

UNIVERSITATEA ALEXANDRU IOAN CUZA IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

ESCAPE ROOM XR

propusă de

Alexandru-Daniel Marian

Sesiunea: iulie, 2019

Coordonator științific
Prof. Colab. Florin Olariu

UNIVERSITATEA ALEXANDRU IOAN CUZA IAȘI
FACULTATEA DE INFORMATICĂ

ESCAPE ROOM XR

Alexandru-Daniel Marian

Sesiunea: iulie, 2019

Coordonator științific
Prof. Colab. Florin Olariu

Avizat,
Îndrumător Lucrare de Licență
Prof. Colab. Florin Olariu

Data _____ Semnătura _____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemnatul **Alexandru-Daniel Marian** cu domiciliul în **Nicolae Bălcescu, Bacău**, născut la data de **10 Ianuarie 1998**, identificat prin CNP **1980110046215**, absolvent al Universității „Alexandru Ioan Cuza” din Iași, Facultatea de **Informatică**, specializarea **Informatică**, promoția **2016-2019**, declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art. 143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul: **ESCAPE ROOM XR** elaborată sub îndrumarea dl. **Florin Olariu**, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi, _____

Semnătură student, _____

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „ESCAPE ROOM XR”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași,

Absolvent Alexandru-Daniel Marian

(semnătura în original)

Cuprins

Cuprins	5
Introducere	7
Abstract	7
Motivație	7
Context	9
Aplicații existente	10
Contribuții	14
Capitolul I. Descrierea problemei	15
Capitolul II. Descrierea soluției	16
II.1. Descrierea soluției pe scurt	16
II.2. Arhitectură și implementare	16
Concluziile lucrării	19
Bibliografie	20
Anexa 1	23
A1.0. Descrierea aplicației pe scurt	23
A1.1. Arhitectura aplicației	23
A1.2. Implementarea aplicației	25
A1.2.a. Clasele AR Foundation	25
A1.2.b. Obiectul Ui Logic și Meniurile	26
A1.2.c. Obiectul GameLogic și Placement Indicator	29
A1.2.d. SimpleScreenCanvas și SplitScreenCanvas	31
A1.2.e. Scriptul „Interact With Objects With Controller”	32
Anexa 2	33
Termeni, noțiuni și device-uri	33

A2.1. Raycast	33
A2.2. Coliziune	33
A2.3. KeyCode	34
A2.4. Oculus Rift și Vive	34
Anexa 3	36
Tehnologii utilizate	36
A3.1. Unity	36
A3.1.a. Motorul de dezvoltare	36
A3.1.b. Setul de pași logici în dezvoltarea oricărui tip de joc	36
A3.2. Pachetul AR Foundation	38
A3.3. Android Studio	39
A3.4. Paint.net	39
Anexa 4	40
Manual de utilizare	40
A4.1. Meniul principal	40
A4.2. Meniurile secundare	40
A4.3. Prima interacțiune cu jocul	41
A4.3.a. Pregătirea mediului	41
A4.3.b. Jocul propriu-zis	42

Introducere

Abstract

Aceasta lucrare de licență ilustrează posibilitatea îmbinării mediului virtual cu cel real printr-un joc de tip „Cross Reality” (XR). Acest concept de realitate mixată reprezintă un mod de a combina conceptul de realitate augmentată (AR) cu realitatea virtuală (VR) prin crearea de noi medii propice unor noi stiluri de aplicații. În jocul „Escape Room XR”, poziția jucătorului este realizată cu ajutorul pachetului AR Foundation, pachet ce ușurează munca dezvoltatorilor de aplicații augmentate în ceea ce privește tracking-ul device-ului în mediul real. Pentru partea realității virtuale, camera virtuală ce este plasată în mediul real, este redimensionată astfel încât utilizatorul să se afle în interiorul acesteia, feed-ul camerei de la telefon fiind randat o singură dată și afișat de două ori pe ecran pentru posibilitatea utilizării unei perechi de ochelari de realitate virtuală.

Motivație

Totul a început de când am văzut un anime¹ pe internet intitulat „Sword Art Online”. Povestea nu este una prea fericită, multe întâmplări ciudate având loc, însă conceptul în sine m-a fascinat. Fiind pasionat de calculator și de ceea ce înseamnă jocuri video, atât jucatul cât și crearea lor, am început încetul cu încetul să descopăr acest concept de realitate virtuală. Cu timpul Youtube-ul a devenit tot mai populat de filmulețe cu diferite jocuri VR, utilizatorii folosind un headset pe nume Oculus (vezi Figura 1 de mai jos) sau Vive (vezi Figura 2 de mai jos) pentru a le juca. Urmărind aceste filmulețe, dorința de a încerca și eu un device de genul a crescut enorm (pentru mai multe informații despre device-uri vezi secțiunea A2.4 din Anexa 2).

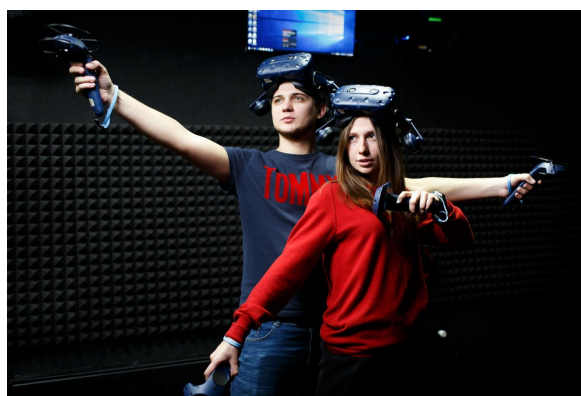
¹ În „Sword Art Online” este vorba despre niște persoane, în mare parte copii, care își achiziționează un device intitulat Nerve Gear. Acțiunea petrecându-se undeva după anul 2020, tehnologia fiind una suficient de dezvoltată. Acele persoane își puneau pe cap o cască ca de motociclist, stăteau întinși în pat, și se conectau la un mediu virtual, un joc de tip MMORPG unde ei își petreceau o mare parte din timpul lor.

După ce am intrat la facultate și am început să îmi aprofundez cunoștințele de programare, am avut norocul într-o zi să testez o aplicație VR creată special pentru Oculus/Vive. Experiența a fost una de nedescris. Am ajuns să fiu fascinat de ideea de a crea eu astfel de aplicații. Știam pe atunci că este posibil să îți dezvolti o aplicație de realitate augmentată sau de realitate virtuală pentru telefonul mobil, însă totul pare destul de limitat. Dacă îți creezi o aplicație AR, fără un device precum HoloLens, orice augmentezi în lumea reală poți să vezi doar prin ecranul telefonului tău mobil, ceea ce nu îți oferă o oarecare imersiune. Dacă în schimb îți creezi o aplicație VR, ai o lume complet virtuală însă modul în care te deplasezi în acea lume prin apăsarea unui buton iar reduce nivelul de imersiune. Desigur că s-au creat mai multe posibilități de a te deplasa în lumea virtuală pentru a nu reduce nivelul de imersiune, însă acele posibilități sunt costisitoare monetar.

Într-o zi, mi-a venit ideea de a încerca să sparg această barieră care strică imersiunea în aplicațiile AR și în cele VR făcute pe mobil. Cercetând mai amănunțit despre cum funcționează cele două concepte am realizat că este posibil. Așadar am început să realizez acest lucru, iar un mod în care să arăt că acest lucru este posibil, a fost creând un joc de tipul escape room. Aici și-a pus amprenta și o oarecare pasiune pentru tot ceea ce înseamnă puzzle pentru mine.



(Figura 1) Oculus Rift (sursa²)



(Figura 2) Vive Pro Eye (sursa³)

2

https://unsplash.com/photos/zj_saLHzwk?fbclid=IwAR3WUtn6lmm6JW-XmdFPRWJME3D2Huux02U_sVBdYPs_h_jc-8Jw51u0yig

3

<https://unsplash.com/photos/f376CkFI5iY?fbclid=IwAR2mXA64DBe2zUHuhylAAdu1jkWGr9biHAdF32KazQkQsag4p5wE6HMR1I4>

Context

Conform unui articol de pe Quora (vezi sursa⁴) postat în Februarie 2019, un escape room este un joc de tipul puzzle, ce se joacă în echipe. Jucătorii sunt închiși într-o încăpere unde au de rezolvat diferite puzzle-uri și să completeze anumite cerințe împreună, pentru a evada din acea încăpere. Uneori este și o poveste peste care este axat jocul, motivând astfel jucătorii, ei fiind protagoniștii.

Ideea de Escape Room a fost preluată din jocurile video de acest tip. Acest joc se bazează pe anumite criterii, însă pot varia cu ușurință de la ele. Anumite criterii sunt:

- Un obiectiv cât mai clar - De obicei în acest tip de joc, pasul final este găsirea unei chei, o combinație la încuietoare sau activarea unui mecanism care să deblocheze ușa finală.
- Puzzle-uri - Jocul în sine se axează pe rezolvarea mai multor puzzle-uri cum ar fi: găsirea unor obiecte ascunse prin încăpere, găsirea unor cuvinte cheie, a unor coduri, recunoașterea unor pattern-uri sau potrivirea lor, deblocarea unor cifruri etc.
- O limită de timp - Se impune de obicei o limită de timp pentru a evada din încăpere. Timpul este de aproximativ 60 de minute, perioada în care toți cei închiși pot rezolva cât mai multe puzzle-uri cu scopul de a le finaliza și de a ieși din încăpere.
- O poveste - Cui nu-i plac poveștile? Dacă ești dornic să faci parte dintr-una, trebuie să încerci un astfel de joc. Unele Escape Rooms sunt construite pe baza unor povești, în care jucătorii sunt anumite personaje din narațiunea respectivă.
- Să existe o echipă - Numărul maxim de jucători într-o cameră variază de la 4 până la 12, însă sunt momente când sunt și camere ce se joacă în cel mult 2 persoane. Lucrul în echipă este necesar, indiciile fiind descoperite mai ușor când mai multe persoane colaborează în căutarea lor.
- Tematica - Camera trebuie să fie decorată astfel încât să se potrivească cu povestea, să îi inducă jucătorului o experiență de neuitat. În camerele cu cele mai mari venituri se investește mult efort și multă experiență în a crea o atmosferă cât mai realistă, cu cele mai amănunțite detalii.

Unele elemente tipice ce se regăsesc în majoritatea jocurilor de escape room sunt: o ușă încuiată (ușa finală pe care trebuie să o deschizi), încuietori și chei (pentru a bloca și a

4

https://www.quora.com/What-is-an-escape-room-game?fbclid=IwAR230f1IFTPWrqfyby5pfw2TT5MvJ6UQiSEssnlw_T6Mn2NrgBZZ82crvME

deschide diferite obiecte: lăzi, uși), uși ascunse (întrări către alte încăperi, ce sunt activate prin întrerupătoare magnetice sau sisteme electronice), mai multe încăperi (uneori camera în care te afli nu este camera prin care poți să evadezi).

Aplicații existente

Odată cu apariția celor două noi trenduri realitatea augmentată și realitatea virtuală, dezvoltatorii de jocuri video au început să creeze și jocuri pentru device-uri care folosesc aceste trenduri. Pe lângă multitudinea de joculețe în browser, de tipul escape room, unii developeri au creat jocuri de escape room pentru device-uri precum Oculus Rift, Vive cât și pentru device-urile mobile.

