



MESTNA OBČINA CELJE

MESTNA OBČINA CELJE
KOMISIJA MLADI ZA CELJE

ANDROID APLIKACIJA ZAVAROVALNIŠKI AGENT

Raziskovalna naloga

AVTORJI

Matej Simonič
Sergey Lobanov
Andraž Podpečan

MENTOR

mag. Boštjan Resinovič, univ. dipl. inž. rač. inf.

Mestna občina Celje, Mladi za Celje
Celje, 2013/2014



ŠOLSKI CENTER CELJE
Srednja šola za kemijo, elektrotehniko in računalništvo
Pot na Lavo 22
3000 Celje

ANDROID APLIKACIJA ZAVAROVALNIŠKI AGENT

Raziskovalna naloga

AVTORJI:

Matej Simonič, R4B
Sergey Lobanov, R4B
Andraž Podpečan, R4B

MENTOR:

mag. Boštjan Resinovič, univ. dipl. inž. inf.

Mestna občina Celje, Mladi za Celje
Celje, 2013/2014

IZJAVA*

Mentor, mag. Boštjan Resinovič v skladu z 2. In 17. Členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z

Naslovom ANDROID APLIKACIJA ZAVAROVALNIŠKI AGENT,

Katere avtorji so Matej Simonič, Sergey Lobanov, Andraž Podpečan:

- Besedilo v tiskani elektronski obliki istovetno,
- Pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- De je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- Da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na spletnih portalih z navedbo, da je nastala v okviru projekta Mladi za Celje,
- Da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- Da smo seznanjeni z razispnimi pogoji projekta Mladi za Celje.

Celje, 13.3.2013

Žig šole

Šola

Šolski center Celje, Srednja
šola za
kemijo, elektrotehniko in
računalništvo

Podpis mentorja

Podpis odgovorne osebe

*

POJASNILO

V skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja(-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja(-ice) fotografskega gradiva, katerega ni avtor(-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

KAZALO

KAZALO SLIK.....	5
POVZETEK	6
ABSTRACT	6
KLJUČNE BESEDE	7
UVOD	8
PREDSTAVITEV PROBLEMA	8
HIPOTEZE	8
RAZISKOVALNE METODE	9
REZULTATI ANKETE.....	9
MOBILNA APLIKACIJA	11
ANDROID	12
APP INVENTOR	13
APP INVENTOR APLIKACIJA	14
IGRA AVTOMOBIL	15
IGRA COLLECTOR	16
IGRA KVIZ.....	17
NAJPOGOSTEJE UPORABLJENE FUNKCIJE:	18
Math (matematične funkcije).....	18
Variable (spremenljivka).....	19
Orientation sensor(senzor orijentacije)	19
TinyDB	19
Clock component (ura)	20
Label (komponenta za prikazovanje teksta).....	21
Button (gumb)	22
ImageSprite	23
Canvas (ozadje).....	25
Player (predvajalnik).....	26
ZAKLJUČEK	27
VIRI	28
ZAHVALA.....	29

KAZALO SLIK

Slika 1:1. vprašanje ankete	9
Slika 2:2. vprašanje ankete	10
Slika 3:3. vprašanje ankete	10
Slika 4:4. vprašanje ankete	10
Slika 5: mobilne aplikacije	11
Slika 6:Android logotip	12
Slika 7:AppInventor	13
Slika 8:AppInventor aplikacije	14
Slika 9:Igra avtomobil	15
Slika 10:Igra collector	16
Slika 11:Igra kviz	17
Slika 12:Matematične funkcije	18
Slika 13:Spremenljivka.....	19
Slika 14:TinyDB sklici	19
Slika 15:Ura.....	20
Slika 16:Label	21
Slika 17:Gumb.....	22
Slika 18:Lastnosti ure.....	22
Slika 19:ImageSprite	23
Slika 20:Lastnosti ImageSprita.....	23
Slika 21:Ozadje	25
Slika 22:Predvajalnik.....	26

POVZETEK

V naši raziskovalni nalogi smo si zadali problem. Želeli smo ustvariti aplikacijo, ki bo tako mlajšim, kot starejšim na zanimiv in zabaven način predstavila zavarovalništvo. Raziskava je zajela predvsem področji pametnih telefonov in izdelavo aplikacije na platformi android. Pri pripravi raziskovalne naloge smo tudi uporabili anketo z mnenji anketirancev. Preden smo anketo objavili za reševanje, smo morali pripraviti tudi analizo problema. Anketa je vsebovala vprašanja kot so, ali anketiranec uporablja pametni telefon, ali uporablja na telefonu operacijski sistem android, ali igra igre na svojem telefonu.

ABSTRACT

With this research paper, we wanted to create an android application, which will introduce insurance policies in a fun and interesting way to our users. Our research primarily focused on smartphones and android application development. We created a short survey to get our potential users' opinion, but before that, we prepared an analysis of our problem. The survey included questions such as, does the person use a smart phone, does he use a smartphone with an android operating system, does the person play games on his smartphone.

KLJUČNE BESEDE

Android – operacijski sistem za mobilne naprave.

AppInventor – orodje za razvoj android aplikacij.

Pametni telefoni – je mobilni telefon, ki ponuja naprednejše računalniške sposobnosti.

Mobilna aplikacija – program za mobilni telefon.

Symbian – operacijski sistem za mobilne naprave.

UVOD

Naša pot skozi srednjo šolo je imela dosti ovir, težav a seveda smo spoznali tudi ogromno novih in zanimivih stvari. S pomočjo profesorjev, smo skozi leta spoznali veliko računalniških področij, med katerimi je bilo tudi programiranje. Kot ljubitelji iger, bodi si na računalniku, pametnem telefonu ali igralni konzoli, smo se odločili za izdelavo android igre kot temo naše maturitetne naloge. Kot vemo, je spoznavanje vseh zavarovalniških pol lahko mučno, dolgočasno ter dolgotrajno, da izberemo najboljšo za naše potrebe, zato smo se odločili, da izdelamo android igro, ki vas bo na zanimiv in zabaven način seznanila z zavarovalništvom.

PREDSTAVITEV PROBLEMA

Naša naloga je bila izdelati igro za pametne telefone na platformi android. Namen je bil izdelati igro ki bo imela različne stopnje, za vsako stopnjo bo drugačna igra. Skozi vsako stopnjo bo igralec dobil informacije o različnih zavarovalniških polah, katerih znanje bo uporabljal v zadnji stopnji, ki bo kviz o zavarovalništvu.

Za izdelavo tega programa smo potrebovali app inventor, ki nam omogoča razvoj iger in drugih aplikacij na platformi android, na enostaven način.

HIPOTEZE

- Vsaj 80% anketiranih uporablja pametne telefone, z android operacijskim sistemom.
- Anketirani v večini igrajo igre na svojih pametnih telefonih.
- Igra bo za mlajše in za starejše.
- Igralci bodo lahko na zabaven in na zanimiv način prišli do novega znanja.

RAZISKOVALNE METODE

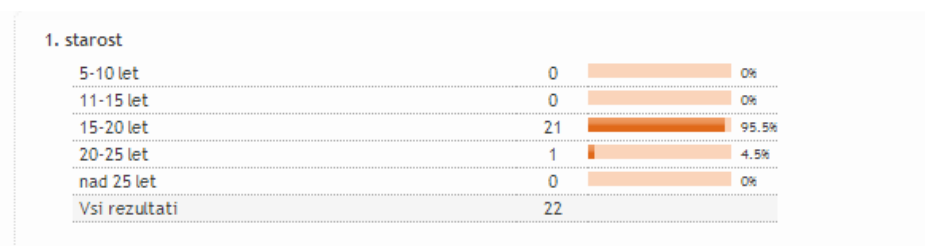
Naloga, ki smo si jo zadali, se nam je že na začetku zdela zahtevna, vendar uporabna, še posebno osebam, ki so se komaj začeli seznanjati z zavarovalništvom. Ko smo pridobili potrebno gradivo, smo najprej naredili načrt, kjer smo združili ideje in predloge.

