

**SVEUČILIŠTE U SPLITU  
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I  
BRODOGRADNJE**

**SEMINARSKI RAD IZ RAČUNALNE GRAFIKE**

**STENCYL**

**Ante Tokić**

Split, siječanj 2021.

## Sadržaj:

1	Uvod.....	1
2	STENCYL.....	2
2.1	Značajke.....	2
2.2	Razvojno okruženje .....	2
2.3	Vizualni programski jezik .....	3
2.4	„Pohrana u oblaku“ .....	3
2.5	Integracija i proširenja .....	4
3	Flappy Bird igra .....	5
3.1	Originalna verzija igre Flappy Bird.....	5
3.2	Ideja i namjena.....	5
4	Implementacija igre.....	7
4.1	Okruženje.....	7
4.2	Stvaranje projekta .....	8
4.3	Resursi igre .....	10
4.3.1	Kreiranje događaja za pticu.....	11
4.3.2	Kreiranje događaja za prepreke.....	14
4.3.3	Kreiranje događaja za bodovanje prepreka .....	14
4.4	Scene.....	15
4.4.1	Prva scena.....	15
4.4.2	Druga scena .....	15
4.5	Konačni rezultat.....	16
5	Literatura .....	18

# **1 Uvod**

Tema seminarskog rada u kolegiju Računalna grafika je Stencyl, alat koji omogućuje stvaranje video igara. Na početku ćemo objasniti detaljnije što je to Stencyl, navesti njegove najvažnije značajke, vidjeti koje su to platforme na kojima se Stencyl pokreće, te upoznati Stencyl razvojno okruženje. Također rad bi trebao predstaviti i jednostavni primjer video igrice stvorene pomoću Stencyl-a. Zašto je odabrana baš ta igra te koji je cilj implementirane igre bit će opisano u jednom od poglavlja ovog seminarskog rada.

## 2 STENCIL

Stencyl je alat za razvoj video igara koji korisnicima omogućuje stvaranje 2D video igara za računala, mobilne uređaje i web. Softver se izvorno zvao "StencylWorks" dok je bio u razvoju i za početno izdanje, ali je kasnije skraćen na samo "Stencyl". Softver je dostupan besplatno, a odabrane mogućnosti objavljivanja dostupne su za kupnju.

### 2.1 Značajke

Igre stvorene u Stencyl-u mogu se izvesti na web-u kao HTML5 ili putem Adobe Flash Player-a, na osobnim računalima kao izvršne igre, te na raznim mobilnim uređajima kao iOS i Android aplikacije. Fizikom i sudarima upravlja Box2D, koji se može selektivno ili potpuno onemogućiti kako bi se smanjio potencijalni utjecaj na performanse igara koje ne zahtijevaju potpunu simulaciju fizike. Počevši od verzije 3.0, projekti u Stencyl-u koriste programski jezik Haxe i okvir igre OpenFL kako bi omogućili fleksibilan stil pisanja i pokretanja bilo gdje.

### 2.2 Razvojno okruženje

Stencyl je autorski alat i integrirano razvojno okruženje (engl. IDE - Integrated development environment). Aplikacija uključuje nekoliko modula koji se koriste za izvršavanje potrebnih zadataka za stvaranje igara sa softverom.

- Uređivač ponašanja koristi se za stvaranje i uređivanje koda i logike igre u modularnim dijelovima poznatim kao ponašanja i događaji.
- Uređivač glumca koristi se za stvaranje i uređivanje entiteta igara (Glumci) i njihovih postavki, uključujući ponašanje, fiziku i animacije.
- Dizajner scena koristi se za stvaranje i uređivanje razina i stanja igara (Scene) pomoću glumaca, skupova pločica i ponašanja.
- Uređivač pločica (Tileset) koristi se za uvoz i uređivanje skupova pločica, uključujući oblike sudara, izgled i animacije.

Dodatni alati omogućuju korisniku uvoz slika za upotrebu pozadina i pozadine u scenama, uvoz i uređivanje fontova, uvoz zvukova i glazbenih datoteka (podržani su MP3 i OGG, ovisno o cilju izvoza) i promjenu postavki igre kao što su kontrole igrača i rezolucija igre. Biblioteka uobičajenih ponašanja uključena je u Stencyl kako bi se smanjila potreba za

ponovnim stvaranjem uobičajenih ponašanja u igrama, a nekoliko "kompleta" igara pruža funkcionalna polazišta za uobičajene 2D žanrove igara.

### **2.3 Vizualni programski jezik**

Tijekom kreiranja novog ponašanja, predstavljena je opcija za njegovo stvaranje u načinu koda ili načinu dizajna. Korištenje načina rada koda za ponašanje omogućuje korisniku programiranje logike u tradicionalnom tekstualnom obliku i po želji otvaranje koda u vanjskom uređivaču kao što je Atom, Visual Studio Code ili neki drugi uređivač.

Alternativno, Dizajn način rada je grafičko sučelje koje korisnicima omogućuje stvaranje modularne logike igre za glumce i scene pomoću vizualnog programskog jezika (engl. Visual programming language, VLM). Koncept dizajnerskog načina rada kao oblika razvoja krajnjeg korisnika nastao je u MIT-ovom okruženju za učenje računalnog jezika Scratch, a korišten je uz dopuštenje za Stencyl.

