Izrada igre u Blender programu

Božić, Sanja

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:195:491041

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-26**

Sveučilište u Rijeci
Fakultet informatike
i digitalnih tehnologija

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository

zir.nsk.hr

DIGITALNA KNJIŽNICA

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Sveučilište u Rijeci – Odjel za informatiku

Preddiplomski jednopredmetni studij informatike

Sanja Božić

Izrada igre u Blender programu

Završni rad

Mentor: Doc. dr. sc. Nataša Hoić-Božić

Rijeka, 23.9.2018.

Zadatak

ODJEL ZA INFORMATIKU
SVEUČILIŠTE U RIJECI

Rijeka, 12. 3. 2018.

Zadatak za završni rad

Pristupnik: Sanja Božić

Naziv završnog rada: Izrada igre u Blender programu

Naziv završnog rada na eng. jeziku: Designing a game in Blender

Sadržaj zadatka:

Blender je besplatan softver otvorenog koda namijenjen izradi 3D računalne grafike koji se osim za izradu animacija, vizualnih efekata, 3D modela, interaktivnih 3D aplikacija i sl. također može koristiti i za izradu 3D video igara. U završnom radu opisati će se proces izrade igara u Blender programu pri čemu, kao praktični dio rada, navedene karakteristike programa treba ilustrirati na vlastitom primjeru izrade 3D video igre.

Mentor

Prof. dr. sc. Nataša Hoić-Božić

Voditelj za završne radove

Dr. sc. Miran Pobar

Zadatak preuzet: 22.3.2018.

Božić Sanja
(potpis pristupnika)

Sadržaj

1. Sažetak 4
2. Uvod 5
3. O Blender programu 6
   3.1. Otvoreni kod 6
      3.1.1. Blender zajednica 6
      3.1.2. Primjer izvornog koda 7
4. Izrada igre 8
   4.1. 3D modeliranje 9
   4.2. Vizualno programiranje 10
   4.3. Python programiranje 11
   4.4. Izvoz samostojeće igre 11
5. Učenje kroz igru 13
   5.1. GLAT projekt 13
   5.2. Igre za djecu 13
      5.2.1. Igra Labirint 13
      5.2.2. Igra Skupljačica 14
      5.2.3. Igra Uhvati Me 16
6. Zaključak 18
7. Popis slika 19
8. Popis literature 20
9. Popis izvora 21
10. Popis priloga 22

1. Sažetak

Blender je besplatan 3D program u kojemu se mogu izraditi 3D igre. U ovom radu objašnjeni su potrebni koraci za izgradnju igre u tom programu te dijelovi programa koji su potrebni za izgradnju kao što je vizualno programiranje, modeliranje te programiranje pomoću koda.

Objašnjeno je i kako igre utječu na bolje razmišljanje djece i kako im pomažu u daljnjem napredovanju njihovih sposobnosti.

Također su opisane tri igre kao primjeri 3D igara za djecu koje ih uče raspoznavanju različitih objekata, snalaženja u prostoru te grade natjecateljski duh među skupinom djece.

Ključne riječi: Blender, program, 3D, igra, programiranje, kod, modeliranje, djeca, učenje.

2. Uvod

Igranje igara i korištenje kompjutera, tableta i pametnih telefona je sad svakodnevnica za malu djecu. Sve više i više postoje nove igre za djecu koje potiču učenje i algebarsko razmišljanje. To djeci pomaže u boljim školskim rezultatima i socijalnim vještinama (Bolton, 2016). Tako umjesto samo 2D igra sve popularnije počinju biti 3D igre koje su više privlačnije mlađim naraštajima.

Jedan od programa za izradu 3D igra je Blender koji je popularan zbog svog velikog broja korisnika ali i zbog toga što je besplatan i lak za korištenje (Gumster, 2015). Jedna od poznatijih igra napravljena u tom programu je *Yo Frankie!* (Yo Frankie!, 2009).

3. O Blender programu

Blender je besplatan program otvorenog koda[1] za 3D modeliranje, izradu igara, simulacije, praćenje pokreta, čak i obradu videa. Naprednim korisnicima omogućava i pisanje koda u Python jeziku uz pomoć aplikacijskog programskog sučelja (*eng. Application programming interface*) kojim mogu napraviti posebne alate koji su im potrebni te prilagoditi program svojim potrebama (Foundation, Blender, 2018).

Osim što je program besplatan, može se koristiti na različitim operacijskim sustavima, kao što su Linux, Windows i Macintosh. Napravljen je za korištenje od strane malih studija ili samostalnih umjetnika te programera. Prvo izdanje programa bilo je izdano 1998. godine (Gumster, 2015).

3.1. Otvoreni kod

Cijeli program Blender je otvorenog koda pošto je bio napravljen kao projekt koji vodi zajednica (*eng. Community-driven project*) tako da bi svaka osoba mogla napraviti male ali i velike preinake u osnovnom kodu programa sa ciljem izrade programa većih mogućnosti pristupan svima. Zbog toga mnogo problema i grešaka koji je izvorni kod imao mogao se lakše riješiti uz pomoć zajednice koja je stalno dala svoju podršku i moguća rješenja za probleme. Licencom kojom je program zaštićen je GNU (*eng. General Public License*) tj. općom javnom licencom (Gumster, 2015).

