

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

**Umestitev interaktivnih elementov in elementov igrifikacije na
vnaprej zastavljeni učni poti**

(Placing interactive elements and elements of gamification on previously defined
education trail)

Ime in priimek: Borut Kolar

Študijski program: Računalništvo in informatika

Mentor: doc. dr. Matjaž Kljun

Somentor: doc. dr. Klen Čopič Pucihar

Koper, junij 2018

Ključna dokumentacijska informacija

Ime in PRIIMEK: Borut KOLAR

Naslov zaključne naloge: Umestitev interaktivnih elementov in elementov igrifikacije na vnaprej zastavljeni učni poti

Kraj: Koper

Leto: 2018

Število listov: 45 Število slik: 12

Število referenc: 24

Mentor: doc. dr. Matjaž Kljun

Somentor: doc. dr. Klen Čopič Pucihar

Ključne besede: Učna pot, Informacijske table, Igrifikacija, Digitalno učenje, GPS lokacija, NFC, Prostorski zvok

Izvleček:

Zaključna naloga obravnava možnosti umestitve interaktivnih elementov in elementov igrifikacije na učni poti Parka Škocjanske jame. Učna pot vsebuje veliko informacijskih tabel, zato te najprej predstavimo ter opišemo njihove prednosti in slabosti. Nato je predstavljena sama učna pot in njena infrastruktura. Za tem sledi pregled raziskav in obstoječih aplikacij, ki omogočajo oz. raziskujejo enega ali več načinov umestitve digitalnih gradiv z namenom učenja in obveščanja. Prav tako to poglavje predstavi nekatere tehnologije, ki smo jih vključili pri izdelavi svoje rešitve. Opisane so tudi funkcijske in sistemske zahteve, ki jim mora zadostiti naša rešitev. Ta je podrobneje predstavljena v naslednjem podpoglavju. Zadnje poglavje se osredotoča na implementacijo, v njem pa so predstavljena orodja in algoritmi, ki smo jih uporabili za implementacijo večjega dela aplikacije.

Key words documentation

Name and SURNAME: Borut KOLAR

Title of final project paper: Placing interactive elements and elements of gamification on previously defined education trail

Place: Koper

Year: 2018

Number of pages: 45 Number of figures: 12

Number of references: 24

Mentor: Assist. Prof. Matjaž Kljun, PhD

Co-Mentor: Assist. Prof. Klen Čopič Pucihar, PhD

Keywords: Education trail, Information tables, Gamification, Mobile learning, GPS location, NFC, Spatial audio

Abstract:

The thesis represents the possibilities of placing interactive elements and elements of gamification on education trail in the Škocjan Caves Park. Firstly, we present the advantages and disadvantages of information tables located on the education trail. After that, we present the education trail itself and its infrastructure. This is followed by an overview of previous research and existing applications that include one or more techniques of integrating digital content in order to educate and inform. This chapter also presents some of the technologies that we included in our solution. Next, we describe functional and system requirements that must be satisfied by our solution. In the next subsection, we present the solution in details. The last chapter focuses on implementation and presents tools and algorithms used to implement the majority of the application.

Zahvala

Zahvaljujem se mentorju, doc. dr. Matjažu Kljunu, ter somentorju, doc. dr. Klenu Čopiču Puciharju, za organizacijo in vso pomoč pri pripravi zaključne naloge.

Kazalo vsebine

1	UVOD	1
2	PREDSTAVITEV TRENUTNEGA STANJA	4
3	PREGLED PODROČJA	7
3.1	Digitalno učenje izven šolskih klopi	8
3.2	Lokacijske igre	10
3.2.1	Geolov	10
3.2.2	Pokemon Go	10
3.2.3	QuizTrail	11
3.2.4	PAC-LAN	11
3.2.5	BIG GAME HUNTR	11
3.3	Prostorski zvok	12
3.4	Tehnologija NFC	13
3.5	Oblikovni prostor interakcije nadgradnje učne poti	14
4	ZASNOVA	15
4.1	Funkcijske zahteve	15
4.2	Sistemske zahteve	16
4.3	Predlagana rešitev	17
5	IMPLEMENTACIJA	20
5.1	Mobilna aplikacija Android	21
5.1.1	Uporabniški vmesnik	21
5.1.2	Algoritem prepoznavanja GPS lokacije	22
5.1.3	Algoritem prepoznavanja NFC značke	24
5.1.4	Implementacija vprašanj s slikami	26
5.1.5	Podpora različnim jezikom	29
5.2	Razvoj mobilne aplikacije virtualne resničnosti z Unity	30
6	ZAKLJUČEK	33

7 LITERATURA

35

Kazalo slik

1	Učna pot Škocjan [19]	4
2	Začetna tabla ob vstopu na učno pot	5
3	Primer manjše table na učni poti - opazovalna točka 4	6
4	Razgledna točka nad Veliko dolino pri tabli številka 3 [19]	6
5	Potovanje zvoka do ušes, pri čemer zvok eno uho doseže prej kot drugo [23]	12
6	Primer obeh tipov vprašanj	18
7	Primer razlage na zastavljeno vprašanje in možnosti ponovnega odgo- varjanja	18
8	Primer prikaza števila prebranih značk	19
9	Arhitektura sistema	20
10	Uporabniški vmesnik in preklapljanje med stanji aplikacije	21
11	Pojavno okno, ki se uporabniku prikaže v primeru onemogočenega iska- nja GPS lokacije	23
12	Pojavno okno, ki se pokaže v primeru, da uporabnik nima vklopljenega NFC-ja	25

Seznam kratic

<i>NFC</i>	Near-field communication
<i>GPS</i>	Global Positioning System
<i>VR</i>	Virtual reality
<i>RFID</i>	Radio-frequency identification
<i>SDK</i>	Software development kit
<i>HRTF</i>	Head-related transfer function
<i>PDA</i>	Personal digital assistant

1 UVOD

V današnjem času se veliko ljudi odloča za obisk naravnih parkov. V parkih upravitelji pogosto postavijo informacijske table ali celotne učne poti, ki so namenjene temu, da se obiskovalci seznanijo z rastlinskim in živalskim svetom, tradicionalnimi obrtmi, naravnimi pojavi ter ostalimi naravnimi in kulturnimi znamenitostmi v parku [16].

Primer učne poti je tudi učna pot Škocjan, ki se nahaja v Regijskem parku Škocjanske jame v bližini Divače na slovenskem Krasu [19]. Učno pot spremljajo informacijske table, ki obiskovalcem podajajo informacije o značilnostih parka. Table tako omogočajo pridobivanje znanja, ki je podkrepljeno z opazovanjem realnega sveta. Prav tako so dobrodošle za tiste, ki učne poti ne obišejo v obratovalnem času parka in zato ne morejo dostopati do vsebin, ki so ponujene na informacijski točki parka.

Ena od omejitev tovrstnih tabel je razpoložljiv prostor, kar upravljavce parka omejuje pri količini informacij, ki jih lahko podajo. Za rešitev te težave so v parku izdali vodnik v obliki žepne knjižice, kar je odličen dodatek tablam na učni poti, saj dopolnjuje njihovo vsebino. Glavna prednost informacijskih tabel in žepnih vodnikov je njihova kakovost tiska in relativno nizka cena izdelave.

Kljub naštetim prednostim pa imajo informacijske table in vodniki nekaj slabosti. Obiskovalci morajo za vodnik vedeti in si ga morajo predhodno priskrbeti pred obiskom parka v primeru, da je informacijska točka zaprta. Vodnik je tudi dodaten predmet, ki ga je potrebno nositi s seboj. Zaradi tega obstaja velika verjetnost, da ga bodo ljudje brali v informacijskem centru ali doma, ne pa ob raziskovanju učne poti. Poleg tega težavo predstavlja tudi težko spreminjanje vsebin tabel in vodnikov. Pri vsaki spremembi je potrebno naročiti tiskanje nove table ali vodnika, kar predstavlja strošek za vzdrževalca poti. Nazadnje naj omenimo, da tako table kot vodnik ne omogočajo interaktivnosti.

Velik problem omenjenih tiskovin predstavlja tudi dejstvo, da lahko nanje umestimo le slike in besedilo. V Škocjanskih jamah imajo na voljo veliko dodatnih večpredstavnostnih vsebin, ki jih je na informacijske table oz. v žepne vodnike težko umestiti, recimo zaradi omejitve s prostorom. Tovrstno gradivo bi bilo odlično dopolnilo tabel na učni poti, saj bi lahko z njim popestrili obisk parka. Večpredstavnostne vsebine vključujejo

slike, zvočne posnetke ptic, videoposnetke in 360-stopinjske posnetke naravne in kulturne dediščine parka. S pomočjo prikaza tovrstnih vsebin bi imeli obiskovalci možnost ogleda predelov jam, kamor jim ni dovoljeno oz. dostop do tam ni mogoč, pojavov, ki jih ob določenem obdobju v letu obiskovalci ne morejo doživeti, saj se taki pojavi zgodijo bolj poredko (npr. poplava jame z zamašitvijo podzemnih sifonov ob neurjih), poslušali pa bi lahko tudi posnetke nekaterih živali, ki so v parku prisotne le v določenih obdobjih v letu ali se oglašajo le ob določenih urah dneva, zaradi česar se večina obiskovalcev v času obiska z njimi ne sreča.

Veliko opisanih gradiv je v okviru parka že predstavljenih, vendar je njihova predstavitev zelo razdrobljena. Poleg vsebin, predstavljenih na informacijskih tablah in v žepnem vodniku, jih je nekaj na voljo v jamarskem muzeju, nekaj v prostorih javnega zavoda v vasi Škocjan, nekaj pa na spletni strani zavoda. Kljub vsem dejavnostim in trudu upravljavcev parka pri izdelavi omenjenih gradiv, sklepamo iz lastnih izkušenj, da so te vsebine pogosto spregledane ali površno prebrane, še posebno, kadar govorimo o mlajši generaciji.

Eden od glavnih namenov upravljavcev parka je izobraževanje in ozaveščanje obiskovalcev o glavnih značilnostih območja, v tem primeru o naravni in biotski raznovrstnosti kraškega sveta. To se delno doseže z informacijskimi tablamami in žepnim vodnikom, vendar bogat nabor digitalnih vsebin in možnost interakcije z le-temi predstavlja dodatno priložnost, da se privabi obiskovalce k raziskovanju naravne in kulturne dediščine. Primeren način podajanja predstavljenega gradiva prek aplikacije mobilnih naprav bi lahko pripomogel k uresničitvi omenjenega namena.

V zaključni nalogi bomo predstavili, kako lahko zgoraj opisane informacijske table dopolnimo z digitalnimi vsebinami. Pri tem se nameravamo osredotočiti predvsem na mlajšo populacijo. Obstoječe večpredstavnostne vsebine bi lahko uporabniku ponudili na sami učni poti, saj bi bile tako predstavljene v kontekstu okolja v katerem se uporabnik nahaja. Ena od možnih rešitev predstavlja združitev vsebin prek aplikacije za pametne telefone.

