

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Žiga Černigoj

**Prenos interaktivnih spletnih vsebin s  
Flasha na HTML5**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Saša Divjak

Ljubljana, 2016

Fakulteta za računalništvo in informatiko podpira javno dostopnost znanstvenih, strokovnih in razvojnih rezultatov. Zato priporoča objavo dela pod katero od licenc, ki omogočajo prosto razširjanje diplomskega dela in/ali možnost nadaljne proste uporabe dela. Ena izmed možnosti je izdaja diplomskega dela pod katero od Creative Commons licenc <http://creativecommons.si>

Morebitno pripadajočo programsko kodo praviloma objavite pod, denimo, licenco *GNU General Public License, različica 3*. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Besedilo teme diplomskega dela študent prepíše iz študijskega informacijskega sistema, kamor ga je vnesel mentor. V nekaj stavkih bo opisal, kaj pričakuje od kandidatovega diplomskega dela. Kaj so cilji, kakšne metode uporabiti, morda bo zapisal tudi ključno literaturo.



*Zahvaljujem se prijateljem in družini za podporo in vzpodbujajoče besede in dejanja.*

*Zahvaljujem se tudi mentorju, prof. dr. Saši Divjaku ter predstojniku Laboratorija za računalniško grafiko in multimedije, doc. dr. Matiji Maroltu za strokovno pomoč.*



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Pregled tehnologij</b>	<b>3</b>
2.1	Flash . . . . .	3
2.2	HTML5 . . . . .	5
<b>3</b>	<b>Orodja za razvoj</b>	<b>9</b>
3.1	Razvoj v Flashu . . . . .	10
3.2	Razvoj v HTML5 . . . . .	12
<b>4</b>	<b>Prehod s Flasha na HTML5</b>	<b>17</b>
4.1	Pretvorba obstoječih Flash aplikacij . . . . .	17
4.2	Razvoj od začetka v HTML5 . . . . .	18
<b>5</b>	<b>Primerjava na primeru</b>	<b>21</b>
5.1	Izbrane igre . . . . .	21
5.2	Uporabljena orodja . . . . .	23
5.3	Rezultati . . . . .	26
<b>6</b>	<b>Sklepne ugotovitve</b>	<b>31</b>
	<b>Literatura</b>	<b>34</b>





# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>HTML</b>	hypertext markup language	jezik za označevanje nadbesedila
<b>HTML5</b>	HTML version 5	HTML verzije 5
<b>JS</b>	JavaScript	JavaScript
<b>CSS</b>	Cascading Style Sheets	kaskadne slogovne predloge
<b>PHP</b>	Hypertext Preprocessor	predprocesor nadbesedila
<b>2D</b>	two dimensional	dvodimenzionalno
<b>3D</b>	three dimensional	tridimenzionalno
<b>API</b>	Application Programming Interface	aplikacijski programski vmesnik
<b>WYSIWYG</b>	What You See Is What You Get	kar vidiš, to dobiš
<b>WebGL</b>	Web Graphics Library	knjižnica za spletno grafiko
<b>AS</b>	ActionScript	ActionScript
<b>IDE</b>	Integrated Development Environment	integrirano razvojno okolje



# Povzetek

**Naslov:** Prenos interaktivnih spletnih vsebin s Flasha na HTML5

**Avtor:** Žiga Černigoj

Diplomsko delo se osredotoča na problematiko prenosa interaktivnih Flash aplikacij in iger na HTML5. Raziskali smo možnosti za prehod s Flasha na HTML5. Možnosti smo razdelili na dve glavni množici, pretvorbo obstoječih izdelkov in razvoj od začetka. Na primeru iger za učenje glasbene teorije smo preizkusili po nekaj možnosti iz vsake množice. Igre smo pretvorili s spletnim orodjem Google Swiffy in s programoma Adobe Flash Professional in Adobe Animate. Nekaj iger smo razvili od začetka. Uporabili smo HTML5 igralno ogrodje Phaser, program za ustvarjanje iger Construct 2 in JavaScript knjižnico JCanvas. Odločili smo se, da je za prehod boljši razvoj od začetka z izbranim orodjem. Čeprav je potrebnih več sredstev, nimamo težav z nedelovanjem iger in imamo možnost za izboljšave.

**Ključne besede:** Flash, HTML5, spletne igre, interaktivne spletne aplikacije, bogate spletne aplikacije, animacija.



# Abstract

**Title:** Porting interactive web content from Flash to HTML5

**Author:** Žiga Černigoj

The thesis focuses on the issue of transferring or porting Flash-based content to HTML5. We have explored options for moving from Flash to HTML5. Options are divided into two main sets, the conversion of existing products and the development from the beginning. We tested and compared the options on games for learning music theory. Games were converted using online converter Google Swiffy and authoring programs Adobe Flash Professional and Adobe Animate. We have developed a few games from the beginning. We used HTML5 game framework Phaser, game creation program Construct 2 and JavaScript library JCanvas. We concluded that development from the beginning with chosen tools is better option for porting. Although we need more resources, games work and we have an opportunity for improvements.

**Keywords:** Flash, HTML5, online games, interactive web applications, rich internet applications, animation.



# Poglavje 1

## Uvod

Na spletu je dostopna velika količina interaktivnih vsebin, katerih velik del predstavlja bogate spletne aplikacije in igre. Te so napisane v tehnologijah, kot so Adobe Flash, Java Applets, Microsoft Silverlight in HTML5 (skupaj tehnologij HTML verzije 5, JS in CSS).

Od izdaje HTML5 se pojavlja težnja po brskalnikih brez vtičnikov, saj nekateri vtičniki, najpogosteje so to NPAPI vtičniki, predstavljajo varnostne luknje v brskalnikih, poleg tega pa so lahko vzrok za počasnejše delovanje brskalnika. Tako se tehnologije Flash, Java in Silverlight od prihoda HTML5 naprej pri razvoju spletnih aplikacij postopoma opuščajo, saj so zanje potrebni (NPAPI) vtičniki.

Silverlight je uradno že opuščen, vendar ga nekateri brskalniki še vedno podpirajo. Med bolj znanimi brskalniki so to Mozilla Firefox, Safari in Internet Explorer.

Oracle razvijalcem Java Applets priporoča, naj razvijajo s tehnologijo Java Web Start, ki ne potrebuje vtičnikov, saj ne vzdržujejo in posodablja več potrebnih vtičnikov.

Flash kljub preteklim napovedanim datumom ukinitve podpore zaenkrat ostaja podprt v večini brskalnikov, predvsem zaradi popularnosti Flash tehnologije in števila aplikacij ter iger, napisanih v njej. Za razliko od Mozille Firefox, ki še vedno podpira NPAPI vtičnik, večina ostalih brskalnikov pod-

pira PPAPI vtičnike, ki veljajo za varnejše ter se jih redno vzdržuje [15, 10].

HTML5 je izraz, ki ga največkrat povezujemo z zbirko tehnologij HTML5, CSS in JS, s katerimi lahko ustvarimo interaktivne spletne strani. Zanje ne potrebujemo vtičnikov. Dodatna prednost je, da je HTML5 zmožen upravljati z multimedijско vsebino, pred njim pa smo morali za to uporabiti eno izmed ostalih že omenjenih tehnologij skupaj s primernim vtičnikom. Slabost tehnologije HTML5 je le to, da brskalniki še ne podpirajo vseh njegovih funkcionalnosti, kar lahko povzroči nepredvideno obnašanje pri nepodprtih funkcionalnostih. Vendar se situacija z aktivnim razvojem brskalnikov stalno izboljšuje.

Problem, ki se ga bomo v diplomski nalogi lotili, je, kaj storiti, da bo naša aplikacijo mogoče uporabljati tudi po ukinitvi podpore Flasha v brskalnikih. Ta problem zadeva vse avtorje Flash aplikacij, ki pa imajo na voljo več možnosti. Prva je uporaba orodja, ki pretvori vsebino SWF datotek (SWF je format datotek, v katerih so shranjene Flash aplikacije) v HTML5. Primeri takih orodij so Google Swiffy, Haxe in Mozilla Shumway. Druga možnost je, da celotno aplikacijo sami razvijejo v tehnologiji HTML5. Pri tem so v pomoč razna orodja in ogrodja in knjižnice, pri igrah pa še igralni pogoni.

Cilj diplomske naloge je izbrati najbolj primeren način za prenos iger in aplikacij s Flasha v HTML5. Rezultat bo primerjava različnih metod, s katerimi lahko pretvorimo oziroma razvijemo igro ali aplikacijo v HTML5. Hkrati pa bo cilj diplomske naloge prenos dela pojmovnika glasbene teorije v nove tehnologije.