Budući da je riječ o vizualnom programskom jeziku, korisnici načina dizajna ne moraju učiti ili upisivati određeni programski jezik, niti se moraju baviti sintaksom. Umjesto toga, dostupne se akcije povlače i ispuštaju iz palete „blokova koda“. Ti će se blokovi međusobno spojiti i ugnijezditi jedan u drugoga, što omogućuje stvaranje napredne logike od osnovnih komponenata. Da bi se izbjegle pogreške u sintaksi tijekom kompilacije, neće se svi blokovi međusobno spojiti. Na primjer, razmak koji zahtijeva logičku vrijednost neće prihvatiti blok koji predstavlja numeričku vrijednost. Oblici vrsta blokova različiti su kako bi pomogli da se to korisniku prikaže kao ograničenje oblikovanja ponašanja.

### **2.4 „Pohrana u oblaku“**

StencylForge je integrirana internetska usluga za pohranu i dijeljenje Stencyl igara i sredstava za igre. Može se koristiti za izradu sigurnosnih kopija projekata i pristup projektima s drugih računala kada ste prijavljeni u Stencyl. Sadržaj koji su prenijeli korisnici prema zadanim je postavkama privatn, ali također se može učiniti javnim za dijeljenje s drugim korisnicima. Osim sadržaja koji su prenijeli korisnici, na StencylForge-u su također dostupni službeno sankcionirani sadržaji kao što su primjeri igara, kompleti igara, ponašanja i slikovna ili zvučna sredstva.

## 2.5 Integracija i proširenja

Integriran sa Stencil-om je Pencil, uređivač slika izveden iz JHLabs Image Editor-a. Njegova upotreba u Stencil-u obično je sekundarna, jer se slike obično uvoze izravno s računala nakon što se odvojeno kreiraju. Stencil se alternativno može postaviti da koristi vanjske uređivače slika, kao što su Photoshop i GIMP, za izmjenu slika koje su već učitane u projekt.

Stencil također podržava korisnička proširenja za dodavanje funkcija softveru. Na primjer, proširenje bi moglo dodati nove blokove za upotrebu u načinu dizajna, poput API-ja treće strane. Drugi je primjer opsežni alat za skriptiranje dijaloga za jednostavno dodavanje dijaloga i podešavanje njegovih postavki za igru.

### 3 Flappy Bird igra

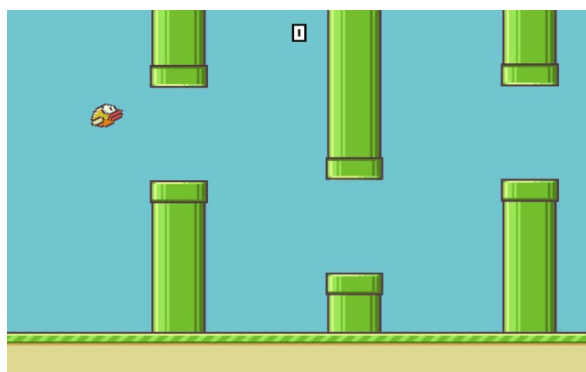
Ovaj seminar prati izradu jednostavne 2D igre u Stencyl-u koja će nalikovati na popularnu igru „Flappy Bird“.

#### 3.1 Originalna verzija igre Flappy Bird

Flappy Bird (Slika 1) je mobilna igra koju je razvio vijetnamski programer video igara Dong Nguyen pod njegovom tvrtkom za razvoj igara dotGears. Igra je napravljena kao bočni klizač gdje igrač kontrolira pticu, pokušavajući letjeti između cijevi, a da ih ne pogodi. Prilikom uspješnog prolaska između cijevi, povećava se rezultat.

Nguyen je igru stvarao tijekom nekoliko dana, koristeći protagonista pticu kojeg je dizajnirao za otkazanu igru 2012. godine. Igra je objavljena u svibnju 2013. godine, ali je naglo porasla popularnost početkom 2014. Krajem siječnja 2014. bila je to najviše preuzimana besplatna igra u App Store-u za iOS. U tom je razdoblju njegov programer rekao da Flappy Bird zarađuje 50.000 USD dnevno od reklama u aplikacijama, kao i od prodaje.

Tvorac Flappy Bird-a uklonio je igru iz App Store-a i Google Play-a 10. veljače 2014. Tvrdi da je osjećao krivnju zbog, kako je smatrao, ovisnosti i pretjerane upotrebe. Popularnost igre i naglo uklanjanje doveli su do toga da se mobilni uređaji s njezinim unaprijed instaliranim sustavom prodaju po visokim cijenama putem Interneta.



*Slika 1 – Originalna igra Flappy Bird*

#### 3.2 Ideja i namjena

Ideja seminara je izrada 2D igre koja će biti slična igri Flappy Bird. Koncept igre će ostati isti, ali vizualno će se razlikovati od izvorne igre. Za igranje igre potreban je samo miš,

odnosno lijeva tipka miša čiji klik utječe na promjenu kretanja ptice. Nakon što je igra završila, također mišem odabiremo ponovni pokušaj igranja.

Namjena je korisniku omogućiti prikaz bodova tijekom igre, kako bi pokrenuo želju za što većim rezultatom, a time potaknuo igrača da što više vremena nadmašiti prethodni rezultat.

Kad igra završi, potrebno je prikazati prozor sa konačnim rezultatom igre, te opciju za ponovni pokušaj igre.