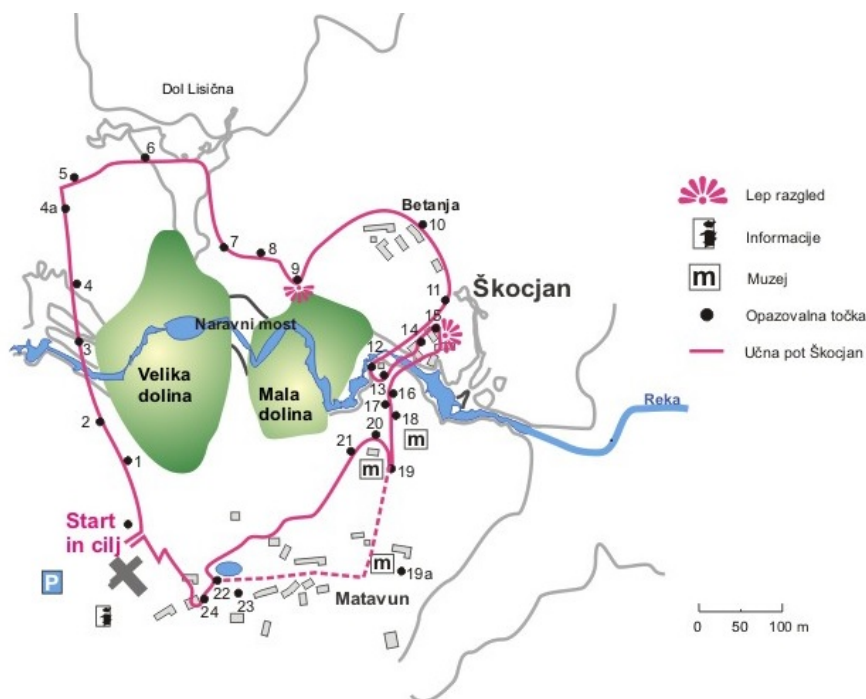
V nalogi je predstavljena naša rešitev z osnovno strukturo in delovanjem aplikacije, ki vsebuje lokacijsko pogojene kvize na učni poti ter nagrade v obliki dodatnih digitalnih vsebin v virtualni resničnosti. Podrobneje je opisana še implementacija enega izmed dveh izbranih tipov vprašanj. Poleg tega je opisana tudi implementacija prostorskega zvoka (ang. spatial audio), ki predstavlja element virtualne resničnosti in omogoča, da uporabnik z usmerjanjem svojega pogleda določa, zvok katere živali bo slišal.

V naslednjem poglavju je predstavljeno trenutno stanje učne poti v Parku Škocjanskih jam. Podrobneje je opisana učna pot z dodanim slikovnim gradivom, prav tako pa je na

kratko opisana sama infrastruktura v parku. Tretje poglavje vsebuje pregled področja, kjer je predstavljenih nekaj že obstoječih rešitev, in sicer na področju digitalno podprtih učnih poti, lokacijsko pogojenih iger in prostorskega zvoka. V četrtem poglavju so opisane funkcijske in sistemske zahteve, ki jim mora zadostiti predlagana rešitev, ta pa je v tem poglavju tudi podrobneje predstavljena. Peto poglavje nudi vpogled v glavne komponente, ki so bile potrebne za implementacijo rešitve. Šesto poglavje zaključuje nalogo.

2 PREDSTAVITEV TRENUTNEGA STANJA

Učna pot v Parku Škocjanske jame je namenjena popestritvi obiska parka in spoznavanju posebnosti klasičnega krasa. Dolga je približno dva kilometra, zahteva športno obutev in traja približno eno uro. Prične in konča se pred informacijskim centrom (glej sliko 1), kjer lahko kupimo tudi vodnik, ki nam nudi dodatne informacije pri spoznavanju območja [19].



Slika 1: Učna pot Škocjan [19]

Po poti lahko občudujemo glavne značilnosti samega območja – globoke udornice, prepadne stene, ponikajočo reko Reko in sosednje hribe. Poleg naravnih znamenitosti imamo tudi možnost vpogleda v etnološko zbirko ter zbirko odkrivanja in raziskovanja Škocjanskih jam. Tisti, ki se ne sprehodijo po celotni učni poti, pa si jo lahko delno ogledajo s pomočjo makete osrednjega območja parka in makete prečnega prereza podzemnega kanjona Reke [19].

Učno pot dodatno popestrijo informacijske table (glej Sliko 2 in 3), ki so napolnjene z raznovrstnimi ilustracijami, fotografijami in besedilom. Vse skupaj pripomore k boljšemu razumevanju naravnih danosti in kulturne dediščine, ki je skupna okolici regijskega parka [19]. Na poti se nahaja 28 tabel, pri čemer vsaka tabla prikazuje in opisuje pojave, ki so značilni za lokacijo (in njeno okolico), na kateri se obiskovalec nahaja. Zaporedna številka informacijske table je enaka številki opazovalne točke na zemljevidu Slike 1.



Slika 2: Začetna tabla ob vstopu na učno pot

Pot ponuja tudi nekaj razglednih točk. Ena se nahaja pri opazovalni točki 3 (glej Sliko 4), kjer imajo obiskovalci pogled na Veliko dolino, druga pa pri opazovalni točki 10, pri kateri se obiskovalci nahajajo nad Malo dolino [19].

Pot je povečini ravna in široka z nekaj ozkimi in strmimi segmenti. Na njej se nahaja tudi nekaj počivališč s klopmi.

Ob prvotnem pregledu poti za potrebe razvoja aplikacije smo bili pozorni tudi na kakovost mobilnega signala in mobilnega interneta. Na mestu, kjer stojijo table, smo preverili njuno moč pri treh različnih operaterjih in ugotovili, da sta kakovost tako mobilnega signala kot mobilnega interneta odlična na celotni učni poti.



Slika 3: Primer manjše table na učni poti - opazovalna točka 4



Slika 4: Razgledna točka nad Veliko dolino pri tabli številka 3 [19]

3 PREGLED PODROČJA

Za problem, ki ga predstavlja ugotavljanje uporabnikove lokacije in podajanje digitalnih interaktivnih informacij prek informacijskih tabel, ki se direktno povezujejo z informacijsko tablo in okolico, kjer se uporabnik nahaja, je na voljo veliko rešitev. Rešitve lahko uporabimo tudi v kombinaciji.

Ugotavljanje lokacije lahko izvedemo z GPS tehnologijo, ki omogoča sledenje uporabniku (kje v parku se trenutno nahaja). Druga možnost ugotavljanja lokacije je tehnologija NFC, kjer uporabnik z napravo prebere NFC značko, nameščeno na poti (na primer na tablah), nato pa se mu na napravi prikaže določena vsebina glede na ukaz iz NFC značke.

Ko je uporabnikova lokacija znana, lahko (interaktivne) vsebine predstavimo na različne načine. Poleg podajanja gradiv v navadnem slikovnem zvočnem ali video zapisu na zaslonu telefona lahko uporabniku omogočimo “potopitveno” izkušnjo tudi s podajanjem vsebin prek elementov dopolnjene resničnosti. S pomočjo tehnologije dopolnjene resničnosti na mobilnih napravah, lahko uporabniku omogočimo, da usmeri kamero mobilnega telefona v informacijsko tablo. Sistem tablo prepozna in uporabniku poda dodatne digitalne vsebine (prostorski zvok, video, slikovno gradivo) kot del informacijske table same.

Pri prikazu vsebin lahko uporabimo tudi elemente igrifikacije, ki pri uporabnikih vzbudijo občutek igre in tako pripomorejo k zdravi tekmovalnosti, doseganju statusa v igri in sodelovanje s parkom.

V naslednjih podpoglavjih je predstavljenih nekaj primerov, ki uporabljajo zgoraj opisane rešitve. Najprej so predstavljene tri raziskave, ki so bile opravljene na področju digitalnega učenja izven šolskih klopi. Nato je navedenih in opisanih nekaj lokacijsko pogojenih iger. Prav tako je predstavljen prostorski zvok, ki omogoča podobno izkušnjo kot poslušanje naravnega zvoka v okolju. Dodatno podpoglavje je namenjeno NFC tehnologiji, ki je prisotna v večini današnjih pametnih telefonov in se je pri nekaterih igrh izkazala kot dober vmesnik med fizičnim in digitalnim svetom.

3.1 Digitalno učenje izven šolskih klopi

V zadnjih dveh desetletjih je bilo opravljenih nekaj raziskav, ki so se osredotočale na učenje v naravnem okolju s pomočjo digitalnih naprav.

Ena od raziskav [2] opisuje sistem Explore!, ki je učence osnovnih šol spremljal pri njihovem obisku arheološkega parka. Sistem temelji na tehniki učenja skozi igro, ki učencem pomaga razumeti zgodovinske pojme, prav tako pa poskuša povečati učinkovitost obiska in tega narediti bolj razburljivega. Raziskava je primerjala izkušnje učencev pri igranju igre “na papirju” in igranju s pomočjo mobilnega telefona, ki je nudil 3D vizualizacijo zgodovinskih zgradb v parku, tudi takih, ki jih v parku ni več. Izkazalo se je, da je bila igra s pomočjo mobilnega telefona bolj cenjena kot tista “na papirju”. Učenci so bili navdušeni nad prikazovanjem 3D vizualizacij, saj se jim je na telefonu prikazala 3D slika zgradbe v preteklosti, ki so jo nato lahko primerjali z dejanskim stanjem zgradbe, kot so ga videli v arheološkem parku, in tako opazili razlike.

Avtorji druge raziskave [8], ki ravno tako govori o možnostih, ki jih ponujajo vseprisotne mobilne tehnologije za učenje izven šolskih klopi, so imeli namen ugotoviti:

- Kakšno vrsto digitalnih vsebin je potrebno ponuditi v takem primeru?
- Koliko, kdaj in kje ponuditi te informacije?
- Kolikšna količina interakcije z uporabnikom je potrebna?
- Kako integrirati digitalne informacije v fizični svet?

Poleg naštetega je bil cilj raziskave učencem omogočiti, da odkrijejo stvari, ki jih sicer morda ne bi opazili. Učencem, starim med 11 in 12 let, so bile ponujene različne digitalne vsebine, odvisno od tega, kje so se v gozdu nahajali. Raziskovalci so ugotovili, da učenci jasno povezujejo stvari iz fizičnega sveta z vsebinami, ki so jim bile ponujene na digitalnih napravah. Prav tako se je izkazalo, da je zelo pomembno, da so informacije ponujene ob pravem času in v pravem kontekstu.

Ena od težav, ki so jih raziskovalci predvideli, je bila, da uporabniki digitalnih naprav ne uporabljajo samo za učenje, ampak tudi za druge stvari. Izkazalo se je, da naprave, ki so jih raziskovalci prinesli od doma (npr. domači periskop), veliko bolje služijo svojemu namenu kot ostale digitalne naprave. Naprave, ki jih je potrebno držati v rokah, bi lahko predstavljale še dodatno težavo pri takem učenju, saj se učenci zaradi polnih rok ne bi mogli dotikati in prijemat stvari, ki se nahajajo v naravi. Študija je pokazala, da temu ni tako, saj si učenci naprave delijo tako, da imajo eno roko vedno prosto, kadar pa naprave ne uporabljajo, jo spravijo v žep.