Kmalu je sledila izbira razvijalnega okolja, v katerem bomo delali. Ker je App Inventor enostaven za uporabo in hrati dosti obširen, smo izbrali App Inventor. Za tem smo pričeli s programiranjem. Najprej smo se morali odločiti, kako bo sploh naša aplikacija zgledala in na kakšen način bo delovala. Odločili smo se za enostaven prvi zaslon, ki bo uporabniku ponudil različne gumbe (igraj, nastavitve, pole, napredek, izhod), kjer bo ob pritisku na gumb »igraj«, lahko izbral različne stopnje igranja. Za vsako stopnjo bo nova igra. Čez različne stopnje, bo igralec zbiral zavarovalniške pole, katere si bo moral vsaj prebrati, se pravi se seznaniti z njihovimi lastnostmi, da bo lahko opravil zadnjo stopnjo, katera bo kviz.

Na koncu smo pripravili še krajšo anketo, katero smo pridobili tudi informacije (uporaba pametnih telefonov, ali uporabljajo na telefonu operacijski sistem android, ali igrajo igre na svojih telefonih) o potencialnih uporabnikih.

REZULTATI ANKETE

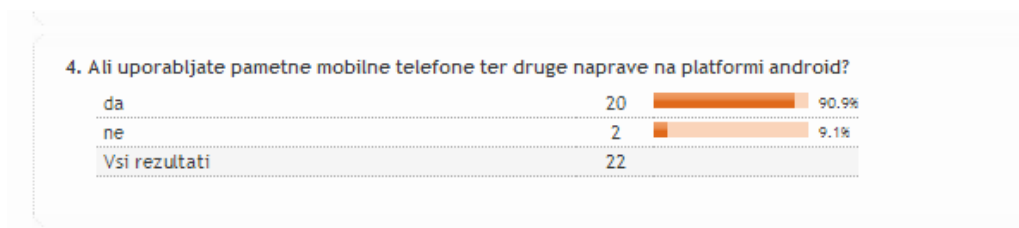
K sodelovanju v naši anketi smo povabili naključno izbrano populacijo v Sloveniji. S pomočjo Facebook-a, smo anketo oglaševali, in dobili rezultate. V anketi je sodelovalo 22 oseb do nad 25 let. Večina odzivov, ki smo jih dobili je bilo oseb do 20 let, kar je dobro, saj z aplikacijo želimo pomagati predvsem mlajšim, ki so še mogoče odvisni od staršev.



Slika 1:1. vprašanje ankete

Anketa je bila sestavljena iz dveh delov. V prvem delu smo želeli izvedeti starost, spol in vrsto izobraževanja. V drugem delu smo želeli izvedeti ali te osebe uporabljajo pametne telefone, ali igrajo igrice na pametnih telefonih in ali uporabljajo pametne telefone z android operacijskim sistemom.

V tem delu ankete, smo anketirance vprašali, ali uporabljajo pametne telefone ter druge naprave na platformi android. Odgovor »da« je izbralo 90.9% anketirancev in 9,1% »ne«, zato si lahko zagotovimo, da smo izbrali pravilno platformo za našo igro.



Slika 2:2. vprašanje ankete

Nato smo anketirance vprašali, ali igrajo igre na svojih pametnih telefonih. Zopet je »da« izbralo 90.9% anketirancev in »ne« 9.1%, iz česa lahko sklepamo, da smo izbrali pravilno vrsto aplikacije za android.



Slika 3:3. vprašanje ankete

V zadnjem vprašanju, nas je zanimalo ali jih zanima zavarovalniška dejavnost. Z »ne«, je odgovorilo 68.2% anketirancev, z »da« 9.1% in z »malo« 22.7%. Glede na te rezultate, si verjetno nismo izbrali najbolj zanimive teme, vendar menimo da vsaj malo znanja o zavarovalništvu je potrebno vsakomur in da bo



Slika 4:4. vprašanje ankete

MOBILNA APLIKACIJA

Mobilna aplikacija (ang. mobile application ali mobile app), je program narejen za pametne telefone, tablice in ostale mobilne naprave.

Začele so se pojavljati v letu 2008 in so večinoma dosegljive preko lastnikov mobilnega operacijskega sistema, kot so Google Play, App Store in Windows Phone Store. Nekatere aplikacije so brezplačne, medtem ko je druge potrebno kupiti.

Kot zanimivost vam lahko povemo, da je beseda "app" oz. "application software" postala zelo popularna. V letu 2010 jo je American Dialect Society označil za besedo leta oz. "Word of the Year". V letu 2009 pa je znani pisec tehnoloških kolumn David Pogue rekel, da bi lahko novim pametnim telefonim dali vzdevek "app phone" tako, da bi jih lahko ločili od prejšnjih manj sefisticiranih telefonov (<http://www.american dialect.org/app-voted-2010-word-of-the-year-by-the-american-dialect-society-updated>(13.3.2014)).

Mobilne aplikacije so bile sprva ponujene uporabnikom za splošno produktivnost in pridobivanje izgubljenih (pozabljenih) podatkov, vključno s spletno pošto, koledarji, vpogledi na borzo, informacijami glede vremena, ipd. Kmalu pa so mobilne aplikacije zaradi velikega zanimanja postale priljubljene tudi na drugih področjih, predvsem pri razvijanju mobilnih iger pa tudi za ugotavljanje lokacije (GPS), pri bančništvu, nakupu kart in še mnogo drugje.

Povpraševanje po teh aplikacijah še vedno narašča in s tem se močno dviga njihova uporaba po vsem svetu. Raziskava, ki so jo opravili pri podjetju comScore (ukvarjajo se s spletnimi analizami) je pokazala, da več uporabnikov mobilnih telefonov uporablja svoje naprave za izvajanje raznih mobilnih aplikacij, kot jih uporabljajo za brskanje po spletu.

Nekateri največji distributerji mobilnih aplikacij Amazon Appstore, Apple App Store, Blackberry App World, Google Play, Microsoft Windows Phone Store.

Zaradi vseh informacij, ki smo jih dobili od drugih, se je tudi naše mišljenje delno spremenilo in sedaj mislimo, da so mobilne aplikacije »velika nova stvar« (the next big thing) na svetu, ker so vedno bolj razširjene, dobesedno vsi jih uporabljamo in menimo da nam bo znanje o njih koristilo tudi v prihodnosti.

Razvijanje aplikacij za mobilne telefone zahteva upoštevanje zmogljivosti telefonov. Mobilni telefoni delujejo na baterijah in imajo manj zmogljive procesorje kot osebni računalniki. Razvijalci morajo tudi razmisliti o velikostih zaslonov, specifikacijah strojne opreme in konfiguraciji zaradi visoke konkurence na področju mobilnih aplikacij.



