



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE**

Rezolvarea unui puzzle folosind Realitatea Augmentată

LUCRARE DE LICENȚĂ

Absolvent: **Cătălin Cuciurean**

Coordonator științific: **Conf. Dr. Ing. Teodor-Traian Ștefănuț**

2020



FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

DECAN,
Prof. dr. ing. Liviu MICLEA

DIRECTOR DEPARTAMENT,
Prof. dr. ing. Rodica POTOLEA

Absolvent: **Cătălin Cuciurean**

Rezolvarea unui puzzle folosind Realitatea Augmentata

1. **Enunțul temei:** *Proiectarea unei aplicații Android, folosind realitatea augmentată având ca scop indicarea utilizatorului în vederea rezolvării unui puzzle.*
2. **Conținutul lucrării:** *Introducere, Obiectivele Proiectului, Studiu Bibliografic, Analiză și Fundamentare Teoretică, Proiectare de Detaliu și Implementare, Testare și Validare, Manual de Instalare și Utilizare, Concluzii, Bibliografie.*
3. **Locul documentării:** Universitatea Tehnică din Cluj-Napoca, Departamentul Calculatoare
4. **Consultanți:**
5. **Data emiterii temei:** 1 noiembrie 2019
6. **Data predării:** 8 septembrie 2020

Absolvent: **Cătălin Cuciurean**

Coordonator științific: Conf. Dr. Ing. Teodor-Traian
Ștefănuț

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE****Declarație pe proprie răspundere privind
autenticitatea lucrării de licență**

Subsemnatul(a) Cuciurean Cătălin, legitimat(ă) cu CI seria XV nr. 371087, CNP 1970615336671, autorul lucrării Rezolvarea unui puzzle folosind realitatea augmentată elaborată în vederea susținerii examenului de finalizare a studiilor de licență la Facultatea de Automatică și Calculatoare, Specializarea Calculatoare din cadrul Universității Tehnice din Cluj-Napoca, sesiunea septembrie a anului universitar 2020, declar pe proprie răspundere, că această lucrare este rezultatul propriei activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate, în textul lucrării, și în bibliografie.

Declar, că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de licență*.

Data

8 septembrie 2020

Nume, Prenume

Cuciurean Cătălin

Semnătura

Cuprins

Capitolul 1. Introducere – Contextul proiectului	1
1.1. Evoluția telefoanelor inteligente și al aplicațiilor mobile	1
1.2. Contextul Proiectului	3
1.3. Motivația	3
Capitolul 2. Obiectivele Proiectului	5
2.1. Tema proiectului	5
2.2. Cerințele funcționale	5
2.3. Cerințe non-funcționale	7
Capitolul 3. Studiu Bibliografic.....	10
3.1. Aplicații similare	10
3.1.1. Your AR Puzzle	10
3.1.2. AMON	10
3.1.3. Ravensburger Puzzle	11
3.1.4. PuzzlAR.....	12
3.1.5. 4D Puzzles	12
3.1.6. ARise	13
3.2. Diferențe între aplicația propusă și cele similare.....	14
Capitolul 4. Analiză și Fundamentare Teoretică.....	15
4.1. Cazuri de utilizare.....	15
4.1.1. Puzzle predefinit	15
4.1.2. Definirea unui puzzle.....	17
4.1.3. Cerințe speciale.....	19
4.2. Tehnologii.....	19
4.2.1. Realitatea Augmentată.....	19
4.2.2. C#.....	21
4.2.3. Vuforia Engine	21
4.2.4. Vuforia SDK.....	23
4.2.5. Vuforia și baza de date	24
4.2.6. Unity Engine.....	25
4.2.7. Microsoft Visual Studio.....	26
4.2.8. Android.....	27
4.2.9. Android Studio	29

4.2.10. JSON.....	29
4.2.11. Fire Base	30
4.2.12. Firebase Autentification	31
4.2.13. API.....	31
Capitolul 5. Proiectare de Detaliu și Implementare	33
5.1. Arhitectura sistemului.....	33
5.1.1. Diagrame de secvențe	34
5.2. Baza de date	36
5.2.1. Baza de date locală	36
5.2.2. Baza de date cloud	37
5.3. Structura proiectului	38
5.3.1. Structura de dosare	38
5.3.2. Structura scenei.....	38
5.4. Diagrama de pachete.....	39
5.4.1. Pachetul Vuforia	39
5.4.2. Pachetul SingIn.....	40
5.5. Funcționalități	41
5.5.1. Autentificare	41
5.5.2. Recunoaștere imagini	41
5.5.3. Conexiune piese.....	42
5.5.4. Legătură piese.....	42
5.5.5. Poziție piese.....	43
5.5.6. Captură ecran	44
5.5.7. Încărcare imagini	44
5.5.8. Recunoaștere imagini din baza de date cloud.....	45
5.5.9. Finalizare puzzle.....	45
Capitolul 6. Testare și Validare.....	46
6.1. Testare funcționalități	46
6.1.1. Recunoaștere imagini	46
6.1.2. Verificare conexiune.....	46
6.1.3. Verificare poziție	47
6.1.4. Verificare linie între piese	48
6.1.5. Realizare captură ecran.....	48
6.1.6. Încărcare imagine în baza de date.....	49

6.1.7. Finalizare puzzle	49
Capitolul 7. Manual de Instalare și Utilizare	51
7.1. Instalare aplicație	51
7.2. Manual de utilizare	52
7.2.1. Permisune cameră	52
7.2.2. Autentificare	52
7.2.3. Meniu Principal	53
7.2.4. Puzzle Predefinit	54
7.2.5. Creare Puzzle	54
7.2.6. Puzzle Definit	55
7.2.7. Informații	55
Capitolul 8. Concluzii	56
8.1. Concluzie	56
8.2. Dezvoltări ulterioare	56
8.2.1. Ochelari inteligenți	57
8.2.2. Îmbunătățire funcție de definire puzzle	57
8.2.3. Crearea unei baze de date proprii	57
8.2.4. Posibilitate recunoaștere puzzle-uri de mărimi mai mari	57
Bibliografie	58

Capitolul 1. Introducere – Contextul proiectului

1.1. Evoluția telefoanelor inteligente și al aplicațiilor mobile

În ultimul deceniu tehnologia în domeniul telefoanelor mobile a avansat considerabil, astfel telefonul inteligent (smartphone), a devenit un obiect esențial și de care nu ne putem lipsi. Datorită acestui fapt, modul în care oamenii comunică, lucrează și socializează, a fost puternic influențat, schimbând radical modul de viață și de desfășurare al activităților.

Telefonul inteligent este la baza un telefon mobil, dar care conține mult mai multe funcții decât acesta și o putere de procesare superioară. A devenit un dispozitiv multifuncțional, care a înlocuit necesitatea de a avea mai multe dispozitive diferite, pentru fiecare activitate în parte, de exemplu: nu mai este necesar un calculator de buzunar, nu mai este necesară achiziția unei camere de fotografiat, deoarece camerele de pe telefoanele inteligente au devenit foarte capabile și performante în ultimii ani.

Una dintre cele mai importante caracteristici ale telefoanelor inteligente este conexiunea acestora la internet, care a devenit mult mai rapidă în ultimii ani ajungând la o conexiune 5G, ce poate atinge viteze de până la 20 gigabiți pe secundă. Datorită acestui fapt, comunicarea și transferul de date între utilizatori a devenit foarte rapidă.

Într-un telefon inteligent se pot găsi o multitudine de senzori integrați. Senzorii standard cu care vine un telefon sunt: un accelerometru, un senzor de poziție sau giroscop, un senzor de proximitate, un senzor magnetic, respectiv o busolă electronică și un senzor de recunoaștere a lumii, dar în funcție de model pot avea mai mulți senzori.

Toate aceste funcționalități sunt susținute de un hardware puternic și avansat. Telefonul inteligent a ajuns un calculator de buzunar, performant cu ajutorul căruia se pot procesa și transfera datele foarte rapid, astfel hardware-ul nu mai este un impediment. Având la dispoziție toată această tehnologie, un telefon inteligent poate să înlocuiască un calculator (desktop) standard, ce este folosit pentru simple activități cum ar fi: verificare mail-uri, navigare pe internet și crearea/modificarea diferitelor fișiere și al transferului acestora.

Datorită acestui avans, smartphone-urile au devenit din ce în ce mai populare, astfel că la nivel global în anul 2020, sunt aproximativ 3,50 miliarde de utilizatori de smartphone-uri, iar 44,85% din oameni dețin un astfel de dispozitiv, fiind o creștere considerabilă față de anul 2016, când erau doar aproximativ 2,5 miliarde de utilizatori. Se estimează faptul că până în anul 2021 se va ajunge undeva la 3,8 miliarde de utilizatori de telefoane inteligente [29].

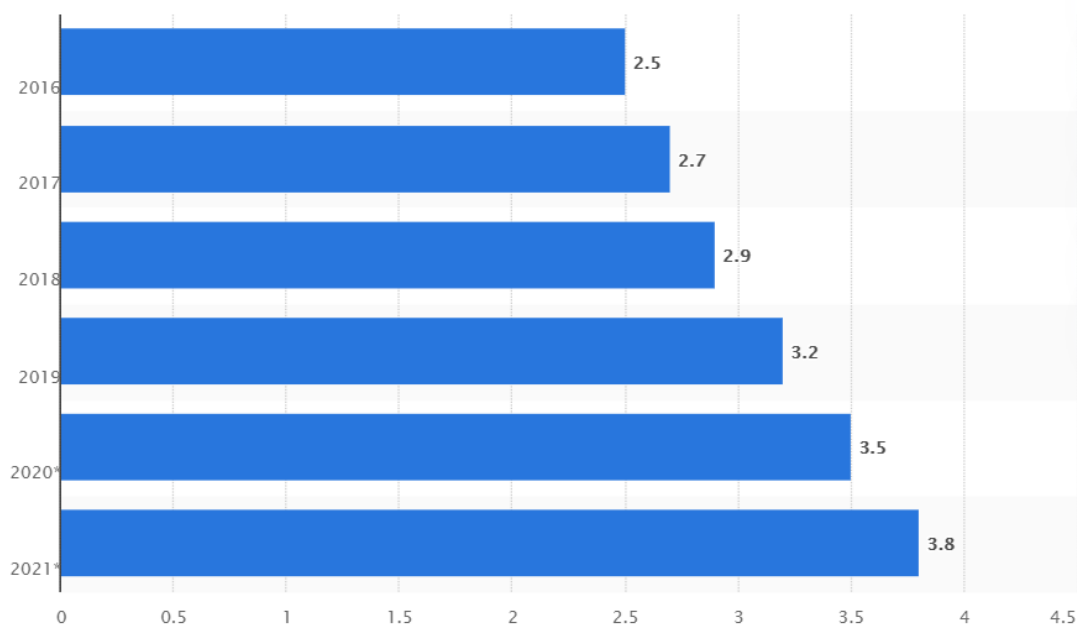


Figura 1.1 – Numărul de utilizator de telefoane inteligente la nivel global [29]

Odată cu avansul acestor dispozitive, și aplicațiile ce rulează pe ele au avut parte de o schimbare majoră și de o evoluție rapidă. Datorită numărului mare de utilizatori, industria aplicațiilor mobile este în plină ascensiune, astfel în medie oamenii își petrec 90% din timpul folosirii unui telefon în diferite aplicații, fie acestea de socializare, sau jocuri. Numărul mare de aplicații dezvoltate pentru dispozitivele mobile, este un rezultat al nevoilor oamenilor de a avea cât mai multe funcționalități la îndemână, într-un singur loc compact și mobil. Având acces la astfel de dispozitive inteligente, cu o gamă foarte diversificată de aplicații, în diferite domenii, oamenii își pot rezolva problemele sau executa activitățile zilnice mult mai ușor și mai rapid, astfel aplicațiile au devenit o parte importantă a vieții cotidiene. De exemplu, putem folosi aplicații specializate pentru a efectua plăți ale facturilor, aplicații pentru cumpărături, dar și transmiterea de informații se poate face mult mai ușor. În ziua de azi, putem efectua aproape orice operațiune de pe un dispozitiv inteligent ce este conectat la o rețea de internet.

Principalele surse de descărcări ale aplicațiilor sunt Google Play și App Store. Google Play aparține de dispozitivele ce folosesc ca și sistem de operare Android, iar App Store aparține de cele ce folosesc iOS. Aproximativ 74% din utilizatori dețin un smartphone cu Android, iar 24% unul cu iOS. Aceste valori sunt influențate și datorită faptului că, politicile celor de la App Store sunt mai stricte, acesta conține aproximativ 1,85 milioane de aplicații, pe când Google Play ajunge la 2,56 de milioane [30].

Android, este un sistem de operare ce se găsește pe majoritatea dispozitivelor și telefoanelor mobile. Acesta este bazat pe nucleul Linux și a fost dezvoltat la început de către compania Google. Deși primele versiuni ale acestui sistem de operare nu au fost reușite, în ziua de azi, sistemul de operare Android, este unul dintre cele mai rapide și performante.

Această creștere și evoluție a avut un impact major și în domeniul realității augmentate. Realitatea augmentată reprezintă o interacțiune a lumii reale cu elemente ce

aparțin lumii virtuale, astfel cu ajutorul unui dispozitiv inteligent și al funcționalităților de care dispune, se poate crea o lume reală augmentată. Aceasta, este o interacțiune în timp real, care combină obiecte și spații aparținând lumii reale, cu obiecte și spații ce aparțin unei lumi virtuale. Folosind această tehnologie, se pot crea diferite aplicații care sunt interactive și mult mai intuitive pentru oameni, deoarece pot folosi obiecte ce aparțin de lumea reală, peste care se pot adăuga informații și obiecte virtuale. Prin intermediul, unui dispozitiv inteligent, folosind camere acestuia, se pot afișa diferite obiecte virtuale, cu care utilizatorul poate interacționa. Astfel realitatea augmentată, oferă o experiență ușor de înțeles și în același timp interactivă, cu rezultate imediat vizibile [31].

1.2. Contextul Proiectului

Deoarece au avut parte de un avans tehnologic atât de ridicat și de rapid, telefoanele inteligente au devenit niște dispozitive indispensabile pentru viața de zi cu zi, pe care ne bazăm din ce în ce mai mult. Fie ca recunoaștem sau nu, petrecem tot mai mult timp în fața ecranelor, atât adulții cât și copiii. Am ajuns să folosim aceste dispozitive pentru aproape orice activitate, pentru a comunica unii cu alții, pentru a ne informa, pentru a consuma mass-media, dar și pentru a ne relaxa sau distra. Dacă avem nevoie să trimitem un mesaj, sau o imagine o vom face folosind telefonul, dacă dorim să plătim o factură, sau chiar la casa de marcat, se poate folosi telefonul, acesta a devenit un mijloc prin care se pot transfera și primi date în orice domeniu de activitate.

Cunoscând toate aceste informații, este evident că și pentru copii tehnologia și telefoanele inteligente, vor face parte din viața lor de zi cu zi. Astfel conținutul, informațiile și aplicațiile la care au acces aceștia, trebuie selectate cu atenție și rigurozitate, deoarece ei sunt mult mai predispuși să preia informații false, fiind foarte ușor de influențat. Deși aceștia poate nu au un telefon inteligent propriu, părinții acestora sigur vor avea unul sau mai multe la îndemână, și cum este normal, orice copil va fi curios și va dori să dețină, sau să folosească un astfel de dispozitiv, mai ales când pentru unii dintre noi este un obiect ce îl folosim aproape tot timpul.

De multe ori se întâmplă să vedem părinți care pentru a liniști copiii, îi lasă să folosească un smartphone, dar fără a lua în considerare ce poate face acesta cu el. Pentru a echilibra această situație, copiilor nu ar trebui să le fie interzis să folosească un telefon inteligent, dar când folosesc unul, trebuie să ne asigurăm că informația este selectată corect. Astfel putem să folosim aceste dispozitive într-un scop educațional pentru cei mici.

1.3. Motivația

Având la cunoștință toate aceste informații explicate mai sus, tema proiectului meu de licență este, o aplicație mobilă pentru Android, care va folosi realitatea augmentată. Mai exact o aplicație ce rezolvă diferite puzzle-uri, folosindu-se de realitatea augmentată, ce încorporează elemente ce aparțin lumii reale, cu elemente ce aparțin de lumea virtuală. Aceasta va conține o listă de puzzle-uri predefinite, dar și o funcție care va permite utilizatorilor să își definească un puzzle propriu.

Am făcut această alegere, deoarece sistemul de operare Android, este mai comun și mai ușor de utilizat decât iOS, dar și datorită faptului că este mai răspândit, astfel mai mulți utilizatori pot avea acces la această aplicație.

Un alt motiv pentru alegerea acestei teme de proiect, este folosirea realității augmentate. Datorită avansului tehnologic din ultimii ani, crearea unei aplicații folosind realitatea augmentată a devenit mult mai ușor de realizat, deoarece această tehnologie a început să fie foarte populară. Din punctul meu de vedere aceasta, este o tehnologie care va fi folosită din ce în ce mai mult în viitor, în cât mai multe domenii, fiind intuitivă pentru oameni. Este potrivită pentru aplicații ce necesită diverși pași în rezolvarea unei probleme, în cazul de față pentru rezolvarea unui puzzle, ajutând utilizatorul prin afișarea diverselor obiecte și informații prin intermediul camerei telefonului.

Capitolul 2. Obiectivele Proiectului

2.1. Tema proiectului

Tema proiectului de licență este crearea și dezvoltarea unei aplicații mobile pentru telefoanele inteligente, ce folosesc sistemul de operare Android. Folosindu-se de tehnologia realității augmentate, aceasta are ca și scop indicarea și informarea utilizatorului în vederea rezolvării unui puzzle. Un joc de puzzle, se referă la reconstruirea unei imagini, sau al unui obiect, având la dispoziție o multitudine de piese care se pot conecta și intercala unele cu altele. Cu ajutorul camerei telefonului, se va recunoaște fiecare piesă a unui puzzle, iar prin afișarea diferitelor obiecte 3D și al altor elemente, aceasta îl va ghida pe utilizator pas cu pas pentru a putea rezolva puzzle-ul. Aplicația va conține o listă de puzzle-uri predefinite, din care se poate alege orice puzzle, dar utilizatorul va avea și opțiunea de ași defini propriul puzzle, pe care aplicația după ce au fost urmați toți pașii îl va putea recunoaște. Astfel din dorința de a studia acest domeniu al realității augmentate și de a oferi o aplicație interactivă pentru oameni, am ales această temă de proiect.

2.2. Cerințele funcționale

Cerințele funcționale, fac referire la modul în care un produs ar trebui să se comporte. Acestea sunt primele obiective ale unui proiect, fiind discutate și alese încă din faza de planificare. Pot fi dorințele pe care și le exprimă clientul, sau utilizatorul, dar și funcțiile pe care dezvoltatorul dorește să le implementeze. Cu ajutorul acestora se pune baza unui proiect [32].