Un joc de tipul escape room folosind telefonul mobil împreună cu o pereche de ochelari VR este „Escape Room The Game” (vezi Figura 3 de mai jos). Acest joc presupune utilizarea unor indicii din lumea reală, sau din lumea virtuală, și rezolvarea unor puzzle-uri fizice în mediul real. Jocul vine cu un pachet cu puzzle-urile necesare unui nivel, și cu o aplicație pentru telefonul mobil. Astfel, o persoana poate utiliza aplicația ca să descopere indicii în mediul virtual în timp ce coechipierul se străduiește să rezolve puzzle-urile în mediul real și să își dea seama de alte indicii care îi pot rezolva puzzle-ul. Jocul se termină atunci când cheia este introdusă în așa numitul „Chrono Decoder” (vezi Figura 4 de mai jos). Acest obiect „Chrono Decoder” acceptă 4 chei ce sunt create cu un design specific. Jocul conține 16 chei, câte 4 din fiecare tip de cheie. Fiecare cheie conține o formă geometrică, litere, simboluri etc. În acest joc, utilizatorul nu se poate deplasa în mediul virtual. Tot ceea ce poate el să facă este să se uite în jur și să interacționeze cu anumite obiecte din împrejurime.



(Figura 3) Escape Room The Game (sursa⁵)



(Figura 4) Chrono Decoder (sursa⁶)

Un joc de tipul escape room în care utilizatorul folosește un headset VR de tipul Oculus Rift sau Vive este PROZE: Prologue (vezi Figura 5 și Figura 6 de mai jos). În acest escape room, utilizatorul se deplasează folosind un buton pe controller. Grafica jocului este una de apreciat. Utilizând și un device performant de realitate virtuală, experiența este una foarte plăcută.

Un lucru interesant în acest joc este prezența unui NPC (caracter ce nu este un jucător real). Introducerea acestuia induce o caracteristică a jocurilor de escape room și anume aceea de lucru în echipă. Acest NPC de-a lungul jocului te ajută cu diferite indicii pe care el este pus să le descopere.

Sunt multe alte jocuri asemănătoare cu acesta, însă fiecare este unic în felul lui, prin poveste, mecanici, puzzle-uri etc. Asemănarea constă în utilizarea unui device de realitate virtuală și de minim un controler pentru a interacționa cu elementele din împrejurime.

5

https://escaperoomthegame.com/en-au/?fbclid=IwAR2kVNvGECa3v3bRXKZGR3c67nmYPGgfSSNZbdt45t_QgEIL8Re37iwqQu8

6 <https://escaperoomthegame.com/en-au/about>



(Figura 5) PROZE: Prologue



(Figura 6) PROZE: Prologue

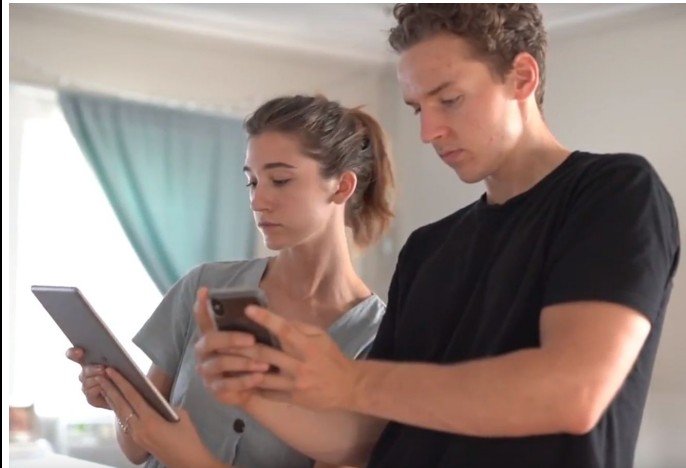
(ambele imagini sursa⁷)

Un joc de tipul escape room care folosește realitatea augmentată ca și concept este Escape The Room: AR (vezi Figura 7 de mai jos). În acest joc, utilizatorul își transformă propria cameră într-un escape room cu ajutorul tehnologiei AR. Jucătorul va scana camera și obiectele vor fi plasate. Acesta, cu ajutorul device-ului cu iOS, va folosi camera device-ului pentru a scana împrejurimile. În urma scanării, puzzle-urile sunt plasate iar jucătorul poate să înceapă rezolvarea lor.

O caracteristică impresionantă a acestui joc este posibilitatea de a te conecta cu încă cel mult 3 prieteni (vezi Figura 8 de mai jos) și a încerca să rezolvați puzzle-urile în echipă.

⁷

https://store.steampowered.com/app/912560/PROZE_Prologue/?fbclid=IwAR2Q1LuoBXo4232SHh2H6Mza7WZZ7Kx74A27xbsfYkVbTc-N96EqDmsEpTM



(Figura 7) Escape The Room: AR

(Figura 8) Escape The Room: AR

(ambele imagini sursa⁸)

8

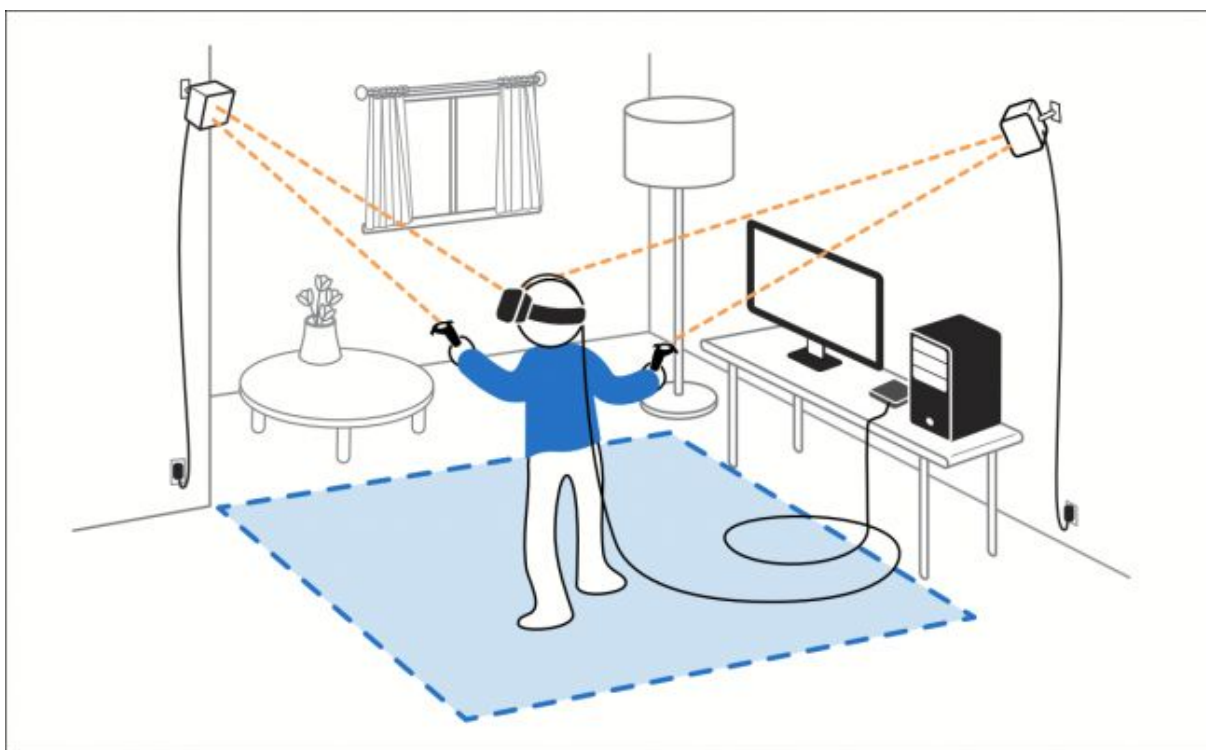
https://www.producthunt.com/posts/escape-the-room-ar?fbclid=IwAR1P_tnIKru2ID-4ZHa5m5Amt1Kna9FbqMVFWl9RlnJUI2aLN-To7uRpRoI

Contribuții

Ceea ce am realizat eu pentru proiectul de licență este un joc de tipul escape room în care am combinat cele două concepte, cel de realitate augmentată cu acela de realitate virtuală. Prin utilizarea realității augmentate folosesc feed-ul camerei video de la device pentru a scana mediul înconjurător și a plasa încăperea virtuală în realitate. Încăperea fiind suficient de mare pentru ca utilizatorul să se afle în interiorul ei, tot ce poate vedea este un mediu virtual. Pentru a ține jucătorul în mediul virtual când acesta trece prin peretele încăperii, am creat un skybox (cer fals) de culoare neagră dând astfel senzația că este captiv într-o cameră ce se află în vid. Folosindu-mă de feed-ul camerei video și de pachetul AR Foundation (vezi Anexa 3 pentru informații despre tehnologii), pot ști în permanență poziția device-ului în camera virtuală. Astfel jucătorul se poate plimba în mediul virtual mergând în mediul real. Ce am adus eu nou față de aplicațiile menționate mai sus a fost să sparg bariera dintre cele două tehnologii combinându-le. Ceea ce face aplicația specială este posibilitatea oricărei persoane posesoare de un device cu android, cu dimensiunea ecranului între 4,5 și 6 inch, de un controler bluetooth și de o pereche de ochelari de realitate virtuală în care se introduce telefonul, ca acesta să poată experimenta un joc de realitate virtuală fără a avea nevoie de o cască de mare performanță precum Oculus Rift, de un controler special pentru acesta și de senzori ce trebuie plasați într-o încăpere, toate conectate la un calculator destul de performant. Costurile sunt așadar enorm reduse. Majoritatea persoanelor sunt deținătoare de un smartphone, cu dimensiunea ecranului ce se încadrează în criteriul menționat mai sus. Un controler bluetooth împreună cu o cască VR pot ajunge la mai puțin de 100 de lei cumulate. De senzori nu este nevoie, aceștia fiind înlocuiți de camera device-ului.

Capitolul I. Descrierea problemei

Jocurile de realitate virtuală care au un perimetru de interacțiune stabilit pentru utilizator, necesită niște senzori adiționali pe lângă headset-ul de un preț ridicat și manete de control pentru acesta. Acești senzori detectează headset-ul VR și manetele de control pe un perimetru în care aceștia sunt amplasați (vezi Figura 9 de mai jos). Odată făcut setup-ul, utilizatorul poate juca aplicații în care este necesar să te deplasezi într-o anumită rază, pe un anumit perimetru ca și cum ar fi întru totul în mediul virtual.



(Figura 9) Senzori pentru tracking-ul device-urilor VR (sursa⁹)

Aplicațiile dezvoltate pentru device-uri mobile de tipul VR nu au posibilitatea integrării unor senzori adiționali pentru a detecta poziția utilizatorului în mediul virtual pe baza poziției acestuia în lumea reală. Așadar toate aplicațiile de tipul VR presupun utilizarea unui controler adițional sau diferite implementări în joc pentru deplasare în mediul virtual. Utilizatorul deplasându-se prin apăsarea unui buton, imersiunea este afectată.