3.1.1. Blender zajednica

Zbog otvorenog koda i zbog svoje pristupačnosti ljudi su sve više počeli koristiti program. Blender zajednica je rasla i sad ima svoje web stranice koje su namijenjene za pomoć novim korisnicima ali i za izmjenjivanje novih ideja i daljnje učenje profesionalnih korisnika. Neke od najpoznatijih stranica za pomoć pri radu sa programom i za izmjenjivanje idea su (Foundation, Community, 2018):

- *Blender artists* - https://blenderartists.org/
- *Blender community* - https://blender.community/c/today
- *Graphic all* - http://graphicall.org/
- *Blender swap* - https://www.blendswap.com/

Osim web stranica postoje posebni kanali u zasebnim programima kao što je Discord na kojima se može razgovarati uživo sa drugim korisnicima Blendera i tako komunicirati sa drugim umjetnicima i programerima koji dijele istu strast prema programu te su spremni pomoći u bilo koje doba dana.

Za profesionalne korisnike postoje i zasebne stranice u kojima se mogu natjecati i dobiti određene nagrade za svoje 3D modele, izrađene igre te simulacije u Blender programu. Najpoznatije natjecanje u zajednici je *Weekly CG Challenge* gdje se korisnici sa svojim 3D modelima i renderanim slikama te videima natječu za prva 3 mjesta svaki tjedan, iako je to natjecanje namijenjeno svim 3D umjetnicima u bilo kojem programu, korisnici najviše koriste Blender ili Maya 3D program kad se natječu (-, 2018). Nagrada za pobjednika je

[1] Generički je naziv za *software* čiji je izvorni kod i/ili nacrt dostupan javnosti na uvid, korištenje, izmjene i daljnje promjene (Petranović, 2018)

nekoliko pretplata na poznatim skupim web stranicama za teksturu, video *tutoriali* za daljnje unaprjeđenje u određenom programu i slično.

3.1.2. Primjer izvornog koda

Programeri u Blender programu programiraju u Python jeziku no sam program Blender je bio napisan pomoću C++ jezika iako ima manjih dijelova koda koji su napisani i u Python-u te u C jeziku (Gumster, 2015). Pošto je program otvorenog koda primjer C++ koda koji je korišten za izradu dijela programa za *game engine* možemo vidjeti na slici (Blender, 2018):

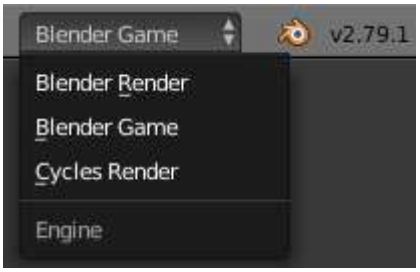
```
37  #include "SCA_AlwaysEventManager.h"
38  #include "SCA_LogicManager.h"
39  #include <vector>
40  #include "SCA_ISensor.h"
41
42  using namespace std;
43
44  SCA_AlwaysEventManager::SCA_AlwaysEventManager(class SCA_LogicManager* logicmgr)
45          : SCA_EventManager(logicmgr, ALWAYS_EVENTMGR)
46  {
47  }
48
49
50
51  void SCA_AlwaysEventManager::NextFrame()
52  {
53          SG_DList::iterator<SCA_ISensor> it(m_sensors);
54          for (it.begin();!it.end();++it)
55          {
56                  (*it)->Activate(m_logicmgr);
57          }
58  }
```

Slika 1. Isječak koda - "always" event manager

Na slici vidimo kod za *manager* 'uvijek' događaja (eng. Always events) koji mogu koristiti pulsni način rada pa se zato trebaju aktivirati. Kao što je prikazano na prijašnjoj slici tako je i ostatak koda napisan sličnim stilom.

4. Izrada igre

Iako nije primarno napravljen za izradnju igara nego za samo 3D modeliranje program Blender ima zasebnu komponentu nazvanu *Blender Game Engine (BGE).* Koristi se i za simulacije, arhitektonske vizualizacije te animacije.

