

# 2D PixelArt RPG igra u Unity Game Engineu

---

**Posavec, Stjepan**

**Undergraduate thesis / Završni rad**

**2019**

*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://um.nsk.hr/um:nbn:hr:200:652106>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-08-26**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

**Preddiplomski studij računarstva**

**2D PixelArt RPG igra u Unity Game Engineu**

**Završni rad**

**Stjepan Posavec**

**Osijek, 2019.**

## Sadržaj

1. UVOD .....	1
1.1. Zadatak završnog rada .....	2
2. KORIŠTENE TEHNOLOGIJE.....	3
2.1. Unity .....	3
2.2. GIMP .....	4
2.3. LMMS .....	5
3. RAZVOJ IGRE .....	6
3.1. Tematika .....	6
3.2. Likovi.....	6
3.2.1. Glavni muški lik .....	6
3.2.2. Glavni ženski lik.....	7
3.2.3. Protivnici .....	8
3.2.4. Likovi kojima ne upravlja igrač – NPC .....	9
3.3. Objekti .....	10
3.4. Grafičko sučelje .....	11
3.5. Grafika .....	12
3.6. Animacije.....	13
3.7. Napredovanje lika.....	15
3.7.1. Iskustvo – engl. Experience .....	15
3.7.2. Životni bodovi – engl. Health points.....	16
3.7.3. Čvrstoća obrane – engl. Strength .....	17
3.7.4. Snaga duha – engl. Spirit.....	18
3.7.5. Kritični udarci – engl. Critical strike.....	19
4. ZAKLJUČAK .....	20
LITERATURA.....	21
POPIS SLIKA .....	22

SAŽETAK.....	23
ABSTRACT .....	24
ŽIVOTOPIS .....	25

## 1. UVOD

Tema završnog rada je igra koja se temelji na igranju uloga (*engl. Role-Playing Game*) izrađena u Unity-u. Inspiracija za igru dobivena je igranjem Blizzardove igre Diablo. Svaka igra igranja uloga se temelji na izvršavanju određenih zadataka (*engl. Quest system*), mehanici borbe (*engl. Combat system*), napredovanju u igru pomoću kojeg razvijamo i osnovne mogućnosti našeg lika u igri (*engl. Progress system*) i drugi. Ovaj završni rad najviše je usmjeren prema vremenskom razvijanju lika u igri; neki od primjera su postupno vraćanje životnih bodova glavnom liku (*engl. Health regeneration*), povećavanje obrane liku (*engl. Strength*), povećavanje jačine udarca (*engl. Damage*), povećavanje mogućnosti za kobni udarac (*engl. Critical chance*), povećavanje jačine kobnog udarca (*engl. Critical damage*) i drugi. Završni rad je napravljen pomoću više programa. Glavni program za izradu igre bio je Unity. Za izradu modela likova i okoliša korišten je GIMP. Dok je za izradu glazbe korišten LMMS. Također, za pisanje koda u C#, koji je potreban za izradu igre, korišten je Microsoft Visual Studio.

## **1.1. Zadatak završnog rada**

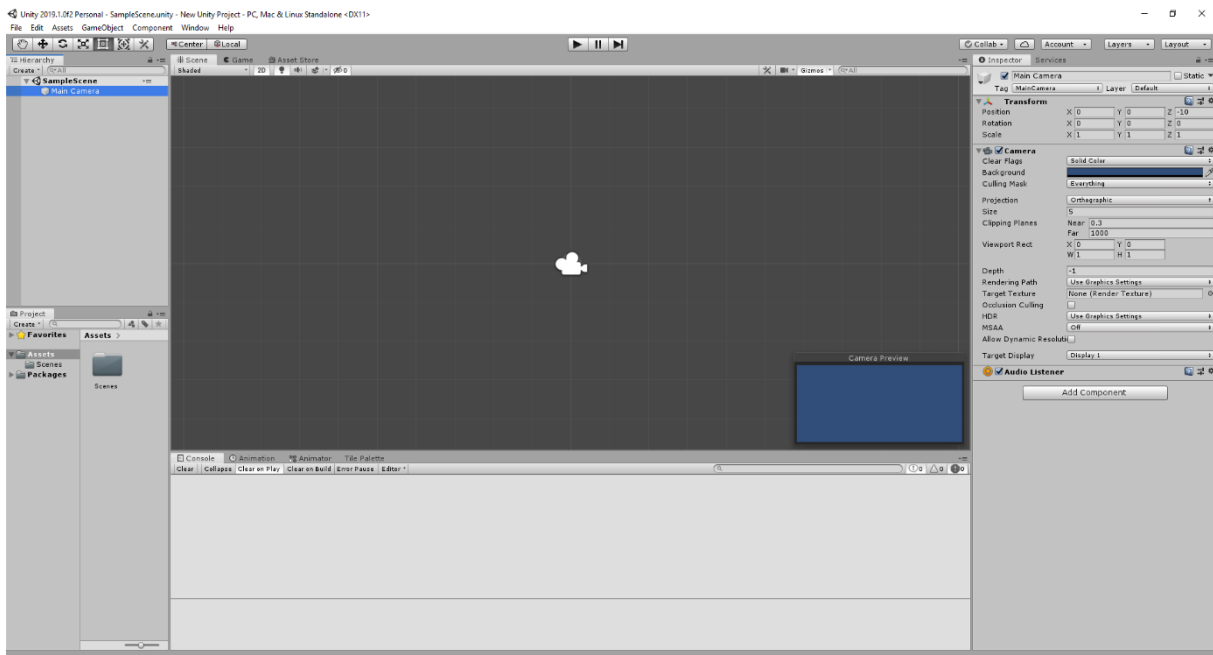
Zadatak završnog rada je napraviti 2D RPG (*engl. Role-playing game*) igru pomoću modela (likova) koji su napravljeni PixelArt tehnikom. Igra treba sadržavati segmente koji su karakteristični za igre igranja uloga kao što su: mehanika borbe, izvršavanje zadataka i drugo. Upotrebljena predznanja u izradi završnog rada su: poznavanje okruženja za rad u Unity game engine-u, rad u GIMP-u pomoću kojega izrađujemo modele, rad u LMMS-u pomoću kojega je izrađena glazba, te programiranje u programskom jeziku C#.