1. Autentificarea utilizatorului

Prima funcționalitate a unei aplicații, trebuie să fie o metodă de înregistrare a utilizatorului. Deoarece este o aplicație dezvoltată pentru android, utilizatorul se putea autentifica în aplicație folosind contul de Google. Atunci când intră în aplicație, utilizatorului i se va cere să își aleagă contul Google cu care este conectat la dispozitiv, pentru a se putea autentifica în aplicație.

2. Acces la camera dispozitivului

Când aplicația este deschisă pentru prima dată, se va cere accesul la camera dispozitivului inteligent. Utilizatorul va putea să accepte acest lucru și să furnizeze acces aplicației sau nu. Dacă va alege să nu permită folosirea camerei, atunci aplicația nu va putea fi utilizată.

3. Recunoașterea pieselor de puzzle

Pentru a putea informa utilizatorul în legătură cu starea conexiunii pieselor de puzzle, aplicația trebuie să fie capabilă să recunoască piesele. Folosind un kit de dezvoltare software pentru realitatea augmentată, numit Vuforia,

integrat în aplicație, aceasta poate procesa imaginile, după care va fi capabilă să le recunoască.

4. Crearea unei baze de data

Pentru a recunoaște diferite imagini aplicația folosește kitul Vuforia, iar pentru ca acest lucru să fie posibil, este necesară crearea unei baze de date pe site-ul acestora. Baza de date va conține o cheie de licență, care va trebui conectată la proiect. Toate imaginile pe care dorim să le recunoaștem vor fi încărcate în această bază de date, acesta având și rolul de a le procesa. După ce sunt procesate de către Vuforia, imaginile primesc un anumit scor și în funcție de acesta, o imagine poate fi mai ușor de recunoscut sau mai greu.

5. Conexiunea pieselor

Ca să putem rezolva un puzzle este necesar să știm dacă două piese au fost conectate corect între ele. Astfel aplicația va afișa un obiect 3D de verificare, care îi va indica utilizatorului ca cele două piese sunt conectate corect și poate merge mai departe.

6. Poziția pieselor

Fiind vorba de un puzzle, acesta conține mai multe piese, iar atunci când acestea sunt împrăștiate cel mai probabil vor fi rotite în toate direcțiile și nu se vor afla într-o poziție ideală. Știind acest lucru, când este identificată o piesă care este rotită, aplicația va afișa o animație a unui obiect 3D, care va arată în ce direcție trebuie rotită piesa respectivă, iar când va ajunge într-o poziție corectă față de camera dispozitivului, obiectul 3D va dispărea.

7. Detectare finalizare puzzle

După ce toate piesele unui puzzle au fost conectate cu succes, aplicația va afișa un text, un mesaj peste puzzle, care are rolul de a anunța utilizatorul că puzzle-ul a fost completat și finalizat cu succes.

8. Captură ecran

Pentru ca utilizatorul să își poată defini propriul său puzzle, acesta trebuie să aibă opțiunea de a fotografia piesele dorite. Astfel la apăsarea unui buton, aplicația va face o captură de ecran a unei zone bine delimitate și va salva imaginea respectivă în memoria locală.

9. Încărcare imagini

După ce utilizatorul a creat diferite capturi de ecran pentru fiecare piesă în parte, va avea opțiunea de a încărca imaginile într-o bază de date cloud. Acest lucru este esențial, deoarece în baza de date ele sunt procesate și mai apoi pot fi recunoscute de către aplicație, iar prin folosirea unei baze de date cloud, imaginile nu vor trebui stocate local și vor putea fi accesate direct din server.

10. Afișare legături piese

Pentru a ști ce piese trebuie conectate între ele, se va implementa o funcționalitate care va desena o linie între două piese. De exemplu aplicația va identifica două piese ce trebuie conectate între ele. În acest moment se va afișa o linie între cele două piese, care îi va indica utilizatorului ca acestea trebuie conectate între ele.

11. Recunoaștere piese din baza de date cloud

După ce imaginile au fost încărcate în baza de date cloud, aplicația va trebui să le poată recunoaște. Acest lucru se va face folosind un exemplu pus la dispoziție de către cei de la Vuforia, care va compara textura unei ținte, cu texturile din baza de date cloud, iar când se găsește o textură identică în baza de date, ținta va fi recunoscută și îi vor fi aplicate interacțiunile virtuale.

2.3. Cerințe non-funcționale

Cerințele non-funcționale fac referire la atributele unui proiect dar și la calitatea acestuia. Descriu capacitățile de funcționare și constrângerile care îmbunătățesc performanța unui proiect. Acestea trebuie să arate anumite trăsături ale unui sistem, cum ar fi: portabilitatea, securitatea, scalabilitatea și altele [32].

1. Performanța

Această caracteristică, definește cât de rapid poate un sistem software să răspundă la anumite sarcini. Pentru ca o aplicație să fie cât mai performantă, aceasta trebuie să aibă un timp de răspuns cât mai mic. Timpul de răspuns, este o caracteristică importantă, care este strâns legată de performanțele dispozitivului pe care rulează, dar și de conexiunea la internet de care dispune acesta. În cazul aplicației de față, atunci când trebuie să scaneze diferite piese, aceasta trebuie să le recunoască cât mai repede, pentru a putea indica utilizatorului pașii necesari într-un timp cât mai scurt și cât mai precis. Pe lângă acest lucru, în momentul în care un utilizator dorește să își definească propriul puzzle, imaginile salvate și încărcate în baza de date, trebuie procesate cât mai repede, pentru a fi disponibile utilizatorului imediat după ce a terminat pașii de definire.

2. Scalabilitate

Se referă la integritatea unei aplicații, atunci când este supusă unor sarcini foarte mari, iar performanțele acesteia nu sunt afectate foarte mult. Aceasta are un rol important în aplicația curentă, deoarece baza de date trebuie să suporte cât mai multe imagini furnizate de către utilizatori. Cu cât baza de date poate susține mai multe imagini, cu atât utilizatorul poate defini cât mai multe puzzle-uri. Aceasta trebuie să fie scalabilă și din

punctul de vedere al dimensiunilor puzzle-urilor, inițial reușind să suporte puzzle-uri de nouă și cincisprezece piese.

3. Securitate

Securitatea, se referă la procesul care menține un nivel acceptabil de risc. Aceasta se ocupă cu protejarea informațiilor, datelor și a sistemelor de către accesul neautorizat. Ca fiecare utilizator să aibă propriile puzzle-uri definite, securitatea este un factor important, deoarece acesta trebuie să asigure o siguranță și o integritate asupra bazei de date în care sunt încărcate imaginile. De aceea atunci când aplicația este deschisă, utilizatorului i se va cere să se conecteze automat cu contul sau Google, pentru a putea diferenția utilizatorii între ei.

4. Uzabilitate

Această caracteristică arată cât de ușor un anumit utilizator, într-un context anume, poate folosi un anumit produs, pentru a ajunge la rezultatul dorit cât mai repede și cât mai eficient posibil. Astfel folosind o interfață cât mai simplistă și ușor de înțeles ne putem asigura ca nu vor exista probleme de înțelegere a aplicației de către un utilizator. Pentru o mai bună înțelegere a funcționalității aplicației, aceștia vor avea la îndemână informații cu privire la folosirea aplicației și ce pași sunt necesari pentru a ajunge la rezultatul dorit.

5. Disponibilitate

Se referă la funcționarea unui sistem într-un anumit interval de timp. În cazul de față disponibilitatea, este un lucru esențial în buna funcționare a aplicației, deoarece utilizatorii trebuie oricând să își poată defini propriul puzzle. Astfel baza de date trebuie să fie disponibilă tot timpul în cazul puzzle-urilor definite de către utilizatori.

6. Portabilitate

Această caracteristică se referă la cât de portabil poate fi un dispozitiv sau un obiect. Fiind vorba de o aplicație ce rulează pe un dispozitiv inteligent, aceasta poate fi mereu la îndemâna utilizatorilor.

7. Fiabilitate

Fiabilitatea, reprezintă timpul în care aplicația poate rula fără să apară probleme în timpul execuției. Este o caracteristică destul de greu de măsurat inițial, deoarece aplicația nu este implementată încă și este foarte greu de prezis cât de fiabilă poate să fie. Doar după ce va fi testată și vor apărea primele rezultate, se va putea spune cu certitudine cât de fiabilă este acesta.

8. Compatibilitate

Definește modul în care un sistem poate coexista cu un alt sistem în același loc, mediu. Astfel aplicația instalată pe un anumit sistem, trebuie să

fie compatibilă cu acesta, pentru a nu exista probleme de configurare și conexiune.

9. Mentenabilitate

Proprietatea unui produs, sistem, sau proiect, de a fi întreținut și reparat cu ușurință. Astfel atunci când apar probleme în aplicație, acestea ar trebui să fie ușor de reparat, într-un timp cât mai scurt.

Capitolul 3. Studiu Bibliografic

3.1. Aplicații similare

3.1.1. *Your AR Puzzle*

Este o aplicație ce a fost lansată pe data de 23 august 2017, care folosește realitatea augmentată în vederea rezolvării unui puzzle. Aceasta conține 408 puzzle-uri, de diferite dimensiuni, de la 30 de piese până la 120. Aplicația rulează pe dispozitivele ce folosesc ca și sistem de operare Android. Versiunea gratuită a acesteia continue doar 30 de puzzle-uri, pentru celelalte 408 trebuie cumpărată.

Aceasta se folosește de camera telefonului pentru a scana suprafețele, atunci când o suprafață este netedă și dreaptă, cu ajutorul unui marker se setează poziția unui șablon pentru puzzle. După ce acești pași au fost făcuți, aplicația începe automat, iar utilizatorul își poate alege puzzle-ul pe care dorește să îl rezolve. În această aplicație utilizatorul trebuie să poziționeze piesele 3D, pe un șablon deja stabilit [36].

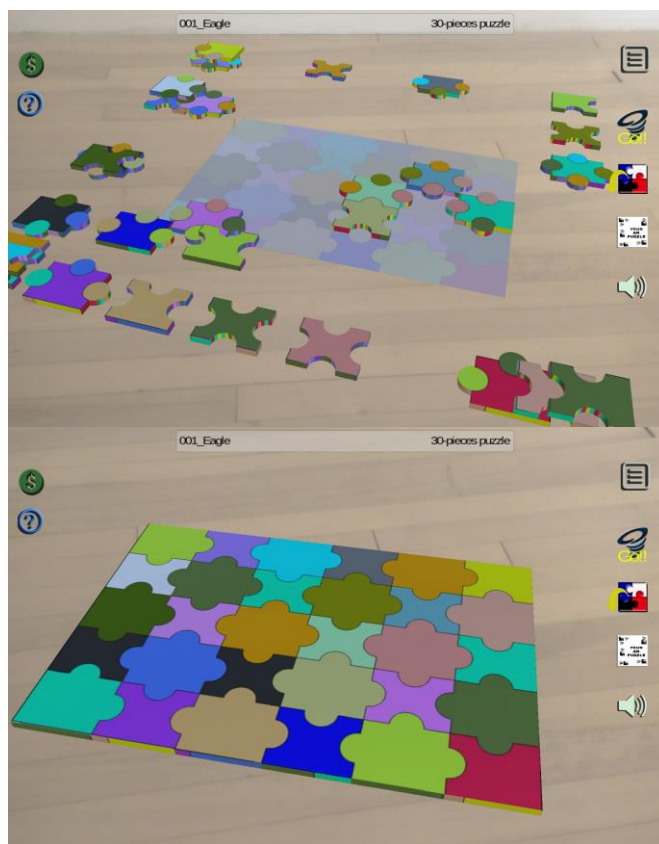


Figura 3.5 – Aplicație ce folosește realitatea augmentată Your AR Puzzle [36].

3.1.2. *AMON*

Amon este un joc de puzzle ce folosește realitatea augmentată, dar într-un mod mai diferit decât proiectul ales de mine. Utilizatorii trebuie să combine diferite părți ale unor obiecte 3D. Puzzle-urile, în sine sunt făcute din sculpturi 3D scanate, antice din

muzee din întreaga lume. După ce se rezolvă un puzzle, se vor afișa mai multe informații despre sculptură, de cine a fost realizată și multe altele. Această aplicație se folosește de tehnologia ARKit, care este dezvoltată de către compania Apple. Jocul plasează mai multe bucăți de sculpturi sparte, iar jucătorul trebuie să se poziționeze într-un mod în care sculptura să apară sub forma sa completă. Uneori, jocul adaugă diferite piese în plus ce trebuie eliminate. Pe măsura ce jucați, vă veți schimba continu poziția, încercând să înțelegeți toate piesele împrăștiate prin cameră [37].

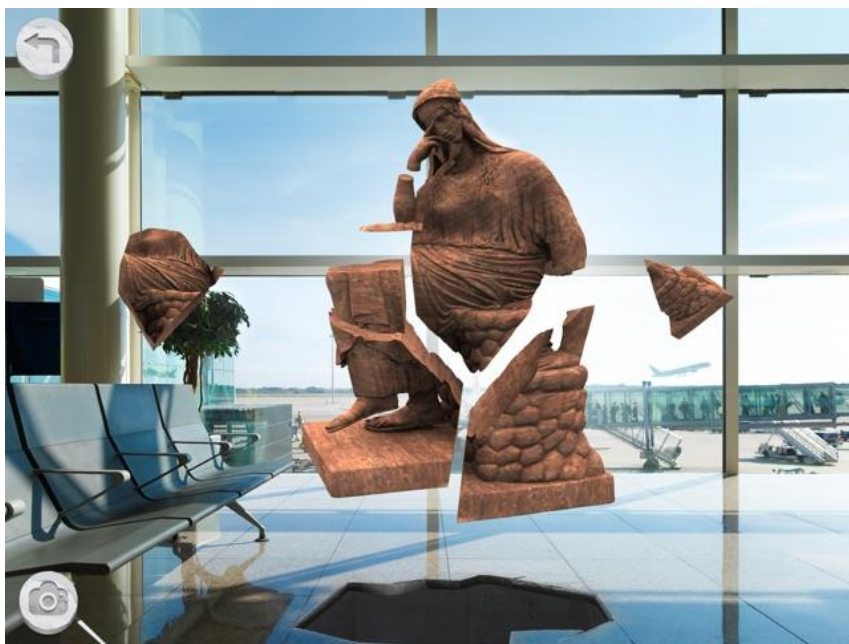


Figura 3.6– Aplicație ce folosește realitatea augmentată Amon [37].

3.1.3. Ravensburger Puzzle

Este o aplicație pentru dispozitivele ce rulează cu sistemul de operare iOS. Aplicația are parte de o interacțiune audio/video cu utilizatorul. Un dezavantaj al acesteia, este faptul că puzzle-ul trebuie să fie deja asamblat pentru a funcționa. Scanează puzzle-ul și dacă acesta este finalizat, se va deschide o imagine care poate fi rotită la 360 de grade. Utilizatorul va putea vedea peisajele în realitatea augmentată și va primi informații despre acestea [38].



Figura 3.7 – Aplicație ce folosește realitatea augmentată Ravensburger Puzzle [38].

3.1.4. *PuzzlAR*

Aplicația a fost creată de o echipă din Liasbona, Portugalia. Obiectivul acestui joc este de a reasambla diferite monumente naționale. Acestea au fost împărțite în mai multe bucăți 3D, iar utilizatorul trebuie să le assembleze la loc. Aplicația se folosește de Magic leap One și de un dispozitiv de control wireless, pentru ca jucătorul să poată roti și plasa obiectele 3D.

Pentru a putea începe jocul, utilizatorul trebuie să scaneze o zonă netedă și plată peste care aplicația va putea afișa monumentele și obiectele 3D, care vor trebui rezolvate [39].

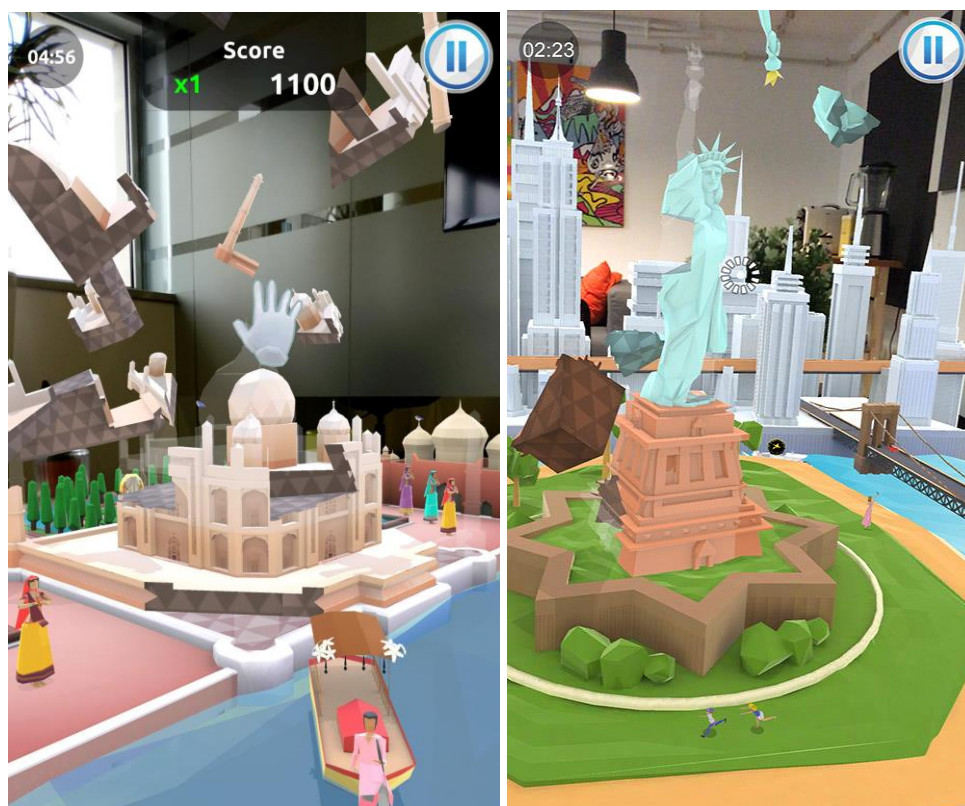


Figura 3.8 – Aplicație ce folosește realitatea augmentată PuzzlAR [39].

3.1.5. *4D Puzzles*

Este o aplicație destinată copiilor. După ce aceștia rezolvă un anumit puzzle, aplicația va scana imaginea formată și va afișa diferite obiecte și interacțiuni legate de puzzle-ul respectiv. Aceasta rulează atât pe iOS cât și pe Android. Deși se folosește de un puzzle pentru a funcționa, acesta nu îl ajută utilizator în rezolvarea, se folosește doar de imaginea finală rezultată [40].



Figura 3.9 – Aplicație ce folosește realitatea augmentată 4D Puzzles [40].