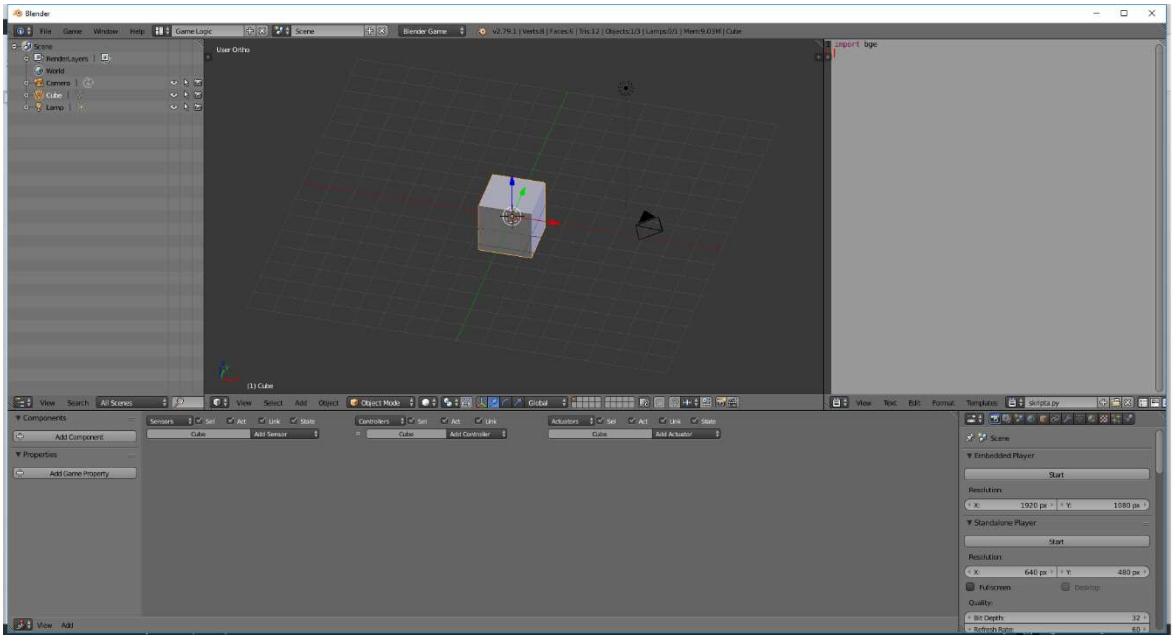


Slika 2. Različite komponente Blender programa

Prilikom izrade igre ili simulacije u BGE imamo četiri osnovna koraka (Aaron Carlisle, 2017):

- Izrada vizualnih elemenata igre kao što su modeli i slike
- Interakcija sa scenom korištenjem vizualnog programiranja ili pisanjem koda
- Stvaranje jedne (ili više) kamera kroz koje će se vidjeti sve scene te modificiranje kamera zbog boljeg izgleda cjelokupne igre
- Pokretanje igre, testiranje, unutar Blender programa sa internim pokretanjem te izvan programa kao samostalnu igru

Sva izrada se radi u glavnom i osnovnom (engl. *default*) rasporedu zaslona (engl. *screen layout*) koje se može mijenjati po potrebi iako se mogu zbog praktičnosti koristiti i druge rasporede koja su već ponuđena kao opcije ovisno o potrebi. Neki drugi rasporedi su za izradu logike igre (engl. *game logic*) koja ima otvorene sve prozore za vizualno programiranje i programiranje korištenjem koda. Donja slika prikazuje navedeni izgled zaslona:



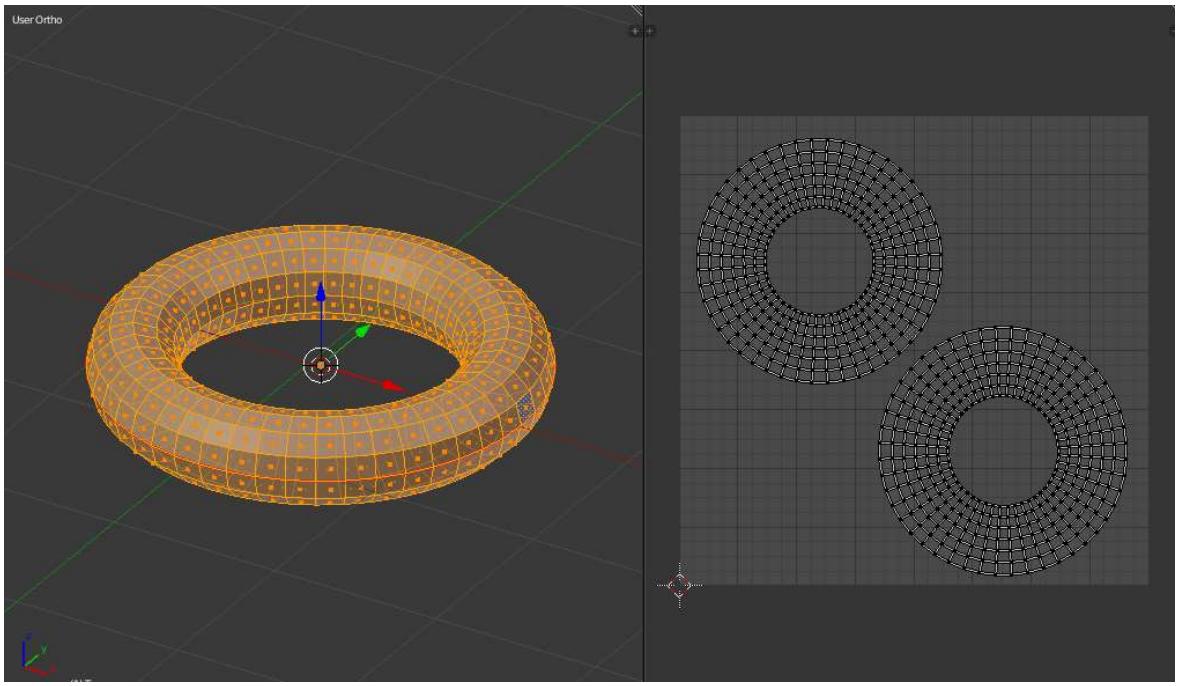
Slika 3. Raspored zaslona za izradu logike igre

4.1. 3D modeliranje

Jedan od koraka kreiranja igre je modeliranje likova i ostalih objekata u sceni. Prilikom otvaranja programa na sredini početne scene se već nalazi kocka. Inače se i samo sa tom početnom kockom može najlakše napraviti cijeli glavni lik koji je potreban za igru. Glavni lik i pozadina (engl. *background*) su najvažniji koraci izgradnje scene. Prvobitno se izgled scene i samih likova nacrta u 2D (eng. *two-dimensional*) obliku pomoću programa kao što su *Photoshop*, *PhotoFiltre*, *Paint* iako se i na papiru, ručno, crta, ovisno kako osoba koja radi igru preferira.