⁹ <https://www.howtogeek.com/246333/oculus-rift-vs.-htc-vive-which-vr-headset-is-right-for-you/>

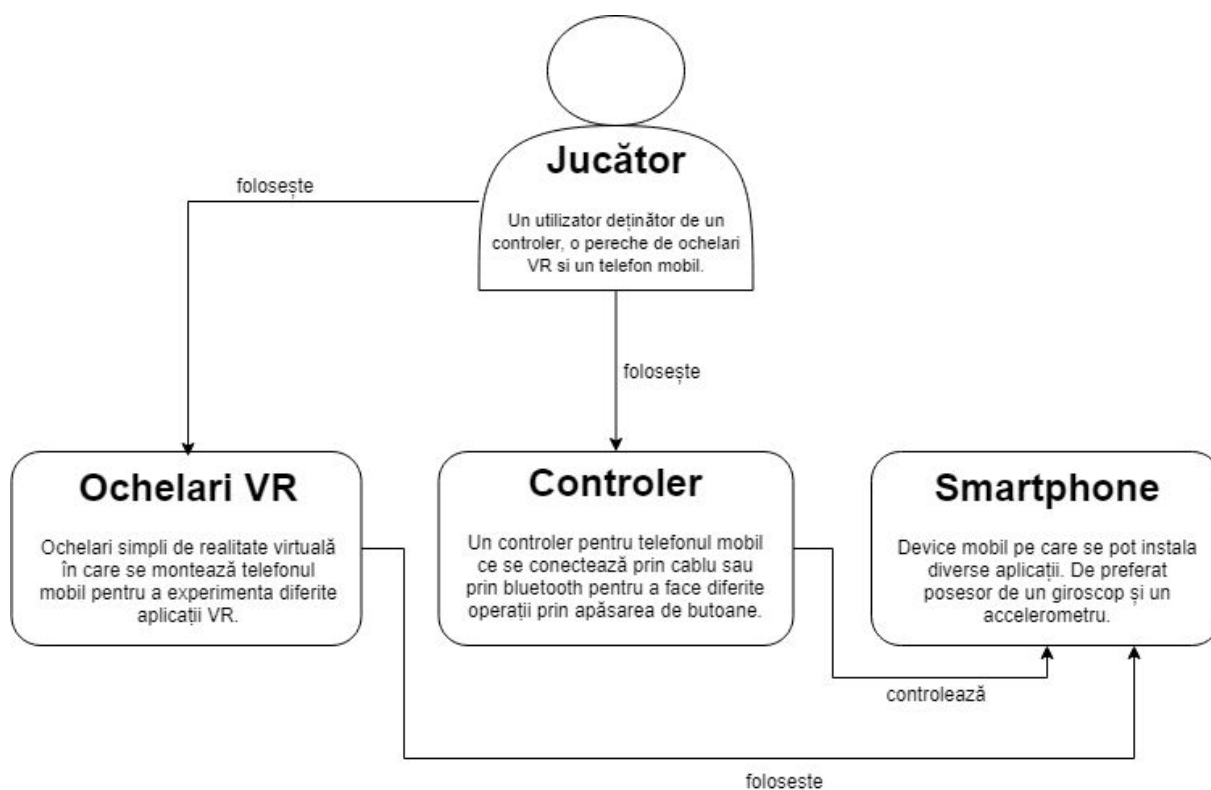
Capitolul II. Descrierea soluției

II.1. Descrierea soluției pe scurt

Cea mai propice metodă de a crea posibilitatea de a detecta o poziție în mediul virtual pe baza poziției utilizatorului în realitate a fost de a utiliza realitatea augmentată. Prin realitate augmentată poziția device-ului în lumea reală este percepută continuu. Așadar m-am gândit să mă folosesc de acest beneficiu și să încerc să ofer o percepție asupra unui mediu virtual, proiectând obiectele la o scală suficient de mare astfel încât jucătorul să se afle în interiorul lor. Astfel este acoperit tot feed-ul video al camerei de la device, rămânând o iluzie cum că tot ce se întâmplă este în mediul virtual. Prin această percepție reiese conceptul de realitate mixată.

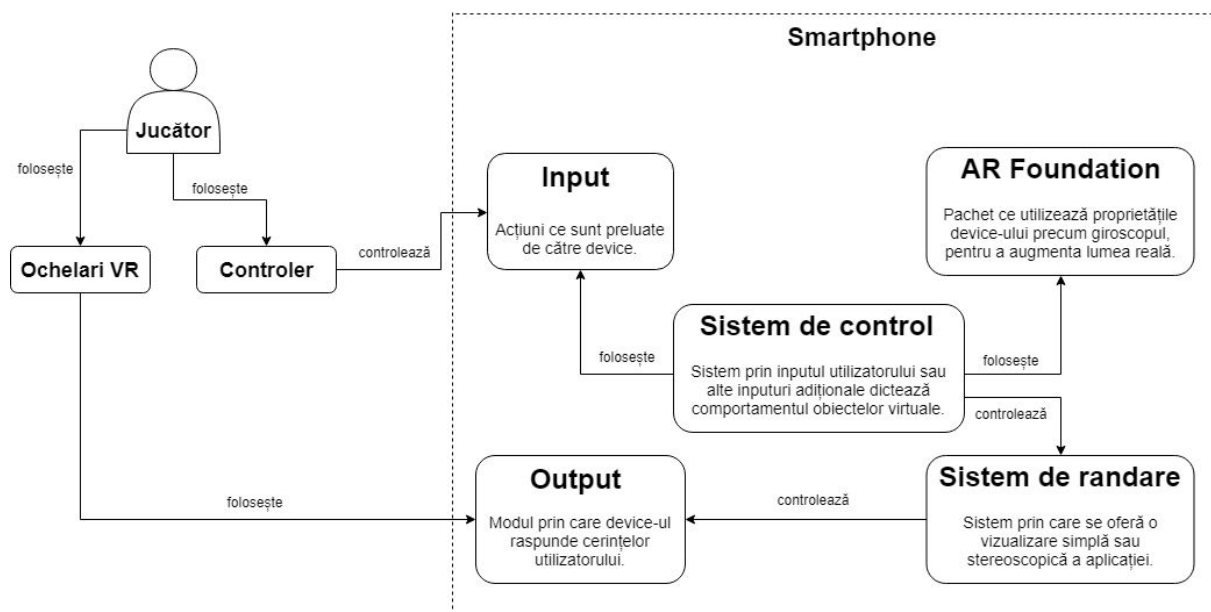
II.2. Arhitectură și implementare

La un prim nivel urmărind arhitectura aplicației de tipul c4 model (vezi Figura 10 de mai jos), se poate observa interacțiunea utilizatorului cu aplicația. Acesta utilizează două device-uri, o pereche de ochelari de realitate virtuală și un controler cu ajutorul căruia comunică diferite acțiuni device-ului mobil ce se află amplasat în ochelarii VR. Smartphone-ul este cel ce suportă toată funcționalitatea de bază, aplicația fiind instalată pe acesta. În urma inputului primit de la utilizator și de la terți, acesta oferă jucătorului un output pe care cu ajutorul ochelarilor VR, acesta este capabil să îl înțeleagă.



(Figura 10) Arhitectura aplicației: C4 Model, nivelul 1

La un zoom făcut pe Smartphone (vezi Figura 11 de mai jos) putem observa la un nivel mai amănunțit ce se întâmplă cu aplicația. În secțiunea mare, marcată de chenarul punctat se află aplicația instalată pe smartphone ce primește două tipuri de input, cel de la utilizator și altul de la funcțiile device-ului, cum ar fi giroscopul, accelerometrul etc. Inputul de la utilizator este preluat în Unity cu ajutorul unor funcții pe care acesta le expune prin clasa Input (vezi Anexa 1, partea de implementare pentru mai multe detalii). AR Foundation se ocupă de preluarea datelor oferite de device și expune niște metode cu ajutorul cărora se poate augmenta un obiect în lumea reală. Clasa GameLogic împreună cu alte clase constituie sistemul de control al arhitecturii. Acestea pe baza informațiilor asimilate, construiesc mediul în care jucătorul își exercită dorința de a „evada” din camera virtuală. Tot aici se stabilește modul în care mediul virtual împreună cu toate obiectele lui îi este afișat utilizatorului. Sistemul de randare este format din două canvas-uri: SimpleScreenCanvas și SplitScreenCanvas. Unul dintre ele este activat în momentul începerii jocului și afișează ceea ce camera device-ului „vede” într-un mod augmentat. Tot aici, pentru a fi optimizat, se crează o singură randare care mai apoi este expusă pe două imagini, în cazul vizualizării stereoscopice (pentru mai multe detalii vezi secțiunea A1.2.d din Anexa 1).



(Figura 11) Arhitectura aplicației: C4 Model, nivelul 2

Concluziile lucrării

Pe parcursul dezvoltării acestei aplicații am întâmpinat multe probleme, neavând o cunoaștere suficient de aprofundată pentru motorul de dezvoltare și pentru lucrul cu pachetul celor de la AR Foundation, însă cu suficient de mult research am reușit să găsesc căi alternative astfel încât să duc ceea ce mi-am propus la bun sfârșit. Odată acumulată această experiență mă simt suficient de încrezător în mine că pot dezvolta și alte aplicații pe același principiu. Sunt perfect convins că multe îmbunătățiri se pot aduce acestei aplicații de la partea de grafică până în partea de implementare. Eu aș adăuga sunete cât mai realiste, de pași, sunete pentru atunci când diferite obiecte cad de la înălțime, sunete pentru absolut orice, adăugând astfel pe lângă headset și controler, o pereche de căști. De asemenea aș modifica grafica, utilizând texturi cât mai realiste pentru a da o altă tentă jocului. Un sistem de nivele pe diferite dificultăți consider a fi necesitatea principală pentru ca toată lumea, de la cei mai experți în rezolvarea puzzle-urilor până la cei mai nepricepuți să poată să se simtă bine, să se simtă provocați să rezolve puzzle-urile. O integrare multiplayer pentru acest joc ar fi o provocare pe cîste, ce ar aduce un plus mare în ceea ce înseamnă un joc de tipul escape room. Posibilitatea de a te conecta cu prietenii, de a rezolva enigmele împreună cu ei ar face jocul mult mai captivant. Scanarea 3D a unui controler și posibilitatea de a-l recunoaște în mediul virtual pentru a crea o reprezentare a acestuia ar apropia mai mult această configurație de ceea ce înseamnă un setup adevărat de realitate virtuală. Senzorii într-un astfel de setup sunt capabili să detecteze și poziția manetelor VR, acestea putând fi mai apoi reprezentate în lumea virtuală.

