

Mladi za napredek Maribora 2017

34. srečanje

## MOBILNA APLIKACIJA/IGRA EduQuest

Računalništvo

Raziskovalna naloga

Avtor: NIKO VUČKO, TOMAŽ KRALJ, LEONARD FIŠER  
Mentor: JOŽE ŠTRUCL  
Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

Maribor, februar 2017 Mladi za napredek Maribora 2017

34. srečanje

## MOBILNA APLIKACIJA/IGRA EduQuest

Računalništvo

Raziskovalna naloga

Maribor, februar 2017

## KAZALO VSEBINE

KAZALO VSEBINE .....	3
KAZALO SLIK .....	4
ZAHVALA .....	5
1 POVZETEK .....	6
2 IDEJA IGRE .....	7
3 DELOVANJE APLIKACIJE.....	8
3.1 Zemljevid .....	8
3.2 Bitke .....	9
3.3 Stopnje in statistični podatki .....	10
3.4 Mesta, predmeti in oprema .....	10
3.5 Meni .....	10
4 OPERACIJSKI SISTEM ANDROID .....	11
4.1 Kaj je platforma Android .....	11
4.2 Zgodovina Operacijskega sistema .....	11
4.3 Arhitektura platforme Android.....	12
4.4 Življenjski cikel aktivnosti Android aplikacije .....	13
4.5 Prednosti platforme Android .....	15
5 RAZVOJNO OKOLJE .....	16
5.1 Razvojno okolje Android Studio .....	16
5.2 Android SDK Android .....	17
5.3 Android API.....	17
5.4 Navidezna mobilna naprava (Emulator) .....	18
5.5 Programski jezik Java .....	18
6 RAZVOJ APLIKACIJE EDUQUEST .....	19
6.1 Začetek aplikacije.....	19
6.2 Premikanje po zemljevidu .....	19
6.3 Formula za napad .....	19
7 IZDELAVA GRAFIČNIH LIKOV .....	20
8 SKLEP .....	23
DRUŽBENA ODGOVORNOST .....	24
VIRI.....	25
SLIKOVNI VIRI .....	25

## KAZALO SLIK

Slika 1: Ikona aplikacije (avtor naloge) .....	7
Slika 2: Primer bitke iz igre Final Fantasy IV.....	9
Slika 3: Arhitektura Operacijskega sistema Android .....	12
Slika 4: Življenjski cikel aktivnosti Android aplikacije .....	14
Slika 5: Android studio Logo.....	16
Slika 6: API za različne verzije Androida .....	17
Slika 7: Java Logo .....	18
Slika 8: Orodja (avtor naloge).....	20
Slika 9: Layeri (avtor naloge) .....	20
Slika 10: Izdelava gumba za glavni meni (avtor naloge) .....	21
Slika 11: Glavni meni (avtor naloge).....	21
Slika 12: Glavni lik igre (avtor naloge) .....	22
Slika 13: Prototip lika (avtor naloge) .....	22
Slika 14: Prototip lika (avtor naloge) .....	22

## **ZAHVALA**

Najprej bi se radi zahvalili našemu mentorju, ki nas je vodil skozi celoten proces in nam nudil pomoč, ko smo kaj potrebovali. Zahvala gre tudi nekaterim sošolcem, ki so nam priskočili na pomoč pri izdelavi aplikacije.

## **1 POVZETEK**

Navdih za izdelavo naloge smo dobili, ker obožujemo video igre in bi tudi sami radi naredili delujočo igro. Hoteli smo se tudi naučiti programiranja v Java programskem jeziku saj smo se z njim prvič srečali v tej nalogi. Torej smo tudi testirali kako hitro se lahko naučimo programskega jezika in njegovo delovanje. Odločili smo se, da bomo aplikacijo napisali za mobilne telefone z operacijskim sistemom Android. Proces je potreboval tudi spretnost grafičnega oblikovanja za ustvarjanje likov v igri, okolja in menijev. Za to smo uporabili program Adobe Photoshop v katerem smo večinoma ustvarjali z bitno grafiko. Igro smo poimenovali EduQuest saj nam s tem pove namen igre. Edu stoji za education, kot neko učenje, Quest pa pustolovščina polna izzivov, torej lahko iz naslova sklepamo, da bo to poučna igra ob kateri se lahko zabavamo. Pri igranju igre bi naj ob zabavi uporabniki oz. igralci tudi pridobivali splošno znanje, saj nas igra spodbuja k temu, da bi se nekaj naučili saj s tem rešujemo glavnemu liku življenje. Igra nas popelje na potovanje polno dogodivščin in ugotovitev.

## 2 IDEJA IGRE

Ko smo bili otroci smo imeli zelo radi video igre. Ta radost se čez leta ni degradirala in tudi v končnem letu srednje šole nas je ta strast motivirala, da izdelamo igro. Zvrst igre je igra igranja vlog (angl. role-playing game oziroma RPG).

Igre igranja vlog so večinoma avanturne in fantastijske. Glavni del igre je premikanje glavnega lika po zemljevidu namišljenega sveta. Vsakič ko premaknemo lik je možnost, da se prične bitka. Bitke temeljijo na potezah, kot na primer šah, med glavnim likom, ki ga mi nadzorujemo, in sovražnimi liki. Na naši potezi lahko ukažemo liku različna dejanja, kot so Napad, Predmeti in Pobeg. Tako kot mi, lahko tudi nasprotni liki napadajo naš lik. Če je naš lik napaden se prikaže vprašanje in trije možni odgovori. Ob pravilnem odgovoru glavnemu liku ne odštejejo življenjskih točk, ampak ob narobnem odgovoru se bo napad nadaljeval in našemu liku bodo odštete življenjske točke. Bitko zmagamo tako, da vse življenjske točke nasprotnih likov zmanjšamo na nič. Če so od katerega lika življenjske točke enake nič, je tako rečeno umrl. V primeru, da se to zgodi našemu liku igra vrne na glavni meni, kjer lahko nadaljujemo od prejšnje shranjene točke v igri.

Na zemljevidu namišljenega sveta so tudi mesta v katera lahko glavni lik vstopi, če ga premaknemo nad njih. Ob vstopu v mesta se zemljevid spremeni v lokalni zemljevid, ki je manjši in vsebuje stavbe, kot so trgovina za predmete, trgovina za opremo in krčma, v kateri lahko za nekakšno ceno, obnovimo likovne življenjske točke. V lokalnem zemljevidu lahko kot v svetovnem zemljevidu premikamo glavni lik, vendar tukaj ni možnosti pričetka bitke. V mestih so tudi drugi liki, ki so nevtralni in ne morejo napasti našega lika ter naš lik njih.