Uz pomoć nacrtane šablone počinjemo rad u 3D programu kao što je *Blender*, *Maya*, *Houdini* i mnogi drugi slični njima. Izgradnja samog modela nije toliko dugotrajan proces koliko je teže pronaći komplementarne boje i teksture koje bi najbolje pristajale tom tipu igre koji se namjerava napraviti.

Modeliranje lika ne znači samo izrada 3D modela kakav god bio nego se treba paziti na puno sitnica koje korisniku nisu toliko važne ali za bolje optimiziranu igru su obavezne. Modeliranje lika ne podrazumijeva samo izradu 3D modela, već i razne procedure čiji rezultati nisu vidljivi korisniku, ali postižu veću optimizaciju igre. Modeli bi trebali imati što manje poligona ne samo zbog optimizacije nego i zbog daljnjeg 3D *unwrapping*-a[2] koji je potreban kod teksturiranja[3] modela. Primjer virtualnog odmatanja (engl. *unwrapping*) prikazan je na sljedećoj slici:

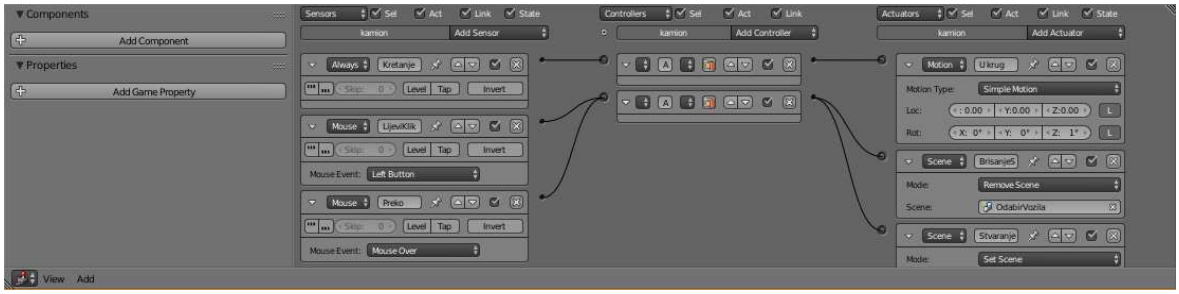


Slika 4. 3D unwrapping

[2] Proces odvijanja 3D modela za kreiranje 2D teksture koja će odgovarati tom modelu (Mairlot, 2015).
[3] „Proces kojim se oživljava model, daje mu se boja i svojstvo.“ (Babić, 2016)

4.2. Vizualno programiranje

Vizualno programiranje se u programu Blender zasniva na tome da se koristi tzv. sistem logičkog blokiranja (engl. *logic brick system)*. To je sistem u kojemu svaki objekt igre (model) može imati zasebne logičke komponente koje kontroliraju ponašanje objekta u toj sceni. Glavni dio kontrole objekta se može postaviti putem logičkog urednika (engl. *logic editor*) (Gumster, 2015). Donja slika prikazuje izgled tog urednika i sami primjer vizualnog programiranja:



Slika 5. Prikaz vizualnog programiranja – logic editor

Sama logika objekta u uredniku je postavljena sa blokovima (engl. *blocks*) koji imaju u sebi preprogramirane osnovne funkcije. Blokovi se mogu namjestiti i prepraviti onako kako je korisniku potrebno za igru koju želi napraviti. Postoje tri tipa blokova:

- Senzori (engl. *sensors*)
- Kontroleri (engl. *controllers*)
- Aktivatori (engl. *actuators*)

Senzori šalju pozitivan puls svim kontrolerima sa kojima su spojeni u trenutku kada se promjeni početno stanje objekta kojeg promatraju. Kontroleri skupljaju podatke koje su im poslali senzori tako da bi na kraju aktivatori proizveli akcije kao što su kretanje, stvaranje novog objekta, sviranje glazbe i slično. Veze koje ih spajaju mogu biti samo od senzora prema kontroleru i od kontrolera prema aktivatoru. Aktivatori se ne mogu natrag povezati sa senzorima ali postoje specijalni tipovi aktivatora i senzora koji nude tu mogućnost (Aaron Carlisle, 2017).

Sam kod blokova se ne može mijenjati pa u nekim slučajevima nedostaje potrebne slobode mijenjanja nekih funkcija. Zato se za neke veće simulacije i kompliciranije igre samostalno piše kod, iako se sve više i više koristi oba dva načina. Za jednostavnije naredbe se koristi vizualno programiranje tako da se brže postigne isto što smo mogli napisati sa kodom, a za kompliciranije dijelove se koristi kod.