## 2. KORIŠTENE TEHNOLOGIJE

U ovome poglavlju pojašnjeni su alati koji su korišteni za izradu igre. Oni su: Unity, GIMP i LMMS.

### 2.1. Unity

Unity<sup>1</sup> je višeplosni game engine koji se koristi za izradu igara. Postoji 27 platformi koje podržavaju Unity, neke od njih su: Windows, OSX, Linux, Android, iOS, Playstation i mnoge druge platforme. Unity je program razvijen od strane Unity Technologies prije 14 godina. Unity osim u industriji video igara također koriste industrije kao što su: filmska industrija, automobilska industrija, arhitektura i druge. Unity koristi programski jezik C#, Javascript i Boo. U izradi završnog rada korišten je samo C#. U Unity-u se mogu razvijati 3D i 2D igre ovisi o tome što korisnik želi. U sklopu s Unity-em za pisanje programa korišten je Microsoft Visual Studio. Unity se može pronaći na službenoj stranici. Program je potpuno besplatan dok se koristi u edukacijske svrhe. Ako ga se želi koristiti u komercijalne svrhe potrebno je kupiti licencu od Unity Technologies.



Slika 1. Izgled Unity game engine-a

<sup>1</sup> Prema [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))

## 2.2. GIMP

GIMP<sup>2</sup> je besplatna aplikacija koja se koristi za stvaranje i obradu grafike. Pomoću programa se može crtati, mijenjati veličinu slike, urediti sliku i druge mogućnosti. GIMP su izvorno objavili Spencer Kimball i Peter Mattis kao General Image Manipulation Program. GIMP podržava 8, 15, 16 i 24-bitne dubine boja, istovremeno se mogu uređivati slike. Podržava razne formate od kojih su neki JPEG, GIF, PNG i drugi. U završnom radu GIMP je korišten za izradu modela likova i izgled okoliša. Također, GIMP podržava i skripte koje su napisane u Script-Fu, C, Scheme, Python i Pearl programskim jezicima.



Slika 2. Izgled GIMP-a

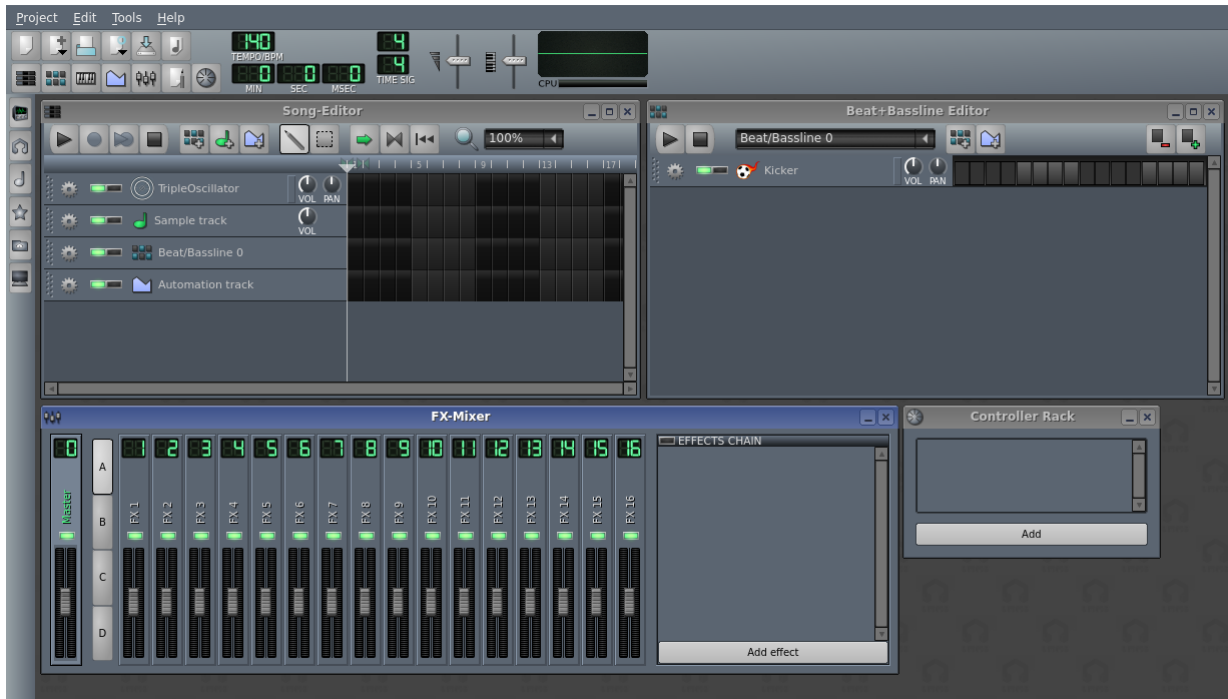
---

<sup>2</sup> Prema <https://hr.wikipedia.org/wiki/GIMP>



## 2.3. LMMS

LMMS<sup>3</sup> je besplatna aplikacija koja se koristi za stvaranje i obradu zvukova. Pomoću ovog programa moguće je snimati nove zvukove pomoću virtualne MIDI tipkovnice ili pomoću fizičke MIDI tipkovnice, moguće je obrađivati već postojeće zvukove i druge mogućnosti. Program podržava Linux, Windows, OpenBSD i macOS. Zahtjeva minimalno procesor od 1 GHz, radnu memoriju od 512 MB i dvokanalnu zvučnu karticu. LMMS su izvorno objavili Paul Giblock i Tobias Junghans.