V nadaljevanju diplomske naloge bomo preleteli tehnologiji Flash in HTML5. Nato bomo raziskali in primerjali različne metode, s katerimi lahko pretvorimo oziroma razvijemo igro ali aplikacijo v HTML5. Na koncu bomo zapisali še ugotovitve ter utemeljili, kateri način je najbolj primeren.



# Poglavje 2

## Pregled tehnologij

V tem poglavju bomo najprej opisali tehnologijo Flash in skupek tehnologij HTML5, CSS in JS. V naslednjem poglavju pa bomo opisali orodja, s katerimi si pomagamo pri razvoju aplikacij in iger v teh tehnologijah.

### 2.1 Flash

Flash je platforma za razvoj bogatih spletnih aplikacij, spletnih iger, animacij in tudi namiznih ter mobilnih aplikacij. Omogoča predvajanje in pretakanje multimedijskih vsebin ter zajem miške, tipkovnice, mikrofona in kamere.

V Flash platformo je zajetih več tehnologij in orodij. Zajeta so orodja za oblikovanje in razvoj aplikacij, ogrodja za lažji razvoj, predvajalniki aplikacij na odjemalčevi strani ter Flash strežniki. Slika 2.1 prikazuje strukturo Flash platforme v letu 2008.

Orodja za razvoj Flash aplikacij so na primer Adobe Flash Builder (prej Adobe Flex Builder), Adobe Animate (prej Adobe Flash Professional) in FlashDevelop. Za razvoj lahko uporabimo tudi katerikoli urejevalnik besedila ali integrirano razvojno okolje, če uporabimo Apache Flex SDK. Adobe Flash Builder in FlashDevelop sta integrirani razvojni okolji, Adobe Animate pa je orodje za izdelavo animacij in obdelavo multimedije ter omogoča objavo animacij v HTML5, WebGL, Flash in drugih formatih.



Slika 2.1: Slika prikazuje strukturo Flash platforme v letu 2008 [1].

Flash aplikacije so med razvojem lahko sestavljene iz več izvornih datotek, najpogosteje v formatih FLA in AS. Končane in prevedene aplikacije združene v eno datoteko formata SWF. Aplikacij v tem formatu ne moremo več urejati.

Za prikaz vsebine v SWF formatu se lahko uporabi Adobe Flash Player, Adobe AIR ali druge predvajalnike kot na primer Scaleform GFX. Adobe Flash Player omogoča predvajanje vsebine v spletnih brskalnikih in je podprt v vseh večjih operacijskih sistemih. Adobe AIR omogoča predvajanje vsebin na starejših pametnih telefonih. Njegov predhodnik je bil Adobe Flash Lite.

Za interaktivne animacije, igre ter aplikacije se uporablja programski jezik ActionScript. To je programski jezik, osnovan na ECMAScriptu, enako kot JavaScript.

Flash je bil pred prihodom HTML5 najbolj razširjena tehnologija, ki je omogočala interaktivnost na spletu. Zdaj pa se v večini uporablja samo še za razvoj iger za mobilne naprave s tehnologijo Adobe AIR.

Slabost Flasha so vtičniki, ki jim brskalniki počasi ukinjajo podporo. Alternativa je HTML5, ki omogoča interaktivnost brez vtičnikov. Na voljo so orodja, s katerimi se da Flash aplikacije in igre pretvoriti v HTML5 [31].

## 2.2 HTML5

Dandanes je najboljša alternativa Flashu tehnologija HTML5. Vendar samo z jezikom HTML ne zmoremo vsega. Obliko in postavitev dosežemo s CSS (kaskadne stilske predloge), obnašanje strani pa lahko definiramo s skriptami v jeziku JavaScript. Ko primerjamo Flash in HTML5, torej z izrazom HTML5 ciljamo na skupek tehnologij HTML5, CSS in JavaScript.

### 2.2.1 HTML

HTML je označevalni jezik, namenjen predstavitvi vsebin na spletu. Skupaj s tehnologijama CSS in JavaScript predstavlja temelj za razvoj spletnih strani ter uporabniške vmesnike za mobilne in spletne aplikacije. Brskalniki preberejo HTML datoteke in jih predstavi kot spletne strani. HTML semantično opredeljuje strukturo spletne strani.

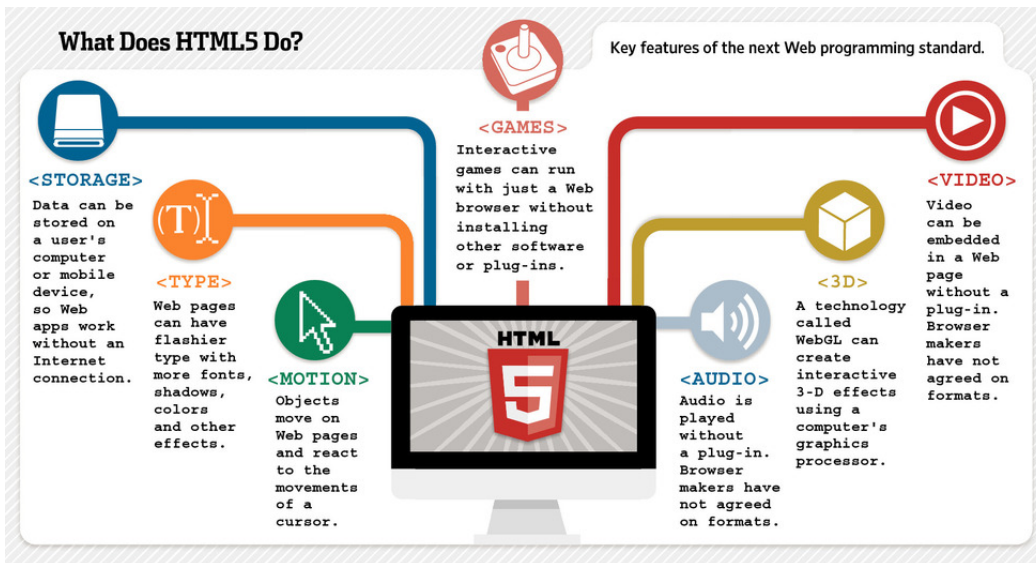
Gradniki HTML dokumentov so HTML elementi, ki so sestavljeni iz začetne in končne značke ter vsebine med značkama oziroma samo iz začetne značke, če element nima vsebine. Značke so zapisane kot `<znacka>` za začetno značko in `</znacka>` za končno značko. Začetne značke lahko vsebujejo še attribute, zapisane na način `imeatributa="lastnostatributa"`.

V HTML lahko vdelamo skripte, napisane v jezikih, kot je JavaScript. Le-te lahko vplivajo na obnašanje spletnih strani. HTML se lahko sklicuje tudi na CSS datoteke, ki opredeljujejo izgled besedila in drugih vsebin na spletnih straneh.

Za razvoj tehnologij HTML in CSS skrbi W3C (World Wide Web Consortium) [33].

### 2.2.2 HTML5

HTML5 je bil izdan leta 2014 in uvaža podporo multimediji v HTML. Vključuje HTML4, XHTML1 ter DOM 2 ter nove funkcionalnosti. Uvaža značke in programske vmesnike za kompleksne spletne aplikacije. Zaradi tega je tudi potencialni kandidat za mobilne aplikacije, ki tečejo na različnih platformah



Slika 2.2: Slika prikazuje zmogljivosti HTML5 [29].

upoštevaje funkcije, zasnovane za mobilne naprave z nizko porabo električne energije. Slika 2.2 prikazuje zmogljivosti HTML5.

V HTML5 so zajete tudi nove sintaktične lastnosti. Za vključevanje ter upravljanje multimedijskih in grafičnih vsebin so bili dodani elementi `<video>`, `<audio>` in `<canvas>`. Vključena je tudi podpora za vektorsko grafiko (SVG) ter matematične formule v MathML. Za obogatitev semantične vsebine so bili dodani novi strukturni elementi, kot so `<main>`, `<section>`, `<article>`, `<header>`, `<footer>`, `<aside>`, `<nav>` in `<figure>`. Nekateri elementi so bili spremenjeni, pri nekaterih so na voljo novi atributi, pri nekaterih pa so bili atributi odstranjeni.

Programski vmesniki (API) in dokumentni objektni model (DOM) so danes temeljni del specifikacije HTML5. HTML5 bolje definira obdelavo neveljavnih dokumentov [34].

### 2.2.3 CSS

Kaskadne stilske predloge uporabljajo slogovni jezik za opis predstavitve dokumentov, napisanih v HTML, XHTML in XML, SVG in drugih označevalnih

jezikih.

CSS ločuje vsebino dokumenta od njegove oblike. Ta ločitev lahko izboljša dostopnost in fleksibilnost. Tako se CSS koda ne ponavlja in lahko določa izgled več HTML dokumentov.

HTML dokument lahko prikažemo v različnih slogih za različne načine upodabljanja ter za zaslone različnih velikosti.