4.3. Python programiranje

Pri izradi kompleksnih igara u Blender programu pišemo kod u Python programskom jeziku koji se piše u tekst uređivaču (engl. *text editor*). Pišu se skripte koje se inače pomoću vizualnog programiranja nadodaju na neki senzor koji šalje potrebne informacije o objektu samoj skripti. Jedan primjer koda za prikaz preko cijelog zaslona u igri možete vidjeti ovdje:

```
from bge import logic, render

left_click = logic.getCurrentController().sensors['MouseLC']
mouse_over = logic.getCurrentController().sensors['MouseOver']

if mouse_over.hitObject and left_click.positive:
    if render.getFullScreen():
        render.setFullScreen(False)
    else:
        render.setFullScreen(True)
```

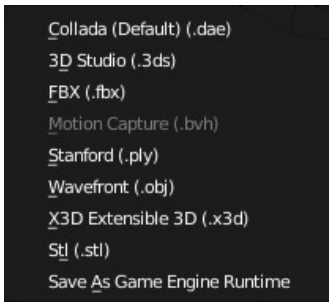
Slika 6. Isječak koda – prikaz preko cijelog zaslona

Sam program Blender dolazi sa već instaliranim programskim jezikom Python i Blender modulima tako da sam korisnik ne treba ništa dodatno instalirati. Tri su glavna modula u programu (Foundation, API, 2017):

- Aplikacijski moduli
- Samostalni moduli
- Moduli za *game engine*

4.4. Izvoz samostojeće igre

Kad je igra gotova vrijeme je da se testira kao samostojeća igra (engl. *standalone game*). U Blenderu većina mogućnosti izvoza (engl. *export*) su moguća pomoću dodataka koji se zasebno trebaju instalirati u postavkama programa (Gumster, 2015). Neki od formata za *export* možete vidjeti na donjoj slici:



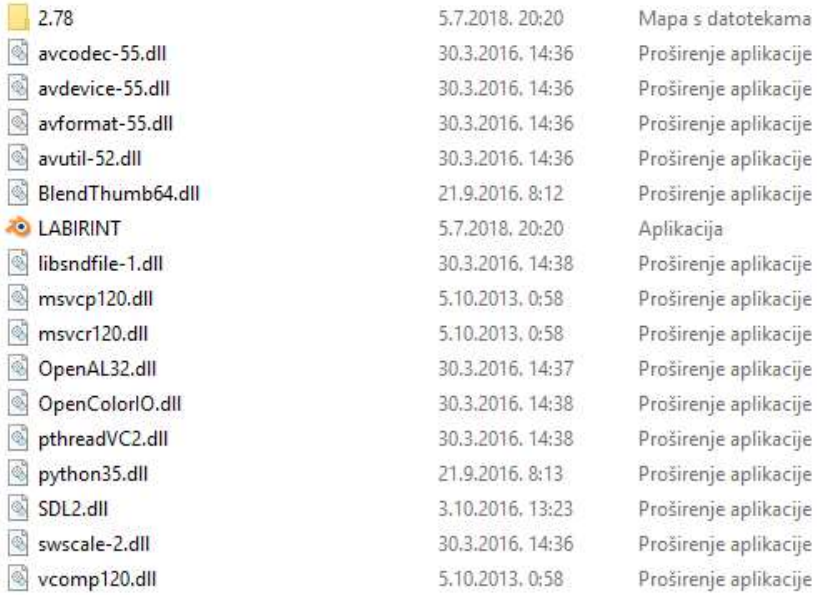
Slika 7. Formati za izvoz

Za samostojeću igru inače se koristi zadnja opcija koja (*save as game engine runtime*) koja sprema igru u .exe[4] formatu. U tom procesu svo vizualno programiranje se pretvara u Python kod koji je u zasebnoj datoteci te kreiraju se dll.[5] datoteke koje su isto potrebne za pokretanje na drugim računalima. Pokretati igru se može bez ikakvog problema, na svim

[4] Predstavlja ekstenziju za samostojeći program (Veizović, 2017).

[5] Predstavlja ekstenziju za skup funkcija i procedura koje mogu koristiti drugi programi (Veizović, 2017).

računalima koji podržavaju .exe datoteke i koje imaju mogućnost prikaza 3D slike. Datoteke koje se kreiraju ovim načinom izvoza možete vidjeti na donjoj slici:



Slika 8. Prikaz kreiranih datoteka pri exportu igre

5. Učenje kroz igru

Dok djeca igraju igre postoji skup ciljeva i interakcija. Iako problem u igri nije isti kao i u stvarnom svijetu osnovna pretpostavka je, ako se ne riješi problem cilj se nikada neće ostvariti. Uz to se problem mora riješiti na način koji je prihvatljiv. Pomoću igara djeca razmišljaju i planiraju strategiju kako doći do cilja. Takvo rješavanje problema može pomoći pri uspjehu u budućnosti djeteta (KPCommunications, 2017) .

