2D interaktivna igra

Zrnić, Matko

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:849110

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij računarstvo

2D INTERAKTIVNA IGRA

Završni rad

Matko Zrnić

Osijek, 2017.

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. CILJEVI 2
2.1. Način ostvarivanja ciljeva 2
3. CONSTRUCT 2 3
4. RAZVOJ IGRE 6
4.1. Glavni likovi 6
4.1.1. Ciljevi za korisnike 9
4.2. Neprijatelji 9
4.3. Razine 11
4.4. Glavni izbornik 12
4.5. Opis igre 14
5. ISPITIVANJE 16
6. ZAKLJUČAK 17
LITERATURA 18
SAŽETAK 19
ABSTRACT 20
ŽIVOTOPIS 21
PRILOZI 22

1. UVOD

U završnom radu se radi o izradi 2D igre namijenjene za jednog igrača. Naziv igre je Neci, a izrađuje korištenjem Construct 2 arhitekture. Problem izrade se rješava korištenjem i kombiniranjem svih elemenata, alata i mogućnosti programa Construct 2. Osim toga potrebno je koristiti dodatne programe kao što je Paint.net za izradu grafičkih elemenata. Izrada osim programskih elemenata zahtjeva korištenje logike objektno orijentiranog programiranja. Upoznavanje osnovnih elemenata i mogućnosti programa je omogućeno na web stranici [5], ali potpuno korištenje svih mogućnosti zahtjeva već stečeno znanje objektno orijentiranog programiranja. Nakon izrade omogućeno je dodavanje vizualnih i zvučnih efekata, te jednostavno testiranje kako bi se što bolje prilagodila igra. Vizualni efekti se ostvaruju dodavanjem svojstava i potrebnih slika, dok se zvučni efekti postižu dodavanjem preuzete glazbe [4]. Testiranje ne zahtjeva prevođenje koda nego je jednostavno i efektivo korištenjem *web* preglednika. Jedna od prednosti korištenja ovog programskog paketa je podržavanje na raznim platformama kao što su preglednici koji podržavaju HTML5 i operacijski sustavi: Mac, Linux i Windows, tako da je vrlo lako prilagoditi igru ovisno o potrebama.

1.1. Zadatak završnog rada

Zadatak završnog rada je izraditi 2D računalnu igricu. Potrebno je ispuniti ciljeve izrade igre (detaljnije objašnjeni u poglavlju 2) korištenjem svih programskih mogućnosti i ponuđenih alata programa Construct 2.

2. CILJEVI

Ključni cilj za razvoj igre je omogućiti interakciju korisnika i igre. Osnovni ciljevi su: postavljanje ciljeva za korisnike, kreiranje neprijatelja, postavljanje više razina različite težine i uvođenje dodatnih opcija za korisnike.

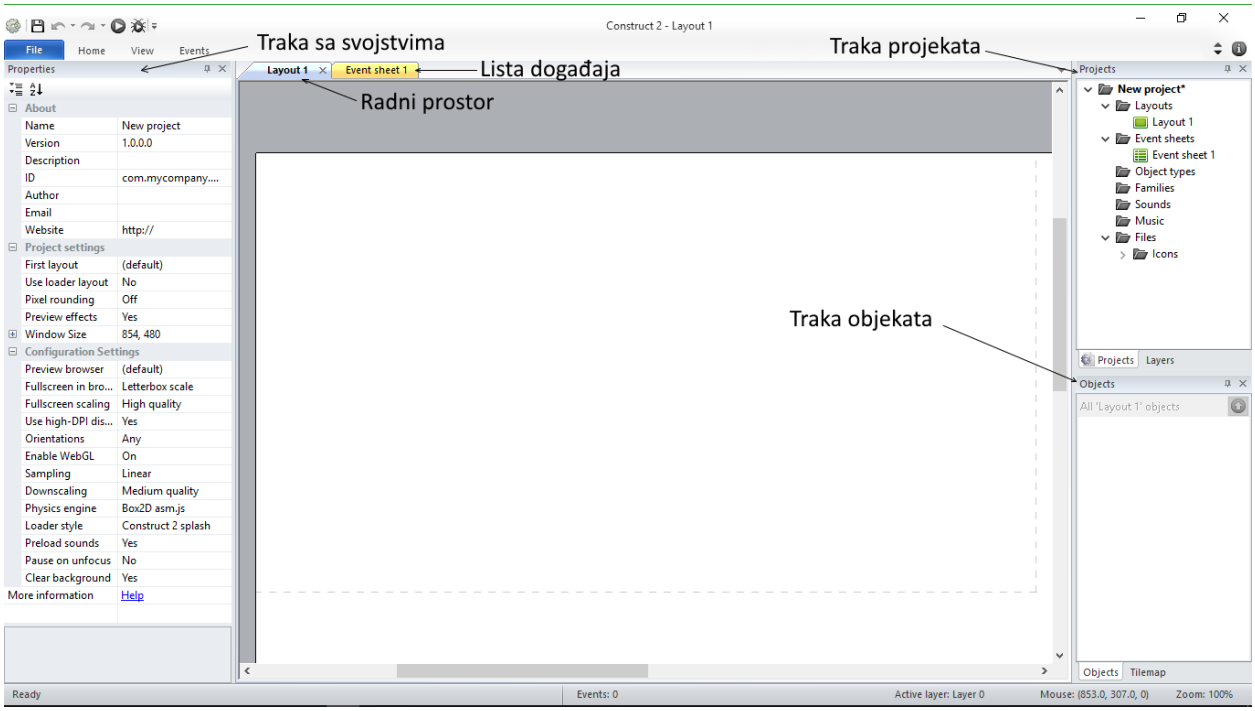
2.1. Način ostvarivanja ciljeva

Interakcija korisnika i igre omogućena je dodavanjem svojstava glavnim likovima za kretanje po prostoru. Postavljanje ciljeva za korisnike se ostvaruje na način dodavanja objekata za prikupljanje. Vrijednosti tih objekata se spremaju u globalne varijable. Postavljeni su događaji koji vrijednosti tih varijabli prikazuju na zaslonu. Kreiranje neprijatelja se postiže dodavanjem mnogobrojnih objekata. Za svaki od tih objekata se postavljaju potrebna svojstva i događaji potrebni za kontrolu.Postavljanje više razina različite težine omogućuje se dodavanjem objekata kojima se dodjeljuje svojstvo čvrstog objekta i raspoređivanje neprijatelja nad kojima se uređuju događaji zaslužni za kontrolu tih objekata. Uvođenje dodatnih opcija za korisnike se postiže kreiranjem glavnog izbornika sa objektima nad kojima se postavljaju događaji sa mogućnostima korištenja miša.