Med igranjem lahko odpremo meni z več izbirami. Te opcije so Predmeti, Oprema, Status, Shrani in Izhod. Predmeti nam omogoča uporabo predmetov tudi izven bitke, z izbiro Oprema lahko spreminjamo opremo našega lika, Status nam prikaže statistične podatke našega lika, Shrani nam ustvari shranjeno točko v igri in Izhod nas vrne v glavni meni.



*Slika 1: Ikona aplikacije (avtor naloge)*

### **3 DELOVANJE APLIKACIJE**

Aplikacija se prične v glavnem meniju, kjer imamo na razpolago tri izbire. Te so Nova igra, Nadaljuj in Izhod. Nova igra nam odpre novo aktivnost, kjer imamo glavni lik v začetnem mestu. Glavni lik ima začetno opremo, ki jo lahko kasneje zamenjamo za boljšo. Na začetku je lik stopnje 1, kar pomeni, da so njegovi statistični podatki, kot na primer napadalna moč in obrambna moč ter življenjske točke, zelo majhni.

Izbira Nadaljuj nam vrne igre na zadnjo shranjeno točko v igri. Shranjena točka vsebuje lokacijo lika, njegovo stopnjo in opremo ter predmete.

Izhod nam pa samo zapre aplikacijo.

#### **3.1 Zemljevid**

V igri sta dve vrsti zemljevida: svetovni in lokalni. Na svetovnem zemljevidu je prikazan namišljen svet in naš lik, ki ga lahko premikamo. S premikanjem lika je možnost, da se prične bitka. Vrsta nasprotnih likov je odvisna od vrste pokrajine po kateri smo premikali lik, ko se je pričela bitka. Na določenih lokacijah zemljevida so mesta, v katera lahko vstopimo, če naš lik premaknemo nad njih. Ob tem premiku se zemljevid spremeni v lokalni zemljevid.

Lokalni zemljevid je večinoma uporabljen za mesta. V mestih so stavbe, v katere lahko vstopamo, če naš lik premaknemo nad njihova vrata. Primer stavb so trgovina za opremo, trgovina za predmete in krčma. V lokalnem zemljevidu ni možnosti pričetka bitke.



### 3.2 Bitke

V svetovnem zemljevidu lahko s premikanjem začnemo bitko. Bitka se dogaja v novi aktivnosti, kjer ne moremo več premikati našega lika. Na desni strani stoji naš lik, na levi pa nasprotni liki. Na spodnji strani zaslona pa je meni, ki prikazuje, na desni strani, ime našega lika in njegove življenjske točke, na levi strani menija pa imena nasprotnih likov in njihove stopnje. V sredini menija pa so tri možnosti, ki jih lahko izberemo. Te možnosti so: Napad, Predmeti in Pobeg.

Če izberemo Napad, bo igra primerjala statistični podatek Hitrost med nasprotnimi liki in našim likom. Zaporedje napadov se nato določi glede na to, kateri lik ima večjo Hitrost. Napad odšteje od nasprotnega lika toliko življenjskih točk, kot je naša napadalna moč, ampak zmanjšano glede na nasprotnikovo brambno moč. Ob napadu na naš lik, se pojavi novi meni kjer je napisano vprašanje. Pod vprašanjem so na voljo trije odgovori, mi moramo izbrati odgovor na vprašanje. Ob pravilnem odgovoru se napad konča brez kakršnih vplivov na naš lik, ampak ob narobnem odgovoru se bo napad nadaljeval in našemu liku bodo odštete življenjske točke. Odštete točke se izračunajo glede na nasprotnikovo napadalno moč in obrambno moč našega lika.

Izbira Predmeti nam odpre novi meni, ki prikazuje vse naše predmete. Tukaj lahko izberemo predmet, ki ga želimo uporabiti, ampak je možno izbrati samo predmete, ki imajo nekakšen vpliv na bitko (ne gre izbrati kosov opreme).

Če izberemo Pobeg potem igra primerja stopnjo našega lika z najvišjo stopnjo nasprotnih likov in če je naša stopnja višja se bitka konča, ampak ne dobimo nagrad za zmago bitke, če pa je naša stopnja manjša pa je 30 odstotna možnost, da se bitka konča.

Bitko zmagamo tako, da vse življenjske točke nasprotnih likov zmanjšamo na nič. Če so od katerega lika življenjske točke enake nič, je tako rečeno umrl in izgine iz zaslona ter ne more več napadati. V primeru, da nasprotniki zmanjšajo življenjske točke našega lika na nič, nas igra vrne na glavni meni.

Z zmagovanjem bitk si prislužimo denar in izkušnjske točke. Denar lahko porabimo v mestu za opremo, predmete ali pa spanje v krčmi. Z nabiranjem izkušnjskih točk pa lahko nadgradimo stopnjo našega lika, kar mu poveča statistične podatke.



Slika 2: Primer bitke iz igre Final Fantasy IV

### 3.3 Stopnje in statistični podatki

Stopnje so števila, ki igralcu lahko predstavljajo težavnost nekaterih nasprotnih likov. Naš lik in nasprotni liki imajo stopnje, vendar nimajo enakih statističnih podatkov. Stopnje nasprotnih likov smo določili, da se ujemajo s primerno stopnjo, ki jo bi naj imel glavni lik, da jih premaga. Naš lik postane višje stopnje, ko dobimo dovolj izkušenjskih točk. Število izkušenjskih točk potrebno za višjo stopnjo lahko pogledamo v meniju za statistične podatke. Vsakič ko naš lik postane višje stopnje se mu povečajo statistični podatki.

Liki v igri EduQuest imajo 4 različne vrste statističnih podatkov, te so: Življenjske točke (ŽP), Napadalna moč (NAP), Obrambna moč (OBR) in Hitrost (HTR). Življenjske točke nam povedo koliko napadov še lahko preživi naš lik ali pa nasprotni. Napadalna moč nam pove kako močni so naši napadi. Obrambna moč zmanjša število, ki je odšteto od Življenjskih točk ob napadu. Po Hitrosti se določi zaporedje potez v bitki.

### 3.4 Mesta, predmeti in oprema

Mesta se nahajajo na določenih lokacijah na svetovnem zemljevidu. V mestu so drugi liki s katerimi se lahko pogovarjamo in stavbe v katere lahko vstopamo. Liki so nevtralni tako, da ne moremo iti z njimi v bitko. V določenih hišah so liki, ki nam prodajo opremo ali pa predmete. Te hiše so označene glede na to kaj prodajajo. Trgovina za predmete je označena s kovancem, trgovina za opremo z mečem in krčma s posteljo.

Trgovina s predmeti nam lahko proda različne predmete: Življenjski napitek, Belo zastavico in Šotor. Življenjski napitek našemu liku obnovi določeno število življenjskih točk. Belo zastavico lahko uporabimo kadar je nasprotni lik višje stopnje od našega lika, da je Pobeg zagotovljen. Šotor lahko uporabimo samo na svetovnem zemljevidu in nam obnovi vse Življenjske točke. Vsi ti predmeti imajo različno ceno.