3.1.6. ARise

Este o aplicație ce aparține de Climax Studios Ltd. Aceasta combină distracția oferită de un puzzle, cu tehnologia realității augmentate. Când este lansată aplicația, caracterul trebuie să navigheze prin împrejurimi, folosindu-se de perspectiva utilizatorului și de indiciile oferite de joc. În această aplicație, trebuie reconstruite drumuri și ocoli obstacole, pentru a ajunge la finalul jocului [41].



Figura 3.10 – Aplicație ce folosește realitatea augmentată ARise [41].

3.2. Diferențe între aplicația propusă și cele similare

Aplicația dezvoltată de mine, deși la bază seamănă cu cele prezentate mai sus, propune o altă abordare. Aplicațiile similare, majoritatea folosesc puzzle-uri 3D ce aparțin lumii virtuale. Se folosesc de mediul lumii reale pentru a putea funcționa, dar toate obiectele sunt virtuale. Spre deosebire de acestea, proiectul meu dorește să rezolve puzzle-uri reale, cu ajutorul realității augmentate. Utilizatorul va trebui să interacționeze cu piesele de puzzle în mod direct și nu va trebui să modifice și să poziționeze diferite obiecte 3D pe ecranul telefonului pentru a-și putea atinge scopul. Aplicația va interacționa live cu piesele de puzzle, primind informații despre acestea prin intermediul camerei telefonului, astfel este esențial ca piesele să se afle în fața camerei.

Altă diferență este tehnologia aleasă pentru dezvoltarea acestora. Proiectul aflat în cauza, este dezvoltat cu ajutorul motorului Vuforia, pe când aplicațiile menționate mai sus folosesc ARKit sau altele. Datorită faptului că sunt folosite diferite tehnologii, fiecare aplicație are un sistem de operare țintă, iOS sau Android. ARKit poate dezvolta aplicații doar pentru dispozitivele ce folosesc ca și sistem de operare iOS, dar Vuforia poate suporta ambele sisteme, atât iOS cât și Android.

Capitolul 4. Analiză și Fundamentare Teoretică

4.1. Cazuri de utilizare

Aplicația are două cazuri de utilizare, dintre care actorul poate alege. Primul caz este selectarea unui puzzle predefinit, pe care utilizatorul îl poate folosi și rezolva direct, iar al doilea caz este cel în care utilizatorul își va putea defini propriul puzzle.

4.1.1. *Puzzle predefinit*

Acesta, este un puzzle care a fost adăugat în aplicație de către dezvoltator și este disponibil utilizatorului din momentul în care accesează aplicația. Este și primul caz de utilizare, actorul având nevoie doar de puzzle-ul fizic, ce se afla deja în aplicație.

4.1.1.1. *Flux de evenimente pentru puzzle-ul predefinit.*

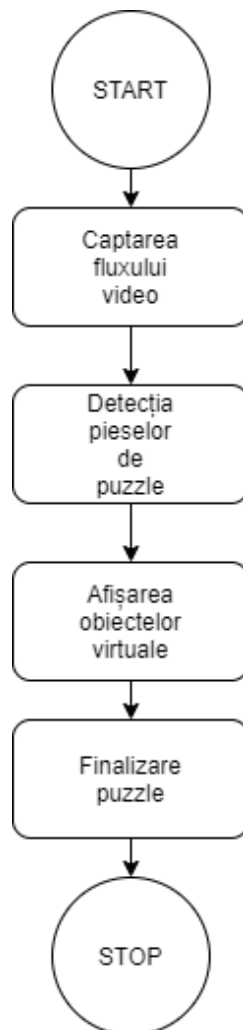


Figura 4.1 – Diagrama fluxului de evenimente pentru un puzzle predefinit.

1. Fluxul de bază

1.1 START- Acesta reprezintă momentul în care utilizatorul a apăsă pe un puzzle din listă, aplicația este redirecționată către respectivul puzzle în vederea rezolvării acestuia.

1.2 Pași de utilizare

- Captarea fluxului video – Va fi captat cu ajutorul camerei dispozitivului inteligent. După ce utilizatorul a ales un puzzle, aplicația accesează camera și începe un live de captare a imaginilor, prin care va reuși să identifice piesele de puzzle.
- Detecția pieselor de puzzle – Începe odată cu fluxul video, astfel utilizatorul poate îndrepta dispozitivul, cu camera spre piesele dorite, iar aplicația va începe să recunoască piesele. Identificarea pieselor se face cu ajutorul kitului Vuforia, iar pentru a putea fi recunoscute, imaginile trebuie încărcate în baza de date a acestora.
- Afișarea obiectelor virtuale – După ce o piesă sau mai multe au fost detectate, aplicația va adăuga obiectele 3D și indicațiile necesare pentru fiecare dintre acestea. Indicațiile, sunt obiecte virtuale ce sunt vizibile cu ajutorul ecranului dispozitivului folosit, doar în momentul în care o piesă este recunoscută.
- Finalizare puzzle – În momentul în care ultima piesă este conectată cu restul, aplicația va recunoaște că s-a ajuns la final și va afișa un mesaj care îi va spune utilizatorului că puzzle-ul a fost completat cu succes.

1.3 STOP – După ce puzzle-ul dorit a fost finalizat, utilizatorul se poate întoarce înapoi în meniul principal, astfel fluxul de detectare a pieselor se oprește.

2. Flux Alternativ

Este posibil ca din cauza condițiilor, a luminii, dar și a dispozitivului, o piesă, sau mai multe să nu fie foarte vizibile, astfel aplicația nu o va putea detecta. Pentru a rezolva aceasta problemă, aplicația poate fi repornită, sau se va cauta o poziție în care lumina și condițiile sunt potrivite pentru o calitate mai bună a imaginii. Dacă este vorba de o singură piesă, utilizatorul poate rezolva puzzle-ul fără a ține cont de acea piesă și o poate adăugat la sfârșit.

3. Precondiții

3.1 Pentru a putea recunoaște o piesă de puzzle , aceasta trebuie să se afle în baza de date de la Vuforia.

3.2 Camera dispozitivului de pe care rulează aplicația, trebuie să fie în parametrii normali de funcționare.

3.3 Lumina nu trebuie să fie foarte intensă, sau foarte scăzută, pentru a nu afecta performanțele camerei, rezultând în erori de recunoaștere a pieselor.

4. Postcondiții

4.1 Aplicația trebuie să identifice toate piesele unui puzzle pentru ca acesta să fie finalizat cu succes.

4.1.2. Definirea unui puzzle

Aplicația permite definirea unui puzzle propriu de către utilizator.

4.1.2.1. Flux de evenimente pentru definirea unui puzzle

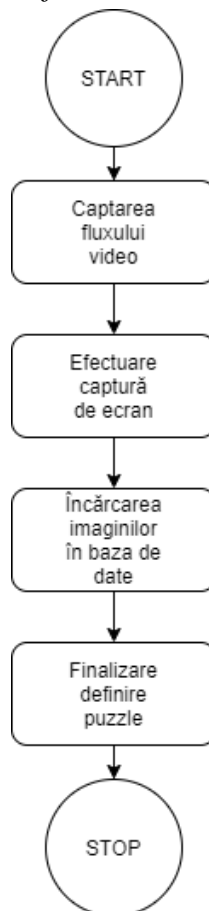


Figura 4.2 – Diagrama fluxului de evenimente pentru definirea unui puzzle.

1. Fluxul de bază

1.1 START – Reprezintă momentul în care utilizatorul a ales din meniul principal al aplicației opțiunea de definire a unui puzzle, fiind redirecționat în meniul de definire specific.

1.2 Pașii de utilizare

- Captarea fluxului video - Va fi captat cu ajutorul camerei dispozitivului inteligent. De această dată, va fi disponibilă doar o zona delimitată, în care utilizatorul trebuie să încadreze piesele dorite.
- Efectuare captură de ecran – După ce piesa a fost poziționată, utilizatorul va apăsa pe un buton, care va efectua o captură de ecran a zonei delimitate și o va salva.
- Încărcarea imaginilor în baza de date – Pentru a putea fi procesate de către Vuforia, imaginea salvată trebuie încărcată în baza de date a acestora. Astfel utilizatorul va avea un buton care va trimite imaginea respectivă către baza de date.
- Finalizare definire puzzle – După ce toate piesele au fost salvate și încărcate, aplicația îi va indica utilizatorului acest lucru prin afișarea unui mesaj text.

1.3 STOP – Aplicația își oprește fluxul în momentul în care utilizatorul părăsește meniul de definire al unui puzzle și este redirecționat către meniul principal.

2. Flux Alternativ

Utilizatorul poate greși sau poate ieși din greșeala din meniul de definire a unui puzzle. În această situație, acesta trebuie să reia definirea puzzle-ului de la început.

3. Precondiții

3.1 Memorie suficientă pentru stocarea imaginilor.

3.2 Camera dispozitivului de pe care rulează aplicația trebuie să fie în parametrii normali de funcționare.

3.3 Lumina nu trebuie să fie foarte intensă, sau foarte scăzută, pentru a nu afecta performanțele camerei, rezultând în erori de recunoaștere a pieselor.

3.4 Conexiune stabilă la internet.

4. Postcondiții

4.1 Toate imaginile salvate trebuie încărcate în baza de date.

4.2 Imaginile trebuie să fie clare pentru a fi ușor de identificat de către aplicație.

4.3 Dacă a fost definit un puzzle, acesta trebuie să rămână salvat pentru a putea fi refolosit.

4.1.3. Cerințe speciale

- Utilizatorul trebuie să dețină în primul rând un dispozitiv inteligent, care folosește sistemul de operare Android. Pe lângă acest lucru trebuie să aibă instalată minim versiunea 6.0 Marshmallow, pentru ca aplicația să funcționeze corespunzător.
- Dispozitivul trebuie să conțină o cameră care poate înregistra un flux live la o rezoluție cât mai bună. Cu cât rezoluția este mai bună și imaginea va fi mai clară, astfel piesele de puzzle vor fi mult mai ușor de recunoscut.
- Este necesară o conexiune la internet, atunci când utilizatorul dorește să definească un puzzle propriu. Conexiunea este necesară, deoarece imaginile salvate trebuie încărcate în baza de date de la Vuforia, pentru a putea fi recunoscute de către aplicație.
- Este necesară deținerea unui puzzle fizic, pe care aplicația să poată fi utilizată.

4.2. Tehnologii

4.2.1. Realitatea Augmentată

Realitatea augmentată, reprezintă o îmbinare a lumii reale cu lumea virtuală. Datorită avansului tehnologic și a faptului că dispozitivele inteligente au devenit tot mai accesibile și mai performante, aplicațiile ce folosesc tehnologia realității augmentate, au crescut în popularitate și au ajuns mult mai ușor de implementat și de folosit. La prima vedere pare ca nu este foarte răspândită, dar dacă ne uităm mai atent, vom observa că este mai aproape de noi decât am crede, un exemplu al acestei tehnologii, este reprezentat de filtrele de la diferite aplicații. Acestea se folosesc de realitatea augmentată pentru a identifica fața unei persoane și aplică un efect sau o mască 3D peste aceasta. Un alt exemplu popular este jocul Pokemon Go, lansat în anul 2016, în care utilizatorii trebuie să localizeze diferite caractere în lumea reală, prin intermediul lumii virtuale [31].

Această tehnologie a apărut pentru prima dată în anul 1990, și a fost folosită pentru a ajuta lucrătorii de aeronave să vizualizeze diferite sisteme complexe. Deși a fost folosită în diverse domenii, abia după anii 2010 a suferit parte de o creștere în

popularitate, deoarece au început să apară tehnologii tot mai avansate care pot să folosească realitatea augmentată în favoarea lor. Această creștere a fost susținută și de dezvoltarea dispozitivelor inteligente, ce au devenit performante și capabile de a rula această tehnologie [32].

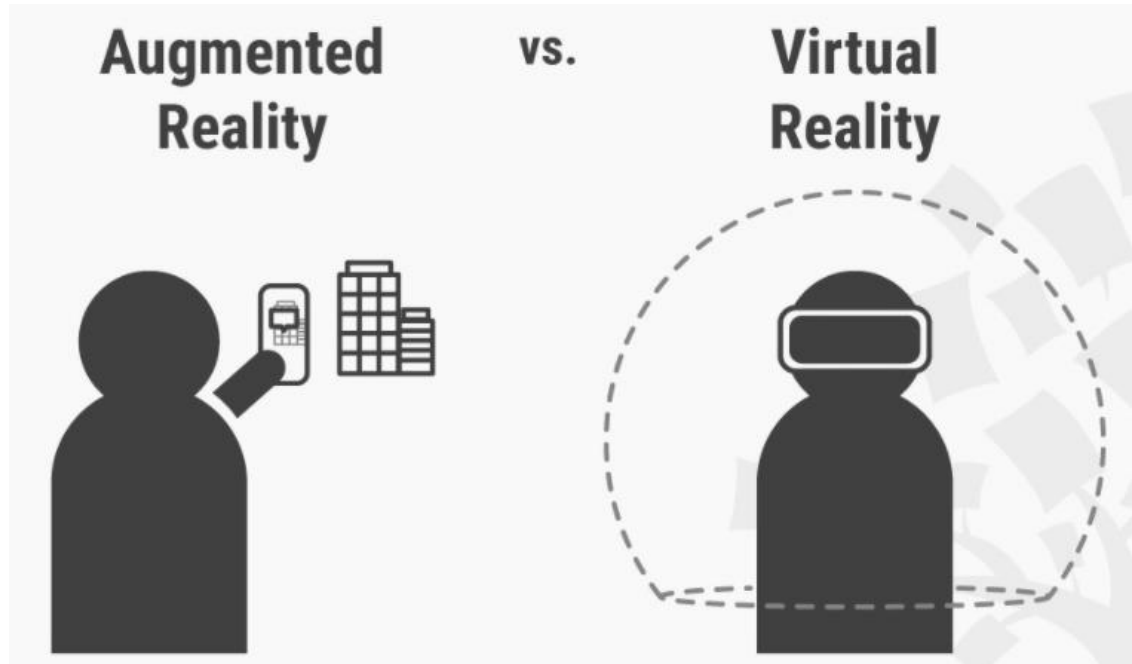


Figura 3.1 – Diferența dintre realitatea augmentată și realitatea virtuală. [34]

Realitatea augmentată, nu trebuie confundată cu realitatea virtuală. Deși ambele aparțin de lumea virtuală, realitatea augmentată este o proiecție digitală peste lumea reală, obiecte virtuale interacționează cu obiecte din lumea reală, adesea prin intermediul dispozitivelor inteligente. Realitatea virtuală izolează utilizatorii complet de lumea reală și îi introduce într-o lume complet nouă, cu ajutorul unor dispozitive speciale cum ar fi ochelarii VR. Astfel, putem spune că realitatea augmentată este o combinație dintre lumea reală și cea virtuală, pe când realitatea virtuală, este o lume complet virtuală [34].

Având parte de o creștere atât de mare, în viitor tot mai multe dispozitive și aplicații vor folosi realitatea augmentată pentru a ne face viața mai ușoară și mai plăcută. Domeniile în care poate fi folosită sunt nelimitate, ceea ce o face o tehnologie de viitor. Acesta poate fi folosită pentru divertisment, educație, dar și pentru diferite locuri de muncă.

Avantajele realității augmentate în educație:

- Este ușor de folosit pentru a prezenta cursanților diferite lucruri, deoarece aceștia pot să le vadă și chiar să interacționeze cu acestea. De exemplu, elevilor li se pot prezenta la ora de biologie anumite elemente ale corpului uman, pentru a fi mai ușor de înțeles de către aceștia, în același timp, activitatea devenind mult mai interactivă și pe placul acestora.
- Cursanții pot opera realitatea augmentată în propriul ritm și rezultatele folosirii se pot vedea imediat.

- Cu ajutorul acestei tehnologii se poate înțelege mai ușor relația dintre lumea reală și lumea virtuală și cum interacționează cele două între ele.

4.2.2. C#

Proiectul este implementat ca o aplicație Android, folosind ca limbaj de programare primar C#. Este un limbaj dezvoltat de către Microsoft, fiind orientat pe obiecte. Deși la bază acesta este apropiat de limbajele tradiționale de nivel înalt, C și C++, se aseamănă mai mult cu Java. Prezintă o multitudine de caracteristici cum ar fi [28]:

- Colectarea gunoaielor (Garbage collection).
- Tratarea excepțiilor (Exception handling).
- Expresii Lambda (Lambda expression).
- Sintaxa de interogare (Query syntax).
- Operațiuni asincrone (Asynchronous operations).
- Potrivirea modelului (Pattern matchin).
- Sistem de tip unificat (Unified type system).

Toate tipurile limbajului moștenesc caracteristici de la un singur obiect, care este de tipul rădăcină. Este un limbaj simplu și mai ușor de înțeles, asigură că fiecare variabilă nu deține valori ale altor tipuri. Fiind un limbaj orientat pe obiect, permite obiecte, clase, moștenire, polimorfism și altele. Toate aceste caracteristici au făcut ca C# să devină unul dintre cele mai populare limbaje de programare, putând fi folosit pentru aplicații mobile, dezvoltarea jocurilor, dar și a aplicațiilor Web [28].

4.2.3. Vuforia Engine

Reprezintă o tehnologie, un kit de dezvoltare pentru dispozitivele inteligente, care folosește realitatea augmentată și permite dezvoltarea de diferite aplicații ce pot folosi această tehnologie. Este un SDK ce folosește tehnologia viziunii calculatorului, pentru a recunoaște și capta diferite imagini plane, dar și obiecte 3D. Aceste imagini și obiecte, sunt recunoscute prin intermediul camerei unui dispozitiv, în timp real, iar Vuforia permite dezvoltatorilor să adauge obiecte virtuale, peste cele reale atunci când au fost recunoscute. Suportă atât tipuri 2D, cât și obiecte 3D, inclusiv configurarea mai multor ținte în același timp. Streamul de date este împărțit în patru module principale: intrările, baza de date, urmărirea și potrivirea țintelor, dar și ieșirile redade. Dispozitivele inteligente, pot prelua imaginile cadru cu cadru și le compară cu cele din baza de date. Dacă sunt găsite în baza de date, se vor adaugă obiecte virtuale peste imaginile recunoscute [16].

