2D RPG igra s proceduralno generiranim razinama u Unityu

Umiljanović, Matej

Master's thesis / Diplomski rad

2018

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:200:047415

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Diplomski studij

2D RPG igra sa proceduralno generiranim razinama u Unityu

Diplomski rad

Matej Umiljanović

Osijek, 2018.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 12.07.2018.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Matej Umiljanović
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D 802 R, 24.09.2017.
OIB studenta:	41720690024
Mentor:	Doc.dr.sc. Časlav Livada
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Alfonzo Baumgartner
Član Povjerenstva:	Izv. prof. dr. sc. Krešimir Nenadić
Naslov diplomskog rada:	2D RPG igra s proceduralno generiranim razinama u Unityu
Znanstvena grana rada:	**Umjetna inteligencija (zn. polje računarstvo)**
Zadatak diplomskog rada:	U ovom radu potrebno je opisati osnove rada u Unity Game Engineu te način njegove primjene za izradu 2D igara. Opisati način stvaranja grafičkih i zvučnih elemenata igre te način njihova povezivanja u C# skriptama. U praktičnom dijelu rada potrebno je napraviti 2D RPG igru s proceduralno generiranim razinama u Unityu. Tehnologije: Unity
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene mentora:	12.07.2018.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 17.07.2018.

Ime i prezime studenta:	Matej Umiljanović
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D 802 R, 24.09.2017.
Ephorus podudaranje [%]:	1

Ovom izjavom izjavljujem da je rad pod nazivom**: 2D RPG igra s proceduralno generiranim razinama u Unityu**

izrađen pod vodstvom mentora Doc.dr.sc. Časlav Livada

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

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

IZJAVA

Ja, Matej Umiljanović, OIB: 41720690024, student/ica na studiju: Diplomski sveučilišni studij Računarstvo, dajem suglasnost Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek da pohrani i javno objavi moj **diplomski rad**:

2D RPG igra s proceduralno generiranim razinama u Unityu

u javno dostupnom fakultetskom, sveučilišnom i nacionalnom repozitoriju.

Osijek, 17.07.2018.

_____
potpis

SADRŽAJ

1. UVOD....1

1.1. Zadatak diplomskog rada....1

2. PRIMIJENJENE TEHNOLOGIJE....2

2.1. Unity 3D 2

2.2. C# programski jezik…….....6

2.3. Paint.NET....6

2.4. Tiled Map Editor....6

3. IZRADA 2D RPG IGRE....7

3.1. Izrada skriptiranih razina 7

3.2. Glavni lik 9

3.3. Neprijatelji....13

3.4. NPC-ovi....14

3.5. Proceduralno generirane razine....16

3.6. Predmeti koji se prikupljaju....19

3.7. Sustav za korištenje predmeta, spremnici podataka i sustav za brzo putovanje....21

3.8. Kupovanje predmeta....24

3.9. HUD....27

4. ZAKLJUČAK....33

5. LITERATURA....34

6. SAŽETAK....35

7. ŽIVOTOPIS....37

1. UVOD

Kao temu diplomskog rada odabrao sam „2D RPG igru sa proceduralno generiranim razinama u Unityu“. Nakon odabira teme bilo je potrebno odrediti koje će sve elemente sadržavati igra, budući da postoje razne vrste RPG-ova. U dogovoru sa mentorom odlučeno je da će se raditi akcijski 2D RPG sa random(proceduralno) generiranim razinama. Igra ima kameru ptičje perspektive. Za engine je odabran Unity pošto smo isti koristili na kolegiju „Razvoj računalnih igara“. Uz korištenje Unity engine-a korišten je programski jezik C# za pisanje skripti, te programi Tiled Map Editor za dizajn razina igre te Paint.NET za kreiranje sprite-ova.

1.1. Zadatak diplomskog rada

Zadatak diplomskog rada je izrada 2D RPG igre sa proceduralno generiranim razinama koristeći Unity engine. Igra treba omogućavati osnovne kretnje likova, borbu sa neprijateljima, skupljanje predmeta i novčane valute s kojom se kupuju jača oružja te napitci za povećanje trenutnog zdravlja glavnog lika. Igra također sadrži nekoliko lokacija koje se sastoje od više razina. Neke od ovih razina će biti slučajno generirane što osigurava da se ista razina neće ponoviti. Osim toga, igra sadržava i funkcionalnu mini kartu na kojoj su bojama prikazani NPC-ovi, neprijatelji, mjesta za prijelaz u drugu lokaciju itd. Za kraj, igra sadrži glavnog neprijatelja nakon čijeg poraza je igra gotova. Uz sve to potreban je i funkcionalni izbornik koji se pojavljuje prilikom pokretanja aplikacije. Ovaj izbornik nudi korisniku informacije kao što su kontrole te ostale informacije koje mogu pomoći igraču.