V trgovini za opremo lahko kupimo dele opreme. Oprema se deli na: Orožje, Čelada, Oklep, Hlače in Obuvalo. Orožje liku poveča Napadalno moč, Obuvalo poveča Hitrost, Čelada in Oklep povečata Obrambno moč, Hlače pa povečajo Obrambno moč in Hitrost. Nekatere dele opreme lahko naš lik obleče samo, če je dovolj visoke Stopnje.

### 3.5 Meni

V svetovnem in lokalnem zemljevidu lahko odpremo meni. V meniju so naslednje izbire: Predmeti, Oprema, Status, Shrani in Izhod. Predmeti nam omogoča uporabo predmetov. Oprema nam odpre novi meni v katerem lahko naš lik oblečemo v druge dele opreme. Status nam prikaže novi meni v katerem so prikazani statistični podatki našega lika in število Izkušenjskih točk potrebnih za naslednjo Stopnjo. Shrani nam ustvari shranjeno točko v igri, do katere lahko dostopamo z izbiro Nadaljuj v glavnem meniju do katerega lahko pridemo z izbiro Izhod.

## **4 OPERACIJSKI SISTEM ANDROID**

Preden smo začeli z izdelavo aplikacije, smo se morali odločiti na kateri platformi bo delovala. Odločitev ni bila težka, saj vsi uporabljamo operacijski sistem android na naših mobilnih napravah, torej nam najbolj ustreza. Pri odločitvi je pomagalo tudi, da je na trgu najbolj uporabljen operacijski sistem, po statistikah iz leta 2015 je pokrival kar 81.2% uporabnikov.

### **4.1 Kaj je platforma Android**

Android je operacijski sistem za pametne mobilne telefone, tablične računalnike, ter ostale prenosne naprave, ki je zgrajen na Linux jedru. Razvija ga Google, ki je v ta namen ustanovil poslovno združenje več podjetij, imenovano Open Handset Alliance (OHA). V tem združenju najdemo podjetja kot so Google, HTC, Sony, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung Electronics, LG Electronics, T-Mobile, Sprint Corporation, Nvidia, and Wind River Systems. Android je v prvi vrsti namenjen mobilnim telefonom, Google je začel tudi opremljanje drugih naprav, kot so Android Wear za pametne ure, Android Auto za avtomobile, Android TV za televizije itd. Vsaka s svojim uporabniškim vmesnikom. Večina ljudi, ki so že slišali za platformo Android, zmotno misli, da je platforma Android namenjena zgolj zahtevnejšim uporabnikom. Vendar temu ni tako. Platforma Android je namenjena tudi uporabnikom, ki nimajo veliko predznanja in izkušenj z napravami, kot so npr. namizni/prenosni računalniki, mobilni telefoni itd., ki omogočajo uporabo aplikacij preko različnih kompleksnih interaktivnih programskih vmesnikov.

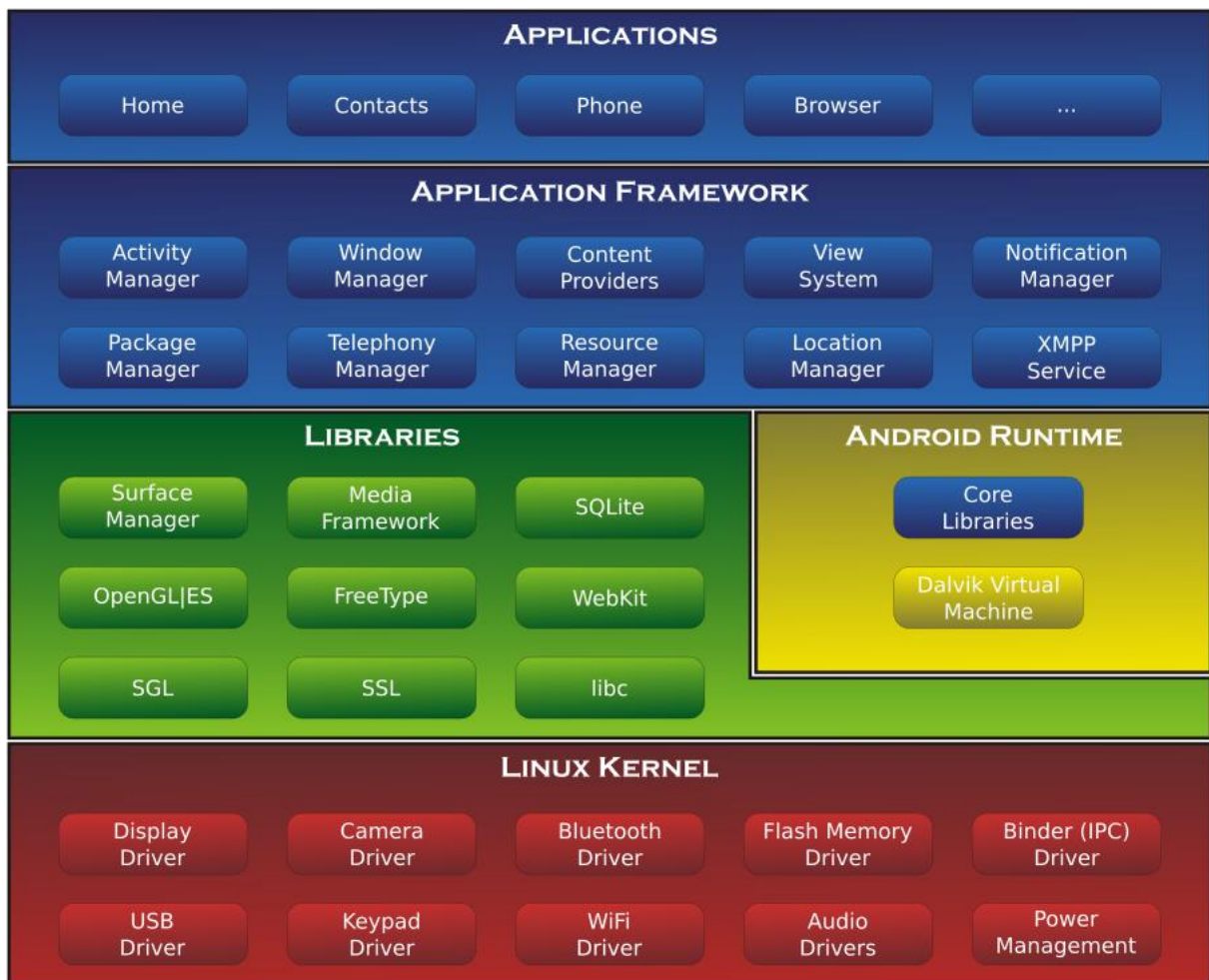
### **4.2 Zgodovina Operacijskega sistema**