5.1. GLAT projekt

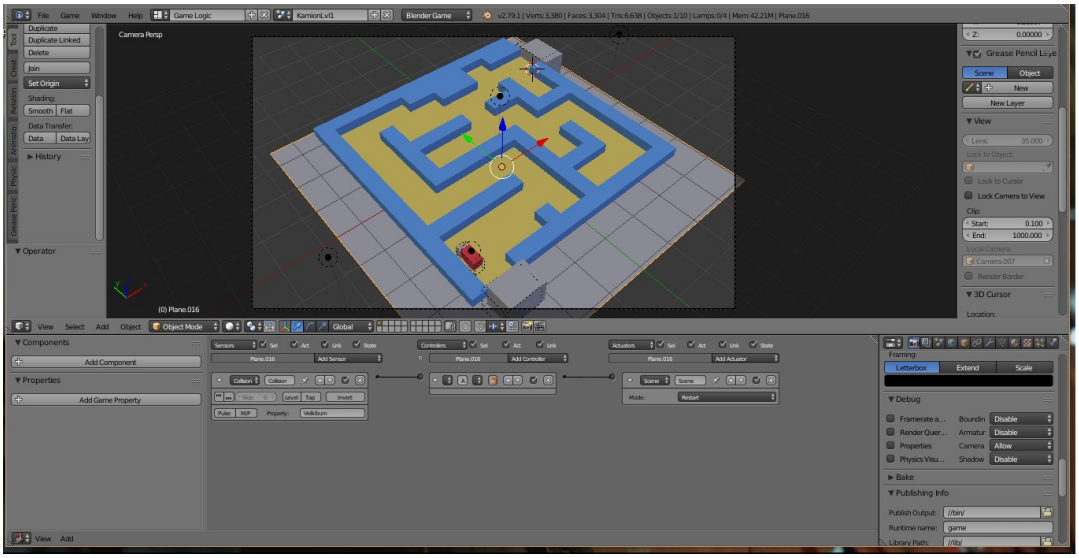
Postoje različiti projekti koji pokušavaju potaknuti djecu na algoritamsko razmišljanje, jedan od njih je projekt GLAT (*Game of Learning Algorithmic Thinking*). Želi se potaknuti učenje učenika od prvog do četvrtog razreda osnovne škole. „Posebna pažnja će se posvetiti učenju uz pomoć didaktičkih igara (engl. *serious games*) odnosno konkretnoj primjeni učenja uz pomoć igara (*Game Based Learning* - GBL) i igrifikacije (engl. *gamification*) za učenje digitalnih vještina općenito te posebice vještina programiranja.“ (Hoić-Božić, Holenko Dlab, Načinović Prskalo, Rugelj, & Nančovska Šerbec, 2018)

5.2. Igre za djecu

Postoje različite vrste igra za djecu koje im dopuštaju učenje uz zabavu i time djeca lakše zapamte. Neke od popularnih video igara za djecu koje ih uče su: igra Sunčica (napravljena od strane hrvatske tvrtke 32bita), Maša i Medvjed (zbog popularnosti crtanog filma ima i razne video igre koja uče djecu o šumskim životinjama i o samim šumama), *Cookie Monster's Challenge* (djeca rješavaju zadatke tako da sakupe dijelove stroja za izradu kolačića), *Spelling Test Free* (uči djecu slovkanju riječi) te mnoge druge (Čizmić, 2016). No na tržištu nema toliko 3D igra koliko ih ima 2D zato u nastavku možemo vidjeti 3 igre napravljene u Blender programu za učenje djece osnovnih stvari. Verzija koja je korištena u izradi sljedećih igara je 2.79 uz malu pomoć UPBGE 0.2.3. koja je zasebna ažurirana vezija Blender *Game Engina*.

5.2.1. Igra Labirint

Prva 3D igra je zvana Labirint u kojoj djeca izaberu između auta i kamiona te prolaze ,kao što ime navodi, labirinte. Igra se igra pomoću strelica i u lijevom gornjem kutu ekrana prikazuje se slika strelice koja je pritisnuta, tako da djeca i vizualno vide koja je točno tipka pritisnuta. Uz popratnu muziku djeca trebaju pronaći put do kraja labirinta kako bi prošla razinu i došla na sljedeću. Labirinti su postepeno teži. Izgled igre u Blender programu je prikazan na slici 9.



Slika 9. Igra Labirint - u programu Blender

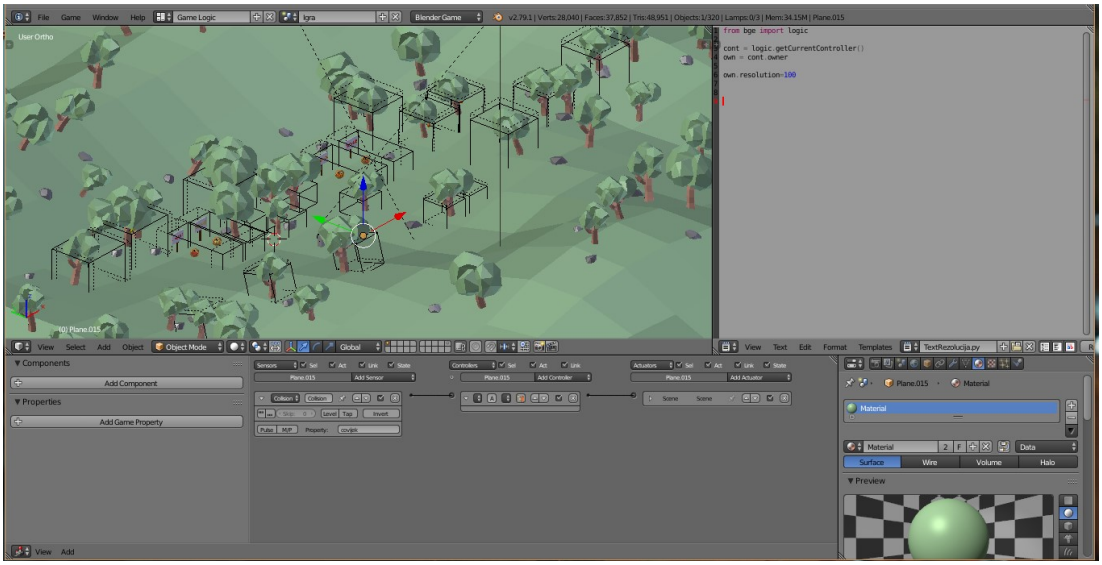
Pošto se može birati između auta i kamiona igra ima različite labirinte za auto, a različite za kamion. Ukupno imamo 8 labirinata, 4 za auto i 4 za kamion, tj. razina pošto je svaki labirint u jednoj razini.