Slika 5: mobilne aplikacije

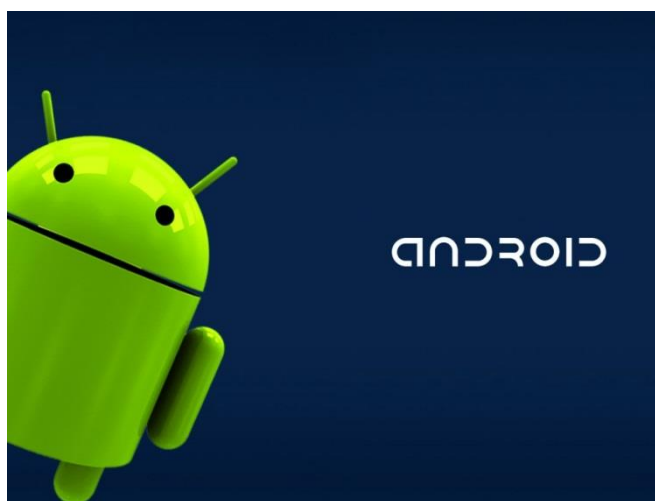
ANDROID

Android je operacijski sistem, ki je baziran na Linuxu in je bil narejen z namenom, da se go bo uporabljalo na pametnih telefonih in na tabličnih računalnikih. Najprej je bil razvijalec Android z finančno podporo Goolga, vendar ga je v letu 2005 Google kupil. Prvi android pametni telefon so predstavili leta 2007. Oktobra 2007 so prodali prvi pametni telefon z operacijskim sistemom Android.

Android je odprtokodni mobilni operacijski sistem, ki ga je google izdal pod licenco Apache. Ta odprta koda in licenca omogoča, da se aplikacije lažje razvijajo in da proizvajalci lažje pspreminjajo programsko kodo v svoje namene. Android ima veliko množico razvijalcev, ki pišejo aplikacije v prilagojeni verziji Jave. Vse te aplikacije se lahko naložijo zastonj za manjšo vsoto denarja iz spletne trgovine Google Play, včasih imenovana Android Market. V letu 2012 je bilo na voljo okoli 600.000 aplikacij za Android in imela je tudi 20 milijonov downloadanih aplikacij([http://techcrunch.com/2012/06/27/google-play/\(13.3.2014\)\)](http://techcrunch.com/2012/06/27/google-play/(13.3.2014)))).

Ta dejstva so Androidu omogočila postati največji in najbolj razširjen razvijalec pametnih telefonov in s tem prevzela vodstvo od Symbiana v drugi polovici 2010([http://www.osnews.com/story/24351/Android_Topples_Symbian_in_Q4_2010/\(13.3.2014\)\)](http://www.osnews.com/story/24351/Android_Topples_Symbian_in_Q4_2010/(13.3.2014)))). To je vodilo tudi v to, da čeprav je bil Android primarno mišljen za pametne telefone in tablične računalnike, so se kasneje začele razvijati tudi aplikacije za televizijo, igralne konzole in druge elektronske naprave.

Android je v drugi polovici 2012 imel 75% tržni delež, s 500 milijoni aktivnih mobilnih naprav in 1.3 milijona novih aktivacij vsak dan([http://techcrunch.com/2012/11/02/idc-android-market-share-reached-75-worldwide-in-q3-2012/\(13.3.2014\)\)](http://techcrunch.com/2012/11/02/idc-android-market-share-reached-75-worldwide-in-q3-2012/(13.3.2014)))) Zaradi svoje popularnosti je velikokrat tarča sporov s patenti v tako imenovani »smartphone vojni« med vodilnimi podjetij.



Slika 6:Android logotip

APP INVENTOR

App Inventor je zelo perspektivno orodje za izdelavo programov za Android. Njegov avtor je Google, vendar pa ga je le-ta kmalu opustil in njegovo izvorno kodo prepustil razvijalcem kot brezplačno in odprto oz. Open Source. To je odlično izkoristil MIT, ki ga je še dopolnil z nadaljnjim razvojem bi lahko postal odlično orodje za lažjo izdelavo androidnih programov.

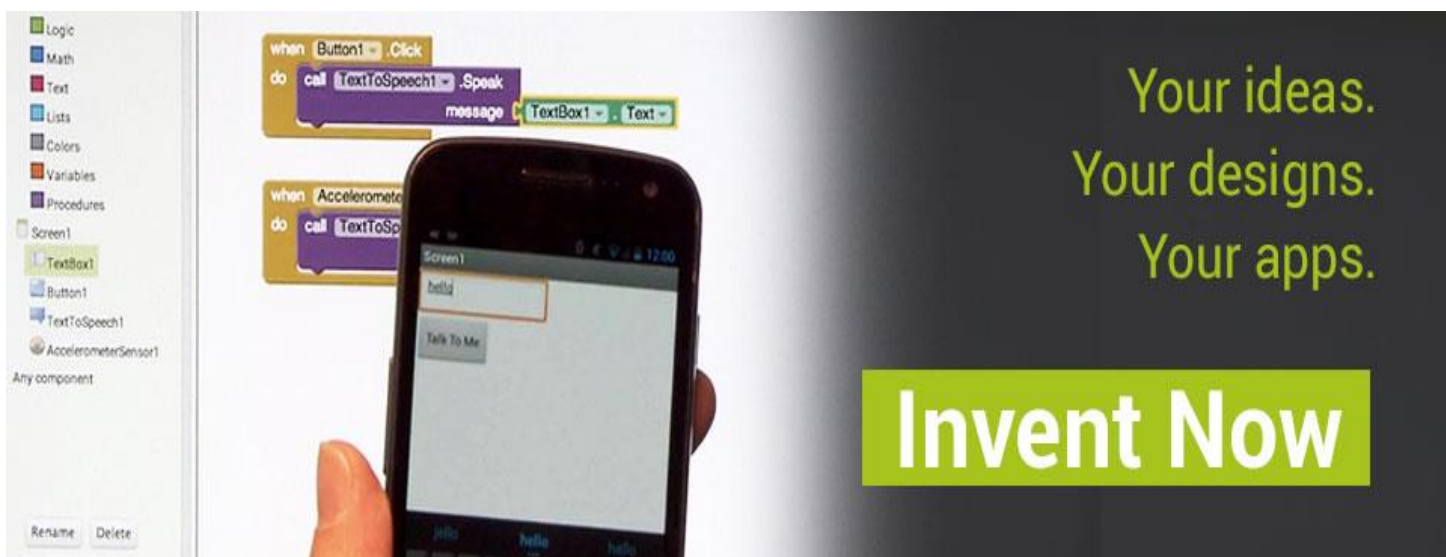
Je orodje, ki bi ga težko uvrstili ali med spletne ali med samostojne, programe saj je razvoj lastnega programa v brskalniku z načrtovanjem uporabniškega vmesnika, medtem ko vso logiko delovanja razvijemo z orodjem, ki je že nameščeno na računalnik. Izdelava programa traja dlje kot z drugimi podobnimi orodji, ima pa veliko prednost, saj lahko zgled popolnoma prilagodimo lastnim potrebam. Na začetku gre za nekakšno zlaganje kock, katerim kasneje določimo lastnosti. Ker pa seveda »gumbek v programu sam po sebi ni zmožen narediti čisto nič, je potrebno vsem elementom določiti še funkcije. Tu sledi malce težji del, kej je potrebno imeti vsaj osnovno programersko logiko, saj je na vrsti zlaganje blokov, ki je nadomestitev pisanja programske kode. Po končanem delu je možno napisani program tudi preizkusiti na mobilni napravi z operacijskim sistemom Android ter ga stestirati. Gre torej za zelo dobro, ne tako zapleteno orodje za katerega pa potrebujete malce več časa



Slika 7:AppInventor

APP INVENTOR APLIKACIJA

V naši nalogi smo z pomočjo programa App Inventor izdelali igro za pametne telefone. Igra je sestavljena iz večih delov. Prva igra je igra avtomobil, nato igra collector na koncu pa igralec prične reševati kviz. Pri igri avtomobil, igralec pobira kovčke, kateri dajo igralcu možnost, da že po preigrani igri lahko bere vsebino kovčkov, kjer so informacije o zavarovalništvu katere bo koristil pri kvizu. Igra collector deluje na podoben način kot igra avtomobil, saj moraš tudi tu pobirati kovčke z informacijami, vendar na malce drugačen način. Zadnja je igra kviz, kjer bo igralec uporabil znanje, ki ga je pridobil z igranjem prejšnjih dveh iger. Vprašanja temeljijo na osnovah zavarovalništva, ki niso zapletena, ampak tudi zelo uporabna. Vprašanja v kvizu se točkujejo z 1 točko, če je odgovor pravilen ali z 0, če je odgovor nepravilen. Igra se zaključi, ko igralec odgovori na vsa vprašanja.