Ime Android pomeni robota, ki s svojim videzom ter delovanjem spominja na človeka, z razvojem je začelo podjetje Android Inc., ki so ga ustanovili Andy Rubin, Rich Miner, Nick Sears in Chris White, leta 2003 v Kaliforniji, v mestu Palo Alto, da bi razvili tako imenovano »pametnejšo mobilno napravo, ki se bolj zaveda lastnikove lokacije in njegovih ljubših nastavitev.« Prvi namen podjetja je bil razviti naprednejši operacijski sistem za digitalne kamere. Vendar ko so se zavedali, da je trg premajhen, so se odločili, da se bodo preusmerili na trg mobilnih telefonov, da bi naredili pametni telefon, ki bi se lahko meril s takratni Symbianom in Microsoftovim Windows Mobile. Leta 2005 je podjetje prevzelo podjetje Google, ki je želelo vstopiti v mobilni trg. Takrat so pričeli z razvojem operacijskega sistema za mobilne naprave, ki bi naj temeljil na Linux jedru.

Prihod odprtokodne mobilne platforme Android so objavili novembra 2007 z ustanovitvijo združenja Open Handset Alliance (OHA). V to združenje so danes vključena velika podjetja s področja proizvodnje računalniških naprav, programske opreme, mobilni operaterji in ponudniki mobilnih telefonov. Leta 2007 je sledila tudi prva izdaja programskega orodja. Avgusta 2008 je bil napovedan trg aplikacij Android Market, na katerem je danes vse več aplikacij in iger.

### 4.3 Arhitektura platforme Android

Odpertokodno platformo Android v grobem sestavlja pet slojev. Vsak sloj lahko uporablja storitve, ki jih ponuja sloj pod njim.



*Slika 3: Arhitektura Operacijskega sistema Android*

Celotni operacijski sistem Android je zgrajen na podlagi Linux jedra, ki predstavlja osnovni sloj z nekaterimi dodatnimi arhitekturami spremembami, ki jih je naredil Google. Slika prikazuje glavne komponente Androida. Arhitektura je sestavljena iz petih elementov: aplikacije, aplikacijsko ogrodje, knjižnice, jedro linuxa in izvajalno okolje.

Jedro deluje kot vmesna plast med strojno opremo in programskim delom. Med pomembnejše osnovne funkcije, ki jih Android uporablja z jedrom, spadajo upravljanje s pomnilnikom, upravljanje procesov, mrežna podpora, gonilniki in varnost.

Naslednji sloj so knjižnice, ki omogočajo obdelavo različnih vrst podatkov. Te niso aplikacije same po sebi, temveč skrbijo za povezavo s komponentami sistema Android. Knjižnice so napisane v programskem jeziku C/C++. Knjižnice vsebujejo grafične knjižnice (Skia, OpenGL ES), knjižnica za podporo medijem, omogoča predvajanje zvoka in video vsebin (MP3, MP4, AAC), knjižnica SQLite za podporo podatkovnim bazam in knjižnica Webkit, ki zagotavlja delovanje spletnega brskalnika.

Glavni del zagonskega okolja predstavlja ART, ki izboljšuje splošno učinkovitost izvajanja in s tem zmanjšuje porabo energije, prinaša pa hitrejše izvajanje aplikacij ter izboljšavo delovanja pomnilnika.

Naslednji sloj je aplikacijsko ogrodje, v katerem se nahajajo vse systemske aplikacije, ki se uporabljajo za razvoj aplikacij. Razvijalci lahko izkoristijo strojno opremo naprave, poganjajo servise v ozadju itd. Omogoča tudi dostop do knjižnic. Spodaj so na kratko opisani arhitekturni gradniki vseh aplikacij Android:

- Pogledi (*ang. Views*) so osnovni gradniki uporabniškega vmesnika. To so razni gumbi, vnosna polja, sezname in brskalniki,
- Ponudnik vsebin (*ang. Content Provider*) aplikacijam omogoča dostop do deljenih podatkov, kot so npr. stiki, in omogoča deljenje podatkov med aplikacijami,
- Upravitelj lokacij (*ang. Location Manager*) omogoča dostop do lokacijskih storitev,
- Upravitelj virov (*ang. Resource Manager*) zagotavlja dostop do virov, ki niso del aplikacije, kot so lokalni nizi, grafični elementi in slike,
- Upravitelj obvestil (*ang. Notification Manager*) skrbi za prikazovanje obvestil v vrstici stanja sistema, omogoča dosledno obveščanje uporabnikov,
- Upravljanje aktivnosti (*ang. Activity Manager*) upravlja aktivnost življenjskega cikla posameznih aplikacije in omogoča medsebojno komunikacijo med njimi.