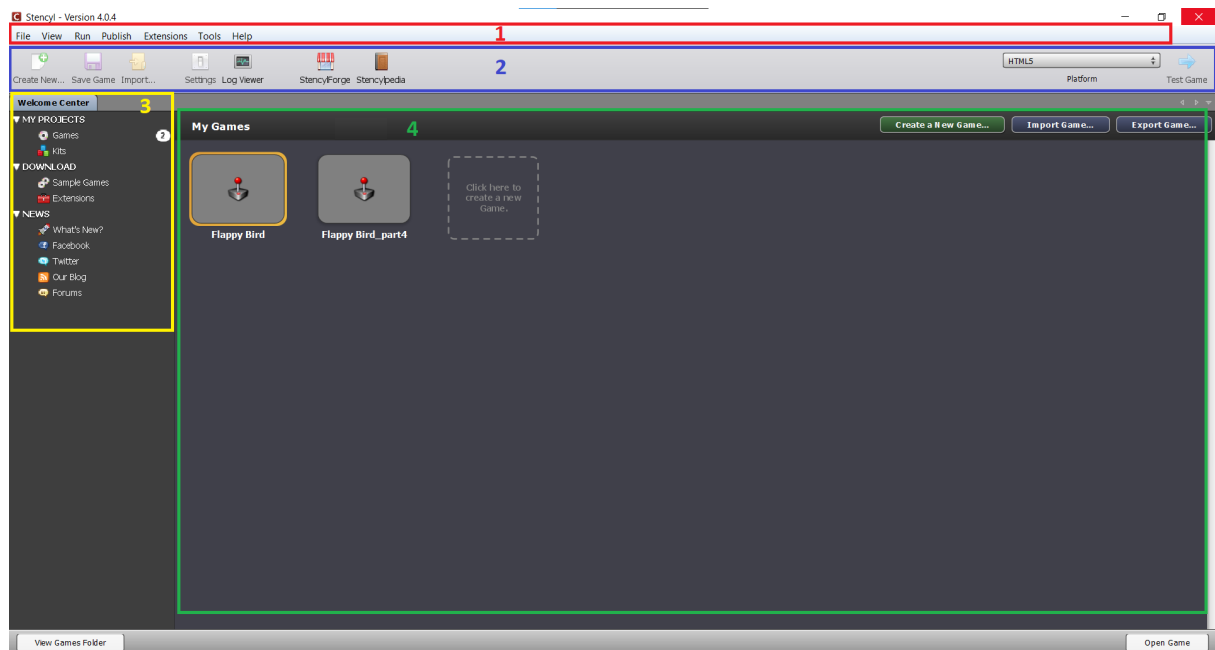
## 4 Implementacija igre

U ovom poglavlju objasniti ćemo razvojno okruženje, njegove najvažnije komponente te nakon toga kreirati novu igru i objasniti logiku same igre.

### 4.1 Okruženje

Stencyl je alat kojeg je moguće instalirati na različite operativne sustave, a nakon pokretanja aplikacije otvara se sučelje (Slika 2) koje se dijeli na sljedeće komponente:

- 1) Glavna navigacija – sadrži razne opcije poput otvaranja projekta, mijenjanja postavki editora, dodavanja proširenja, generiranja zapisa o igri, te objavljivanje igre u raznim formatima
- 2) Alatna traka – sadrži razne opcije poput kreiranja igre, spremanja igre, uvoza te mijenjanja pojedinih postavki unutar igre (primjerice kreiranja varijabli te mijenjanja tipki). Jedna od najvažnijih stvari alatne trake je tipka za testiranje igre. Također u alatnoj traci je moguće promijeniti platformu na kojoj će se izvoditi testna verzija igre. U padajućem izborniku biramo jednu od opcija poput HTML5, Flash Playera, Androida ili Windows sustava
- 3) Sidebar navigacija – padajući izbornik u kojem je moguće odabrati naše projekte, preuzeti predloške gotovih igara, te odabrati novosti o Stencyl-u na nekim od društvenih mreža ili foruma.
- 4) Kolekcija igara – sadrži popis svih dosadašnjih igara koje smo kreirali, te sadrži tipku za kreiranje nove igre, kao i tipke za uvoz i izvoz igre