Bibliografie

1. Articol Quora despre ce este un joc de tipul escape room.

https://www.quora.com/What-is-an-escape-room-game?fbclid=IwAR230f1IFTPWrqfyby5pfw2TT5MvJ6UQiSEssnlw_T6Mn2NrgBZZ82crvME

2. Escape Room The Game

https://escaperoomthegame.com/en-au/?fbclid=IwAR2kVNvGECa3v3bRXKZGR3c67nmYPGgfSSNZbdt45t_OgEIL8Re37iwqQu8

3. Imaginea lui Vladislav Skripchenko cu tematica headset Vive

<https://unsplash.com/photos/f376CkFI5iY?fbclid=IwAR2mXA64DBe2zUHuhyIAAdU1jkWGr9biHAdF32KazOkQsag4p5wE6HMR1l4>

4. Imaginea lui NeONBRAND cu tematica Oculus Rift

https://unsplash.com/photos/zj_saLHazwk?fbclid=IwAR3WUtn6lmm6JW-XmdFPRWJME3D2Huux02U_sVBdYPs_h_jc-8Jw51u0yig

5. PROZE: Prologue

https://store.steampowered.com/app/912560/PROZE_Prologue/?fbclid=IwAR2Q1LuoBXo4232SHh2H6Mza7WZZ7Kx74A27xbsfYkVbTc-N96EqDmsEpTM

6. Escape The Room: AR

https://www.producthunt.com/posts/escape-the-room-ar?fbclid=IwAR1P_tnIKru2lD-4ZHa5m5Amt1Kna9FbqMVFWI9RInJUI2aLN-To7uRpRoI

7. Imaginea cu viewport screen a fost preluată de la 25games din articolul acesta

http://25games.net/camera-tracking-two-objects-camera-unity/?fbclid=IwAR1I781eVSPRXL_Td_abgf7THfIShwCwSC1IVAHxApegu9Vq7RT_Ia5ck3f8

8. Imaginea cu interiorul unui headset VR a fost preluată de pe wareable de la articolul acesta

https://www.wareable.com/vr/how-oculus-rift-works?fbclid=IwAR3P_9eXQkl4-EUIZPGZg8vJOEyPIcsGOLVb5Ri0mmSbYz2voZSI621CA7s

9. Setul de pași logici în crearea unui joc au fost preluate de pe site-ul oficial Unity

https://unity3d.com/programming-in-unity?fbclid=IwAR20_1Y2z4DdFhj2MAWfl3YLtyk8cE7Vvr36U4cMs5fjDn0fO9krSUNdNaQ

10. Android studio

<https://developer.android.com/>

11. Unity

<https://unity.com/>

12. Paint.net

<https://www.getpaint.net/download.html>

13. Documentația Unity pentru versiunea 2018.3

<https://docs.unity3d.com/2018.3/Documentation/Manual/>

14. Modele low poly free

<https://poly.google.com/>

15. Papucii de fotbal făcuți de printable_models

<https://free3d.com/3d-model/soccer-shoes-v1--135829.html>

16. Ușă, monitor, birou, carte și pernă făcute de Poly by Google

<https://poly.google.com/view/ch-fQ-Ruyog>

<https://poly.google.com/view/e8cELDeDuTr>

<https://poly.google.com/view/auSYhIqAZq0>

<https://poly.google.com/view/4S1nr7WmUxm>

<https://poly.google.com/view/f60emm1Xkas>

17. Pioneză făcută de Louis DeScioli

<https://poly.google.com/view/49wQ9jGtT0A>

18. Scaun modelat de către Ryan Donaldson

<https://poly.google.com/view/3txPAhYeu-x>

19. Bibliotecă modelată de Nikki Morin

<https://poly.google.com/view/6p0f-u26DDh>

20. Carte modelată de Anastasiia Ku

https://poly.google.com/view/0-_UGETLPsX

21. Carte modelată de Justin Randall

https://poly.google.com/view/4WPcl72i1_S

22. Noptieră făcută de Hope Moore

<https://poly.google.com/view/55vDM2jR-5H>

23. Pion modelat de Jarlan Perez

<https://poly.google.com/view/0xRVhzhfseb3>

24. Cheie modelată de Dobromir Yordanov

<https://poly.google.com/view/9rQqZlcfGgs>

25. Font preluat de pe google fonts

<https://fonts.google.com/specimen/Open+Sans?selection.family=Open+Sans>

26. Cod inspirat pentru fixarea bug-ului de la scroll de la thinksquirrel_lily

<https://forum.unity.com/threads/access-scrollbar-value-not-possible-through-code.283486/>

27. Iconița pentru butonul de start joc

https://www.flaticon.com/free-icon/music-player-play_70409

28. Figura ilustrativă cu senzori pentru detectarea headset-ului VR

<https://www.howtogeek.com/246333/oculus-rift-vs.-htc-vive-which-vr-headset-is-right-for-you/>

A1.0. Descrierea aplicației pe scurt

Escape Room este o aplicație realizată utilizând motorul de dezvoltare de jocuri 2D și 3D intitulat Unity. Pentru partea de augmentare a realității m-am folosit de pachetul celor de la AR Foundation și ARCore XR Plugin pe care Unity le expune în partea de Package Manager. Pentru a putea lansa aplicația pe telefonul mobil m-am folosit de motorul de dezvoltare Android Studio pentru a descarca SDK-ul android, versiunea 8.1 Oreo pe care mai apoi l-am referențiat în Unity. Unele modele 3D au fost preluate de la creatori de conținut tridimensional și au fost menționați în secțiunea de Bibliografie, restul au fost modelate folosind obiectele deja prestabilite din Unity.

Aplicația Escape Room oferă spre deosebire de celelalte jocuri menționate mai sus, posibilitatea utilizatorului de a se plimba prin mediul virtual fără a utiliza o bandă de alergat VR sau stând pe loc și apăsând un buton pe controler. Puzzle-urile sunt de un nivel stimulant, hint-urile fiind de mare ajutor.

A1.1. Arhitectura aplicației

Aplicația este construită dintr-o scenă, ce conține mai multe obiecte sau pachete de obiecte. Meniul principal, meniul cu controalele și cel de ajutor reprezintă fiecare câte un pachet de obiecte. Ordinea obiectelor din meniuri în pachete contează la randarea scenei. Obiectul ce se află în ierarhie poziționat mai jos, va fi afișat pe ecran în fața obiectelor ce se află în ierarhie poziționate deasupra lui (Vezi Figurile 12 și 13 de mai jos).

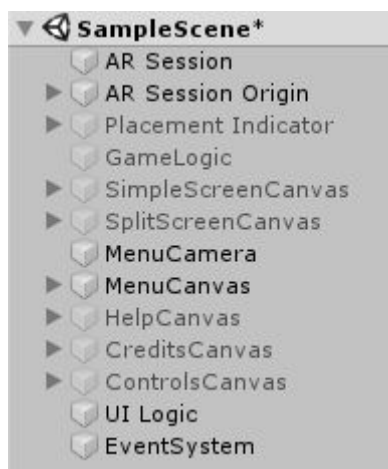


(Figura 12) Ierarhia meniului



(Figura 13) Meniul jocului

Toate meniurile conțin un GameObject de tipul Canvas. Acest obiect, specific Unity, este un wrapper (container) pentru elementele de UI (interfața utilizatorului): butoane, label-uri, input-uri etc.. Toată logica interfeței utilizatorului este menținută de către obiectul din scenă intitulat Ui Logic (vezi Figura 14 de mai jos). Obiectele Ar Session și Ar Session Origin folosesc scripturile pachetului AR Foundation și vor fi în permanență enabled (obiectul este activ și vizibil în scenă). Toată logica jocului este menținută de către obiectul GameLogic. Acesta va crea o cameră 3D în momentul în care utilizatorul va face tap (acțiunea de atingere a ecranului) pe ecranul device-ului cu indicatorul de plasare vizibil, va dezactiva meniul din scenă, indicatorul de plasare, înregistrarea de puncte pentru afișarea marcatorului de plasare a camerei și va activa un mod de afișare: SimpleScreenCanvas sau SplitScreenCanvas pe baza setărilor făcute în Meniu. EventSystem este un obiect ce vine odată cu utilizarea unui element din partea de UI. Acesta se ocupă cu manipularea input-ului, cu partea de raycasting și trimiterea de evenimente. MenuCamera este un obiect ce conține o componentă de tip camera din Unity. Acesta este și el dezactivat la începerea jocului efectiv, rolul lui fiind de a viziona doar meniul.



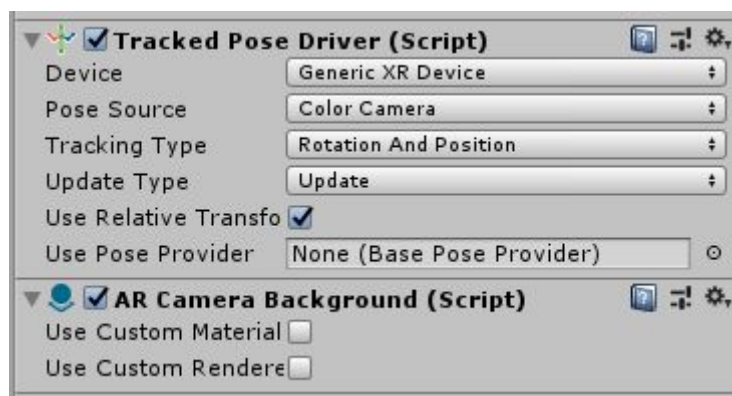
(Figura 14) Ierarhia scenei

A1.2. Implementarea aplicației

A1.2.a. Clasele AR Foundation

În ierarhia scenei, primul obiect conține doar un script pe el, din pachetul AR Foundation. Acest script are o clasă `ARSession` ce expune mai multe metode de lucru cu sesiunea augmentată. Obiectul `AR Session Origin` conține clasa `AR Session Origin` ce expune metode care facilitează lucrul cu mediul real și cu obiectul cameră din mediul virtual. O metodă utilă din această clasă este `Raycast`. Această metodă este capabilă să creeze un Ray localizând puncte din mediul real ce pot fi folosite ca și ancore pentru obiectele virtuale în mediul real.

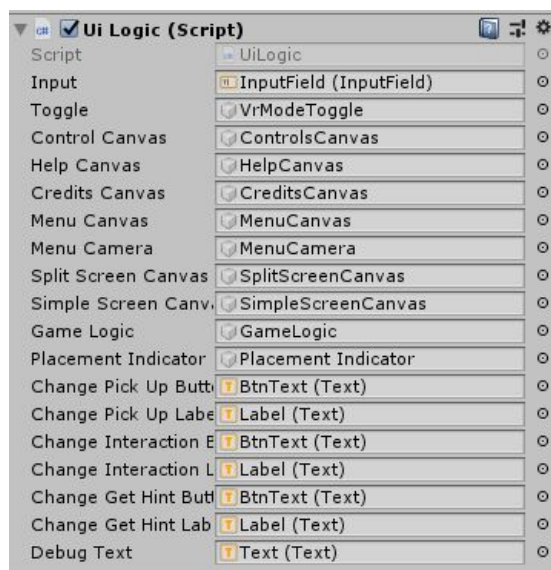
De asemenea, în această clasă se află o metodă numită `GetCameraOriginPose` ce returnează informații despre poziția device-ului și este utilizată în metoda `Update` din acest script, metodă specifică Unity ce este apelată frame cu frame. Cu ajutorul lui `GetCameraOriginPose`, poziția și rotația obiectului Camera din Unity este actualizată. Obiectul Camera, ce conține componenta Camera pe el, necesită încă două componente, mai exact două scripturi și anume: `Tracked Pose Driver` (vezi Figura 15 de mai jos pentru setările acestuia) și `AR Camera Background`, acesta expune diferite metode prin care înlocuiește feed-ul camerei virtuale cu feed-ul video de la camera device-ului.