3. CONSTRUCT 2

Construct 2 spada u moderno objektno orijentirane jezike. [1] To je program za razvoj 2D igara baziran na HTML5 platformi i napisan je u C++ i JavaScript jeziku. Razvijen je od tvrtke Scirra i izašao je na tržište 4. veljače 2011.

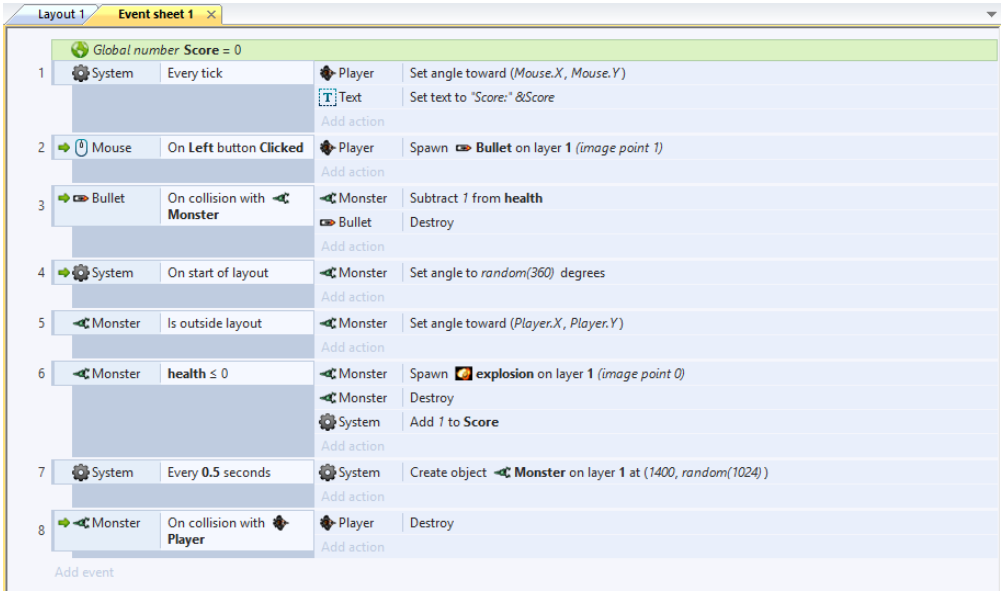
Sučelje od Construct 2 programa (Sl. 3.1) se sastoji od: radnog prostora (eng. *Layout*), liste događaja (eng. *Event sheet*), trake sa svojstvima (eng. *Properties*), trake projekata (eng. *Projects*) i trake objekata (eng. *Objects*).



Sl. 3.1: Construct 2 sučelje

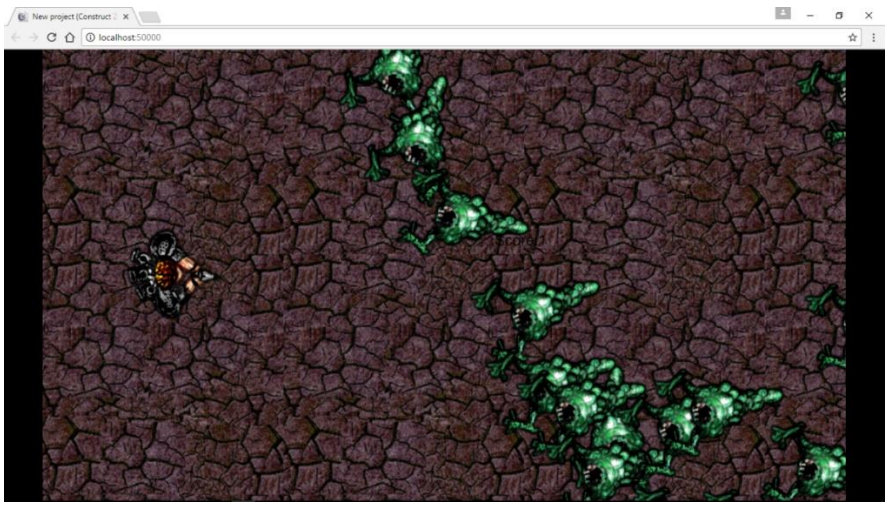
Kreiranjem novog projekta se prikazuje radni prostor, odnosno podloga na koju je moguće smještati i premještati objekte. Objekti se mogu pomicati, rotirati, izmjenjivati veličinu, dodjeljivati vizualni efekti. Postavke se vrlo lagano mijenjaju pomoću trake sa svojstvima, trake projekata i trake objekata, a kontrola nad objektima se postiže događajima na lista događaja. Traka sa svojstvima ima razne mogućnosti od kojih su najvažnije postavljanje osnovnih ponašanja objekata. Svojstvima se određuju svi početni parametri potrebni za razvoj igre. Traka projekata omogućuje jednostavan pristup svakom projektu i zatim pri svakom projektu pogled na sve radne prostore, liste događaja i objekte koje projekt sadrži. Traka objekata je jednostavan prikaz svih objekata za jedan projekt što omogućuje lakše snalaženje pri izradi.

Glavna metoda programiranja u Construct 2 programu se temelji na listama događaja (Sl. 3.2). Lista događaja sadrži događaje (eng. *Event*) i globalne varijable (eng. *Global variable*). Događaji su glavni način programiranja u Construct-u. Rade na način okidača, svaki događaj ima neki uvjet ili uvjete koji se ispunjavaju, a zatim oni obave zadane funkcije.