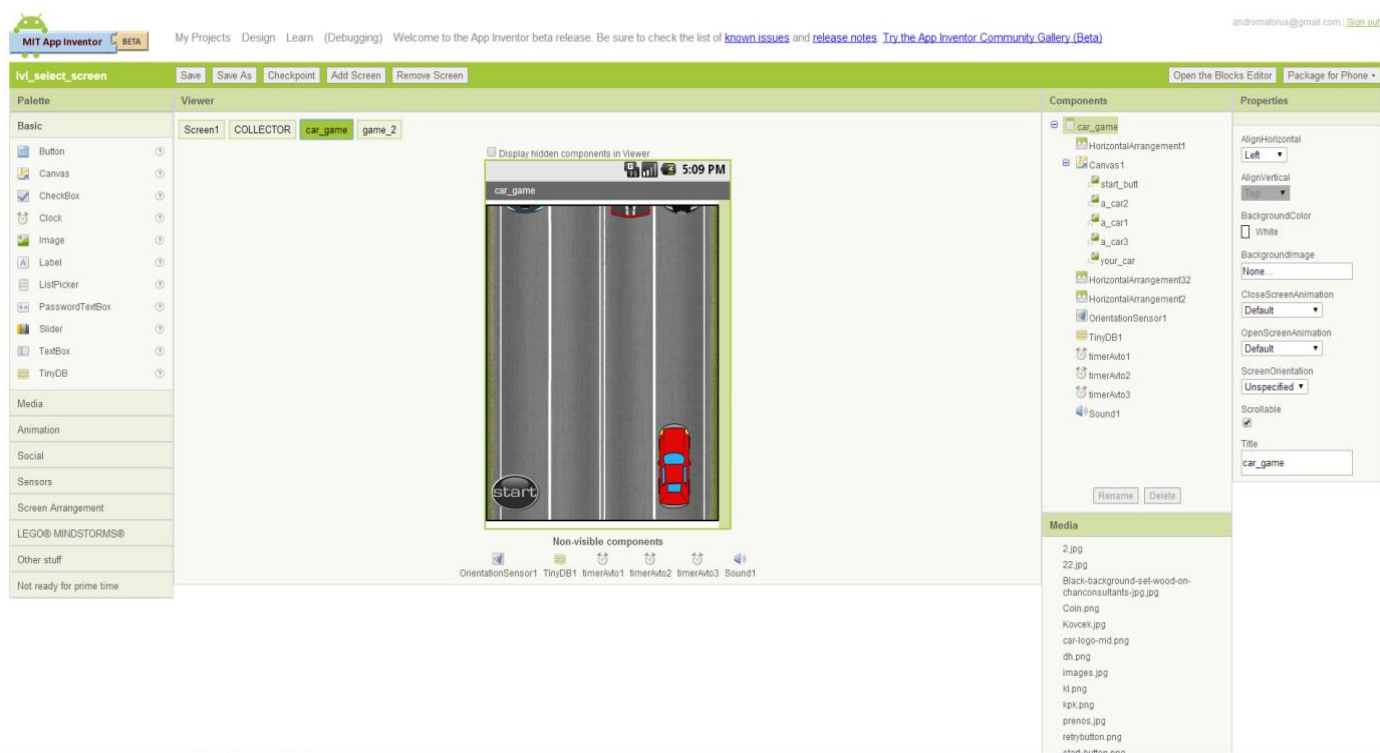
Slika 8:AppInventor aplikacije

IGRA AVTOMOBIL

S to igro je bilo največ dela. Potrebno je bilo uporabiti največ komponent, ki jih ponuja program App Inventor. Namen te igre je, da se izogibaš vozilom, ki se peljejo z nasprotne strani in hkrati poskuša pobrati kovčke, kateri vsebujejo pojme in razlage zavarovalništva. Avtomobil igralec kontrolira z nagibanjem telefona v levo in desno stran. Ko uspešno prevoziš določeno razdaljo napreduješ z naslednjo igro in sicer z igro collector.

Ideja za to igro je prišla na plan, ko smo se odločali, kakšne igre bi lahko uporabili v našem projektu. Avtomobilska igra se nam je zdela primerna, saj imamo v zavarovalništvu tudi veliko možnosti zavarovanj za avtomobile, prav tako tudi za voznike. Ker so igre z avtomobili tudi zelo priljubljene med uporabniki pametnih telefonov, smo se dokončno odločili, za našo prvo izmed treh iger.

Na spodnji sliki je prikazana igra avtomobil. Razvidno je da je zanjo bilo potrebno uporabiti kar veliko komponent, ki so pripomogle k pravilnemu delovanju igre. Igralec prične z igro ob pritisku gumba Start. Ko ga igralec stisne lahko začne upravljati avtomobil z nagibanjem in se izmika avtomobilom, ki se vozijo iz nasprotne smeri. Na cesti se pojavijo tudi kovčki z informacijami o zavarovalništvu.