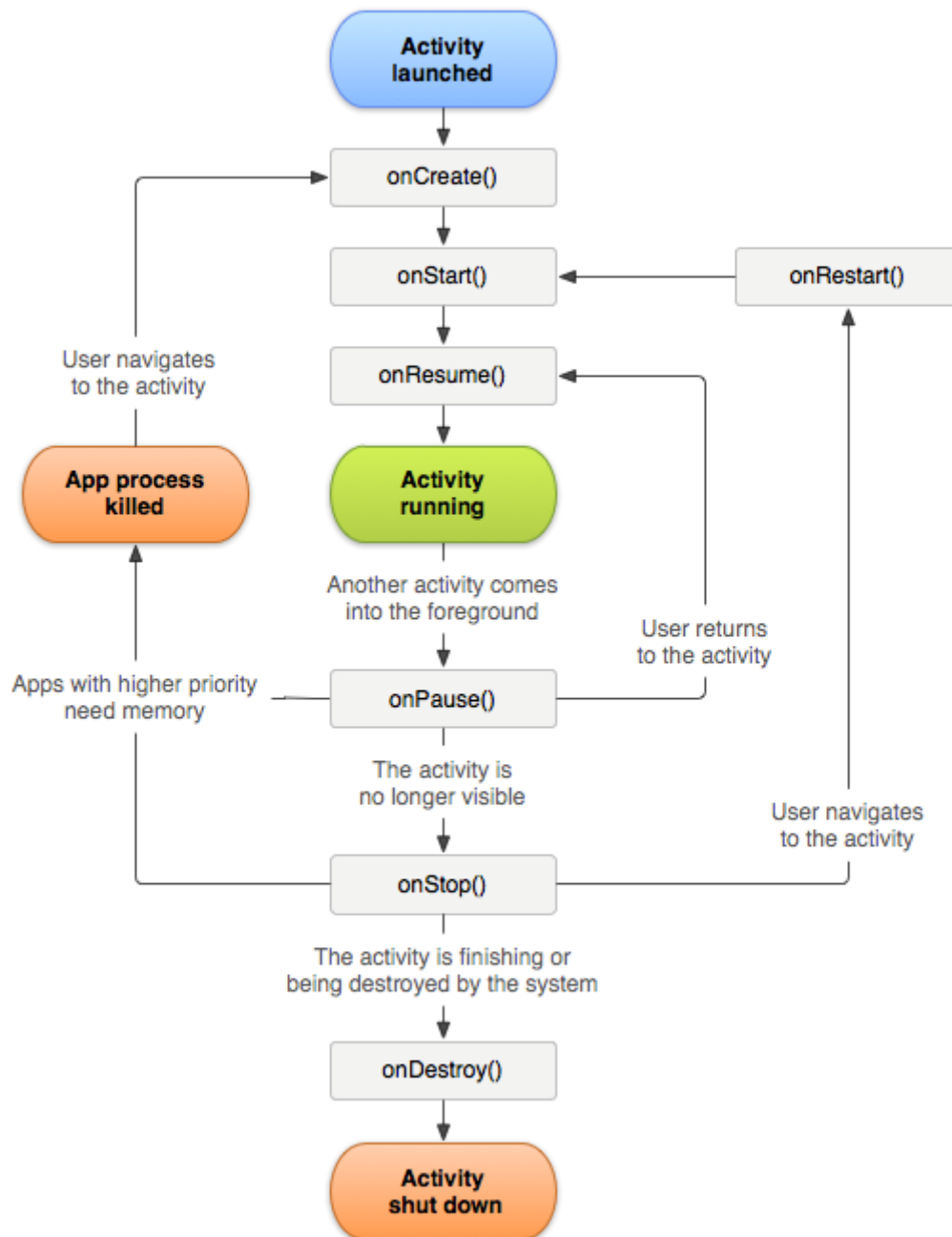
Aplikacije so najvišji sloj arhitekture operacijskega sistema Android. Nekatere aplikacije so vključene že v osnovni različici Androida ob nakupu pametnega mobilnega telefona. To so npr. sporočanja, e-pošta, budilka, koledar, stiki, zemljevid, Google Play in podobno. Vse aplikacije so napisane v programskem jeziku Java. Poleg nje se za razvoj programov uporablja tudi XML. Vse prevedene aplikacije se zapakirajo v datoteko s končnico .apk. To datoteko lahko zaženemo in predstavlja osnovo za distribucijo in nameščanje aplikacije na napravo. Vsaka aplikacija se zažene v svojem Linux procesu.

#### **4.4 Življenjski cikel aktivnosti Android aplikacije**

Koncept aktivnost predstavlja posamezni zaslon aplikacije z uporabniškim vmesnikom, na primer seznam prejetih sporočil. Druga aktivnost lahko vsebuje vsebino sporočil, tretja pa na primer izbiro prejemnika posameznega sporočila. Aktivnost ima štiri pomembna stanja življenjske dobe. Ko preklopimo na drugo aktivnosti se aktivnost, ki je trenutno delovala, neha izvajati in jo sistem potisne v zgodovinski sklad, s tem ohrani podatke za kasnejše nadaljnjo izvajanje. Aktivnost lahko iz sklada tudi odstranimo če je več ne potrebujemo ali pa nočemo po ne potrebi polniti pomnilnik. Življenjski cikel aktivnosti je prikazan na sliki. Štiri pomembnejša stanja življenjske dobe aktivnosti so:

- Aktivna aktivnost (*ang. Active activity*) aktivnost je prikazana na zaslonu v ospredju in se odziva na uporabnikove akcije (dotik zaslona, pritiske tipk ...),
- Prekinjena aktivnost (*ang. Paused activity*) aktivnost se ne odziva na uporabnikove akcije, vendar je še vedno vsaj delno vidna na zaslonu (izbira opcij, potrjevanje dialogov ...),

- Ustavljena aktivnost (*ang. Stopped activity*) aktivnost na zaslonu ni več vidna, a je še vedno na zgodovinskem skladu,
- Uničena aktivnost (*ang. Destroyed activity*) aktivnost ni na zaslonu in je tudi ni več na zgodovinskem skladu.



Slika 4: Življenjski cikel aktivnosti Android aplikacije

#### **4.5 Prednosti platforme Android**

Največja prednost platforme Android v primerjavi z ostalimi je v tem, da je odprtokodna in brezplačna. To je dobro tudi za uporabnike, saj imajo na razpolago mobilne naprave s še več funkcionalnostmi in večje število aplikacij, od tega večina brezplačnih. Domači proizvajalci lahko sedaj ceneje in lažje realizirajo svoje ideje in jih ponudijo na trg. Razvijalci lahko koristijo vse možnosti strojnih platform, kot so npr. zaslon na dotik, GPS, kamera, mikrofoni, kompas ipd. Aplikacije lahko vključujejo Java, JavaScript, HTML in CSS. Za to je na voljo tudi brezplačno razvojno okolje SDK (Software development kit). Vse naprave se samodejno sinhronizirajo z Googlovimi storitvami in omogočajo enostavno posodobitev, ko je na voljo novejša različica.

## 5 RAZVOJNO OKOLJE

Za razvoj Android aplikacij potrebujemo zagonsko in razvojno okolje. Med razvojna okolja spadata tudi Eclipse, z vtičnikom ADT in Android Studio. Za testiranje aplikacij lahko uporabljamo navidezno napravo, ki je vključena v programskem paketu Android SDK, ali pa uporabljamo kar dejanske naprave z operacijskim sistemom Android.

### 5.1 Razvojno okolje Android Studio

Razvojno okolje Android studio je odprtokodni in brezplačni program, ki ga je ustvarilo podjetje Google. Namenjen je razvoju aplikacij na platformi Android. Prvič je bil predstavljen 13.5.2013 na Google I/O konferenci. Temelji na programski opremi JetBrains IntelliJ IDEA in je namenjen samo za razvoju Android aplikacij. Prva stabilna različica je bila »Android Studio v1.0«, ki je bila izdana v decembru 2014, trenutna različica pa je »Android studio v2.2.3«, ki je bila izdana 6. decembra 2016. Razvojno okolje Android Studio je na voljo za Windows, Linux in Mac OS X in je zamenjala Eclipse Android Development Tools (ADT) kot Googlov primarni IDE za razvijanje android aplikacij.