Sl. 3.2: Lista događaja

Najvažnije svojstvo programa je jednostavnost korištenja. Izrada jednostavnih računalnih igara je omogućena na vrlo jednostavan način kreiranjem objekata u dijelu za rad i dodavanje ponašanja svakom objektu te kreiranje događaja za kontrolu tih objekata. Za lakšu prilagodbu pri izradi omogućen je trenutni pregled igre za bilo koje vrijeme izrade. Testiranje je omogućeno tijekom cijele izrade što mnogo olakšava posao i skraćuje vrijeme potrebno za izradu. Primjer testiranja je vidljiv na slici 3.3



Sl. 3.3: Pregled igre u web pretraživaču

Za razliku od ostalih programa kao što je C# kojemu je Construct najsličniji, nema potrebu za prevođenje koda ili neke posebne procese koji oduzimaju mnogo vremena. Sličnosti koje ima sa programom C# su prostor u koji se mogu dodavati razni objekti, mijenjanje svojstava tih objekata i upravljanje objektima. Jedina bitna razlika između tih programa je što Construct koristi mnogo jednostavnije događaje dok se u C#-u koriste razne funkcije. Usporedba je prikazana na slici 3.4 (primjer za Construct 2 program) i slici 3.5 (primjer za program C#). Događaj u Constructu i kod napisan u C# jeziku prikazuju obavljanje iste radnje. Ta radnja je prikaz teksta "Poruka" kada se mišem klikne na gumb.



Sl. 3.4: Događaj za prikaz poruke pomoću gumba u programu Construct 2

```
private void button_Click(object sender, EventArgs e)
{
    MessageBox.Show("Poruka");
}
```

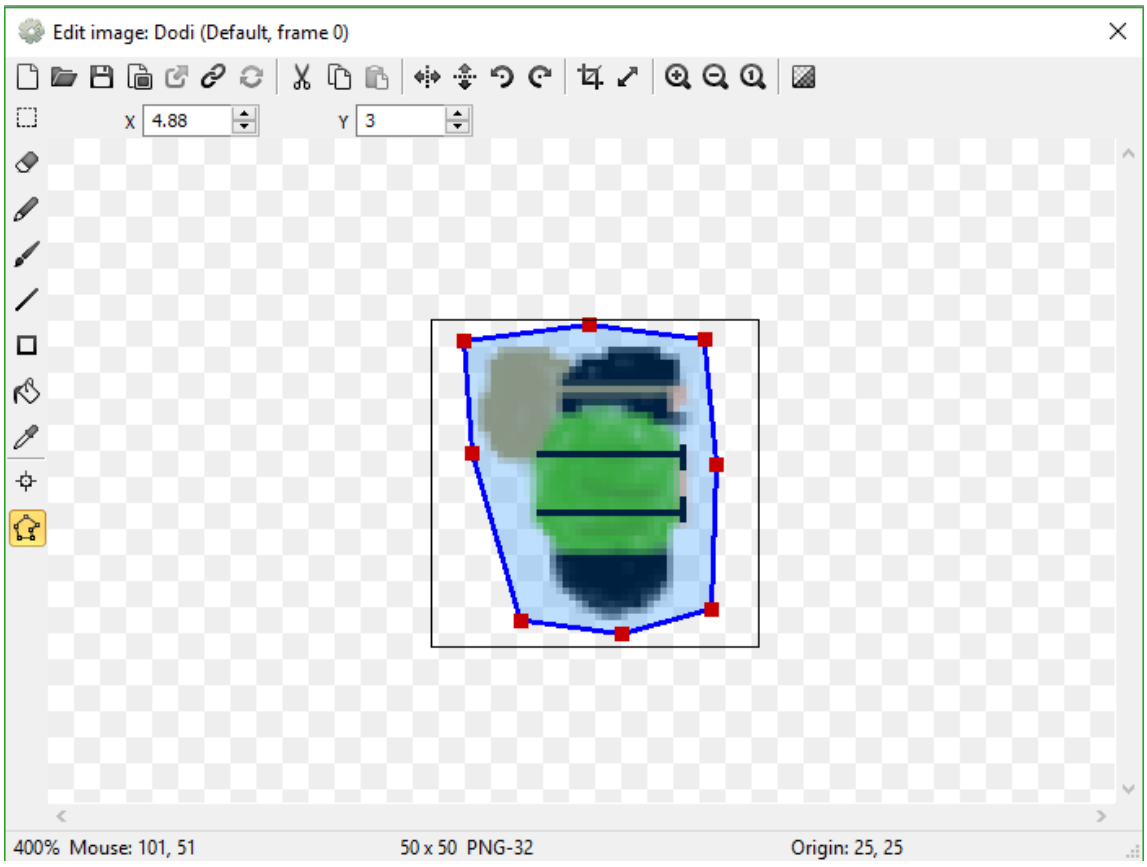
Sl. 3.5: Događaj za prikaz poruke pomoću gumba u programu C#

4. RAZVOJ IGRE

Razvoj igre je podijeljen na faze, a svaka od tih faza predstavlja ispunjavanja jednog cilja izrade igre. Prva faza je postavljanje glavnih likova koji se mogu kretati i postavljanje ciljeva za korisnike. U drugoj fazi se kreiraju neprijatelji. Treća faza je nadograđivanje igre, odnosno postavljanje više razina sa različitim težinama. Posljednja faza razvoja je dodavanje glavnog izbornika kako bi korisnik imao dodatne mogućnosti.

4.1. Glavni likovi

Za kreiranje glavnih likova koristi se objekte *Sprite*. Sprite je objekt koji se koristi pri izradi likova iz razloga što su mu omogućena potrebna svojstva. Kreiranjem objekta *Sprite* se prikazuju mogućnosti postavljanja granica kolizije (eng. *Collision polygon*) i točku slike (eng. *Image point*) prikazano na slici 4.1.



