

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Blaž Kokol

**Razvoj igre za platformo BlackBerry PlayBook s pomočjo
HTML5**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2014

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Blaž Kokol

**Razvoj igre za platformo BlackBerry PlayBook s pomočjo
HTML5**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana, 2014

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuirajo predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuirajo in/ali predelujejo pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Razvoj igre za BlackBerry PlayBook s pomočjo HTML5 zahteva iskanje konceptualnih rešitev, strokovno znanje in pridobivanje novih izkušenj, v smislu uporabe različne tehnologije, in uporabe možnosti, ki jo le-ta ponuja. Opisno in postopkovno predstavite tehnologijo, potek dela oziroma razvoj, implementacijo ter objavo končnega izdelka (igre) na spletnem portalu BlackBerry World.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Blaž Kokol, z vpisno številko **63070091**, sem avtor diplomskega dela z naslovom:

Razvoj igre za platformo BlackBerry PlayBook s pomočjo HTML5

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 22. decembra 2014

Podpis avtorja:

Zahvaljujem se svojemu mentorju, doc. dr. Roku Rupniku, za vodenje pri izdelavi in za strokovni pregled diplomskega dela. Hvala tudi družini, prijateljem in vsem ostalim, ki so me podpirali tekom študija in pri nastajanju diplomskega dela.

Kazalo

Povzetek

Abstract

| | | |
|-------------------|---------------------------------------------------------------|-----------|
| Poglavje 1 | Uvod | 1 |
| Poglavje 2 | BlackBerry PlayBook in uporabljene tehnologije | 2 |
| 2.1 | BlackBerry PlayBook | 2 |
| 2.2 | Spletne tehnologije | 3 |
| 2.2.1 | HTML5 | 3 |
| 2.2.2 | CSS3 | 5 |
| 2.2.3 | JavaScript | 6 |
| 2.3 | Razvojna okolja | 8 |
| 2.3.1 | NetBeans | 8 |
| 2.3.2 | GIMP | 9 |
| Poglavje 3 | Igra Time Attacker | 11 |
| 3.1 | Opis | 11 |
| 3.2 | Programska koda | 13 |
| 3.3 | Implementacija izdelka | 21 |
| 3.3.1 | Testiranje z emulatorjem | 21 |
| 3.3.2 | Testiranje na tabličnem računalniku BlackBerry PlayBook | 22 |
| 3.4 | Objava igre | 27 |
| Poglavje 4 | Sklepne ugotovitve | 32 |

Seznam uporabljenih kratic

| kratica | angleško | slovensko |
|----------------|--------------------------------------|---------------------------------------------|
| HTML | Hypertext Markup Language | Jezik za označevanje nadbesedila |
| CSS | Cascading Style Sheets | Kaskadne stilske podloge |
| XHTML | Extensible Hypertext Markup Language | Razširljiv jezik za označevanje nadbesedila |
| XML | Extensible Markup Language | Razširljiv označevalni jezik |
| JS | JavaScript Programming Language | Programski jezik JavaScript |
| IDE | Integrated development environment | Integrirano razvojno okolje |
| JVM | Java virtual machine | Javanski navidezni stroj |
| W3C | World Wide Web Consortium | Konzorcij za svetovni splet |

Povzetek

V pričujočem diplomskem delu je podrobno predstavljen razvoj igre za platformo BlackBerry PlayBook s pomočjo HTML5, in sicer pot od ideje do izdelka. Uvodoma je obravnavana omenjena platforma, uporabljene tehnologije (HTML5, CSS3, JavaScript) ter razvojna okolja (NetBeans, GIMP). Osrednji del je namenjen predstavitvi razvoja našega izdelka (igra Time Attacker), ki vključuje opis, programsko kodo, implementacijo ter objavo igre. Uporabnost našega dela vidimo v njegovi izobraževalni in praktični funkciji, saj je delo zasnovano tako, da bodočega uporabnika, bodisi študenta računalništva ali drugega zainteresiranega uporabnika, s strokovnimi opisi korakov in razlago vsebin vodi skozi potek izdelave igre, ki smo jo v končni fazi objavili na spletu.

Ključne besede: računalniška igra, BlackBerry PlayBook, HTML5

Abstract

The diploma paper presents in detail the development of game for BlackBerry PlayBook platform using HTML5, i. e. the path from idea to product. First, it examines the previously mentioned platform, the technologies used (HTML5, CSS3, JavaScript) and environments for development (NetBeans, GIMP). The central part is dedicated to the development of our product (Game Time Attacker), which includes a description of the program code, implementation and publication of the game. The usefulness of our work can be seen in its educational and practical function, because the work is designed for the future user or computer science student or other users, it contains professional descriptions of the steps and interpretations of the content through the course of manufacture of the game, which we ultimately posted it online.

Keywords: computer game, BlackBerry PlayBook, HTML5

Poglavje 1 Uvod

Diplomsko delo smo si zastavili kot izziv, prehoditi pot razvoja igre za platformo BlackBerry PlayBook, od ideje do izdelka.

Izhodišče našega razmišljanja je dejstvo, da je uporaba iger v porastu. In naše uvodno spoznanje je bilo, da je za izdelavo igre potrebno ogromno znanja mnogoterih oblik, ter spretnosti in ustvarjalnosti v iskanju aplikativnih rešitev. Za neveščega in neizkušenega uporabnika oziroma programerja razvoj igre predstavlja velik zalogaj. Da bi v strokovno polje prispevali svoj kamenček v mozaiku v smislu uporabnosti lastnega dela, smo si zastavili pomemben globalni cilj, to je: bodočemu uporabniku pokazati pot od ideje do izdelave igre in ponudbe le-te uporabnikom.

V našem delu bomo bralcu pokazali, da HTML okolje omogoča širok diapazon možnosti uporabe. Enako ugotavljamo tudi za JavaScript in CSS.

Omenimo še naše zavedanje, da igra sama po sebi nima neke vzgojne vrednosti, pa vendar je poznavanje njene izdelave za mladega strokovnjaka pomembno v smislu metodološkega preizkusa pridobivanja znanja, ki bo v prihodnje lahko uporabljeno pri reševanju drugih problemov. Naše delo je iz izobraževalnega vidika uporabno kot priročnik bodočemu uporabniku, kar štejemo za naš glavni prispevek diplomske naloge.

Poglavje 2 BlackBerry PlayBook in uporabljene tehnologije

2.1 BlackBerry PlayBook

Tablični računalnik BlackBerry PlayBook je bil izdan 19. aprila 2011 s strani podjetja Research In Motion (RIM) [1], ki se je v začetku leta 2013 preimenovalo in prevzelo ime po svojem najbolj znanem produktu BlackBerry. [2] Podjetje je za svoj tablični računalnik pripravilo nov operacijski sistem BlackBerry Tablet OS, ki temelji na operacijskem sistemu QNX Neutrino. Najnovejša verzija operacijskega sistema BlackBerry Tablet OS je verzija 2.1.0.1917, ki je izšla 31. marca 2014.



Slika 2.1: Tablični računalnik BlackBerry PlayBook [3]

Tablični računalnik, ki ga prikazuje zgornja slika, je opremljen z večočkovnim 17,8-centimetrskim zaslonom ločljivosti 1024*600 točk. Za njegovo delovanje skrbi 1 gigaherčni

dvojedrni procesor *ARM Cortex-A9*, sistemski pomnilnik kapacitete 1 gigabajt in grafična kartica *PowerVR SGX540*.

Tablični računalnik omogoča predvajanje in zajemanje videovsebin do formata Full HD (1080p). Za zajem video vsebin skrbita dve kameri. Sprednja kamera ima ločljivost treh milijonov točk, kamera na zadnji strani ima pet milijonov točk. Opremljen je s senzorji žiroskop, digitalni kompas (GPS) in pospeškometer. Za povezovanje z zunanjimi napravami in internetom so na voljo Wi-Fi (802.11 a/b/g/n), Bluetooth 3.1, microHDMI in microUSB. Slednji se uporablja tudi za polnjenje visokozmogljive neodstranljive Li-Po 5300mAh baterije, ki omogoča 10-urno delovanje tabličnega računalnika ob normalni uporabi. Modeli BlackBerry PlayBook-a se razlikujejo po velikosti pomnilnika. Na voljo sta različici z 32GB in 64GB pomnilnika, prav tako obstaja še model PlayBook 4G LTE, ki vsebuje 32GB pomnilnika in dodatno poleg zgoraj omenjenih specifikacij ponuja še možnost vstavitve SIM kartice in uporabo hitrega mobilnega prenosa podatkov [1], teoretično do hitrosti 326,4/86,4 Mbit/s. [4]

2.2 Spletne tehnologije

2.2.1 HTML5

V nadaljevanju bo predstavljen pregled razvoja HTML, jezika za označevanje nadbesedila, ki je skriptni jezik za izdelavo spletnih strani in je v sedanjem času praktično nepogrešljiv.

Začetki HTML-ja segajo v leto 1990, ko sta začela sodelovati znanstvenika Tim Berners-Lee in Robert Cailliau, takrat oba zaposlena v CERNU, evropski organizaciji za jedrske raziskave. HTML sta prvič predstavila leta 1993 v Genovi na prvi spletni konferenci.

Leta 1994 je Netscape predstavil brskalnik, ki je razvijalcem ponudil kreiranje značk, s katerimi so lahko prevzeli kontrolo nad izgledom spletnih strani. Med drugimi je bila predstavljena značka CENTER, ki besedilo premakne na sredino zaslona.

Leta 1995 je Microsoft predstavil značko MARQUEE, ki je omogočila pomik teksta preko celotnega zaslona. Tega leta se je pojavila verzija HTML3.0.

Dve leti kasneje je bila izdana popravljena verzija HTML3.2, ki je bila prva standardizirana verzija HTML-ja. Ta je nastala pod okriljem W3C. V njej so bile združene in poenotene značke, ki so jih uporabljali pri Netscape-u in Microsoft-u.

Leta 1999 je bila izdana verzija HTML4.01, ki se je obdržala vse do leta 2012. Leta 2000 je bil tudi uradni izid jezika XHTML1.0, ki je vključeval značilnosti HTML4 in XML. Po izidu XHTML1.0 se je organizacija W3C odločila, da ne bo nadaljevala razvoja HTML jezika in se je trudila izdati novo verzijo XHTML2.0, vendar le-ta ni dočakala uradnega izida, saj ni bila kompatibilna s prejšnjimi verzijami HTML in XHTML jezikov.

Leta 2004 so podjetja Apple, Mozilla in Opera ustanovila skupino WHATWG, ki je združevala strokovnjake z namenom ponovnega razvoja HTML jezika in sorodnih tehnologij. [5]

Leta 2007 so napovedali izid spletnega jezika HTML5.

Štiri leta kasneje (l. 2011) je bila uradna predstavitev spletnega jezika HTML5, skupaj s CSS3, ki je bil razvit s strani skupine WHATWG in organizacije W3C, nadomestil je prejšnji verziji spletnega jezika HTML4.01 in XHTML1.0, saj združuje specifikacije obeh. Prav tako se verzija spletnega jezika HTML5 še izboljšuje. [6]

Za leto 2016 je napovedan izid verzije HTML5.1. [7]

```
<!DOCTYPE html>
<html>
  <head>
    <title>Naslov</title>
  </head>
  <body>
    Vsebina spletne strani
  </body>
</html>
```

Slika 2.2: Sintaksa HTML5 (posnetek programske kode HTML5)

Zgornja slika prikazuje sintakso HTML5. Z deklaracijo `<!DOCTYPE html>` definiramo standard HTML5. Sintaksa vsebuje naslednje osnovne elemente (značke), v pravilnem vrstnem redu: `<html>`, `<head>` in `<body>`.