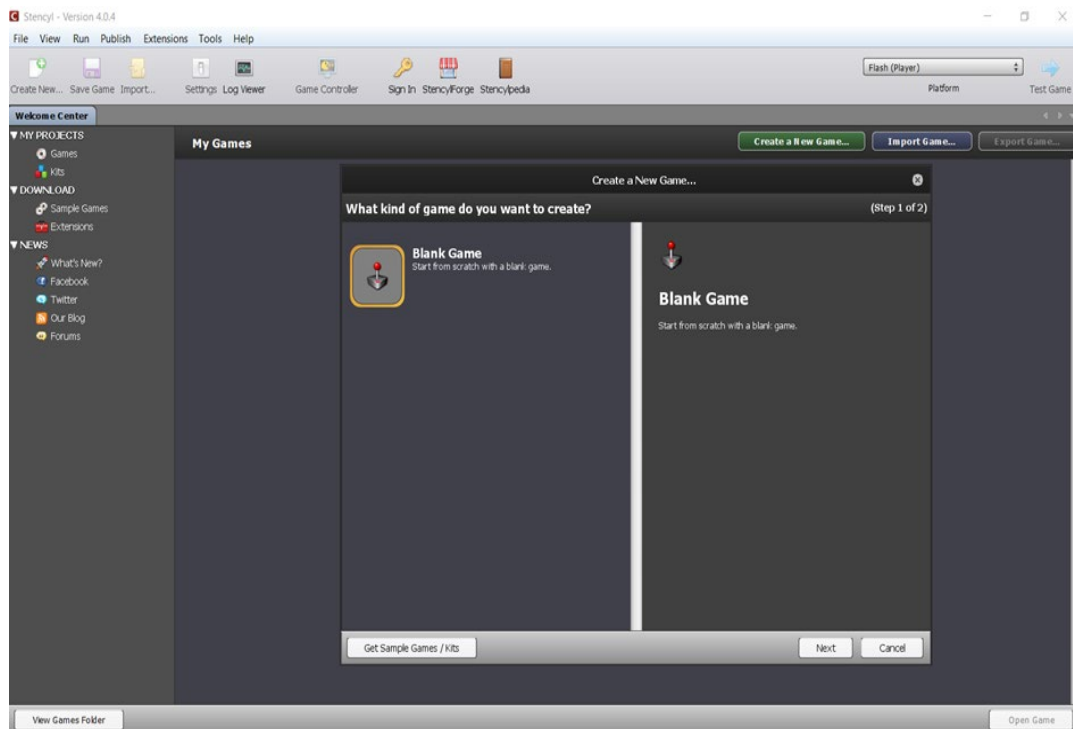


*Slika 2 – Podjela razvojnog okruženja na komponente*

## 4.2 Stvaranje projekta

Osim što je moguće koristiti gotovi predložak igre, moguće je uvesti već gotove igre sa lokalnog računala ukoliko nisu prikazane u odjeljku „My Games“ pritiskom na tipku „Import Game...“, izvesti kreirane igre u *.stencyl* formatu pritiskom na tipku „Export Game“ ili kreirati novu igru od početka pritiskom na tipku „Create a New Game“.

Nakon što smo odabrali kreiranje nove igre, javlja se novi prozor (Slika 4) u kojem imamo opciju kreirati praznu igru ili dohvatiti s Interneta neke unaprijed kreirane igre koje je moguće promijeniti i prilagoditi našim potrebama.



*Slika 3 – Prozor za stvaranje nove igre*

Nakon što smo odabrali kreiranje prazne igre, javlja se još jedan prozor (Slika 5). Tu unosimo naziv igre, te širinu i visinu ekrana na kojem će se prikazivati igra. Nakon unosa podataka, potvrdimo kreiranje igre tipkom „Create“.

*Slika 4 – Prozor za unos informacija o novoj igri*

### 4.3 Resursi igre

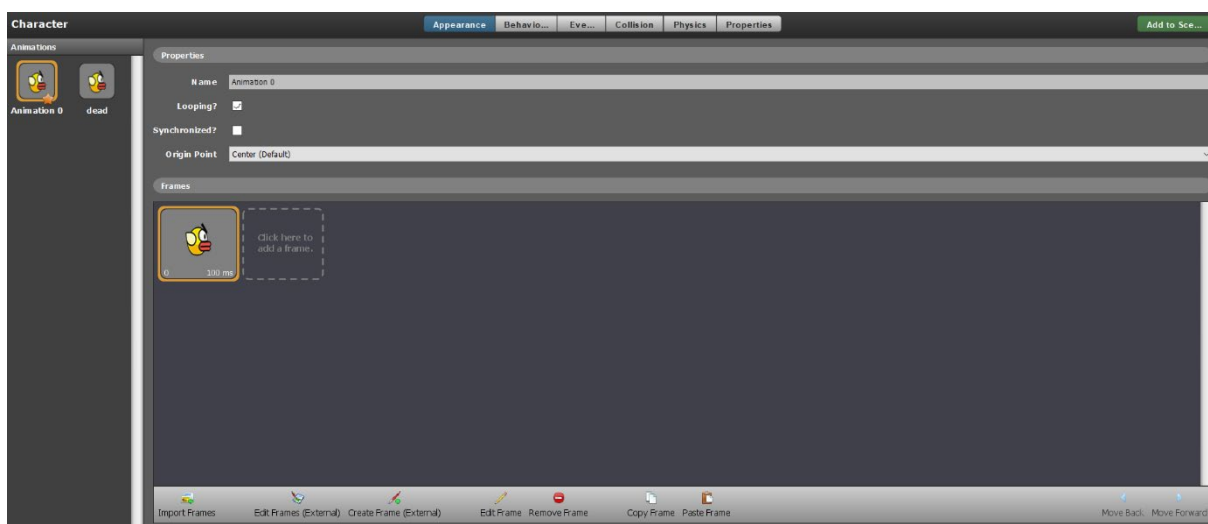
Nakon što smo uspješno stvorili praznu igru, kreira se sučelje u kojem imamo „Dashboard“ ili nadzornu ploču, koja je ujedno i početna stranica naše igre. Sama igra podijeljena je na scene, a scene su sastavljene od glumaca, pozadina, fontova, zvukova i pločica. Stvoreni resursi dostupni su u svim scenama.

Unutar odjeljka resursi u sidebar navigaciji odabiremo glumce, te kreiranje objekte koje ćemo koristiti. Za igru Flappy Bird to će biti ptica koja će prolaziti kroz prepreke, zatim prepreke i prostor između prepreka koji služi kao senzor i brojač uspješnih prolazaka. Osim toga kreirat ćemo 2 tipke koje će se prikazati tijekom pokretanja igre ( Play) te tipka za novi pokušaj igranja (Retry), Još jedan važan objekt je prozor s prikazom rezultata na kraju igre.