Oblikovne spremembe za enega ali več dokumentov je mogoče uveljaviti hitro in enostavno s posodobitvijo ene CSS datoteke, pri tem pa ni potrebno spreminjati značk v dokumentih.

V primeru, da je za nek element več pravil, se le-ta določajo kaskadno, pravilom se dodelijo prioritete, tako da so rezultati predvidljivi.

CSS specifikacije vzdržuje W3C konzorcij. MIME tip `text/css` je registriran za uporabo s CSS, ki je določen v RFC 2318 iz leta 1998. CSS datoteke je možno tudi validirati s CSS validatorjem [32].

## 2.2.4 JavaScript

JavaScript je visokonivojski, dinamično tipiziran programski jezik, ki se interpretira. Standardiziran je v specifikaciji jezika ECMAScript. Podprt je v vseh brskalnikih in zanj ni potrebnih nobenih vtičnikov.

JavaScript podpira delo z besedilom, polji, datumi in regularnimi izrazi. Ne vključuje vhodno-izhodnih funkcionalnosti, kot so omrežne povezave, shranjevanje ali grafične zmogljivosti. Pri teh se zanaša na gostiteljevo okolje, v katerega je vdelan.

Sintaksa JavaScripta izhaja iz jezika C, na semantiko in obliko pa sta vplivala programska jezika Self in Scheme.

JavaScript je uporabljen tudi v namenskih brskalnikih in namiznih aplikacijah ter za strežniške spletne aplikacije.

Običajno se JavaScript na odjemalcu obravnava kot interpretiran jezik, a novejši brskalniki opravljajo "just-in-time" prevajanje.

Uporabljen je tudi za razvoj iger, namiznih in mobilnih aplikacij ter za strežniško omrežno programiranje z izvajalnimi okolji, kot je Node.js [35].



## Poglavje 3

# Orodja za razvoj

Za razvoj običajno uporabljamo integrirana razvojna okolja ali namenska orodja. Integrirana razvoja okolja nam pomagajo s samodokončanjem kode, predlaganjem besed (ukazov, imen funkcij in spremenljivk) in razhroščevanjem. Omogočajo tudi enostavno prevajanje in poganjanje kode. Namenska orodja omogočajo razvoj v tehnologiji, za katero so ustvarjena. Pogosto vključujejo tudi grafični urejevalnik, s katerim urejamo vizualno podobo aplikacije ali igre.

Pri programiranju si pogosto pomagamo z že napisanimi funkcionalnostmi, ki nam olajšajo in pohitijo razvoj. Pri razvoju aplikacij lahko uporabimo knjižnice in ogrodja. Pri razvoju iger običajno uporabimo igralni pogon. Knjižnice in ogrodja vključujejo že napisane najpogostejše in najpomembnejše funkcije, ki so potrebne za delovanje vsake aplikacije oziroma igre. Igralni pogoni zajemajo več knjižnic in olajšajo razvoj, saj skrbijo za upodabljanje grafike, predvajanje zvoka in animacij, v igro vnašajo simulacijo fizikalnih zakonov in podpirajo različne načine interakcije s človekom (branje vnosa preko tipkovnice, miške ali zaslona na dotik ter zajem slike ali zvoka). Tako nam ni treba pisati funkcionalnosti, ki predstavljajo temelje aplikacij ali iger, ampak se lahko osredotočimo samo na glavne funkcionalnosti aplikacije oziroma igre, ki jo razvijamo.

## 3.1 Razvoj v Flashu

### 3.1.1 Orodja za razvoj

Flash aplikacije razvijamo v orodjih, ki omogočajo ustvarjanje in obdelovanje multimedijskih vsebin, animacij ter programiranje logike v jeziku ActionScript.

#### Adobe orodja

Adobe nudi uradna orodja za razvoj Flash aplikacij [31].

Adobe Animate (prej Adobe Flash Professional) omogoča ustvarjanje animacij, hkrati pa tudi urejanje kode v Actionscript-u, ki omogoča interaktivnost. Projekte lahko izvozimo za več platform, tudi za HTML5.

Adobe Flash Builder (prej Adobe Flex Builder) je integrirano razvojno okolje, ki omogoča urejanje, prevajanje, razhroščevanje in poganjanje ActionScript kode.

Adobe Scout analizira in grafično prikaže zmogljivost namiznih in mobilnih aplikacij.

Apache Flex (prej Adobe Flex) je paket oziroma ogrodje za razvoj aplikacij v integriranih razvojnih okoljih.

CrossBridge je odprtokoden nabor orodij za navzkrižno prevajanje (cross-compile) aplikacij, napisanih v C ali C++, da jih lahko poganjamo v Adobe Flash Player-ju ali v Adobe AIR.

#### Orodja drugih ponudnikov

Na voljo pa so nam tudi neuradna orodja [31]. Nekatera med njimi so tudi brezplačna ali odprtokodna.

FlashBuilder je odprtokodno integrirano razvojno okolje, ki omogoča razvoj v ActionScriptu, podpira pa tudi razvoj v tehnologijah HTML, CSS, JS ter uporabo sistemov za upravljanje z izvorno kodo.



Powerflasher FDT je na Eclipse platformi osnovano integrirano razvojno okolje za razvoj Flash vsebin.

Haxe zajema odprtokoden visokonivojski programski jezik in prevajalnik, ki je zmožen kodo iz tega jezika prevesti v jezike ActionScript 3, JavaScript, Java, C#, C++, Python, PHP, ter Lua. OpenFL in Flambe sta popularni na Haxe osnovani ogrodji, ki omogočata izvoz aplikacij za več platform.

SWFTools je odprtokodna zbirka orodij za ustvarjanje in upravljanje SWF datotek.

Swfmill je brezplačen program, ki omogoča pretvarjanje med XML in SWF formatoma. Uporablja SWFML, dialekt formata XML, ki je podoben SWF formatu.

### 3.1.2 Ogrodja in igralni pogoni

#### Ogrodja

Na voljo je kar nekaj ogrodij za razvoj v Flashu. Med njimi so Apache Flex, SomaCore, AsWing, CASA Lib, PureMVC, Wrapper, Zest3D in drugi [28]. V naslednjem odstavku bomo opisali Apache Flex, eno izmed najbolj razširjenih odprtokodnih resitev. Za razliko od drugih, je Apache Flex še aktivno vzdrževan.

Apache Flex je odprtokodno Flash ogrodje za razvoj spletnih in mobilnih aplikacij. Z njim lahko z istim programskim modelom, orodji in programsko osnovo enostavno ustvarimo mobilne aplikacije za iOS, Android in BlackBerry Tablet OS ter spletne in namizne aplikacije. Vključen paket SDK nudi prevajalnik in omogoča, da aplikacije razvijamo in prevajamo v poljubnem integriranem razvojnem okolju. Prevajalnik združi MXML dokumente in datoteke z ActionScript kodo ter izvozi SWF datoteko, ki predstavlja končano aplikacijo. Datoteko lahko poganja Flash player, lahko pa jo z Adobe AIR prevedemo v aplikacije za Windows, Mac OS, Android, iOS ali Blackberry [7].

## Igralni pogoni

Za razvoj iger pa imamo na izbiro igralne pogone, kot so Starling, Flixel, Flare3D, FlashPunk, Citrus ter Away3D [6].

Igralna pogona Starling in Away3D sta zajeta v Adobe Gaming SDK, uporablja pa ju tudi igralni pogon Citrus. Adobe Gaming SDK je zbirka ogrodiv, vzorcev kode in učnih sredstev, ki so v pomoč razvijalcem pri razvoju iger v ActionScriptu za več platform [2].

Starling je igralno ogrodje za razvoj iger za namizne in mobilne platforme. Razvit je v ActionScriptu. Nekateri razvijalci so svoje odprtokodne knjižnice osnovali na njem in razširili njegove obstoječe funkcionalnosti. Tak je na primer StarlingPunk, ki združuje funkcionalnosti Starlinga in FlashPunka. Uporablja Stage3D, programski vmesnik za izrisovanje 3D interaktivne grafike. V Starlingu je napisanih nekaj popularnejših komercialnih iger, na primer Angry Birds [16].

Away3D je platforma, ki združuje grafični urejevalnik in integrirano razvojno orodje Away Builder, grafični pogon Away3D ter pogon za simulacijo fizike Away Physics. Platformo je razvila neprofitna organizacija Away Foundation. Pripravili so tudi AwayJS, odprtokodni grafični pogon, ki uporablja WebGL. Napisan je v TypeScriptu, strogo tipizirani različici Javascripta. AwayJS je konsistenten z Away3D, tako lahko iz obstoječih modelov in animacij z istim orodjem ustvarimo HTML5 igre [8].