Splošna ugotovitev raziskave je bila, da kljub morebitnim težavam digitalne tehnologije omogočajo primeren način za popestritev in dopolnitev učenja. Učinkovitost učenja pa je odvisna od načrtovanja takih tehnologij. Potrebno je upoštevati tip naprave, način, kako so informacije podane, in stvari, ki jih oseba v danem trenutku počne poleg pridobivanja znanja.

Še ena raziskava [9], ki opisuje mobilne tehnologije, ki lahko vsakodnevne dogodke pretvorijo v priložnosti za učenje, govori o digitalni knjižnici s kakovostnimi vsebinami za učenje na področju naravoslovja, matematike, inženirstva in računalništva za učence osnovnih in srednjih šol, pa tudi za nekoliko starejše. V sklopu projekta sta bila razvita dva prototipa mobilnih aplikacij. Prvi prototip so uporabljali učenci osnovnih šol, ki so se v parih sprehajali okoli šole, iskali različna mehanska orodja in jih fotografirali. S pomočjo dlančnika (PDA), ki je vseboval sistem za prepoznavanje orodij, so nato ta orodja lažje identificirali in se o njih naučili nekaj novega.

Druga skupina pa so bili dijaki, ki so uporabljali pametni telefon z GPS signalom. Naloga vsakega dijaka je bila, da se sprehodi po bližnji okolici in najde pet lokacij z digitalnimi vsebinami. Na vsaki od teh lokacij mu je bilo postavljeno vprašanje, ki se je navezovalo na lokacijo, na kateri se je nahajal, in imelo več možnih odgovorov. Po izbiri odgovora je bil študentom prikazan video, na katerem so strokovnjaki podali pravilen odgovor na zastavljeno vprašanje in razložili, zakaj je temu tako.

V raziskavi so ugotovili, da dijaki, ki so uporabljali mobilne telefone v času svojega gibanja po okolici, niso bili zainteresirani za branje dolgih člankov ali gledanje dolgih videoposnetkov na tako majhni napravi. Bolj so jih zanimale izobraževalne vsebine o lokaciji, na kateri so se nahajali. Prav tako so se morali dijaki veliko gibati, da so našli stvari, ki jim jih je predstavil mobilni telefon.

Študija kaže tudi na to, da morajo take tehnologije nuditi ravno dovolj informacij na zaslonu, da lahko uporabnik hkrati raziskuje tudi fizični svet. V nasprotnem primeru lahko prevelika količina informacij vpliva na to, da je uporabnik preveč zaposlen z napravo in je čas, namenjen raziskovanju okolice, okrnjen. Prav tako je dobro, da imajo učenci in študenti možnost ponovno videti ponujene vsebine kasneje v drugem okolju, npr. v učilnici. Poleg tega je dobrodošlo tudi to, da imajo uporabniki možnost dopolnjevanja vsebin, tako da lahko zapiske in slike, ki jih ustvarijo sami, dodajo k že obstoječim vsebinam in tako bodočim uporabnikom ponudijo dopolnjene informacije.

3.2 Lokacijske igre

Na trgu je prisotnih veliko iger za mobilne telefone, katerih vsebina je prilagojena lokaciji, na kateri se uporabnik nahaja. Glede na lokacijo te igre uporabniku podajajo naloge in informacije, ki so v kontekstu s prostorom, v katerem se uporabnik nahaja. Lahko so v kontekstu tudi časovno (letni čas ali dogodki iz zgodovine). V naslednjih podpoglavjih bomo predstavili nekaj lokacijsko pogojenih iger, ki so na voljo širši množici uporabnikov ali so nastale v raziskovalne namene.

3.2.1 Geolov

Geolov (ang. Geocaching) [6] je aktivnost, ki je predstavljena tudi kot mobilna aplikacija in uporabnikom omogoča, da s pomočjo GPS lokacije iščejo t. i. geotočke (ang. geocaches/caches) oz. zaklade. Geotočke so predstavljene v obliki GPS koordinat in so objavljene na svetovnem spletu. Lov na zaklad poteka tako, da uporabniki najprej zaklad najdejo na spletnih straneh ali mobilni aplikaciji, kjer so ti zabeleženi, nato pa njegove koordinate vpišejo v GPS sprejemnik (ali si pomagajo z GPS sprejemnikom mobilnega telefona), ki jih popelje do lokacije željene geotočke. Zaklad predstavlja nekakšen nepremočljiv zaboj, v katerem se nahaja vpisni dnevnik, kamor uporabniki vnesejo datum najdbe in svoj podpis. Na ta način dokažejo, da so našli zaklad, ki se tam nahaja. Ko se uporabniki vpišejo, morajo zaklad vrniti na isto mesto, kjer so ga našli. Lovci na zaklade lahko svojo izkušnjo delijo tudi na različnih socialnih omrežjih.

3.2.2 Pokemon Go

Gre za še eno igro, katere vsebina se spreminja na podlagi uporabnikove lokacije, poleg tega pa vsebuje tudi elemente dopolnjene resničnosti [20]. Glavne akterje igre predstavljajo virtualne živali, imenovane Pokemoni, ki jih lahko igralci najdejo, lovijo, trenirajo in se proti njim borijo.

Na začetku igre uporabnik ustvari “avatarja”, ki predstavlja igralca samega. Ko je ta narejen, je na zemljevidu prikazan na isti lokaciji, kjer se nahaja igralec v realnem svetu. Če se uporabnik sprehaja po realnem svetu, se hkrati spreminja tudi lokacija njegovega “avatarja” v igri. Prav tako se Pokemoni uporabnikom prikazujejo v skladu z realnim okoljem. Na lokacijo posameznega Pokemona vpliva tudi njegova vrsta (npr. vodni Pokemoni se po navadi nahajajo nekje blizu vode).

3.2.3 QuizTrail

QuizTrail [22] aplikacija ponuja uporabnikom kvize, ki so vezani na določeno lokacijo. Če se uporabnik nahaja na lokaciji, na katero je vezan določen kviz, ga aplikacija na to opozori. Prek kvizov se uporabnik uči in odkriva zanimivosti in znamenitosti nekega območja. Ko uporabnik prispe do določene lokacije, mu aplikacija ponudi vprašanje, na katerega naj bi odgovoril, preden nadaljuje pot. Dosežen rezultat se shrani, tako da lahko uporabniki primerjajo svoje dosežke z drugimi uporabniki, ki so odgovarjali na vprašanja, povezana z obiskanim področjem.

3.2.4 PAC-LAN

Za namen raziskave, ki ugotavlja, ali lahko običajni predmeti vplivajo na interakcijo med realnim in virtualnim svetom [7], so raziskovalci priljubljeno arkadno igro Pacman nadgradili v lokacijsko igro PAC-LAN. V ta namen so sodelujoči v raziskavi uporabljali mobilne telefone s čitalnikom radiofrekvenčne identifikacije (RFID), s pomočjo katerih so lahko vzpostavili interakcijo z objekti v realnem svetu. Ugotovljeno je bilo, da objekti v realnem svetu zelo pomagajo pri igranju igre, prav tako pa lahko RFID tehnologija predstavlja dobro sredstvo za učinkovito interakcijo. Uporabnikom je bila ideja o nadgradnji igre zelo všeč, med njenim igranjem so se zabavali, prav tako bi jo veliko ljudi igralo ponovno.

3.2.5 BIG GAME HUNTR

Raziskovalci so razvili platformo Big Game Huntr, s katero so uporabnike želeli spodbuditi k razmišljanju o sestavi lokacijsko pogojenih iger [5]. Uporabniki si lahko ustvarijo lastno lokacijsko pogojeno igro, določijo dogodke oz. aktivnosti znotraj igre, njeno igralno območje, težavnost in sistem točkovanja. Ko je igra sestavljena, jo lahko uporabniki igrajo. Lahko pa izberejo eno od iger, ki so jih pripravili ostali uporabniki.

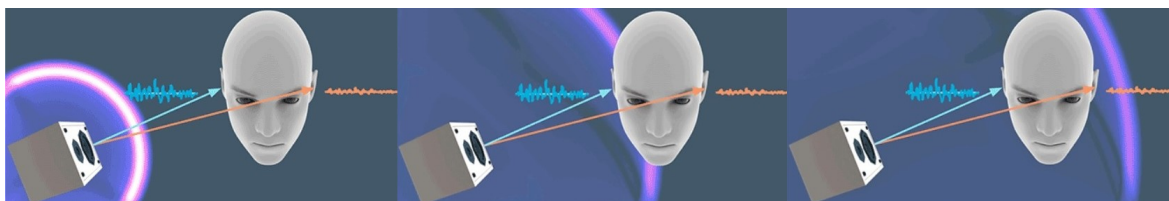
Raziskava je pokazala, da je kreativnost uporabnikov lahko ključnega pomeni pri ustvarjanju lokacijsko pogojenih iger. Igre so uporabniki ustvarili v različne namene: kot pripomoček za krajšanje časa med sprehodom po naravi ali pa nudenje interaktivnih vsebin uporabnikom določene prireditve. Take igre uporabnike motivirajo pri raziskovanju in odkrivanju okolice in jim tako pomagajo najti razne znamenitosti. Če igre vsebujejo točkovni sistem in lestvico najboljših rezultatov, so lahko take igre tudi motivacijske, saj želijo uporabniki doseči čim večje število točk in se primerjati z ostalimi uporabniki.

3.3 Prostorski zvok

Googlov paket za razvoj programske opreme (SDK) na področju prostorskega zvoka, imenovan "Resonance Audio", simulira način človeškega zaznavanja prostorskega zvoka. Z njim lahko uporabniku pričramo realistično izkušnjo zvoka in usmerjamo njegovo pozornost, saj mu s podajanjem zvokov iz različnih smeri namigujemo, kam mora usmeriti svoj pogled [23].

Paket temelji na treh načelih, ki jih ljudje uporabljamo za lokalizacijo zvoka:

- *Interauralna časovna razlika*: ko zvočni val trči ob uporabnikovo glavo, potrebuje različno količino časa, da pride do levega oz. desnega ušesa. Zamik je odvisen od horizontalne razdalje, ki jo zvok prepotuje - večja kot je, večji bo zamik (glej Sliko 5).
- *Interauralna razlika v ravni*: interauralno časovno razliko lahko človeško uho uporabi samo za nizkofrekvenčne zvoke. Visokofrekvenčni zvok človeško uho prepozna na podlagi njegove glasnosti s pomočjo akustične sence glave [11].
- *Spektralni efekti*: z njimi ne določamo lokacije zvoka, temveč njegovo višino. Zvoki, ki prihajajo iz različnih smeri, se na različne načine odbijajo od naših zunanjih ušes. S pomočjo teh razlik v frekvencah določimo višino zvoka.



Slika 5: Potovanje zvoka do ušesa, pri čemer zvok eno uho doseže prej kot drugo [23]

Za simuliranje interakcije zvočnih valov z našimi ušesi se uporabljajo funkcije zunanjega ušesa (HRTFs) [15]. Z njihovo pomočjo se ob poslušanju s slušalkami ustvari iluzija, da zvoki prihajajo iz različnih smeri.