V znački `<head>` se opišejo informacije o html dokumentu, v znački `<body>` pa se nahaja vsebina spletne strani.

Nekaj najpomembnejših novosti, ki jih je prinesla nova verzija HTML5:

- krajša oblika deklaracije, ki služi spletnim brskalnikom za branje HTML dokumentov in prikaz vsebine: `<!DOCTYPE html>`;
- elementi za večpredstavnostne vsebine: `<video>`, `<audio>`;
- grafični elementi: `<svg>`, `<canvas>`;
- semantični elementi: `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, `<footer>`. [8]

HTML5 podpirajo vsi sodobni spletni brskalniki (Internet Explorer, Firefox, Google Chrome, Safari itd). [9] Za samo izdelavo spletnih strani v HTML5 nam zadostuje že navadna beležnica, v kateri kreiramo dokument s končnico .htm ali .html. Pri izdelavi dinamičnih spletnih strani uporabljamo HTML5 zgolj za opredelitev vsebine spletne strani, za stilsko oblikovanje spletne strani uporabljamo CSS3, z JS pa nadzorujemo obnašanje spletne strani. [10]

2.2.2 CSS3

Kaskadno stilska podloga (CSS) je preprost stilski jezik, ki skrbi za implementacijo in izgled spletnih strani oziroma spletnih aplikacij. CSS deluje pod okriljem skupine CSS Working Group, le-ta pa znotraj organizacije World Wide Web Consortium (W3C). [11]

Začetki razvoja CSS segajo v leto 1994, ko je Hakon Wium Lie objavil osnutek kaskadnih HTML stilskih predlog (CHSS). Bert Bos, ki je razvijal spletni brskalnik, ki bi podpiral stilske predloge z naslovom Argo, se je odločil pridružiti Hakon-u pri nadaljnjem razvoju CHSS, iz katerega sta odstranila besedo HTML in obdržala zgolj CSS.

Bert in Hakon sta še pred uradnim izidom predstavila CSS na konferenci aprila 1995 leta. Bert je predstavil podporo za stilske predloge v spletnem brskalniku Argo, Hakon pa je predstavil verzijo spletnega brskalnika Arena, ki je bil oblikovan tako, da je omogočal podporo CSS-u. Pri razvoju spletnega brskalnika Arena je sodeloval Dave Raggett, služil pa je kot testno okolje za nove razvojne možnosti, eden izmed njih je bil tudi CSS. Na konzorciju W3C leta 1995 je bil podprt nadaljnji razvoj CSS in čeprav še ni bilo uradnega izida, sta ga podprla spletna brskalnika Netscape in Microsoft Explorer, ki sta poenotila uporabo CSS-a.

Kmalu zatem, decembra 1996, je združenje W3C potrdilo končno verzijo CSS verzija 1. Znotraj organizacije W3C se je razvila skupina CSS Working Group, ki je in bo poskrbela za nadaljnji razvoj.

Skupina je že v maju 1998 izdala novo verzijo CSS2.

Leta 1998 so začeli vzporedno razvijati tudi verzijo CSS3, ki je razdeljena na module. [12] Izšla je leta 2011, vendar se moduli še zmeraj razvijajo in dodajajo, prav tako najnovejša verzija CSS3 podpira vse starejše verzije.

Leta 2005 je bila tudi izdana verzija CSS2.1, ki je odpravila vse odkrite pomanjkljivosti prejšnje verzije CSS2. [11]

Vse verzije CSS-a imajo enako sintakso (glej spodnjo sliko). Sestavljene so iz selektorja in deklaracije.



Slika 2.3: Sintaksa CSS [13]

Selektor je del pred zavitim oklepajem in kaže na element v HTML besedilu, katerega želimo oblikovati. Deklaracija se nahaja znotraj zavutih oklepajev in je sestavljena iz dveh delov, in sicer lastnosti in vrednosti. Vsakemu selektorju lahko dodelimo eno ali več deklaracij. Primer na zgornji sliki vsebuje 2 deklaraciji, prva določi lastnosti barve selektorja na vrednost modra, druga deklaracija določi lastnosti velikosti črk na vrednost 12px. Kot v drugih programskih jeziki lahko tudi v CSS uporabljamo komentarje. Komentar se začne z `»/*«` in konča z `»*/«`. Primer:

`/* besedilo h1 se obarva z modro barvo in določi velikost črk na 12px */.` [13]

V CSS3 se uporabljajo tudi predpone, v kolikor verzija spletnega brskalnika še ne podpira standardnega zapisa deklaracije. Predpone za spletne brskalnike:

- Internet Explorer: `-ms-`;
- Firefox: `-moz-`;
- Google Chrome, Safari, Opera: `-webkit-`.

Poglejmo primer standardnega zapisa `column-count: 3;` kateri deluje zgolj v Internet Explorerju. Za delovanje v spletnem brskalniku Firefox potrebujemo dodatno še deklaracijo `-moz-column-count: 3;`, za delovanje v spletnih brskalnikih Google Chrome-u, Safariju in Operi potrebujemo še dodatno zapis: `-webkit-column-count: 3;`. [14]

2.2.3 JavaScript

Skriptni programski jezik JS se je prvič pojavil leta 1995. Razvil ga je Brendan Eich, za spletni brskalnik Netscape Navigator 2, pod imenom LiveScript, vendar se je kmalu preimenoval v JS. Avgusta 1996 so tudi pri podjetju Microsoft vključili svojo verzijo JS v spletni brskalnik Internet Explorer, pod imenom JScript. [15], [16].

Novembra 1996 je podjetje Netscape naznanilo, da so poslali zahtevo za standardizacijo JS mednarodni neprofitni organizaciji za standardizacijo informacijskih in komunikacijskih sistemov (Ecma International). Standardizirano verzijo JS so poimenovali ECMAScript.

Junija 1997 je bila objavljena prva verzija standarda ECMA-262, ki temelji na standardu ISO/IEC-16262. Ta je bila uporabljena v JS verziji 1.2.

Že leta 2000 je bila objavljena tretja verzija standarda ECMA-262. Uporabljena je bila v JS verziji 1.5 in JScript verziji 5.5. [15] JS, JScript in tudi ActionScript so bolj znane implementacije ECMAScript oziroma standarda ECMA-262.

Junija 2011 je bil izdan trenutno aktualni standard ECMA-262 verzija 5.1. Uporabljajo jo JS verzija 1.8.5 in JScript verzija 9.0.

Za junij 2015 je napovedan izid novega standarda ECMA-262 verzija 6, prav tako je že tudi v fazi razvoja verzija 7, v kateri so objavljeni novosti. [17]

V nadaljevanju bomo predstavili sintakso oziroma način, kako so oblikovani stavki programskega jezika JS.

JS je občutljiv na velike in majhne črke. Poznamo:

- **Spremenljivke**

- Definiramo jih z besedo *var*. Primer: *var spremenljivka;*
- Tip spremenljivke se določi avtomatično, ko ji priredimo konkretno vrednost (številski, znakovni niz, polja znakov, objekti itd.).

- **Operatorje**

- Priredilni, aritmetični in bitni operatorji: =, +, -, *, / itd.
Primer: *spremenljivka = 6;*
- Pogojni, primerjalni in logični operatorji: ==, !=, <, > itd.

- **Funkcije**

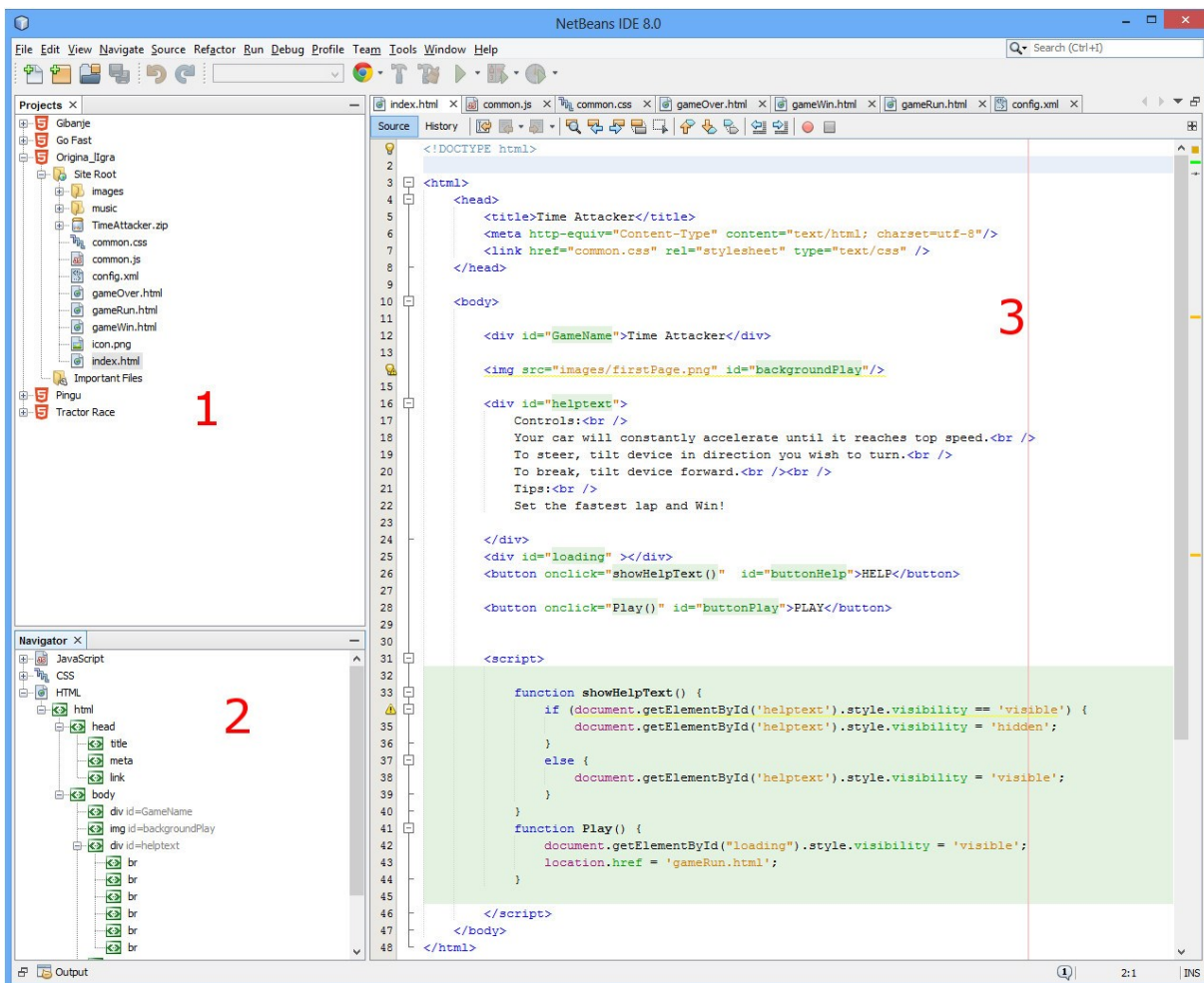
- Definiramo jih z besedo *function*.
Primer: *function MojaFunkcija(a, b){ return a+b;}*. [18]

2.3 Razvojna okolja

2.3.1 NetBeans

NetBeans IDE je odprtokodno integrirano razvojno okolje, razvito leta 2000 s strani podjetja Sun Microsystems, od leta 2010 pa je v lasti podjetja Oracle, ki je prevzelo podjetje Sun Microsystems. [19] Razvojno okolje je razvito s pomočjo tehnologije Java, kar pomeni, da se lahko uporablja na vseh operacijskih sistemih, kjer je nameščen JVM.