## 3.2 Razvoj v HTML5

### 3.2.1 Orodja za razvoj

Za razvoj aplikacij in iger v HTML5 obstaja velika množica različnih orodij. HTML5 aplikacije lahko razvijamo v običajnem urejevalniku besedil, vendar pa nam je na voljo veliko integriranih razvojnih okolij, ki nam olajšajo razvoj. Prav tako kot za razvoj v Flashu obstajajo tudi za razvoj v HTML5 orodja z grafičnim urejevalnikom. Taka orodja so znana tudi pod vzdevkom wysiwyg

- kratica za "what you see is what you get", kar pomeni, da kar vidimo, to bomo dobili. Pojavila pa so se tudi spletna orodja, ki omogočajo razvoj kar v brskalniku [30].

WebStorm in PhpStorm sta plačljivi<sup>1</sup> integrirani razvojni okolji za razvoj spletnih aplikacij. Vključujeta podporo za veliko knjižnic in ogrodij ter sistem za upravljanje z izvorno kodo. Imata še funkcijo hitre navigacije po datotekah [19, 18].

Podobno, vendar naprednejšo funkcijo navigacije ima tudi urejevalnik kode Sublime Text, s katero lahko navigiramo po datotekah in po kodi znotraj datotek [27].

NetBeans je brezplačno integrirano razvojno okolje. Vsebuje funkcionalnost predogleda v živo - sproti ob spreminjanju kode posodablja izgled v brskalniku. Podpira pa sistem za upravljanje z izvorno kodo ter ogrodje za razvoj mobilnih aplikacij Cordova [23].

Funkcijo predogleda v živo ima tudi odprtokodni urejevalnik kode Adobe Brackets, ki je v celoti napisan v tehnologijah HTML, CSS in JavaScript. Brackets podpira predprocesiranje LESS in SCSS datotek [4].

Na spletnih tehnologijah je osnovan tudi urejevalnik kode Visual Studio Code. Code vsebuje IntelliSense, tehnologijo pametnega dokončevanja, ki jo najdemo v integriranem razvojnem okolju Visual Studio. Podpira pa tudi sistem za upravljanje z izvorno kodo Git. Spletne aplikacije je možno razvijati tudi v Visual Studiu [21].

Znani urejevalniki besedil in integrirana razvojna okolja so še Eclipse, na njem osnovan Aptana Studio, brezplačni urejevalnik kode Komodo Edit ter njegova naprednejša različica, plačljivo razvojno okolje Komodo IDE.

Primeri namiznih orodij z grafičnimi urejevalniki so BlueGriffon, WYSIWYG Web Builder, Adobe Dreamweaver in Adobe Muse. Vsi omogočajo oblikovanje v grafičnem urejevalniku. Vsi razen Muse pa tudi urejanje kode. Prvi je na voljo v brezplačni in plačljivi različici, ostali so plačljivi, ponujajo pa brezplačno preizkusno obdobje.

---

<sup>1</sup>za študente brezplačni

Spletna orodja za razvoj v brskalniku so Aloha, Mercury Editor in Rendera. Prva dva sta tipa WYSIWYG, Rendera pa je urejevalnik kode.

Orodja za razvoj iger so podobna WYSIWYG urejevalnikom za razvoj aplikacij.

Construct 2 je program za ustvarjanje HTML5 iger z vizualnim urejevalnikom. Namenjen je neprogramerjem, saj lahko z načinom povleci-in-spusti grafično urejajo izgled igre. Tok igre določamo z dodajanjem dogodkov na seznam dogodkov. Dogodki so sestavljeni iz pogoja ter akcije, ki se izvede ob pogoju. Igro lahko izvozimo kot HTML5 spletno stran ali kot aplikacijo za Android, Blackberry, Tizen ali Windows. Brezplačna verzija programa dovoljuje le 100 dogodkov, štiri plasti ter dva efekta na projekt (igro), izvoz v HTML5 spletno stran. Ima še nekaj omejitev funkcionalnosti programa samega [26].

Wade je HTML5 igralni pogon s spletnim WYSIWYG urejevalnikom za 2D igre. V urejevalniku igri dodajamo scene, na katerih so objekti. Objekti so lahko gumbi, besedilo, slike oziroma drugi grafični in avdio elementi. Objektom lahko določimo obnašanje, ki označuje animacijo objekta in funkcije, ki določajo, kaj se zgodi ob uporabnikovi interakciji. Omogoča urejanje kode in je brezplačen [11].

Podjetje MightyFingers je razvilo odprtokodni spletni urejevalnik iger MightyEditor. Urejevalnik lahko uporabljamo na njihovi spletni strani, lahko pa si ga postavimo tudi lokalno. Končane igre lahko izvozimo kot spletne igre ali pa kot igre za Android. MightyEditor uporablja igralni pogon Phaser [22]. Igralni pogon Phaser uporablja še en spletni urejevalnik iger, Phaser Editor, ki je plačljiv.

Na voljo so še druga orodja za urejanje iger, kot so program Booty5 in spletni urejevalniki Ceilfire, Wimi, GDevelop in SuperPowers.

### 3.2.2 Knjižnice, ogrodja in igralni pogoni

Za razvoj aplikacij so na voljo številna ogrodja. Večinoma so to JavaScript ogrodja, ki nam olajšajo prilagajanje obnašanja aplikacije. Za razvoj iger pa

je na voljo veliko igralnih pogonov, na primer Quintus, Defold, Panda.js ter PlayCanvas. Primer ogrodja za razvoj iger je Phaser [13]. Ogrodja in igralni pogoni za igre omogočajo razvoj iger v dvodimenzionalnem ali tridimenzionalnem prostoru. Pri razvoju pa so nam na voljo tudi številne knjižnice.

Razdelili jih bomo v dve skupini. Prva skupina olajša delo pri razvoju spletnih strani. Druga skupina pa omogoča, da lahko tehnologije HTML5 uporabimo za razvoj interaktivnih aplikacij in iger, nekako nadomešča Flash.

Ogrodja v prvi skupini se delijo na dva dela: ogrodja, ki upravljajo z začeljem in ogrodja, ki upravljajo z zaledjem aplikacije. Tista, ki upravljajo z začeljem, skrbijo za čim večjo učinkovitost na odjemalčevi strani ter čim boljše uporabniško izkušnjo. Ogrodja, ki upravljajo z zaledjem, skrbijo za vso logiko aplikacije ter za podatke, s katerimi aplikacija upravlja.

Ogrodja, ki upravljajo z začeljem, so običajno napisana v Javascriptu in upravljajo s tehnologijami Javascript, HTML in CSS. Najbolj popularna ogrodja so AngularJS, Bootstrap ter Foundation.

Ogrodja, ki upravljajo z zaledjem, so napisana v različnih jezikih, na primer PHP, Javascript ali Python. Meteor in Express sta primera Javascript ogrodij, Laravel je primer ogrodja, napisanega v jeziku PHP, Django in Flask pa sta Python ogrodji.

V prvo skupino spada še knjižnica jQuery, ki poenostavi delo s HTML elementi, obravnavo dogodkov, kreiranje animacij itd.

V drugi skupini so knjižnice, ki olajšajo delo s HTML canvas elementom, omogočajo uporabo WebGL ter igralni pogoni in ogrodja.

JCanvas je Javascript knjižnica, napisana z uporabo knjižnice jQuery. Nudi programski vmesnik za enostavno manipulacijo s HTML5 canvas elementom. Oviije se okoli HTML5 canvas programskega vmesnika, tako omogoča več funkcionalnosti. Nekatere funkcionalnosti so plasti, dogodki, povleci-in-spusti funkcionalnost in animacije [14].

CreateJS je skupek knjižnic, ki jih lahko uporabimo skupaj ali posamično. EaselJS poenostavi delo s HTML5 canvas elementom. TweenJS omogoča enostavno kreiranje animacij. SoundJS omogoča delo z zvokom v HTML5.

PreloadJS pa nudi upravljanje in usklajevanje nalaganja sredstev (slikovni, video in avdio material) [12].

Phaser je odprtokodno HTML5 igralno ogrodje. Za upodabljanje uporablja Javascript knjižnico Pixi.js, ki glede na podporo brskalnika uporablja HTML5 canvas ali WebGL. Je zelo popularen, uporablja ga veliko razvijalcev iger in ima dobro podporo. Uporabljata ga že prej omenjena urejevalnika iger, MightyEditor in Phaser Editor [24].

PlayCanvas je odprtokodni igralni pogon za razvoj 3D iger in interaktivnih aplikacij, ki za upodabljanje uporablja WebGL. Ponuja platformo v oblaku, ki omogoča urejanje igre z več računalnikov hkrati ter gostovanje končanih iger [25].

Igralnih ogrodij je veliko, izbira pa se še večja, saj se razvoj novih igralnih pogonov še ni ustavil, kar nakazuje obetaven igralni pogon Quintus, ki je že v zgodnji fazi razvoja precej popularen.