(Figura 15) Scripturile camerei

A1.2.b. Obiectul Ui Logic și Meniurile

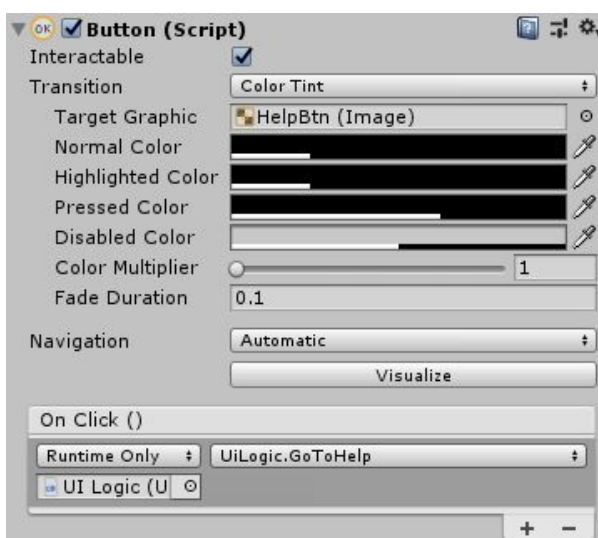
Obiectul Ui Logic este un Empty GameObject (obiect ce nu conține nici o altă componentă pe el în afară de Transform, acesta indicând poziția, rotația și dimensiunea în scenă) ce are atașat un script intitulat Ui Logic (vezi Figura 16 de mai jos) având clasa cu același nume, care are asignat toate referințele către obiectele din meniuri cât și cele din afara meniurilor cum ar fi: GameLogic, Placement Indicator etc.;



(Figura 16) Variabilele publice din scriptul Ui Logic

La startul acestui script este golit PlayerPrefs de chei și valori, este setat ecranul device-ului pentru a nu intra în modul sleep cât timp jocul rulează și este setat modul de afișare sub formă de cheie în dicționarul PlayerPrefs, pe baza obiectului de tip toggle din UI.

În această clasă am creat niște metode pentru a afișa meniul dorit de către utilizator, o metodă cu numele GoToGameScreen ce încarcă (activează obiectele din scenă ce se ocupă cu jocul propriu-zis și dezactivează ceea ce ține de meniu) și de asemenea niște metode ce se ocupă cu schimbarea controalelor. În metoda Update am expus logica pentru schimbarea controalelor întrucât în acest loc pot face listen pentru a vedea când un buton este apăsă pe controler de către utilizator. Pentru a explica mai concret cum se realizează trecerea de la un meniu la altul, voi alege meniul de ajutor ca și exemplu. În meniul principal avem un obiect de tipul buton cu textul „HELP” ce are setată o metodă pe evenimentul de OnClick (vezi Figura 17 de mai jos) care setează canvas-ul meniului de ajutor pe activ și cel al meniului principal pe inactiv (vezi Tabela 1 de mai jos).



(Figura 17) Butonul „HELP”

```
public void GoToHelp()
{
    helpCanvas.SetActive(true);
    menuCanvas.SetActive(false);
}
```

(Tabela 1) Metoda GoToHelp

Metoda utilizată pentru a schimba meniul principal cu acela de afișare a controalelor mai conține câteva lucruri în plus. Această metodă setează și textul pe butoanele de schimbare a controalelor pe „CHANGE” cât și label-urile fiecăruia cu numele KeyCode-urilor corespunzătoare butoanelor pe controler prestabilite. Odată afișat meniul controalelor, în acesta se regăsesc trei butoane pentru schimbarea fiecărui control în parte. Asemeni butonului de „HELP”, o metodă corespunzătoare este atașată. Fiecare metodă în parte setează label-ul corespunzător butonului de pe interfață ce este apăsă în „PRESS A BUTTON” și o variabilă pentru a ști în metoda de Update ce control urmează a fi modificat (vezi Tabela 2 de mai jos pentru un exemplu). Odată setată această variabilă, în metoda Update se verifică în fiecare frame în parte dacă unul din aproximativ 509 de KeyCode-uri este apăsă. Dacă un buton pe

controler este apăsat, ce are un KeyCode deja prestabilit pentru un alt control din joc, acestea se vor schimba între ele. Vechiul KeyCode al controlului ce se dorește a schimba va fi setat pentru controlul ce deține noul KeyCode ce trebuie înregistrat (vezi Tabela 3 de mai jos pentru un exemplu).

```
public void ChangePickUpButton()
{
    isChangePickUpButtonPressed = true;
    ChangePickUpButtonText.text = "PRESS A BUTTON";
}
```

(Tabela 2) Metoda schimbării butonului de pick up și drop de obiecte

```
public void Update()
{
    if (isChangePickUpButtonPressed)
    {
        foreach (KeyCode kcode in Enum.GetValues(typeof(KeyCode)))
        {
            if (Input.GetKeyDown(kcode))
            {
                if ((int)kcode == PlayerPrefs.GetInt(INTERACT_BTN, INTERACT_BTN_VALUE))
                {
                    PlayerPrefs.SetInt(INTERACT_BTN, PlayerPrefs.GetInt(PICK_UP_BTN, PICK_UP_BTN_VALUE));
                    ChangeInteractionLabelText.text = Enum.GetName(typeof(KeyCode), PlayerPrefs.GetInt(PICK_UP_BTN, PICK_UP_BTN_VALUE));
                }
                else
                if ((int)kcode == PlayerPrefs.GetInt(GET_HINT_BTN, GET_HINT_BTN_VALUE))
                {
                    PlayerPrefs.SetInt(GET_HINT_BTN, PlayerPrefs.GetInt(PICK_UP_BTN, PICK_UP_BTN_VALUE));
                    ChangeGetHintLabelText.text = Enum.GetName(typeof(KeyCode), PlayerPrefs.GetInt(PICK_UP_BTN, PICK_UP_BTN_VALUE));
                }
                isChangePickUpButtonPressed = false;
                ChangePickUpButtonText.text = "CHANGE";
                ChangePickUpLabelText.text = kcode.ToString();
                PlayerPrefs.SetInt(PICK_UP_BTN, (int)kcode);
                break;
            }
        }
    }
}
```

(Tabela 3) Listen pentru un nou control și setarea acestuia

Metoda implementată pentru a porni jocul este GoToGameScreen (vezi mai jos Tabela 4). În aceasta setez pentru început în dicționarul global o cheie cu numele „HintTime” cu valoarea introdusă de jucător sau cea prestabilită pentru a ști în timpul jocului la ce perioadă de timp utilizatorul poate cere un hint. De asemenea, sunt dezactivate din scenă meniul și camera pentru afișarea meniului. Pe baza input-ului de tip toggle cu label-ul Vr Mode din UI, setez în dicționarul global o altă cheie intitulată „isVrMode” pe care o preiau în această metodă și în funcție de care activez unul dintre cele două moduri de afișare implementate de mine ce vor fi explicate mai jos. Și nu în ultimul rând, activez indicatorul pentru a marca poziția unde camera virtuală poate fi plasată și obiectul gameLogic ce se va ocupa mai departe de logica aplicației.

```

public void GoToGameScreen()
{
    try
    {
        Int32.Parse(input.text);
        PlayerPrefs.SetInt("HintTime", Int32.Parse(input.text));
    }
    catch {}
    menuCamera.SetActive(false);
    menuCanvas.SetActive(false);
    if (PlayerPrefs.GetInt("isVrMode") == 1)
    {
        splitScreenCanvas.SetActive(true);
    }
    else
    {
        simpleScreenCanvas.SetActive(true);
    }
    placementIndicator.SetActive(true);
    gameLogic.SetActive(true);
}

```

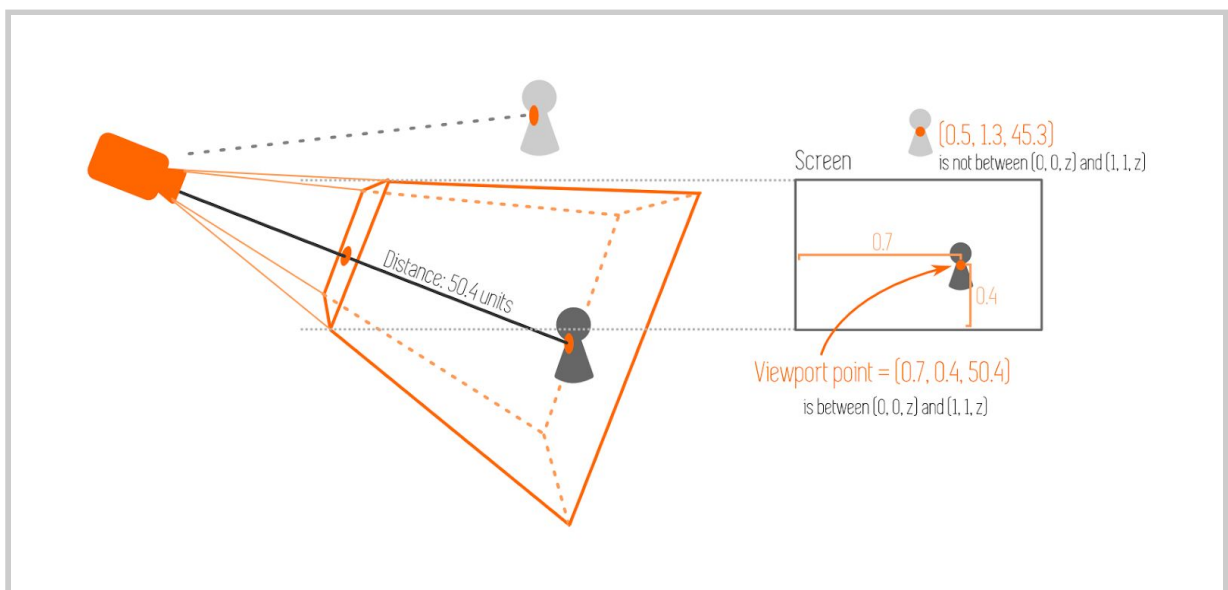
(Tabela 4) Metoda GoToGameScreen

A1.2.c. Obiectul GameLogic și Placement Indicator

GameObject-ul GameLogic este un empty game object ce conține atașat pe el două scripturi: Place Object (care este activ) și Interact With Objects With Controller (care este inactiv). Odată făcut activ acest obiect, scriptul Place Object își începe execuția, metoda Start fiind prima metodă apelată și o singură dată pe parcursul rulării acestui script. În această metodă se caută și se memorează într-o variabilă obiectul de tipul ARSessionOrigin din scenă. Frame cu frame este apelată metoda Update din acest script care cât timp nu avem escape room-ul plasat în scenă, utilizează alte metode create în interiorul scriptului pentru a plasa obiectul Placement Indicator într-o poziție validă ce este setată de metoda UpdatePlacementPose. Această metodă fiind una mai complexă o voi explica în ultima frază. Prin interacțiune cu device-ul prin „tap” pe ecran, indicatorul pentru plasarea camerei nu se va mai afișa și va fi creat și poziționat escape room-ul. De asemenea, scriptul „Interact With Objects With Controller” va fi setat ca și activ.