Pri našem delu smo se za uporabo razvojnega okolja NetBeans odločili predvsem zaradi dejstva, da je brezplačen in enostaven za uporabo ter podpira vse uporabljene spletne tehnologije, in sicer HTML, CSS in JS. [20]



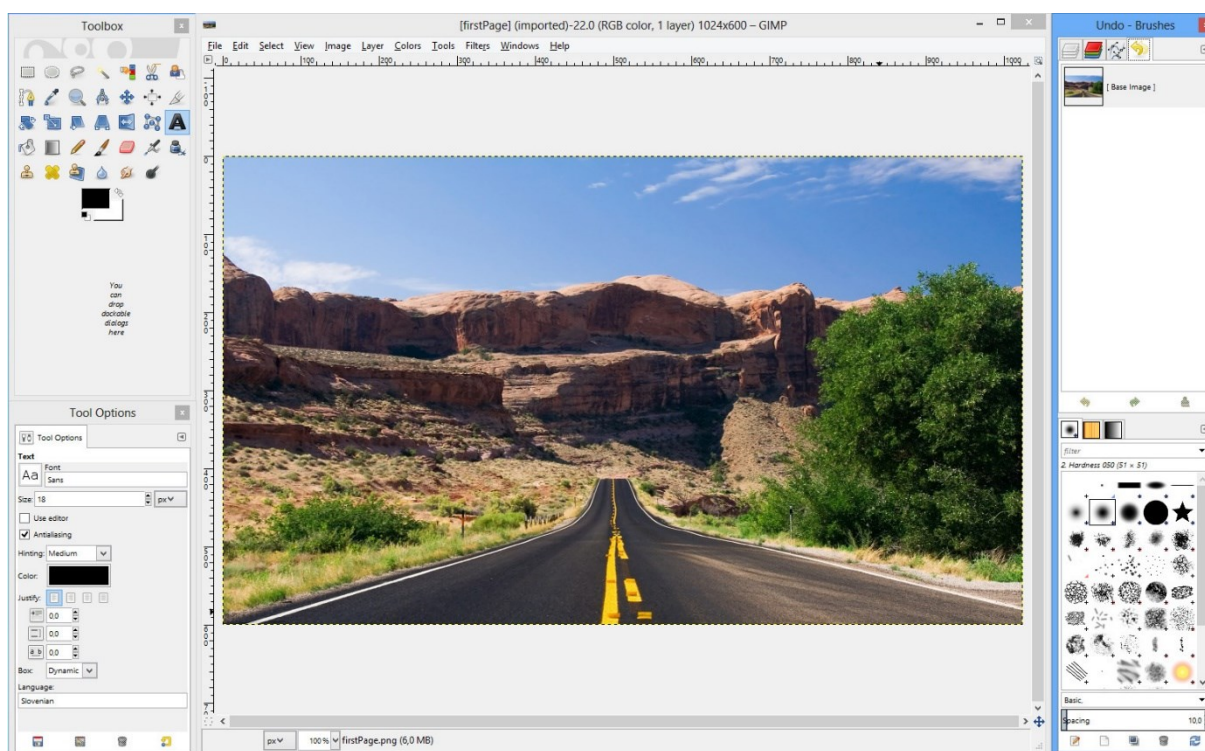
Slika 2.4: Razvojno okolje NetBeans (posnetek strani programa NetBeans IDE 8.0)

Razvojno okolje NetBeans je razdeljeno na 4 dele, kar je razvidno iz zgornje slike. V prvem delu imamo pregled nad datotekami in ustvarjenimi projekti, ki sestavljajo aplikacije. V drugem

delu imamo zapisane elemente oziroma funkcije trenutno odprte datoteke, kar nam omogoča lažjo navigacijo znotraj projekta. Tretji, osrednji del, vsebuje programsko kodo. V tem delu pride do izraza razvojno okolje IDE, ki nam ob vpisu prve črke predlaga nadaljevanje z opisom posamezne funkcije oziroma razreda, opozarja nas na sintetične napake in nam različno obarva besedilo za lažji pregled nad programsko kodo. [21]

2.3.2 GIMP

GIMP je brezplačni program za urejanje rastrske grafike. Prva uradna verzija GIMP 0.54 je izšla januarja 1996. Za njegov razvoj si delita zasluge razvijalca Spence Kimball in Peter Mattis. Program je enostaven za uporabo, omogoča manipulacijo s slikami, kot so ostritev, nastavljanje kontrasta in svetlobe, prav tako program omogoča tudi večplastno urejanje grafike, kar je bil glavni namen uporabe programa GIMP. [22]



Slika 2.5: Razvojno okolje GIMP (posnetek strani programa GIMP 2.8.10)

Program je v grobem razdeljen na 4 dele, kar je razvidno tudi iz zgornje slike. V levem zgornjem delu imamo možnost izbire zelenega orodja. V levem spodnjem kotu imamo prikazane možnosti, ki jih lahko urejamo pri izbranem orodju. V sredini je osrednji del, v katerem je prikazana slika, nad katero lahko izvajamo manipulacije, prav tako imamo možnost odprtja več

slik hkrati. V desnem delu imamo prikazane plasti izbrane slike, kar nam omogoča manipulacijo nad izbrano plastjo. [23]

Poglavje 3 Igra Time Attacker

V tem poglavju predstavljamo opis igre, programsko kodo, implementacijo izdelka in testiranje aplikacije na tabličnem računalniku BlackBerry PlayBook. Slike, vključene v to poglavje, so pridobljene kot posnetek naše aplikacije (*print screen*), vendar si avtorstva grafične podobe ne pripisujemo. Enako velja za zvočno ozadje.

3.1 Opis

Igra Time Attacker je dirkaška 3D igra, ki smo jo razvili za uporabo na tabličnem računalniku BlackBerry PlayBook.

Vožnja avtomobila poteka zgolj z nagibanjem naprave (levo, desno, naprej, nazaj), brez motečega dotikanja zaslona s prsti med samo vožnjo. Avtomobil avtomatsko pospešuje, razen v kolikor tablični računalnik nagnemo močno naprej, takrat pričnemo z zaviranjem. Zavijanje poteka z nagibanjem naprave levo ali desno. Ob večjem nagibu tabličnega računalnika tudi avto bolj močno zavije v smer gibanja.



Slika 3.1: Začetni meni igre Time Attacker (posnetek aplikacije)

Na zgornji sliki je predstavljena začetna stran, kjer se nahaja gumb z imenom pomoč (*Help*), v katerem prejmemo vsa potrebna navodila za igranje igre. Ob pritisku na gumb igray (*Play*) pa pričnemo z igranjem igre.



Slika 3.2: Dirka v igri Time Attacker (posnetek aplikacije)

Na cesti, kot prikazuje zgornja slika, so tudi ostali udeleženci prometa, na katere je potrebno biti pozoren. Dirka se lahko ob prekomernih trkih z ostalimi udeleženci prometa predčasno zaključi. Prav tako je potrebno biti pazljiv na izlete s steze. Ti izleti sicer niso usodni, vendar lahko voznika močno ovirajo pri doseganju najhitrejšega časa, saj je ob cesti veliko ovir, ki zadržujejo voznika.



Slika 3.3: Zmaga v igri Time Attacker (posnetek aplikacije)

Cilj igre je doseči najhitrejši čas na približno 4-kilometrski progi sredi puščave, in zmagati. V kolikor vozniku uspe zmagati, se mu prikaže zgornja slika, ki prikaže tudi čas vožnje. V primeru predčasnega končanja z dirko oziroma kadar doseženi čas ni rekorden, se pojavi okno za konec igre z možnostjo ponovnega poizkusa.

3.2 Programska koda

Programska koda za predstavljeno igro [24] je ustvarjena za uporabo s tipkovnico in obsega približno 1.000 vrstic kode.

Pri ustvarjanju naše aplikacije Time Attacker je bilo potrebno omenjeno kodo prilagoditi za uporabo na BlackBerry PlayBook, za uporabo brez tipkovnice, ter poiskati nekatere konceptualne rešitve, ki pritegnejo uporabnika.

V nadaljevanju jih obravnavamo po vsebinskih sklopih:

- **Ozadje**

Ozadje spreminjamo tako, da odpremo obstoječo background.png sliko ali ustvarimo novo sliko, ki jo lahko po želji oblikujemo, npr.: hribi, nebo, oblaki, ptice, sonce itd.

Vse to lahko vključimo v programsko kodo, ter v programski kodi izberemo pozicijo slike in določimo njeno širino in višino (glej spodnji sliki).

```

loadImages: function(names, callback) { // load multiple images and callback when ALL images have loaded
  var result = [];
  var count = names.length;

  var onload = function() {
    if (--count == 0)
      callback(result);
  };

  for (var n = 0; n < names.length; n++) {
    var name = names[n];
    result[n] = document.createElement('img');
    Dom.on(result[n], 'load', onload);
    result[n].src = "images/" + name + ".png";
  }
},

```

Slika 3.4: Izsek iz programske kode – izbira poti do slike

```

Game.run({
  canvas: canvas, render: render, update: update, step: step,
  images: ["background", "sprites"],
  ready: function(images) {
    background = images[0];
    sprites = images[1];
    reset();
    Dom.storage.fast_lap_time = Dom.storage.fast_lap_time || 180;
    updateHud('fast_lap_time', formatTime(Util.toFloat(Dom.storage.fast_lap_time)));
  }
});

```

Slika 3.5: Izsek iz programske kode – izbira imena slike

- **Določanje širine ceste, število pasov, število krogov, število avtomobilov**

Želenim spremenljivkam (glej sliko) smo določili naslednje vrednosti: širina ceste: 2000, število pasov: 3, število krogov: 1, število avtomobilov: 120.

```

var roadWidth = 2000; // actually half the roads width, easier math if the road spans from -roadWidth to +roadWidth
var lanes = 3; // number of lanes
var maxLap = 1; // number of max laps
var totalCars = 120; // total number of cars on the road

```

Slika 3.6: Izsek iz programske kode – širina, pasovi, št. krogov, št. avtomobilov

- **Sprememba proge**

Za popestritev voznikove izkušnje lahko spreminjamo tudi progo, npr.: ravnica, hrib, zavoj ipd. Našo izbiro prikazuje spodnja slika.

```
function resetRoad() {
  segments = [];

  addStraight(ROAD.LENGTH.SHORT);
  addLowRollingHills();
  addSCurves();
  addCurve(ROAD.LENGTH.MEDIUM, ROAD.CURVE.MEDIUM, ROAD.HILL.LOW);
  addBumps();
  addLowRollingHills();
  addCurve(ROAD.LENGTH.LONG * 2, ROAD.CURVE.MEDIUM, ROAD.HILL.MEDIUM);
  addStraight();
  addHill(ROAD.LENGTH.MEDIUM, ROAD.HILL.HIGH);
  addSCurves();
  addCurve(ROAD.LENGTH.LONG, -ROAD.CURVE.MEDIUM, ROAD.HILL.NONE);
  addHill(ROAD.LENGTH.LONG, ROAD.HILL.HIGH);
  addCurve(ROAD.LENGTH.LONG, ROAD.CURVE.MEDIUM, -ROAD.HILL.LOW);
  addBumps();
  addHill(ROAD.LENGTH.LONG, -ROAD.HILL.MEDIUM);
  addStraight();
  addSCurves();
  addDownhillToEnd();

  resetSprites();
  resetCars();

  segments[findSegment(playerZ).index + 2].color = COLORS.START;
  segments[findSegment(playerZ).index + 3].color = COLORS.START;
  for (var n = 0; n < rumbleLength; n++)
    segments[segments.length - 1 - n].color = COLORS.FINISH;

  trackLength = segments.length * segmentLength;
}
```