Prilikom kreiranja glumca javlja se sučelje (Slika 5) u kojem imamo navigaciju sa sljedećim komponentama:

- Izgled – vizualne postavke glumca u kojima je moguće dodavati animacije, uređivati pojedine okvira unutar Stencyl-a ili pomoću vanjskog uređivača, dodavati animacije te mijenjati svojstva animacija

- Ponašanje – uređivanje ponašanja glumca kako bi igra bila što realnija. U Stencyl-u postoje već kreirana ponašanja koja je moguće prilagođavati našim potrebama.
- Događaji –gradivni blokovi ponašanja. Ukratko, to su stvari koje se događaju u vašoj igri koje mogu pokrenuti neku vrstu radnje ili odgovora. (Primjer događaja u igri je klik mišem, a odgovor na to je pomicanje ptice prema gore.)
- Sudari - budući da Stencyl pokreće Box2D, sudari se događaju automatski, kao da svaki objekt postoji u „stvarnom životu“. Za zadane slučajeve nije potrebna nikakva radnja da bi došlo do sudara.
- Fizika – odjeljak u kojem je moguće promijeniti utjecaj gravitacije na objekt, materijal od kojeg je izgrađen objekt, težinu objekta te brojne druge fizičke vrijednosti
- Svojstva – odjeljak u kojem mijenjamo naziv objekta, njegov opis i grupu kojoj pripada



*Slika 5 – Kreiranje objekta*

Za potrebe naše igre, slike su kreirane u Adobe-ovom alatu „Adobe Photoshop 2021“, dok su zvučni efekte preuzeti sa web stranice <https://freesound.org/>. Kasnije su slike uvezene kao okviri za objekte, dok su zvukovi uvezeni kao .mp3 i .ogg dokumenti.

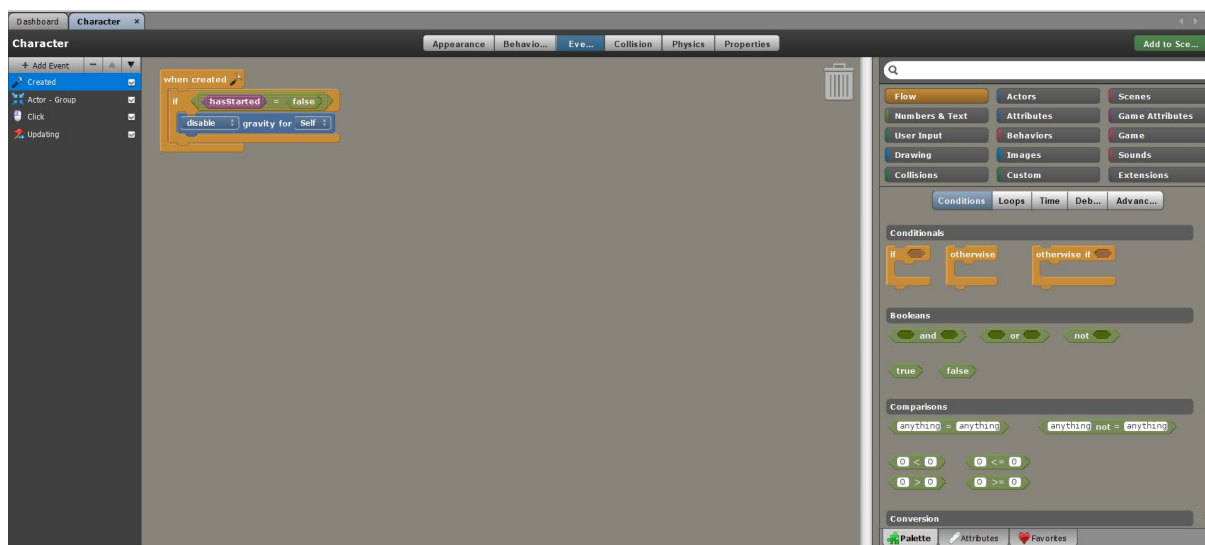
Nakon što smo objasnili stvaranje svih glumaca, sada ćemo objasniti događaje za pojedine glumce.

#### *4.3.1 Kreiranje događaja za pticu*

Kada odaberemo opciju Događaji za pojedini objekt, otvara se sučelje u kojem s lijeve strane imamo izbornik za dodavanje događaja. Također je moguće mijenjati raspored

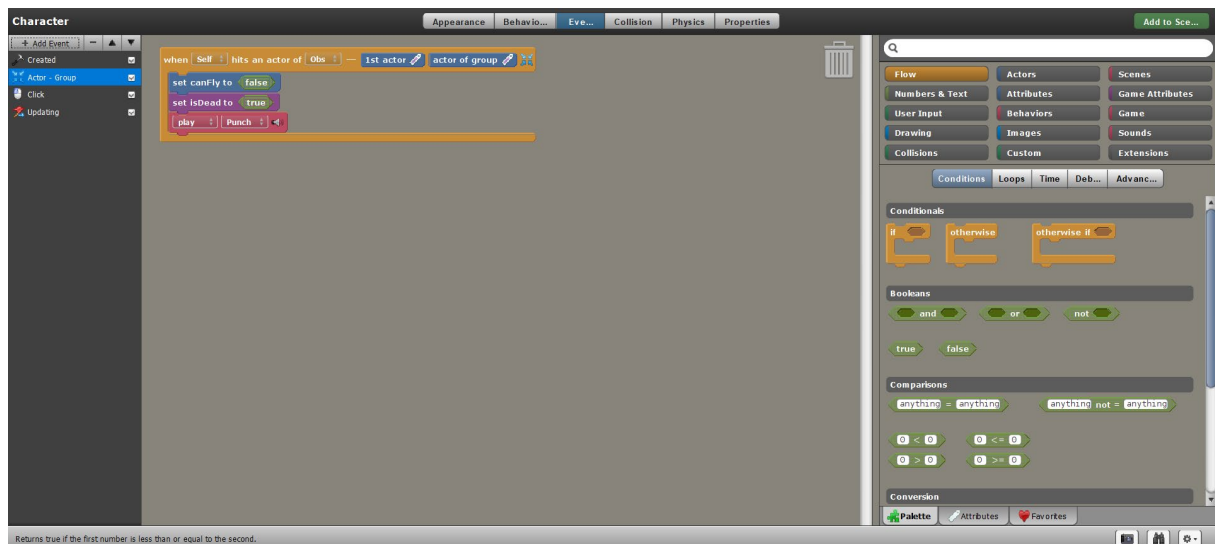
događaja, duplicirati ih, brisati i uređivati. S desne strane sučelje imamo tražilicu, te kategorije u kojima su blokovi koda. Nije potrebno pisati kod, nego samo implementirati programsku logiku pomoću blokova.

