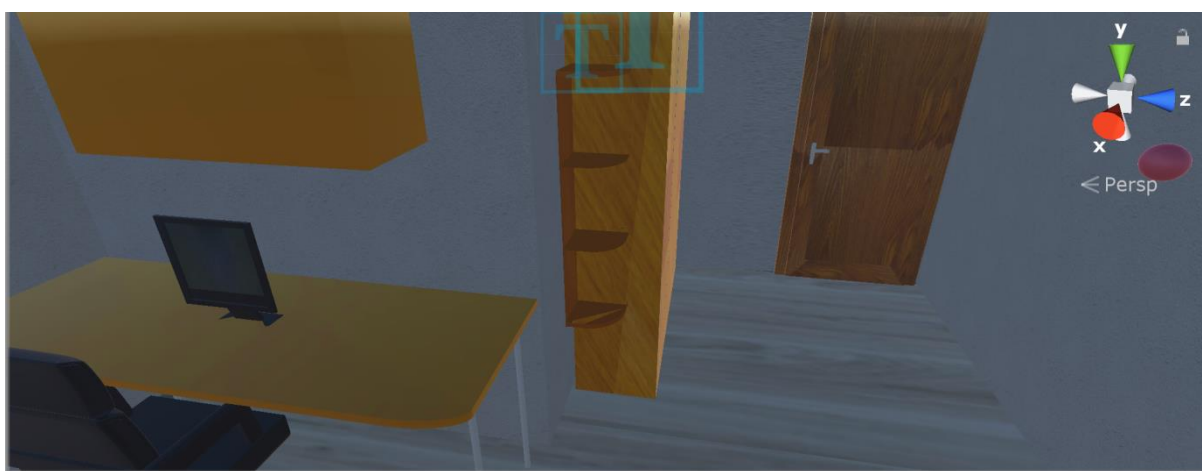


Figura 3.3 Biroul domnului Prof. Dr. Ing. Gabriel Rădulescu.

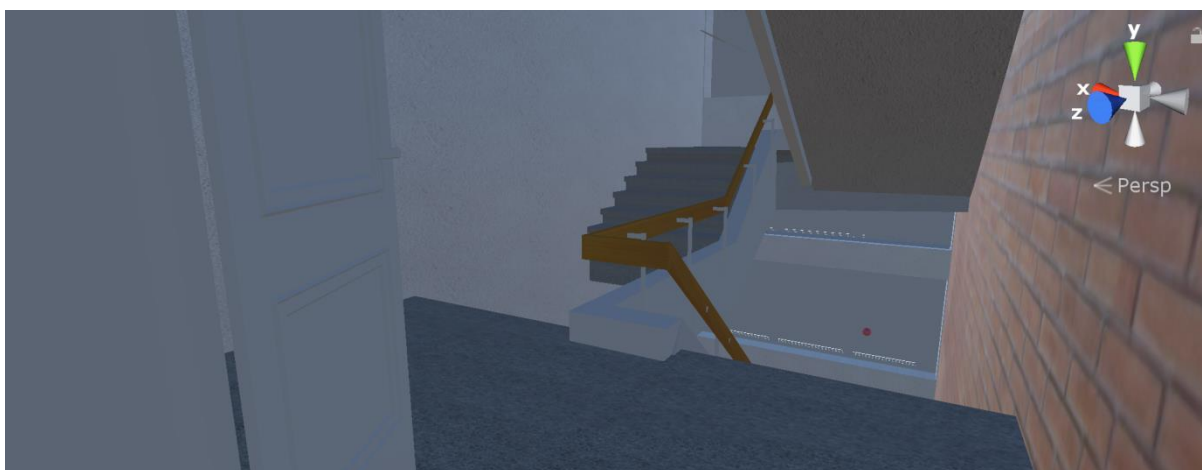


Figura 3.4. Scarile Universității.

Capitolul 4. Concluzii și direcții de cercetare ulterioare

Testarea aplicației a fost realizată în două etape. Prima etapă de testare a fost realizată în motorul grafic Unity pentru eventualele erori de funcționare și corectare a acestora, de asemenea și pentru problemele legate de plasarea obiectelor 3D. Cea de-a doua etapă a testării a fost realizată utilizând ochelarii Google Daydream și telefonul compatibil ochelarilor pe care rula aplicația.