Slika 3.7: Izsek iz programske kode – atributi proge

- **Sprememba barv na cestišču in izven njega**

Za spremembo barv moramo spremeniti vrednosti v datoteki *common.js*

```
//=====
// RACING GAME CONSTANTS
//=====

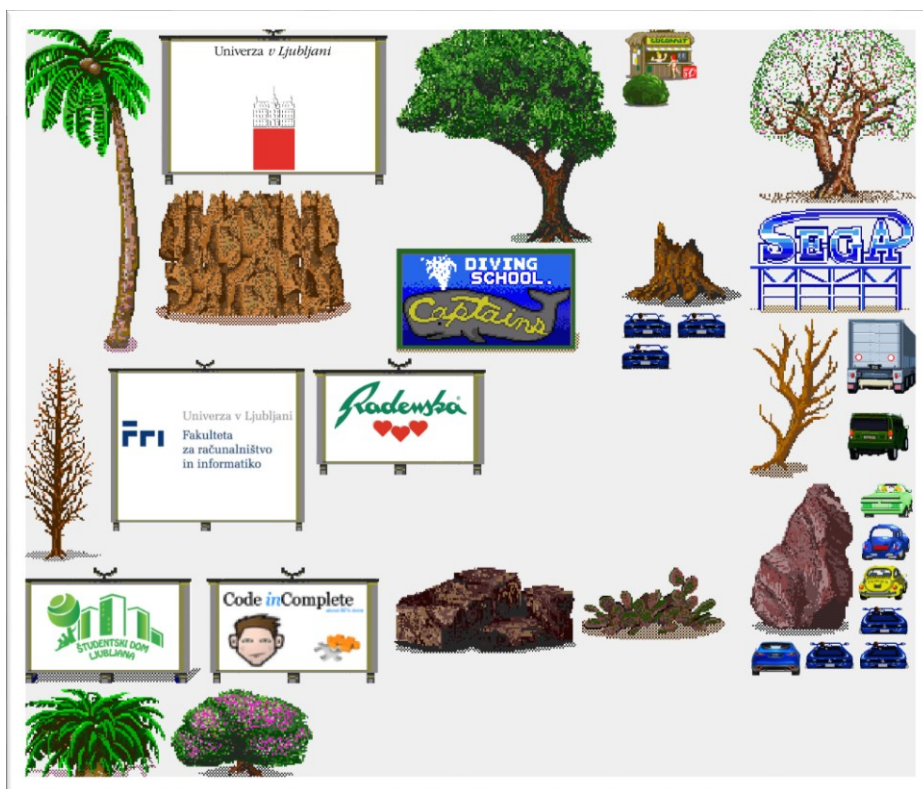
var COLORS = {
  SKY: '#72D7EE',
  TREE: '#005108',
  FOG: '#005108',
  // sand
  LIGHT: {road: '#6B6B6B', grass: '#c9ae83', rumble: '#555555', lane: '#CCCCCC'},
  DARK: {road: '#696969', grass: '#c3a87d', rumble: '#BBBBBB'},
  //grass
  // LIGHT: {road: '#6B6B6B', grass: '#10AA10', rumble: '#555555', lane: '#CCCCCC'},
  // DARK: {road: '#696969', grass: '#009A00', rumble: '#BBBBBB'},
  START: {road: 'white', grass: 'white', rumble: 'white'},
  FINISH: {road: 'black', grass: 'black', rumble: 'black'}
};
```

Slika 3.8: Izsek iz programske kode – sprememba barv

Zgornja slika prikazuje programsko kodo, kjer lahko spremenimo barve. Pod *grass* lahko spremenimo barvo, s katero določimo, ali bo zunaj cestišča trava, pesek ali voda. Pod *road* spremenimo barvo cestišča, pod *rumble* določimo barvo robnikov, barvo sredinske črte določimo pri *lane*. Tukaj lahko določimo tudi barve za meglo *fog*, drevesa *tree* in nebo *sky*.

- **Sprememba barv avtomobilov in objektov, ki se nahajajo ob cesti**

Če želimo spremeniti barve avtomobilov in objektov, ki se nahajajo ob cesti, to storimo tako, da odpremo obstoječo *sprites.png* sliko, določimo novo barvo, in jo vključimo v programsko kodo, ter v programski kodi izberemo pozicijo slike in določimo njeno širino in višino (glej spodnji sliko).

Slika 3.9: Primer slike *sprites.png*

```
var SPRITES = {
  PALM_TREE: {x: 5, y: 5, w: 215, h: 540},
  BILLBOARD08: {x: 230, y: 5, w: 385, h: 265},
  TREE1: {x: 625, y: 5, w: 360, h: 360},
  DEAD_TREE1: {x: 5, y: 555, w: 135, h: 332},
  BILLBOARD09: {x: 150, y: 555, w: 328, h: 282},
  BOULDER3: {x: 230, y: 280, w: 320, h: 220},
```

Slika 3.10: Izsek iz programske kode – izbira pozicij slike

○ Upravljanje z avtomobilom

Za upravljanje avtomobila zgolj z nagibom naprave potrebujemo poslušalec dogodkov (*Device Motion Event*). Poslušalec dogodkov ob vsakem premiku vrača parametre, ki jih pridobi iz pospeškometra. Koda je prikazana na spodnji sliki.

```
/*=====
// EVENT LISTENER FOR DEVICE MOTION
//=====

function detectMotion() {
  // Blackberry Playbook does not support 'deviceorientation'
  if (window.DeviceMotionEvent) {

    window.addEventListener('devicemotion', function(eventData) {

      // Grab x parameter from results of acceleration including gravity. Is for tilt device from left to right
      var LeftRight = eventData.accelerationIncludingGravity.x;

      // Grab y parameter from results of acceleration including gravity. Is for tilt device from front to back
      var FrontBack = -eventData.accelerationIncludingGravity.y;

      // call our motion event handler
      deviceMotionHandler(LeftRight, FrontBack);
    }, false);

  } else {
    document.getElementById("canvas").innerHTML = "Not supported on your device or browser. Sorry. _";
  }
}
```

Slika 3.11: Izsek iz programske kode - poslušalec dogodkov

Iz programske kode JS je razvidna funkcija *detectMotion*, znotraj katere je postavljen pogoj, v katerem preverjamo, če naprava podpira poslušalce dogodkov (*devicemotion*). V kolikor ga podpira, pridobimo podatke o nagibu naprave, vrednosti se gibljejo od -9.81 do 9.81, in kličemo funkcijo *deviceMotionHandler*. Omenjena funkcija je prikazana na spodnji sliki.


```
//  
// UPDATE THE GAME WORLD  
//  
function deviceMotionHandler(LeftRight, FrontBack) {  
  
  //  
  if (LeftRight > 6 || LeftRight < -6) {  
    centrifugal = 0.2;  
  }  
  else {  
    centrifugal = 0.3;  
  }  
  
  if (FrontBack >= 0) {  
    if (FrontBack < 3) {  
      keyFaster = false;  
      keySlower = true;  
    } else {  
      keySlower = false;  
      keyFaster = true;  
    }  
  
    if (LeftRight > 1) {  
      keyLeft = false;  
      keyRight = true;  
    }  
  
    else if (LeftRight < -1) {  
      keyRight = false;  
      keyLeft = true;  
    }  
    else {  
      keyLeft = false;  
      keyRight = false;  
    }  
  }  
  else {  
    if (FrontBack > -3) {  
      keyFaster = false;  
      keySlower = true;  
    } else {  
      keyFaster = true;  
      keySlower = false;  
    }  
  }  
}
```

Slika 3.12: Izsek iz programske kode – del funkcije *deviceMotionHandler*

Iz slike je razvidna funkcija, znotraj katere postavljamo pogoje. V kolikor nagib naprave levo oziroma desno presega vrednost 6 oziroma je manjši od -6, se centrifugalna spremenljivka zniža iz 0.3 na 0.2, kar nam omogoča boljši oprijem na cestišču. V kolikor nam omenjeni oprijem ne ustreza, ga lahko spreminjamo s spremenljivko *centrifugal*. Višja kot je vrednost, slabši oprijem ima avtomobil. Znotraj funkcije prav tako tudi določamo spremenljivkam za levo, desno, hitreje in počasneje vrednosti *true*, v kolikor je aktivna, oziroma *false*, v kolikor ni aktivna. [25]

- **Zvok**

Če želimo spremeniti zvočno ozadje, to storimo tako, da v programski kodi spremenimo pot do nove datoteke, in ime (glej spodnjo sliko).

```

<div id="musicP">
  <audio id='music' autoplay loop>
    <source src="music/racer.ogg">
    <source src="music/racer.mp3">
  </audio>
  <span id="mute"></span>
</div>

```

Slika 3.13: Izsek iz programske kode – sprememba zvočnega ozadja

○ Kreiranje datoteke *config.xml*

Pred testiranjem aplikacije ter objavo na portalu BBW je potrebno kreirati datoteko *config.xml*, ki vsebuje podatke o aplikaciji: ime aplikacije, verzija aplikacije, avtor, vsebina, opis aplikacije, prikazana ikona, orientacija ipd. (glej spodnjo sliko). [26]

```

<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.w3.org/ns/widgets"
  xmlns:rim="http://www.blackberry.com/ns/widgets"
  version="1.0.0.0" id="TimeAttacker">

  <name>Time Attacker</name>
  <author>BlazKokol</author>
  <content src="index.html"/>
  <description>Show your skills and be the fastest racer!</description>
  <icon src="icon.png"/>
  <rim:orientation mode="landscape" />
  <autoOrients>false</autoOrients>
</widget>

```

Slika 3.14: Izsek iz programske kode – datoteka *config.xml*

○ Nekatere druge možnosti

Pri razvoju aplikacije imamo znotraj programske kode številne aplikativne možnosti v smislu spreminjanja in različnih nastavitev. V nadaljevanju nanizamo nekaj primerov: dolžina segmenta, gostota megle, hitrost pospeševanja, hitrost zaviranja, maksimalna hitrost, širina in višina risalnega platna (*canvas*), število narisanih okvirjev na sekundo (*fps*), število segmentov (dolžina proge), višina kamere itd.

3.3 Implementacija izdelka

3.3.1 Testiranje z emulatorjem

Pred samo implementacijo izdelka na tablični računalnik BlackBerry PlayBook smo igro testirali z emulatorjem Ripple Emulator, ki deluje kot dodatek v spletnem brskalniku Google Chrome. S pomočjo Ripple Emulatorja lahko na enostaven in hiter način preverimo, kako bo izgledala in delovala aplikacija na tabličnem računalniku. Prav tako lahko dodatek Ripple Emulator uporabljamo tudi za testiranje aplikacij na ostalih mobilnih napravah, kot so: BlackBerry Q10, iPhone 4, iPhone 5, Nexus 7, itd. Izgled testnega okolja Ripple Emulator prikazuje spodnja slika.



Slika 3.15: Ripple Emulator (posnetek strani programa Google Chrome)

V levem spodnjem kotu imamo pospeškometer, ki je bil za naše testiranje ključnega pomena. Z nagibom slike, ki je prikazana v pospeškometru, simuliramo nagib tabličnega računalnika. Z nagibom tabličnega računalnika smo testirali zaviranje in vožnjo avtomobila v igri. [27]

3.3.2 Testiranje na tabličnem računalniku BlackBerry PlayBook

- **Registracija tabličnega računalnika BlackBerry PlayBook**

Na uradni spletni strani podjetja BlackBerry se moramo najprej registrirati in pridobiti BlackBerry identifikator (*ID*). Uporabniško ime in geslo, ki si ga izberemo, bomo uporabili na samem tabličnem računalniku za dostop do portala *BlackBerry World*, kjer si lahko namestimo razne aplikacije, in tudi za dostop do portala *Vendor Portal for BlackBerry World*, preko katerega lahko objavimo našo aplikacij.