Acest SDK conține mai multe tipuri de scanări, ce pot fi folosite pentru diferite situații și obiecte, astfel Vuforia conține următoarele servicii [15]:

- ImageTargets – Este cea mai ușoară metodă de implementat și cea mai ușor de recunoscut de către aplicație, deoarece acesta urmărește și recunoaște imagini plane, 2D. Imaginea este detectată prin compararea caracteristicilor, astfel pentru a putea face această comparație, imaginile

trebuie mai întâi adăugate într-o bază de date de la Vuforia, unde sunt procesate și pregătite pentru a fi recunoscute. După ce imaginea este detectată, aceasta va fi urmărită în permanență, iar aplicația va afișa obiectele și trăsăturile virtuale peste ea

- **Multi Targets** – În principal, această opțiune are ca scop recunoașterea obiectelor 3D, cu suprafețe plane și drepte, un exemplu de astfel de obiect este un cub. Un obiect multiplu este o colecție de mai multe imagini combinate, astfel Vuforia permite urmărirea obiectului din orice direcție. De exemplu, dacă dorim să detectăm un cub, se vor scana toate laturile acestuia, iar imaginile vor fi adăugate într-o bază de date specială pentru astfel de obiecte. Din nou aplicația va procesa toate imaginile și le va asigna unui singur obiect, astfel reușind să îl detecteze din orice poziție ar fi acesta. Imaginile trebuie adăugate în ordinea corespunzătoare, pentru a nu apărea erori atunci când se va încerca recunoașterea obiectului.
- **Cylinder Targets** – Această funcție este folosită pentru detectarea unei imagini, care se află pe o suprafață cilindrică. Un exemplu de astfel de obiect, este o doză de suc. Metoda, funcționează similar cu celelalte două funcții discutate mai sus, imaginea de pe forma cilindrică se încarcă în baza de date pentru a putea fi procesată. Diferența constă în faptul că se pot recunoaște părți ale imaginii pe rând și nu este necesar ca aceasta să fie vizibilă în totalitate.
- **Object Target** – Se referă la recunoașterea diferitelor obiecte 3D, în special a jucăriilor de dimensiuni nu foarte mari. Pentru a putea detecta un astfel de obiect, se folosește un scanner de obiecte, pus la dispoziție pe site-ul celor de la Vuforia. Obiectul trebuie pus pe o foaie specială, ce se găsește în tutorialul oficial, iar acesta trebuie să conțină cât mai puține părți mobile și suprafață lui să fie cât mai complexă și plină de diferite texturi, pentru a putea fi scanat și detectat cu ușurință. După ce obiectul în cauză a fost scanat, fișierul rezultat trebuie încărcat în Vuforia Target Manager, unde va fi generat obiectul. La finalizarea acestui proces, baza de date rezultată se va descarca și va fi adăugată în proiectul dorit.
- **Model Target** – Aplicațiile ce folosesc această funcție, creează un model al unui obiect 3D, mai exact conturul acestuia. Atunci când obiectul se află în cadrul camerei, conturul acestuia se va poziționa peste obiect, iar dacă se detectează o potrivire a conturului cu obiectul, atunci aplicația îl va putea recunoaște. Această modalitate de recunoaștere, se bazează tot pe imagini plane, deoarece se urmărește conturul obiectului și nu forma lui 3D.

4.2.4. Vuforia SDK

Este un SDK ce folosește realitatea augmentată pentru dispozitivele inteligente. Permite executarea aplicațiilor AR, într-un live obținut cu camera dispozitivelor. Folosește capacitățile tehnologiei de viziune computerizată, pentru a reuși să recunoască individual obiectele captate.

Capacitatea acestui SDK de a înregistra imagini, permite dezvoltatorilor să poziționeze și să orienteze în spațiu obiecte virtuale, în principal obiecte 3D, care vor fi vizualizate prin intermediul camerei unui dispozitiv inteligent. Obiectele virtuale pot urmări în timp real orientarea și poziția imaginii, astfel încât perspectiva utilizatorului asupra imaginii se va regăsi și pentru obiectele virtuale.

Principalele componente pentru o aplicație ce folosește Vuforia sunt:

- Camera - Se asigură că fiecare cadru este capturat și transmis către următor. Dezvoltatorul trebuie doar să spună camerei când să urmărească o imagine și când ar trebui să se oprească.
- Convertor de imagine – Convertește formatul camerei care capturează imaginea, într-un format potrivit pentru redare OpenGL și pentru urmărire.
- Urmăritor – Conține algoritmul de viziune computerizată, care detectează și urmărește obiecte din lumea reală în cadre video. Pe baza imaginii redade, diferiți algoritmi au grija să detecteze noi ținte.
- Redare video pe fundal – Redă imaginea camerei, care este stocată în obiectul de stare.
- Codul aplicației – Dezvoltatorii trebuie să inițializeze toate componentele prezentate. Pentru fiecare cadru procesat, starea obiectului este actualizată.
- Ținte – Resursele țintă sunt create folosind sistemul de gestionare a țintei online. Setul de date care se descarcă, conține un fișier XML și unul binar care conține baza de date.

Urmărirea, reprezintă principala caracteristică a acestui sistem. Este constituită din obiectele lumii reale, urmate de cele virtuale atunci când urmărirea are succes și se recunoaște un obiect.

Sunt mai multe tipuri de urmărire:

- Unknown_type (tip necunoscut) – Când aplicația nu știe ce tip de țintă urmărește.
- Image_Target (Imagine țintă) – O țintă urmărită de tipul ImageTarget.
- Multi_Target (Ținte multiple) – Mai multe ținte urmărite simultan.
- Marker – O țintă urmărită de tipul Marker.

4.2.5. Vuforia și baza de date

Vuforia Engine, este o platformă software care permite dezvoltatorilor să creeze diferite aplicații folosind realitatea augmentată. Este una dintre cele mai populare platforme, aceasta fiind compatibilă cu sistemele de operare Android, iOS și UWP. Permite mai multe tipuri de recunoaștere a imaginilor, dar pentru acest proiect se va folosi Image Targets. Se referă la imagini pe care Vuforia le poate detecta și urmări în timp real, prin compararea diferitelor caracteristici ale unei imagini, cu o bază de date. Astfel dacă imaginea pe care o identifică se află în baza de date, o poate recunoaște și îi poate aplica conținutul virtual [17].

Pentru a putea procesa imaginile, acestea trebuie să aibă o textură cât mai complexă, pentru a putea fi identificate caracteristici unice, pentru ca imaginea să fie mult mai ușor de recunoscut. Imaginile trebuie să aibă un format JPG, sau PNG, și să fie RGB sau în tonuri de gri. Pentru a putea fi încărcate în baza de date, trebuie să aibă o dimensiune de maxim 2,25 MB și o lățime minimă de 320 de pixeli [18].

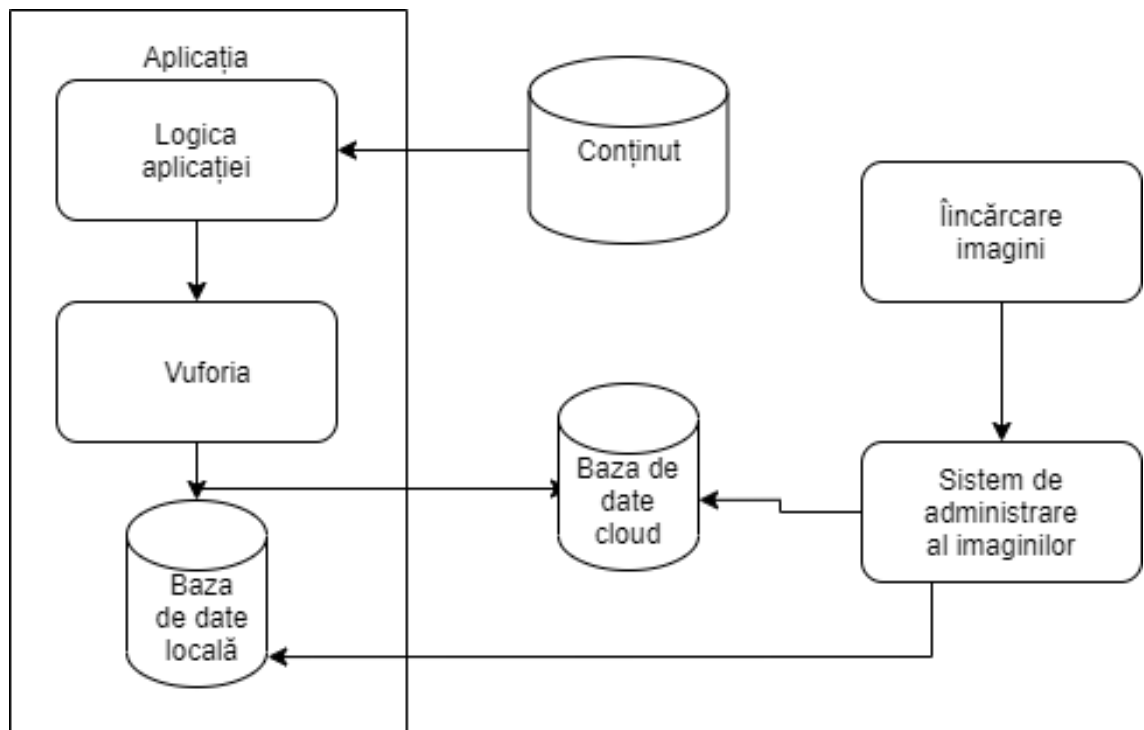


Figura 4.3 – Diagramă de funcționare a sistemului Vuforia [17].

Aplicația folosește o arhitectură de tip client-server, prin care comunică cu baza de date cloud. După cum se poate observa în figura 4.2, aplicația preia imagini din baza de date locală, dar și din baza de date cloud, deoarece pentru un puzzle predefinit este necesar ca imaginile să fie salvate în proiect, iar pentru un puzzle definit de către utilizator, imaginile sunt stocate cloud.

1. Baza de date locală

Este folosită pentru a încărca imaginile direct în proiect, fiind accesibile rapid și local. Se pot descărca mai multe baze de date ce pot fi incluse în aplicație, dar acestea trebuie să fie sub aceeași licență Vuforia. Într-o bază de date pot fi stocate până la 1000 de imagini. Aceasta se creează pe site-ul celor de la Vuforia, în portalul pentru dezvoltatori. După ce a fost creată, se pot încărca imaginile dorite, pe care baza de date le procesează. Fiecare imagine va primi o notă ce este reprezentată prin stele de la 0 la 5. Dacă primește 0 stele, înseamnă ca imaginea nu poate fi recunoscută de către aplicație, iar dacă primește 5, înseamnă că este foarte ușor de identificat. În final baza de date se poate descărca și va putea fi introdusă în proiect [18].

2. Baza de date cloud

Aceasta diferă de cea locală, prin faptul că nu mai este necesară descărcarea acesteia și introducerea în proiect, ci imaginile sunt recunoscute direct din cloud. Poate susține până la 1000 de imagini, dar un dezavantaj mare al acesteia, este că poate recunoaște simultan decât o imagine. Un avantaj îl reprezintă posibilitatea de a încărca imagini direct din aplicație, astfel nu mai este necesar ca imaginile să fie introduse manual [18].

4.2.6. Unity Engine

Este o platformă dezvoltată de compania Unity Technologies, fiind destinată în principal pentru crearea și dezvoltarea jocurilor video. Deși la baza este un motor de dezvoltare a jocurilor, acesta a fost extins pentru a suporta mai multe platforme printre care și Android. Poate fi folosit pentru a crea proiecte și aplicații tridimensionale, dar și bidimensionale. O caracteristică importantă, este faptul că suportă tehnologii care folosesc realitatea augmentată, incluzând Vuforia Engine [4]. Unity este responsabil pentru crearea a 60% dintre jocurile și aplicațiile ce au apărut pe telefoanele mobile, care au legătură cu realitatea augmentată, dar și cu realitatea virtuală. A avut parte de o creștere considerabilă a popularității în ultimii ani, devenind unul dintre cele mai mari sisteme de dezvoltare a jocurilor, pe lângă Unreal Engine [8] .

Pentru dezvoltarea aplicațiilor, limbajul de bază al acestui motor de dezvoltare este C#, dar a fost adăugată și o versiune de Java Script, numită Unity Script.

Interfață ușor de înțeles și ușurința cu care se pot dezvolta aplicațiile, au făcut ca Unity Engine să fie unul dintre cele mai populare și folosite motoare de dezvoltare.

După ce a fost creat un proiect, Unity îl va structura în scene. Fiecare scenă este ca un nivel de joc și este încărcată doar când este nevoie de ea. Scenele reprezintă proiectul în sine, deoarece în acestea se introduc obiectele și interfețele ce urmează a fi dezvoltate în aplicație. Scena care va fi responsabilă pentru aplicațiile ce folosesc realitate augmentată, trebuie să conțină o cameră AR specială, care poate capta și urmări

obiectele din lumea reală. Tot ce va vedea utilizatorul, este prin camera AR a aplicației, aceasta putând fi rotită și poziționată în funcție de nevoie.

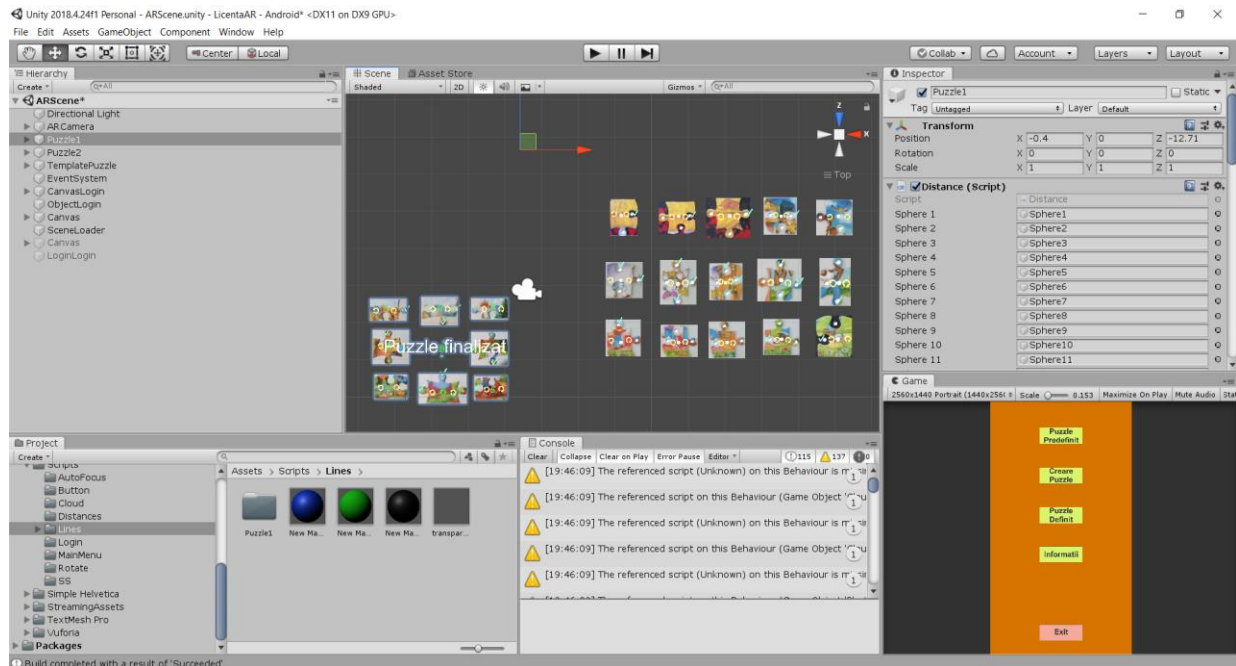


Figura 3.2– Unity [Proiect propriu].

4.2.7. Microsoft Visual Studio

A fost dezvoltat de către compania Microsoft, fiind un mediu de dezvoltare integrat, pentru programe de calculator, site-uri web, dar și al aplicațiilor mobile.

Un mediu de dezvoltare integrat, reprezintă un program ce cuprinde o multitudine de caracteristici, ce pot fi utilizate pentru diferite dezvoltări software. Poate suporta o multitudine de limbaje de programare cum ar fi: Java Script, C, C++, Python, dar limbajul care stă la baza acestuia este C# [14].

Conține anumite caracteristici ce îl fac un mediu de dezvoltare productiv, ușor de utilizat și performant [14] :

- Squiggles – Reprezintă o atenționare, care indică anumite erori ce au apărut în cod. Este folosit pentru a prezenta informații legate despre eroarea apărută, dar și a anumitor metode rapide care ar putea fi soluția rezolvării erorilor, fără a fi nevoie ca programatorul să identifice cauza, atunci când acestea sunt simple și ușor de gestionat de către program.
- Curățarea codului – Visual studio permite formatarea codului și curățarea acestuia. Astfel codul este mai vizibil și mai ușor de înțeles de către programator.

- IntelliSense- Are ca scop informarea programatorului despre anumite bucăți de cod ce stau la baza mediului de dezvoltare. De exemplu dacă dorim să folosim un anumit tip de dată, vor fi prezentate informații legate de acel tip de dată, cum ar fi: ce reprezintă acesta și în ce context poate fi folosită.
- Peek definition – Prezintă definiția unei anumite metode, fără a fi necesară deschiderea unui fișier separate. Astfel programatorul poate vedea cu ușurință ce reprezintă o metodă apelată în cod.

4.2.8. Android

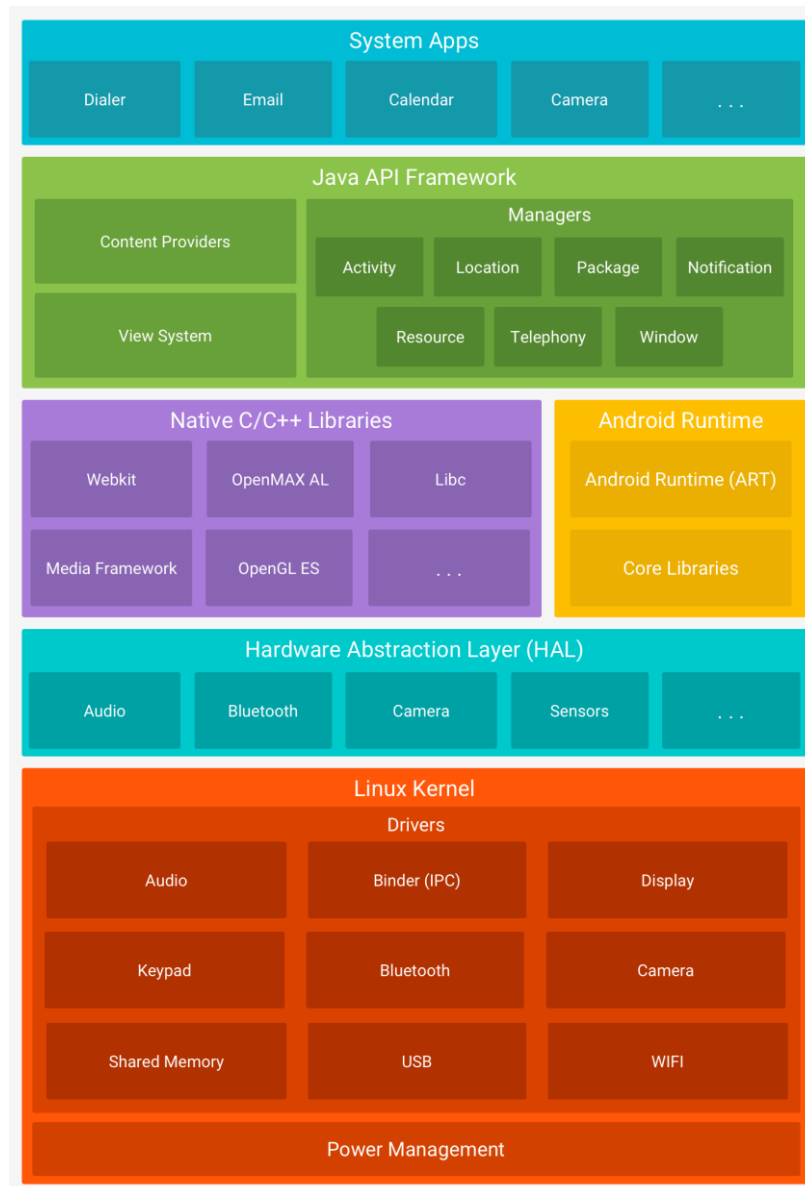


Figura 3.3– Arhitectura sistemului de operare Android. [22]

Arhitectura unui sistem Android cuprinde următoarele:

- Linux Kernel
- Strat de abstracție hardware (HAL)
- Android Runtime
- Livrării native C/C++
- Java API Framework
- Aplicațiile sistemului

Android, este un sistem de operare utilizat de un număr foarte mare de dispozitive inteligente, aproximativ 2,5 miliarde, cum ar fi: telefoane inteligente, tablete și chiar ceasuri inteligente [21]. Un sistem de operare, este un ansamblu de programe care lucrează împreună pentru a gestiona diferite activități și de-ai oferi utilizatorului o interfață accesibilă și ușor de utilizat. Acest sistem de operare este bazat pe nucleul Linux. Inițial a fost dezvoltat de către compania Android, dar în anul 2005 a fost cumpărată de către Google.