Sl. 4.1: Kreiranje objekta sprite

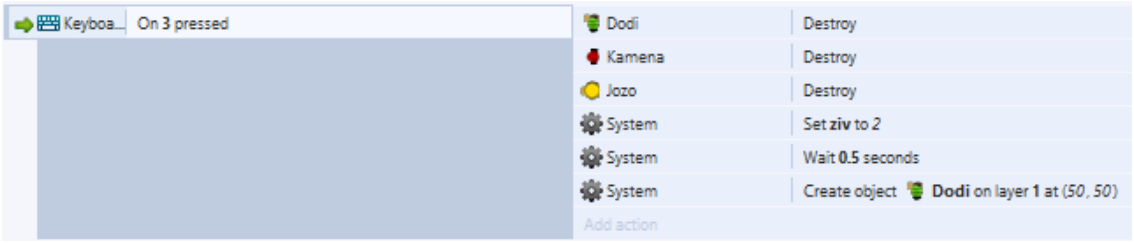
Granice kolizije se postavljaju kako bi se izostavio nebitan prostor, ali je ključno ostaviti malo prostora kako bi se izbjegle greške korištenjem računala sa slabim mogućnostima procesora. Kolizija je jedan od najbitnijih čimbenika pri izradi računalnih igara iz razloga što omogućuje ograničenost kretanja. Kolizija funkcionira tako da varijablu koja sprema vrijednost brzine objekta postavlja na nula u slučaju da se granice dva ili više objekta imaju iste vrijednosti pozicije.

Točka slike služi kako bi se odredila točna pozicija objekta u svakom trenutku. Za svaku sličicu u sekundi (eng. *Frame Per Second*, FPS) pozicija točke slike se sprema u varijable objekta X i Y. Varijabla X predstavlja vrijednost x koordinate objekta na radnom prostoru dok Y varijabla predstavlja vrijednost y koordinate. Svojstva prikazana na slici 4.2 prikazuju početne vrijednosti varijabli X i Y pod svojstvom *Position* i mogu se mijenjati tako da se početna pozicija objekta postavi prema potrebi.

Name	Dodi
Plugin	Sprite
UID	0
Global	No
Common	
Layer	likovi
Angle	0
Opacity	100
Position	1165.983, -448.445
Size	50, 50
Instance variables	
Add / edit	Instance variables
Behaviors	
⊞ BoundToLayout	
⊟ 8Direction	
Max speed	400
Acceleration	600
Deceleration	500
Directions	8 directions
Set angle	360 degree (smo...
Default contr...	Yes
Initial state	Enabled
Add / edit	Behaviors
Effects	
Container	
Properties	
Animations	Edit
Size	Make 1:1
Initial visibility	Visible
Initial animation	Default
Initial frame	0
Collisions	Enabled

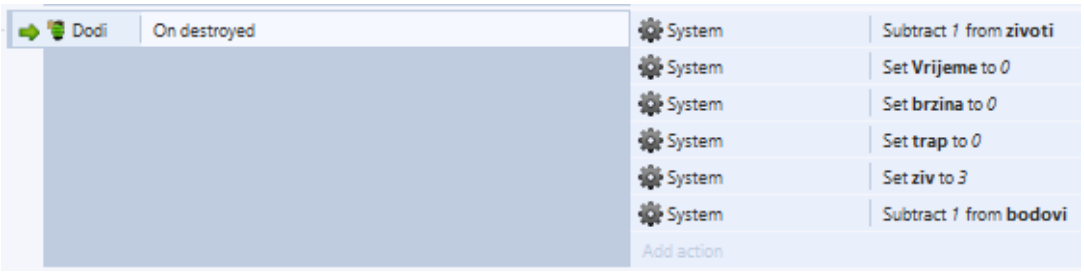
Sl. 4.2: Svojstva objekta Dodi

Kako bi omogućili kretanje objekta dodaje mu se svojstvo *8Direction* prikazano na slici 4.2. *8Direction* je već predefinirana funkcija koja omogućuje da se objekt pomiče po radnom prostoru korištenjem strjelica na tipkovnici u osam smjerova. Kada je dodana mogućnost kretanja, moguća je izmjena brzine kretanja, ubrzavanja i usporavanja.. Dodavanjem kretanja se javlja problem izlaska iz radnog prostora tako da se postavlja svojstvo *BoundToLayout*. Dodavanje svojstva *BoundToLayout* prikazanog na slici 4.2 objekt je ograničen na granice radnog prostora. To svojstvo funkcionira na način da pri koliziji objekta i granice radnog prostora zaustavlja objekt kako se ne bi mogao kretati u tom smjeru. Nakon što se kreiraju glavni likovi, potrebno je omogućiti kreiranje i kontrolu nad uništavanjem tih likova. Za primjer kreiranja glavnog lika je uzet primjer kreiranja objekta Dodi. Događaj na slici 4.3 uz uvjet da se pritisne tipka 3 na tipkovnici uništava sve glavne likove kako ne bi kreirali više glavnih likova. Nakon uništavanja postavlja globalnu varijablu *ziv* na vrijednost 2, a globalna varijabla *ziv* sluzi kao spremnik informacije koji se glavni lik trenutno koristi. Za postizanje efekta kreiranja objekta, korištena je jednostavna funkcija koja odgodi izvršavanje sljedeće funkcije za 0.5 sekundi. Nakon izvršavanja tih funkcija, kreira se objekt Dodi na predefinirani koordinatama.



Sl. 4.3: Događaj za kreiranje glavnog lika

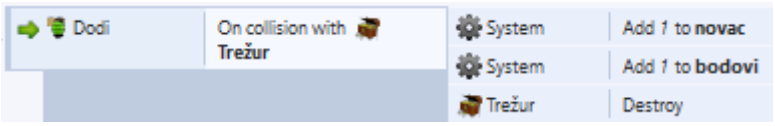
Kontrola nad uništavanjem prikazana događajem na slici 4.4 je jednostavno postavljanje vrijednosti globalnih varijabli na vrijednosti koje su nam potrebne.



Sl. 4.4: Događaj za kontrolu nad uništavanjem glavnog lika

4.1.1. Ciljevi za korisnike

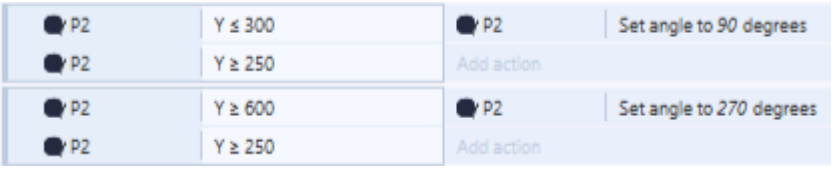
Postavljanje cilja za korisnika je jako bitno jer je cilj za korisnika ujedno i svrha korištenja računalne igre. Ostvarivanje cilja za korisnike se postiže dodavanjem novih objekata nazivom *Trežur* koji se postavlja na točno određenim pozicijama na radnom prostoru. Na slici 4.5 je prikazan događaj za kontrolu objekta *Trežur*. Događaj se sastoji od uvjeta što je kolizija glavnog lika i objekta *Trežur*. Nakon kolizije, obrađuju se funkcije dodavanja vrijednost 1 globalnim varijablama novac i bodovi, te uništava objekt *Trežur*.