În prima variabilă, screenCenter preiau centrul ecranului transformând punctul din centrul Viewport-ului (vezi Figura 18 de mai jos), după care inițializez o listă de puncte de

tipul `ARRaycastHit`. Utilizez metoda `Raycast` din scriptul `ARSessionOrigin` pentru a detecta puncte de tip `Trackable`, ce vor fi stocate în lista de `ARRaycastHit`. În urma creării de `Raycast`-uri pornind din centrul ecranului, punctele ce sunt considerate ca făcând parte din același plan, sunt stocate în listă. Dacă un astfel de punct este găsit, setez faptul că poziția de plasare a camerei virtuale este validă, și actualizez această poziție și rotație în variabila `placementPose` (vezi Tabela 5 de mai jos). Odată ce poziția de plasare a camerei este una validă, afișarea indicatorului de plasare a camerei se face împreună cu actualizarea poziției și rotației acestuia în metoda `UpdatePlacementIndicator`.



(Figura 18) Viewport point to Screen point (sursa¹⁰)

```
private void UpdatePlacementPose()
{
    var screenCenter = Camera.current.ViewportToScreenPoint(new Vector3(0.5f, 0.5f));
    var raycastHits = new List<ARRaycastHit>();
    arOrigin.Raycast(screenCenter, raycastHits, TrackableType.Planes);

    placementPoseIsValid = raycastHits.Count > 0;
    if (placementPoseIsValid)
    {
        placementPose = raycastHits[0].pose;

        var cameraForward = Camera.current.transform.forward;
        var cameraBearing = new Vector3(cameraForward.x, 0, cameraForward.z).normalized;
        placementPose.rotation = Quaternion.LookRotation(cameraBearing);
    }
}
```

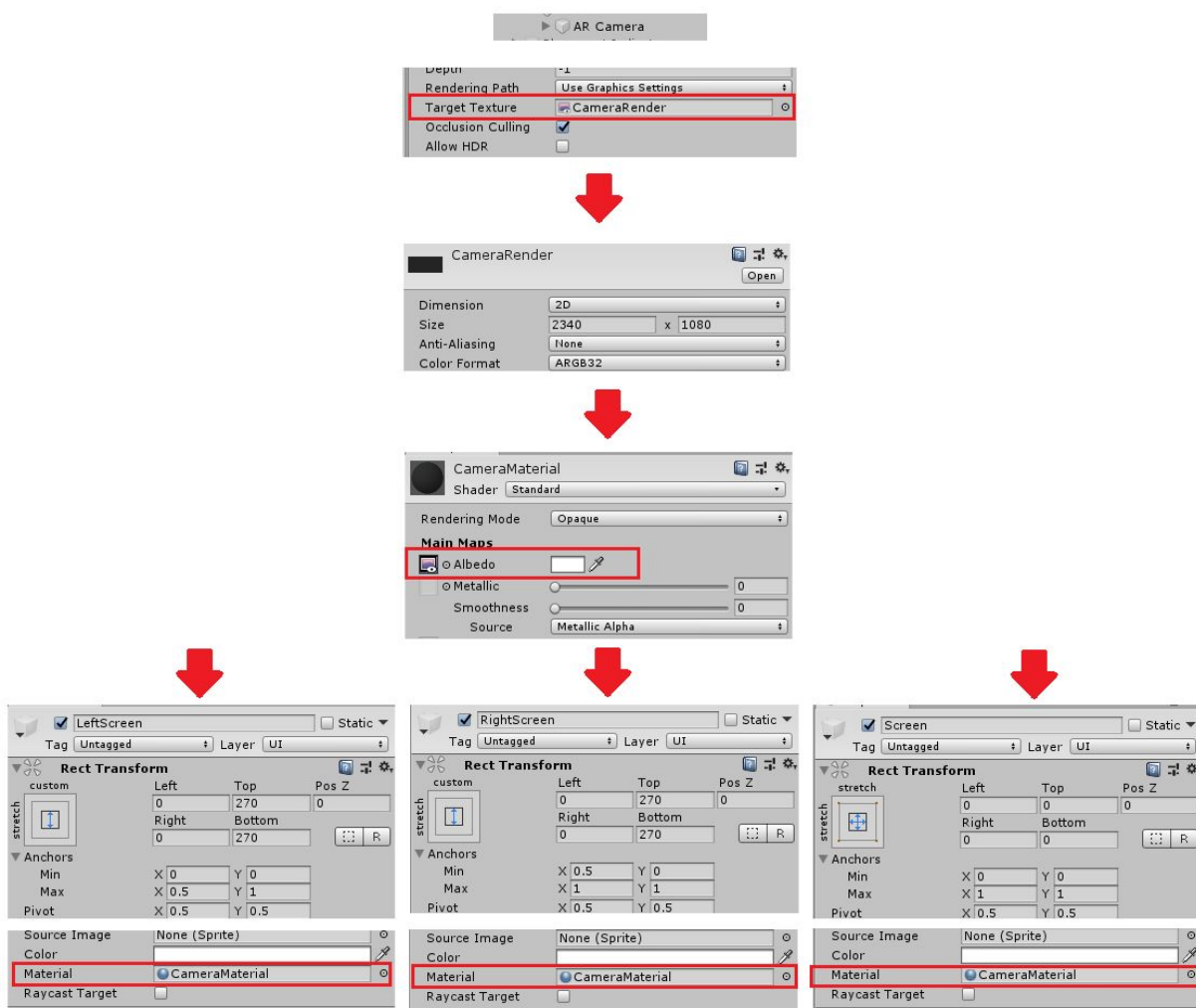
10

http://25games.net/camera-tracking-two-objects-camera-unity/?fbclid=IwAR1I781eVSPRXLTd_abgf7THfIShwCwSC1IVAHxApegu9Vq7RT_Ia5ck3f8

(Tabela 5) Metoda UpdatePlacementPose

A1.2.d. SimpleScreenCanvas și SplitScreenCanvas

SimpleScreenCanvas și SplitScreenCanvas sunt două obiecte de tipul Canvas din Ui. SimpleScreenCanvas are ca fiu un obiect de tipul Image în timp ce SplitScreenCanvas are doi fii cu două obiecte de acest tip ce au asignat ca și material, un material numit CameraMaterial. Camera device-ului, în cazul nostru ARCamera, stochează feed-ul într-o textură. Această textură este transpusă pe un material numit CameraMaterial ce este mai apoi asignat obiectelor de tip Image din aceste două canvas-uri. Acest lucru l-am realizat ca și optimizare cu scopul de a nu utiliza mai multe camere în realizarea modului stereoscopic pentru a folosi telefonul împreună cu un headset VR (vezi Figura 19 de mai jos).



(Figura 19) Modul de randare a jocului

A1.2.e. Scriptul „Interact With Objects With Controller”

Odată activat acest script, sunt setate în metoda de Start toate variabilele necesare puzzle-urilor, fiind căutate în scenă după tag. Obiectului ARCamera i-am atașat ca și fiu un empty game object pe post de ancoră pentru obiectele pe care jucătorul le ia cu ajutorul controlerului. În metoda Update, se face listen pentru toate butoanele setate în meniul de controale. De fiecare dată când un buton este apăsă, se face un raycast din centrul ecranului pentru a verifica dacă utilizatorul dorește să interacționeze cu un anumit tip de obiect precum: ușa, sertarul, butoanele de la panoul ușii, cryptex-ul etc., sau dacă butonul de pick up/ drop este apăsă, în cazul în care jucătorul nu are deja un obiect ancorat, se verifică dacă obiectul lovit de raycast are tagul „Interactable”, pentru a-l ancora. Dacă deține deja un obiect, acesta va fi dezancorat. Ancorarea unui obiect este făcută în modul următor (vezi Tabela 6 de mai jos). Se memorează într-o variabilă de tipul GameObject, obiectul lovit de raycast, îi este setată proprietatea isKinematic din componenta Rigidbody pe true, îi este setată poziția obiectului ca fiind poziția ancorei și de asemenea, ca și părinte, ancora îi va fi setată. Scriptul este mult mai diversificat însă cu explicațiile de până acum, înțelegerea lui nu este dificilă.

```
if (hit.transform.gameObject.tag == "Interactable" || hit.transform.gameObject.tag == "DrawerKey")
{
    isPickedUp = true;
    objectPicked = hit.transform.gameObject;
    objectPicked.GetComponent<Rigidbody>().isKinematic = true;
    objectPicked.transform.position = Anchor.transform.position;
    objectPicked.transform.parent = Anchor.transform;
}
```

(Tabela 6) Ancorarea unui obiect de tipul Interactable

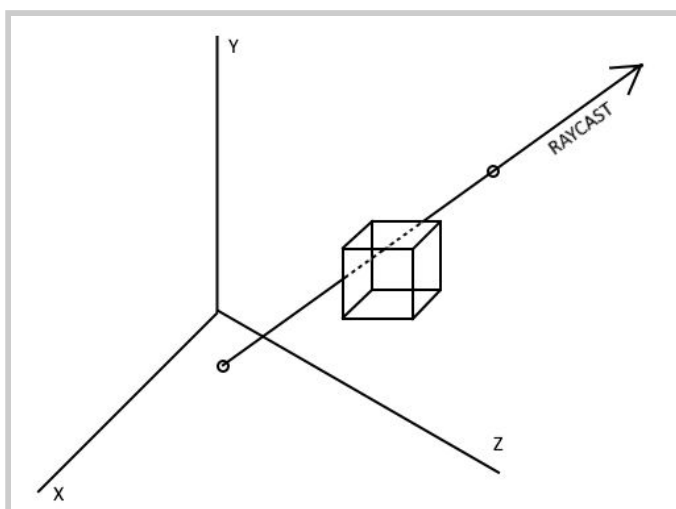
Termeni, noțiuni și device-uri

A2.1. Raycast

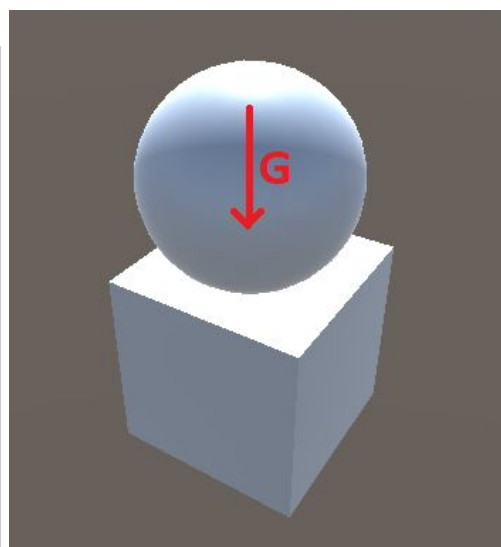
O rază este ca o linie ce pornește dintr-un punct și se continuă într-o direcție dată. Ideea unui raycast este de a urmări mai apoi această linie și de a vedea dacă se întâmplă o coliziune pe parcursul acesteia (vezi Figura 20 de mai jos).