Codul sursă al acestui sistem de operare este disponibil gratuit, inițial fiind sub licența Apache. Deși este disponibil gratuit, acesta nu este în totalitate o sursă deschisă, unele părți ale SDK-ului încă sunt o sursă închisă.

Prima versiune oficială de Android a fost 1.0, lansată la data de 23 Septembrie 2008, permițând folosirea și descărcarea de aplicații, cu ajutorul unui magazin destinat acestora. A doua versiune a fost 1.1, lansată pe 9 februarie 2009, dar aceasta a adus puține îmbunătățiri față de cea precedentă. Versiunea 1.5, numită și Cupcake, a fost lansată în același an și a adus o mulțime de îmbunătățiri cum ar fi: predicția de text pentru tastaturi, înregistrarea de videoclipuri, abilitatea de a încărca filmulețe pe Youtube și multe altele. Următoarea versiune a fost 1.6 Donut, care a îmbunătățit accesul la cameră și galerie, dar a adus în plus și diferite gesturi ale sistemului. Eclair, a fost versiunea 2.0, care a perfecționat funcțiile deja existente, și a implementat suport pentru Bluetooth 2.1. A urmat versiunea 2.2, sau Froyo, aducând optimizări la performanță, memorie și viteză. Cele mai importante funcții noi ale acesteia sunt: suport pentru cloud, hot spot funcțional, suport pentru Adobe Flash și o îmbunătățire a securității. Gingerbread, a fost următoarea versiune, venind cu o interfață mult mai simplă și rapidă, dar și cu suport pentru ecranele de dimensiuni crescute, cu rezoluții mai mari. A urmat Honeycomb, care a fost optimizată pentru a suporta tabletele. Aceasta aduce cu ea sistemul de bară, care oferă acces rapid la anumite aplicații și funcționalități. Versiunea 4.0, sau Ice Cream Sandwich, a îmbunătățit și mai mult interfața, atât din punct de vedere vizual, cât și al performanței. A fost integrată funcția de captură a ecranului și abilitatea de a accesa aplicații din ecranul de blocare. Jelly Bean, versiunea 4.1, aduce din nou îmbunătățiri mari asupra interfeței. Această versiune permite extinderea notificărilor din bara de acces și îmbunătățiri ale camerei. Următoarea versiune a fost 4.4 KitKat. Interfața a fost complet schimbată față de versiunile precedente, aplicațiile putând fi instalate pe un card extern și a fost implementată și tehnologia NFC pentru prima dată. Acesta a adus o mulțime de îmbunătățiri și de optimizări, fiind un update foarte important al sistemului Android. După KitKat, a urmat Lollipop, aceasta suporta procesoare pe 64 de biți, OpenGL ES 3.1 și a fost îmbunătățit sistemul de colectare a datelor nefolositoare (garbage data). Pe data de 5 octombrie 2015, apare versiunea 6.0 numită Marshmallow, aducând o mulțime de optimizări, pentru folosirea procesorului, a memoriei, dar și pentru a reduce consumul de

energie, astfel încât bateria să dureze cât mai mult. A fost introdus un sistem de recunoaștere a amprentelor, suport pentru USB-C, dar și suport al rezoluțiilor 4K. Nougat, a fost lansată pe data de 22 august 2016, și avea abilitatea de a calibra culorile, de a deschide rapid ultima aplicație folosită și abilitatea de a da zoom în ecran. Ca la orice versiune nou apărută, performanțele și viteza au fost îmbunătățite. Android 8.0, numit și Oreo, a fost cea mai mare schimbare a arhitecturii sistemului. Au fost îmbunătățite și regândite majoritatea funcțiilor existente. Experiența folosirii unui dispozitiv cu un astfel de sistem devenind una diferită, dar în același timp mai bună. Android 9.0 Pie, aduce schimbări în interfață și optimizări față de versiunea precedentă. Ultima versiune oficial apărută, este Android 10, lansată pe data de 3 septembrie 2019. Aduce îmbunătățiri și performante mai ridicate decât versiunea anterioară, una dintre cele mai importante îmbunătățiri ale acesteia, fiind One UI 2.0, care schimbă modul în care poate fi folosită interfața sistemului [35] .

4.2.9. *Android Studio*

Este un mediu de dezvoltare integrat. Folosit pentru a dezvolta aplicații compatibile cu dispozitivele ce folosesc sistemul de operare Android și se bazează pe IntelliJ. A fost creat special pentru dezvoltarea aplicațiilor în mediul android, fiind optimizat pentru acest lucru.

Acesta conține o mulțime de caracteristici cum ar fi: un editor de aspect vizual, un analizator APK, ce este folosit pentru a reduce dimensiunea aplicației, un emulator rapid folosit pentru testarea aplicațiilor, fără a avea nevoie mereu de un dispozitiv fizic, un editor de cod inteligent și un sistem flexibil de construire [19].

Deși aplicația mea va fi dezvoltată în mediul Unity, este necesară instalarea acestui mediu de dezvoltare, deoarece vine însoțit de un SDK, care este esențial pentru dezvoltarea aplicațiilor mobile pentru Android. Un SDK este o trusă de dezvoltare, un set de unelte folosite pentru a implementa un program, pentru un anumit sistem de operare [24] .

4.2.10. *JSON*

JavaScript Object Notation, prescurtat JSON, reprezintă un schimb de date. Acesta este un format de tip text și poate converti obiecte în JSON, pentru a le putea trimite către un server, dar poate și să convertească orice JSON primit de la un server în obiecte, permițând lucrul cu date și obiecte mult mai ușor și eficient. Este conturat folosind două structuri generale pe care orice limbaj de programare îl poate susține [27]:

- O colecție de perechi
- O listă ordonată de valori

Aplicația în cauză, folosește formatul JSON pentru a comunica cu serverul de la Google, în cazul autentificării. Clientul se autentifică folosind contul de Google, acest lucru se face prin trimiterea unei cereri din partea aplicației, către

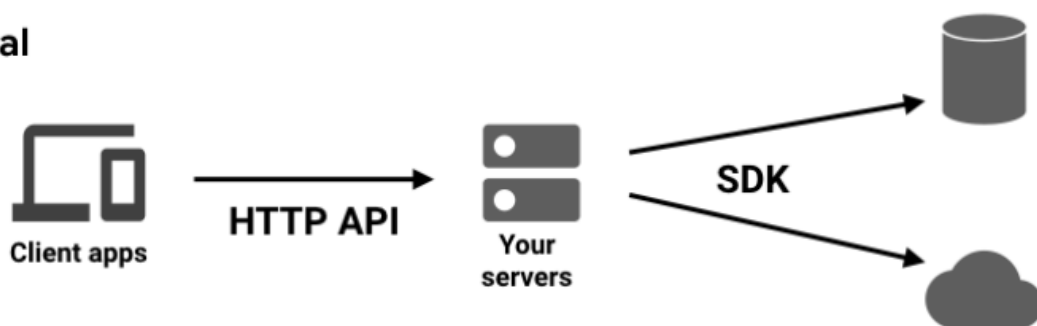
furnizorul de resurse. Dar folosește formatul și pentru trimiterea capturilor de ecran către baza de date [27].

4.2.11. Fire Base

Reprezintă un set de instrumente, care pot fi folosite pentru a îmbunătăți, sau adaugă diferite funcții noi. Oferă o multitudine de servicii, pe care dezvoltatorul le poate folosi, nefiind necesară implementarea lor directă de către programator. De exemplu se poate implementa autentificarea într-o aplicație, folosind sistemul acestora. Serviciile oferite de către firebase sunt găzduite în cloud [25].

O parte dintre aceste servicii de care dispune sunt: autentificare, cloud, bază de date în timp real, spațiu de stocare și machine learning.

Traditional



Firebase



Figura 3.4 – Diferența între o aplicație clasică și o una ce folosește firebase [26]

În imaginea de mai sus, se poate observa diferența de design între o aplicație clasică și una ce folosește sistemul firebase. Pentru a putea comunica cu o bază de date sau cu un serviciu de tip cloud, dezvoltatorii trebuie să implementeze un server propriu. Firebase scutește dezvoltatorii de acest lucru, iar legătura cu diversele necesități ale aplicației este făcută prin intermediul acestora.

Atunci când se dezvoltă o aplicație, care va fi folosită de o multitudine de utilizatori, este necesar să se cunoască identitatea acestora. Astfel firebase oferă servicii de backend și SDK-uri, pentru ca utilizatorii să se poate conecta cu contul lor propriu, datele acestora fiind ținute în siguranță[26].

4.2.12. Firebase Autentificare

Este o platformă dezvoltată pentru a crea aplicații rapid, fără a fi nevoie de a implementa anumite funcții în plus. Acesta se ocupă de back end, permițând funcții de autentificare, o bază de date și alte servicii cloud. La bază este dezvoltată pe o infrastructură Google, permițând o comunicare foarte bună a aplicațiilor cu serviciile oferite de Google [25].

Autentificare, folosind Firebase este creată pentru a oferi securitate aplicațiilor, și o modalitate simplă de autentificare a utilizatorilor cu diferite conturi, Google, Facebook, GitHub, dar suportă și conturi cu email și parola [25].



Figura 4.3 – Autentificare folosind firebase [25].

4.2.13. API

Reprezintă un nivel al versiunii de Android, care facilitează compatibilitatea platformei cu diverse aplicații. Fiecare versiune de Android are un număr de API diferit, astfel nu toate aplicațiile dezvoltate pentru această platformă sunt compatibile cu toate versiunile. Atunci când se dezvoltă o aplicație, trebuie selectată o versiune minimă de API și una maximă, pentru a defini pe ce versiune de Android poate rula acesta. Dacă numărul API nu este compatibil, atunci nu se va putea instala aplicația dorită, sau vor apărea diferite erori [24].

Un API este alcătuit dintr-un:

- Set de pachete și clase.
- Set de elemente și atribute XML.
- Set de intenții.
- Set de permisiuni.

Când o aplicație este în dezvoltare, setarea unui anumit nivel de API înseamnă că aceasta a fost testată și nu au apărut probleme de compatibilitate.

Odată cu evoluția și dezvoltarea sistemului Android, când apare o versiune nouă, și nivelul API este actualizat. Dacă un dispozitiv are ultimul nivel de API disponibil, acesta este compatibil cu aplicații ce au un nivel mai scăzut, adică poate suporta versiunile mai vechi de API [24].

Capitolul 5. Proiectare de Detaliu și Implementare

5.1. Arhitectura sistemului

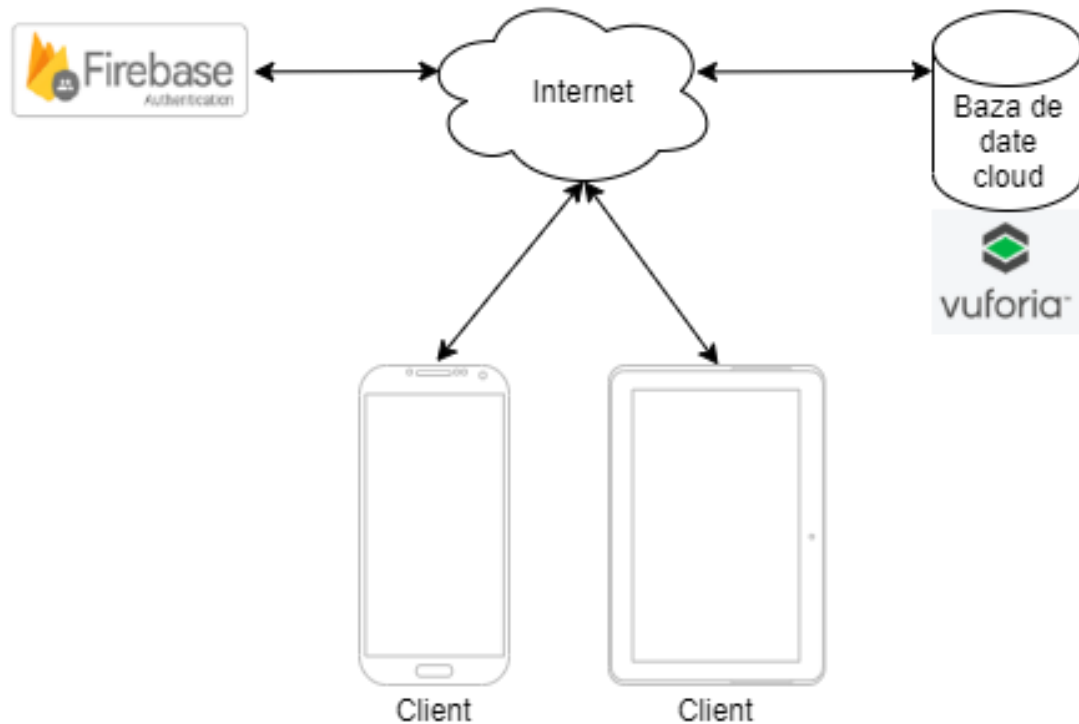


Figura 5.1 – Arhitectura generala a sistemului.

1. **Clientul** – Acesta reprezintă un utilizator de dispozitiv inteligent care rulează cu sistemul de operare Android. Clientul are instalată aplicația pe dispozitiv, iar pentru ca acesta să funcționeze corect, necesită conexiune la internet pentru a se putea conecta la contul Google și pentru a putea predefini un puzzle propriu.
2. **Internetul** – Este calea prin care aplicația comunică cu componentele necesare, prin acesta se face transferul de date.
3. **Baza de date cloud** – Aceasta aparține de Vuforia, care este și kit-ul folosit pentru recunoașterea imaginilor și pentru aplicarea obiectelor și a interacțiunilor virtuale. Facilitatea acestei baze de date, este faptul că nu mai este necesar ca imaginile să fie introduse manual în proiect, pentru ca mai apoi să fie folosite, ci după ce clientul a încărcat imaginea dorită în baza de date cloud, aplicația o va putea recunoaște imediat ce aceasta a fost procesată. Astfel funcția de adăugare a unui puzzle de către utilizator este posibilă.

4. **Firestore Authentication** – Sistemul Firestore permite clienților să se autentifice în aplicație cu contul Google, cu care sunt conectați la dispozitivul inteligent.

5.1.1. Diagrame de secvențe

1. Puzzle predefinit

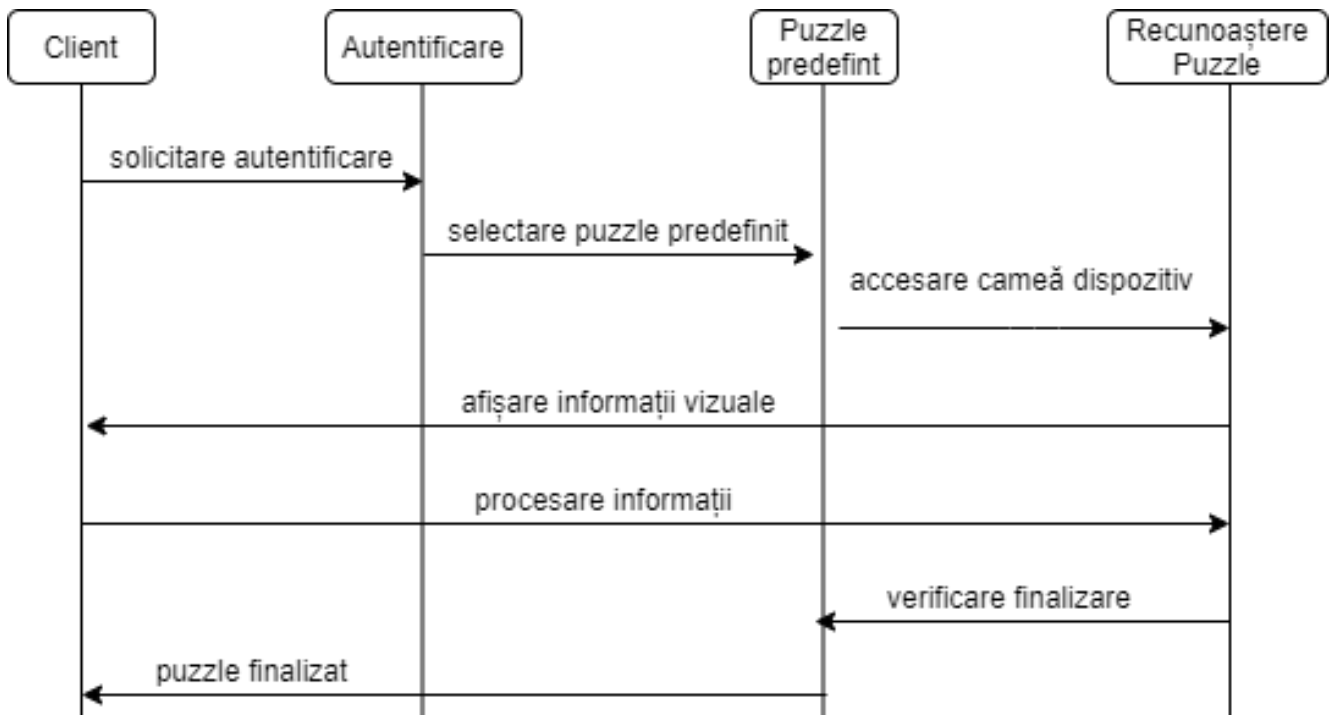


Figura 5.2 – Diagrama de secvențe pentru un puzzle predefinit.