Sl. 4.5: Događaji za objekt škrinja

4.2. Neprijatelji

Neprijatelji su objekti kojima je potrebno pomoću događaja kontrolirati kretanje po radnom prostoru neovisno o korisniku ili prilagođavati to kretanje ovisno o korisniku. Kreiraju se različiti objekti koji predstavljaju neprijatelje kako bi se mogle koristiti razine različitih težina. Jedno od glavnih obilježja igre je izazov koji predstavljaju neprijatelji, odnosno objekti koji smetaju korisniku da ispuni ciljeve. Prikazani na slici 4.6 su događaji za jednostavno automatsko kretanje neprijatelja, odnosno objekta *P2*. Objekt *P2* ne ovisi o korisniku nego mu se dodaje svojstvo pravocrtnog kretanja naziva *Bullet* prikazano na slici 4.7, a događajima se prilagođava tako da se može kretati samo vodoravno po x osi radnog prostora.

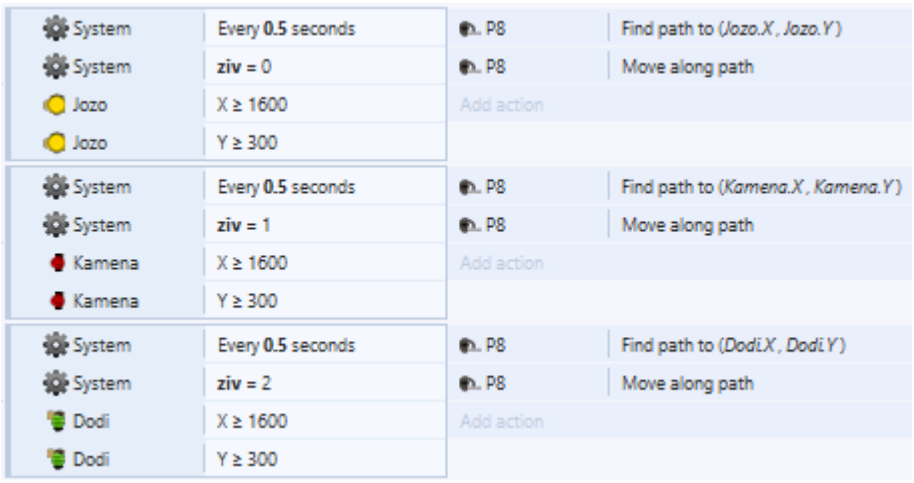


Sl. 4.6: Događaji za kontrolu kretanja objekta P2

Bullet	
Speed	300
Acceleration	0
Gravity	0
Bounce off s...	No
Set angle	Yes
Initial state	Enabled

Sl. 4.7: Svojstvo Bullet objekta P2

Neki neprijatelji imaju mnogo složenije kretanje prikazano događajima na slici 4.8 i svojstvom *Pathfinding* na slici 4.9 za objekt P8. Svojstvo *Pathfinding* je predefinirana funkcija koja automatski kalkulira najpovoljniji put od objekta do određene točke. Događaji se uređuju da objekt za bilo koji od glavnih likova svakih 0.5 sekundi uz uvjet da je glavni lik u određenom prostoru kalkulira novi put do x i y koordinate glavnog lika i zatim se kreće tim putem.

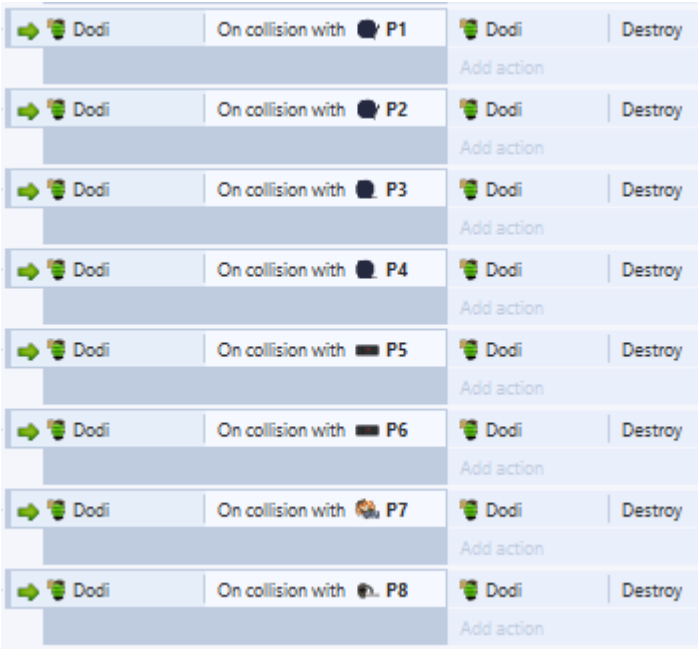


Sl. 4.8: Događaji za objekt P8

Pathfinding	
Cell size	30
Cell border	-1
Obstacles	Solids
Max speed	400
Acceleration	2000
Deceleration	2000
Rotate speed	360
Rotate object	Yes
Diagonals	Enabled
Initial state	Enabled

Sl. 4.9: Svojstvo Pathfinding za objekt P8

Zajedničko za sve neprijatelje je uništavanje objekata koji prikazuju glavne likove. Događaji prikazani na slici 4.10 imaju funkciju uništavanja glavnog lika Dodi u slučaju kolizije između glavnog lika i bilo kojeg od neprijatelja.