A2.2. Coliziune

Coliziunea între două obiecte din Unity are loc atunci când cele două obiecte dețin amândouă câte o componentă de tipul Collider. Când un collider încearcă să pătrundă în interiorul altui collider, atunci se detectează o coliziune. Dacă unul din obiecte este deținător de un Rigidbody, motorul de dezvoltare va aplica fizici pe acesta, cum ar fi gravitația. În momentul în care un obiect încearcă să pătrundă în interiorul altuia, ambele fiind posesoare de collider, acesta nu va putea (vezi Figura 21 de mai jos).



(Figura 20) Ilustrație raycast



(Figura 21) Obiecte suprapuse cu collider

A2.3. KeyCode

Unity expune o enumerație de KeyCode-uri, fiecare reprezentând un întreg ca și valoare, iar cheia este dată de denumirea controlului respectiv. De exemplu, pentru KeyCode-ul cu valoarea 13 este cheia „Return”. Parcurgând această listă de coduri se poate detecta ce buton este apăsat într-un anumit moment.

A2.4. Oculus Rift și Vive

Oculus Rift este un headset de realitate virtuală lansată de cei de la Oculus VR în 2016. Aceasta este prima cască din tipul Rift creată de ei și a fost nișa companiei. Cu timpul au venit cu îmbunătățiri și au lansat Oculus Rift S, succesorul acestuia. Un astfel de device se conectează la un calculator, el fiind posesor de două lentile convergente și de două ecrane, fiecare dintre ele făcând randare pentru fiecare ochi. De asemenea vine cu un magnetometru, giroscop și un accelerometru în interior, combinate pentru a reda cu acuratețe tracking-ul într-o lume tridimensională (vezi Figura 22 de mai jos).

Vive este o companie ce crează headset-uri de realitate virtuală, printre care și headset-ul Vive, sau Vive VR cum i se mai spune. Aceleași principii ca mai sus sunt aplicate și aici. În ultima vreme headset-urile au devenit tot mai performante. Au camere montate pentru monitorizarea mișcării pupilelor, camere externe device-ului pentru tracking-ul mâinilor sau a mediului înconjurător.



(Figura 22) Interiorul unui headset (sursa¹¹)

11

https://www.wareable.com/vr/how-oculus-rift-works?fbclid=IwAR3P_9eXQkl4-EUIZPGZg8vJQEYPlcsGQLVb5Ri0mmSbYz2voZSI621CA7s

Tehnologii utilizate

A3.1. Unity

A3.1.a. Motorul de dezvoltare

Am ales Unity ca și motor de dezvoltare de jocuri întru-cât l-am considerat cea mai bună opțiune dintre toate aplicațiile ce pot dezvolta jocuri de realitate augmentată sau virtuală. Fiind destul de popular, la multe probleme întâmpinate am putut să găsesc deja o rezolvare în documentația celor de la Unity, sau la alți programatori ce le-au mai întâmpinat. Acest software a apărut odată cu mult cunoscuta platformă de conținut Youtube, având la bază trei tipuri de limbaje de programare suportate în versiunile mai vechi însă cu versiunea din 2017 numai C# este cel ce are mentenanță (engine-ul fiind dezvoltat cu C++ și C#).

Această platformă este expusă utilizatorilor gratuit cu o condiție, aceea de a nu depăși anual din jocurile dezvoltate un venit mai mare de 100.000\$. Programul a fost conceput pentru crearea jocurilor 3D, cu timpul ajungând să devină un motor de dezvoltare și pentru jocuri 2D, aplicații de realitate augmentată, de realitate virtuală, virtualitate augmentată, aplicații mobile etc.; Multitudinea platformelor pentru care acest software este dispus să creeze aplicații fac din acesta un competitor desăvârșit în lista de game engine-uri disponibile creatorilor de jocuri.

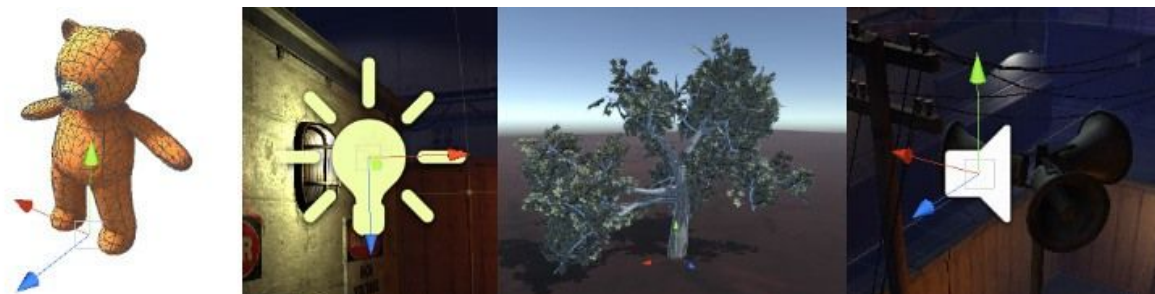
A3.1.b. Setul de pași logici în dezvoltarea oricărui tip de joc

(conform articolului de pe pagina celor de la Unity)

GameObject-urile

Orice conținut din scenă în Unity este un GameObject. Orice obiect în joc este un GameObject: caracterele, luminile, efectele speciale, proprietățile, totul (vezi Figura 23 de

mai jos). GameObject-urile nu pot face nimic de unele singure. Pentru ca să aibă un rol în joc, acestea trebuie să aibă niște proprietăți, pe care le putem adăuga prin adăugarea de componente.



(Figura 23) Tipuri de GameObject-uri în Unity (sursa¹²)

Componente

Componentele definesc și controlează comportamentul GameObject-urilor de care acestea sunt atașate. Un exemplu simplu este crearea unei lumini în scenă. Pentru a face acest lucru, unui GameObject (care neavând nimic atașat se mai numește Empty GameObject) îi atașăm o componentă „Light” (vezi Figura 24 de mai jos). O componentă ce se regăsește pe orice GameObject este componenta Transform. Aceasta dictează obiectului poziția, rotația și dimensiunea acestuia în scenă. Această componentă setează toate aceste atribute GameObject-ului de care este atașat relativ la părintele acestuia în meniul de ierarhie.

¹²

https://unity3d.com/programming-in-unity?fbclid=IwAR20_1Y2z4DdFhj2MAWfl3YLtyk8cE7Vvr36U4cMs5fjDn0fQ9krSUNdNaQ



(Figura 24) Componenta Light atașată unui GameObject (sursa¹³)

Variable

Componentele au nenumărate proprietăți ce pot fi editate, acestea pot fi modificate în fereastra Inspector, în editor sau din interiorul scripturilor. În exemplul de mai sus, unele proprietăți ai componentei Light sunt raza, culoarea și intensitatea.

A3.2. Pachetul AR Foundation

AR Foundation este un framework specific creat pentru dezvoltatorii de aplicații de realitate augmentată. Oferă posibilitatea de a crea aplicația o singură dată și de a o lansa pe device-urile ce suportă AR Kit și AR Core. De asemenea, include funcționalități adiționale pentru a facilita lucrul cu ancorarea obiectelor virtuale în mediul real, menținerea acestora și multe altele. Având în principal lucru cu device-urile mobile, aceasta are o conversie a randării elementelor din HD în lightweight pentru a optimiza partea de randare. Cu ajutorul

¹³

https://unity3d.com/programming-in-unity?fbclid=IwAR20_1Y2z4DdFhj2MAWfi3YLtyk8cE7Vvr36U4cMs5fjDn0fQ9krSUNdNaQ

acestui pachet, dezvoltatorii pot prelua modelele 3D pe care le-au folosit în dezvoltarea jocurilor normale și să le augmenteze în lumea reală, să creeze un joc augmentat care arată bine, rulează cu ușurință și este disponibil la peste 1 miliard de device-uri mobile care permit lucrul cu aplicații de realitate augmentată. Unity expune pachetul celor de la AR Foundation direct în Package Manager pentru a fi ușor de importat.

Am ales acest framework deoarece este ușor de integrat cu Unity, lucrul cu device-urile de tip android fiind punctul lui forte. Posibilitatea de a face tracking pe orice tip de teren plat este cireașa de pe tort în aplicația mea pe care acest pachet mi-o oferă.

A3.3. Android Studio

Am avut nevoie de Android Studio pentru a prelua SDK-ul corespunzător versiunii de android a telefonului. Folosind un Huawei P20 Mate Lite cu android-ul 8.1 Oreo, am preluat din Android Studio SDK-ul corespunzător și l-am pus în Unity pentru ca acesta să știe mai departe, când build-ul este făcut, tot ceea ce are el nevoie.

A3.4. Paint.net

Acest program de desenare/editare de imagini mi-a fost de ajutor în crearea texturilor transparente. Pe lângă Paint normal de la windows, acesta vine cu multe alte beneficii. Cum ar fi diferite funcții ce modifică aspectul imaginii, posibilitatea de a modifica contrastul, saturația imaginii etc. Am utilizat Paint.net la crearea imaginilor ce se afișează pe monitorul din încăpere, pentru crearea hărții din cameră, crearea puzzle-urilor.

Manual de utilizare

A4.1. Meniul principal

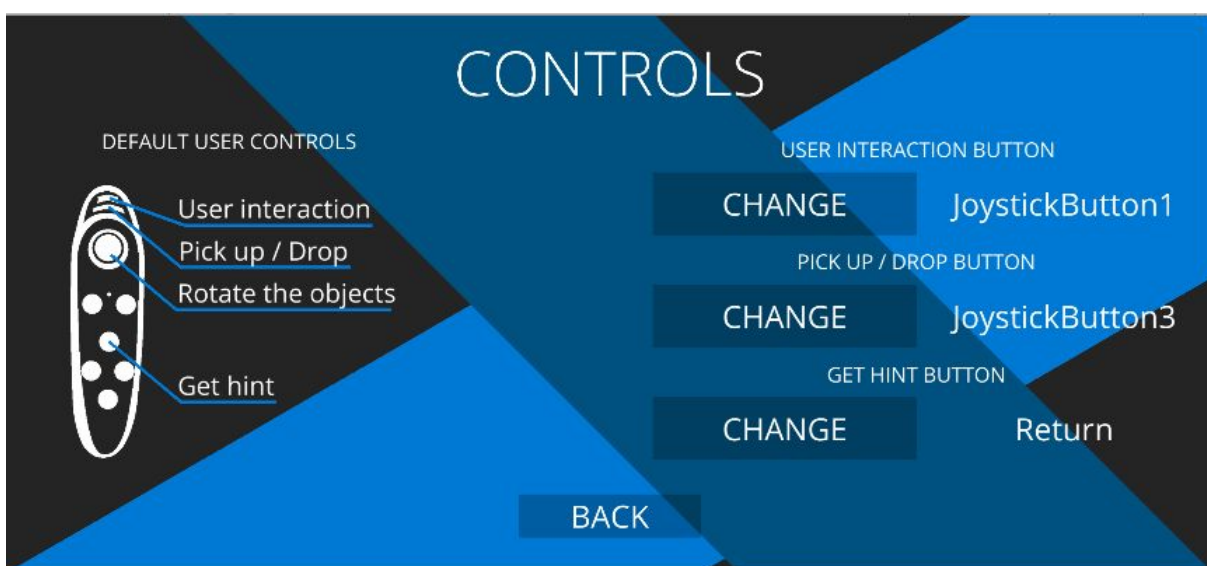
Meniul este unul simplist, interfața fiind una foarte comună zilelor noastre (vezi Figura 13 de mai sus). În partea din stânga sus, este poziționat un buton de tip toggle, pentru a selecta dacă afișarea să fie de tipul stereoscopic pentru un headset VR sau să fie afișarea normală, pe tot ecranul telefonului.