Urmărind pașii de realizare a aplicației văzuți în capitolul anterior, am observat faptul că realizarea pe o scară 1:1 a corpului B este foarte greu de obținut. Pe lângă datele colectate de noi cu ajutorul telemetrului și a pozelor realizate, a fost nevoie și de realizarea câtorva ajustări. Obiectele care au foarte multe detalii au fost foarte greu de realizat, chiar dacă am încercat să combinăm cele două moduri de implementare utilizate, cea prin măsurători și cea prin imagini.

Pe parcursul realizării aplicației, pornind de la modelarea în Blender, până la implementarea aplicației în Unity am întâmpinat o serie de probleme, pe care a trebuit să le rezolvăm.

Din cauza faptului că există o continuă îmbunătățire și schimbare a versiunii software-ului grafic 3D Blender, am avut o problemă în identificarea instrumentelor necesare pentru modelarea corpului B deoarece o parte din aceste instrumente au fost fie redenumite, fie modificate (eliminarea sau adăugarea anumitor setări a ustensilelor respective), fie mutate în altă secțiune, fie lipseau și a trebuit să le adăugăm manual. Suportul pe care l-am găsit nu a fost al versiunii curente, acesta neexistând și a trebuit să ne dăm seama de anumite metode de activare și utilizare a uneltelor de lucru.

La nivelul motorului grafic Unity, problema principală a fost modificarea versiunii deoarece noi am lucrat pe o versiune din 2018, fiind compatibil cu kit-ul de dezvoltare pentru Android. Pe parcursul dezvoltării aplicației, am încercat să trecem la versiunea de Unity lansată în 2019 și am constatat că nu mai puteam continua dezvoltarea aplicației începută în 2018, deoarece nu există o versiune actualizată și compatibilă a kit-ului de dezvoltare Android și a trebuit să revenim la o versiune anterioară.

Principalele probleme care împiedicau finalizarea aplicației au fost legate de volatilitatea suportului Realității Virtuale deoarece SDK-ul a fost modificat, anumite componente necesare din secțiunea Prefab a kit-ului de dezvoltare Google au fost eliminate

Din punct de vedere al costului pentru realizarea aplicației, atât software-ul grafic 3D Blender cât și motorul grafic Unity sunt medii de dezvoltare gratuite, însă a trebuit să achiziționăm telemetrul, ochelarii Google Daydream și suportul informațional.

Din punctul de vedere al resurselor utilizate pentru dezvoltarea aplicației, am folosit un calculator ale cărui componente sunt: procesor AMD RYZEN 5, 8 GB RAM, placa video Nvidia GTX 1070 8GB.

Pentru rularea aplicației ca și resurse am folosit o pereche de ochelari Google DayDream și un telefon Samsung Galaxy S9 dotat cu un procesor Exynos 9810 Octa Core, 4GB RAM, placa video Mali-G72 MP18, memorie internă 64 GB. Am folosit acest model de telefon deoarece este construit pentru Realitatea Virtuală cu o rezoluție mare a display-ului, nivelul de grafică foarte ridicat și senzori de înaltă fidelitate.

La rularea aplicației pe echipamentul utilizat nu au existat probleme de funcționare.

Din punctul de vedere al optimizării, inițial, am creat un script pentru deplasare având ca metodă de deplasare înclinarea capului la un unghi de 30 de grade, însă s-a dovedit inefficient și cauza probleme senzoriale/neuronale utilizatorului și inducea o stare de amețală. Pentru eliminarea acestui impediment, am schimbat metoda de deplasare utilizând o metodă mai eficientă, aceea fiind realizarea mișcării prin waypoint. Astfel se acordă un grad mai mare de libertate din punct de vedere al vizualizării al utilizatorului în timpul deplasării.

Aplicația realizată poate fi utilizată de către viitorii studenți sau oricare altă persoană care dorește să viziteze Universitatea.

Câteva direcții viitoare ar fi implementarea folosind un controller pentru interacționarea utilizatorului cu obiectele, adăugarea unor obiecte cât mai detaliate și a mai multor camere, poate chiar și utilizatori care se plimbă prin Universitate.

Rezumat

În această lucrare am avut drept scop reprezentarea corpului B al Universității, modelarea fiind realizată prin măsurători/imagini de referință.

Pentru a realiza acest proiect am utilizat software-ul grafic 3D Blender și motorul grafic Unity, ambele fiind open-source. Blender a fost utilizat pentru crearea modelelor 3D, iar Unity pentru implementarea aplicației.

În cadrul acestui proiect au fost prezentate noțiuni introductive și generale legate de Realitatea Virtuală, avantajele utilizării acesteia pe deoparte, dar și dezavantajele utilizării ei pe de altă parte.