Na slici 6 je implementirana logika prilikom kreiranja objekta ptice. Imamo boolean vrijednost nazvanu `hasStarted` koja provjerava je li igra započela. Ako igra nije započela, ptica stoji bez utjecaja gravitacije, kako bi omogućila korisniku da se pripremi za igru, odnosno da ptica ne padne dok nije započeo igru.



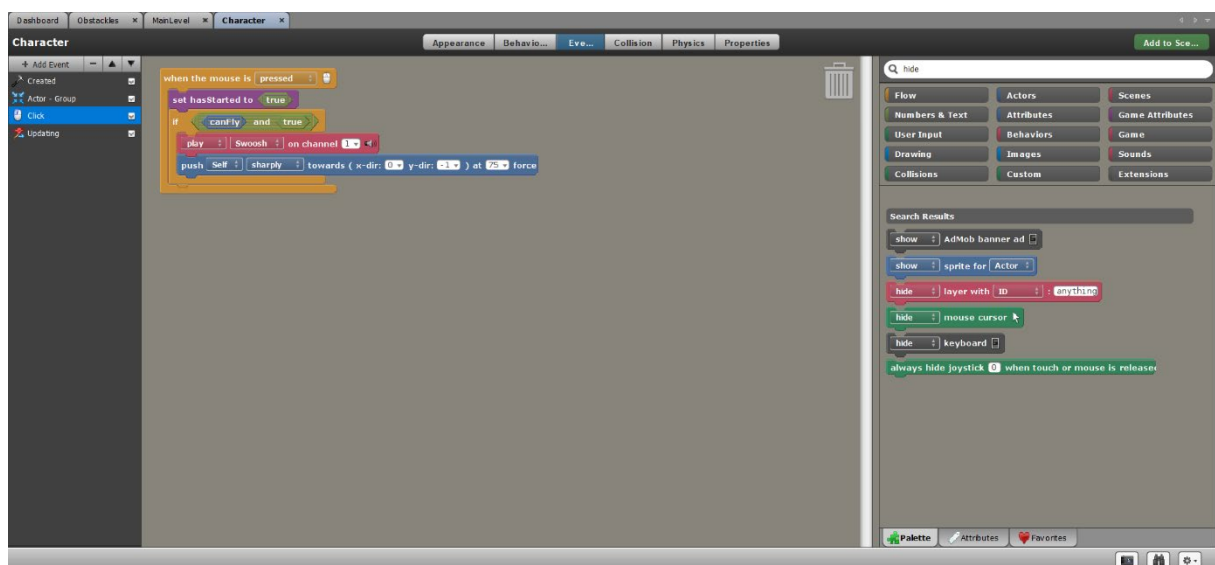
*Slika 6 – Događaj Created*

Nadalje, kreiramo događaj Actor – Group u kojem je implementirano ponašanje ptice prilikom udarca u prepreke (Slika 7). Prepreke su grupirane zajedno u grupu i ukoliko ptica udari u jednu od prepreka, postavlja se boolean varijabla `canFly` na vrijednost neistina, dok se vrijednost boolean varijable `isDead` mijenja na vrijednost istina. Na kraju pokreće se zvuk udarca iz resursa zvuk, kako bi stvorili bolji doživljaj igre.



Slika 7 – Događaj Actor – Group

Sljedeće što implementiramo je ponašanje nakon što pritisnemo tipku miša (Slika 8). Nakon što je pritisnuta tipka miša, mijenja se vrijednost varijable hasStarted na vrijednost istina. Zatim postavljamo zvuk prilikom svakog klika i funkcijom push mijenjamo koordinate ptice određenom silom.



Slika 8 – Događaj Click

Sljedeći događaj se naziva Updating u kojem je izvršena provjera varijable hasStarted. Ako je igra započela, varijabla će se promijeniti iznos, a time će se i promijeniti gravitacijsko djelovanje, odnosno ptica će početi naglo padati ukoliko nismo pritisnuli tipku.