Jednu razinu u igri možete vidjeti na donjoj slici:

Slika 10. Igra Labirint – primjer razine

5.2.2. Igra Skupljačica

Druga igra zvana Skupljačica je igra u kojoj djeca skupljaju određena voća iz drveća. Voće koje se treba skupljati je napisano na gornjoj lijevoj strani ekrana. Na drveću ima raznih vrsta voća koje je ono koje se skuplja i ono koje ne. Djeca trebaju odlučiti koja je točno voće, ako pokušaju uzeti krivo voće u ukupnom zbroju bodova, koji piše na gornjoj desnoj strani zaslona, izgubiti će bodove. U suprotnom, ako uzmu točno voće dobiti će bodove. U programu Blender igra izgleda ovako:



Slika 11. Igra Skupljačica - u Blender programu

Sam cilj igre je uzeti točno voće iz drveća i staviti ga u točnu košaru. Svaka košara iza sebe ima natpis koje voće treba biti u njoj. Djeca spuštaju voće u košaru tako da pritisnu prvo slovo imena voća na tipkovnici, ako je prvo slovo tog voća točno voće se spušta u košaru ako nije voće ostaje kod igrača.

Na kraju igre koja ima dvije razine (razina gdje se skupljaju jabuke i kruške i razina gdje se skupljaju trešnje i naranče) na ekranu će se prikazati lista pet najboljih rezultata tako da će djeca moći uspoređivati svoj rezultat sa drugom djecom koja su igrala tu igru prije njih i tako razvijati svoj natjecateljski duh (Slika 12.). Osim natjecateljskog duha djeca uče raspoznavati različita vrsta voća i imena tih voća.

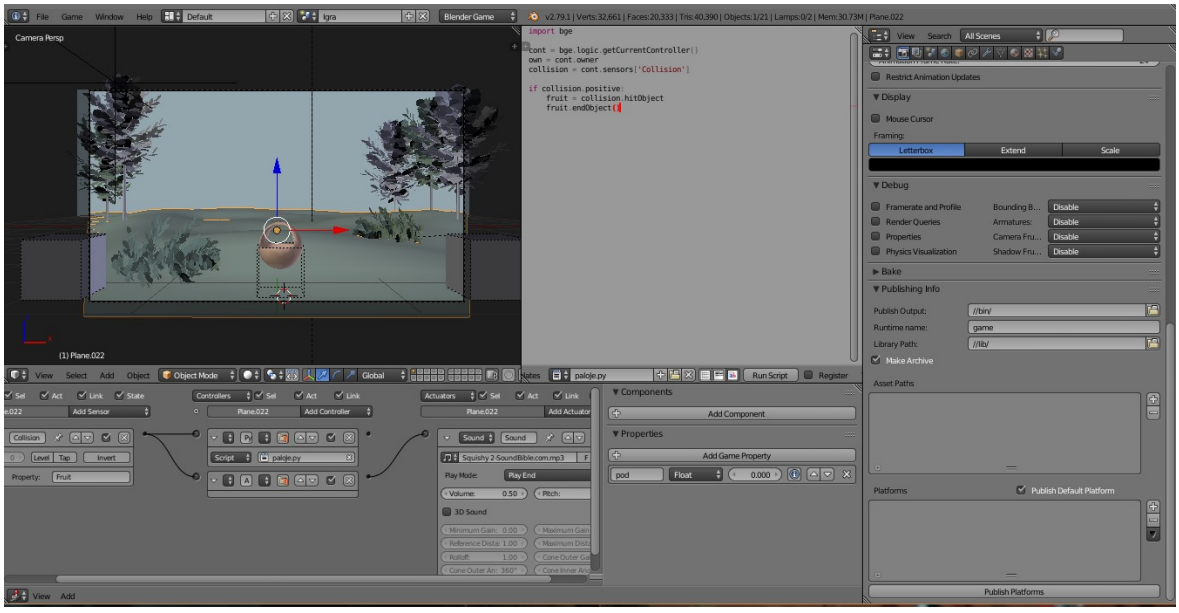
Slika 12. Igra Skupljačica - najbolji rezultati

Malom svijetom gdje su drveća sa voćem može se kretati pomoću strelica ili WASD tipkama na tipkovnici, a sa mišem se rotira pogled. Izgled samo igre možete vidjeti na donjoj slici:

Slika 13. Igra Skupljačica - izgled

5.2.3. Igra Uhvati Me

Zadnja igra je slična kao igra Skupljačica ali jednostavnija pošto se kontrolira košara sa strelicama lijevo-desno na tipkovnici. Treba se sakupiti voće koje se prikazuje u desnom donjem kutu ekrana kao slika i koje se nasumično mijenja. Od gore pada voće koje se isto nasumice generira. Ako je uzme voće koje nije na slici izgubi se život, ako se uzme točno voće dobije se jedan bod. Kako to izgleda u Blender programu možete vidjeti ovdje:



Slika 14. Igra Uhvati Me - u Blender programu

Kad se izgube sva tri života igra je gotova i dolazimo do scene sa najboljim rezultatima u igri, koja je slična kao i kod igre Skupljačica. Izgled igre Uhvati Me možete vidjeti na donjoj slici:

Slika 15. Igra Uhvati Me

6. Zaključak

Djeci je lakše učiti vizualno i pošto su u današnjem svijetu okruženi tehnologijom koja im to dopušta učenje uz tehnologiju je korak naprijed. Izgradnjom ozbiljnih igara (engl. *serious games*) možemo sve sate koje djeca provode gledajući u ekran iskoristiti za njihovo dobro i za njihovu budućnost. Uz to možemo ih naučiti osnovama programiranja i logičkog razmišljanja koje će moći upotrijebiti u svakodnevnom životu.