Poleg simuliranja interakcije z našimi ušesi, Googlov paket simulirata tudi interakcijo zvočnih valov z okoljem, in sicer tako, da zvočni vir, ki mu ga podamo, razporedi po zvočni sferi. Ko nato uporabnik svojo glavo obrne v določeno stran, se ta sfera zavrti v nasprotno smer. Tako se ustvari iluzija, da se zvoki nahajajo na različnih lokacijah v virtualnem svetu [23].

Da bi bil zvok čim bolj realističen, paket upošteva tudi ovire, ki so na poti med virom in poslušalcem. To naredi tako, da visoke frekvence blokira bolj pogosto kot nizke, kar na nek način simulira dejansko situacijo v realnem okolju [23].

Poleg tega na kakovost zvoka vpliva tudi smer, iz katere zvok prihaja oz. vzorec. Npr. če se sprehajamo v krogu okoli nekoga, ki igra kitaro, bo zvok kitare glasnejši na strani, kjer so strune in zvočna luknja, saj na nasprotni strani prihod zvoka blokirata kitara in človeško telo. V Googlovem paketu lahko to simuliramo s pomočjo dveh parametrov: alfa (ang. alpha) predstavlja obliko vzorca zvoka, ostrina (ang. sharpness) pa predstavlja širino vzorca zvoka [23].

3.4 Tehnologija NFC

NFC (ang. near field communication) je tehnologija, ki omogoča brezstično komunikacijo med dvema napravama oz. komunikacijo naprave s tako imenovanimi NFC značkami. Za pretok informacij ni potreben fizičen stik med napravama, prav tako ni potrebe po vzpostavitvi kakršnekoli povezave (kot je to potrebno npr. pri tehnologiji Bluetooth). Dovolj je le, da sta si napravi dovolj blizu [17].

Povezovanje dveh naprav ali naprave z NFC objektom temelji na radiofrekvenčni identifikaciji (RFID), prenos podatkov pa poteka po magnetnem polju. Naprave ali objekti, ki uporabljajo tehnologijo NFC, so lahko pasivni ali aktivni. Pasivni objekti, kot so npr. NFC značke, vsebujejo informacije, ki jih lahko druge naprave preberejo, sami pa informacij ne morejo brati. Naloga takih objektov je le prenos lastnih informacij na druge naprave. Aktivne naprave, kot so npr. pametni telefoni, pa lahko poleg tega, da oddajajo informacije drugim NFC napravam, tudi prejemaajo podatke od ostalih objektov in naprav. Prav tako so zmožni tudi spreminjanja podatkov na pasivnih objektih, če to le-ti omogočajo [17].

Najbolj znana oblika uporabe tehnologije NFC je brezstično plačevanje s plačilnimi karticami. Za plačilo določenega zneska uporabniku ni potrebno fizično vstaviti kartice v POS terminal; dovolj je, da jo prisloni nanj. Brezstično plačevanje lahko opravimo tudi z mobilnim telefonom in uporabo aplikacije Android Pay, ki je na voljo uporabnikom operacijskega sistema Android in deluje kot digitalna denarnica.

Novejši uporabniki telefonov z operacijskim sistemom Android lahko uporabijo funkcionalnost Android Beam. Gre za funkcionalnost samega operacijskega sistema, s katero si lahko izmenjujemo spletne zaznamke, kontakte v imeniku, posnetke kanala YouTube itd. Ta funkcionalnost za izmenjavo podatkov uporablja tehnologijo NFC [13].

S pomočjo tehnologije NFC se lahko med pošiljateljem in prejemnikom prenašajo zelo občutljivi podatki, zato morajo biti le-ti na nek način zavarovani. V ta namen se ob uporabi prenosa s tehnologijo NFC vzpostavi zanesljiv kanal, ki uporablja kodiranje oz. enkripcijo ob pošiljanju občutljivih informacij (npr. pri plačevanju z brezstično kartico ni nikoli posredovana dejanska številka plačilne kartice, ampak se ustvari neka virtualna, saj se tako zavarujejo podatki plačnika) [4]. Uporabniki lahko podatke zaščitijo tudi tako, da dodajo zaščitno geslo, ki ga naprava zahteva za izvedbo določene akcije. Tako se lahko zaščitijo v primeru, če napravo izgubijo ali jim to ukradejo [17].

3.5 Oblikovni prostor interakcije nadgradnje učne poti

Osnovni vodili pri tvorbi rešitve sta bili, da je aplikacija namenjena mlajši populaciji in da z aplikacijo želimo prebuditi zanimanje za naravno in biotsko raznovrstnost kraškega sveta.

Pregled področja je pokazal, da je prostor oblikovanja interakcije precej širok in da lahko uporabimo kar nekaj omenjenih rešitev za aplikacijo, namenjeno obiskovalcem učne poti Škocjan, ki bo nadgradila obstoječo učno pot z digitalnimi vsebinami. Rešitve, ki bi jih lahko uporabili vključujejo na primer prikazovanje vsebine, ki je povezana z lokacijo in kontekstom uporabnika (kot na primer pri lokacijskih igrah) in igrifikacijo, ki temelji na napredovanju uporabnika po poti.

4 ZASNOVA

V tem poglavju so predstavljene zahteve, ki jih mora izpolniti naša rešitev. Razdeljene so na funkcijske in sistemske, opisano pa je tudi, kako bomo s predlagano rešitvijo poskušali tem zahtevam zadostiti.

4.1 Funkcijske zahteve

Funkcijske zahteve aplikacije so naštet v naslednjih točkah:

- Ponuditi lokacijsko pogojene vsebine le na točno določenih lokacijah
- Vsebine morajo biti kratke in zanimive
- Opozoriti uporabnika na ponujene vsebine, kadar se ta nahaja na določeni lokaciji
- Odvrniti uporabnika od uporabe elektronske naprave, kadar vsebine niso na voljo
- Motivirati uporabnika z elementi igrifikacije

Uporabniku želimo torej ponuditi digitalne vsebine samo na določenih lokacijah. Te vsebine so vezane na lokacijo. Ko uporabnik zapusti to lokacijo, mu onemogočimo dostop do teh vsebin. To je zelo pomembno, saj želimo uporabnika odvrniti od ne-nehnega spremljanja uporabljene naprave. To je ena ključnih težav z večino zgoraj opisanih rešitev, saj večina od njih zahteva, da so imeli uporabniki naprave, ki so ponujale digitalne vsebine, večino časa v roki, da bi spremljali pojavljanje nove vsebine. S ponujanjem vsebin samo na za to primernih lokacijah bi lahko naša rešitev odpravila to težavo, saj bi lahko uporabniki v času, ko se digitalne vsebine ne pojavljajo, imeli napravo v žepu.

Omenjene težave pri zgoraj opisanih prototipih vključujejo tudi preobširne informacije (preveč slik in predolga besedila), zato so uporabniki veliko časa porabili na napravah. Posledično je bilo uporabnikovo raziskovanje okolice zapostavljeno. Naša rešitev bi to lahko preprečila s tem, da bi imela kratka besedila, vprašalnike, zanimivo slikovno in video gradivo (na primer vsebine virtualne resničnosti). Zaradi zahteve po čim krajšem času, preživetem na napravi, količina ponujenih digitalnih vsebin ne sme biti prevelika, ogled ne sme zavzeti preveč časa, saj ne želimo spreminjati klasičnega obiska parka in raziskovanja okolice, temveč le popestriti doživetje obiskovalcev.

Glede na to, da bo izbrana rešitev s svojimi vsebinami temeljila na uporabnikovi lokaciji, mora sistem celoten čas svojega delovanja omogočati zaznavanje uporabnikove lokacije. Potrebna je približna in natančna zaznava uporabnikove lokacije. V primeru, da je s približno lokacijo uporabnika, zaznana preko GPS signala, povezana določena digitalna vsebina, mora sistem uporabnika o tem obvestiti (na primer z vibriranjem). Nato mora uporabniku omogočiti, da s pravilnimi dejanji, za katera prejme navodila na ekranu pametnega telefona, pridobi dostop do digitalnih vsebin.

Sistem mora na primeren način motivirati obiskovalca. Želimo namreč, da bi vsak uporabnik prišel do konca “igre” in tako pridobil kar največ znanja. Zato moramo uporabniku prikazati, koliko vsebin si je uporabnik že ogledal in koliko jih ima še na voljo (prikazovanje uporabnikovega napredovanja v učni vsebini obravnava raziskovalno področje Open Learner Model [3]). Prav tako moramo uporabnika na primeren način za njegovo sodelovanje tudi nagraditi, na primer z zbiranjem neke vrste žetonov ali nazivov pri napredovanju skozi park, s čimer se lahko primerja tudi z ostalimi uporabniki.

4.2 Sistemske zahteve

Sistemske zahteve:

- Rešitev mora biti mobilna
- Sistem mora ugotavljati približno in točno lokacijo uporabnika
- Sistem mora uporabnika opozoriti na prisotnost vsebin tako, da ga ne zmoti pri hoji in opazovanju okolice
- Sistem mora biti večjezičen

Sistem mora biti izdelan tako, da bo njegova uporaba mogoča na učni poti Škocjan. Rešitev mora biti mobilna in ne sme dodatno obremenjevati obiskovalca učne poti, saj je ta na določenih mestih zahtevna in strma.

Sistem mora varčevati z energijo naprave, saj ne želimo, da bi uporabniki porabili preveč baterije pri uporabi aplikacije. To bomo dosegli tako, da bodo lahko imeli uporabniki sistem med sprehodom po parku v “spečem” stanju. Le pri prisotnih vsebinah na neki lokaciji mora sistem uporabnika opozoriti na to.

Sistem mora biti zmožen slediti lokaciji uporabnika na celotnem območju učne poti. Potrebni sta dve natančnosti sledenja. Maksimalna napaka sledenja za “približno sledenje”, ki jo sistem uporablja za opozarjanje uporabnika o navzočnosti dodanih digitalnih vsebin, je 20 metrov. Maksimalna napaka za “natančno sledenje”, ki jo sistem uporablja za opozarjanje uporabnika na odpiranje digitalnih vsebin, je nekaj centimetrov.

Da bi še dodatno razširili množico potencialnih uporabnikov, mora biti naša rešitev tudi jezikovno podprta. Zato mora biti na voljo v več jezikih (slovenščini in angleščini), saj obiskovalci Škocjanskih jam prihajajo iz mnogih različnih držav, ne samo iz širše okolice.