Slika 3. Izgled LMMS-a

<sup>3</sup> Prema <https://en.wikipedia.org/wiki/LMMS>

### 3. RAZVOJ IGRE

#### 3.1. Tematika

Misao vodilja u igri je kako će glavni muški lik izbaviti sebe i glavnog ženskog lika (njegovu sestru) iz podzemlja gdje su zarobljeni i okruženi brojnim neprijateljima. Budući da se igra zasniva na igranju uloga glavni zadatak glavnog lika je pobijediti sve protivnike tijekom igre tj. oslobađanja. Na početku igre protivnici su slabiji jer je i igrač slabiji, tj. rangirani su po jačini igrača. No kako igrač napreduje tako napreduju i njegovi neprijatelji. U konačnici, glavni lik može spasiti sebe i ženskog lika samo ukoliko pobijedi i najjačeg protivnika koji ima moć multipliciranja.

#### 3.2. Likovi

U ovoj igri postoji nekoliko likova. Jedan s kojim upravlja naš korisnik, odnosno igrač tzv. glavni lik. Uz glavnog lika postoji i ženski (interaktivni) lik kojem ne upravlja igrač nego je samo s njim u interakciji. Također postoje i neprijatelji glavnog lika.

##### 3.2.1. Glavni muški lik

Glavni muški lik, Ivan, nakon što je nepoznati neprijatelj zarobio njega i njegovu sestru, te ih odveo u podzemlje, svu svoju snagu usmjerava ka tome da ih izbavi iz podzemlja.

U borbi mu stoje na raspolaganju isključivo njegove šake, te mu se snaga pojačava isključivo ubijanjem protivnika. Osim što se povećava čvrstoća obrane, snaga duha i jačina udarca, povećavaju se također i životni bodovi, te svakom eliminacijom protivnika dobije određeni broj iskustvenih bodova.



*Slika 4. Izgled glavnog muškog lika*

### 3.2.2. Glavni ženski lik

Glavni ženski lik, sestra Marija ne sudjeluje u borbi, ali ima vrlo važnu ulogu u samoj igri jer usmjerava igrača, naime, ona ima povremene vizije koji otkrivaju što točno njen brat Ivan mora odraditi kako bi ih izbavio iz podzemlja u kojem su zarobljeni.



*Slika 5. Izgled glavnog ženskog lika*

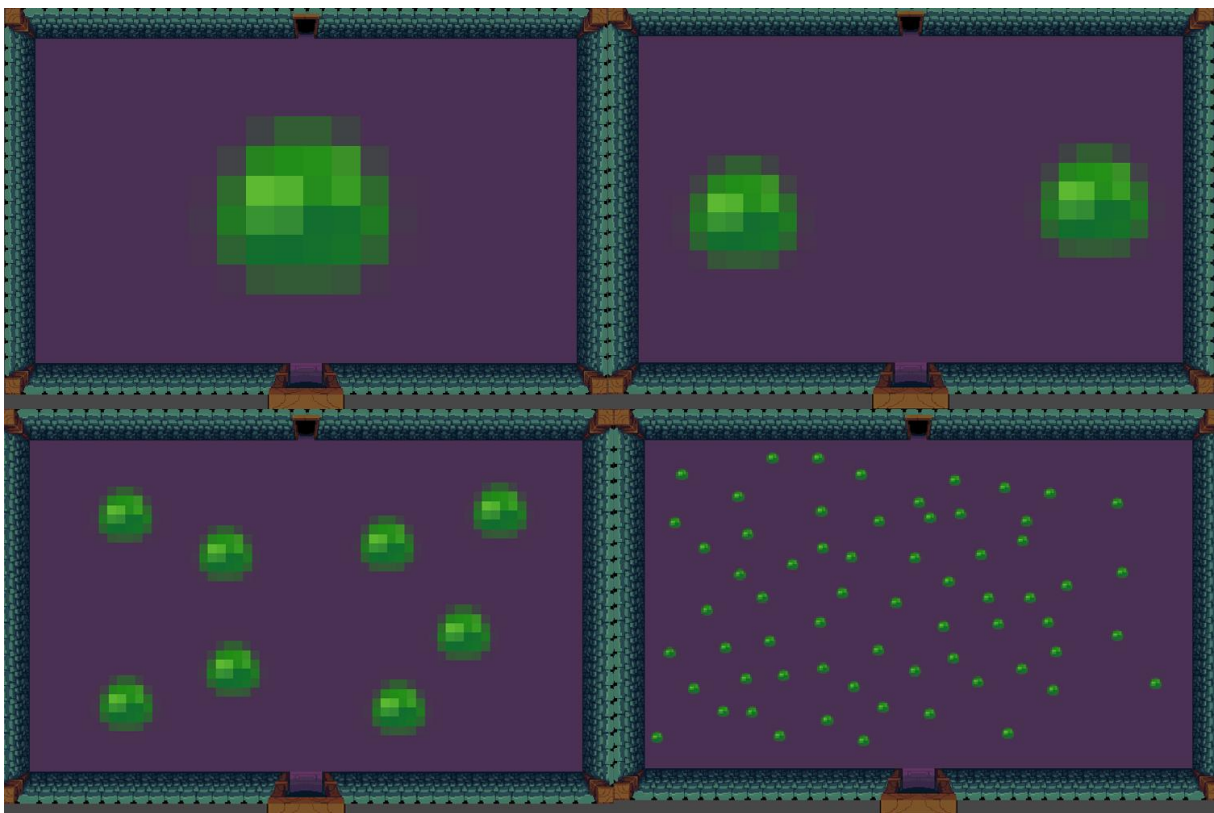
Npr. na samom početku daje informaciju koliko je točno protivnika potrebno ubiti kako bi se zaradio ključ pomoću kojeg glavni lik može otići u sobu u kojoj prikuplja različita pojačanja (*engl. buff room*), te se tako ojačati i daje informaciju o tome da kada igrač dosegne razinu 10 može otići u borbu protiv glavnog neprijatelja.