În partea dreaptă sus se regăsește un input ce setează numărul de secunde la care jucătorul poate să ceară un indiciu folosind butonul corespunzător controlului ce face acest lucru. În centrul ecranului se regăsește mare și simplist butonul de start joc. Odată apăsă, interfața utilizatorului dispare și acesta începe să scaneze împrejurimile pentru a detecta un spațiu plan, unde ulterior camera virtuală va fi plasată. În partea de jos a ecranului patru butoane secundare, de aici și dimensiunea lor mai mică. Primul buton de la stânga spre dreapta este butonul ce redirecționează jucătorul către meniul de control. Următorul buton este cel de HELP, acesta redirecționează jucătorul către meniul de ajutor. Al treilea buton este cel de CREDITS. Prin apăsarea lui jucătorul este redirecționat la meniul ce conține persoanele de la care unele resurse au fost preluate în crearea acestei aplicații. Ultimul buton este cel pentru închiderea jocului.

A4.2. Meniurile secundare

Meniul de ajutor, de creditare și cel de controale sunt toate meniuri secundare și conțin un butonul BACK pentru a reveni la meniul principal. În interiorul meniului de control (vezi Figura 25 de mai jos), este afișat o imagine cu butoanele de control setate implicit pentru un joystick Esperanza BT VR EMV101. Dacă utilizatorul dorește să schimbe controalele sau să își configureze propriul controler, acest lucru se realizează foarte ușor. Jucătorul trebuie să atingă butonul de CHANGE al controlului pe care dorește să îl schimbe și să apese butonul

dorit de acesta pe controlerul conectat de el. Dacă jucătorul dorește să știe cum se joacă acest joc, acesta poate naviga către meniul de ajutor unde se regăsesc explicațiile necesare.



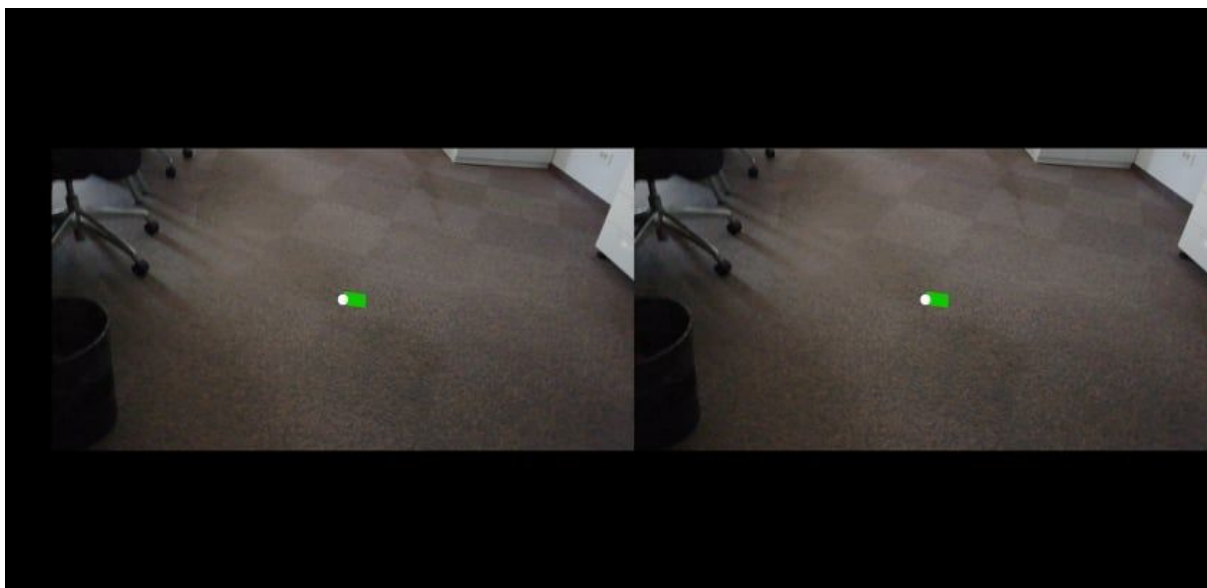
(Figura 25) Meniul de control

A4.3. Prima interacțiune cu jocul

A4.3.a. Pregătirea mediului

Primul pas pentru a juca acest joc este ca să vă conectați un controler la telefon. Odată conectat și ochelarii VR sunt pregătiți, este recomandat pentru a merge în meniul de control și a vă configura controlerul (dacă acesta diferă de unul „Esperanza BT VR EMV101”). Asigurați-vă că ecranul dispozitivului are luminozitatea ajustată astfel încât la utilizarea împreună cu headset-ul, să nu vă doară ochii sau să vă chinuiți să vedeți ceva. Odată ce totul este în regulă, ne putem întoarce la meniul principal și să apăsăm butonul de start joc. În acest moment, veți vedea exact ceea ce vede camera telefonului (vezi Figura 26 de mai jos). Vă puteți introduce acum telefonul în casca de realitate virtuală și vă puteți face ajustările necesare astfel încât să vedeți clar. De îndată ce toți acești pași sunt îndepliniți, începeți să scanați zona prin a vă uita cât mai mult la sol și a vă plimba în zona în care doriți să proiectați camera virtuală. Pe toată perioada jocului evitați să acoperiți camera telefonului pentru a nu perturba tracking-ul. Cu cât scanați mai mult, cu atât tracking-ul pe parcursul jocului va fi mai bun. De asemenea, un teren cât mai diversificat în modele, cu textura cât mai variată ajută tracking-ul. Pentru o scanare cât mai bună asigurați-vă că mediul în care vă aflați este bine

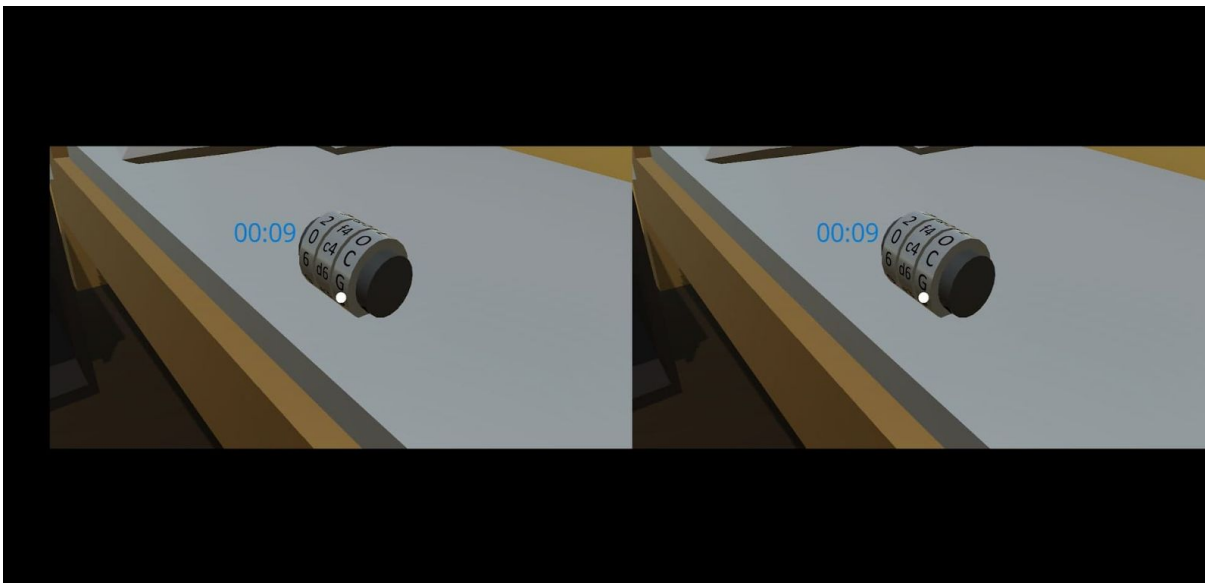
iluminat. Ca să nu riscați să vă loviți de obiecte pe parcursul jocului, mergeți într-un loc spațios, fără obiecte pe un perimetru de aproximativ 3 metri, de care v-ați putea accidenta. Un loc cât mai spațios va împiedica acoperirea camerei de către pereții înconjurători sau de către obiectele mari din împrejurime. Când vă simțiți pregătiți, atingeți ecranul dispozitivului, sau dacă acesta este introdus în headset și nu aveți acces la ecran, folosiți butonul de interacțiune pentru a proiecta camera.



(Figura 26) Scanarea mediului înconjurător

A4.3.b. Jocul propriu-zis

Camera virtuală este proiectată. Uită-te în jurul tău, ești închis într-o încăpăre. Dacă vei încerca să ieși din aceasta prin zid, vei constata că tot ce vezi este negru. Acesta este acel Skybox de care am menționat pe la începutul acestui document. Dacă te vei roti 180 de grade, vei observa că peretele camerei este invizibil, pentru ca să poți să te reîntorci cu ușurință înapoi. Nu ai idee cum să începi, de unde să începi? Apasă butonul de HINT de pe controler pentru a primi un indiciu. Acesta ar trebui să fie îndeajuns de sugestiv ca să știi ce ai de făcut. Luați obiectele de prin cameră poziționând gaze-ul pe acestea (vezi Figura 27 de mai jos) și folosind butonul de Pick Up / Drop și rotiți-le utilizând maneta de pe joystick.



(Figura 27) Pick Up / Drop de obiecte

Pentru a roti roțițele de pe cryptex, apasă butonul de interacțiune după ce ai luat cryptex-ul „în mână” pentru a interacționa mai departe cu el. Folosește maneta de pe controler stânga - dreapta pentru a schimba roțița pe care dorești să o controlezi și axa sus - jos pentru a o roti. Dacă consideri că ai pus codul corect între acele două săgeți marcatoare de pe cryptex ce indică locul unde ar trebui cheile să fie aliniate, apasă din nou butonul de interacțiune pentru a deschide cryptex-ul. Dacă nu se întâmplă nimic cu acesta, atunci codul pus nu este cel bun, sau nu este aliniat corespunzător. Pentru a reveni la a roti întregul cryptex, folosește iar butonul de interacțiune, acesta face trecerea înainte și înapoi în interacțiunea cu acesta.

Pentru a plasa o pioneză în tabloul cu harta lumii, se apasă butonul de interacțiune având pioneză „în mână” și gaze-ul (punctul alb din centrul ecranului) poziționat în locul în care se dorește plasarea acesteia.

Pentru a scrie un cod pe cifrul de la ușă, îndreptați gaze-ul către butonul cu cifra dorită și apăsați butonul de interacțiune (vezi Figura 28 de mai jos). Odată introduse patru cifre, după puțin timp codul se șterge. Dacă codul introdus este cel corect, atunci cele patru cifre vor fi înlocuite de litera X. Dacă veți deschide ușa prin interacționarea cu aceasta, veți vedea că ați evadat din „lumea virtuală” pentru că puteți vedea lucrurile din lumea reală.



(Figura 28) Interacțiune cu cifrul ușii