4.3 Predlagana rešitev

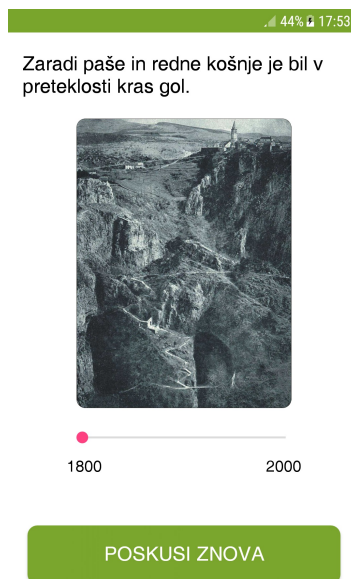
Vsi od naštetih prototipov v poglavju “Pregled področja” za svoje delovanje uporabljajo pametni telefon. To pa zato, ker so pametni telefoni dostopni vedno širšim množicam in imajo vedno večje zmogljivosti. S tem razlogom smo se tudi mi odločili, da izberemo mobilno napravo kot platformo za naš sistem in izdelamo mobilno aplikacijo. Glede na dejstvo, da večina pametnih telefonov Android omogoča implementacijo sistemskih in funkcijskih zahtev, smo se odločili, da bomo implementirali aplikacijo za pametne telefone z operacijskim sistemom Android.

Aplikacija, ki smo si jo zamislili, bo sestavljena v obliki prikazovanja vsebin na vnaprej izbranih določenih lokacijah. Obiskovalci se bodo sprehajali po učni poti, pri čemer bodo lahko imeli mobilni telefon spravljen na primer v žepu. Ko bodo prišli v bližino lokacije, s katero je povezana določena digitalna vsebina, jih bo sistem na to opozoril. Za to lahko uporabimo GPS sistem, ki bo ugotavljal približno lokacijo uporabnika. Po nevsiljivem opozorilu o prisotnosti digitalnih vsebin (na primer preko vibriranja) bodo lahko uporabniki poiskali NFC značko, ki se bo nahajala v bližini (na primer na informacijski tabli). Tehnologija NFC “določi” točno lokacijo telefona, saj deluje na nekaj centimetrov. Hkrati omogoča tudi sprožitev prikaza določene vsebine na telefonu.

Pri prikazu vsebin smo se odločili za uporabo kratkih vprašanj s slikovnim gradivom. Vprašanja morajo biti povezana z vsebino na informacijski tabli na določeni lokaciji. Lokacij z vprašanji bo osem, le-ta pa bodo dveh tipov (glej sliko 6): pri prvih bodo morali uporabniki odgovoriti z “DA” ali “NE”, pri drugih pa bodo imeli na voljo več slik, izbrati pa bodo morali tiste, ki predstavljajo pravilen odgovor na zastavljeno vprašanje (npr. izbrati bodo morali slike tistih ptic, ki živijo na prepadnih stenah). Sistem bo narejen tako, da bo pri prvem tipu vprašanj uporabniku po izbiri odgovora podana razlaga na temo vprašanja, ob morebitnem napačnem odgovoru pa bo imel možnost ponovnega odgovarjanja (kot prikazuje slika 7). Pri drugem tipu vprašanj pa bodo poleg slik na voljo tudi zvoki, ki bodo uporabniku še dodatno v pomoč pri izbiri ustreznega odgovora (npr. zvoki ptic, ki se nahajajo na slikah).



Slika 6: Primer obeh tipov vprašanj



Slika 7: Primer razlage na zastavljeno vprašanje in možnosti ponovnega odgovarjanja

Pri uporabi elementov igrifikacije smo se odločili, da bo uporabnik po vsakem pravilno odgovorjenem vprašanju prejel naziv. Nato bo uporabniku prikazano tudi število prebranih značk. Na ta način želimo dodatno motivirati uporabnike, da bodo dokončali celotno pot (glej Sliko 8).

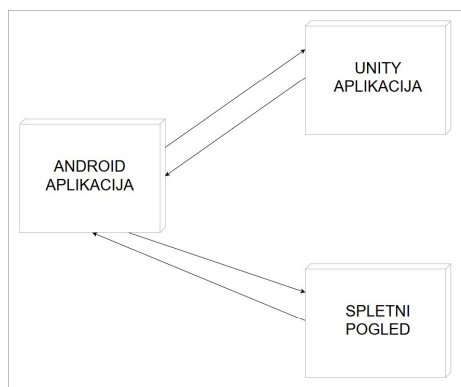


Slika 8: Primer prikaza števila prebranih značk

Da bi ponudili dodatne vsebine, bosta poleg osmih lokacij z vprašanji na učni poti tudi dve lokaciji, povezani z vsebinami virtualne resničnosti. Tudi v bližini teh lokacij bodo uporabniki o tem obveščeni, zatem pa bodo prav tako morali v okolju poiskati NFC značko in jo z napravo prebrati. Vsebine virtualne resničnosti bodo vsebovale 360-stopinjsko sliko, na kateri bodo pozicionirane živali, na katerih se bo, ko bo pogled usmerjen proti njim, začela izvajati določena animacija in predvajati zvok.

5 IMPLEMENTACIJA

Po določitvi funkcijskih in sistemskih zahtev in predlogu rešitve, smo začeli z implementacijo mobilne aplikacije. Mobilna platforma Android ponuja številna orodja za implementacijo različnih aplikacij. Jedro našega sistema predstavlja mobilna aplikacija Android, poleg nje pa sta del sistema tudi mobilna aplikacija virtualne resničnosti ter spletna stran z različnimi digitalnimi vsebinami. Arhitekturo sistema ponazarja Slika 9.



Slika 9: Arhitektura sistema

Kot lahko vidimo, je naš sistem sestavljen iz treh komponent. Glavno komponento predstavlja aplikacija za Android, ki je najbolj obsežna, vsebuje največ vsebin, prav tako pa jo bodo uporabniki uporabljali največ časa. Za implementacijo te komponente smo uporabili orodje Android Studio [14], ki podpira programski jezik Java. Iz te komponente sistem skoči na Unity aplikacijo v primeru, da uporabnik prebere NFC značko, s katero je povezana ena izmed vsebin VR. Za implementacijo te komponente smo uporabili orodje Unity [24], v katerem pisanje kode poteka v programskem jeziku C#. Ko si uporabnik ogleda vsebino virtualne resničnosti, s pritiskom gumba za izhod sistem ponovno skoči na aplikacijo za Android. Iz te komponente lahko sistem skoči tudi na spletni pogled, in sicer takrat, ko uporabnik na zaslonu, kjer ima prikazano število pridobljenih značk, klikne na eno izmed značk, ki prikazujejo ogledane vsebine VR. V sistemu se odpre nova aktivnost v Androidu, ki ni predstavljena v obliki VR, ampak ima v sebi integriran spletni pogled in nalaga vsebine s spletne strani, na kateri so naloženi modeli, ki si jih je uporabnik ogledal v vsebinah virtualne resničnosti.

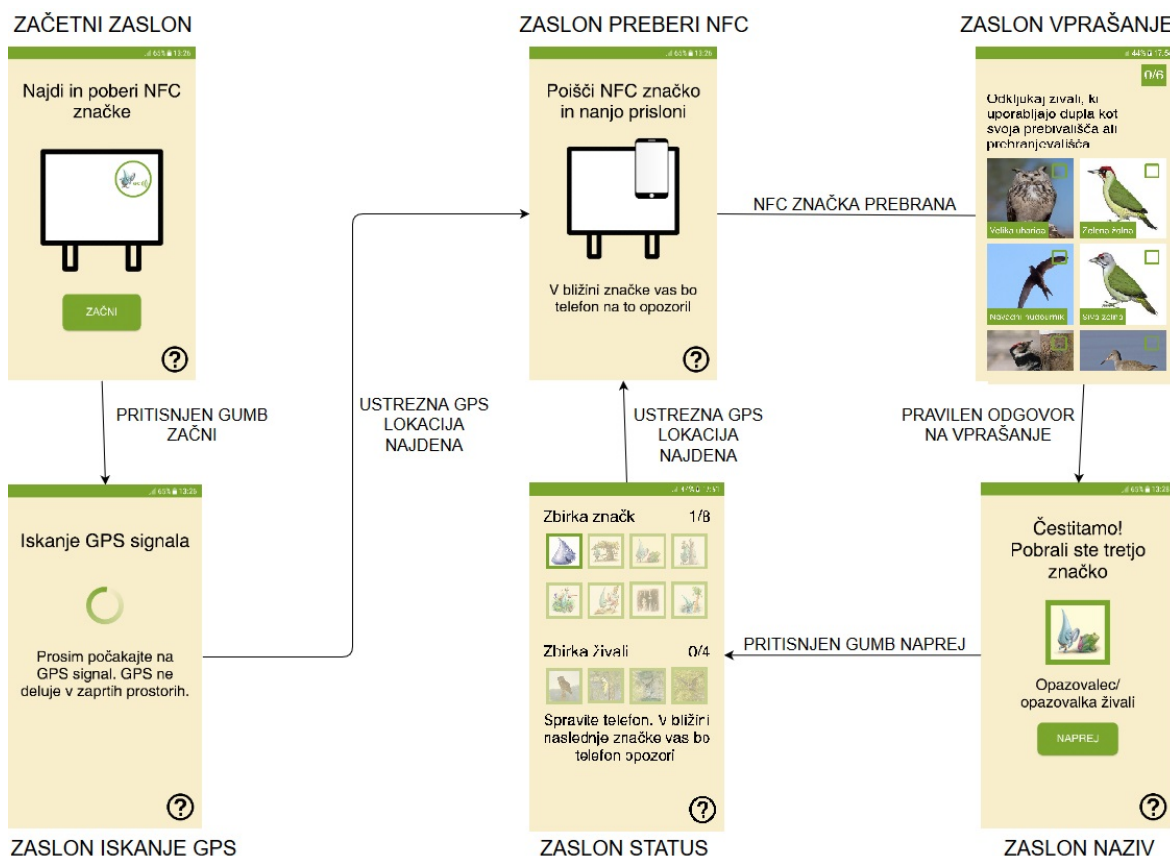
V nadaljevanju sta podrobneje predstavljene aplikaciji za Android in Unity.

5.1 Mobilna aplikacija Android

Kot omenjeno, glavni del oz. jedro celotnega sistema predstavlja mobilna aplikacija Android. Za razvoj tega dela aplikacije smo uporabili orodje Android Studio [14] in programski jezik Java.

5.1.1 Uporabniški vmesnik

Uporabniški vmesnik (prikazan na Sliki 10) smo zasnovali po načelu minimizacije interakcijskih elementov. Glavni namen stremjenja k tovrstni enostavnosti sistema je dejstvo, da bo večina uporabnikov našo aplikacijo uporabila le enkrat, zato je ključnega pomena, da oblikujemo vmesnik, ki je kar se da razumljiv in ne vmesnik, ki je fleksibilen in omogoča čim večji nabor funkcionalnosti.



Slika 10: Uporabniški vmesnik in preklapljanje med stanji aplikacije

Ob zagonu aplikacije se pojavi “Začetni zaslon”. Ob pritisku na gumb “Začni” nato aplikacija preide na “Zaslon iskanje GPS”. V tem stanju se aplikacija nahaja toliko časa, dokler ne najde ustrezne GPS lokacije, s katero je povezana določena digitalna vsebina. Ko tako lokacijo najde, se na napravi pojavi “Zaslon preberi NFC”. Ko se aplikacija nahaja na tem zaslonu, uporabnika pozove, da v svoji bližnji okolici poišče NFC značko in jo s telefonom prebere.