*Slika 6. Informacije koje igraču daje glavni ženski lik*

### 3.2.3. Protivnici

Kroz igru se protežu varijacije jednog protivnika. To je „Slime“. Oni su u početku malih dimenzija, dok je glavni „Slime“ puno veći od ostatka. „Mini Slime“ umire kada izgubi životne bodove, dok „Big Slime“, koji predstavlja glavnog neprijatelja, kada izgubi životne bodove umire i pretvara se u dva manja „Slime-a“, od kojih se svaki pretvori u četiri još manja, od kojih se svaki u konačnici pretvori u osam još manjih „Slime-ova“ čije se dimenzije podudaraju s onima s početka igre. Također, ovisno o veličini protivnika se mijenjaju i njegove karakteristike. Tako onaj koji je veći ima više životnih bodova i jači napad od onog koji je manji.



*Slika 7. Mehanika glavnog protivnika („Big Slime“)*

### 3.2.4. Likovi kojima ne upravlja igrač – NPC

Likovi kojima igrač ne upravlja su općenito likovi koji su isprogramirani da reagiraju na određene akcije lika s kojim igrač upravlja. Kao što su određeni dijalozi, zadaci, i mnoge druge opcije. U ovome slučaju glavni ženski lik je lik kojim igrač ne upravlja. Taj lik je isprogramiran tako da u trenutku kada mu se približi lik s kojim igrač upravlja započne dijalog kojim upućuje igrača što bi trebao napraviti kako bi mogao dalje nastaviti s igrom.



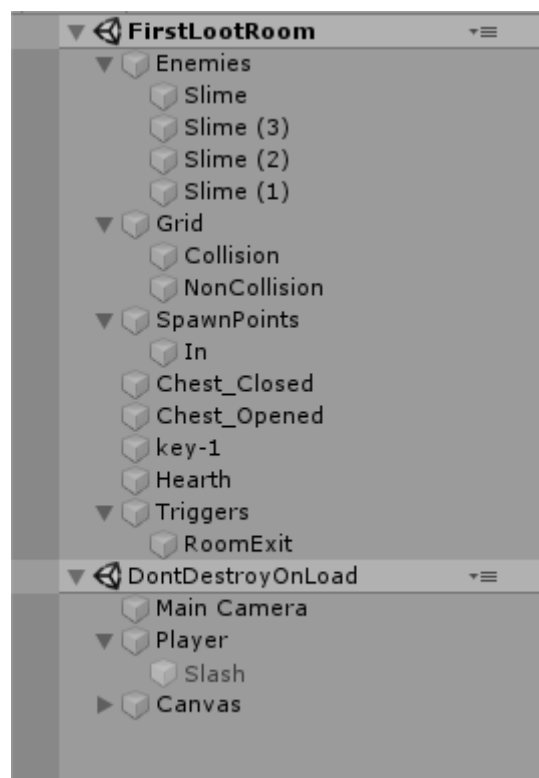
*Slika 8. Zeleni okvir prikazuje zonu početka dijaloga*

### 3.3. Objekti

Prilikom izrade projekta korišteni su brojni objekti. Najpotrebniji objekti pri izradi igre su „Game Objects“ koji u sebi mogu sadržavati sve segmente koji su potrebni u izradi igre. Neki od segmenata su programski kod koji omogućava manipuliranje nad objektom, sliku odnosno „Sprite“ određenog lika. Također, „Game Object“ može biti i u slučaju ovoga projekta ključić kojega igrač može pokupiti ili vrata koja igrač treba otključati, te mnogi drugi objekti.

Također, vrlo važan objekt je „Camera“ koji omogućava vidljivost igre igraču tj. vidljivost svih komponenata igre kao što je područje u kojem igrač igra, lika kojim igrač igra, te izbornika u slučaju pauziranja igre. Također, i na taj objekt je moguće priključiti programski kod, u ovome slučaju je to kod koji izvršava zadaću praćenja lika s kojim igrač upravlja.

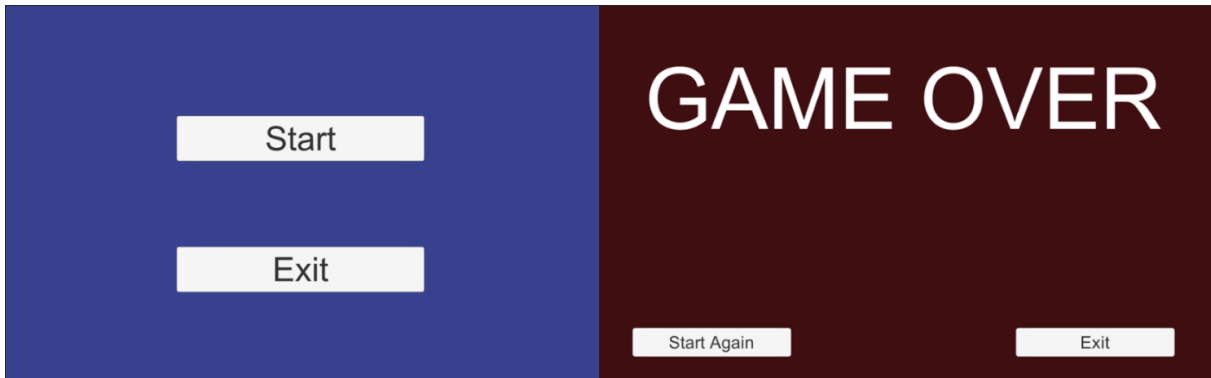
Osim toga u grafičkom sučelju se pojavljuju i objekti. Neki od njih su „Button“ objekti koji omogućavaju korištenje grafičkog sučelja, jer klikom na taj objekt kod izvršava određenu zadaću, pokreće igru, izlazi iz igre ili resetira igru.



Slika 9. Objekti korišteni za izradu „buff“ sobe

### 3.4. Grafičko sučelje

Grafičko sučelje omogućava igraču da pokrene igru, pauzira ju ili kada njegov lik umre resetira igru. Osim toga igrač uvijek ima opciju da napusti igru. Grafičko sučelje se sastoji od gumba koji reagiraju na pritisak miša na njih. Pritiskom na tipku „Start“ igra se pokreće. Pritiskom na tipku „Exit“ igra se gasi. Pritiskom na tipku „Start Again“ igra se pokreće ponovno. Također, igru je moguće pauzirati pritiskom na tipku „Escape“ na tipkovnici, te ju ponovno pokrenuti pritiskom na tipku „Start“.