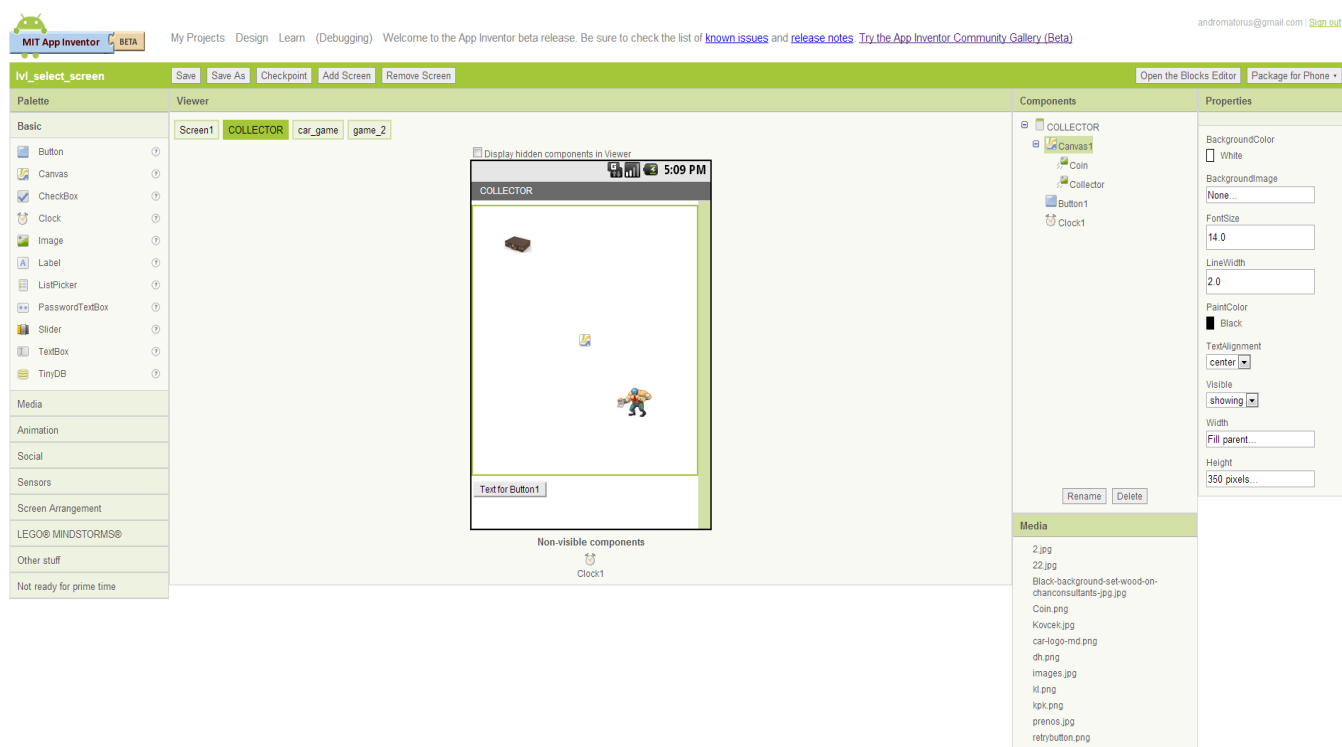
Slika 9:Igra avtomobil

IGRA COLLECTOR

Za to igro smo si lahko pomagali z vodiči na spletni strani App Inventor-ja, zato pri kreiranju ni prišlo do resnejših problemov. Pri tej igri igralec pobira kovčke, ki zopet vsebujejo pojme in razlage zavarovalništva. Kovčki se pojavijo in izginjajo v obratu treh sekund. Igralec mora za napredovanje pobrati deset polic. Ko zbere vse police napreduje v zadnjo igro, ki se imenuje kviz.

Za to igro smo se odločili zato, ker smo bili vsi v otroštvu ljubitelji igre Snake, katere igranje se nam zdi podobno, kot pri naši igri. Sprva si jo lahko igral samo na arkadah. Leta 1998 pa je bila igra na vseh Nokia mobilnih telefonih. Koncept igre je da s kačo, ki jo upravlja igralec pobereš vse predmete, ki so na igralni površini in se izmikaš robovom ekrana ter oviram. Kača se je večala z vsakim predmetom, ki si ga pobral ter postajala hitrejša. Ti spremembi sta igro zelo otežili.

Naša igra tako temelji na igri Snake. Edina razlika je, da se collector ne večja ter ne postaja hitrejši. Ker smo morali narediti igro tudi poučno, smo vpeljali vanjo zavarovalne pole, kjer



Slika 10:Igra collector

igralec pridobiva znanje o zavarovalništvu.

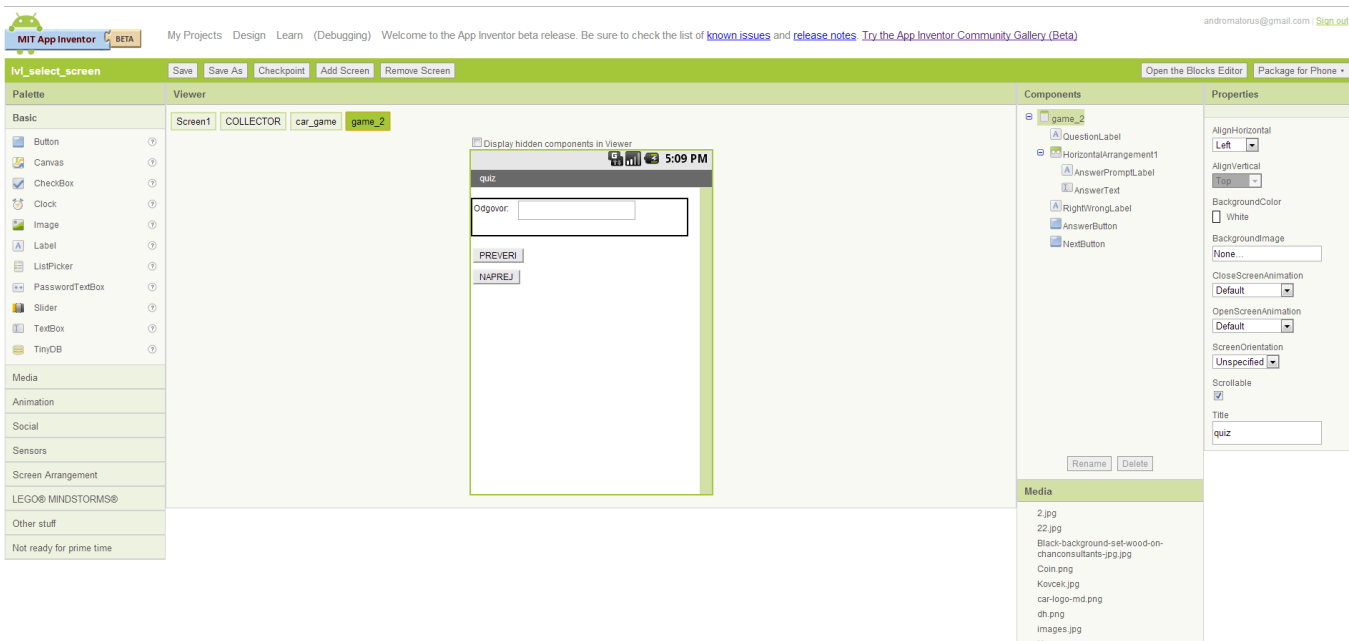
IGRA KVIZ

To je zadnja igra v naši aplikaciji. Tukaj igralec uporabi znanje, ki ga je pridobil z igranjem igre avtomobil in igre collector, kjer je pobiral kovčke, ki so vsebovali podatke o zavarovalništvu. Vprašanja v kvizu se točkujejo z 1 točko, če je odgovor pravilen ali z 0, če je odgovor nepravilen. Igra se zaključi, ko igralec odgovori na vsa vprašanja pravilno in s tem dokaže, da je s pomočjo naše igre prišel do novega znanja.

Na to idejo smo prišli tako, da smo se vprašali kako lahko igralec izve kaj se je naučil z igranjem naše igre. Vsem je na misel prišel kviz. V kvizu se lahko igralec na zabaven način preizkusi v znanju zavarovalništva.

Na tej sliki je prikazana zadnja igra, kviz. Ko igralec izbere odgovor z pritiskom na gumb PREVERI, če je izbral pravilni odgovor na vprašanje. Nato klikne gumb NAPREJ in prične z naslednjim vprašanjem.

Cilj ige je zbrati čim več točk. Odgovor mora igralec napisati. Odločitev je vedno dokončna, saj igralec svojega odgovora ne more več spremeniti.



Slika 11:Igra kviz

NAJPOGOSTEJE UPORABLJENE FUNKCIJE:

Math (matematične funkcije) – matematične funkcije so razdeljene na več delov:

- Določitev števila.
- Večje.
- Manjše.
- Večje ali enako.
- Manjše ali enako.
- Je enako.
- Seštevanje.
- Odštevanje.
- Množenje.
- Deljenje.
- Korenjenje.
- Naključen del (vrača naključne vrednosti med 0 in 1).
- Naključno celo število (vrne naključno celo število med dvema določenima vrednostima, vključno z obema vrednostima sama).
- Random set seed (to polje uporabimo za ustvarjanje ponovljivih zaporedij).