*Slika 5: Android studio Logo*



## 5.2 Android SDK Android

Android software development kit je programski paket, ki vsebuje celovit paket orodji za razvijalce. Vsebuje izvorno kodo, razvojno orodje, navidezno napravo (Emulator), dokumentacijo, razhroščevalnik, knjižnice, ki so ključnega pomena pri izgradnji Android aplikacij. Podpira vse novejšie različice Linux, Mac OS X 10.5.8 in kasnejše in Microsoft Windows 7 in kasnejše verzije. Knjižnice so napisane v programskem jeziku Java, poganja pa jih navidezni stroj ART. Navidezna naprava (Emulator) omogoča izvajanje aplikacije, ki jo razvijamo na navidezni ciljni napravi in je v veliko pomoč pri testiranju aplikacij. V programskem paketu so na začetku samo osnovna orodja, v nadaljevanju pa moramo dodajati komponente, ki jih potrebujemo pri razvoju kompleksnejših aplikacij. Orodja SDK dodajamo preko managerja SDK, ki ga vključuje program Android Studio. Android SDK vsebuje tudi starejše verzije platforme Android, tako da lahko razvijalci razvijajo aplikacije na starejših platformah.

## 5.3 Android API

API je ogrodje, ki ima unikatno oznako za določeno verzijo platforme Android. Operacijski sistem razvijalci ves čas razvijajo in posodablajo, prav tako pa tudi ogrodje API. Vsak novejši API je združljiv s predhodnim, vendar le do določene verzije. Razvijalci morajo pri razvoju določiti najnižjo in ciljno številko različice API. Na spodnji sliki lahko vidima stopnje Android API in distribucijo določene verzije.

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.6%
4.1.x	Jelly Bean	16	6.0%
4.2.x		17	8.3%
4.3		18	2.4%
4.4	KitKat	19	29.2%
5.0	Lollipop	21	14.1%
5.1		22	21.4%
6.0	Marshmallow	23	15.2%

Slika 6: API za različne verzije Androida

#### 5.4 Navidezna mobilna naprava (Emulator)

V paket Android SDK je vključena navidezna mobilna naprava, ki omogoča sprotno testiranje aplikacije in iskanje napak z razhroščevalnikom. Priporočljivo je, da aplikacijo, ki jo razvijamo, testiramo na čim več različnih mobilnih napravah, saj je na ta način mogoče, da najdemo napako, ki se morda pojavi smo na nekaterih napravah, katero pol odpravimo, če najdemo razlog za to napako. Navidezna naprava ima možnost predstavljanja različnih naprav, saj ji lahko določimo velikost in ločljivost zaslona, različico ogrodja API, procesorjev in nastavljanje velikosti delovnega in notranjega pomnilnika. Lahko pa uporabimo tudi dejanske naprave, ki morajo imeti operacijski sistem Android in ki jih priklopimo na računalnik. Izkazalo se je, da je testiranje na dejanskih napravah boljše ter hitrejše, saj navidezne naprave potrebujejo več časa za zagon in zmogljivejše računalnike na katerih jih zaženemo.

#### 5.5 Programski jezik Java

Java je objektno usmerjeni, prenosljivi programski jezik, ki ga je razvil James Gosling s sodelavci v podjetju Sun Microsystems. Projekt, ki se je v začetku (leta 1991) imenoval Oak (hrast), je bil razvit kot zamenjava za C++. Z javo se da programirati tudi aplikacije na mobilnih telefonih (J2ME) in pametnih telefonih z operacijskim sistemom Android. Trenutna različica je Java 8, ki je edina različica, ki jo podpira Oracle brezplačno, starejše različice pa so podprte na podlagi tržnih zakonitosti.



*Slika 7: Java Logo*

## **6 RAZVOJ APLIKACIJE EDUQUEST**

Za razvoj aplikacije smo se odločili, ker smo želeli ustvariti igro, s katero preverimo splošno znanje uporabnika/igralca. Aplikacija oz. igra deluje na pametnih mobilnih napravah z operacijskim sistemom Android. Aplikacijo smo razvijali v razvojnem okolju Android Studio, ki je opisan v poglavju 5. Aplikacija še ni imela nobenih nadgradenj in je trenutno še prototip. Pri razvoju aplikacije smo uporabili dva programska jezika, to sta Java, za funkcionalnost aplikacije in XML, ki zajema grafični vmesnik aplikacije.

### **6.1 Začetek aplikacije**

Ko zaženemo aplikacijo se pojavi glavni meni. V glavnem meniju imamo gumbе: Nova igra, Nadaljuj in Izhod. Če pritisnemo na Nova igra se zažene nova aktivnost, ki postavi glavnega lika na določeno mesto v začetnem mestu. Statistične podatke glavnega lika an Stopnji 1 smo določili tako, da so prvi nasprotni liki v okolici mesta primerne težavnosti.

Ob pritisku na gumb Nadaljuj aplikacija poišče datoteko, v kateri so shranjeni podatki o najnovejši shranjeni točki v igri. Glede na podatke datoteke aplikacija nastavi lika na pravilno lokacijo, Stopnjo, doda predmete, opremo in denar ter Izkušnjske točke.

Gumb Izhod nam zapre aplikacijo.

### **6.2 Premikanje po zemljevidu**

Vsak zemljevid je sestavljen iz ploščic tako, da vsak premik v eno smer je v bistvu premik glavnega lika za eno ploščico v tisto smer. Za nadzor lokacije glavnega lika smo uporabili koordinate. Vsaka ploščica ima svoje koordinate (x, y), ki aplikaciji povedo kje se nahaja lik.

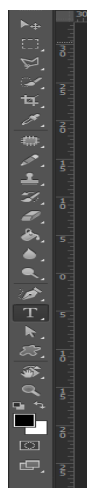
### **6.3 Formula za napad**

Ob bitki se zažene nova aktivnost. Delovanje smo opisali v poglavju 3.2. Ob napadu aplikacija vzame Napadalno moč napadalca in Obrambno moč napadenega in od Napadalne moči odšteje Obrambno moč ter rezultat odšteje od Življenjskih točk napadenega.

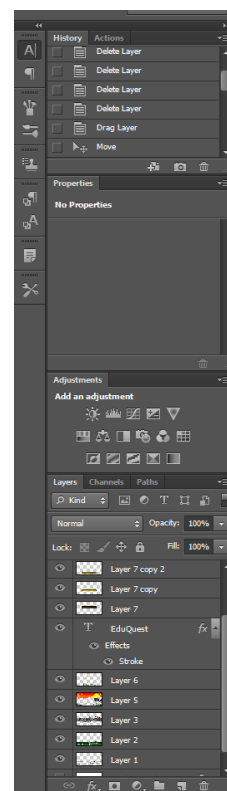
## 7 IZDELAVA GRAFIČNIH LIKOV

Adobe Photoshop je profesionalni računalniški program za obdelavo fotografij in drugih grafik. Razvili so ga pri podjetju Adobe Systems, ki je trenutno eden izmed vodilnih izdelovalcev grafičnih programov na svetu. Za izdelavo grafičnih likov smo uporabljali program Adobe Photoshop, v katerem je zelo preprosto risati, ker imamo možnost delati z layeri, ki pohitrijo delo. Photoshop je zelo preprost za uporabo, ker je uporabniku prijazen in ponuja veliko orodji, ki so zelo uporabne in olajšajo delo. Kot na primer:

- svinčnik,
- čopič,
- layeri,
- layer maske,
- radirka,
- hitra izbira,
- in tako dalje...