Program Blender je jedan od takvih programa koji je dobar u izradi 3D video igara te modeliranja općenito. Iako su osnivači Blendera najavili ukidanje *Game Engina* u verziji 3.0 ona još nije uvedena i još se diskutira o mogućnosti ponovnog uvođenja *Game Engina* u nadolazećim verzijama. No i da dođe do ukidanja postoji još dobrih 3D programa u kojima se lako može napraviti igra za učenje kao što je npr. Unity te to neće spriječiti autore u izradi igara za nove generacije djece.

7. Popis slika

Slika 1. Isječak koda - "always" event manager 7
Slika 2. Različite komponente Blender programa..... 8
Slika 3. Raspored zaslona za izradu logike igre..... 8
Slika 4. 3D unwrapping 9
Slika 5. Prikaz vizualnog programiranja – logic editor..... 10
Slika 6. Isječak koda – prikaz preko cijelog zaslona 11
Slika 7. Formati za izvoz..... 11
Slika 8. Prikaz kreiranih datoteka pri exportu igre 12
Slika 9. Igra Labirint - u programu Blender..... 14
Slika 10. Igra Labirint – primjer razine..... 14
Slika 11. Igra Skupljačica - u Blender programu..... 15
Slika 12. Igra Skupljačica - najbolji rezultati..... 15
Slika 13. Igra Skupljačica - izgled 16
Slika 14. Igra Uhvati Me - u Blender programu 16
Slika 15. Igra Uhvati Me..... 17

8. Popis literature

Aaron Carlisle, F. S. (2017). *Manual.* Dohvaćeno iz Blender Documentation: https://docs.blender.org/manual/en/dev/editors/logic_editor.html

Babić, D. (2016). *Montaža teksture na trodimenzijske objekte.* Dohvaćeno iz https://bib.irb.hr/datoteka/831030.Final_0036470256_411.pdf

Bolton, D. (2016). *Video games may improve children's intellectual and social skills*. Dohvaćeno iz Independent: https://www.independent.co.uk/news/science/video-games-children-learning-intelligence-social-skills-study-a6920961.html

Čizmić, M. (18. 1 2016). *Edukativne aplikacije za djecu*. Dohvaćeno iz Woman in Adria: http://www.womeninadria.com/edukativne-aplikacije-za-djecu/

Foundation, B. (3. 10 2017). *API.* Dohvaćeno iz Blender Documentation: https://docs.blender.org/api/2.79/

Foundation, B. (2018). Dohvaćeno iz Blender: https://www.blender.org/about/

Foundation, B. (2018). *Community*. Dohvaćeno iz Blender: https://www.blender.org/community/

Gumster, J. v. (2015). *Blender for Dummies.*

Hoić-Božić, N., Holenko Dlab, M., Načinović Prskalo, L., Rugelj, J., & Nančovska Šerbec, I. (2018). *Projekt GLAT.* Dohvaćeno iz docs.mipro-proceedings.com: http://docs.mipro-proceedings.com/ce/ce_65_4809.pdf

KPCommunications. (31. 1 2017). *The relationship between learning and video games among children*. Dohvaćeno iz Kids Space: https://www.kidspeace.org/the-relationship-between-learning-and-video-games-among-children/

Mairlot, R. (2015). Dohvaćeno iz Blender stackexchange: https://blender.stackexchange.com/questions/38651/what-does-unwrapping-a-model-do-and-why-is-it-important

Petranović, P. S. (2018). *Otvoreni kod*. Dohvaćeno iz http://otvorenikod.weebly.com/.

Veizović, M. (2017). *PE FORMAT (.EXE, .DLL).* Dohvaćeno iz https://lecto-player.lecto.org/recordings/fer/predmeti/racfor/2016/seminari/mveizovic/seminar.pdf

9. Popis izvora

-. (2018). Dohvaćeno iz Weekly CG Challenge: http://weeklycgchallenge.com/

Aaron Carlisle, F. S. (2017). *Manual.* Dohvaćeno iz Blender Documentation: https://docs.blender.org/manual/en/dev/editors/logic_editor.html

Blender. (2018). Dohvaćeno iz GitHub: https://github.com/sobotka/blender

Foundation, B. (3. 10 2017). *API.* Dohvaćeno iz Blender Documentation: https://docs.blender.org/api/2.79/

Foundation, B. (2018). Dohvaćeno iz Blender: https://www.blender.org/about/

Foundation, B. (2018). *Community*. Dohvaćeno iz Blender: https://www.blender.org/community/

Mairlot, R. (2015). Dohvaćeno iz Blender stackexchange: https://blender.stackexchange.com/questions/38651/what-does-unwrapping-a-model-do-and-why-is-it-important

Yo Frankie! (2009). Dohvaćeno iz https://apricot.blender.org/

10. Popis priloga

Prilog uz ovaj rad su tri igre:

- Skupljačica.zip
- UhvatiMe.zip
- Labirint.zip