Slika 12: Matematične funkcije

Variable (spremenljivka) – ustvari vrednost, ki se lahko spreminja tudi, ko je aplikacija zagnana in dodeli tej vrednosti ime. Spremenljivke so »global«, kar pomeni da se lahko na njih sklicujemo, iz katerega koli dela kode aplikacije.



Slika 13:Spremenljivka

Orientation sensor(senzor orijentacije)

Senzor orijentacije je nevidna komponenta ki vrača tri nasljednje vrednosti prostorske orijentacije v stopinjah:

-Roll ima vrednost 0 ko je naprava v vodoravni legi in se večja do 90 ko se njena leva stran dviguje in se spušča do -90 medtem ko se desna dviguje.

-Pitch ima vrednost 0 ko je naprava v vodoravni legi in se večja do 90 ko je nagnjena z zgornjim delom navzdol in se začne zmanjševati ko se obrne ter obratno ko je nagnjena tako se zgornja stran dviguje se vrednost zmanjšuje do -90 in se nato začne vračati na 0 dokler

-Azimuth ima vrednost 0 ko je zgornja stran naprave obrnjena na sever, 90 ko je obrnjena na vzhod, 180 ko je obrnjena na jug, 270 ko je obrnjena na zahod ter vse vmesne vrednosti od 0 do 359.

TinyDB – app inventor ima poenostavljeno verzijo podatkovne baze, ki se imenuje TinyDB, ki shranjuje trajne podatke neposredno na napravo Android. (App Inventor ima tudi TinyWebDB komponento, ki shranjuje podatke na strežniku.

Metode:

- StoreValue (text tag, valueToStore) – shranjuje vrednosti po določeno oznako. Oznaka mora biti besedilni niz (text string), vrednost je lahko niz (string) ali seznam (list).

- GetValue (text tag) – dobi vrednost, ki je shranjeno pod določeno oznako. Če ni bilo shranjene vrednosti, vrne prazno besedilo.



Slika 14:TinyDB sklici

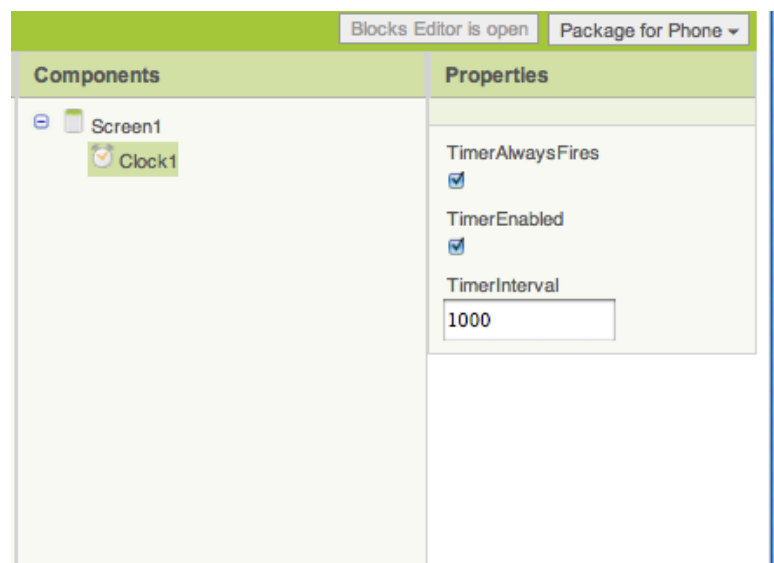
Clock component (ura) – ustvarimo časovnik, ki signalizira dogodke v rednih časovnih intervalih. Ura pa tudi omogoča različne konverzije in manipulacije s časovnimi enotami. Ena uporaba ure je časovnik, ki bo nastavil časovni interval (npr. 2 sekundi) in časovnik bo glede na nastavljen časovni interval, sprožil dogodek.

Lastnosti:

- TimerInterval - časovni interval v milisekundah.
- TimerEnabled - če nastavljen, bo časovnik sprožil dogodek.
- TimerAlwaysFires - če nastavljen, časovnik bo sprožil dogodek, tudi če aplikacija ni prikazana na zaslonu.

Dogodki:

- Timer() - ta dogodek se sproži, ko se časovnik sproži.

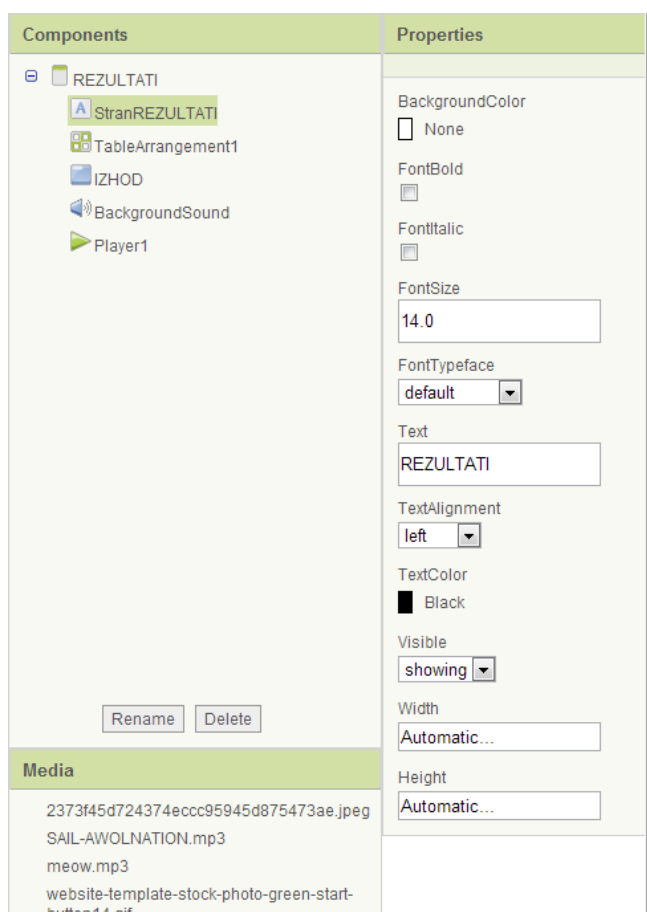


Slika 15:Ura

Label (komponenta za prikazovanje teksta) – prikazuje tekst, ki je določen z lastnostjo Text. S pomočjo »Blocks Editor-ja«, lahko razvijalec ureja izgled in postavitev teksta.

Lastnosti:

- BackgroundColor - barva ozadja.
- FontSize - velikost teksta.
- Height - višina.
- Width - širina.
- Text - tekst, ki bo prikazan.
- TextAlignment - poravnava teksta levo, desno ali sredina.
- TextColor - barva teksta.
- Visible - če nastavljeno, je tekst viden.



Slika 16:Label

Button (gumb) – gumbi so komponente, katerih se uporabnik dotakne, da v aplikaciji sproži nek dogodek. Veliko stvari glede izgleda je nastavljenih, po lastnih željah. Izberemo lastnost »Enabled«, da izberemo ali je gumb mogoče pritisniti.

Lastosti:

- BackgroundColor - barva gumba.
- Enabled - če nastavljeno, uporabnik lahko pritisne na gumb, za izvedbo ukaza.
- FontSize - velikost teksta v gumbu.
- Height - višina gumba.