*Slika 10. Grafičko sučelje (lijevo – izbornik, desno – igra nakon izgubljenog života)*

Nadalje, igrač tijekom igranja ima uvid u grafičko sučelje koje mu pokazuje određene informacije. Kao što su: životni bodovi, iskustveni bodovi, razina igrača, broj ubijenih protivnika prije nego dobije ključ te broj trenutnih ključeva.

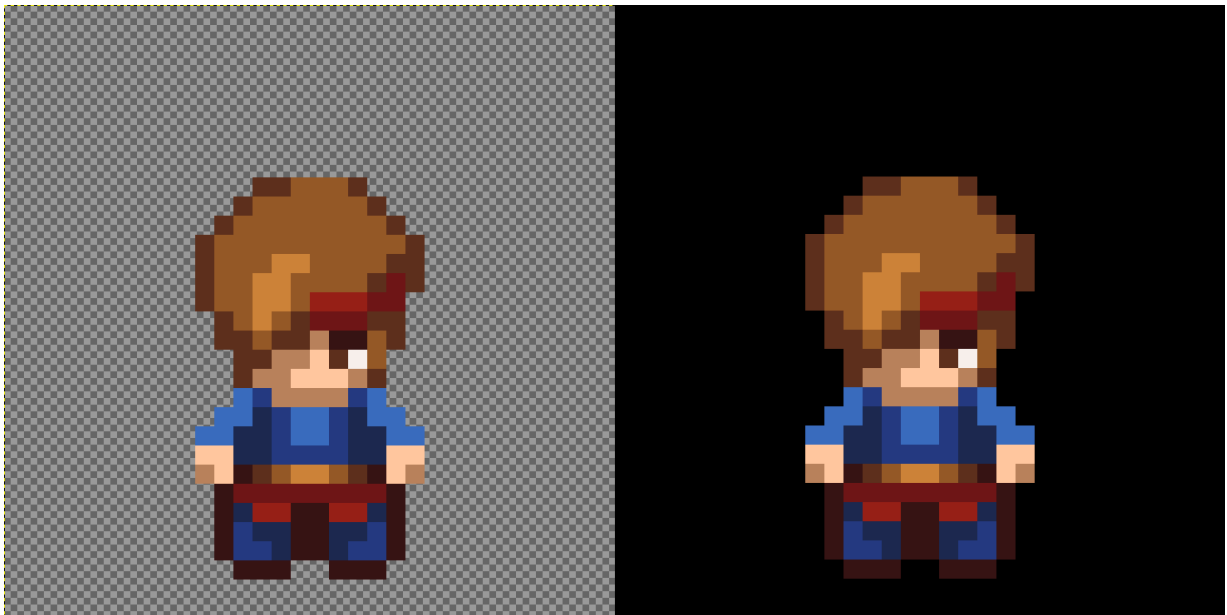


*Slika 11. Grafičko sučelje koje pruža informacije o glavnom liku tokom igre*

### 3.5. Grafika

Pri izradi modela likova (*engl. Sprites*) i okoliša kombinirani su vlastoručno izrađeni modeli likova korištenjem programa GIMP, a dio je preuzet s Interneta kao besplatan sadržaj dan na slobodno korištenje.

Format svih slika je PNG koji omogućava transparentnu (prozirnu) pozadinu. Taj format je korišten upravo iz tog razloga jer je u igri potrebno imati prozirnju pozadinu na modelima likova kako svi likovi ne bi bili strogo pravokutni, odnosno kako bi mogli imati likove prirodnog izgleda.

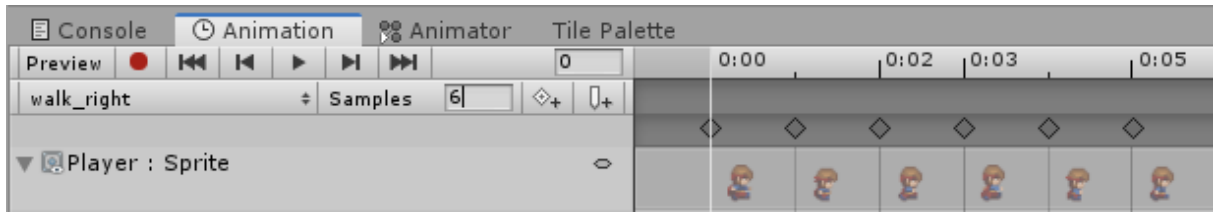


*Slika 12. Usporedba transparentne pozadine i pune (solid) pozadine*



### 3.6. Animacije

U igri je bilo potrebno animirati likove. Glavni lik ima više skupina animacija. Posjeduje animaciju koja prikazuje hodanje, animaciju koja prikazuje napadanje, animaciju trčanja i u konačnici i animaciju umiranja.