*Slika 8: Orodja  
(avtor naloge)*

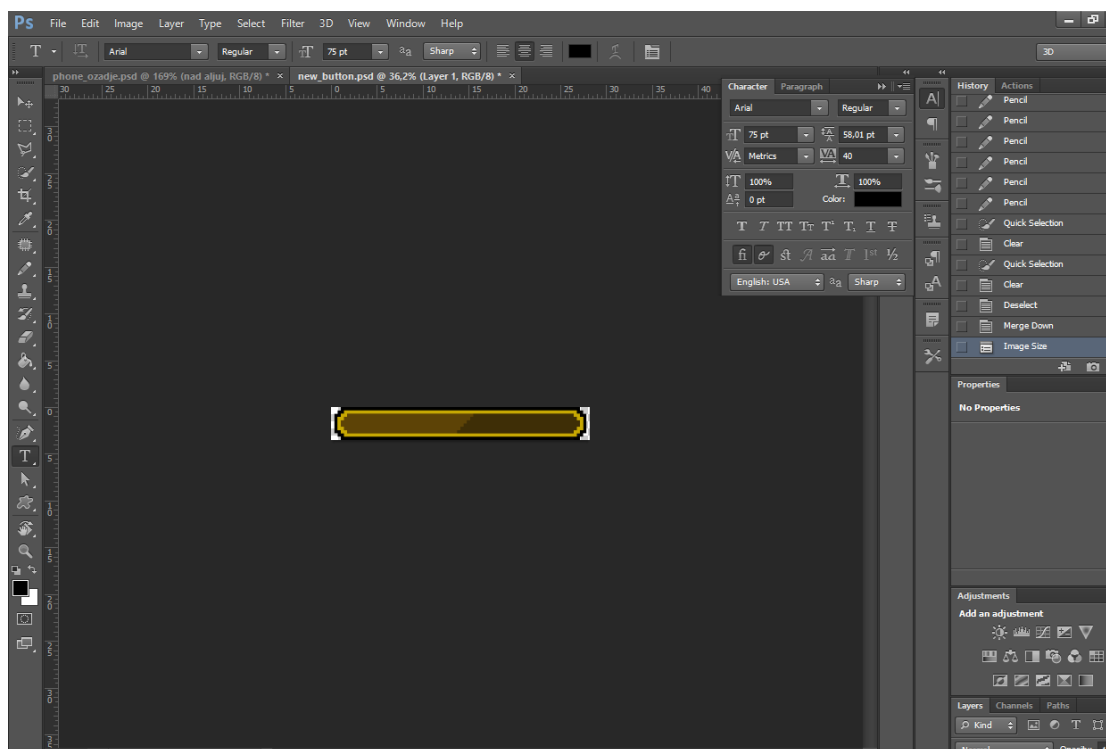


*Slika 9: Layeri (avtor naloge)*

Za nas je bilo najbolj pomembno, da smo lahko izbirali velikosti dokumenta, saj smo se odločili, da bodo naši liki narisani v bitni obliki (pixli). Zato smo uporabljali majhne velikosti dokumentov kot so na primer: 20x40px ali pa 10x80px in tako dalje. Ko smo dokončali naš lik smo samo povečali velikost dokumenta in lik se je z njim povečal. Na tak način smo iz majhne slike (10x80px) dobili 100x800px veliko sliko. Zelo dolgo smo mislili kako naj začnemo z liki in kako naj sploh izgledajo ali bodo realistični, fantastični, miti, legende, ali si jih bomo sami izmislili, ali jih bomo kopirali iz interneta. Na koncu smo se odločili da bomo sami poskusili risati. Potem smo se morali odločiti ali bodo ljudje ali pošasti in smo prišli do zaključka da bo vsega malo. Nato smo se še morali odločiti kakšen izgled bodo imeli naši grafični liki ali bodo v bitni grafiki (pixli), ali bodo realistični ali, narisani in tako dalje. Na koncu smo se odločili da nam je vsem všeč bitna grafika torej bodo bitni. Pogledali smo si tudi ostale igre na Google Play Storju in smo videli da se je večina iger, ki so podobna naši, odločila za risanje v bitni grafiki zato smo še bili bolj prepričani, da bitna grafika najbolj ustreza igri. Rišemo v RGB (red, green, blue) načinu, ta način nam najbolj ustreza saj nam omogoča veliko izbiro barv, prosto izbiro in likov tudi ne bomo natisnili, torej ne bomo rabili CMYK modela (cyan, magenta, yellow, black). Če ba bomo v nadaljevanju rabili bomo pa pretvorili RGB v CMYK model saj je z Photoshopom to zelo preprosto. Morali smo tudi paziti na barve v naši igri, ali bomo uporabljali tople ali mrzle barve, svetle ali temne, in verjetno bomo uporabljali vse barve. Inspiracijo smo vsi dobili iz

naših izkušenj s takšnimi igrami, saj vsi igramo igre takšne vrste in potem smo se spomnili naših najlepših iger, kakšni liki so tam bili in smo poskusili izdelati nekaj podobnega. Poskušamo tudi uporabljati takšne barve, ki označujejo vsakega lika posebej na primer: Lik z rdečo barvo se bo zaljubil ali pa kaj podobnega, lik modre barve bo pogumen in pameten, lik zelene barve bo spreten in potrpežljiv ipd. V prihodnosti bomo poskusili to čim bolj temu ustrezati. Nato pridemo do ozadja vsake scene, ki je tudi čustveno povezana glede na to kako vpliva na igralca. Če je temna lahko igralec pričakuje, da so bo zgodilo nekaj groznega, če bo svetla, bo nekaj dobrega in podobno. Barve so zelo pomembne, ker zelo vplivajo na igralčevo dožemanje igre. Poimenovanje vsakega izdelka je tudi pomembno saj s programi lažje najdemo sliko, ki jo potrebujemo tisti pri kakšnem klicanju. Slike oz. izdelke shranjujemo v dveh oblikah: .jpeg – joint photographic expert group in .psd – photoshop data file. Shranjujemo pa zato ker .jpeg potrebujemo za programiranje, .psd pa uporabljamo zato, da če se zmotimo lahko popravimo napako ali pa, če si premislimo lahko spremenimo in še potem enkrat shranimo v .jpeg.