## Poglavje 4

# Prehod s Flasha na HTML5

Prehod lahko izvedemo na dva načina. Pri prvem načinu z orodji obstoječo Flash aplikacijo ali igro pretvorimo v HTML5. Drugi način pa je razvoj aplikacije ali igre od začetka z uporabo tehnologij HTML5.

### 4.1 Pretvorba obstoječih Flash aplikacij

V primeru, ko želimo Flash aplikacijo prenesti na HTML5, pa je ne želimo na novo sami razviti, lahko uporabimo program ali orodje za pretvorbo, ki Flash aplikacije pretvori v HTML5.

Spletni pretvornik Swiffy je razvil Google predvsem za pretvorbo animacij in oglasov, ki so bili narejeni v Flashu. Orodje ni več na voljo od prvega julija letos, ko je Google končal s prehodom vseh oglasov na HTML5 in prenehal sprejemati spletne oglase v Flashu [9]. Swiffy je sprejel SWF datoteko in vrnil povezavo za shranitev ustvarjenih HTML5 datotek [17]. Na spletu najdemo več pretvornikov, ki so podobni Google Swiffy-ju.

Programa Adobe Flash Professional in Adobe Animate omogočata, da Flash aplikacije pretvorimo v HTML5 [5]. Oba programa sta zmožna urejanja datotek formatov AS, ASC, FLA, JSFL, XFL in XML, ne pa formata SWF. To pomeni, da moramo imeti izvirne datoteke aplikacije, če želimo izvesti pretvorbo. Če imamo na voljo samo SWF datoteke, si lahko pomagamo

s programi, ki so zmožni urejati SWF datoteke in jih pretvoriti v format FLA. Primerki takih programov so urejevalnik SWF datotek Sothink SWF Editor ter Sothink SWF Decompiler in Flash Decompiler Trillix, ki SWF datoteke prevedeta nazaj v FLA format. V programu Flash Professional lahko aplikacijo pretvorimo z orodjem Toolkit for CreateJS. Program nam na izbrano mesto shrani ustvarjene datoteke ter odpre pretvorjeno aplikacijo v brskalniku. Adobe Animate pa omogoča tudi ustvarjanje in izvoz aplikacij v tehnologiji HTML5 [3].

## 4.2 Razvoj od začetka v HTML5

Ponoven razvoj od začetka je težak, saj moramo vse narediti še enkrat. Vendar pa je hkrati priložnost za izboljšave in večje spremembe. Pomembno je, da razmislimo in si naredimo načrt, kaj želimo narediti in kako želimo, da izgleda končan izdelek.

Na začetku moramo zbrati besedila, slike, fotografije, druge grafične elemente in avdio ter video posnetke, ki jih bomo uporabili. Pri razvoju aplikacij ta korak običajno ni tako zapleten, saj lahko uporabimo navadne gumbe, že narejene grafične elemente, sami posnamemo fotografije ali pa na spletu najdemo take, ki jih lahko prosto uporabimo. Pri igrah pa običajno moramo oziroma želimo sami oblikovati vse grafične elemente igre (gumbe, napise, glavne like in ozadje).

Če razvijamo sami, je izbira razvojnega orodja odvisna od nas samih. Če pa aplikacijo ali igro razvijamo v skupini, je dobro najti razvojno okolje, ki vsem dovolj ustreza. Soglasje pri uporabi enega orodja je pomembno, če izbiramo med orodjem z grafičnim urejevalnikom in orodjem brez njega. Orodje z grafičnim urejevalnikom običajno samo generira kodo, ki določa postavitev grafičnih objektov. Če bi nekdo sam napisal to kodo, bi lahko med uporabo aplikacije prišlo do težav ali neželenega obnašanja.

Pri izbiri razvojnega orodja je potrebno upoštevati tudi naše znanje in izkušnost. Orodje Construct 2 omogoča razvoj iger ljudem, ki ne znajo pro-



gramirati. Programerji z manj izkušnjami bodo verjetno raje izbrali orodja z grafičnim urejevalnikom, saj jim tako ne bo potrebno vsega sprogramirati.

Ogrodje oziroma igralni pogon izberemo upoštevajoč zahteve, ki smo si jih zamislili. Če želimo izdelati 3D aplikacijo ali igro, moramo izbrati ogrodje ali pogon, ki to omogoča. Ker je izbire dovolj, lahko tudi pri upoštevanju vseh zahtev izbiramo med več ogrodji oziroma igralnimi pogoni in izberemo tistega, v katerem se bolj znajdemo.

Tudi, če razvijamo sami, je priporočljiva uporaba sistema za upravljanje z izvorno kodo. Pri razvoju v ekipi pa je skoraj nujna. Sistem za upravljanje z izvorno kodo omogoča, da lažje upravljamo s kodo. Najbolj uporabna funkcionalnost takega sistema pa je, da lahko v primeru, da gre kaj narobe, enostavno povrnemo stanje kode na prejšnjo, delujočo verzijo.



# Poglavje 5

## Primerjava na primeru

Različne načine prehoda smo primerjali na primeru spletnih iger za učenje glasbene teorije. Igre so bile izvirno narejene v Flashu, logika je bila napisana v ActionScriptu verzije 2.

Uporabili smo različna orodja - orodja za pretvorbo, knjižnico, ki olajša delo s HTML5 canvasom, HTML5 igralna ogrodja ter programe z vdelanim igralnim pogonom in grafičnim urejevalnikom. Tako smo zajeli vse vrste možnosti prehoda, enostavno pretvorbo s spletnim pretvornikom, pretvorbo z uradnim orodjem za razvoj Flash aplikacij, razvoj v urejevalniku kode z igralnim pogonom in brez njega ter razvoj v programu z grafičnim urejevalnikom.

Pri pretvorbi smo uporabili Google Swiffy ter preizkusnima verzijama programov Adobe Flash Professional CS6 in Adobe Animate CC.

Za razvoj od začetka smo uporabili knjižnico JCanvas, ogrodje Phaser ter urejevalnik iger Construct 2. Pri razvoju s knjižnico in igralnimi ogrodji smo uporabili integrirano razvojno okolje PhpStorm.

### 5.1 Izbrane igre

Za osnovno igro primerjave smo izbrali igro Vadnico not. Slika 5.3 prikazuje star izgled igre Vadnica not, preden smo se lotili prenosa na HTML5.

Z njo igralec vadi poimenovanje not na notnem črtovju. Igra prikazuje izbrane note, uporabnik pa mora klikniti na gumb s pravilnim imenom note. Igro smo razvili na novo z uporabo Phaserja, Construct 2 ter JCanvas ter jo pretvorili z Google Swiffy, Adobe Flash Professional CS6 in Adobe Animate CC.

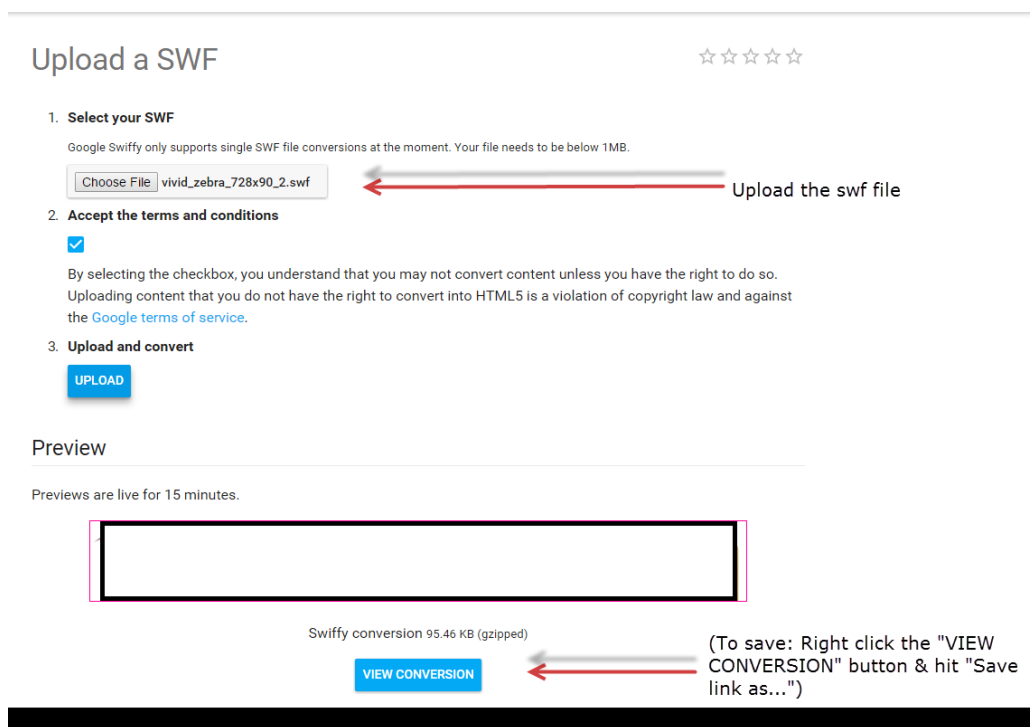
Izbrali smo še igre Vadnica intervalov, Vadnica akordov in Risanje.