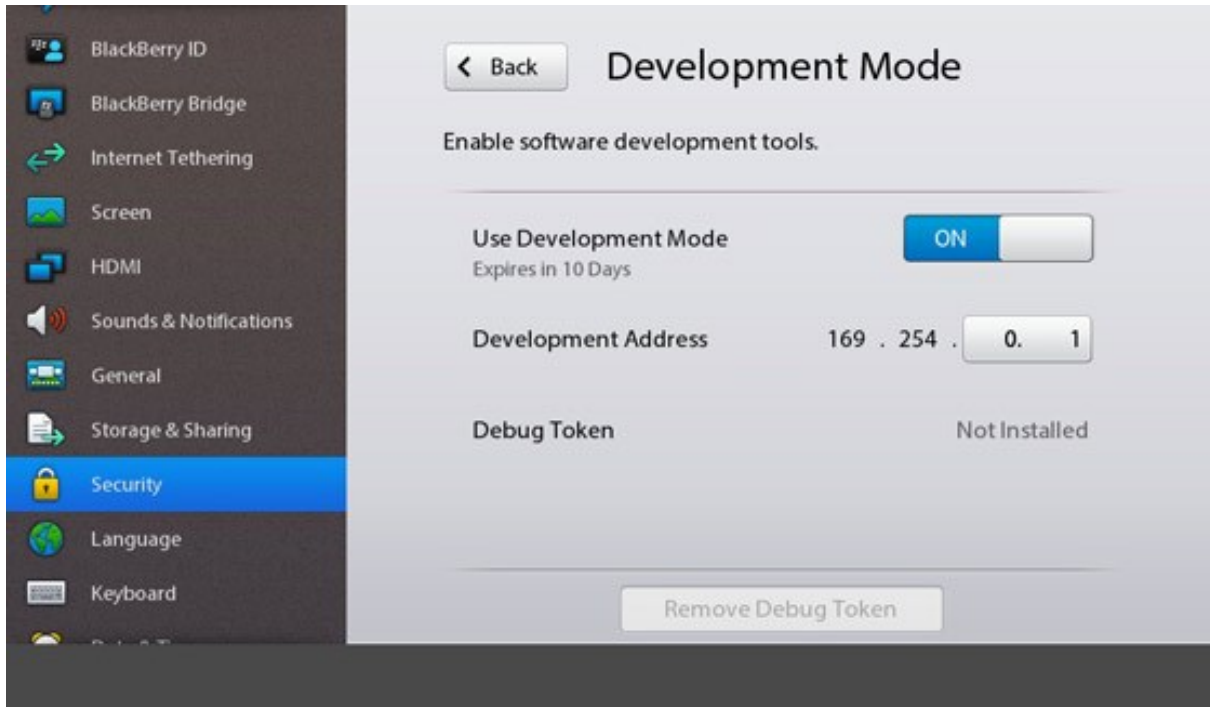
- **Priprava aplikacije za implementacijo**

Za implementacijo aplikacije na tablični računalnik moramo ustvariti stisnjeno datoteko s končnico *.zip*, ki mora vsebovati vse datoteke, ki jih vsebuje tudi aplikacija.

- **Priprava naprav za implementacijo**

Ko imamo stisnjeno datoteko, pripravimo računalnik (iz katerega implementiramo) in tablični računalnik (na katerega implementiramo), na nadaljnji postopek implementacije.

Na tabličnem računalniku BlackBerry PlayBook moramo vklopiti t.i. razvijalski način delovanja.



Slika 3.16: Prikaz vklopa razvijalskega načina [28]

Na zgornji sliki je prikazano, kje v meniju na tabličnem računalniku BlackBerry PlayBook se vklopi razvijalski način. Na osnovni strani BlackBerry PlayBook imamo v desnem zgornjem kotu ikono za nastavitve. Ko se nam odprejo nastavitve, moramo na levi strani izbrati možnost varnost (*Security*), nato izberemo možnost razvijalski način. Ko imamo odprto okno razvijalskega načina, samo preklopimo gumb na ON, le-ta se nahaja na desni strani besedila .

- **Pridobitev podpisanih ključev**

Na računalniku, na katerem imamo naloženo aplikacijo, potrebujemo BlackBerry podpisane ključe (*signing keys*). To sta 2 ključa, en ključ nam služi za podpis razhroščevalnega žetona (*debug token*), drugi ključ pa nam služi za podpis aplikacije, in ga potrebujemo pri podpisu aplikacije, preden jo objavimo na portalu *BlackBerry World*. Ključne pridobimo na spletni strani podjetja BlackBerry [29], prejmemo jih na vpisani elektronski naslov in jih nato shranimo na računalnik.

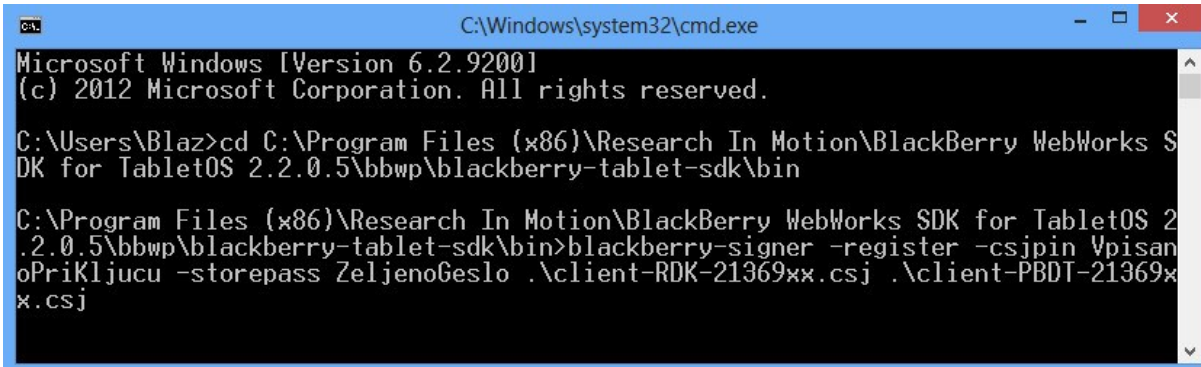
- **Registracija podpisanih ključev**

V naslednjem koraku moramo pridobljena ključa registrirati. Registracijo, kot tudi nadaljnjo implementacijo, lahko na enostaven način izvajamo s pomočjo programa Command Prompt (v nadaljevanju cmd). V cmd se postavimo v direktorij:

```
C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS  
<version>\bbwp\blackberry-tablet-sdk\bin.
```

V tem direktoriju vpišemo ukaz in ga s pritiskom na tipko enter izvedemo:

```
blackberry-signer -register -csjpin <csj pin> -storepass <KeystorePassword> <client-RDK-  
xxxxxx.csj file> <client-PBDT-xxxxxx.csj file>.
```



```
Microsoft Windows [Version 6.2.9200]  
(c) 2012 Microsoft Corporation. All rights reserved.  
  
C:\Users\Blaz>cd C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks S  
DK for TabletOS 2.2.0.5\bbwp\blackberry-tablet-sdk\bin  
  
C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2  
.2.0.5\bbwp\blackberry-tablet-sdk\bin>blackberry-signer -register -csjpin Vpisan  
oPrikljucu -storepass ZeljenoGeslo .\client-RDK-21369xx.csj .\client-PBDT-21369x  
x.csj
```

Slika 3.17: Primer registracije ključev (posnetek programa cmd)

Na zgornji sliki je prikazan primer, kako se premaknemo v omenjen direktorij. Prav tako je prikazana tudi koda, s katero opravimo registracijo. Pri *-csjpin* moramo vpisati geslo, ki si ga

določimo na portalu, kjer se registriramo za pridobitev ključev. Pri `-storepass` pa vpišemo novo geslo, katerega bomo pri naših ključih uporabljali. Pri `client-RDK-xxxxxx.csj file` določimo pot, kjer se nahaja naš RDK ključ, ki ga uporabljamo za podpis aplikacij in objavo na portalu *BlackBerry World*. Pri `client-PBDT-xxxxxx.csj file` pa določimo pot, kjer se nahaja naš PBDT ključ, ki ga uporabljamo za generiranje razhroščevalnega žetona, ki nam služi za testiranje aplikacije na našem tabličnem računalniku.

Po opravljeni registraciji se kreirajo 3 datoteke (`author.p12`, `barsigner.csk` in `barsigner.db`) v skrito mapo z naslovom poti:

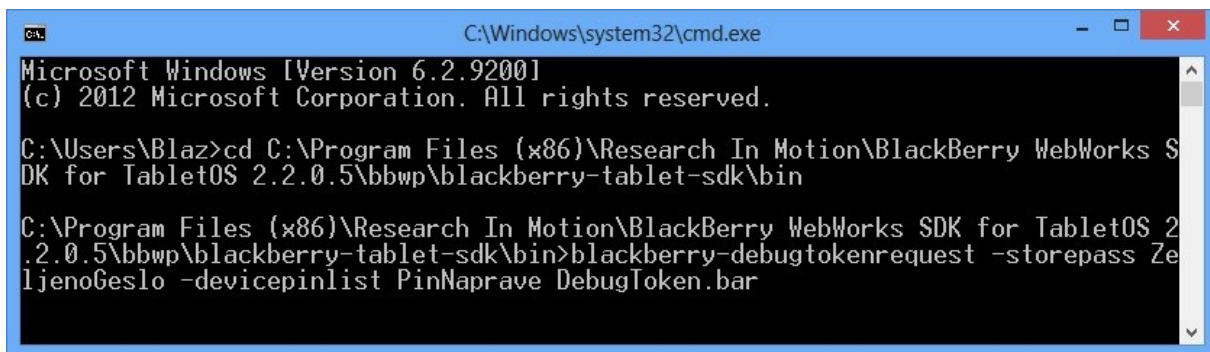
`%HOMEPATH%\AppData\Local\Research In Motion`, za katere je priporočljivo, da se naredi varnostna kopija. [30]

V izogib morebitnim težavam priporočamo, da se datoteka `author.p12` kopira v direktorij `C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS <version>\bbwp\bin` in preimenuje v `sigtool.p12` [31]

o Kreiranje razhroščevalnega žetona

V naslednjem koraku bomo kreirali razhroščevalni žeton (`debug token`). Na spodnji sliki je primer, kako ga kreiramo. Najprej se je spet potrebno premakniti v ustrezni direktorij. Nato izvedemo kodo:

```
blackberry-debugtokenrequest -storepass <KeystorePassword> -devicepinlist
<device_PIN_list_file> <debug_token_file_name.bar>.
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The text in the window is as follows:

```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Blaz>cd C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks S
DK for TabletOS 2.2.0.5\bbwp\blackberry-tablet-sdk\bin