Ko uporabnik značko najde in jo uspešno prebere, aplikacija preide na “Zaslon vprašanje”. Na tem zaslonu se pojavi vprašanje, ki je povezano z enoličnim identifikatorjem (ID-jem) NFC značke (algoritma prepoznavanja GPS lokacije in NFC značke sta predstavljena v nadaljevanju). Vsaka lokacija ima vnaprej določeno vprašanje. Aplikacija ponuja dva tipa vprašanj, ki sta opisana v poglavju 4.3. Ob pravilnem odgovoru na zastavljeno vprašanje se v aplikaciji naloži “Zaslon naziv”. Tukaj se uporabniku prikaže naziv, ki si ga je prislužil s pravilnim odgovorom. Ob pritisku gumba “Naprej” aplikacija preide na “Zaslon status”. Tukaj se uporabniku prikaže stanje pridobljenih značk. V zgornjem delu zaslona je prikazano število pridobljenih značk s pravilnim odgovarjanjem na vprašanja, v spodnjem delu zaslona pa število pridobljenih značk z ogledom vsebin virtualne resničnosti. Ko aplikacija ponovno najde ustrezno GPS lokacijo, s katero je povezana določena digitalna vsebina, ponovno preide na “Zaslon preberi NFC”. V zadnjih štirih opisanih stanjih aplikacija kroži večino časa oz. dokler uporabnik ne prebere vseh značk. Takrat lahko uporabnik na “Zaslonu status” bodisi zapre aplikacijo, saj je prišel do konca učne poti oz. so mu bile ponujene že vse digitalne vsebine bodisi ponovno začne celoten proces od začetka.

5.1.2 Algoritem prepoznavanja GPS lokacije

Da aplikacija lahko dostopa do GPS lokacije Android naprave, mora imeti dovoljenje za to dejanje. Vsa dovoljenja, ki jih aplikacija ima, so deklarirana v datoteki *AndroidManifest.xml*. Poleg dovoljenj ta datoteka vsebuje tudi ime aplikacije, njene komponente, funkcije strojne in programske opreme, ki jih potrebuje za nemoteno delovanje itd. [14]. Da lahko dostopamo do lokacije, moramo v tej datoteki dodati naslednjo vrstico kode, ki aplikaciji omogoča, da dostopa do natančne lokacije:

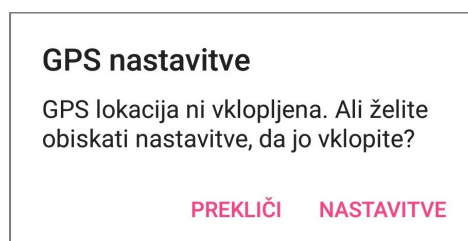
```
<uses-permission  
    android:name="android.permission.ACCESS_FINE_LOCATION"  
>
```

Vse funkcije, potrebne za prepoznavanje GPS lokacije, se nahajajo v razredu *GPSTracker* [12]. Ob ustvarjeni aktivnosti, ki ji pripada “Zaslon iskanje GPS” (*Location-*

Search), ali aktivnosti, ki ji pripada "Zaslonski status" (*Status*), moramo narediti novo instanco omenjenega razreda, da lahko dostopamo do vseh njegovih funkcij.

```
GPSTracker gpsTracker = new GPSTracker(LocationSearch.this);  
if(!gpsTracker.canGetLocation()) {  
    gpsTracker.showSettingsAlert();  
}
```

Najprej moramo preveriti, če ima uporabnik na svoji napravi omogočeno uporabo iskanja GPS lokacije. To preverimo s funkcijo *canGetLocation*, ki se nahaja v razredu *GPSTracker*. V primeru, da uporabnik nima omogočenega iskanja GPS lokacije, nam funkcija vrne *false* in izvede funkcijo *showSettingsAlert*, ki uporabniku prikaže pojavno okno (glej Sliko 11) in ga pozove, da omogoči GPS.



Slika 11: Pojavno okno, ki se uporabniku prikaže v primeru onemogočenega iskanja GPS lokacije

V primeru, da ima uporabnik omogočeno iskanje GPS lokacije, pa nam funkcija vrne *true*, zato se zgornji del kode ne izvede, sistem pa lahko začne s pridobivanjem GPS lokacije.

```
@Override  
public void onLocationChanged(Location location) {  
    if(location != null) {  
        latitude = location.getLatitude();  
        longitude = location.getLongitude();  
        float finalDistance = 1000000;  
        for (int i = 0 ; i < Constants.locations.length; i++) {  
            float distance = getDistance(  
                latitude, longitude, Constants.locations[i][0],  
                Constants.locations[i][1]  
            );  
            if (distance < finalDistance) {
```

```
        finalDistance = distance;
    }
}
if (finalDistance < 20) {
    Vibrator vibrator = (Vibrator)
        mContext.getSystemService(Context.VIBRATOR_SERVICE);
    vibrator.vibrate(vibrations, -1);
    Intent intent = new Intent(mContext, NFCScan.class);
    mContext.startActivity(intent);
}
}
}
```

Najpomembnejša funkcija v razredu *GPSTracker* je funkcija *onLocationChanged*, ki se zgodi vsakič, ko se uporabnikova GPS lokacija spremeni. Znotraj te funkcije najprej pokličemo funkciji *getLatitude* in *getLongitude*, da pridobimo zemljepisno širino in dolžino GPS lokacije. Nato se s *for* zanko sprehodimo čez polje lokacij, na katerih se nahajajo digitalne vsebine, ter v vsaki iteraciji pokličemo funkcijo *getDistance*, ki nam izračuna razdaljo med trenutno lokacijo in lokacijo table z digitalno vsebino (prva dva argumenta sta zemljepisna dolžina in širina trenutne lokacije, druga dva pa lokacije table). S pomočjo tega ugotovimo, kateri tabli se uporabnik približuje. Ko je razdalja med lokacijo uporabnika in tablo, ki se ji približuje, manjša od dvajsetih metrov, uporabnika z vibriranjem njegove naprave na to opozorimo, prav tako pa poženemo aktivnost *NFCScan* (*mContext* predstavlja aktivnost, iz katere ustvarimo instanco razreda *GPSTracker*) in prikažemo “Zaslon preberi NFC”. To je za uporabnika znak, da se v njegovi bližnji okolici nahaja NFC značka, ki jo mora poiskati in prebrati.

5.1.3 Algoritem prepoznavanja NFC značke

Podobno kot aplikacija potrebuje dovoljenje za dostop do GPS lokacije, potrebuje tudi dovoljenje za dostop do tehnologije NFC v napravi. V datoteki *AndroidManifest.xml* dodamo naslednjo vrstico kode:

```
<uses-permission android:name="android.permission.NFC" />
```

Poleg tega lahko aplikaciji omogočimo, da se v trgovini Google Play pojavi samo na tistih napravah, ki vsebujejo tehnologijo NFC. Da to storimo, moramo datoteki *AndroidManifest.xml* dodati naslednje:


```
<uses-feature
    android:name="android.hardware.nfc"
    android:required="true"
/>
```

V primeru, da naprava ne vsebuje tehnologije NFC, njeni lastniki do naše aplikacije ne bodo imeli dostopa.

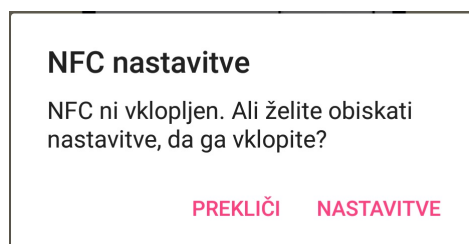
Vse akcije, povezane s prepoznavanjem NFC značke, se izvedejo v aktivnosti *NFCScan*, ki ji pripada "Zaslon preberi NFC". Ob tvorbi omenjene aktivnosti moramo najprej iz naprave z Androidom prebrati privzeti NFC vmesnik (večina naprav Android ima samo en tak vmesnik) [10].

```
NfcAdapter nfcAdapter = NfcAdapter.getDefaultAdapter(this);
```

Potem moramo preveriti, ali je na napravi vklopljen NFC. To storimo s pomočjo funkcije *isEnabled*, ki jo vsebuje vmesnik.

```
if (!nfcAdapter.isEnabled()) {
    showSettingsAlert();
}
```

V primeru, da na napravi ni vklopljen NFC, pokličemo funkcijo *showSettingsAlert*, ki uporabniku prikaže pojavno okno (glej Sliko 12) in ga pozove, da vklopi NFC na napravi.



Slika 12: Pojavno okno, ki se pokaže v primeru, da uporabnik nima vklopljenega NFC-ja

Ker ne vemo natančno, kdaj bo uporabnik NFC značko prebral, moramo vso logiko prepoznavanja značke implementirati znotraj funkcije *onNewIntent*, ki se izvede, ko se zgodi določena akcija, ki je povezana z aplikacijo [14].

```
protected void onNewIntent(Intent intent) {  
    byte[] tagBytes = intent.getBytesExtra(NfcAdapter.EXTRA_ID);  
    String tagID = ByteArrayToHexString(tagBytes);  
}
```

Edini podatek, ki ga moramo iz NFC značke razbrati, je njen ID. Na njej je ta podatek tipa *EXTRA_ID* in shranjen kot zaporedje zlogov (ang. bytes) [18], zato moramo uporabiti funkcijo *ByteArrayToHexString*, ki nam zaporedje zlogov pretvori v tekst (*String*) v šestnajstiškem številskem sistemu. Na podlagi ID-ja značke nato odpremo vprašanje, ki je s to značko povezano.