Z Vadnico intervalov igralec vadi poznavanje intervalov. V igri se prikazujejo intervali, ki jih igralec izbere. Igralec mora izbrati gumb s pravilnim imenom intervala. Igro smo realizirali z uporabo Phaserja ter jo pretvorili z Adobe Flash Professional CS6.

V igri Vadnica akordov igralec vadi poimenovanje akordov. Igralec izbere akorde, ki se mu nato prikazujejo. Klikniti mora na gumb z imenom prikazanega akorda. Igro smo pretvorili z Adobe Flash Professional CS6 ter z Google Swiffy.

Igra Risanje je namenjena zabavi. Niti ni igra, temveč interaktivno platno, na katerega lahko igralec z miško riše črte. Ko klikne na gumb "Zaigraj", se predvajajo note, ki sledijo črti. Igro smo pretvorili z Google Swiffy.

Funkcionalne zahteve iger so generiranje naključnih novih primerov upoštevajoč izbrane nastavitve, ugotavljanje pravilnosti izbire, računanje uspešnosti ter predvajanje primerne zvočnega posnetka ob določenih situacijah (napačna izbira pri vadnicah in predvajanje pravih tonov pri igri Risanje). Najpomembnejši zahtevi sta zaznavanje človekove interakcije z miško (klik, premik) ter proženje animacij ob primernem času njihova ustavitve, ko je potrebno. Če ti dve zahtevi nista upoštevani, se igra obnaša kot animacija, saj se animacije prožijo samodejno, brez uporabniške interakcije ter se predvajajo v zanki brez ustavitve, igralec pa ne more uporabljati igre, saj igra ne zazna njegovih akcij. V takem primeru ostale zahteve niso pomembne, saj jih ne moremo koristiti.



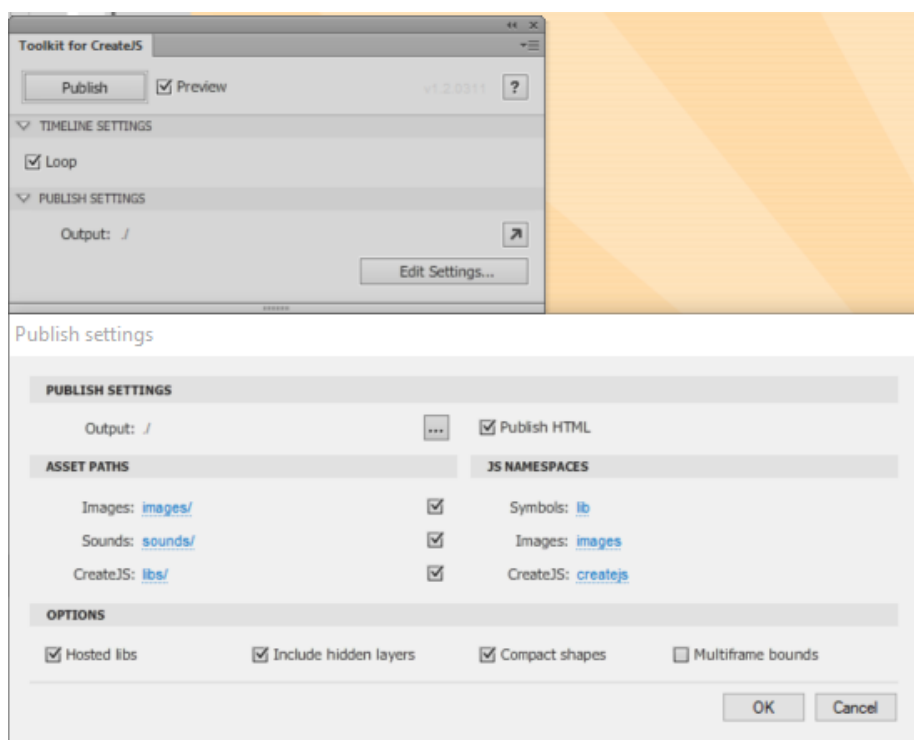
Slika 5.1: Slika prikazuje navodila za uporabo orodja Google Swiffy [20].

## 5.2 Uporabljena orodja

### Google Swiffy

Uporaba spletnega pretvornika Google Swiffy je bila enostavna. Na strani je bil gumb, ki je odprl okno za izbiro datoteke. V oknu smo izbrali datoteko in potrdili izbiro. Datoteka se je naložila in pričela se je pretvorba. Po pretvorbi se je prikazala povezava do datoteke, ki je nastala ob pretvorbi. Na povezavi je bila spletna stran, saj je Swiffy pretvoril naloženo datoteko v HTML5. Datoteko smo lahko shranili z izbiro "Save povezavo kot ..." oziroma "Shrani kot ..." v kontekstnem meniju, ki se prikaže ob kliku na desni miškin gumb. Slika 5.2 prikazuje navodila, kako uporabiti Google Swiffy.

Koda, ki jo je med pretvorbo naših iger ustvaril Google Swiffy, je dolga in zapletena, zato je težko karkoli spremeniti.



Slika 5.2: Slika prikazuje možnosti pri pretvorbi v programu Adobe Flash Professional.

## Adobe Flash Professional / Adobe Animate

V programu Adobe Flash Professional CS6 smo za pretvorbo uporabili orodje Toolkit for CreateJS. Pred pretvorbo je na voljo nekaj nastavitvev, med drugim, kam naj se shranijo pretvorjene datoteke in vključevanje skritih plasti. Pretvorjena aplikacija uporablja skupek knjižnic, zajetih v CreateJS. Slika 5.2 prikazuje možnosti pri pretvorbi v programu Adobe Flash Professional.

V programu Adobe Animate imamo enako orodje, le drugače je poimenovano. Pretvorjena aplikacija prav tako uporablja CreateJS, le novejšo verzijo.

Koda pretvorjenih iger je bolj jasna kot pri pretvorbi Google Swiffy, zato so spremembe lažje izvedljive.

## JCanvas

S knjižnico JCanvas je uporaba HTML5 elementa canvas enostavna. Sintaksa je podobna kot pri knjižnici JQuery, zato je za poznavalce knjižnice JQuery razvoj hiter. Ker nima sistema za simulacijo fizike, je primeren le za igre, kjer ne potrebujemo simulacije fizike.

Pri razvoju naših iger v JCanvasu smo naleteli na težavo le pri potrditvenih poljih in izbirnih gumbih. Ker jih HTML podpira (kot vrsto vnosnih polj) smo jih realizirali kar v HTML in ne z JCanvasom v elementu canvas. Če bi jih hoteli realizirati v elementu canvas, bi morali sami razviti algoritem, ki bi omogočal pričakovano obnašanje.

Nobena od iger ne vsebuje zahteve po simulaciji fizikalnih zakonov, zato pri uporabi knjižnice JCanvas ne potrebujemo dodatnega sistema za simulacijo fizike. Tak sistem pa bi potrebovali pri igrah, kjer igralec upravlja lik, ki se sprehaja po prostoru in lahko skače in pada.

## Phaser

Razvoj v igralnem ogrodju Phaser je razvijalcu prijazno, saj ima dobro dokumentacijo, veliko primerov ter skupnost, ki na spletnem forumu odgovarja na vprašanja. Na voljo je veliko dodatkov, ki še razširjajo funkcionalnosti igralnega pogona.

Razvoj naših iger je bil enostaven in hiter. Uporabili smo dodatek za potrditvena polja, ki smo mu dodali še funkcionalnost izbirnih gumbov. Izkoristili smo funkcijo Preload, ki preprečuje začetek igre dokler se ne naložijo vsi grafični in avdio elementi (slike, zvočni posnetki).

## Construct2

Razvijanje v programu Construct 2 je zanimivo in nezanimivo hkrati. Veliko je klikanja ali uporabe bližnjic na tipkovnici, saj program ne omogoča urejanja kode. Programerju bi bilo verjetno lažje razvijati v kakšnem drugem orodju.

Razvoj iger je bil v začetku počasen, saj je bilo kar zahtevno preklopiti v pravo paradigmo programa. Tudi tu smo morali za potrditvena polja in izbirne gumbе uporabiti vtičnik, ki smo ga morali za naše potrebe rahlo spremeniti. Brezplačna verzija programa Construct 2 ima omejitvev 100 dogodkov, tako smo se morali omejiti pri možnostih iger. Igre imajo polno funkcionalnost, le manj različnih možnosti pri nastavitvah.

## Ostali

Pri primerjavi smo uporabili še orodja z grafičnim urejevalnikom Wade, GDevelop, Mighty, igralna pogona Quintus in P5js ter knjižnico CreateJS. Razvoj v njih smo začeli in nato opustili iz različnih razlogov.