C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2
.2.0.5\bbwp\blackberry-tablet-sdk\bin>blackberry-debugtokenrequest -storepass Ze
ljenoGeslo -devicepinlist PinNaprave DebugToken.bar
```

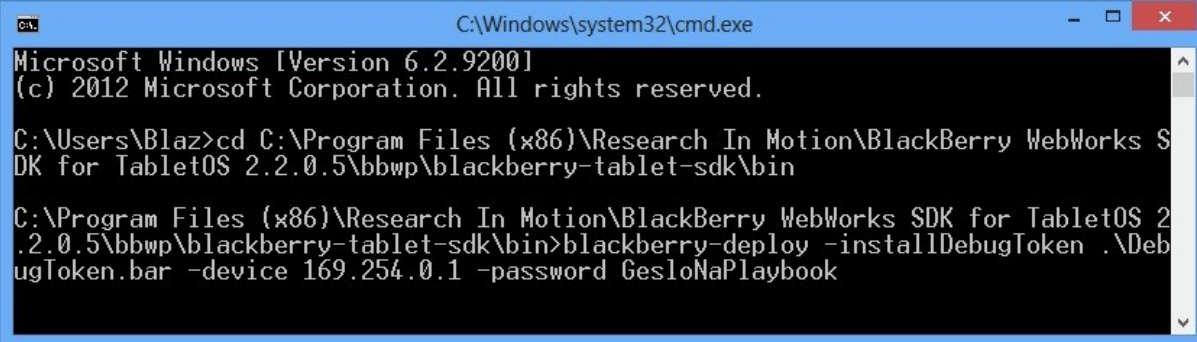
Slika 3.18: Primer kreiranja razhroščevalnega žetona (posnetek programa cmd)

Pri `-storepass` vpišemo geslo, ki smo ga kreirali v prejšnjem odseku. Pri `-devicepinlist` vpišemo PIN številko tabličnega računalnika, na katerega želimo namestiti aplikacijo. PIN številko najdemo v nastavitvah. V meniju izberemo gumb o napravi in nato še izberemo strojna oprema. Prva lastnost, ki se nam prikaže, je PIN naprave, katerega vpišemo. Pri `-debug_token_file_name.bar` pa vpišemo naslov direktorija, kamor želimo shraniti

razhroščevalni žeton, in ime, ki ga mu želimo dodeliti. V našem primeru bomo shranili razhroščevalni žeton z imenom *DebugToken.bar* v direktorij, v katerem se nahajamo.

- **Namestitev razhroščevalnega žetona na BlackBerry PlayBook**

Kreiran razhroščevalni žeton je potrebno namestiti na tablični računalnik BlackBerry PlayBook. Za vpis ukaza se moramo nahajati v enakem direktoriju kot pri kreiranju razhroščevalnega žetona. Primer, kako se namesti na napravo, prikazuje spodnja slika.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Blaz>cd C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2.2.0.5\bbwp\blackberry-tablet-sdk\bin

C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2.2.0.5\bbwp\blackberry-tablet-sdk\bin>blackberry-deploy -installDebugToken .\DebugToken.bar -device 169.254.0.1 -password GesloNaPlaybook
```

Slika 3.19: Primer nameščanja razhroščevalnega žetona (posnetek programa cmd)

Pri *-installDebugToken* moramo vpisati pot do razhroščevalnega žetona in njegovo ime. V našem primeru se nahaja v isti mapi. Pri *-device* moramo vpisati IP naslov naprave oziroma t.i. razvijalski naslov naprave, ki ga najdemo na tabličnem računalniku med nastavitvami, pot do njega je v opisana v poglavju Priprava naprav za implementacijo na strani 22–23. Pri *-password* pa vpišemo geslo, ki ga uporabljamo na tabličnem računalniku BlackBerry PlayBook. Razhroščevalni žeton je na tabličnem računalniku veljaven 30 dni, po preteku tega obdobja ga je spet potrebno na novo namestiti oziroma po potrebi kreirati novega.

- **Priprava datotek na razhroščevalni žeton**

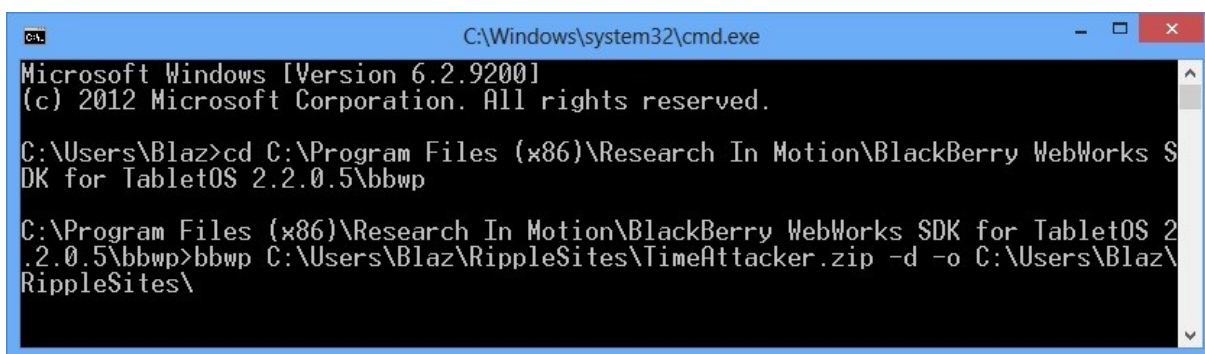
Datoteki *bbwp.properties*, ki se nahaja v direktoriju:

C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2.2.0.5\bbwp\bin«, moramo spremeniti naslov do razhroščevalnega žetona. Datoteko lahko odpremo v navadni beležnici. V kolikor tega ne spremenimo in se pot in ime razhroščevalnega žetona ne ujema z vpisanim, implementacija aplikacije brez podpisa na tablični računalnik ne bo mogoča. V datoteki je potrebno spremeniti naslednjo vrstico:

<debug_token>pot do razhroščevalnega žetona</debug_token>.

- **Priprava aplikacije za implementiranje**

Po uspešno kreiranem razhroščevalnem žetonu, nameščenem na tabličnem računalniku in vpisani poti do razhroščevalnega žetona v datoteki *bbwp.properties*, ki je opisan v zgornjem odstavku, lahko nadaljujemo s pripravo aplikacije za nadaljnjo implementacijo. Stisnjeno datoteko s končnico *.zip* (*TimeAttacker.zip*) bomo v tem delu pripravili tako, da jo bomo lahko namestili na tablični računalnik. S pomočjo kode, ki je predstavljena na spodnji sliki, bomo kreirali novo datoteko, v našem primeru *TimeAttacker* s končnico *.bar*, ki je nepodpisana aplikacija.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Blaz>cd C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2.2.0.5\bbwp

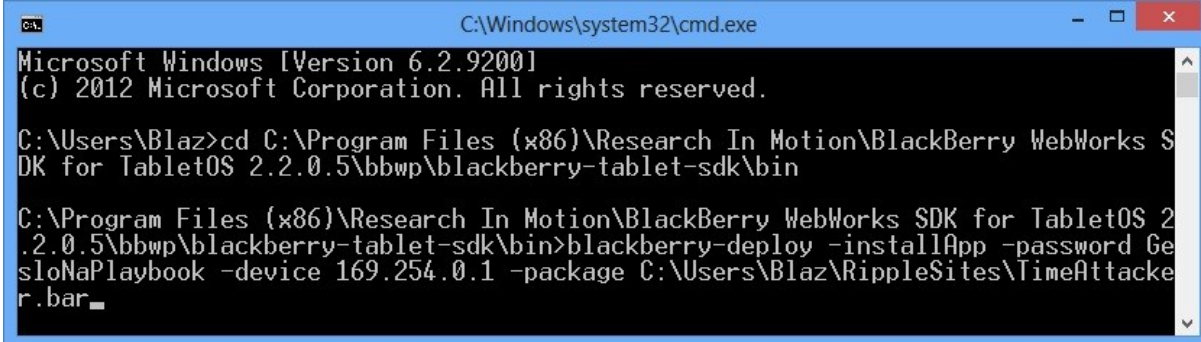
C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2.2.0.5\bbwp>bbwp C:\Users\Blaz\RippleSites\TimeAttacker.zip -d -o C:\Users\Blaz\RippleSites\
```

Slika 3.20: Primer priprave aplikacije za nadaljnjo implementacijo (posnetek programa cmd)

Iz zgornje slike je razvidno, da se moramo premakniti v drug direktorij, v katerem bomo izvedli ukaz. Ukaz vsebuje direktorij, ime naše stisnjene datoteke, parameter *-d*, ki pripravi našo aplikacijo v razvijalskem načinu in jo lahko s pomočjo predhodno nameščenega razhroščevalnega žetona namestimo na tablični računalnik, ter tudi parameter *-o*, ki vključuje direktorij, v katerega želimo novo nastalo datoteko shraniti.

- **Implementacija aplikacije na BlackBerry PlayBook**

Zadnji korak pred testiranjem aplikacije na tabličnem računalniku je implementacija datoteke *TimeAttacker.bar*, ki smo jo predhodno kreirali (glej prejšnji odstavek).



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Blaz>cd C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2.2.0.5\bbwp\blackberry-tablet-sdk\bin

C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2.2.0.5\bbwp\blackberry-tablet-sdk\bin>blackberry-deploy -installApp -password GesloNaPlaybook -device 169.254.0.1 -package C:\Users\Blaz\RippleSites\TimeAttacker.bar_
```

Slika 3.21: Primer implementacije aplikacije na BlackBerry PlayBook (posnetek programa cmd)

Zgornja slika prikazuje primer, kako se izvede implementacija aplikacije na tablični računalnik. Spet je potrebno menjati direktorij, v katerem bomo izvajali ukaze. Poglejmo sestavo ukaza: *blackberry-deploy -installApp -password <device password> -device <IP_address> -package <BAR file path>*.

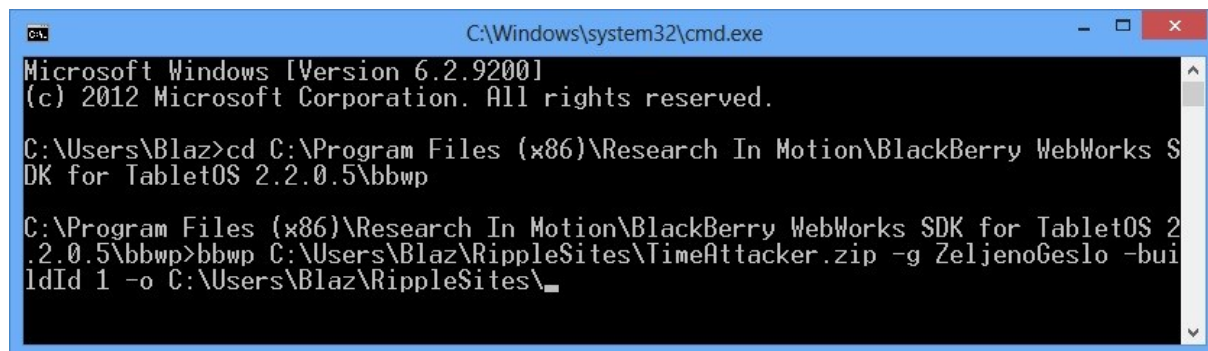
Pri *-installApp* vpišemo geslo, ki ga uporabljamo na BlackBerry PlayBook. Pri *-device* vpišemo IP naslov BlackBerry PlayBook. Pri *-package* pa vpišemo direktorij, v katerem se naša aplikacija nahaja in njeno ime. [32]

3.4 Objava igre

- **Podpisovanje in priprava aplikacije za objavo na spletu**

Za odobritev distribuiranja aplikacije preko spletnega portala *BlackBerry World* moramo aplikacijo podpisati, da dokažemo avtorstvo ostalim uporabnikom in operacijskim sistemom. [33] *BlackBerry World* je primarni distribucijski kanal za zaupanja vredne in preverjene BlackBerry aplikacije. [34]

V nadaljevanju bo predstavljena možnost podpisa aplikacije s pomočjo programa cmd, enako kot smo jo implementirali na tablični računalnik. Med tem postopkom prav tako kreiramo novo datoteko s končnico *.bar*, z razliko, da je ta datoteka podpisana z našim BlackBerry ključem.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Blaz>cd C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2.2.0.5\bbwp