2. PRIMIJENJENE TEHNOLOGIJE

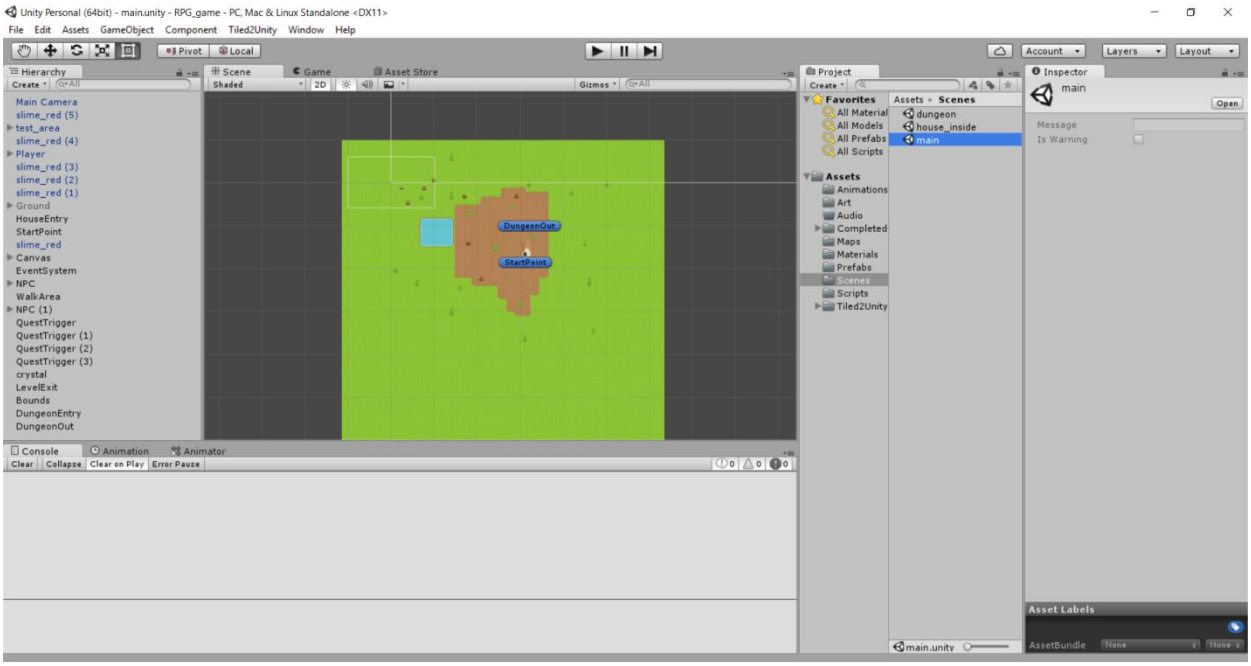
2.1. Unity 3D

Unity je game engine (pogonski sklop igre) kreiran od strane tvrtke Unity Technologies. Služi za kreiranje videoigara za više platformi (od PC-a, igraćih konzola do mobilnih uređaja i web preglednika). Unity je trenutno jedan od najpopularnijih engine-a u svijetu te se konstantno poboljšava izlascima novih verzija programa. Prvotno je predstavljen 2005. Godine, kada je bio podržan jedino na OS X operacijskom sustavu. Od tada do danas proširio se na 27 platformi. Unity se koristi za izradu i 2D i 3D videoigara. Cilj kreatora Unity-a je da svi mogu kreirati videoigre, bilo da ste profesionalac u industriji ili student.

Unity dolazi u četiri verzije:

1. Osobna (engl. personal) verzija
2. Plus verzija
3. Pro verzija
4. Enterprise verzija

Osobna inačica je besplatna te se može koristiti ako generirana zarada korisnika nije veća od 100,000 dolara godišnje. Plus verzija se plaća 35 dolara mjesečno, ali korisnik smije zaraditi do 200,000 dolara. Pro i Enterprise verzije se koriste ako je zarada veća od ranije navedenog iznosa te se Pro verzija plaća 125 dolara mjesečno, dok se Enterprise plaća po dogovoru odnosno o njoj se more pregovarati. Na slici 2.1 vidljivo je sučelje Unity-a.



Sl. 2.1 Unity sučelje

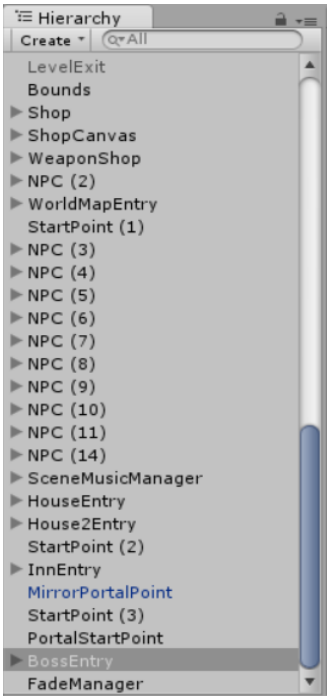
Sučelje Unity-a se sastoji od nekoliko bitnih dijelova koji služe za izradu videoigre te se svaki od elemenata može zamijeniti nekim drugim elementom ovisno o preferencijama korisnika. Navedeno znači da razmještaj elemenata nije isti kod svakog korisnika. Najbitnija svojstva za rad se nalaze u alatnoj traci koja je vidljiva na slici 2.2.