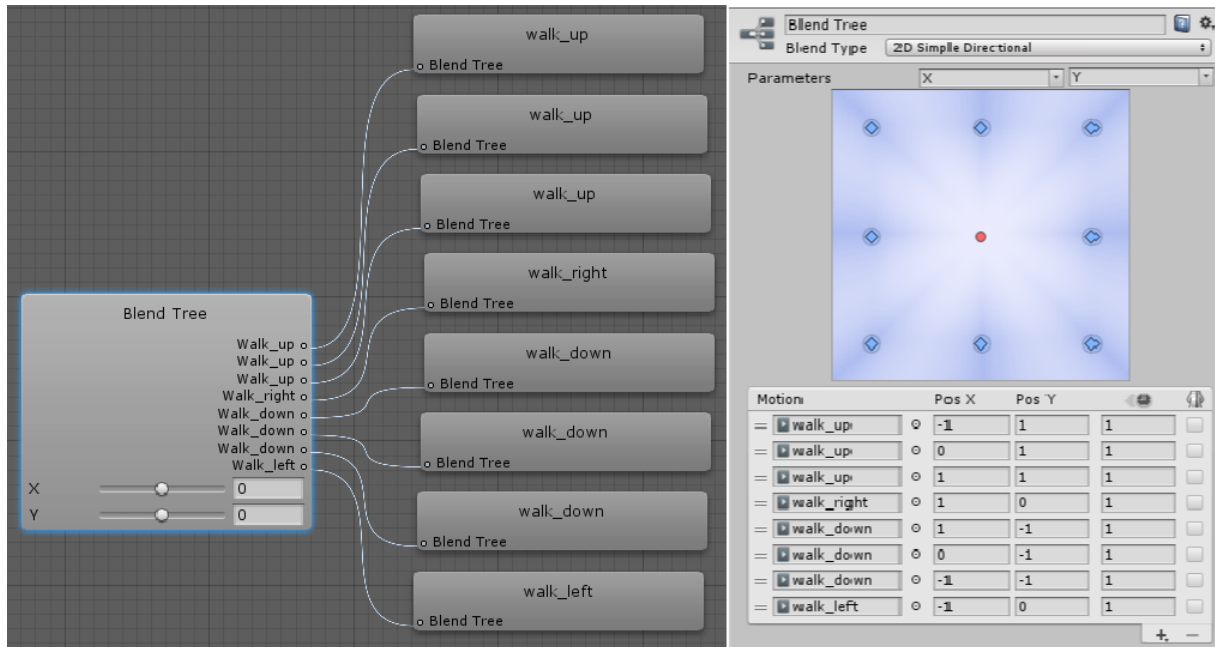
*Slika 13. Izgled animacijskog prozora u Unity-u*

Općenito, animacije služe kako bi likovima dali određene ljudske karakteristike kako bi se igrači mogli poistovjetiti sa stvarnim životnim situacijama. Za animaciju lika potrebno je više slika od koje svaka pokazuje slijedeći izgled lika, te pomoću izmjene tih sličica sustav kreira animaciju lika npr. privid kretanja. Ta mogućnost je ugrađena u Unity, a animacije se odrađuju direktno u samom programu.



*Slika 14. Skupina sličica za kretanje lika u desno*

Animator je posebni dio koji se koristi u Unity-u kako bi povezali određene vrijednosti unutar igre sa određenim animacijama, te uz animator pomoću koda programa povezujemo određenu komandu sa animacijom odnosno vrijednosti. Npr. tipke W, A, S, D koristimo kako bi promijenili položaj glavnog lika te pokrenuli izvršavanje skupine animacija za kretnju.



Slika 15. Izgled animatora s popratnim vrijednostima koje upravljaju animacijama

### 3.7. Napredovanje lika

Najveći naglasak stavljen je na segmentu napredovanja lika. U to ubrajamo formulu po kojoj se povećavaju bodovi potrebni za podizanje nivoa igrača. To je formula na kojoj se zasnivaju svi ostali segmenti igre, jer kako igrač podigne nivo svog lika tako se mijenjaju (podižu/smanjuju) ostali segmenti koji su vezani uz lika, a oni su: životni bodovi, čvrstoća obrane, snaga duha (pozitivna ili negativna), jačina udaraca te sposobnost izvršavanja kritičnog udarca kao i njegova jačina.

#### 3.7.1. Iskustvo – engl. Experience

Iskustveni bodovi, općenito u igrama, služe kako bi igrač vremenom napredovao kroz igru. Tako i u ovom završnom radu iskustveni bodovi pokazuju igračevo napredovanje.

Kako bi igra bila što više prilagođena napredovanju igrača tako protivnici povećavaju broj iskustvenih bodova koje igrač dobije kada ih pobijedi. Iz priložene formule vidljivo je kako se ponaša broj iskustvenih bodova koje igrač dobije kada pobijedi protivnika u odnosu na njegovu trenutnu razinu.

$$y = 10 + e^{\frac{x}{2}}$$

U igri postoje četiri sobe koje igrač otključa prethodno osvojenim ključem. Kada ubije sve protivnike u toj sobi dobije ključ za kovčeg koji se u njoj nalazi. U kovčegu se nalazi pojačanje (*engl. buff*) kojim kada ga uzme unapređuje svoje sposobnosti. Tako u jednoj od soba kao pojačanje dobije 1/3 broja iskustvenih bodova trenutne razine igrača.

Kako bi matematički što preciznije odredili broj potrebnih iskustvenih bodova primjenjujemo formulu koja najbolje odgovara težini igre, te kao što je vidljivo iz formule kao bazni broj potrebnih iskustvenih bodova za slijedeću razinu (prelazak sa razine 1 na razinu 2) koristimo otprilike 23 iskustvena boda, koji se eksponencijalno povećavaju sve do posljednje, 10 razine za koju je potrebno oko 22.000 iskustvenih bodova.

$$y = 20 + e^x$$

Kao zaključak odnosa broja potrebnih iskustvenih bodova te broja iskustvenih bodova koje igrač dobije kada pobijedi protivnika zaključujemo: kako igrač podiže svoju razinu on mora ubiti sve više protivnika kako bi došao na slijedeću razinu.

```

2 references
public void LevelUp()
{
    currentLevel++;
    currentXP = 0;
    currATK = levelATK[currentLevel];
    tempCurrATK = levelATK[currentLevel];
    currDEF = levelDEF[currentLevel];
    currHP = levelHP[currentLevel];
    currSpirit = levelSpirit[currentLevel];
    playerHealth.maxHP = currHP;
    playerHealth.SetMaxHealth();
}

1 reference
public void AddXPtoPlayer(int xp)
{
    currentXP += xp;
}

```

Slika 16. Funkcija za podizanje razine lika i funkcija za dodavanje iskustva igraču

### 3.7.2. Životni bodovi – engl. Health points

Općenito u igrama životni bodovi predstavljaju izdržljivost igrača, odnosno koliko dugo igrač može ostati živ. U trenutku kada životni bodovi dosegnu broj 0 ili manje od nule igrač je mrtav te igru mora započeti ispočetka.