Pisavi za našo igro sta Aria in Old English Text MT. Pisavi sta zelo različni saj je naslov tudi sestavljen iz dveh besed, ki imata različne pomene. Edu kot education, kar pomeni izobraževanje, torej iz tega lahko uporabniki sklepajo, da naša igra ne bo brez pomena, ampak se bo ob igranju veliko naučil in pridobil tudi nekaj splošnega znanja. Quest pa pomeni nekakšno potovanje, naloga, ki jo moreš dokončati, ki bo polna različnih dogodkov ipd. Pisavi sta si zelo različni, kar je zelo ne vsakdanje in zato tudi izstopamo od konkurence, kar pomeni, da bo uporabniku, ki brska Google Play takoj padlo v oči, da ni enaka pisava, kot so navajeni pri drugih igrah.



*Slika 10: Izdelava gumba za glavni meni (avtor naloge)*



*Slika 12: Glavni meni (avtor naloge)*



*Slika 11: Glavni lik igre (avtor naloge)*



*Slika 14: Prototip lika (avtor naloge)*



*Slika 13: Prototip lika (avtor naloge)*

## 8 SKLEP

Na začetku raziskovalne naloge smo najprej predstavili funkcionalnost aplikacije, njeno delovanje in smisel. Predstavili smo tudi tehnologije, potrebne za razvoj zastavljene aplikacije. Podrobno smo predstavili mobilno platformo Android ter prikazali trenutno najbolj uporabljene različice operacijskega sistema Android. Ugotovili smo, da je najbolj uporabna različica 5.0 in 5.1 Lollipop. Nato smo razčlenili sloje arhitekture operacijskega sistema in prikazali stanja življenjskega cikla aktivnosti aplikacije Android.

V nadaljevanju smo predstavili uporabljen prosto dostopno razvojno okolje za razvijanje aplikacij na mobilni platformi Android. To okolje je Android Studio. Poleg tega smo predstavili še dodatke, brez katerih v programskem jeziku Java ne moremo programirati Android aplikacij. Nekaj smo še povedali o programskem jeziku Java, s katerim smo napisali kodo za aplikacijo.

Predstavili smo tudi okolje za grafično oblikovanje Adobe Photoshop brez katerega naša aplikacija nebi mogla delovati, saj je potrebovala izgled, pri katerem smo se odločili za bitno grafiko saj najbolj ustreza takšnemu tipu igre. Adobe Photoshop nudi tudi funkcije, ki nam olajšajo delo in je preprost za uporabo.

V praktičnem delu raziskovalne naloge smo se osredotočili na razvoj igre EduQuest, ki smo jo razvili za mobilne naprave z nameščenim operacijskim sistemom Android. Aplikacija je namenjena vsem, ki si želijo zabave obenem pa tudi pridobiti nekaj splošnega znanja pri igranju igre. Na koncu smo izvedli preizkušanje aplikacije na naših lastnih mobilnih telefonih, kar je pripomoglo k uspešnejšemu ugotavljanju napak v sistemu in odkrivanju nekaterih hroščev, ki jih je bilo potrebno odpraviti. Skozi celoten proces razvoja so bili v pomoč prosto dostopni primeri na spletu, s katerimi smo si tudi pomagali pri razumevanju kode.

V nadaljevanju še želimo izboljšati aplikacijo saj je trenutna različica le prototip, izboljšave vsebujejo izboljšanje kode in pa tudi prenoviti/preurediti nekatera okolja, saj se z razvojem igre in ideje spreminja tudi scena. Igri imamo tudi namen dodati svojo zgodbo, ki jo bomo pripovedovali z igranjem in nadaljevanjem igre, našemu glavnemu liku bomo dodali smisel za njegovo pustolovščino.

## **DRUŽBENA ODGOVORNOST**

Trdimo, da naša aplikacija pozitivno vpliva na igralce/uporabnike, saj jim pomaga pri učenju. V aplikacijo bi lahko v prihodnosti vnesli vprašanja, ki so bolj specializirana za nekatera področja, kar bi ljudem, ki se učijo tista področja zelo pomagalo. Uporabniki so nagrajeni za njihovo znanje, kar jih vzpodbuja k učenju.



## **VIRI**

Operacijski sistem Android [online]. slideshare.net. 2016. [25.1.2016]. Dostopno na: <http://www.slideshare.net/akleema/presentation-on-android-operating-system>

Android Studio [online]. developer.android.com. 2016. [26.1.2016]. Dostopno na: <https://developer.android.com/studio/intro/index.html>

Java programski jezik [online]. ibm.com. 2016. [27.1.2016]. Dostopno na: <https://www.ibm.com/developerworks/java/tutorials/j-introtojava1/>

Adobe Photoshop [online]. jura.wi.mit.edu. 2016. [27.1.2016]. Dostopno na: <http://jura.wi.mit.edu/bio/graphics/photoshop/intro.php>

## **SLIKOVNI VIRI**

Slika 1: Ikona aplikacije (avtor naloge)

Slika 2: Primer bitke iz igre Final Fantasy IV. (26.1.2017) Dostopno na: [http://vignette3.wikia.nocookie.net/finalfantasy/images/e/ec/FFIV\\_Fight.jpg/revision/latest?cb=20080418215525](http://vignette3.wikia.nocookie.net/finalfantasy/images/e/ec/FFIV_Fight.jpg/revision/latest?cb=20080418215525)

Slika 3: Arhitektura Operacijskega sistema Android. (26.1.2017) Dostopno na: [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)#/media/File:Android-System-Architecture.svg](https://en.wikipedia.org/wiki/Android_(operating_system)#/media/File:Android-System-Architecture.svg)

Slika 4: Življenjski cikel aktivnosti Android aplikacije. (26.1.2017) Dostopno na: <http://www.javatpoint.com/images/androidimages/Android-Activity-Lifecycle.png>

Slika 5: Android studio Logo. (26.1.2017) Dostopno na: [http://www.androidpolice.com/wp-content/uploads/2014/11/nexus2cee\\_new-splash.png](http://www.androidpolice.com/wp-content/uploads/2014/11/nexus2cee_new-splash.png)

Slika 6: API za različne verzije Androida. (26.1.2017) Dostopno na: <https://i.stack.imgur.com/8tesD.png>

Slika 7: Java Logo. (26.1.2017) Dostopno na: [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)#/media/File:Java\\_programming\\_language\\_logo.svg](https://en.wikipedia.org/wiki/Java_(programming_language)#/media/File:Java_programming_language_logo.svg)

Slika 8: Orodja (avtor naloge)

Slika 9: Layeri (avtor naloge)

Slika 10: Izdelava gumba za glavni meni (avtor naloge)

Slika 11: Glavni meni (avtor naloge)

Slika 12: Glavni lik igre (avtor naloge)

Slika 13: Prototip lika (avtor naloge)

Slika 14: Prototip lika (avtor naloge)