Această diagramă de secvență reprezintă scenariul în care un utilizator alege să rezolve un puzzle ce este predefinit în aplicație. Clientul solicită autentificarea cu contul Google, iar după ce a fost autentificat, din meniul principal va alege opțiunea de puzzle predefinit. Se alege un puzzle din lista celor disponibile, iar aplicația va accesa camera telefonului și va trimite informațiile vizuale către client. Utilizatorul va compara piesele și aplicația va procesa datele necesare. După un anumit număr de pași se verifică dacă puzzle-ul este finalizat, dacă se confirmă acest lucru, utilizatorului i se va indica faptul că puzzle-ul este complet, prin afișarea unui mesaj virtual.

- Autentificare.
- Selectare opțiune puzzle predefinit.
- Accesarea camerei dispozitivului.
- Recunoașterea imaginilor.

- Afișarea informațiilor virtuale.
- Procesare informații.
- Verificare informații.
- Finalizare puzzle.

2. Definire puzzle

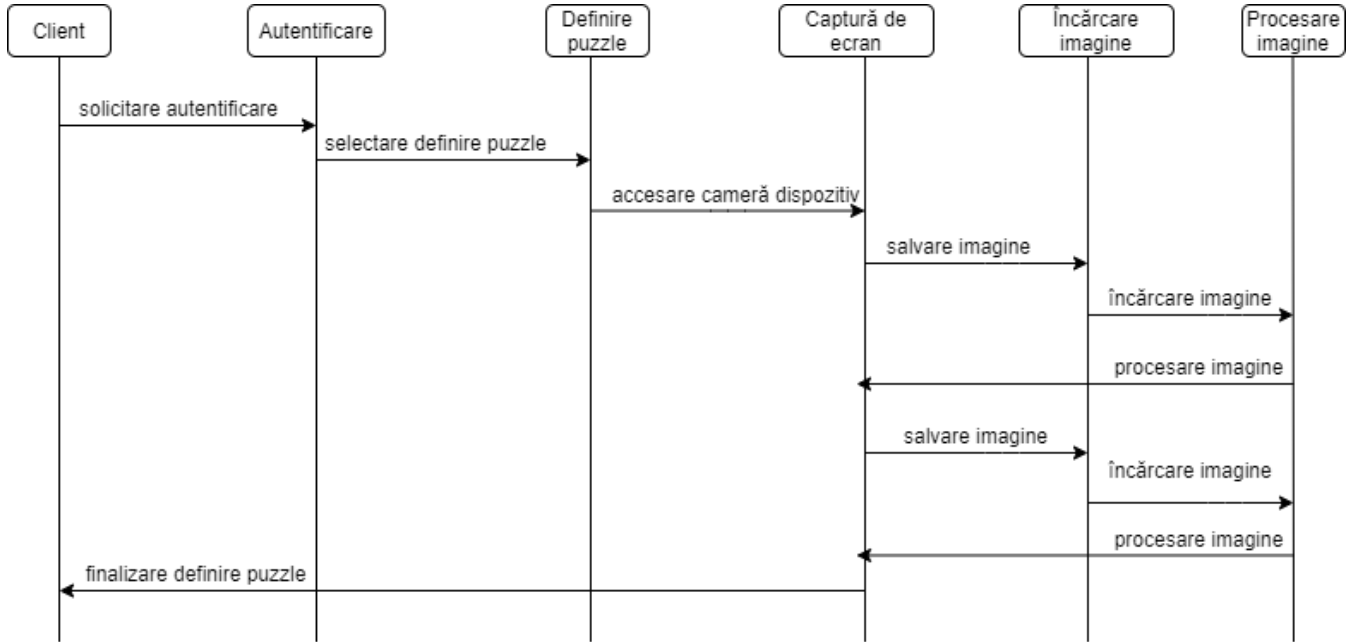


Figura 5.3 – Diagrama de secvențe pentru definirea unui puzzle.

În această diagramă de secvențe, este prezentat scenariul în care un utilizator își definește propriul său puzzle. Acesta se autentifică în aplicație cu contul Google și solicită opțiunea de definire puzzle. Apoi este redirecționat către meniul de definire, unde aplicația accesează camera dispozitivului pentru a putea face capturi de ecran. Utilizatorul va încadra piesa în zona delimitată și va apăsa pe butonul foto pentru a salva imaginea. După ce a fost salvată, acesta va apăsa butonul de încărcare, care va trimite imaginea către baza de date cloud. Ajunsă în baza de date imaginea va fi procesată și pregătită pentru a putea fi recunoscută în aplicație. Utilizatorul repetă acești pași până când va încarcă toate piesele necesare unui puzzle. Aplicația îi va indica utilizatorului că a terminat de definit puzzle-ul și îl va direcționa către meniul principal.

- Autentificare.
- Selectare definire puzzle.
- Accesare cameră dispozitiv.
- Salvarea imaginii.
- Încărcarea imaginii în baza de date.

- Procesarea imaginii.
- Finalizare definire puzzle.

5.2. Baza de date

Pentru acest proiect am ales să folosesc baza de date deținută de Vuforia, deoarece în această bază de date imaginile sunt deja procesate de aplicație și pot fi recunoscute. Pentru a putea avea acces la acesta, trebuie creat un cont și mai apoi creată o bază de date. Fiecare bază de date va avea o cheie unică.

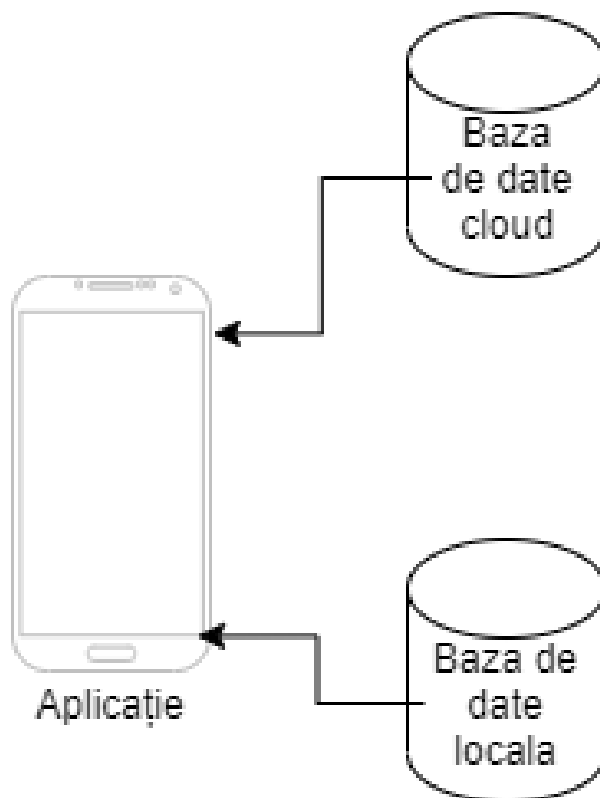


Figura 5.5 – Diagrama bazei de date

5.2.1. Baza de date locală

Aceasta reprezintă baza de date ce se află salvată în aplicație. După ce a fost creată pe site-ul celor de la Vuforia, se pot încărca imagini în acesta. O dată încărcate, acestea sunt procesate și pregătite pentru a fi identificate. Fiecare bază de date poate susține până la o mie de imagini. Pentru a o putea utiliza, aceasta trebuie descărcată și introdusă manual în aplicație. Odată adăugată, imaginile pot fi folosite fără probleme.

Licenta [Edit Name](#)
Type: Device

Targets (24)

Add Target

Download Database (All)

<input type="checkbox"/> Target Name	Type	Rating ①	Status ▾	Date Modified
<input type="checkbox"/> Puzzle7	Single Image	★★★★★	Active	May 31, 2020 19:02
<input type="checkbox"/> Puzzle3	Single Image	★★★★★	Active	May 31, 2020 19:02
<input type="checkbox"/> P2Img2	Single Image	★★★★★	Active	May 31, 2020 19:01
<input type="checkbox"/> P2Img1	Single Image	★★★★★	Active	May 31, 2020 18:57
<input type="checkbox"/> P2Image15	Single Image	★★★★★	Active	May 31, 2020 18:07
<input type="checkbox"/> P2Image14	Single Image	★★★★★	Active	May 31, 2020 18:07
<input type="checkbox"/> P2Image13	Single Image	★★★★★	Active	May 31, 2020 18:06
<input type="checkbox"/> P2Image12	Single Image	★★★★★	Active	May 31, 2020 18:06
<input type="checkbox"/> P2Image11	Single Image	★★★★★	Active	May 31, 2020 18:06
<input type="checkbox"/> P2Image10	Single Image	★★★★★	Active	May 31, 2020 18:06

Figura 5.4 – Bază de date locală

5.2.2. Baza de date cloud

Acesta este susținută de către cei de la Vuforia și nu trebuie introdusă manual în aplicație. Imaginile pot fi încărcate atât manual pe site-ul lor, dar și direct din aplicație dacă se implementează funcționalitatea necesară. Este folosită atunci când se dorește definirea unei noi imagini dinamic, fără a fi necesar să introducem din nou o bază de date manual și să configurăm imaginea static. Cel mai mare dezavantaj al acesteia, este faptul că poate recunoaște doar o singură imagine, nu se pot recunoaște mai multe imagini simultan.

vuforia engine™
developer portal

Home Pricing Downloads Library **Develop** Support

Hello Cuci ▾ | Log Out

License Manager Target Manager

Target Manager > Cloud

Cloud [Edit Name](#)
Type: Cloud
License Key: CuciLicenta

Targets (2) [Database Access Keys](#)

Add Target

Search by target name or target ID

<input type="checkbox"/> Target Name	Rating ①	Recos ▾	Status ▾	Date Modified ▾
<input type="checkbox"/> P0001	★★★★★	0	Active	Sep 04, 2020 14:54
<input type="checkbox"/> P0000	★★★★★	25	Active	Sep 04, 2020 14:38

Figura 5.4 – Bază de date cloud

5.3. Structura proiectului

Deși nu pare un lucru foarte important, structura unui proiect trebuie să fie una bună încă de la început. Aceasta poate economisi timp, să fie eficientă și mai ușor de modificat. La început, aplicația nu poate părea foarte mare, sau foarte complicată, dar cu dezvoltarea ei vor apărea tot mai multe componente și va deveni din ce în ce mai mare, astfel o proiectare bună a structurii de la început, poate face ca extinderea proiectului să fie ușoară fără prea mari bătaii de cap.

5.3.1. Structura de dosare

Unity este un sistem ce se bazează pe această structură. Are un folder rădăcină numit „Assets”, în care se află restul structurii proiectului. Deoarece Unity oferă libertate maximă în organizarea structurii, acesta poate deveni foarte ușor dezorganizată și complicată. Este bine să nu se stocheze fișierele folosite în directorul rădăcină, ci trebuie create subdirectoare suplimentare care să le conțină. Denumirea acestora trebuie să fie potrivită și consecventă. Astfel este indicat să se folosească subdirectoare, de exemplu pentru materialele create să existe un subdirector, pentru imaginile stocate să existe altul, pentru fișierele ce conțin cod și funcții altul și tot așa.

5.3.2. Structura scenei

Pe lângă Structura de dosare Unity are și o ierarhie a scenei. Acesta conține toate obiectele ce sunt instanțiate în scenă. De exemplu în cazul proiectului meu, am două puzzle-uri predefinite, fiecare puzzle este un obiect ce conține la rândul lui mai multe obiecte, care la rândul lor conțin și ele obiecte. Pe lângă obiecte, în scenă se mai găsește și cameră care face posibilă captarea imaginilor din scenă. Ca și în cazul de mai sus, o bună structurare de la început a scenei, o poate face mai ușor de înțeles și modificat dacă este nevoie.

5.4. Diagrama de pachete

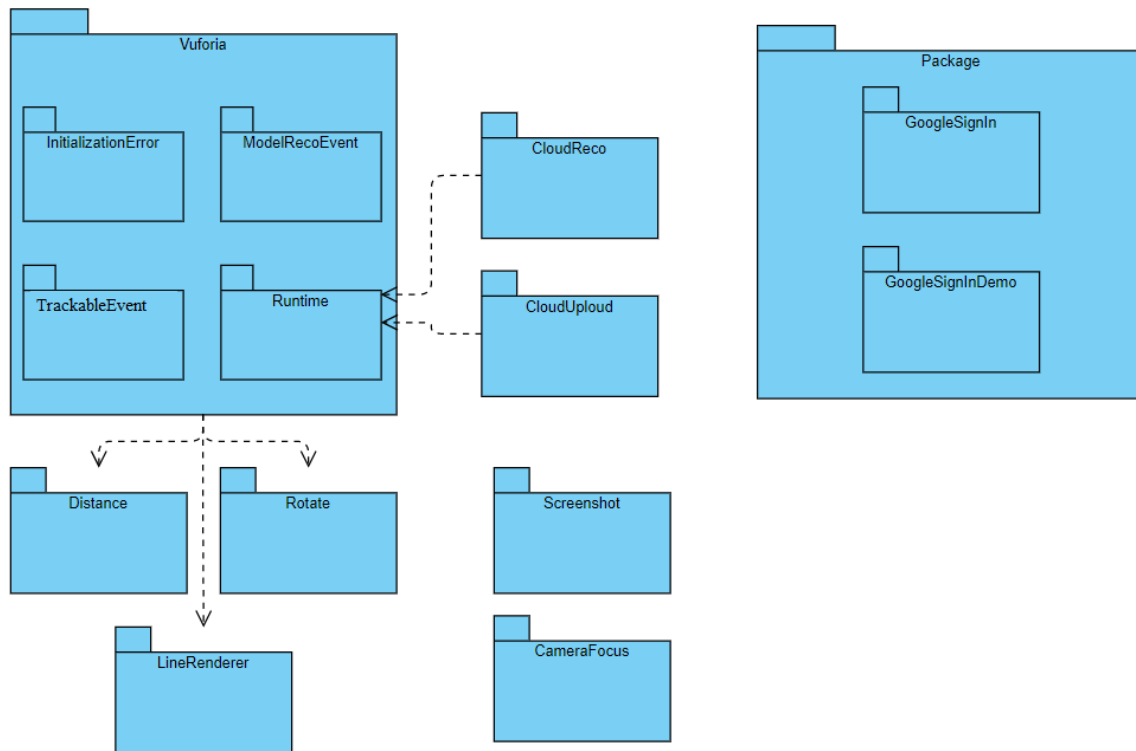


Figura 5.5 – Diagrama de pachete

Arhitectura acestui proiect este structurată pe mai multe pachete, care conțin diferite clase și funcționalități.

5.4.1. Pachetul Vuforia

Acesta conține clasele pe care le folosește kitul Vuforia pentru a funcționa corect.

`InitializationError`, este clasa responsabilă de tratarea erorilor. Atunci când aplicația este pornită, această clasă se inițializează și verifică diferite aspecte și returnează următoarele erori dacă este cazul:

- Dacă dispozitivul nu este detectat, sau nu conține un hardware minim.
- În cazul în care cheia de licență lipsește din proiect, nu se va putea face legătura între Vuforia și aplicație.
- Dacă cheia de licență nu este validă.
- Dacă nu există conexiune la internet, nu se va putea comunica cu serverul.
- În cazul în care cheia de licență a fost anulată.
- Dacă nu a fost permis accesul la cameră.
- Dacă dispozitivul nu este compatibil cu Vuforia.

`ModelRecoEvent`, este folosită pentru a afișa diferite erori în interfața aplicației UI. Cu ajutorul acestei clase, putem spune când Vuforia ar trebui să caute noi ținte și când ar trebui să se oprească.

TrackableEvent, este responsabilă pentru găsirea și pierderea unei ținte. Acesta conține trei metode principale. Prima este OnTrackingFound(), adică dacă o țintă a fost găsită, se vor inițializa componente ale acesteia. InTrackingLost, anunță că ținta a fost pierdută și îi dezactivează toate componentele acesteia. OnTrackavleStateChange, informează asupra situației unei ținte, dacă a fost recunoscută, sau pierdută de către aplicație. Acesta indică statusul în care a fost ținta și cel în care este acum, dar și informații despre țintă cum ar fi: dacă este detectată, dacă este urmărită, sau dacă a fost pierdută.

RuntimeOpensourceInitializer, atunci când programul este compilat, această clasă face legătura corect între Unity și Vuforia, pentru a funcționa între ele, dar face și legătura pentru dispozitivele Android. Aceasta cere permisiuni și se ocupă de camera ce randează imagini.

Acest Pachet, este legat și la câteva clase externe, care se folosesc de Vuforia.

CloudReco, este clasa ce face legătura între aplicație și baza de date cloud. Aceasta se inițializează la deschiderea aplicației și verifică dacă sunt ținte compatibile cu cele din baza de date. Dacă se găsește o țintă ce corespunde unei texturi, o va recunoaște, dar dacă mai adăugăm în scenă o imagine care știm siguri că aparține de baza de date cloud, nu va reuși să o identifice, deoarece poate recunoaște doar o imagine simultan. Pentru a putea recunoaște alta țintă, este nevoie să piardă țintă curentă, pentru a putea începe din nou căutarea.

CloudUploading, este clasa folosită pentru a trimite imaginile la baza de date cloud. Acesta conține cheile de acces ale imaginii, dar și numele țintei pe care dorim să o încărcăm.

Distanța, în această clasa sunt instanțiate obiectele 3D folosite pentru a calcula distanțe. Fiecărei imaginii, îi sunt atribuite câteva obiecte virtuale. Pe baza acestora se pot calcula distanțe reale, deoarece nu se poate calcula distanța dintre două imagini recunoscute altfel. Aici sunt, verificate conexiunile pieselor și tot aici este verificat dacă puzzle-ul este finalizat.

Rotate, conține la rândul ei anumite obiecte 3D, pe care le folosește pentru a determina poziția piesei recunoscute. Clasă responsabilă de afișarea poziției și indicarea informațiilor necesare, pentru ca utilizatorul să poată poziționa corect piesele.

LineRenderer, se ocupă de indicarea pieselor ce trebuie conectate între ele. Atunci când două piese sunt recunoscute, între ele se va afișa o linie, ce indică faptul că aceste piese trebuie conectate una de alta.

Cele două clase separate, Screenshot și CameraFoscus, sunt folosite independent de Vuforia. Clasa CameraFocus are rolul de a focusa automat pe diferite ținte. A fost necesară implementarea acesteia, deoarece camera în timpul folosirii aplicației nu focusa cum trebuie. Clasa Screenshot, este responsabilă de salvarea capturilor de ecran. Atunci când se definește un puzzle propriu, trebuie create poze ale țintelor dorite.

5.4.2. Pachetul SingIn

Acesta este responsabil pentru autentificarea utilizatorului cu contul Google. Conține două clase importante, GoogleSignInDemo și GoogleSignIn.

GoogleSignInDemo, este inițializat atunci când se cere autentificarea în aplicație, prima dată verifică legătura cu Firebase și dacă se poate face. Dacă legătura a fost realizată cu succes, se va returna lista cu conturile Google ale utilizatorului, iar dacă este valid contul ales, se vor cere permisiunile necesare și se va afișa numele și adresa de email a contului respectiv, autentificarea realizându-se cu succes.