5.1.4 Implementacija vprašanj s slikami

Kot že omenjeno, bodo vprašanja, na katera bodo morali uporabniki odgovarjati, dveh tipov. V tem poglavju je podrobneje predstavljena implementacija drugega tipa, pri katerem bodo imeli uporabniki na voljo več slik, izbrati pa bodo morali tiste, ki predstavljajo pravi odgovor na zastavljeno vprašanje. Aplikacija bo vsebovala dve vprašanji tega tipa, in sicer:

- “Odkljukaj živali, ki uporabljajo dupla kot svoja prebivališča ali prehranjevališča” pri informacijski tabli št. 4 (glej Sliko 1) in
- “Ali prikazano na sliki sodi v kal?” pri informacijski tabli št. 26 (glej Sliko 1)

Vse funkcije, potrebne za prikaz vsebin pri vprašanju, se nahajajo v vmesniku *ImageAdapter* [14]. Ob ustvarjeni aktivnosti za informacijske table na Opazovalnih točkah 4 in 26 naredimo novo instanco omenjenega razreda, ki nam vrne vse potrebne vsebine za prikaz.

```
ImageAdapter tabla4ImageAdapter = new ImageAdapter  
    (Tabla4.this, images, sounds, checkboxSelection,  
    answers, popupTexts, imageTexts, button, textView, counter);
```

Kot lahko vidimo, vmesnik vsebuje 10 argumentov:

- *Tabla4.this*: aktivnost, iz katere kličemo vmesnik (poleg informacijske table 4 še informacijska tabla 26),
- *images*: polje slik, ki bodo prikazane pri vprašanju,
- *sounds*: polje zvokov, ki se bodo predvajali ob kliku na sliko,
- *checkboxSelection*: polje tipa *boolean*, ki nam pove, katere od slik je uporabnik označil (na začetku so vse vrednosti *false*),

- *answers*: polje tipa *boolean*, ki nam pove, katere od slik predstavljajo pravilen odgovor na zastavljeno vprašanje,
- *popupTexts*: polje besedil, ki se bodo prikazala ob kliku na sliko, ki ne predstavlja pravilnega odgovora (pri slikah, ki predstavljajo pravilen odgovor, je vrednost *null*),
- *imageTexts*: polje imen elementov, ki so prikazani na sliki,
- *button*: gumb "Odgovori",
- *textView*: besedilo, ki prikazuje vrednost argumenta *counter*
- *counter*: število, ki pove, koliko slik je uporabnik že izbral (na začetku je 0)

V vmesniku *ImageAdapter* so elementi na zaslonu razporejeni v obliki mreže (ang. Grid) [14]. Vsak element na mreži je tipa *ViewHolder*, ki ga definiramo v vmesniku.

```
class ViewHolder {  
    ImageView imageView;  
    CheckBox checkBox;  
    TextView textView;  
}
```

Kot lahko vidimo, ima vsak element svojo sliko (*ImageView* – slika ptice oz. elementa), potrdilno polje (*CheckBox* – polje, ki se označi, ko uporabnik pritisne na sliko) in besedilo (*TextView* – ime ptice oz. elementa). Vsakemu elementu na mreži določimo sliko, potrdilno polje in besedilo na naslednji način:

```
viewHolder.imageView.setId(position);  
viewHolder.imageView.setImageResource(images[position]);  
viewHolder.checkBox.setId(position);  
viewHolder.textView.setId(position);  
viewHolder.textView.setText(imageTexts[position]);
```

Najpomembnejši del zgornje kode je argument *position*, ki nam pove pozicijo elementa na mreži. Gre za indeks, ki je samodejno dodan vsakemu elementu ob tvorbi mreže, pri čemer ima element v levem zgornjem kotu mreže indeks 0, element desno poleg njega indeks 1 itd. Na podlagi tega indeksa lahko vsaki sliki, potrdilnemu polju in besedilu določimo enolični identifikator (ID), na podlagi katerega lahko do teh elementov dostopamo. Tako v zgornji kodi v drugi in peti vrstici sliki določimo vir ter besedilu tekst. S pomočjo indeksa najdemo pozicijo v poljih *images* oz. *imageTexts*, iz katerih dobimo vir pripadajoče slike oz. tekst pripadajočega besedila.

Najpomembnejša funkcija v vmesniku *ImageAdapter* je tista, ki se zgodi ob kliku na določeno sliko.

```
public void onClick(View v) {
    stopPlaying();
    int idImage = viewHolder.imageView.getId();
    int idCheckbox = viewHolder.checkBox.getId();
    if (answers[idImage] == true) {
        if (checkboxSelection[idCheckbox] == true) {
            counter--;
            textView.setText(Integer.toString(counter));
            viewHolder.checkBox.setChecked(false);
            checkboxSelection[idCheckbox] = false;
        } else {
            counter++;
            textView.setText(Integer.toString(counter));
            viewHolder.checkBox.setChecked(true);
            checkboxSelection[idCheckbox] = true;
            createSound(idImage);
        }
    } else {
        createSound(idImage);
        showPopup(idImage);
    }
}
```

Najprej pokličemo funkcijo *stopPlaying*, ki nam ustavi predvajanje zvoka, če se ta predvaja. Nato pridobimo ID-ja slike in potrdilnega polja, da lahko kasneje do njiju dostopamo. Zatem s pomočjo polja pravih odgovorov *answers* preverimo, ali slika, na katero je uporabnik kliknil, predstavlja pravih odgovor. Če temu ni tako (če ima v polju *answers* element na indeksu slike vrednost *false*), se izvede koda znotraj drugega *else* bloka, s katero predvajamo zvok (v tem primeru gre za pisk, ki ponazarja, da odgovor ni pravih) in prikažemo pojavno okno z besedilom, ki poda razlago, zakaj ta odgovor ni pravih. V nasprotnem primeru (slika predstavlja pravih odgovor – v polju *answers* ima element na indeksu slike vrednost *true*) pa se izvede koda znotraj prvega *if* bloka. S pomočjo polja izbranih potrdilnih polj *checkboxSelection* še preverimo, če je potrdilno polje na mestu slike že označeno. V primeru, da je temu tako, najprej zmanjšamo število *counter*, ki nam pove, koliko slik je uporabnik že izbral, in ga nato prikažemo v *textView* ter nato še izbrišemo oznako na potrdilnem polju in v polju *checkboxSelection* na mestu slike vrednost nastavimo na *false*. V nasprotnem primeru (potrdilno polje na mestu slike ni označeno) najprej povečamo število *counter* in ga nato prikažemo v *textView*, nato pa še dodamo oznako na potrdilnem polju in

v polju `checkboxSelection` na mestu slike nastavimo vrednost na `true`. Poleg tega še predvajamo zvok, ki pripada sliki, na katero je uporabnik kliknil (v tem primeru gre za zvok oglašanja ptice).

5.1.5 Podpora različnim jezikom

Glede na to, da obiskovalci Škocjanskih jam prihajajo iz številnih držav, smo se odločili, da aplikacijo naredimo v dveh jezikih. Osnovni jezik aplikacije je angleščina, poleg tega pa aplikacija podpira tudi slovenski jezik.

Android Studio ponuja odlično orodje, s pomočjo katerega lahko aplikacijo naredimo večjezično. Vsaka aplikacija v svoji strukturi vsebuje direktorij `res`, ki vsebuje gradiva, ki jih prikazuje uporabniški vmesnik. Kot razvijalci lahko definiramo različna gradiva za različne narode in jezike. Aplikacija nato na podlagi jezika, ki je nastavljen na napravi sami (v operacijskem sistemu Android), določi, katero mapo gradiv bo uporabila za uporabniški vmesnik [14].

Za našo aplikacijo smo se odločili, da bomo vsa besedila, ki se prikazujejo na uporabniškem vmesniku, imeli v dveh jezikih (angleščini in slovenščini). Ta se nahajajo v datoteki `res/values/strings.xml`. V tej datoteki so vsa besedila v angleškem jeziku. Če bo imela naprava nastavljen katerikoli jezik, ki ni slovenščina, se bodo vsa besedila nalagala iz te datoteke.

Vsa besedila v aplikaciji smo želeli dodati še v slovenskem jeziku za domače obiskovalce. To smo storili tako, da smo v direktorij `res` dodali nov direktorij `values-b+sl+SI`. Pripona `-b+sl+SI` nam pove, da gre za slovenski jezik in državo Slovenijo. V tej mapi se ravno tako nahaja datoteka `strings.xml`, v njej pa so vsa besedila v slovenskem jeziku. Če bo imela naprava kot jezik nastavljeno slovenščino, se bodo vsa besedila nalagala iz te datoteke.

V datotekah `strings.xml` ima vsako besedilo enolični identifikator `name`, kot prikazujeta spodnji dve vrstici kode.

```
<string name="tabla4">
    Tick/Mark the animals that use tree holes for nesting or hunting
</string>

<string name="tabla4">
    Odkljukaj živali, ki uporabljajo dupla kot svoja prebivališča ali
    prehranjevališča
</string>
```

Prvo besedilo se nahaja v datoteki za angleški jezik, drugo pa v datoteki za slovenski jezik. Besedili imata enak enolični identifikator in različni vsebini. Ko bomo v neki aktivnosti poklicali string z identifikatorjem "tabla4", bo od jezika, ki je nastavljen na napravi, odvisno, katero besedilo se bo naložilo. Če bo nastavljena slovenščina, se bo naložilo drugo, v nasprotnem primeru pa prvo.

5.2 Razvoj mobilne aplikacije virtualne resničnosti z Unity

Pomemben del našega sistema predstavlja tudi mobilna aplikacija virtualne resničnosti, ki smo jo implementirali s pomočjo orodja Unity [24]. Za tega smo se odločili, ker je implementacija vsebin virtualne resničnosti z njim lažja kot z orodjem Android Studio, in ker ponuja zelo obsežno dokumentacijo in obširno število obstoječih rešitev.

Za razvoj vsebin smo uporabili Googlov paket za razvoj programske opreme (SDK) [21]. Ta paket vsebuje tudi kazalec (ang. pointer), ki nam pomaga določiti, kam je usmerjen uporabnikov pogled. Vsebuje tudi skripto, ki na posamezen objekt v virtualnem svetu aplicira izbran prostorski zvok, ki bo v našem primeru predstavljal oglašanje živali.

Aplikacija za Unity predstavlja sceno, v kateri uporabimo dva objekta: virtualno okolje, tekstura ali statičen 360-stopinjski posnetek določene okolice na učni poti (okolici pri informacijskih tablah št. 3 in št. 10), ki ga vpnemo na kroglo, in animirani 3D objekti (živali) znotraj krogle. Ob zagonu scene se aktivira ena izmed krogel, in sicer glede na to, katero NFC značko je uporabnik prebral (informacijska tabla 3 ali 10). Animirane 3D objekte oz. živali uporabnik aktivira, ko jih pogleda. Takrat se kazalec, ki nam pomaga določiti, kam je usmerjen uporabnikov pogled, razširi, s čimer daje uporabniku vedeti, da se s pogledom nahaja na živali. Istočasno se začne predvajati še zvok oglašanja te živali. V primeru, da se s pogledom od živali malce oddaljimo, se glasnost zvoka zmanjša. Uporabnik dobi občutek, kot da bi se od živali dejansko oddaljil, tako iluzijo pa pričara tudi krogla, ki zvok razporedi po svojem prostoru [23].

Največji izziv v sklopu celotne implementacije je predstavljalo povezovanje aplikacije za Android in Unity. Kljub temu, da gre za ločeni komponenti našega sistema, smo si želeli, da bi bili obe komponenti povezani znotraj ene aplikacije. V nasprotnem primeru bi morali uporabniki imeti na svoji napravi nameščeni dve aplikaciji, kar pa zmanjša enostavnost uporabe našega sistema in lahko znatno vpliva na uporabniško izkušnjo.

Zaradi omenjene težave smo se odločili, da aplikacijo za Unity integriramo v aplikacijo za Android kot modul. Orodje Unity poleg klasičnega generiranja datotek .apk

omogoča tudi izvoz celotnega projekta kot modul za Android. Tega je nato potrebno s pomočjo orodja Android Studio spremeniti tako, da se obnaša kot knjižnica in ne kot aplikacija. S tem dobimo datoteko s končnico `.aar`, ki jo nato lahko uvozimo v našo glavno komponento [1].