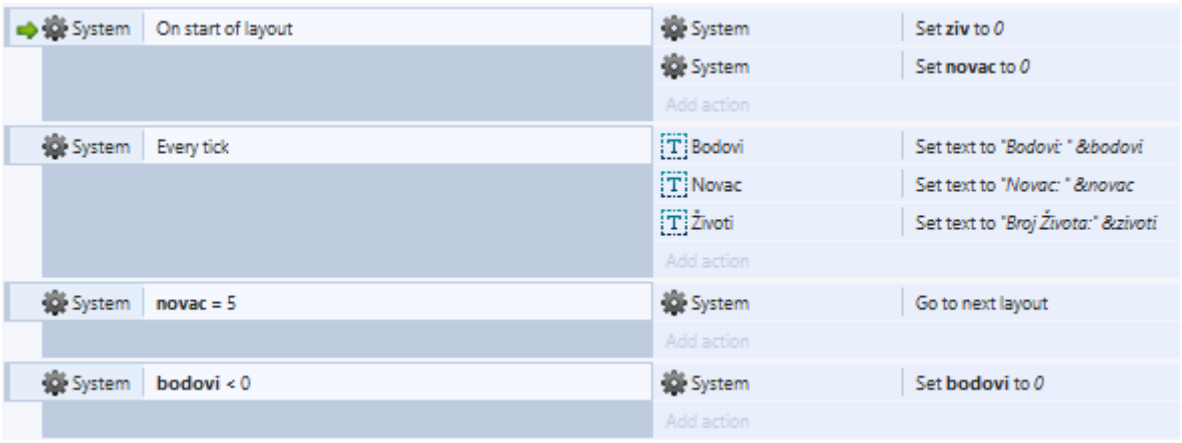


Sl. 4.10: Događaji za koliziju glavnog lika i bilo kojeg neprijatelja

4.3. Razine

Zadatak je napraviti više razina sa različitim težinama na način da svaka sljedeća težina bude teža od prethodne. Prelazak na sljedeće razine se postiže korištenjem globalnih varijabli i događaja zaslužnih za kontrolu prelaska između razina. Globalne varijable služe isključivo kao spremnik informacije i inicijalno se postavljaju na vrijednost nula.

Na slici 4.11 su prikazana četiri ključna događaja zaslužnih za kontrolu prelaska između razina. Prvi događaj postavlja vrijednost globalne varijable novac na nula početkom svake razine. Korisnik ispunjavanjem vlastitih ciljeva u igri povećava tu varijablu, a drugim događajem se na zaslonu prikazuju vrijednosti globalnih varijabli koje su bitne za korisnika. Treći događaj je zaslužan za prelazak na sljedeću razinu uz uvjet da je korisnik ispunio vlastiti cilj razine, odnosno povećao vrijednost globalne varijable na pet. Zadnji događaj služi isključivo za kontrolu nad greškama, odnosno ne dopušta da vrijednost globalne varijable pređe u negativno područje.



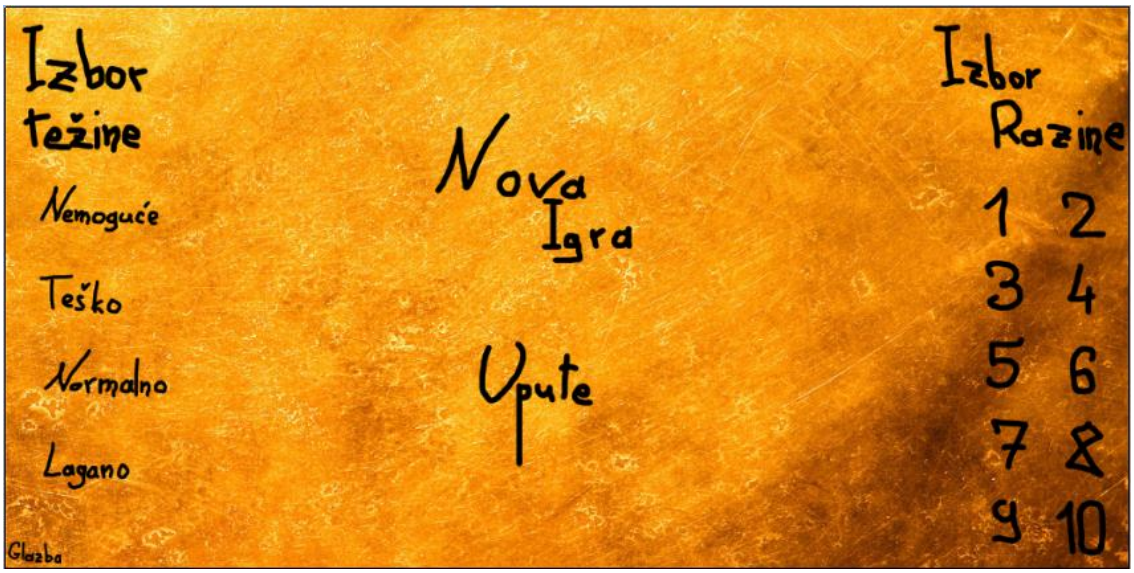
Sl. 4.11: Događaji za prelazak razina

Različite težine za razine su postignute jednostavnim raspoređivanjem objekata po radnom prostoru i mijenjanjem svojstava svih tih objekata kako bi se dobio efekt težine.

4.4. Glavni izbornik

Zadnja faza razvoja igre je kreiranje glavnog izbornika. Glavni izbornik služi kako bi korisnik imao dodatnih opcija pri igranju. Glavni izbornik sa potrebnim objektima je prikazan na slici 4.12.

Prvobitno se kreira objekt Nova igra i jednostavan događaj koji omogućuje da se klikom miša na taj objekt pređe na prvu razinu. Potrebno je povezati sve razine sa glavnim izbornikom, a to se postiže tako da se kreiraju objekti koji predstavljaju razine. Za svaki od tih objekata su napravljeni događaji koji klikom miša omogućuj prijelaz na željenu razinu. Za prilagodbu korištenja igre po želji dodana je opcija izbora težine. Izbor težine se postiže na način da se dodaju objekti koji predstavljaju različite težine. Tim objektima se zatim dodaju događaji koji izmjenjuju parametre neprijatelja i tako se dobije efekt težine za sve razine.



Sl. 4.12: Izgled glavnog izbornika

Nakon što se poveže glavni izbornik sa razinama, potrebno je napraviti i obrnuto, odnosno povezati razine sa glavnim izbornikom. To se postiže uvođenjem pauze tijekom korištenja igre. Pauza prikazana na slici 4.13 zaustavlja sve radnje i korisniku nudi razne opcije detaljnije opisane u sljedećem odlomku.