Wade, GDevelop in Mighty so po uporabniški izkušnji mešanica razvoja v programu Construct 2 in ogrodju Phaser. Elemente se ureja grafično, nato pa je potrebno napisati kodo za dogodke, ki se jih v Construct 2 izbira preko grafičnega vmesnika. Poleg tega niso tako intuitivni, zato je bil razvoj počasen in je bilo večino časa porabljenega za učenje uporabe orodij, ne pa razvoju.

Quintus in P5js sta še v zgodnji fazi razvoja, razvoj z njima pa je podoben razvoju z Phaserjem. Nimata pa še veliko uporabnikov in dodatkov kot na primer dodatek za izbirne gumbе pri Phaserju.

CreateJS je zanimiva knjižnica z naprednimi možnostmi, vendar manj intuitivna kot JCanvas.

## 5.3 Rezultati

Pretvorba po naših izkušnjah in rezultatih ni najbolj primerna za igre. Igre, pretvorjene z Google Swiffy, ne delujejo, saj se animacije nenadzorovano predvajajo in jih ni mogoče ustaviti. Igre ne zaznajo igralčevih akcij. Tudi pri igrah, pretvorjenih z Adobe Flash Professional ali Adobe Animate se animacije prožijo samodejno in se ne ustavijo. Tudi te igre ne zaznajo igralčevih akcij.



Razvoj od začetka je dal bistveno boljše rezultate. Po naših izkušnjah se je najbolje obnesel Phaser, predvsem zaradi lahke uporabe ter dobre dokumentacije in veliko dodatkov. Construct 2 je poseben, saj omogoča razvoj tudi neprogramerjem. Programerjem pa zaradi količine klikanja verjetno ni prva izbira. Prednost Construct 2 je ta, da se igre, razvite z njim, samodejno prilagajajo velikosti okna<sup>1</sup>. Pri uporabi Phaserja pa moramo za to sami dodati ukaz. JCanvas ni slab, vendar ponuja le del funkcionalnosti, ki jih ponujata Construct 2 in Phaser. Če bi želeli simulacijo fizike, bi morali uporabiti sistem za simulacijo fizike, ki pa ima svoje specifične lastnosti.

Rezultati vseh pretvorb in razvoja s posameznim orodjem so dostopni na spletu [36].

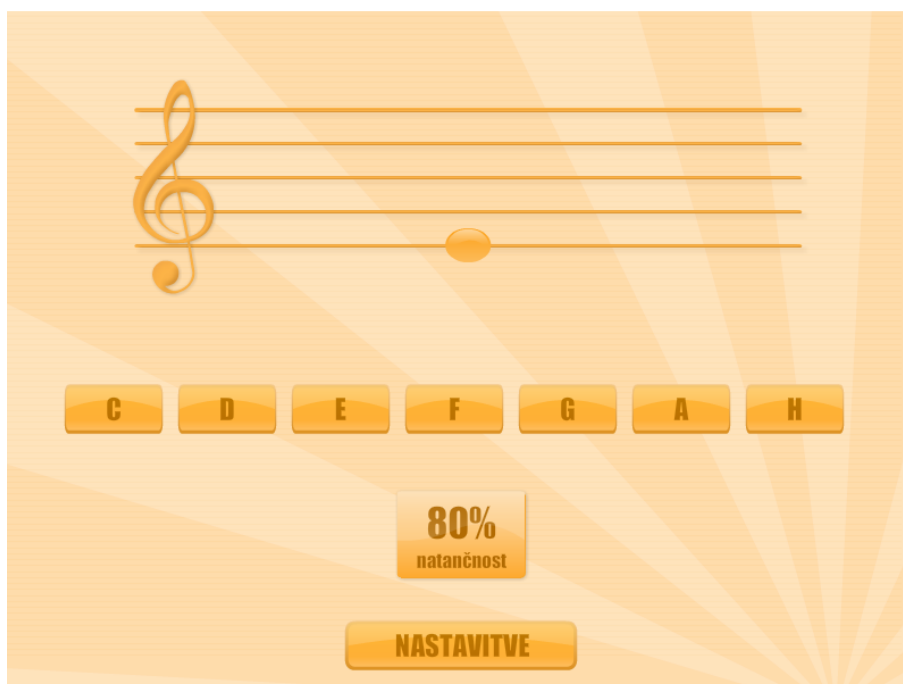
## Razvojni čas

Čas razvoja je odvisen od razsežnosti igre (kako velik je svet, koliko različnih grafičnih elementov vsebuje) ter zapletenosti logike, ki upravlja igro. Pri razvoju naših iger se čas razvoja pri uporabi posameznih orodij ni veliko razlikoval. Od začetka razvite igre niso grafično tako dovršene kot so bile prej. Nov izgled prikazuje slika 5.4. Igralec lahko izbira note oziroma intervale ali akorde, ki jih želi vaditi. V igri Vadnica not ima na voljo nekaj več možnosti kot v stari igri, to se vidi na slikah 5.5 in 5.6. Igre pa morajo upoštevati upoštevati njegovo izbiro.

V primeru igre z več grafičnimi elementi in/ali nezapleteno logiko, je čas razvoja krajši s programom Construct 2. V obratnem primeru pa je hitrejši razvoj s Phaserjem, saj nimamo omejitev pri oblikovanju algoritmov. Pri več grafičnih elementih je pri razvoju s Phaserjem vsakega deklarirati in mu določiti položaj, v programu Construct 2 pa element le postavimo na zeleno mesto. Zapleteno logiko je težje in bolj zamudno realizirati v programu Construct 2. Zato je razlika v razvojnem času v prvem primeru majhna, v drugem primeru pa lahko kar velika. Ker je logika pri igri Vadnica not dokaj zapletena, je bil razvojni čas v programu Construct 2 približno

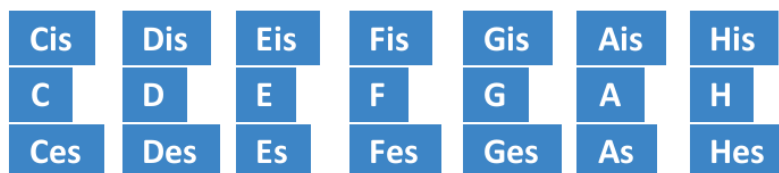
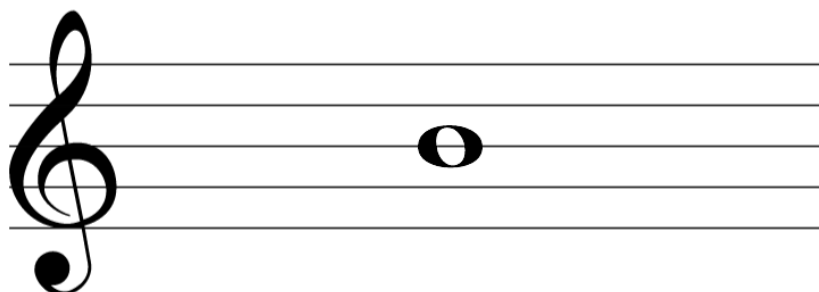
---

<sup>1</sup>Pri tem ohranjajo prvotno razmerje širine in višine



Slika 5.3: Slika prikazuje star izgled igre Vadnica not.

Vadnica not Primeri: ∞ Čas: ∞ Uspeh: 33% || ■



Slika 5.4: Slika prikazuje nov izgled igre Vadnica not.



Slika 5.5: Slika prikazuje možnosti stare igre Vadnica not.

## Nastavitve

### Izberi note

- |                                       |                              |                              |
|---------------------------------------|------------------------------|------------------------------|
| <input checked="" type="checkbox"/> C | <input type="checkbox"/> Cis | <input type="checkbox"/> Ces |
| <input checked="" type="checkbox"/> D | <input type="checkbox"/> Dis | <input type="checkbox"/> Des |
| <input checked="" type="checkbox"/> E | <input type="checkbox"/> Eis | <input type="checkbox"/> Es  |
| <input checked="" type="checkbox"/> F | <input type="checkbox"/> Fis | <input type="checkbox"/> Fes |
| <input checked="" type="checkbox"/> G | <input type="checkbox"/> Gis | <input type="checkbox"/> Ges |
| <input checked="" type="checkbox"/> A | <input type="checkbox"/> Ais | <input type="checkbox"/> As  |
| <input checked="" type="checkbox"/> H | <input type="checkbox"/> His | <input type="checkbox"/> Hes |

### Izberi ključe

- violinski
- basovski
- altovski
- sopranski
- tenorski

### Izberi način

- neomejeno
- omeji število primerov
- omeji čas

< Nazaj

Naprej >

Slika 5.6: Slika prikazuje možnosti nove igre Vadnica not.

enak času razvoja z Phaserjem, približno en teden. Pri Phaserju zaradi prve implementacije algoritma za naključno izbiranje not, pri Construct 2 pa zaradi omejitev in iskanja alternativ pri implementaciji tega algoritma. Čas razvoja z JCanvasom ni upoštevan, saj smo večino kode uporabili iz igre, realizirane s Phaserjem.