Uvožen modul nato v želeni aktivnosti uporabimo na način, ki ga predstavlja spodnja koda.

```
import com.parskocjanskejame.vr.UnityPlayerActivity;
```

Zgornja vrstica predstavlja uporabo modula, pri čemer je `com.parskocjanskejame.vr` ime njegovega programskega paketa, `UnityPlayerActivity` pa aktivnost, ki vsebuje celotno sceno v Unity. To zaženemo le v primeru, da je NFC značka, ki jo je uporabnik prebral, povezana z eno izmed vsebin virtualne resničnosti.

Glede na to, da v danem trenutku prikažemo le eno izmed dveh vsebin virtualne resničnosti (glede na lokacijo uporabnika pri informacijski tabli 3 ali 10), mora aplikacija za Unity vedeti, kateri objekt (kroglo) aktivirati. Glede na to, da uporabnik značko prebere v aplikaciji za Android, mora ta Unity aplikaciji poslati podatek o prebrani znački. Na podlagi tega podatka se nato v Unity sceni aktivira ena izmed krogel z vsebinami virtualne resničnosti.

Odločili smo se, da bomo s komponente za Android pošiljali številko (*Integer*), ki bo povedala, katera značka je bila prebrana. Ustvarili smo globalno spremenljivko `VRScene`, ki je na začetku enaka 0. Ob prebrani znački pri informacijski tabli 3 jo nastavimo na 1, ob prebrani znački pri informacijski tabli 10 pa jo nastavimo na 2. Da bi lahko aplikacija za Unity prebrala ustvarjeno spremenljivko, smo morali v Androidu narediti funkcijo, ki vrne njeno vrednost.

```
public static int chosenVRScene() {  
    return VRScene;  
}
```

Da aplikacija za Unity dostopa do spremenljivke `VRScene`, smo morali objektu, ki se aktivira ob zagonu scene, dodati skripto, ki ob zagonu pokliče zgornjo funkcijo za Android. To smo storili s pomočjo spodnje kode.

```
AndroidJavaClass jc =  
    new AndroidJavaClass ("android.parskocjanskejame.NFCScan");  
int scene = jc.CallStatic<int> ("chosenVRScene");  
if (scene == 1) {
```

```
    space1.SetActive (true);  
} else {  
    space2.SetActive (true);  
}
```

Najprej ustvarimo novo instanco razreda za Android, kar nam Unity omogoča s tipom *AndroidJavaClass* [24]. Ob tvorbi navedemo ime programskega paketa aplikacije za Android (*android.parkskocjanskejame*) ter razred, do katerega želimo dostopati (v *NFCScan*, saj v tej aktivnosti uporabnik prebere NFC značko). Nato s pomočjo instance tega razreda pokličemo njegovo funkcijo *chosenVRScene*, ki nam vrne vrednost spremenljivke *VRScene*, ki si jo shranimo v spremenljivko *scene*. Na podlagi njene vrednosti nato aktiviramo enega izmed objektov z vsebinami virtualne resničnosti, pri čemer je *space1* objekt, ki ga aktiviramo pri informacijski tabli 3, *space2* pa objekt, ki ga aktiviramo pri informacijski tabli 10.

6 ZAKLJUČEK

Zaključna naloga predstavlja možnost nadgradnje učne poti Regijskega parka Škocjanske jame in informacijskih tabel z digitalnimi vsebinami. Kljub prednostim imajo informacijske table nekaj slabosti, saj so omejene s prostorom, na njih lahko umestimo le določene vsebine, prav tako pa ne omogočajo interaktivnosti in preprostega spreminjanja vsebin.

Glede na to, da imajo v Škocjanskih jamah na voljo veliko digitalnih gradiv, ki bi obiskovalcem še dodatno popestrila obisk, smo se v zaključni nalogi osredotočili na to, kako bi lahko te vsebine obiskovalcem predstavili. Odločili smo se za implementacijo aplikacije za pametne telefone z operacijskim sistemom Android. Za mobilno platformo smo se odločili zaradi njene priljubljenosti in dostopnosti ter vedno večje zmogljivosti pametnih telefonov.

Sistem smo zasnovali tako, da vključuje elemente igrifikacije, poleg tega pa digitalne vsebine ponuja na podlagi uporabnikove GPS lokacije, saj smo želeli, da bi bile informacije ponujene v pravem kontekstu. Zaradi tega smo v pregledu področja predstavili nekatere že obstoječe lokacijske igre in opisali nekaj raziskav, ki so bile opravljene na področju uporabe digitalnih vsebin pri učenju v naravnem okolju. Analiza raziskav je pokazala, da optimalna rešitev ne obstaja, saj ima vsaka od njih določene slabosti. Glavna pomanjkljivost je bilo dejstvo, da so morali uporabniki vseskozi opazovati zaslonski telefon in spremljati pojavljanje nove vsebine oz. brati dolga besedila, pri čemer so zapostavljali raziskovanje okolice. To slabost smo z našo rešitvijo odpravili tako, da uporabniku onemogočimo dostop do vsebin, če se ne nahaja na lokaciji, ki take vsebine ponuja.

Razvita aplikacija vsebuje vnaprej določene lokacije, ki so povezane z digitalnimi vsebinami. Uporabniki imajo lahko ob uporabi aplikacije telefon v žepu, saj jih ta z vibriranjem opozori na možnost ogleda digitalnih vsebin. Nato morajo uporabniki v bližnji okolici poiskati NFC značko in jo prebrati, da dostopajo do digitalne vsebine, ki je bodisi vprašanje, povezano s trenutno lokacijo, bodisi vsebina virtualne resničnosti. Sistem jim nato še prikaže naziv, ki so si ga prislužili s pravilnim odgovorom na vprašanje, ter število prebranih značk kot element igrifikacije.

Glede na to, da sistem ne beleži uporabnikovega početja, bi to lahko bila naloga za prihodnost. Sistem bi lahko beležil naslednje: prehojeno pot uporabnika, odgovore na vprašanja in preostalo interakcijo. S takšnim zajemom podatkov bi lahko upravljavci Škocjanskih jam opravili podrobnejše analize in ugotovili, koliko zanimanja obstaja za ogled digitalnih vsebin, kateri deli učne poti so najbolj obiskani ter kako dobro informacijske table obiskovalce informirajo o značilnostih obiskanega območja.

7 LITERATURA

- [1] D. BELOOSESKY, *Embedded Unity within Android App*, <https://medium.com/@davidbeloosesky/embedded-unity-within-android-app-7061f4f473a>. (Datum ogleda: 25. 5. 2018.) (*Citirano na strani 31.*)
- [2] M.F. COSTABILE, A. DE ANGELI, R. LANZILOTTI, C. ARDITTO, P. BUONO in T. PEDERSON, Explore! possibilities and challenges of mobile learning. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008) 145–154. (*Citirano na strani 8.*)
- [3] M.C. DESMARAIS in R.S. BAKER, A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction* 22(1–2) (2012) 9–38. (*Citirano na strani 16.*)
- [4] C. FAULKNER, *What is NFC? Everything you need to know*, <https://www.techradar.com/news/what-is-nfc>. (Datum ogleda: 30. 3. 2018.) (*Citirano na strani 14.*)
- [5] K. LUND, M. LOCHRIE in P. COULTON, Enabling emergent behaviour in location based games. *Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments* (2010) 78–85. (*Citirano na strani 11.*)
- [6] K. O'HARA, Understanding geocaching practices and motivations. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008) 1177–1186. (*Citirano na strani 10.*)
- [7] O. RASHID, W. BAMFORD, P. COULTON, R. EDWARDS in J. SCHEIBLE, PAC-LAN: mixed-reality gaming with RFID-enabled mobile phones. *Computers in Entertainment (CIE)* 4(4) (2006) 4. (*Citirano na strani 11.*)
- [8] Y. ROGERS, S. PRICE, G. FITZPATRICK, R.. FLECK, E. HARRIS, H. SMITH, C. RANDELL, H. MULLER, C. O'MALLEY, D. STANTON in M. THOMPSON, Ambient wood: designing new forms of digital augmentation for

- learning outdoors. *Proceedings of the 2004 conference on Interaction design and children: building a community* (2004) 3–10. (Citirano na strani 8.)
- [9] K. RYOKAI, A.M. AGOGINO in L. OEHLBERG, Mobile learning with the engineering pathway digital library. *International Journal of Engineering Education* 28(5) (2012) 1119. 82–91 (Citirano na strani 9.)
- [10] R. WONDRATSCHEK, *Reading NFC Tags With Android*, <https://code.tutsplus.com/tutorials/reading-nfc-tags-with-android--mobile-17278> . (Datum ogleda: 5. 3. 2018.) (Citirano na strani 25.)
- [11] *Acoustic shadow*, https://en.wikipedia.org/wiki/Acoustic_shadow. (Datum ogleda: 27. 3. 2018.) (Citirano na strani 12.)
- [12] *Android - Location based services*, https://www.tutorialspoint.com/android/android_location_based_services.htm. (Datum ogleda: 1. 3. 2018.) (Citirano na strani 22.)
- [13] *Android Beam*, <https://en.wikipedia.org/wiki/Android.Beam>. (Datum ogleda: 28. 3. 2018.) (Citirano na strani 13.)
- [14] *Android Studio*, <https://developer.android.com/studio/>. (Datum ogleda: 20. 2. 2018.) (Citirano na straneh 20, 21, 22, 25, 26, 27 in 29.)
- [15] *Head-related transfer function*, https://en.wikipedia.org/wiki/Head-related_transfer_function. (Datum ogleda: 27. 3. 2018.) (Citirano na strani 12.)
- [16] *Naravoslovna učna pot*, https://sl.wikipedia.org/wiki/Naravoslovna_učna_pot. (Datum ogleda: 15. 3. 2018.) (Citirano na strani 1.)
- [17] *Near field communication*, <http://nearfieldcommunication.org>. (Datum ogleda: 28. 3. 2018.) (Citirano na straneh 13 in 14.)
- [18] *NFC Forum Specifications*, <https://members.nfc-forum.org/specs/>. (Datum ogleda: 28. 3. 2018.) (Citirano na strani 26.)

- [19] *Park Škocjanske jame*,
<http://www.park-skocjanske-jame.si/>. (Datum ogleda: 15. 3. 2018.) (*Citirano na straneh VII, 1, 4, 5 in 6.*)
- [20] *Pokémon Go*,
https://en.wikipedia.org/wiki/Pokémon_Go. (Datum ogleda: 25. 3. 2018.)
(*Citirano na strani 10.*)
- [21] *Quickstart for Google VR SDK for Unity with Android*,
<https://developers.google.com/vr/develop/unity/get-started-android>.
(Datum ogleda: 10. 3. 2018.) (*Citirano na strani 30.*)
- [22] *QuizTrail*,
<http://quiztrail.com>. (Datum ogleda: 26. 3. 2018.) (*Citirano na strani 11.*)
- [23] *Resonance Audio*,
<https://developers.google.com/resonance-audio/discover/concepts>.
(Datum ogleda: 27. 3. 2018.) (*Citirano na straneh VII, 12, 13 in 30.*)
- [24] *Unity User Manual*,
<https://docs.unity3d.com/Manual/index.html>. (Datum ogleda: 5. 3. 2018.)
(*Citirano na straneh 20, 30 in 32.*)