Sl. 4.13: Pauzirana razina

Na slici 4.14 su prikazani događaji zaslužni za postizanje pauze. Prvi događaj zaustavlja sve radnje, odnosno korištenjem tipke *Escape* se vrijeme izvođenja postavlja na 0 i kreiraju se objekti sa različitim mogućnostima. U drugom događaju je postavljen uvjet za klik na objekt Nastavi, a

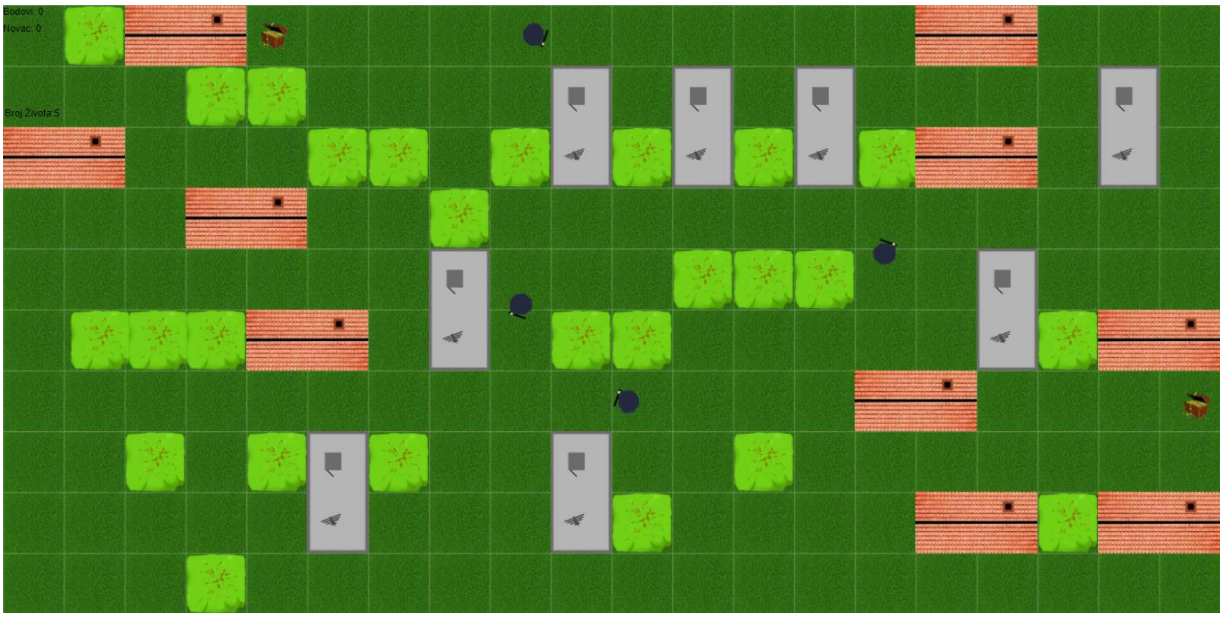
funkcija briše sve kreirane objekte i igra se nastavlja. Treći događaj je za ponovno pokretanje razine, odnosno klikom miša na objekt Ponovi razinu se obavljaju funkcije koje brišu sve kreirane objekte i ponovno pokreću razinu. Zadnji događaj je za objekt Izlaz što je prelazak u Glavni izbornik. Uz uvjet da se klikne objekt Izlaz, funkcije brišu sve kreirane objekte i omogućavaju prijelaz u glavni izbornik.



Sl. 4.14: Događaji za pauzu

4.5. Opis igre

Pokretanjem igre otvara se Glavni izbornik. Na glavnom izborniku se nalaze gumbi za: pokretanja nove igre, izbora težine, izbora razine, upute i glazbu. Pokretanjem igre se pokreće prva razina na kojoj se nalaze neprijatelji i škrinje. Na svakoj razini se automatski broje bodovi ostvareni igranjem. Moguće je stvoriti tri različita glavna lika, a svaki ima posebne mogućnosti. Prvi glavni lik imena Jozo ima najmanju brzinu, ali može stvarati objekt zamku i tako uništiti neprijatelje. Drugi lik Kamena ima za 100 piksela/sekundi veću brzinu od Joze i može usporiti vrijeme odvijanja igre. Zadnji lik Dodi najvećom brzinom ima mogućnost povećavanja te brzine dvostruko i dvostrukog smanjivanja veličine. Prikaz prve razine se nalazi na slici 4.15.



Sl. 4.14: Prva razina

Svaka razina ima različite neprijatelje i različitu količinu škrinja koje je potrebno pokupiti. Najzahtjevnija i posljednja razina je razina deset. Kako bi igra bila mnogo zahtjevnija, nekim neprijateljima je dodana mogućnost pucanja, nekima i mogućnost ciljanja glavnog lika. Tako je na desetoj razini kombinacija svih neprijatelja. Za posebne opcije postoji mogućnost pauziranja igre što omogućuje dodatne opcije poput izlaska u glavni izbornik i ponovno ponavljanje trenutne igre. Za igrača je posebno dodana mogućnost od pet života na svakoj razini, a količina preostalih života je cijelo vrijeme prikazana na ekranu. Tako da iako ne postoji mogućnost spremanja trenutnog napretka, moguće je ponavljati razinu neodređenu količinu puta. Na kraju ako igrač uspije preći zadnju razinu, pojavljuje mu se konačan broj ostvarenih bodova.

5. ISPITIVANJE

Glavna zadaća testiranja igre je otkrivanje programskih grešaka, ali to većinom radi sam programer tako da je pažnja usmjerena na grupno testiranje koje daje rezultate o programskom ostvarivanju ciljeva za izradu računalne igre. Za testiranje je korišteno 10 ispitanika koji su ocijenili igru za pojedina područja izrade. Ocjenjuje se područje funkcionalnosti igre, greške koje programer nije uočio, kompleksnost izrade i originalnost programskog ostvarivanja. Pojedina područja su ocijenjena ocjenama od 1 do 10, a rezultati su prikazani tablicom 5.1.