Za razvoj igre, ki bi vključevala več animacij in premikanje glavnega lika s tipkami bi razvoj s Phaserjem in v Construct 2 trajal vsaj tri tedne, odvisno od izkušenj razvijalca s posameznim orodjem. Vendar verjetno več časa s Phaserjem. Zagotovo pa bi dlje trajal razvoj z JCanvasom. Bolj točne ocene ne moremo podati zaradi premalo izkušenj z vsakim od uporabljenih orodij.

### **Končna odločitev**

Najbolj primerno se nam zdi igralno ogrodje Phaser, s katerim smo dosegli najboljše rezultate. Ima največ funkcionalnosti, fleksibilnosti ter možnosti razširitve. Zdi se bolj profesionalno in zagotavlja večji pregled nad kodo.

## Poglavje 6

### Sklepne ugotovitve

Ukinitiv vtičnika za Flash v brskalnikih je gotova. Vendar to še ne preprečuje razvoja aplikacij ali iger v Flashu oziroma v orodjih za razvoj Flash aplikacij. Ta ista orodja, ki so bila v preteklosti namenjena razvoju zgolj v Flashu, so sedaj namenjena razvoju za več platform. Primer je Adobe Animate, ki omogoča pretvorbo in izvoz Flash aplikacij v tehnologijo HTML5. Če smo poznavalci tega programa, lahko še vedno razvijamo v njem, nato pa izdelek izvozimo v željenem formatu. Težave se pojavijo le ob pretvorbi starejših Flash aplikacij, napisanih z ActionScript verzije 1 ali 2.

Težave se pojavijo tudi pri pretvorbi s spletnimi orodji za avtomatsko pretvorbo, kot je Google Swiffy. Taka orodja so res namenjena pretvorbi Flash oglasov in animacij.

Za najboljše rezultate pri prehodu s Flasha na HTML5 se je treba potruditi in izdelek ponovno razviti od začetka. Res je, da ta način vzame več časa, vendar pa je priložnost, da obstoječe aplikacije še izboljšamo. Tudi težav s podporo HTML5 funkcionalnosti v brskalnikih v veliki večini ni.

Razširjenost HTML5 je prinesla veliko orodij za razvoj, ki pa se zaradi napovedane ukinitve podpore Flasha še bolj razvijajo, kot bi se sicer. Na voljo je velika vrsta ogrodij, ki nam olajšajo razvoj z že napisanimi funkcionalnostmi in igralnimi pogoni, ki omogočajo simuliranje fizike ter enostavno izdelavo 2D in 3D iger. Urejevalniki iger, ki omogočajo grafično urejanje

iger, približajo razvoj iger neprogramerjem z željo po ustvarjanju iger. Uporabimo pa lahko tudi eno izmed mnogih integriranih razvojnih okolij, ki nam s samodokončanjem kode in možnostjo razhroščevanja prihranijo muke.

Največja težava prehoda oziroma razvoja v HTML5 ni več pomanjkanje orodij, ampak (ne)znanje in volja razvijalcev.







# Literatura

- [1] Adobe. Adobe flash platform. [https://www.adobe.com/platform/whitepapers/platform\\_overview.pdf](https://www.adobe.com/platform/whitepapers/platform_overview.pdf). [Dostopno na spletu; obiskano 24. 8. 2016].
- [2] Adobe. Adobe gaming sdk. <https://www.adobe.com/products/gaming-sdk.html>. [Dostopno na spletu, obiskano 18. 8. 2016].
- [3] Adobe. Animate cc vs. flash pro cs6. <http://www.adobe.com/products/animate/versions.html>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [4] Adobe. Brackets. <http://brackets.io/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [5] Adobe. Create html5 canvas documents in animate cc. <https://helpx.adobe.com/animate/using/creating-publishing-html5-canvas-document.html>. [Dostopno na spletu; obiskano 31. 3. 2016].
- [6] Adobe. Gaming engines. [http://www.adobe.com/devnet/games/gaming\\_engines.html](http://www.adobe.com/devnet/games/gaming_engines.html). [Dostopno na spletu; obiskano 31. 3. 2016].
- [7] Apache. Apache flex®. <http://flex.apache.org/>. [Dostopno na spletu; obiskano 31. 3. 2016].
- [8] Away3D. Away3d. <http://away3d.com/>. [Dostopno na spletu; obiskano 18. 8. 2016].

- 
- [9] Google Ads Developer Blog. Swiffy. <http://googleadsdeveloper.blogspot.si/2016/06/sunset-of-googles-swiffy-tool.html>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [10] Chrome. Npapi deprecation. <https://sites.google.com/a/chromium.org/dev/developers/npapi-deprecation>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [11] Clockworkchilli. Wade. <http://clockworkchilli.com/wade/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [12] CreateJS. Createjs. <http://createjs.com/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [13] HTML5 Game Development. Html5 engines. <http://html5gamedevelopment.com/category/html5-engines/>. [Dostopno na spletu; obiskano 9. 4. 2016].
- [14] Caleb Evans. jcanvas. <http://projects.calebevans.me/jcanvas/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [15] Firefox. Npapi plugins. <https://blog.mozilla.org/futurereleases/2015/10/08/npapi-plugins-in-firefox/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [16] Gamua. Starling. <http://gamua.com/starling/>. [Dostopno na spletu; obiskano 31. 3. 2016].
- [17] Google. Swiffy. <https://developers.google.com/swiffy/>. [Dostopno na spletu; obiskano 9. 4. 2016].
- [18] JetBrains. PhpStorm. <https://www.jetbrains.com/phpstorm/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [19] JetBrains. Webstorm. <https://www.jetbrains.com/webstorm/>. [Dostopno na spletu; obiskano 20. 8. 2016].

- [20] Ad-Tag Macros. How to use google swiffy to convert flash into html5. <http://adtagmacros.blogspot.si/2015/09/how-to-use-google-swiffy-to-convert-flash-into-html5.html>. [Dostopno na spletu; obiskano 24. 8. 2016].
- [21] Microsoft. Visual studio code. <https://code.visualstudio.com/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [22] MightyFingers. Mightyfingers. <http://mightyfingers.com/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [23] NetBeans. Netbeans. <https://netbeans.org/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [24] PhotonStorm. Phaser. <http://phaser.io/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [25] PlayCanvas. Playcanvas. <https://playcanvas.com/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [26] Scirra. Construct 2. <https://www.scirra.com/construct2>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [27] Sublime Text. Sublime text. <https://www.sublimetext.com/>. [Dostopno na spletu; obiskano 20. 8. 2016].
- [28] TutorialFeed. 10 open source flash frameworks. <http://www.tutorialfeed.org/2010/05/10-open-source-flash-framework-for.html>. [Dostopno na spletu; obiskano 18. 8. 2016].
- [29] USC. Html5 features. <http://www-scf.usc.edu/~chenemil/itp104/webtech.html>. [Dostopno na spletu; obiskano 24. 8. 2016].
- [30] Wikipedia. Comparison of html editors. [https://en.wikipedia.org/w/index.php?title=Comparison\\_of\\_HTML\\_editors&oldid=729990619](https://en.wikipedia.org/w/index.php?title=Comparison_of_HTML_editors&oldid=729990619). [Dostopno na spletu; obiskano 23. 8. 2016].

- [31] Wikipedia. Adobe Flash. <http://en.wikipedia.org/w/index.php?title=Adobe%20Flash&oldid=710317143>, 2016. [Dostopno na spletu; obiskano 23. 3. 2016].
- [32] Wikipedia. Cascading Style Sheets. <http://en.wikipedia.org/w/index.php?title=Cascading%20Style%20Sheets&oldid=712557847>, 2016. [Dostopno na spletu; obiskano 30. 3. 2016].
- [33] Wikipedia. HTML. <http://en.wikipedia.org/w/index.php?title=HTML&oldid=711874632>, 2016. [Dostopno na spletu; obiskano 29. 3. 2016].
- [34] Wikipedia. HTML5. <http://en.wikipedia.org/w/index.php?title=HTML5&oldid=709499105>, 2016. [Dostopno na spletu; obiskano 23. 3. 2016].
- [35] Wikipedia. JavaScript. <http://en.wikipedia.org/w/index.php?title=JavaScript&oldid=712202340>, 2016. [Dostopno na spletu; obiskano 30. 3. 2016].
- [36] Žiga Černigoj. Diploma. <https://zigacernigoj.github.io/diploma/>. [Dostopno na spletu; obiskano 25. 8. 2016].