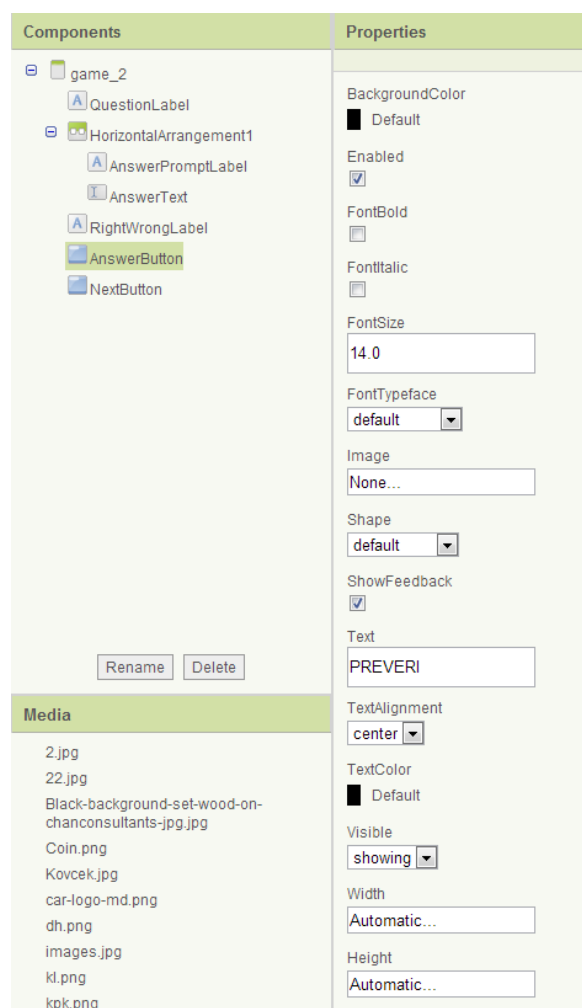
Slika 17:Gumb

- Width - širina gumba.
- Image - slika, ki bo prikazovala gumb.



Dogodki:

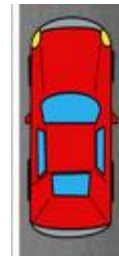
- Click () - uporabnik je pritisnil in spustil gumb.
- GotFocus() - gumb je postal osredotočena komponenta.
- LostFocus() - gumb ni več osredotočena komponenta.



Slika 18:Lastnosti ure

ImageSprite

ImageSprite je animiran objekt ki ga lahko uporabimo v zvezi z Canvas(ozadje), drugimi ImageSpritei, robi ozadja, se premika glede na njegove lastnosti ter reagira na dotike.



Slika
19:ImageSprite

Lastnosti:

Picture: slika prikazana za ImageSprite

Enable: če je vrednost true je ImageSprite aktiven

Interval: frekvenca gibanja v milisekundah

Picture: slika izgleda ImageSprita

Rotates: če je vrednost true se slika obrne tako da se uskladi z naslovom. Če je vrednost false se ImageSprite ne rotira ko se naslov spremeni

Visible: če je vrednost true je ImageSprite viden

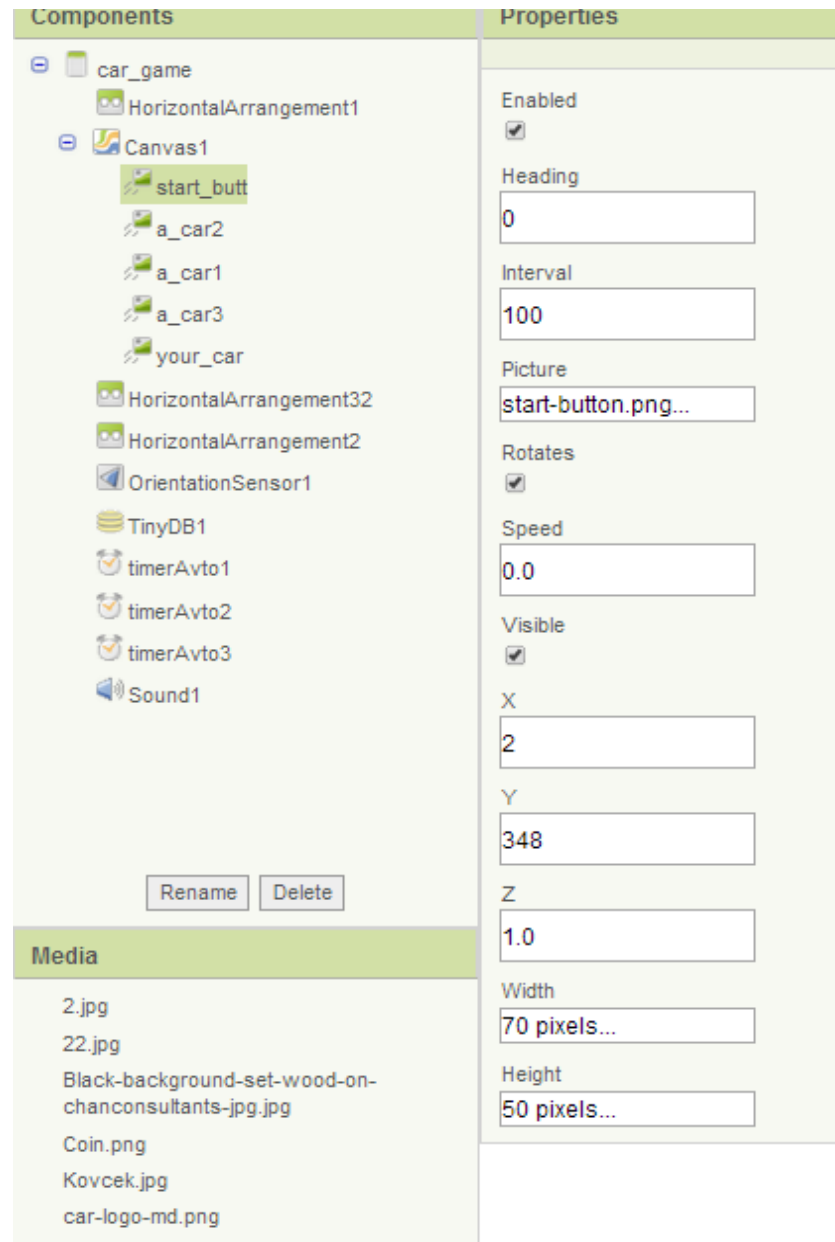
Heading: Smer premikanja image sprita v stopinjah, 0 je vodoravno desno(vzhod), 90 je navzgor, 180 je v levo in 270 je navzdol.

X: x koordinata ImageSprita. če ima vrednost 0 je ImageSprite na skrajnem levem robu ozadja

Y: y koordinata ImageSprita. če ima vrednost 0 je ImageSprite na skrajnem zgornjem robu ozadja

Speed: število za kolikor pixlov se vsakic premakne ImageSprite

Wight: Širina ImageSprita



Slika 20:Lastnosti ImageSprita

Height: Višina ImageSprit

Dogodki:

CollidedWith(): se kliše ko se dva animirana objekta zabijeta

Dragged(): ko je Image sprite povlečen

EdgeReached(): ta dogodek je klican ko ImageSprite pride do roba zaslona. Argument edge je številka roba ki se ga dotakne (1=sever, 2=severovzhod, 3=vzhod, 4=jugovzhod, -1=jug, -2=jugozahod, -3=zahod, -4=severozahod)

NoLongerCollidingWith(): je klicant ko se dva animirana objekta nehata dotikati

Touched(): je klican ko uporabnik pritisne ImageSprite

Metode:

Bounce(): odbije ImageSprite kot od zidu oziroma kota.