C:\Program Files (x86)\Research In Motion\BlackBerry WebWorks SDK for TabletOS 2.2.0.5\bbwp>bbwp C:\Users\Blaz\RippleSites\TimeAttacker.zip -g ZeljenoGeslo -buildId 1 -o C:\Users\Blaz\RippleSites\
```

Slika 3.22: Primer podpisa aplikacije za nadaljnjo obdelavo (Posnetek programa cmd)

Na zgornji sliki je primer, kako se premaknemo v direktorij, kjer bomo izvedli ukaz in zapis samega ukaza. Sestava ukaza:

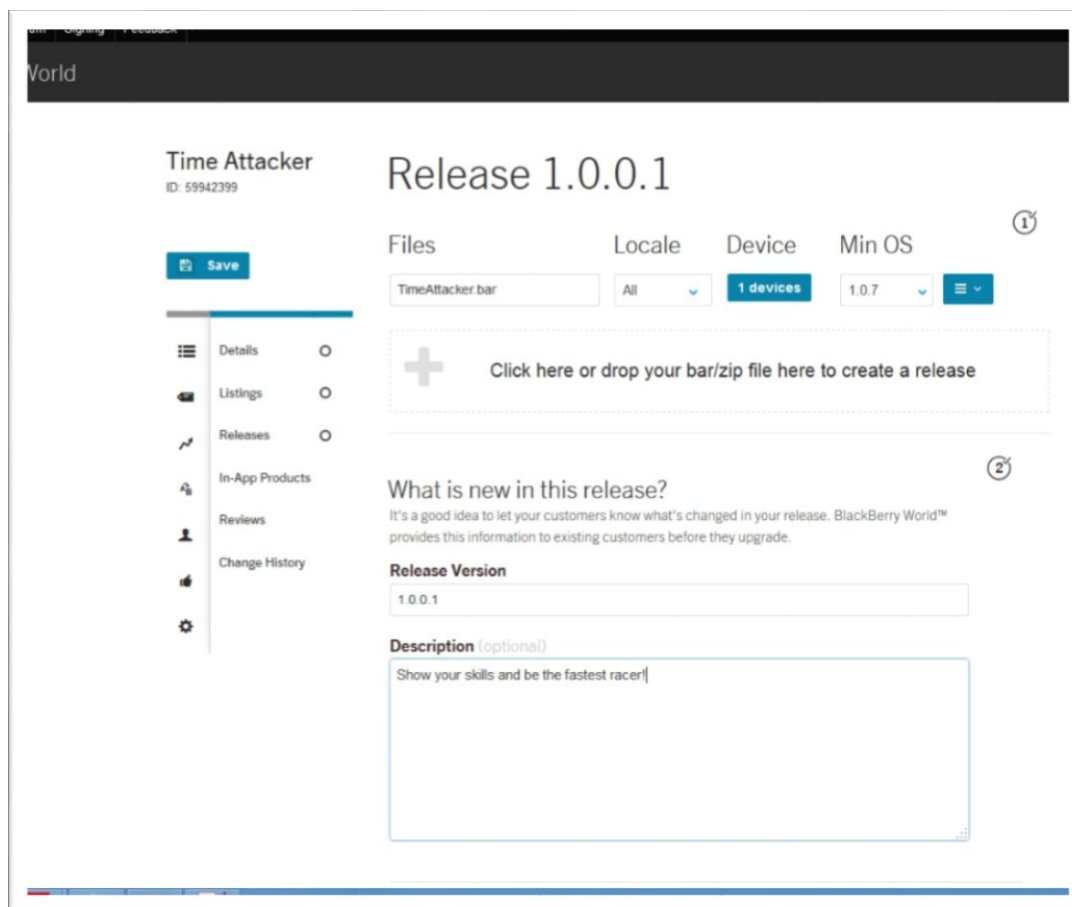
```
bbwp C:\myapp\myarchive.zip -g myKeystorePassword -buildId 10 -o C:\myapp\output.
```

Pri *-g* vpišemo geslo, ki smo ga izbrali pri registraciji BlackBerry ključev. Pri *-buildId* vpišemo verzijo aplikacije, v našem primeru je prva podpisana verzija. Pri *-o* vpišemo direktorij, v katerem želimo, da se kreira podpisana aplikacija s končnico *.bar*.

Po uspešno izvedenem ukazu dobimo novo podpisano datoteko *TimeAttacker.bar*, ki jo lahko uporabimo na našem tabličnem računalniku, in jo tudi objavimo na spletnem portalu *BlackBerry World*. [35]

- **Spletni portal *Vendor Portal for BlackBerry World***

Portal *Vendor Portal for BlackBerry World* nam služi za pregled naših podpisanih aplikacij in objavo le-teh na portalu *BlackBerry World*. Pri prvem dostopu do portala *Vendor Portal for BlackBerry World* [36] je potrebno opraviti registracijo. Registracija in članstvo sta brezplačna. Po prijavi na portalu je še potrebna preverba avtentičnosti. In sicer tako, da prejmemo elektronsko sporočilo, v katerem so zapisana dodatna navodila. Fizična oseba mora poslati po elektronski pošti ali faksu kopijo osebne dokumenta, iz katerega sta razvidna ime in letnica rojstva skupaj s spremnim besedilom. Šele, ko dobimo potrditev, smo uspešno zaključili z registracijo na portal. Za prijavo pa uporabljamo naše BlackBerry ID uporabniško ime in geslo.



Slika 3.23: Dodajanje nove aplikacije na portalu *Vendor Portal for BlackBerry World* (posnetek portala *Vendor Portal for BlackBerry World*)

Na zgornji sliki je prikazana začetna spletna stran, kjer pričnemo z nalaganjem naše podpisane aplikacije na portal. Po izpolnitvi vseh podatkov, ki jih je potrebno vpisati o sami aplikaciji, je potrebno dodati še nekaj slik, ki bodo prikazane na spletnem portalu *BlackBerry World*. Po uspešno vneseni aplikaciji na portalu jo pošljemo v pregled. Šele po pregledu jo lahko objavimo na spletnem portalu *BlackBerry World*. Iz tega vidika ima BlackBerry zelo dobro poskrbljeno za varnost svojih naprav. [37]

Po uspešno prestatnem pregledu smo igro *Time Attacker* s pomočjo portala *Vendor Portal for BlackBerry World* objavili na portalu *BlackBerry World* [37], kar prikazuje tudi spodnja slika.

The screenshot shows the BlackBerry Developer Vendor Portal interface. At the top, there is a navigation bar with links for BlackBerry Developer, Blog, YouTube, Forum, Signing, Feedback, and Logout. The main heading is "Products", with a "Filter" button, a "+ New Product" button, and a search bar labeled "Search for Products".

The user profile "Blaz Kokol" is displayed. A sidebar menu on the left includes: Products, Promotions, Reports, Change Requests, Users, Testers, and Settings. The main content area features the product "Time Attacker" with a 5-star rating. A vertical menu on the left of the product page lists: Details, Listings, Releases, In-App Products, Reviews, and Change History, each with a green checkmark. To the right of this menu are buttons for "Pin to top" and "More".

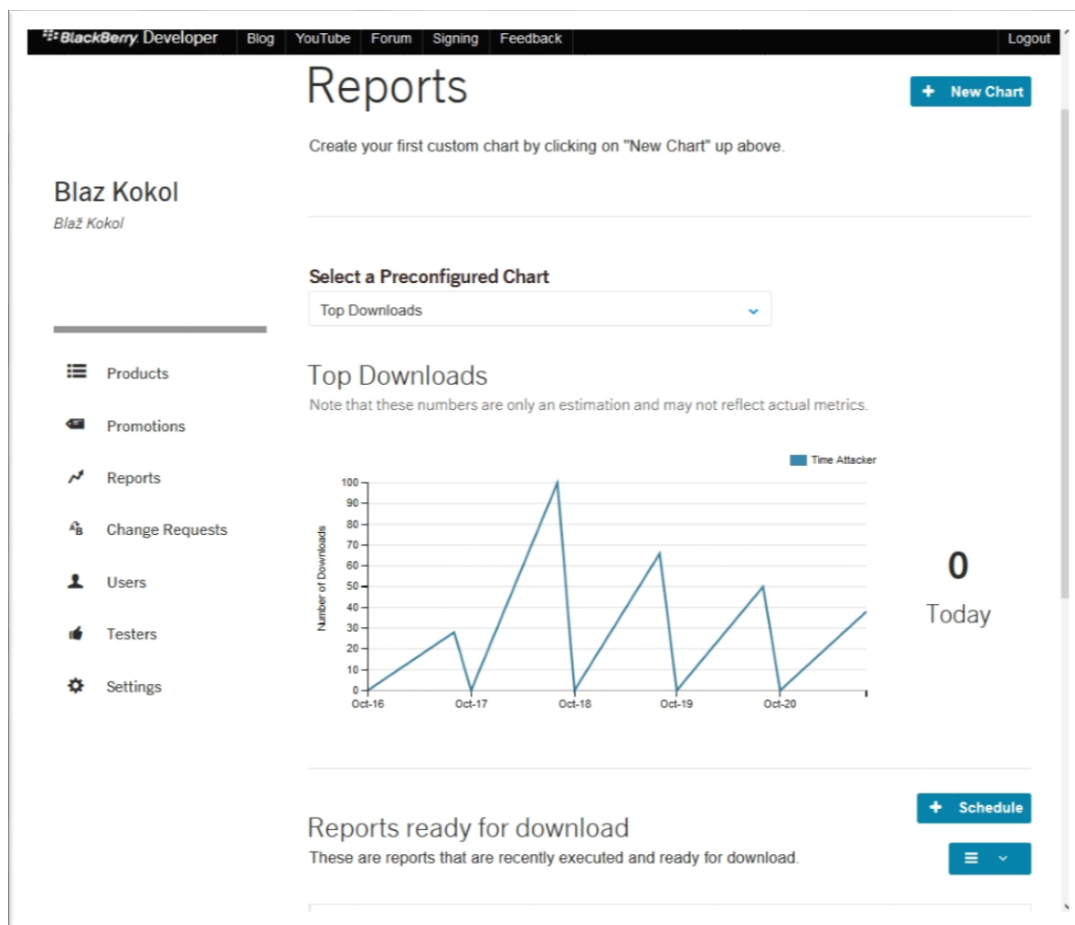
Key statistics for the product are shown in a grid:

- 0 New reviews today
- 38 Downloads today
- US\$0 Money earned today
- 0 Purchases today

A notification box indicates "Release 1.0.0.1 is now Approved" with a link to "More recent events".

The footer contains links for Technologies (Native), Programs (Beta Zone), Support (Code Signing Help), BlackBerry World (Get your signing key), and social media icons for Facebook, YouTube, and Twitter.

Slika 3.24: Podatki o igri Time Attacker na portalu *Vendor Portal for BlackBerry World* (posnetek portala *Vendor Portal for BlackBerry World*)



Slika 3.25: Poročilo o igri Time Attacker na portalu *Vendor Portal for BlackBerry World* (posnetek portala *Vendor Portal for BlackBerry World*)

Zgornja slika nam prikazuje statistiko prenosov aplikacije s spleta po dnevih.

Poglavje 4 Sklepne ugotovitve

Na podlagi v tem delu prehojene poti od ideje do izdelka ugotavljamo, da je za razvoj računalniške igre za platformo BlackBerry PlayBook potrebno veliko strokovnega znanja in izkušenj pri iskanju ustreznih rešitev. Posebej težavno je, kadar se s takšnim izzivom srečaš prvič. Tehnologije HTML, JS in CSS ponujajo mnogotere možnosti uporabe. Nekatere med njimi smo obravnavali opisno in postopkovno, kar povečuje uporabno vrednost našega dela.

Naš izvirni prispevek vidimo v iskanju konceptualnih rešitev pri razvoju igre (Time Attacker) ter prilagoditvi aplikacije za uporabo na BlackBerry PlayBook oziroma za uporabo na mobilnih napravah brez uporabe tipkovnice.

Kot glavni prispevek štejemo, da je naše delo iz izobraževalnega vidika uporabno kot priročnik bodočemu uporabniku.

Želimo si, da bo naše delo pritegnilo številne mlade bralce in pri njih vzbudilo zanimanje in motivacijo, da se bodo tudi sami podali na tako zanimivo in poučno »pot«.

Na osnovi našega dela, spoznanj in izkušenj, vidimo tudi možnosti za nadaljnji razvoj. Vsekakor bi nove izpeljave in rešitve zahtevale ažurno sledenje razvoju programske opreme in naprav, v smislu čim večjega izkoristka potenciala, ki ga bo nova tehnologija ponujala. Aplikacija in razširitev naših zamisli je z nekaj prilagoditvami izvedljiva tudi na drugih napravah, ki uporabljajo kateri drugi operacijski sistem (npr. Android, Windows Phone). Od programerjeve ustvarjalnosti je odvisna pot razvoja in končni izdelek. S tem mislimo predvsem na avtorsko in izvirno zasnovo koncepta igre ter izpeljave posameznih rešitev (zvok, grafika in razni parametri ter nastavitve), kar v bistvu povzema naše celotno delo.

Kazalo slik

| | |
|----------------------------------------------------------------------------------------|----|
| Slika 2.1: Tablični računalnik BlackBerry PlayBook [3]..... | 2 |
| Slika 2.2: Sintaksa HTML5 (posnetek programske kode HTML5) | 4 |
| Slika 2.3: Sintaksa CSS [13] | 6 |
| Slika 2.4: Razvojno okolje NetBeans (posnetek strani programa NetBeans IDE 8.0)..... | 8 |
| Slika 2.5: Razvojno okolje GIMP (posnetek strani programa GIMP 2.8.10) | 9 |
| Slika 3.1: Začetni meni igre Time Attacker (posnetek aplikacije) | 11 |
| Slika 3.2: Dirka v igri Time Attacker (posnetek aplikacije) | 12 |
| Slika 3.3: Zmaga v igri Time Attacker (posnetek aplikacije)..... | 13 |
| Slika 3.4: Izsek iz programske kode – izbira poti do slike | 14 |
| Slika 3.5: Izsek iz programske kode – izbira imena slike | 14 |
| Slika 3.6: Izsek iz programske kode – širina, pasovi, št. krogov, št. avtomobilov..... | 14 |
| Slika 3.7: Izsek iz programske kode – atributi proge | 15 |