Sl. 2.2 Alatna traka

Alatna traka na lijevoj strani sadrži osnovne alate za manipuliranje scene i objektima unutar scene. U sredini se nalaze tipke za pokretanje i pauziranje igre. Tipke na desnoj strani alatne trake pružaju pristup korisničkom Unity Cloud servisu i Unity računu, te odabir vidljivih slojeva u igri. Ovdje je također i uređivač izgleda sučelja koji sadrži nekoliko predefiniranih rasporeda elemenata. Pogled scene služi za navigaciju i uređivanje scene. Njime možemo prikazati scenu u 2D i 3D perspektivi. Primjer je vidljiv na slici 2.3.

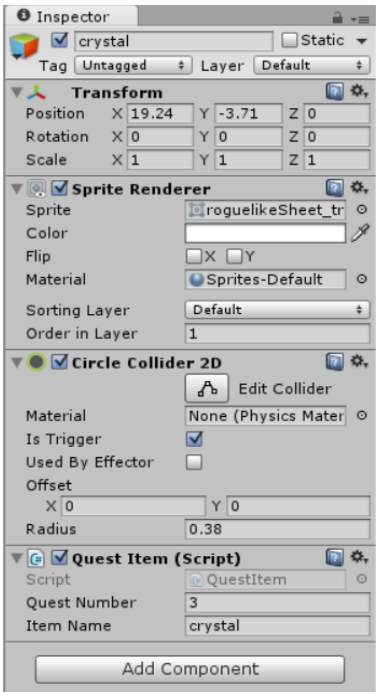
Sl. 2.3 Pogled scene

Hijerarhija objekata sadrži popis svih objekata/elemenata unutar scene. Svaki objekt koji se doda unutar scene automatski je uvršten u hijerarhiju. Izgled hijerarhije vidljiv je na slici 2.4. Također prikazuje i objekte koji su pridodani nekom drugom objektu.



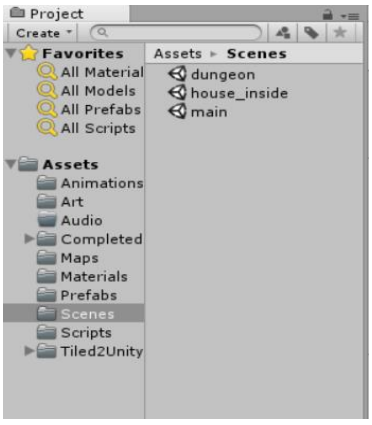
Sl. 2.4 Hijerarhija objekata

Sljedeći bitan element Unity sučelja je inspektor. Inspektor služi za pregled i uređivanje svih svojstava odabranog objekta. Neke osnovne stvari su slika objekta, pozicija unutar scene, collider, skripta objekta itd. Primjer inspektora jednog objekta je vidljiv na slici 2.5.



Sl. 2.5 Inspektor objekta

Slika 2.6 prikazuje prozor projekta koji sadrži sve resurse koje imamo na raspolaganju pri izradi projekta. Prilikom uvoza resursa u projekt isti se ovdje prikazuju. Resursi su najčešće slike, skripte, materijali, animacije i zvukovi.