GoogleSignIn, este clasa ce implementează GoogleSignInAPI pentru Unity. Conectarea la Google simplifică sistemul de autentificare al aplicației, permite utilizatorului să se conecteze cu un cont Google deja existent, fără a mai fi necesară o înregistrare nouă. Prin această conectare, este permis accesul la serviciile Google.

5.5. Funcționalități

5.5.1. Autentificare

Fiecare utilizator trebuie să poată fi recunoscut ca fiind unic.

Am decis pentru această aplicație să folosesc sistemul celor de la Firebase. Atunci când un utilizator va deschide pentru prima dată aplicația, pentru a putea accesa meniul, trebuie să se autentifice cu contul Google. Acesta se va afișa automat când se va apăsa butonul specific.

Inițial trebuie creat un cont pe site-ul celor de la Firebase, în care se va crea un nou proiect, pentru care se va alege opțiunea Android. Pentru a conecta proiectul la Firebase, va trebui pus același nume de proiect, astfel stabilindu-se legătura. Este nevoie de o cheie numită SHA-1, care este o funcție hash criptografică, ce preia o intrare și produce o valoare hash. După ce proiectul a fost setat, este necesară introducerea pachetelor de autentificare, care vor permite legătura cu Google, pentru a cere conturile cu care este autentificat utilizatorul. În proiect mai trebuie atașat un fișier numit „google-services.json”, ce conține detaliile de configurare cum ar fi: cheile de acces pentru serviciul de autentificare.

Serviciul Google de autentificare folosește OAuth 2.0. Acesta este o funcție care permite accesul autorizat la API-urile Google.

5.5.2. Recunoaștere imaginii

Primul pas în crearea și dezvoltarea unei aplicații cu realitatea augmentată, este reușirea recunoașterii, identificării a unei imagini, obiect, sau a unei zone din lumea reală, pentru a putea aplica lumea virtuală peste aceasta.

Pentru acest lucru am ales să folosesc kitul de la Vuforia, care se integrează destul de ușor în Unity. Pentru acest proiect folosesc versiune 2018.4.24f1 de Unity, în care am instalat pachetele de la Vuforia. Pentru a putea recunoaște imaginile, acestea trebuie adăugate în baza de date de la Vuforia. Imaginile odată încărcate, sunt procesate și li se atribuie un scor, pe baza acelui scor știm dacă se pot recunoaște ușor, sau mai greu. Apoi se descarcă aceasta baza de date pentru Unity. În Unity se va crea o scena, care va conține o cameră numită ARCamera, folosită pentru a identifica și recunoaște imaginile. În scenă se va adăuga din opțiunile de la Vuforia ImageTarget, care reprezintă baza pe care se va pune o imagine din baza de date. După ce am atribuit o imagine, pentru a testa faptul că este recunoscută, vom adăuga un obiect 3D, pe care îl atribuim imaginii. Astfel

în momentul în care aplicația va rula, când imaginea este identificata peste acesta se va afișa obiectul 3D. Dacă se întâmpla acest lucru știm ca aplicația funcționează corect.

5.5.3. Conexiune piese

Pentru a rezolva un puzzle este necesar să știm dacă piesele acestuia sunt conectate între ele corect. Aplicație trebuie să știe să facă acest lucru pentru ai confirma utilizatorului ca a conectat piesele potrivite.

Inițial, am decis să adaug cate un obiect 3D pentru fiecare imagine. Acesta era poziționat în mijlocul piesei de puzzle, iar pentru a verifica dacă două piese sunt conectate, am calculat distanța între obiecte. Astfel pentru a ști dacă prima piesă de puzzle este conectată cu cea dea doua, calculez distanța dintre obiectul de pe prima și obiectul de pe a doua. Când distanța ajunge să fie mai mică de o anumită valoare, aplicația considera că piesele sunt conectate și va afișa un obiect care îi va indica utilizatorului că piesele se potrivesc și sunt conectate corect. Acest obiect este vizibil doar când distanța dintre piese este foarte mică. După câteva teste, au apărut niște probleme ale implementării. Deoarece am ales să adaug doar un obiect în mijlocul piesei, aplicația îmi recunoștea conexiunea între două piese, în orice poziție ar fi fost conectate acestea.

Pentru a rezolva problemele apărute, am decis să adaug cate un obiect pentru fiecare conexiune pe care o poate avea o piesă. De exemplu piesa din coltul din stânga sus, poate avea două conexiuni, astfel pentru fiecare din ele am adăugat un obiect pentru ca distanța ce este calculată să fie mult mai mică, astfel nu va mai apărea obiectul de verificare în orice poziție conectăm piesele.

```
float distance = Vector3.Distance(sphere1.transform.position, sphere3.transform.position);

/*Afișare dinstanta pe ecran*/
float walid = distance / 10;
guiText = walid.ToString();

Cube.SetActive(false);
if (distance > 0.5)
{
    Cube.SetActive(false);
}
else if (0.3 > distance)
{
    Cube.SetActive(true);
}
```

Figura 5.6 – Exemplu de calcul al distanței.

5.5.4. Legătură piese

Deoarece aplicația trebuie să indice utilizatorului cum să rezolve un puzzle, am decis să implementez o funcționalitate care îmi va arăta legătura dintre piese.

Atunci când aplicația va recunoaște două piese, ce trebuie conectate una lângă alta, se va afișa o linie între cele două piese. Acesta se va mări sau micșora în funcție de distanța dintre piese.

Ca să implementez această funcționalitate, am folosit o componentă din Unity numită „Line Renderer”. Acesta are în compoziția ei cel puțin două puncte în spațiul 3D și este folosită pentru a desena linii între diferite obiecte. Pentru a adăuga

această componentă, se accesează efectele din Unity și se selectează linia. Apoi se pune pe obiectul dorit această componentă și se implementează codul ei de funcționare. În cazul meu linia apare între două puncte în funcție de distanța dintre ele.

```
lr.SetPosition(0, Positions[0].position);
lr.SetPosition(1, Vector3.MoveTowards(transform.position, Positions[1].position, speed * Time.time));

float distance = Vector3.Distance(Cylinder.transform.position, Cylinder1.transform.position);
if (distance > 1 && distance < 5)
{
    // draw();
}
else if (0.7 > distance)
{
    lr.enabled = false;
}
Debug.Log(distance);
```

Figura 5.6 – Exemplu de calcul al liniei

5.5.5. Poziție piese

Fiind vorba de un puzzle, atunci când piesele sunt scoase din cutie și împrăștiate pe o suprafață, cel mai probabil acestea vor fi rotite în diferite direcții.

Am decis să implementez o funcție pentru aplicație, care va indica în ce direcție trebuie să rotească utilizatorul piesa pentru ca aceasta să ajungă în poziția corectă față de cameră.

Pentru fiecare imagine a unui puzzle, am adăugat un obiect 3D în centrul acestora. Acesta este poziționat drept și perpendicular față de camera din scenă. Poziția acestuia indică dacă piesa este rotită sau nu, deoarece obiectul se rotește odată cu imaginea. Verificarea se face folosind unghiul la care este rotit obiectul față de centrul său. Dacă depășește un anumit unghi, se va afișa un obiect sub forma de săgeată, ce se rotește în direcția în care trebuie rotită și piesa de puzzle. Am adăugat două obiecte, unul care se rotește la stânga și unul care se rotește la dreapta, pentru ca utilizatorul să vadă mai ușor direcția de rotire. Când pe piese nu se observă acești indicatori, înseamnă că este deja în poziția corespunzătoare față de cameră și nu mai este necesară nici o modificare a acesteia.

```

Indicator3.transform.Rotate(0, 0, speed * -Time.deltaTime);
Indicator4.transform.Rotate(0, 0, speed * -Time.deltaTime);
if (Cylinder1.transform.rotation.eulerAngles.y >= 30 && Cylinder1.transform.rotation.eulerAngles.y <= 180)
{
    Indicator3.SetActive(true);
}
else
{
    Indicator3.SetActive(false);
}
if (Cylinder1.transform.rotation.eulerAngles.y >= 180 && Cylinder1.transform.rotation.eulerAngles.y <= 330)
{
    Indicator4.SetActive(true);
}
else
{
    Indicator4.SetActive(false);
}

```

Figura 5.7 – Exemplu de calcul al poziției.

5.5.6. Captură ecran

Pentru a putea crea o metodă prin care un utilizator să își definească propriul puzzle, este necesar ca acesta să poată captura imagini ale puzzle-ului dorit.

Am ales să implementez o funcție care reușește să facă o captură de ecran și să salveze imaginea atât în editor, cât și în memoria telefonului. Pentru a nu fi probleme cu baza de date în care urmează să fie încărcate imaginile, am ales să le salvez ca și format JPG.

Funcția, prima dată, va căuta directorul în care trebuie salvate capturile de ecran, dacă acesta nu există îl va crea. După ce directorul este găsit, imaginile vor fi salvate sub numele de „P0000”, număr care se incrementează de fiecare dată când se salvează o captură nouă. Imaginile sunt văzute de program ca niște texture, definite în cod C# sub forma Texture2D.

Atunci când se apasă butonul specific, se va face o captură a ecranului în acel moment, iar acestea i se va atribui numele ce urmează și va fi salvată.

5.5.7. Încărcare imagini

Metoda prin care utilizatorul își definește un puzzle propriu, trebuie să poată să trimită imaginile salvate de acestea către o baza de date de la Vuforia, pentru a putea fi procesate.

Am decis să folosesc o bază de date cloud, care este pusă la dispoziție de către Vuforia. Cu ajutorul acesteia se pot încarca imagini și mai apoi pot fi recunoscute, fără a fi necesară o predefinire în aplicație. Această funcție va căuta după nume fiecare imagine, de exemplu pentru imaginea „P0000”, este un fișier care va căuta denumirea aceasta, iar dacă o găsește o va trimite mai departe. Prima dată se verifică textura imaginii, dacă este RGB nu se aplică modificări, dar dacă este de alt tip, se va modifica să fie RGB, după care se face o cerere către server. Se va verifica numele imaginii, dacă se regeste în baza de date, dacă nu se regăsește, se va putea adăuga, în caz contrar, se va returna o eroare. Fiecare imagine după ce a fost încărcată va avea un id propriu și un nume unic, care sunt folosite pentru a le diferenția. În această funcție, trebuie specificate și cheile de acces ale bazei de date.

Fiecare bază de data cloud are un set de chei de acces unice, ce trebuie specificate, ca aplicația să poată comunica cu aceasta.

5.5.8. Recunoaștere imagini din baza de date cloud

Dacă imaginile au fost salvate și încărcate cu succes în baza de date cloud, aplicație trebuie să poată recunoaște aceste imagini.

Pentru această parte, am ales să folosesc un exemplu de la cei de la Vuforia care reușește să identifice imaginile și să le găsească în baza de date. În scena din aplicație, trebuie adăugat un „CloudReco”, ce servește ca bază pe care vor fi aplicate imaginile, dar și obiectele virtuale necesare. Identificarea imaginilor se face după texturile acestora, astfel când se găsește o textură, se compară cu cele din baza de date, iar dacă se returnează că a fost găsită, aplicația va recunoaște imaginea.

5.5.9. Finalizare puzzle

Utilizatorul trebuie anunțat atunci când puzzle-ul a fost finalizat cu succes.

Am decis să fac acest lucru, prin afișarea unui text virtual peste puzzle, în momentul în care aplicația recunoaște că ultima piesă a fost adăugată cu succes. De exemplu la un puzzle de 9 piese, ultima piesă este considerată cea din mijloc, iar aplicația deja calculează distanțele între piese pentru a verifica conexiunea. Deci când acele conexiuni sunt corecte, se va afișa mesajul, astfel utilizatorul va ști că puzzle-ul este terminat.

Capitolul 6. Testare și Validare

6.1. Testare funcționalități

Pentru a fi siguri că o funcție pe care dorim să o adăugăm proiectului funcționează în parametrii normali. În acest caz fiecare dintre următoarele funcționalități au fost mai întâi testate și pe urmă adăugate în proiectul final.

6.1.1. Recunoaștere imagini

Primul pas în realizarea unei aplicații ce folosește realitatea augmentată, este acela de testare a kitului folosit pentru recunoașterea imaginilor. Trebuie să vedem câte imagini poate recunoaște simultan, de la ce distanță și cât de ușor. Astfel am implementat în proiect kitul de la Vuforia, care a întrunit toate cerințele necesare.



Figura 6.1 – Testare recunoaștere piese de puzzle

În figura 6.1, se poate observa că aplicația reușește să recunoască cu succes mai multe piese de puzzle simultan și să aplice indicatorii virtuali peste acestea.

6.1.2. Verificare conexiune

Pentru a putea rezolva un puzzle corect, este nevoie să cunoaștem când două piese sunt conectate corect între ele. Pentru acest lucru este nevoie să știm piesele care le recunoaștem și ce distanță este între ele.



Figura 6.2 – Testare conexiune între piese

După cum se poate observa în imaginea de mai sus, conexiunea corectă dintre piesele de puzzle a fost testată cu succes. Astfel atunci când conectam două piese și va apărea indicatorul de verificare, știm că acestea se potrivesc și se poate trece la următorul pas.

6.1.3. Verificare poziție

Un puzzle conține mai multe piese, iar când acestea sunt împrăștiate, cel mai probabil vor fi rotite în poziții diferite. Pentru a rezolva această problemă, se va testa poziția piesei față de camera telefonului. Astfel dacă piesa este rotită, se va afișa un indicator, care se rotește în direcția în care trebuie rotită și piesa, pentru a ajunge la poziția corectă față de cameră.



Figura 6.3 – Testare poziție piesă

6.1.4. Verificare linie între piese

Pentru a ști care două piese trebuie conectate între ele, se va afișa o linie continuă între acestea, care indică utilizatorului, care piesă trebuie conectată cu care.

Am testa această funcție, dar din păcate apare un bug ce nu am reușit să îl rezolv. Linia ar trebui să apară doar când două piese de puzzle ce trebuie conectate sunt recunoscute de aplicație, dar în unele cazuri linia apare și dacă este recunoscută o singură piesă.



Figura 6.4 – Testare legătură piese

După cum se poate observa în imaginile de mai sus, în prima este prezentat modul în care linia nu funcționează corect, totuși când a două piese este identificată, legătura se afișează și funcționează cum trebuie. Aceasta apare din motive necunoscute și nu la toate piesele.

6.1.5. Realizare captură ecran

Una din funcțiile aplicației este definirea unui puzzle de către utilizator. Pentru a putea fi posibil acest lucru, este necesară o captură de ecran cu fiecare piesă din puzzle-ul ce se definește.

Am testat salvarea capturii de ecran atât în editor, Unity, cât și pe aplicația mobilă, astfel pe ambele platforme imaginile sunt salvate cu succes. Imaginile se salvează cu un nume unic ce se incrementează după fiecare imagine salvată.

6.1.6. Încărcare imagine în baza de date

Când utilizatorul definește un puzzle nou, pentru ca imaginile salvate de acesta să fie recunoscute, acestea trebuie încărcate în baza de date cloud, pentru a fi procesate.

Proiectul a fost testat cu succes în editor. Capturile de ecran făcute pentru definirea puzzle-ului, pot fi încărcate imediat în baza de date cu succes. În schimb pentru aplicația mobilă se pot încărcă imaginile, dar doar dacă acestea au fost făcute în editor, imaginile salvate în memoria telefonului nu pot fi încărcate.

The screenshot shows the 'vuforia engine™ developer portal' with a navigation bar including Home, Pricing, Downloads, Library, Develop, and Support. The 'Develop' tab is active, and the 'Target Manager' sub-tab is selected. The main content area is titled 'Cloud' and shows a 'Type: Cloud' and 'License Key: Cucilicenta'. Below this, there are tabs for 'Targets (2)' and 'Database Access Keys'. A table lists two targets:

Target Name	Rating	Recos	Status	Date Modified
P0001	★★★★☆	0	Active	Sep 04, 2020 14:54
P0000	★★★★☆	19	Active	Sep 04, 2020 14:38

Figura 6.5 – Testare încărcare imagini în baza de date cloud

6.1.7. Finalizare puzzle

După ce toate piesele unui puzzle au fost conectate corect între ele, utilizatorului i se va indica acest lucru prin afișarea unui mesaj peste puzzle, care va indica că a fost finalizat cu succes.

Am testat acest lucru în funcție de distanțe, deoarece erau calculate deja, am ținut cont de ele, astfel când distanțele pentru ultima piesă sunt calculate corect, se va afișa mesajul. Acesta este unul simplu și clar.



Figura 6.6 – Testare finalizare puzzle

Capitolul 7. Manual de Instalare și Utilizare

7.1. Instalare aplicație

Pentru a instala aplicația există două metode.

1. Instalare pe un dispozitiv cu Android

Pentru a putea instala aplicația este necesar un dispozitiv inteligent ce folosește sistemul de operare Android. Pe lângă acesta, versiunea minimă pe care poate rula aplicația este 6.0 Marshmallow. Aplicația nu se află pe magazinul celor de la Google, astfel, pentru a o putea instala este necesar fișierul.apk, care este rezultat din construirea aplicației în Unity.

Se descarcă fișierul.apk și se accesează pentru a fi instalat. Telefon va avertiza utilizatorul cu privire la instalarea aplicației, deoarece aceasta nu este dintr-o sursă sigură, la acest pas trebuie să permitem telefonului să instaleze aplicația. După ce aceasta a fost instalată cu succes se poate utiliza fără probleme.

2. Instalare pe dispozitiv de tip desktop

Fiind necesar un dispozitiv cu Android și deoarece aplicația nu este accesibilă de pe Magazin Play, aceasta poate fi instalată pe dispozitivul de tip desktop.

- Primul pas este descărcarea portalului celor de la Unity, ce se numește Unity Hub. Acesta se poate descărca de pe link-ul următor <https://unity3d.com/get-unity/download>, care face o trimitere spre pagina oficială a celor de la Unity.
- Al doilea pas constă din instalarea versiunii 2018.4.24f1 de Unity. Aceasta se instalează din Unity Hub. În momentul în care se selectează versiunea, trebuie selectat pentru instalare și pachetul Android build Support.
- Al treilea pas constă în instalarea programului Android Studio. Acesta poate fi descărcat de pe link-ul următor <https://developer.android.com/studio>, care face o trimitere spre pagina oficială de descărcare. Este necesară instalarea acestui program, deoarece pentru a dezvolta o aplicație pentru Android este nevoie de un SDK și de un API. Acestea se instalează odată cu Android Studio.
- În Unity proiectul poate fi importat din opțiunea File, Open Project.
- După ce proiectul a fost importat cu succes, se poate porni direct din Unity apăsând pe butonul play.

- Dacă se dorește instalarea acestuia pe un dispozitiv, din meniul de sus se alege File, Build Settings și se apasă pe butonul Build. Proiectul va fi convertit într-un fișier.apk.
- Se conectează dispozitivul prin usb și se copiază fișierul rezultat în telefon.
- Aplicația poate fi instala accesând fișierul din telefon.