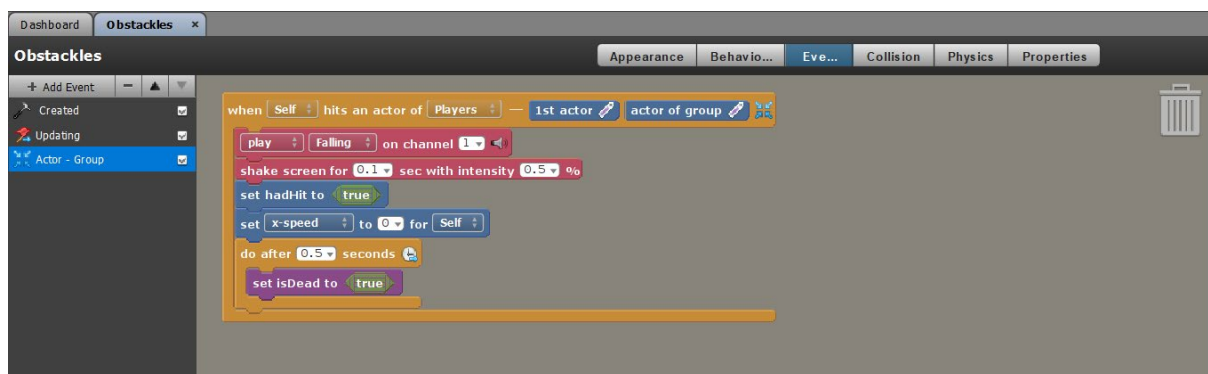
### 4.3.2 Kreiranje događaja za prepreke

Prilikom kreiranja objekta prepreke, kreirali smo 6 različitih vrsta prepreka, kako igra ne bi bila monotona, te kako bi bilo teže igrati.

Prvo kreiramo događaj koji smo nazvali Created. Tu smo napravili funkciju unutar koje smo postavili slučajni odabir animacija, kako bi prepreka bila različita svaki put.

Nadalje, imamo događaj Updating kod kojeg se prilikom udarca ptice u prepreku, brzina kretanja prepreke postavlja na vrijednost 0. U suprotnom, ukoliko je ptica živa, brzina se mijenja pomoću varijable Obstacklespeed.

Zadnji događaj koji smo kreirali za prepreke je prikazan na slici 9. Ukoliko ptica udari u jednu prepreku, pokreće se zvuk udarca i ekran se počinje tresti blagim intenzitetom. Varijabla hadHit se postavlja na vrijednost istina, a brzina se postavlja na 0. Nakon odgode vremena, kako bi pad bio realističniji, vrijednost isDead se postavlja na vrijednost istina, te se igra prekida.



Slika 9 – Događaj Actor – Group

### 4.3.3 Kreiranje događaja za bodovanje prepreka

Najvažniji događaj koji kreiramo kod objekta nazvanog ObstacklesScore je događaj prikazan na slici 10. Služi kao broj rezultata, odnosno svakim prolaskom između prepreka, rezultat se povećava za 1.





Slika 10 – Događaj Actor – Group za ObstaclesScore

## 4.4 Scene

Igra se sastoji od 2 scene. Prva scene je uvodna scena, dok se u drugoj sceni odvija igranje igre.

### 4.4.1 Prva scena

Prva scena kao što smo rekli je uvodna scena i nazvana je MainMenu. U njoj imamo prikaz pozadine sa tipkom Play kojom se pokreće igra. Prilikom kreiranja scene, pali se glazba u pozadini.

### 4.4.2 Druga scena

Druga scena je glavna scena i nazvana je MainLevel. Pritiskom tipke Play u prethodnoj sceni, pokreće se druga scena. Kreira se scena u kojoj se javlja ptica koja miruje. Klikom miša, igra započinje, te se javljaju prepreke koje je potrebno zaobilaziti i time skupljati bodove.

Kod druge scene kreiramo prvo događaj Created u kojem se postavlja brzina kretanja prepreka, te boolean varijable hasStarted na vrijednost neistina i varijabla isDead na vrijednost neistina. Rezultat se postavlja na 0. Ukoliko je igra započela, odnosno detektiran je klik mišem, stvaraju se zapreke na ekranu.

Sljedeći događaj koji imamo je Drawing u kojem se nalazi samo ispis trenutnog rezultata u gornjem lijevom kutu, te ukoliko je igra gotova ispisuje se u prozoru koji se sastoji upravo od tog rezultata i tipke za novi pokušaj.

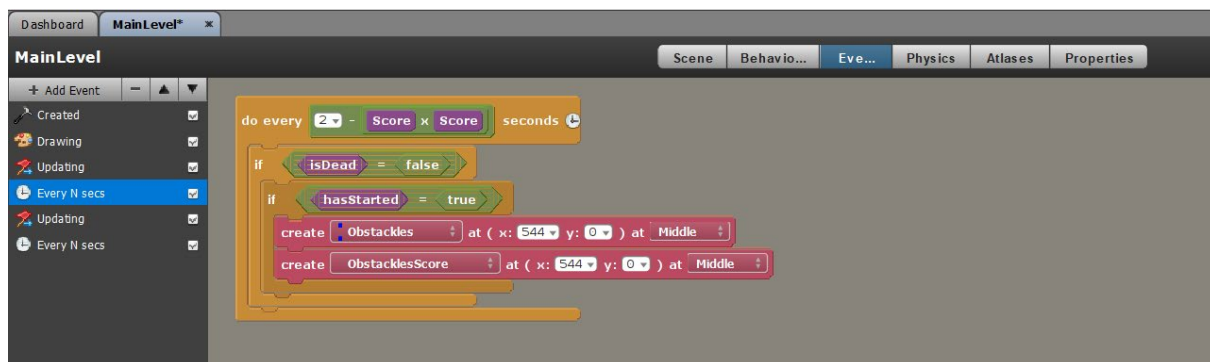
Kod događaja Updating prikazanog na slici 11 imamo ponašanje ukoliko objekt dodirne dno ekrana i time se varijabla isDead postavlja na vrijednost točno te se skriva sloj sa preprekama

i poziva se prikaz prozora s rezultatom i tipkom za ponovnu igru. Ukoliko je pritisnuta tipka Retry, scena se ponovo učitava sa efektom koji traje 1 sekundu.