Sl 2.6 Prozor projekta

2.2. C# programski jezik

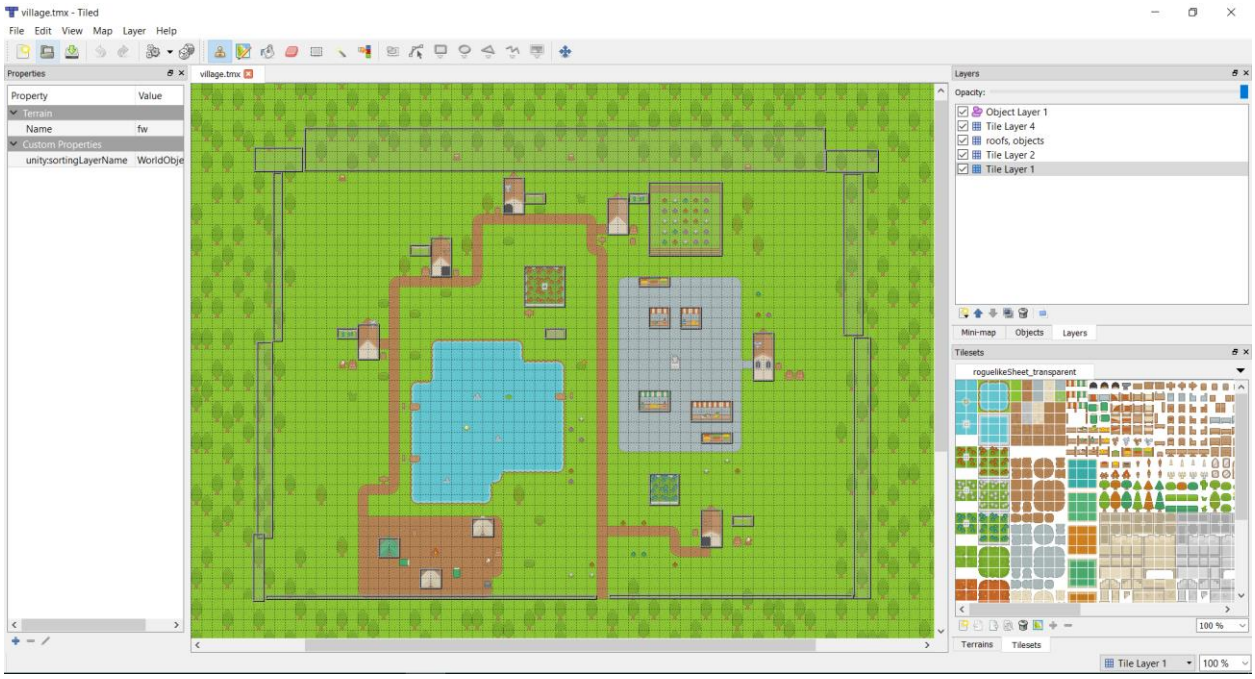
Unity podržava dva programska jezika, a to su C# i UnityScript, kreiran specifično za radu u Unity-u koji je baziran na JavaScript jeziku. Kao što je ranije navedeno, za izradu ovog diplomskog rada koristi se C#. C# je objektno-orijentirani programski jezik razvijen od strane Microsoft-a te se prvi puta pojavio 2000. godine. U Unity engine-u se koristi za pisanje skripti kako bi objekti unutar igre znali reagirati na npr. pritisak tipke od strane korisnika, za određivanje događaja koji se trebaju dogoditi ako se zadovolje uvjeti. Ovaj programski jezik također omogućuje kreiranje grafičkih efekta, upravljanje fizičkim ponašanjem objekata te za kreiranje umjetne inteligencije likova unutar igre. Drugim riječima, svaka dinamička komponenta igre zahtjeva pisanje skripte. Mogućnosti su, moglo bi se reći, beskonačne.

2.3. Paint.NET

Paint.NET je besplatni program za uređivanje slika razvijen za Microsoft Windows. Za izradu ovog rada potreban je program u kojemu je moguće nacrtati i uređivati različite slike, stoga je u izradi ove igre korišten Paint.NET. Osim toga, navedeni program je prilično jednostavan, a također i besplatan.

2.4. Tiled Map Editor

Tiled je open source program za kreiranje 2D razina koje se koriste unutar igre pomoću blokova/ploča (engl. tile). Na slici 2.7 vidljiv je primjer izgleda razine kreirane pomoću Tiled Map Editora. Koriste se već nacrtani kvadratni blokovi/ploče koji se dodaju u program, te se pomoću istih kreira razina. Tiled podržava i slojevitost što znači da se nacrtani objekti mogu postaviti na razne slojeve tako da mogu prekriti jedan drugoga (npr. kuća prekriva travu). Jedna od mogućnosti Tiled Map Editora, koja je uvelike pomogla pri izradi ovoga rada, je podrška collider-a. Primjer korisnosti su objekti unutar razina kod kojih se, pri kreiranju razine, može postaviti da likovi u igri ne mogu proći kroz njih (kao što je kuća, drvo, itd.) bez da se unutar Unity-a mora zasebno svakom objektu razine dodavati komponenta Collider.



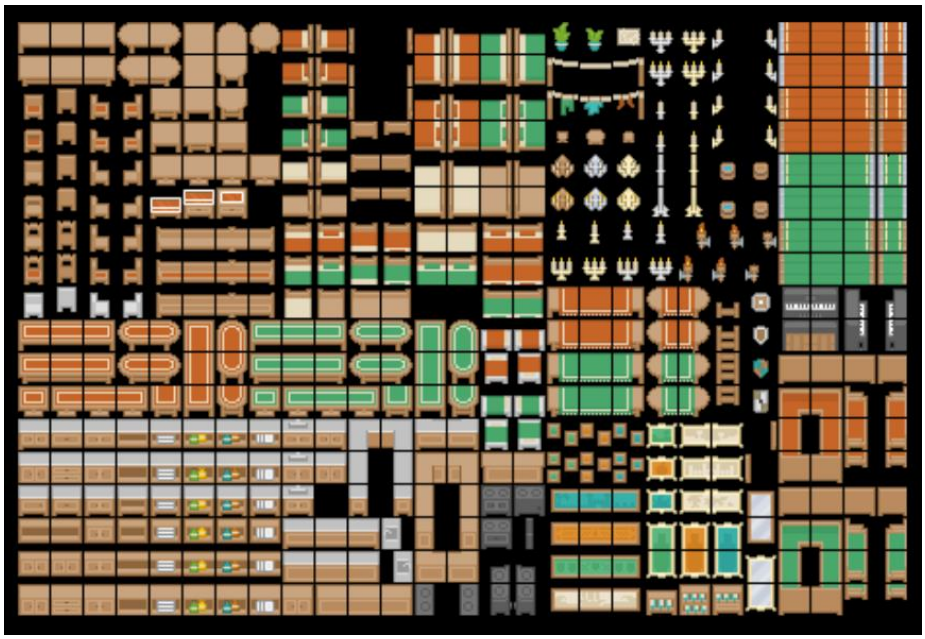
Sl. 2.7 Primjer razine kreirane u Tiled Map Editor-u