7.2. Manual de utilizare

7.2.1. Permisune cameră

După ce aplicația a fost instalată, prima data când se accesează va apărea un mesaj, care va întreba dacă se permite accesul aplicației la camera dispozitivului. Pentru ca acesta să poată fi folosită, este necesar să i se permită accesul.

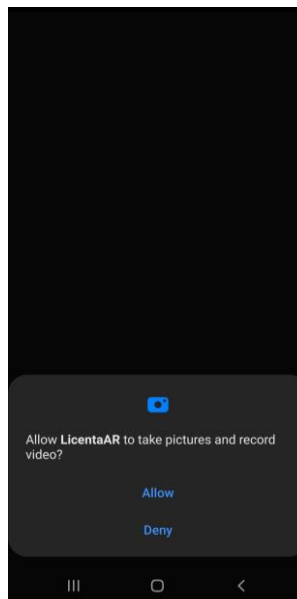


Figura 7.1 – Permisune cameră

7.2.2. Autentificare

După ce a fost acceptată permisiunea de a folosi camera, se va deschide primul meniu al aplicației. Meniul de autentificare inițial prezintă un buton de exit și un buton pentru autentificare. La apăsarea butonului „Cont”, se va deschide o fereastră care conține conturile Google care sunt conectate la dispozitiv. Utilizatorul trebuie să aleagă un cont pentru a se putea autentifica în aplicație. După ce a fost ales un cont valid, va apărea butonul „Meniu”. La apăsarea acestuia utilizatorul va fi trimis în meniul principal al aplicației.

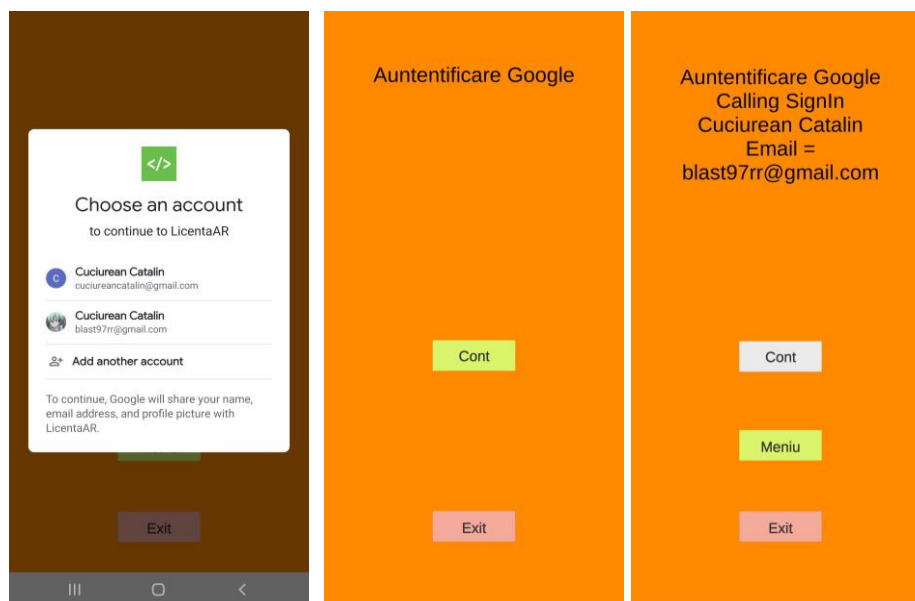


Figura 7.2 – Meniu de autentificare

7.2.3. Meniu Principal

Meniul principal conține următoarele opțiuni din care se poate alege:

- Puzzle Predefinit – Conține o lista de puzzle-uri predefinite în aplicație..
- Creare Puzzle – Reprezintă metoda prin care un utilizator își va putea defini propriul puzzle în aplicație.
- Puzzle Definit – Este meniul din care se selectează puzzle-ului definit de către utilizator.
- Informații – Conține informații legate despre aplicație. Ce fel de aplicație este și ce tehnologii folosește aceasta și cum le aplica.

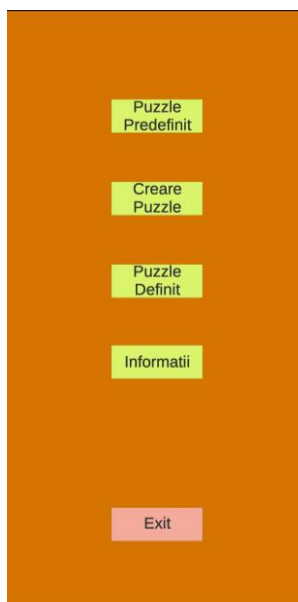


Figura 7.3 – Meniu Principal

7.2.4. Puzzle Predefinit

Acest meniu conține o listă de puzzle-uri care au fost predefinite în aplicație. Conține un buton pentru fiecare puzzle și alături de acesta o imagine reprezentativă. Pentru a rezolva un puzzle predefinit nu este necesară conexiunea la internet, deoarece toate imaginile sunt salvate în aplicație. Pentru a alege un puzzle se apasă pe buton, de exemplu pentru primul din listă, se apasă „Puzzle 1”. La apăsarea butonului „Back”, se va ieși din acest meniu și se va reveni la meniul principal.

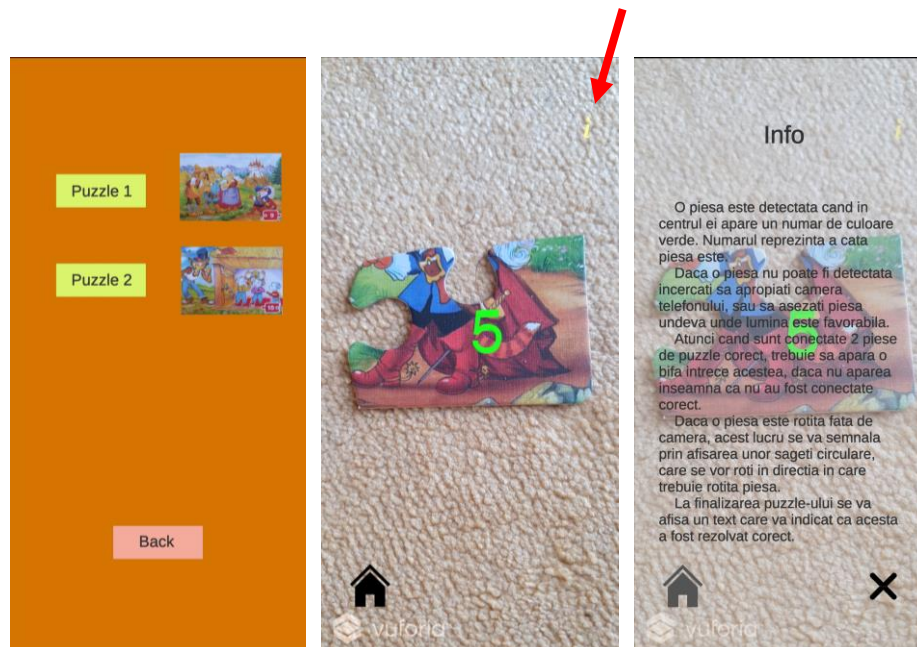


Figura 7.2 – Meniu puzzle predefinit.

Dacă un puzzle a fost ales, se va deschide o noua interfață, care va permite accesarea camerei pentru a putea vizualiza piesele de puzzle și pentru a le putea recunoaște. În această fereastră, se vor identifica și se va rezolva puzzle-ul ales. În colțul din dreapta sus, indicat cu săgeata roșie pe imagine, se afla un buton ce deschide o fereastră care conține informații despre obiectele ce se vor afișa și ce înseamnă fiecare. Mai exista un buton în forma de casă, care la apăsare, va ieși din fereastră curentă și va reveni la lista de puzzle-uri.

7.2.5. Creare Puzzle

Dacă utilizatorul dorește să definească un puzzle propriu, trebuie apăsat butonul „Creare Puzzle”, din meniul principal. Inițial se va deschide o fereastră ce conține instrucțiunile care trebuie urmate astfel încât să nu apară probleme în timpul definirii. După ce au fost citite instrucțiunile se va apăsa butonul „Next”, care va deschide o interfață nou în aplicație.



Figura 7.2 – Meniu pentru crearea unui puzzle.

În aceste ferestre se va indica numărul piesei și care este aceasta. De exemplu în imaginea de mai sus, trebuie definită prima piesă din puzzle, aceasta fiind piesa din colțul din stânga sus al unui puzzle. În zona delimitată, trebuie încadrată piesa de puzzle. După ce piesa a fost poziționată corect, se va apăsa pe butonul din mijloc, reprezentat printr-o cameră foto, pentru a se face captura de ecran, iar apoi se va apăsa butonul „Încărcare”, care va trimite imaginea spre o baza de date cloud, pentru a putea fi procesată. Se vor urma aceeași pași până când cele 9 piese de puzzle au fost încărcate în baza de date. Finalizarea acestui proces va fi indicat printr-un mesaj, iar utilizatorul se va putea întoarce la meniul principal al aplicației.

7.2.6. *Puzzle Definit*

Conține o fereastră de unde se poate selecta un puzzle definit de către utilizator. După ce acesta a parcurs toți pașii de definire corect, în această nouă fereastră, trebuie să fie recunoscute piesele adăugate de către acesta.

7.2.7. *Informații*

Această fereastră conține informații despre aplicație și cum funcționează. Este explicat procesul și modul în care funcționează și ce tehnologii au fost necesare pentru dezvoltarea ei.

Capitolul 8. Concluzii

8.1. Concluzie

În ultimii ani tehnologia a avut parte de un avans considerabil, devenind mai performantă și mai accesibilă. Din acest motiv majoritatea oamenilor au început să dețină cel puțin un dispozitiv inteligent, fie el tabletă sau telefon, deoarece acesta conține o mulțime de funcții care ne ajută să comunicăm, să ne informăm și să rezolvăm anumite sarcini fără a fi nevoie de alte dispozitive necesare. Eu cred că cel mai mare avantaj al acestor dispozitive este timpul pe care reușim să îl salvăm. În ziua de azi putem plăti aproape orice de pe telefon, fiind vorba de facturi, sau de cumpărăturile online, iar faptul că mereu avem un dispozitiv inteligent la îndemână, îl face perfect pentru astfel de activități.

Pe lângă avansul dispozitivelor inteligente, tehnologia realității augmentate a avut parte de o creștere foarte mare în popularitate. Au apărut o multitudine de noi aplicații ce se bazează pe această tehnologie, care este tot mai prezentă. Este folosită tot mai des, deoarece este intuitivă și este mai ușor să înțelegem lucrurile, atunci când le vedem și putem interacționa efectiv cu ele, cu rezultate imediate. Un exemplu foarte bun al folosirii tehnologiei, este integrarea ei în Google Maps, astfel utilizatorul dacă nu se poate orienta pe harta 2D, acesta poate intra în modul AR, care pe baza hărții va indica diferite săgeți, și informații spre destinație. Astfel orientarea în spațiu este mult mai ușoară și intuitivă.

Un alt avantaj al realității augmentate este acela educațional. Această tehnologie se conturează foarte bine pe acest domeniu, deoarece este mult mai ușor să înțelegem anumite obiecte și funcționalități prin intermediul lumii virtuale. Un exemplu poate fi integrarea manualelor cu tehnologie AR, astfel elevii când învață despre un anumit lucru, folosind un dispozitiv inteligent, pot să îl vadă într-o lume virtuală 3D și să interacționeze cu acesta, devenind mai ușor de înțeles.

Datorită acestor aspecte am fost motivat să implementez o aplicație care se folosește de tehnologia dispozitivelor și de realitatea augmentată. Am vrut ca aplicația să fie una distractivă, dar și educațională în același timp. Aplicația reușește să recunoască și să indice utilizatorului pașii spre rezolvare. S-a ajuns în acest punct prin multiple încercări și testări a funcționalităților. Deși este funcțională, se poate dezvolta mult mai mult, devenind mai intuitivă și mai performantă.

8.2. Dezvoltări ulterioare

Fiind o aplicație ce utilizează realitatea augmentată, exista o mulțime de moduri prin care acesta poate fi dezvoltată. În funcție de avansul tehnologiei, atât din punct de vedere al tehnologiei realității augmentate, dar și al dispozitivelor inteligente. Pentru acest lucru am pregătit câteva dezvoltări ulterioare.

8.2.1. *Ochelari inteligenți*

Cu siguranță în viitor ochelarii inteligenți vor deveni un lucru destul de comun odată cu avansul tehnologie. Momentan aceștia sunt costisitori și nu foarte performanți. Conțin un afișaj optic, care trimite imagini către lentilă, astfel utilizatorul poate vedea diferite informații. Pentru a folosi astfel de ochelari se folosesc comenzi vocale.

Aceștia sunt candidații perfecți pentru utilizarea aplicațiilor cu realitate augmentată, deoarece informația virtuală poate fi văzută direct prin lentilele lor, fără a mai fi necesar să folosim un dispozitiv inteligent precum un telefon.

8.2.2. *Îmbunătățire funcție de definire puzzle*

În aceste momente, funcționalitatea în care un utilizator își definește propriu puzzle este limitată de tehnologia folosită și anume Vuforia. Aceasta poate să recunoască simultan o singură țintă ce aparține de baza de date cloud.

Dacă în viitor se vor putea recunoaște mai multe imagini simultan, partea de definire va putea fi dezvoltată mai în detaliu, și va putea beneficia de toate funcționalitățile unui puzzle predefinit.

8.2.3. *Crearea unei baze de date proprii*

Momentan aplicația folosește baza de date de la Vuforia. O dezvoltare ulterioară ar fi crearea unei baze de date proprii pentru stocarea imaginilor. Acestea vor trebui să fie totuși procesate în baza de date de la Vuforia, dar după acest pas, să fie trimise către noua bază de date, fiind mai ușor de gestionat.

8.2.4. *Posibilitate recunoaștere puzzle-uri de mărimi mai mari*

În acest moment aplicația poate recunoaște puzzle-uri de 9 și 15 piese și se pot defini puzzle-uri doar cu 9 piese. Un pas următor ar fi definire de puzzle-uri de 15 piese. Din cauza modului în care a fost dezvoltată aplicația, aceasta nu este foarte scalabilă, astfel dacă se dorește rezolvarea unui puzzle mai mare de 100 de piese, este aproape imposibil de realizat.

Bibliografie

- [1] Azuma, R., Baillet, Y. and Behringer, R. (2001) Recent Advances in Augmented Reality. IEEE Computers and Graphic, 21, 34-47
<http://dl.acm.org/citation.cfm?id=618862>
- [2] Billinghurst, M. (2003) Augmented Reality in Education.
http://www.solomonalexis.com/downloads/ar_edu.pdf
- [3] Gu Jiawei, Zhang Yanpeng, Cao Dayuan, et al. Design and Simulation of virtual experiment system based on Unity 3D and AR Technology[J]. 2014.
- [4] Krystian Babilinski : Augmented Reality for Developers: Build practical augmented reality applications with Unity, ARCore, ARKit, and Vuforia
- [5] K. H. Ahlers, C. Crampton, D. Greer, E. Rose, and M. Tuceryan. Augmented vision: A technical introduction to the Grasp 1.2 system. Technical Report ECRC-94-14, ECRC, Munich, Germany, 1994.
- [6] Svetlin Nakov : Fundamentals of Computer Programming with C#: The Bulgarian C# Book
- [7] Ma, H.L. and Wu, D.B. (1998) The Computer and Multimedia Technologies in Education. Ling-Tung Journal, 8, 432- 448.
- [8] Unity Manual <https://docs.unity3d.com/Manual/index.html>
- [9] Lin, S.H. (2009) The Influence of Infusing Augmented Reality into Mobile Learning on Elementary School Students' Learning Motivation and Achievement in Science and Technology. Master's Thesis, National University of Tainan, Taiwan.
- [10] Liarokapis, F. (2007) An Augmented Reality Interface for Visualizing and Interacting with Virtual Content. Journal Virtual Reality, 11, 23-43.
<http://dx.doi.org/10.1007/s10055-006-0055-1>
- [11] Lu, C.C. and Yao, N.D. (2002) The Application and Introspection of Information Education in Elementary-School Curriculum. Elementary Education, 42, 19-24.
- [12] Li Jie. Research on an Application of AR System Based on Android[D]. Agricultural University Of Hebei, 2015.
- [13] Xu Min,Tong Qiang, et al. Development of AR Interactive App on Unity 3D +Vuforia[J]. Modern computer, 2016(12):71-75.

- [14] Microsoft Visual Studio IDE <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>
- [15] Vuforia Features <https://engine.vuforia.com/features>
- [16] Vuforia Engine <https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html#betas>
- [17] Vuforia Engine for Android Development <https://library.vuforia.com/articles/Solution/Getting-Started-with-Vuforia-for-Android-Development.html>
- [18] Vuforia Database <https://library.vuforia.com/articles/Solution/How-To-Work-with-Device-Databases.html>
- [19] Android Studio <https://developer.android.com/studio>
- [20] Android <https://www.android.com/what-is-android/>
- [21] What is Android <https://www.android.com/what-is-android/>
- [22] Android Architecture <https://developer.android.com/guide/platform>
- [23] Android Operating System <https://www.investopedia.com/terms/a/android-operating-system.asp>
- [24] Android SDK and API <https://developer.android.com/guide/topics/manifest/uses-sdk-element>
- [25] Firebase Authentication <https://firebase.google.com/docs/auth>
- [26] What is Firebase <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>
- [27] JSON <https://www.json.org/json-en.html>
- [28] Microsoft C# language <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- [29] Number of smartphone worldwide from 2016 to 2021 <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>
- [30] Mobile Operating System Market Worldwide <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [31] What is Augmented reality ? <https://www.fi.edu/what-is-augmented-reality>
- [32] Functional vs non functional Requirement <https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/>
- [33] Augmented Reality Introduction <https://www.interaction-design.org/literature/topics/augmented-reality>
- [34] Augmented Reality VS Virtual Reality <https://www.androidauthority.com/ar-vs-vr-782176/>
- [35] Android version history https://en.wikipedia.org/wiki/Android_version_history
- [36] Your AR Puzzle <https://www.amazon.com/YMBSMartin-YOUR-AR-PUZZLE/dp/B0752R55YT>
- [37] AMON <https://apps.apple.com/gb/app/amon/id1276464043>
- [38] Ravensburger Puzzle <https://newatlas.com/ravensburger-augmented-reality-jigsaw-puzzles/22079/>
- [39] PuzzlAR <https://www.magicleap.com/en-us/news/partner-stories/puzzlar-world-tour-first-magic-leap-icp-app-live>

- [40] 4D Puzzles <https://pl4d.net/product/4d-puzzles/>
- [41] ARise <https://www.gearbrain.com/arise-augmented-reality-game-review-2563236586.html>
- [42] Google Sign-In <https://developers.google.com/identity>