	Funkcionalnost	Greške	Kompleksnost	Originalnost	Konačna ocjena
Ispitanik 1	10	9	5	8	8
Ispitanik 2	8	8	7	8	7.8
Ispitanik 3	9	9	6	7	7.8
Ispitanik 4	9	6	7	7	7.3
Ispitanik 5	9	10	8	10	9.3
Ispitanik 6	10	10	10	10	10
Ispitanik 7	9	9	4	8	7.5
Ispitanik 8	10	10	9	10	9.8
Ispitanik 9	8	10	7	9	8.5
Ispitanik 10	10	10	8	8	9
Prosjek	9.2	9.1	7.1	8.5	8.5

Tablica 5.1.: Testiranje: ocjene ispitanika

Funkcionalnost prikazuje programsko ostvarivanje kontrole nad likovima. Svojstvo funkcionalnosti govori o efikasnosti korištenja ponuđenih alata i primjenjivanje tih alata za ostvarivanje postavljenih ciljeva. Greške (eng. *Bug*) su obrnuto svojstvo funkcionalnosti, odnosno ukazuju na sve ne efikasne načine korištenja ponuđenih alata. Ukazuje na sve nedostatke programskog sadržaja i na sve dijelove programskog izvođenja koji su nepravilno ostvareni. Kompleksnost ukazuje na uspješnost ostvarivanja programskog sadržaja sa ciljem prilagodbe za korisnike. Programski sadržaj mora biti usklađen i pravilno korišten kako bi prilagodba za korisnika bila što efikasnija. Originalnost je osvrt na jedinstvenost korištenja mogućnosti dostupnih programskih alata.

6. ZAKLJUČAK

Ideja je bila napraviti 2D interaktivnu igru za jednog igrača. Na temelju ideje je napravljen model korištenjem programa Construct 2. Pri izradi se koriste događaji, odnosno već napravljene funkcije u C++ i JavaScript jeziku. Svaki događaj ima svoju namjenu za ostvarivanje određenog rezultata. Događaji upravljaju objektima, a objekti su sredstva pomoću kojih je napravljen sam model igre. Izrada se postiže korištenjem i kombiniranjem svih elemenata programskog paketa i kombiniranjem znanja objektno orijentiranog programiranja. Sastoji se od dvanaest slojeva kojima se kontrolira pomoću 1331 događaja.. Za vizualni prikaz elemenata igrice korišteni su preuzeti sadržaji poput glazbe [4] i pozadinskih slika [2], a ostali elementi su napravljeni pomoću programa Paint.net [3]. Igra se temelji na HTML5 platformi koja omoguće brzo testiranje i prilagođavanje na pogreške. Pomoću povratne informacije testiranja je prilagođen model za što bolju funkcionalnost. Za prilagodbu po potrebi je korišten JavaScript jezik koji omogućuje pokretanje na Windows, Linux i Mac operacijskim sustavima.

LITERATURA

[1] Construct 2, https://www.scirra.com/construct2 (pristupljeno 10.03.2017)

[2] Clipart slike, https://www.google.hr/search?q=clipart&client=opera&hs=1AP&tbm=isch&tbo=u&source=univ&sa=X&ved=0ahUKEwj4i4qpoNzVAhWDDxoKHSNdDJoQsAQILg&biw=1366&bih=696 (pristupljeno 18.07.2017)

[3] Paint.net, https://www.getpaint.net/index.html pristupljeno (18.07.2017)

[4] Glazba, http://freesound.org/search/?q=bullet+fall&f=&s=score+desc&advanced=0&g=1 pristupljeno (18.07.2017)

[5] https://www.scirra.com/tutorials/37/beginners-guide-to-construct-2 pristupljeno (10.03.2017)

SAŽETAK

Cilj završnog rada je izrada 2D interaktivne igre. Igra je napravljena u programu Construct 2. Construct 2 je program koji ne zahtjeva dodatno predznanje jer ne zahtjeva pisanje koda. Osnove Construct 2 programa je vrlo lako za naučiti preko uputa na službenoj stranici programa. Iako nije potrebno pisanje programa, izrada igre zahtjeva mnogo vremena i kreativnosti. Sve što se nalazi u igri je napravljeno ili obrađeno nakon preuzimanja u programu Paint.net, a glazba je preuzeta. Za lakše snalaženje i kompleksnije iskustvo igranja, igra je podijeljena na 12 različitih prostora i ima 12 različitih listi događaja. Od tih 12 prostora i listi događaja, dva su za glavni izbornik i kraj igrice, a ostatak za razine. Glavni izbornik omogućuje kretanje po svim razinama i prostorima, osim kraju igrice kojemu je moguće pristupiti prelaženjem zadnje razine. Za prelazak razine potrebno je kontrolirati likove i izbjegavajući sve neprijatelje pokupiti sve škrinje na razini.

Ključne riječi: 2D igra, Construct 2, Paint.net, razina, lopov, policajac, škrinja.

ABSTRACT

Title: 2D interactive game

The goal of final paper is creating 2D interactive game. Game was made in program Construct 2. Construct 2 is a program which does not require additional knowledge in programing because it does not require writing code. Basics of Construct 2 program are very easy to learn with directions on official site of the program. Although it is not necessary to write code, making a game requires a lot of time and creativity. Everything that is in a game was made or altered after downloading in program Paint.net. For easier moving and more complex experience of playing, the game is divided on 12 different layouts and has 12 different event sheets. From these 12 layouts and event sheets, two are for game menu and game end and the rest are for levels. Menu allows moving on all layouts and levels except the game end which can only be accessed by passing the final level. For passing a level it is required to control all characters and avoiding all enemies, pick up all chests on level.

Key words: 2D game, Construct 2, Paint.net, level.

ŽIVOTOPIS

Matko Zrnić je rođen 25. lipnja 1996. godine u Virovitici. Trenutno živi u naselju Korija blizu Virovitice. Osnovnu školu Vladimira Nazora je završio u Križevcima i nakon toga upisao srednju školu Gimnaziju Petra Preradovića u Virovitici. U srednjoj školi se počeo baviti Body-building-om i biciklizmom. Nakon završetka srednje škole, upisuje se u Sveučilište Josipa Jurja Strossmayera u Osijeku, Elektrotehnički Fakultet Osijek, smjer računarstvo. Na fakultetu znatno proširuje znanje o računalima i stranim jezicima. Zbog kreativnosti radi na programiranju i pokazuje interese u izradi računalnih igara.

PRILOZI

P1 Programska realizacija nalazi se u .rar datoteci.