3. IZRADA 2D RPG IGRE

U ovom dijelu diplomskog rada prikazani su i objašnjeni glavni dijelovi izrade igre, kao što su izrada glavnog lika te animacija raznih akcija i pokreta, kreiranje mapa pomoću Tiled Map Editor-a, generiranje proceduralnih razina, izrada sustava za prikupljene predmete koje igrač ima na raspolaganju te samih predmeta koji se skupljaju i kupovanje predmeta te povezivanje svega sa korisničkim sučeljem. Važno je napomenuti da je velika većina sprite-ova koja se koristi napravljena od strane tvrtke Kenney.nl koji kreiraju besplatni 2D sadržaj za videoigre.

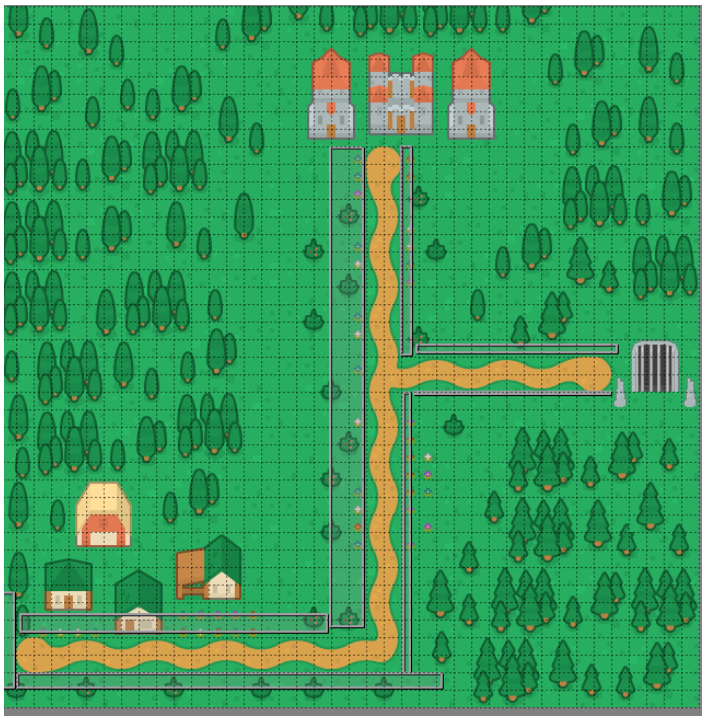
3.1. Izrada skriptiranih razina

Kao što je navedeno, za izradu određenih područja je odabran klasični pristup, odnosno kreiranje pomoću Tiled Editor-a. Za izradu razina potrebno je u program dodati kolekciju sprite-ova koja sadrži sve potrebno za kreiranje istih. Primjer jedne od korištenih kolekcija sprite-ova je vidljiva na slici 3.1.



Sl. 3.1 Sprite kolekcija

Sama izrada razina zahtjeva podosta vremena jer je potrebno sve pomno isplanirati kako bi razina imala smisla. Prva kreirana razina služi kao mapa svijeta na kojoj glavni lik ima ograničeno kretanje te omogućuje pristup ostalim razinama kao što je vidljivo na slici 3.2.



Sl. 3.2 Mapa svijeta

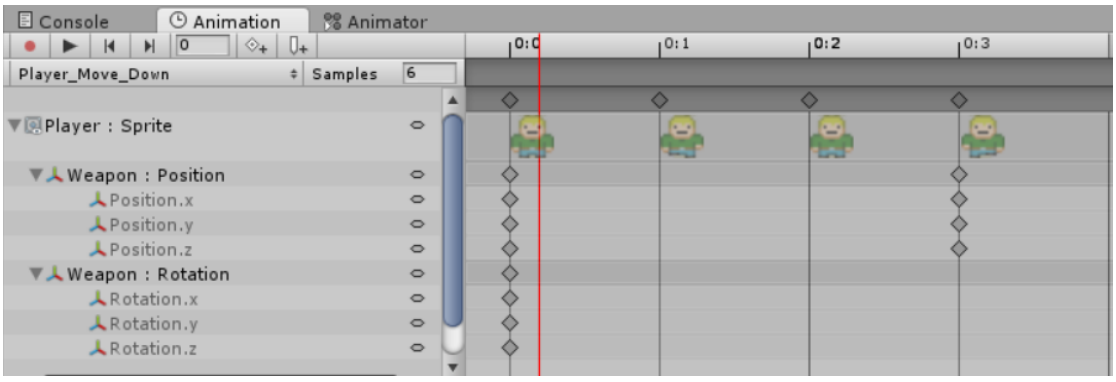
Kretanje je ograničeno pomoću mogućnosti unutar Tiled Editora koja dozvoljava dodavanje collidera unutar mape koje Unity automatski prepoznaje. Ova mogućnost je vrlo korisna jer nije potrebno svakom objektu zasebno dodavati collider.