Boolean CollidingWith(): vrne ali se imagesprite dotika podanega argumenta

MoveIntoBounds(): če je ImageSprite izven okvirja ga metoda pomakne nazaj v okvir

MoveTo(): premakne ImageSprite točno na dano točko.

Canvas (ozadje) – dvodimenzionalna, na dotik občutljiva plošča, katero je možno premikati in vstaviti »image sprite«. Lahko nastavimo tudi poljubno sliko ozadja.

Katera koli lokacija na ozadju je določena z vrednostmi x in y kjer:

- X je število pikslov od levega roba ozadja.
- Y je število pikslov od zgornjega roba ozadja.

Lastnosti:

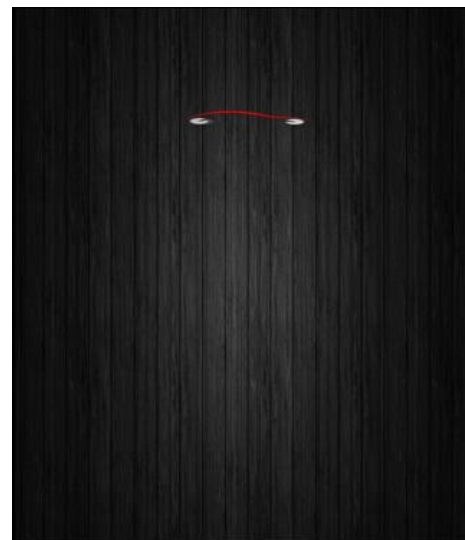
- BackgroundColor - barva ozadja.
- BackgroundImage - ime slike, ki je nastavljena kot slika ozadja.
- Height - višina.
- LineWidth - širina črt na ozadju.
- Visible - ali bo komponenta vidna.
- Width - širina.

Metode:

- Clear () - počisti ozadje, ne da bi odstranil.
- DrawCircle (number x , number y , number r) - nariše krog na danih koordinatah, pri določenem radiju.
- DrawLine (number x_1 , number y_1 , number x_2 , number y_2) – nariše črto med danimi koordinatami.
- DrawPoint (number x , number y) – nariše piko na danih koordinatah.
- Save () – shrani sliko tega ozadja na zunanji pomnilnik naprave in vrne polno ime poti do shranjene datoteke. Če pride do napake se izvede dogodek imenovan ErrorOccurred.

Dogodki:

- Dragged (number startX, number startY, number prevX, number prevY, number currentX, boolean draggedSprite) – Ko uporabnik povleče iz ene točke do druge.
- Flung (number startX, number startY, number xSpeed, number ySpeed) – ko uporabnik na hitro prst potegne čez ekran (fling).
- Touched (number x , number y , boolean touchedSprite) – ko se uporabnik dotakne ozadja.



Slika 21:Ozadje

Player (predvajalnik) – predvajalnik je komponenta, ki nam omogoča predvajanje zvočne in video datoteke ali vibracijo telefona.

Lastosti:

- IsLooping - če nastavljeno, se medij ponavlja.
- IsPlaying - ali se medij predvaja.
- Source - medij, ki je povezan s predvajalnikom.
- Volume - izbiramo glasnost med 0 in 100.

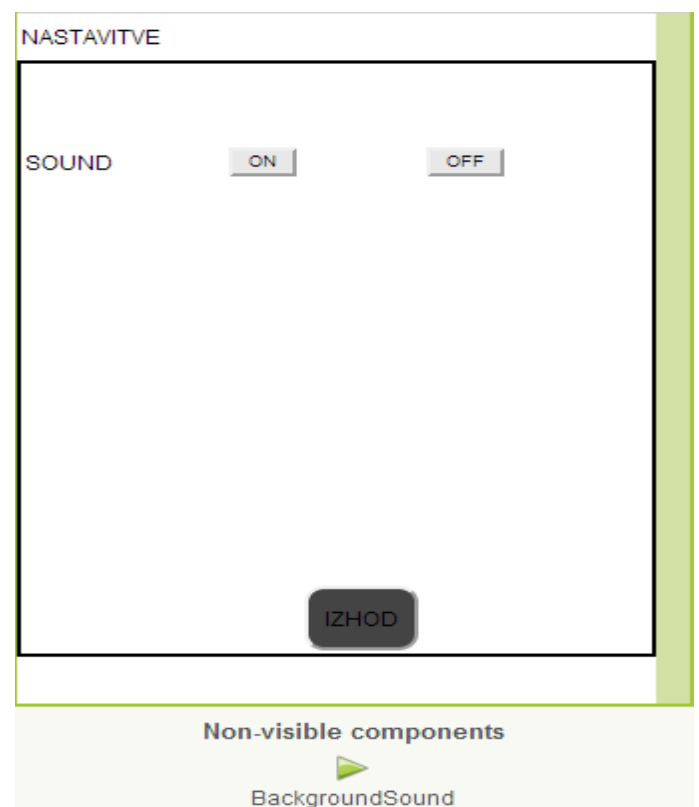
Metode:

- Pause () - predvajan medij da na premor.
- Start () - začne predvajati medij.
- Stop () - prekine predvajanje medija.
- Vibrate (število milisekund) – aktivira telefonovo vibracijo, za določeno število milisekund.

Dogodki:

- Completed (medij je končal predvajanje).

Slika 22:Predvajalnik



ZAKLJUČEK

Namen naše naloge je bil ustvariti mobilno aplikacijo, ki uporabniku pomaga pridobiti znanje o zavarovalništvu. S to nalogo smo uporabili znanje pridobljeno v štirih letih šolanja in ga nadgradili z ustvarjanjem aplikacije v popolnoma novem programskem okolju. To kar smo se naučili pri izdelavi nam lahko koristi v nadaljnjih okoliščinah programiranja. Ključnega pomena je bilo ekipno delo, saj je bila naloga izjemno obsežna. Prišli smo do nekaj zapletov, vendar na koncu nam je uspelo in z končnim izdelkom smo bili zadovoljni.

Glede na zastavljene cilje smo z izdelkom zelo zadovoljni. Vemo da naša aplikacija ne bo težko dostopna saj je brezplačna in jo lahko uporablja vsak, ki ima pametni telefon. Potrdimo lahko tudi to, da bo aplikacija zanimiva tako mlajšim kot starejšim, ki na pametnih telefonih igrajo igre in si želijo znanje pridobiti na zabaven način.

VIRI

App inventor. [splet]. [11. mar. 2014; 13:41]. Dostopno na spletnem mestu:
<http://appinventor.mit.edu/explore/>

App inventor forum. [splet]. [11. mar. 2014; 18.50]. Dostopno na spletnem mestu:
<http://appinventor.mit.edu/explore/forums/>

App inventor tutorials. [splet]. [2. feb. 2014; 15.00]. Dostopno na spletnem mestu:
<http://appinventor.mit.edu/explore/ai2/tutorials>

Aplikacija Get the gold. [splet]. [11. mar. 2014; 18.30]. Dostopno na spletnem mestu:
<http://appinventor.mit.edu/explore/get-gold.html>

ZAHVALA

Zahvalili bi se profesorju mag. Boštjanu Resinoviču, univ. dipl. inž. rač. inf. za podporo in pomoč pri izdelavi raziskovalne naloge. Z njegovo pomočjo nam je uspelo sprotno odpravljati napake in reševati težave, ki so nastajale pri nalogi.