| Slika 3.8: Izsek iz programske kode – sprememba barv | 16 |
| Slika 3.9: Primer slike <i>sprites.png</i> | 17 |
| Slika 3.10: Izsek iz programske kode – izbira pozicij slike | 17 |
| Slika 3.11: Izsek iz programske kode - poslušalec dogodkov | 18 |
| Slika 3.12: Izsek iz programske kode – del funkcije <i>deviceMotionHandler</i> | 19 |
| Slika 3.13: Izsek iz programske kode – sprememba zvočnega ozadja | 20 |
| Slika 3.14: Izsek iz programske kode – datoteka <i>config.xml</i> | 20 |
| Slika 3.15: Ripple Emulator (posnetek strani programa Google Chrome)..... | 21 |
| Slika 3.16: Prikaz vklopa razvijalskega načina [28]..... | 22 |
| Slika 3.17: Primer registracije ključev (posnetek programa cmd) | 23 |
| Slika 3.18: Primer kreiranja razhroščevalnega žetona (posnetek programa cmd) | 24 |
| Slika 3.19: Primer nameščanja razhroščevalnega žetona (posnetek programa cmd) | 25 |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Slika 3.20: Primer priprave aplikacije za nadaljnjo implementacijo (posnetek programa cmd) | 26 |
| Slika 3.21: Primer implementacije aplikacije na BlackBerry PlayBook (posnetek programa cmd) | 27 |
| Slika 3.22: Primer podpisa aplikacije za nadaljnjo obdelavo (Posnetek programa cmd) | 28 |
| Slika 3.23: Dodajanje nove aplikacije na portalu <i>Vendor Portal for BlackBerry World</i> (posnetek portala <i>Vendor Portal for BlackBerry World</i>) | 29 |
| Slika 3.24: Podatki o igri Time Attacker na portalu <i>Vendor Portal for BlackBerry World</i> (posnetek portala <i>Vendor Portal for BlackBerry World</i>) | 30 |
| Slika 3.25: Poročilo o igri Time Attacker na portalu <i>Vendor Portal for BlackBerry World</i> (posnetek portala <i>Vendor Portal for BlackBerry World</i>) | 31 |

Literatura

- [1] BlackBerry PlayBook, *Wikipedia* [Online]. Dosegljivo: http://en.wikipedia.org/wiki/BlackBerry_PlayBook
- [2] BlackBerry Limited, *Wikipedia* [Online]. Dosegljivo: http://en.wikipedia.org/wiki/BlackBerry_Limited
- [3] BlackBerry PlayBook Help & Support [Online]. *Uradna spletna stran podjetja BlackBerry* Dosegljivo: <http://us.blackberry.com/support/tablets/playbook/get-started.html>
- [4] LTE (telecommunication), *Wikipedia* [Online]. Dosegljivo: http://en.wikipedia.org/wiki/LTE_%28telecommunication%29
- [5] Introduction — HTML5, *W3C* [Online]. Dosegljivo: <http://www.w3.org/TR/html5/introduction.html#history-0>
- [6] HTML5, *Wikipedia* [Online]. Dosegljivo: <http://en.wikipedia.org/wiki/HTML5>
- [7] Plan 2014, *W3C* [Online]. Dosegljivo: <http://dev.w3.org/html5/decision-policy/html5-2014-plan.html#plan>
- [8] HTML5 New Elements, *W3Schools* [Online]. Dosegljivo: http://www.w3schools.com/html/html5_new_elements.asp
- [9] HTML5 Browser Support, *W3Schools* [Online]. Dosegljivo: http://www.w3schools.com/html/html5_browsers.asp
- [10] JavaScript Tutorial, *W3Schools* [Online]. Dosegljivo: <http://www.w3schools.com/js/default.asp>
- [11] Cascading Style Sheets, *Wikipedia* [Online]. Dosegljivo: http://en.wikipedia.org/wiki/Cascading_Style_Sheets#CSS_3

- [12] H. W. Lie in B. Bos, *Cascading Style Sheets: Designing for the Web*, 3rd ed., Indiana: Addison-Wesley Professional, 2005.
- [13] CSS Syntax, *W3Schools* [Online]. Dosegljivo:
http://www.w3schools.com/css/css_syntax.asp
- [14] CSS Browser Support Reference, *W3Schools* [Online]. Dosegljivo:
http://www.w3schools.com/cssref/css3_browsersupport.asp
- [15] JavaScript, *Wikipedia* [Online]. Dosegljivo <http://en.wikipedia.org/wiki/JavaScript>
- [16] P. Wilton and J. McPeak, "Introduction to JavaScript and the Web" in *Beginning JavaScript*, 4th ed. Indiana: Wiley Publishing, Inc., 2010, str. 1-16
- [17] ECMAScript, *Wikipedia* [Online]. Dosegljivo:
<http://en.wikipedia.org/wiki/ECMAScript>
- [18] JavaScript Syntax, *W3Schools* [Online]. Dosegljivo:
http://www.w3schools.com/js/js_syntax.asp
- [19] An Introduction to NetBeans, *NetBeans* [Online]. Dosegljivo:
<https://netbeans.org/about/>
- [20] NetBeans, *Wikipedia* [Online]. Dosegljivo: <http://en.wikipedia.org/wiki/NetBeans>
- [21] NetBeans IDE Features, *NetBeans* [Online]. Dosegljivo:
<https://netbeans.org/features/index.html>
- [22] GIMP, *Wikipedia* [Online]. Dosegljivo: <http://en.wikipedia.org/wiki/GIMP>
- [23] Main Windows, *GIMP* [Online]. Dosegljivo: <http://docs.gimp.org/2.8/en/gimp-concepts-main-windows.html>
- [24] How to build a racing game - conclusion, *Code in Complete* [Online]. Dosegljivo:
http://codeincomplete.com/posts/2012/6/30/javascript_racer_v4_final/
- [25] HTML5 Device Motion Event, *HTML WebWorks on BlackBerry Playbook* [Online].
Dosegljivo:
<https://developer.blackberry.com/playbook/html5/apis/devicemotionevent.html>
- [26] Create your config.xml file, *HTML WebWorks on BlackBerry Playbook* [Online].
Dosegljivo:

https://developer.blackberry.com/playbook/html5/documentation/working_with_config_xml_file_1866970_11.html

- [27] Ripple Emulator (Beta), *Chrome web store* [Online]. Dosegljivo:
<https://chrome.google.com/webstore/detail/ripple-emulator-beta/geelfhphabnejhdalkjhgipohgpdnoc?hl=en>
- [28] Rooting your BlackBerry PlayBook is easier than ever with DingleBerry 3.x , *CrackBerry* [Online]. Dosegljivo: <http://crackberry.com/rooting-your-blackberry-playbook-easier-ever-dingleberry-3x>
- [29] BlackBerry Keys Order Form, *BlackBerry Jam Zone* [Online]. Dosegljivo:
<https://www.blackberry.com/SignedKeys/codesigning.html>
- [30] Set up for signing, *HTML WebWorks on BlackBerry Playbook* [Online]. Dosegljivo:
https://developer.blackberry.com/playbook/html5/documentation/signing_setup.html
- [31] (PlayBook) Code signing request failed because this file has been previously signed, *Network Programming in .NET* [Online]. Dosegljivo:
<http://networkprogramming.wordpress.com/2012/01/25/playbook-code-signing-request-failed-because-this-file-has-been-previously-signed/>
- [32] Deploy to a device, *HTML WebWorks on BlackBerry Playbook* [Online]. Dosegljivo:
https://developer.blackberry.com/playbook/html5/documentation/running_your_app_on_a_device.html
- [33] Distributing your app, *HTML WebWorks on BlackBerry Playbook* [Online].
Dosegljivo:
https://developer.blackberry.com/playbook/html5/documentation/distributing_your_app_1866990.html
- [34] BlackBerry World, *BlackBerry Developer* [Online]. Dosegljivo:
<https://developer.blackberry.com/blackberryworld/>
- [35] Signing your app, *HTML WebWorks on BlackBerry Playbook* [Online]. Dosegljivo:
https://developer.blackberry.com/playbook/html5/documentation/signing_your_app_1920008_11.html
- [36] Vendor Portal for BlackBerry World, *BlackBerry Developer* [Online]. Dosegljivo:
<https://appworld.blackberry.com/isvportal/login.do>

[37] Apply for a BlackBerry World vendor account, *BlackBerry Developer* [Online].

Dosegljivo:

https://developer.blackberry.com/devzone/blackberryworld/apply_for_a_blackberry_world_membership_account.html#pde1350406051752

[38] Time Attacker, *BlackBerry World* [Online]. Dosegljivo:

<http://appworld.blackberry.com/webstore/content/59942400/?lang=en&countrycode=SI>