3.2. Glavni lik

Idući korak je izrada mehanike glavnog lika koja uključuje njegovo kretanje, interakciju sa objektima unutar igre te sa ostalim likovima(NPC i neprijatelji). Za potrebe kretanja i borbe potrebno je dodati skriptu u kojoj se definira brzina kretanja te brzina napada. Slična stvar je napravljena kod neprijatelja unutar igre što će biti uskoro prikazano.

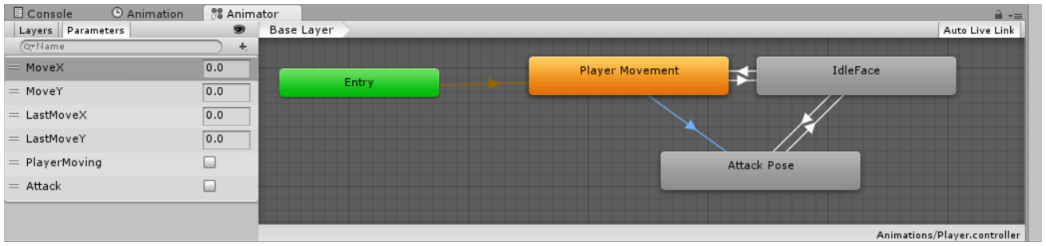
Sl. 3.3 Glavne varijable skripte glavnog lika

Kao što je navedeno u uvodu ovoga rada, igra se odvija u ptičjoj perspektivi što znači da je potrebno napraviti animaciju glavnog lika za svaki od četiri smjera. Animiranje je napravljeno pomoću Animator komponente unutar Unity-a dodavanjem slika pokreta glavnog lika kao što je vidljivo na slici ispod.

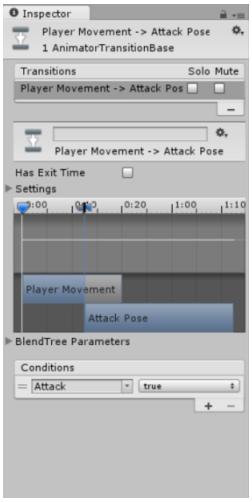


Sl. 3.4 Animacija glavnog lika pri kretanju po y-osi prema dolje

Glavni lik osim hodanja posjeduje i stanje u kojemu miruje te stanje u kojemu se bori, tj. napada neprijatelje. Prelazak iz jednog stanja u drugo odrađeno je pomoću animator sučelja gdje pomoću parametara koji se mijenjaju radimo prijelaz u drugo stanje. Izmjena vrijednosti parametara se izvodi unutar skripti.

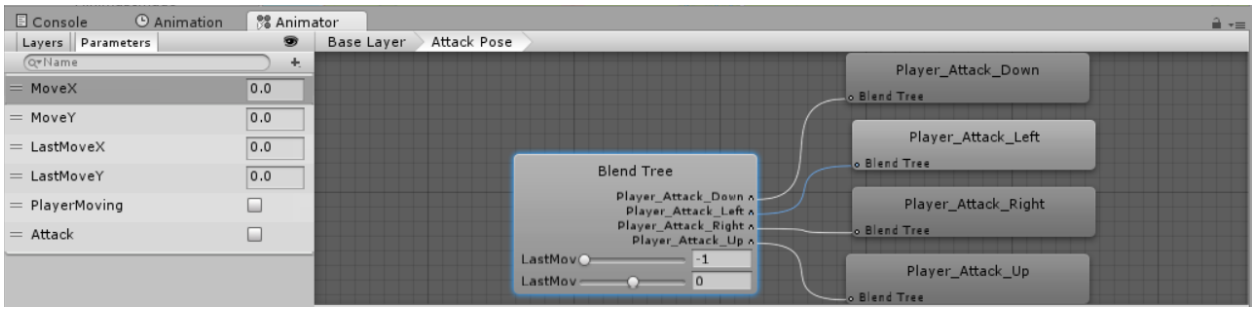


Sl. 3.5 Animator sučelje u kojem se nalaze parametri za glavnog lika



Sl. 3.6 Prikaz tranzicije u drugo stanje u inspektoru

Kao što je vidljivo na slici 3.6 za prijelaz iz stanja kretanja u stanje napada potrebno je promijeniti vrijednost parametra „Attack“ u „True“. Kao što je navedeno, svako od stanja sadrži četiri animacije. Izmjena animacija unutar jednog stanja se izvodi pomoću „Blend Tree“ opcije. Prilikom rada se, kao i kod izmjena između samih stanja, promjenom vrijednosti parametara u animator komponenti mijenja animacija koja se izvodi.