Kao kod iskustvenih bodova, tako i kod životnih bodova, životni bodovi protivnika se također povećavaju u odnosu na razinu igrača. Taj odnos razine igrača i povećavanja životnih bodova protivnika možemo vidjeti iz priložene formule:

$$y = 15 + e^{1,5 \cdot \frac{x}{2}}$$

I u slučaju životnih bodova, također postoji soba u kojoj se kao pojačanje dobije 15% više maksimalnih životnih bodova igrača.

Kao i ostatak segmenata napredovanja igrača tako i ponašanje veličine životnih bodova donosimo priloženom formulom koja kao i ostale formule opisuje ponašanje veličine životnih bodova u odnosu na trenutnu razinu igrača:

$$y = 50 + \left(\frac{e}{2}\right)^{1,5x}$$

Iz gore navedenog odnosa ponašanja veličine životnih bodova igrača i protivnika možemo doći do zaključka da igrač čim postiže veću razinu igre mora udarati protivnika sve

više puta kako bi ga savladao, no isto tako i protivnik mora udariti više puta igrača, dakle odnos je proporcionalan.

### 3.7.3. Čvrstoća obrane – engl. Strength

Općenito u igrama igranja uloga (*engl. Role-play game*) segment čvrstoće obrane (*eng. Strength*) se uobičajeno koristi kao dodatak životnim bodovima. Odnosno čvrstoća obrane opisuje kolika je jaka obrana igrača izuzev broja životnih bodova. Objašnjenje ponašanja slijedi u primjeru:

Pretpostavimo da protivnik glavnom liku oduzima 10 životnih bodova sa jednim udarcem u slučaju kada je čvrstoća obrane glavnog lika jednaka nuli, no kada je čvrstoća obrane glavnog lika jednaka 5, a napad protivnika ostaje i dalje 10 u tome trenutku protivnik nanosi štetu glavnom liku u iznosu od 5 životnih bodova.

```
References
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.name == "Player")
    {
        if (damage - playerStats.currDEF <= 0)
        {
            currentDmg = (int)Random.Range(-0.1f, 1.1f);
        }
        else
        {
            currentDmg = damage - playerStats.currDEF;
        }

        collision.gameObject.GetComponent<PlayerHealthManager>().DealDamageToPlayer(currentDmg);

        var clone = (GameObject)Instantiate(damageNumber, collision.transform.position, Quaternion.Euler(Vector3.zero));
        clone.GetComponent<FloatingNumbers>().damageNumber = currentDmg;
        clone.GetComponentInChildren<Text>().color = Color.red;
    }
}
```

Slika 17. Funkcija pomoću koje protivnik nanosi štetu igraču

Slika 17. prikazuje slučaj kada igračeva čvrstoća obrane u potpunosti negira napad protivnika. Protivnik i dalje ima mogućnost nanijeti određenu štetu igraču (u ovom slučaju štetu od jednog životnog boda), ali postoji mogućnost da štetu uopće neće nanijeti. Uokvirena linija koda prikazuje kod koji izvršava prije objašnjenu situaciju.

Iz priložene formule vidljivo je ponašanje veličina čvrstoće obrane u odnosu na razinu igrača.

$$y = \left(\frac{e}{2}\right)^{1.8x}$$

### 3.7.4. Snaga duha – engl. Spirit

Općenito u igrama igranja uloga (*engl. Role-play game*) segment snage duha (*engl. Spirit*) se koristi kako bi se opisalo kojim tempom će se životni bodovi prirodno vraćati igraču. Uzmimo za primjer da igrač u datom trenutku ima 64 životnih bodova od 100 mogućih. U slijedećem trenutku igrač ako nije primio ni jedan udarac, odnosno štetu na svoje životne bodove i njegova razina je 5, će imati 69 životnih bodova od mogućih 100. Taj rezultat je dobiven na taj način da se iz priložene formule dobije da snaga duha na petoj razini iznosi oko 18. A količina životnih bodova koji se vraćaju svakih tri sekundi iznosi 25% od trenutne veličine snage duha (u ovom slučaju to je oko 5 životnih bodova).

$$y = 2 + \left(\frac{e}{2}\right)^{1.8x}$$

```
1 reference
private void HealthRegen()
{
    if(currHP < maxHP && regenCounter <= 0)
    {
        if(currHP + (int)(playerStats.currSpirit * 0.25f) > maxHP)
        {
            currHP = maxHP;
        }
        else
        {
            currHP += (int)(playerStats.currSpirit * 0.25f);
        }
        regenCounter = 5f;
    }
    regenCounter -= Time.deltaTime;
}
```

Slika 18. Funkcija koja izvršava zadaću prirodnog vraćanja životnih bodova

### 3.7.5. Kritični udarci – engl. Critical strike

Kritični udarci se, općenito u igrama sastoje od dva segmenta: mogućnosti pojave kritičnog udarca, te jačine kritičnog udarca. Kritičan udarac se najlakše može objasniti kao dosta jači udarac, odnosno napad u odnosu na jačinu normalnog napada. Općenito, pa tako i u ovoj igri mogućnosti izvođenja kritičnog udarca su na nižim razinama igrača vrlo rijetki, dok na višim razinama postaju učestaliji, međutim nikada ne prelaze 50% mogućnosti njihove pojave. U slučaju ove igre maksimalna mogućnost pojave kritičnog udarca (na zadnjoj razini igrača) iznosi 33%, što je manje od spomenutih 50%. Iz priložene formule možemo vidjeti ponašanje veličina za mogućnost pojave kritičnog udarca.

$$y = 0.045x^{0.85}$$

U ovom slučaju, kao i kod životnih bodova i iskustva, postoji soba u kojoj kao nagradu igrač dobiva 15% jači kritičan udarac na svakoj razini igrača.