În cele ce au urmat au fost prezentați pașii pentru proiectarea și implementarea aplicației (determinare caracteristici geometrice, realizare model în mediul Blender, importare model în motorul grafic Unity, importare și configurare aplicație pe platformă mobilă, testare aplicație, corecții).

În urma realizării acestei aplicații am dedus una din principalele concluzii că este dificil și durează foarte mult timp pentru a realiza un model la o scară 1:1 a nivelului de detaliu. Din cauza volatilității domeniului VR, există incompatibilități între versiunile motorului grafic, ceea ce necesită rescrierea în unele cazuri a scripturilor, eliminarea anumitor componente chiar și adăugarea unor componente noi.

În acest proiect s-a urmărit implementarea unei modalități eficiente de deplasare și de interacționare cu obiectele(uși, panouri). Dezvoltarea proiectului poate continua prin extinderea metodelor de interacțiune a utilizatorului cu obiectele realizate, de asemenea și modificarea aplicației în vederea compatibilizării cu alte echipamente care pot reda aplicații ale realității virtuale precum calculatorul sau consolă (xbox și playstation).

Referinte

- The focal length and camera orientation estimation is based on [Using Vanishing Points for Camera Calibration and Coarse 3D Reconstruction from a Single Image](#) by E. Guillou, D. Meneveaux, E. Maisel, K. Bouatouch.
- BLAM does least squares solution of systems of linear equations using the pure python linear algebra routines available [here](#) and does not depend on external packages such as scipy.

Bibliografie

[1] United States Congress, Office of Technology Assessment, “Virtual Reality and Technologies for Combat Simulation: Background Paper.” OTA-BP-ISS-136, U. S. Government Printing Office, Washington, DC, September 1994. Available on the In-ternet by anonymous ftp from OTA’s ftp server, at <ftp://otabbs.ota.gov/pub/vr.military/> The file <ftp://otabbs.ota.gov/pub/vr.military/references.txt> lists references, many of them available on-line by ftp or http. A system used in the SAGE Air Defense System (1958) included distributed interactive live and virtual simulation (p. 20.).

[2] J. M. Rolfe and K. J. Staples, eds., Flight Simulation. Cambridge University Press, Cambridge, 1985. For example, the U. S. Air Force Advanced Simulator for Pilot Training was delivered in 1974 (p. 154.).

[3] R. P. Greenfield, “Navy VR Flight Simulator,” Virtual Reality Special Report, Vol. 1, No. 3, Fall 1994, p. 61–64

[4] M. Moshell, “Virtual Environments in the U.S. Military,” Computer, Vol. 26, Feb. 1993, pp. 81–82.

[5] E. A. Alluisi, “The Development of Technology for Collective Training: SIMNET, a Case History,” Human Factors, Vol. 33, No. 3, June 1991, pp 343–362.

[6] T. W. Mastaglio and R. Callahan, “A Large-Scale Complex Virtual Environment for Team Training,” Computer, Vol. 28, No. 7, July 1995, pp. 49–56.

[7] L. Brill, “Metaphors for the Traveling Cybernaut,” Virtual Reality World, Vol. 1, No. 1, Spring 1993, pp. q–s. (Note: this issue of Virtual Reality World is an insert in Multimedia Review.)

- [8] P. Jackson, “Applications of Virtual Reality in Training Simulation,” in K. Warwick, J. Gray, and D. Roberts, eds., *Virtual Reality in Engineering*. The Institution of Electrical Engineers, Stevenage, Herts, UK, 1993, p. 134.
- [9] R. B. Loftin, M. Engelberg, and R. Benedetti, “Applying Virtual Reality in Education: A Prototypical Virtual Physics Laboratory,” in *Proceedings of the IEEE 1993 Symposium on Research Frontiers in Virtual Reality*. IEEE Computer Society Press, Los Alamitos, CA, 1993, pp. 67–74.
- [10] P. Yam, “Surreal Science: Virtual Reality Finds a Place in the Classroom,” *Sci. Am.*, Vol. 266, No. 2, Feb. 1993, pp. 103–104.
- [11] C. Dede and L. Fontana, “Reconceptualizing Distance Learning in Science and Technology Education,” *Speculat. Sci. Technol.*, Fall 1994.
- [12] “Vanderbilt University: Creating Virtual Engineering Laboratories,” *Syllabus*, Vol. 8, No. 2, Oct 1994, pp. 33–35.
- [13] J. R. Bourne. Personal communication with Veronica Pantelidis.
- [14] K. Nemire, “Virtual Laboratory for Disabled Student,” in Harry J. Murphy, ed., *Proceedings of the Second Annual International Conference on Virtual Reality and Persons With Disabilities*. California State University, Northridge, CA, 1994.
- [15] K. Nemire, A. Burke, and R. Jacoby, “Human Factors Engineering of a Virtual Laboratory for Students with Physical Disabilities,” *Presence*, Vol. 3, No. 3, Summer 1994, pp. 216–226.
- [16] Gibson, D. Brown, S. Cobb, and R. Eastgate, “Virtual Reality and Rapid Prototyping,” in K. Warwick, J. Gray, and D. Roberts, eds., *Virtual Reality in Engineering*. The Institution of Electrical Engineers, Stevenage, Herts, UK, 1993, pp. 51–63.
- [17] C. Bridgewater, “Computer-Aided Building Design and Construction,” in K. Warwick, J. Gray, and D. Roberts, eds., *Virtual Reality in Engineering*. The Institution of Electrical Engineers, Stevenage Herts, UK, 1993, pp. 65–89.
- [18] D. Harrison, M. Jaques, and P. Strickland, “Design by Manufacture Simulation Using a Glove Input,” in K. Warwick, J. Gray, and D. Roberts, eds., *Virtual Reality in Engineering*. The Institution of Electrical Engineers, Stevenage, Herts, UK, 1993, pp. 91–103.