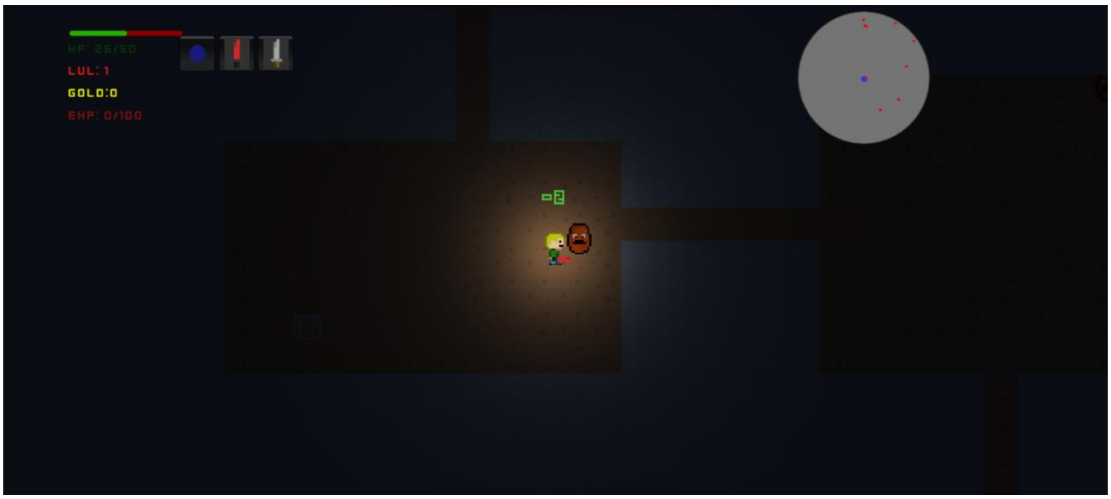


Sl. 3.7 Blend Tree stanja u kojemu glavni lik napada

Iduća bitna stvar je kreirati način borbe sa neprijateljima unutar igre. Prvo je potrebno napraviti skriptu kojom određujemo životne bodove (engl. HP) glavnoga lika nakon čega se definira način nanošenja štete (slika 3.8).

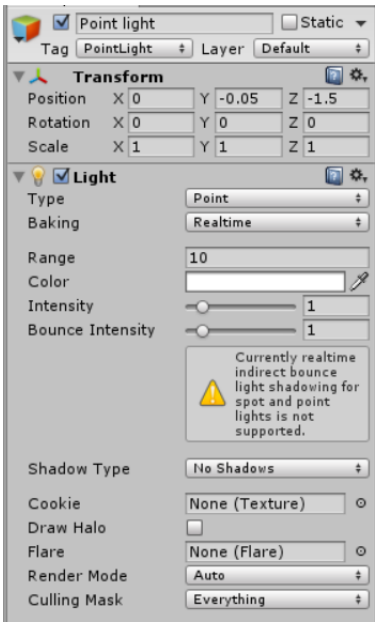
Sl. 3.8 Skripta za određivanje životnih bodova

Kao što je vidljivo na slici 3.8, skripta sadrži maksimalan broj i trenutni broj životnih bodova. Prilikom primanja štete također je dodan efekt bljeska glavnoga lika. Prilikom sudara sa neprijateljem glavnom liku se umanjuju životni bodovi za iznos koji je definiran u skripti. Skripta se nalazi na objektima koji predstavljaju neprijatelje te je prikazana na slici 3.10. Iznos primljene štete je prikazan zelenim brojem koji lebdi iznad glavnoga lika kao što je prikazano na slici 3.9.



Sl. 3.9 Prikaz nanošenja štete glavnom liku

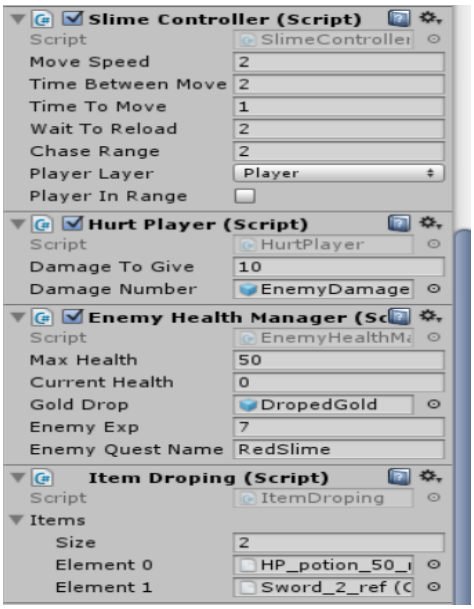
Kao što je vidljiv na slici iznad ako se glavni lik nalazi u tamnim područjima, u ovom slučaju tamnici, oko njega se pojavljuje svjetlo kako bi igrač bolje vidio što okružuje glavnog lika. Svjetlo je zapravo jedna od komponenti Unity-a koja je dodana objektu. Postoje razne vrste svjetla, a u ovom slučaju se koristi „Point light“. Njena specifičnost je što emitira svjetlost u svim smjerovima te se intenzitet svjetlosti smanjuje s obzirom na udaljenost od izvora(glavnog lika). Objekt sa komponentom svjetla je prikazan na slici 3.10.