Slika 11 – Događaj Updating

Posljednje ponašanje igre je prikazano na slici 12, odnosno kreiranje prepreka različitom brzinom koja se mijenja ovisno o trenutnom rezultatu. Na početku igre je rezultat 0 pa se prepreka kreira nakon 2 sekunde. Osim prepreke kreira se i prostor između prepreke, koji služi kao senzor prolaska, odnosno brojač rezultata.

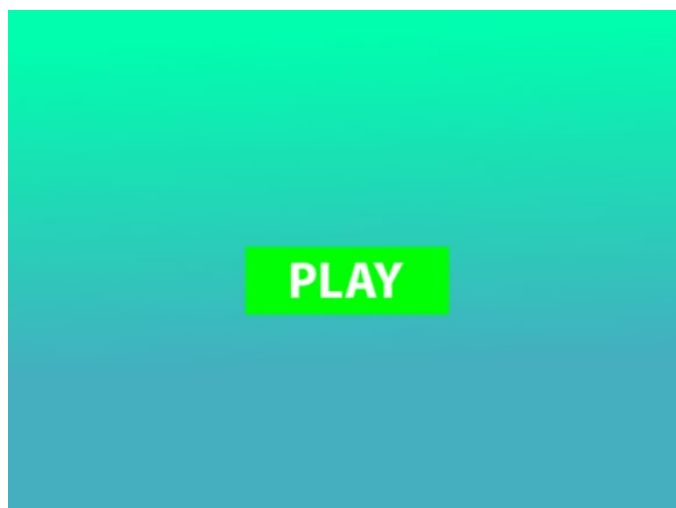


Slika 12 – Događaj Every N secs

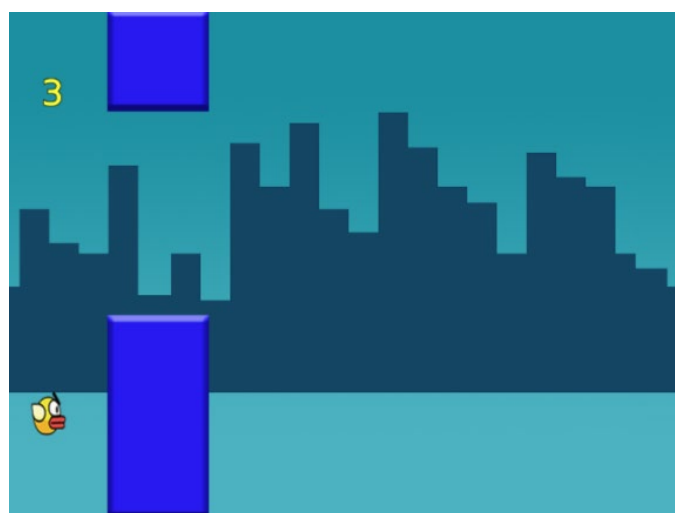
## 4.5 Konačni rezultat

Igra je dostupna na sljedećem linku: <http://www.stencyl.com/game/play/43273>, a za njeno igranje potrebno je omogućiti Flash Player u pretraživaču. Također na linku <https://github.com/atokic/Stencyl> nalaze se resursi korišteni u implementaciji igre, kao i prezentacijski video igre.

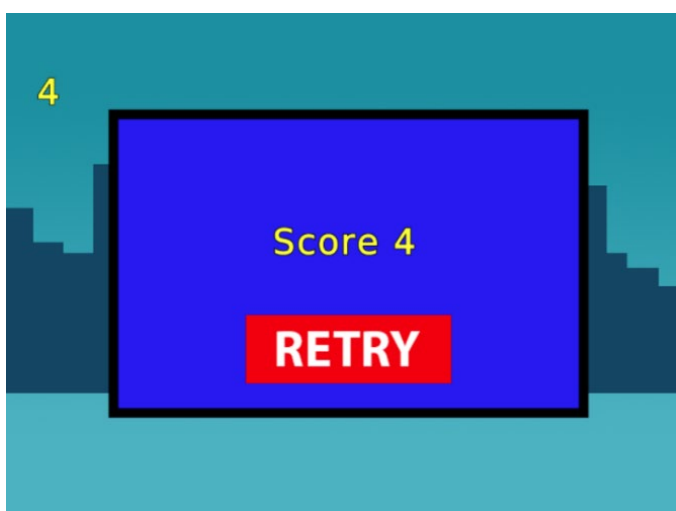
Trenutno je postavljeno da se igra ubrzava svakim zaobilaskom prepreka, ali moguće je igru učiniti još dinamičnijom kreiranjem novih prepreka u vanjskom uređivaču slika.



*Slika 13 - Scena 1*



*Slika 14 - Igra u tijeku*



*Slika 15 - Prikaz prozora s rezultatom i tipkom za ponovni pokušaj*

## 5 Literatura

[1] Stencyl, s Interneta, <https://en.wikipedia.org/wiki/Stencyl>

[2] Stencylpedia, s Interneta, <http://www.stencyl.com/help/>