Jačina kritičnog udarca je prikazana priloženom formulom. Što je razina igrača veća to je razlika između normalnog napada i kritičnog napada sve veća.

$$y = 10 + \left(\frac{e}{2.15}\right)^{2.75x}$$

#### **4. ZAKLJUČAK**

U ovome završnom radu obuhvaćena je većina segmenata potrebnih za izradu jedne računalne igre. Veći fokus je stavljen na samu pozadinu lika s kojim igrač upravlja, odnosno njegov razvitak kroz igru. Vođeno je računa o izbalansiranosti napretka igrača u odnosu na napredak neprijatelja unutar igre. Za izradu završnog rada bilo je potrebno poznavati programski jezik C# te Unity „dodatke“ tom jeziku. Uz poznavanje programskog jezika bilo je potrebno izraditi grafiku te zvukove za igru pomoću spomenutih programa. Igra još ima mjesta za doradu. Neki od ciljeva su dodavanje više različitih likova s kojim igrač može upravljati, više zadataka koje igrač mora ispuniti te otvaranje igre prema „multiplayer-u“.



## LITERATURA

1. [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))  
pristupanje svibanj 2019.
2. <https://hr.wikipedia.org/wiki/GIMP>  
pristupanje svibanj 2019.
3. <https://en.wikipedia.org/wiki/LMMS>  
pristupanje svibanj 2019.
4. Unity User Manual - <https://docs.unity3d.com/Manual/index.html>  
pristupanje veljača 2019.
5. Unity youtube official channel - <https://www.youtube.com/user/Unity3D>  
pristupanje veljača 2019.
6. Unity Beginner Tutorials -  
<https://www.youtube.com/watch?v=j48LtUkZRjU&list=PLPV2KyIb3jR5QFsefuO2RlAgWEz6EvVi6>  
pristupanje veljača 2019.
7. PixelArt Tutorial -  
<https://www.youtube.com/watch?v=RLvDDZnccQE&list=PLC6BCB8E64F315574>  
pristupanje svibanj 2019.

## POPIS SLIKA

Slika 1. Izgled Unity game engine-a .....	3
Slika 2. Izgled GIMP-a.....	4
Slika 3. Izgled LMMS-a.....	5
Slika 4. Izgled glavnog muškog lika .....	6
Slika 5. Izgled glavnog ženskog lika.....	7
Slika 6. Informacije koje igraču daje glavni ženski lik .....	7
Slika 7. Mehanika glavnog protivnika („Big Slime“) .....	8
Slika 8. Zeleni okvir prikazuje zonu početka dijaloga .....	9
Slika 9. Objekti korišteni za izradu „buff“ sobe.....	10
Slika 10. Grafičko sučelje (lijevo – izbornik, desno – igra nakon izgubljenog života) .....	11
Slika 11. Grafičko sučelje koje pruža informacije o glavnom liku tokom igre.....	11
Slika 12. Usporedba transparentne pozadine i pune (solid) pozadine.....	12
Slika 13. Izgled animacijskog prozora u Unity-u.....	13
Slika 14. Skupina sličica za kretanje lika u desno.....	13
Slika 15. Izgled animatora s popratnim vrijednostima koje upravljaju animacijama .....	14
Slika 16. Funkcija za podizanje razine lika i funkcija za dodavanje iskustva igraču.....	16
Slika 17. Funkcija pomoću koje protivnik nanosi štetu igraču .....	17
Slika 18. Funkcija koja izvršava zadaću prirodnog vraćanja životnih bodova .....	18

## SAŽETAK

Završni rad opisuje izradu igre pomoću Unity-a i programskog jezika C# uz popratne programe kao što su LMMS za zvuk i GIMP za grafiku. Igra posjeduje priču u kojoj glavni lik pokušava izbaviti sebe i sestru iz smrtonosne situacije. Kako bi postao što jači mora prvo savladavati slabije protivnike kako bi osvojio ključ koji mu omogućava ulaz u sobe u kojima dobiva nagrade pomoću koji dodatno napreduje. Kada jednom dosegne najvišu razinu svoje snage (razina 10) otvaraju mu se posebna vrata iza kojih je njegov glavni neprijatelj „Big Slime“. Mehanika koja se krije iza glavnog protivnika je takva da se iz protivnika jednom kad je ubijen stvore dva nova manja i slabija i tako sve dok se ne dođe do slučaja kada se iz jednog protivnika stvori njih 8. Igra je osmišljena da ne postoje klasične razine (*engl. level*) koje igrač prelazi nego se svi protivnici, osim zadnjeg protivnika, ravnaju po trenutnoj snazi glavnog lika.

Ključne riječi: 2D igra igranja uloga, 2D RPG, segmenti napredovanja igrača, Unity

## **ABSTRACT**

This final project describes a game development process in Unity as well as using C# programming language with an addition of utility software such as LMMS for sound editing and GIMP for graphics. The game contains a story in which the main character is trying to make his way out of a lethal situation, as well as saving his sister. In order to become stronger, he has to outstand some less strong enemies. If he does succeed in that particular task, he'll be obtaining a key which allows him an entrance to the rooms where special bonus prosperity rewards are hidden. Once he reaches the maximum level of his strength (level 10), a special door opens behind of which is his nemesis called "Big Slime". The mechanics of this particular main boss is such that once the boss is defeated, he is multiplied into two new smaller pieces of himself and repeats the same mechanic up to the point where there's 8 pieces of slimes, after killing those, the fight ends. The concept of the game is such that there are no regular levels for player to overcome. Instead, enemy in the game is scaled to the current strength of a player's character, with an obvious exception of a final boss.

Key words: 2D Role-playing game, 2D RPG, player statistics, Unity

## **ŽIVOTOPIS**

Stjepan Posavec rođen je u Osijeku, 03.09.1997. godine. Prva četiri razreda osnovne škole pohađao je OŠ Frana Krste Frankopana, a slijedeća četiri razreda OŠ Ivana Filipovića Osijek. U tom razdoblju je aktivno trenirao vaterpolo, te sudjelovao na brojnim športskim natjecanjima vezano uz vodene sportove, a također i na školskim športskim natjecanjima tijekom svog osnovno školskog obrazovanja. 2012. godine upisuje željenu Prirodoslovno matematičku gimnaziju u Osijeku te tijekom svog srednjoškolskog obrazovanja pokazuje sve više interesa za programiranjem, te sudjeluje u natjecanjima na području Informatike. Nakon završetka srednje škole upisuje Fakultet elektronike, računarstva i informacijskih tehnologija u Osijeku.