[19] P. du Pont, “Applied Virtual Reality,” in K. Warwick, J. Gray, and D. Roberts, eds., *Virtual Reality in Engineering*. The Institution of Electrical Engineers, Stevenage, Herts, UK, 1993, pp. 153–167.

[20] W. Webster, “Telerobotics in the Nuclear Industry,” in K. Warwick, J. Gray, and D. Roberts, eds., *Virtual Reality in Engineering*. The Institution of Electrical Engineers, Stevenage, Herts, UK, 1993, pp. 183–188.

[21] N. I. Durlach and A. S. Mavor, eds., *Virtual Reality: Scientific and Technological Challenges*. National Academy Press, Washington, 1995, pp. 435–436.

[22] N. Adams and L. Lang, “VR Improves Motorola Training Program,” *AI Expert*, May 1995, pp. 13– 14.

[23] The author of this article has seen an explosive growth in interest in using VR in education, reflected in a continual increase in information requests made to the Virtual Reality and Education Laboratory, of which she has been co-director since 1992. The majority of these requests come via the Internet.

[24] “Avatar Partners to Develop Full Body Wireless Virtual Reality System for Dismounted Infantry Training; U. S. Army Simulation Training and Instrumentation Command Funds Development of ‘Holodeck-like’ Virtual Reality System.” Press release, February 17, 1995, by Avatar Partners, Inc., 13140 Central Avenue, Boulder Creek, CA 95006-9120.

[25] Virtual Reality and Engineering Education - VERONICA S. PANTELIDIS
Virtual Reality and Education Laboratory, School of Education, East Carolina University,
Greenville, NC 27858.

Webografie

- [25]<https://www.blender.org/about/>
- [26]<https://unity3d.com/unity>
- [27]http://c3.icvl.eu/papers2016/cniv/documente/pdf/sectiuneaA/sectiuneaA_lucrarea7.pdf
- [28]<https://www.audio.ro/stiatica/cum-ne-poateschimba-realitatea-virtuala-viata/>
- [29]<https://www.udemy.com/build-30-mini-virtual-reality-games-in-unity-3d-from-scratch/learn/lecture/6230709#overview>
- [30]<http://www.skills2learn.com/virtual-reality-benefits.html>
- [31]<https://docs.blender.org/manual/en/latest/preferences/addons.html>
- [32]<https://github.com/stuffmatic/blam>
- [33]<https://github.com/stuffmatic/blam/wiki/User%27s-guide>
- [34]<https://github.com/stuffmatic/blam/releases>
- [35]<https://www.youtube.com/watch?v=yeWzQYh7iH0&fbclid=IwAR23zdTTHqsKWuZuPN0DpzcI4JsSmKydTJZL0Lht4MxcRjksteNizZBEhdc>
- [36]https://www.youtube.com/watch?v=SmxtF_D82xk&t=62s&fbclid=IwAR1VQP8rII-CRITe57rvo40dyuO3deghe-LZYNOjxZDvWHA6na2KHFMRXns
- [37]https://www.youtube.com/watch?v=SmxtF_D82xk&t=62s&fbclid=IwAR1VQP8rII-CRITe57rvo40dyuO3deghe-LZYNOjxZDvWHA6na2KHFMRXns
- [38]https://www.youtube.com/watch?v=_rDfJlyqDC8