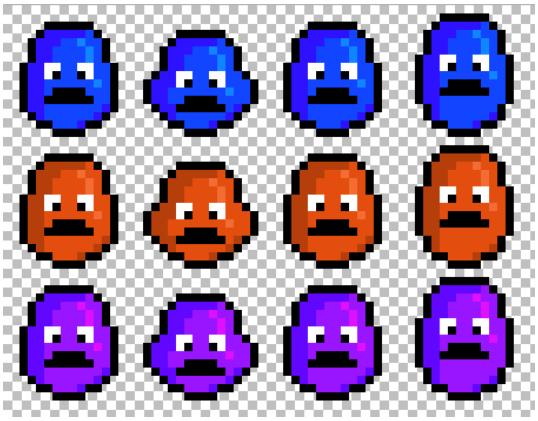
Sl. 3.10 Objekt sa komponentom svjetla

3.3. Neprijatelji

Igra sadrži nekoliko vrsta neprijatelja koji različito reagiraju na glavnog lika te svaki posjeduje drugačija svojstva. Neka od tih svojstava su brzina kretanja, health, zatim koliko štete rade glavnom liku ako dođu s njim u kontakt i koliko njihovo uništenje nosi iskustvene bodove. Svakom neprijatelju su dodane skripte kojima se definiraju navedena svojstva koja su deklarirana kao „public“, kako bi se vrijednosti mogle mijenjati unutar editora po potrebi kao na slici 3.11. Tipovi neprijatelja su prikazani na slici 3.12.



Sl. 3.11 Osnovne skripte neprijatelja

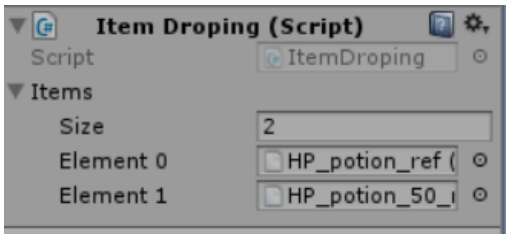


Sl. 3.12 Neprijatelji

Prilikom napada neprijatelja prikazani su broj štete koji je glavni lika napravio neprijatelju te koliko iskustvenih bodova tj. bodova potrebnih za unaprijeđenje smo dobili. Sam način unaprjeđivanja glavnoga lika je objašnjen u daljnjem tekstu. Uništavanjem neprijatelja se stvara novac koji se, ako se pokupi, pridodaje već skupljenoj količini te služi za kupovanje boljeg oružja ili jednog od napitaka za povećanje trenutnog zdravlja.

Sl.3.13 Prikaz uništenja neprijatelja

Svaki neprijatelj sadrži skriptu u kojoj je definirana lista napitaka koje on posjeduje. Nakon uništenja neprijatelja slučajnim odabirom stvara se jedan od tih napitaka koje glavni lik može pokupiti i koristiti. Vjerojatnost da će neprijatelj nakon smrti ostaviti neki napitak je mala te iznosi do 10%.



Sl. 3.14 Skripta kojom se omogućuje dobivanje napitka

3.4. NPC-ovi

Sljedeće je potrebno kreirati NPC-ove (engl. non-player character) koji „žive“ unutar igre te služe kako bi igrači saznali nešto više o svijetu u kojemu se igra odvija ili kako bi od istih kupovali oružja ili napitke (slika 3.15).

Sl.3.15 Interakcija glavnog lika i NPC-a

Slika 3.16 prikazuje jednu od trgovina u kojoj je moguće kupiti napitke za prikupljeni novac koji dobivamo uništavanjem neprijatelja, kao što je već ranije objašnjeno. Sama mehanika kupovanja predmeta će biti objašnjena kasnije.

Sl. 3.16 NPC koji omogućuje kupnju različitih napitaka

Kretanje NPC-ova u igri je potpuno slučajno. Kod svakog NPC-a je određena brzina kretanja i vrijeme trajanja kretanja te vrijeme čekanja. Prilikom isteka vremena koje NPC čeka, odnosno stoji na mjestu, izvršava se funkcija u kojoj se dobiva slučajni broj u rasponu od jedan do četiri te svaki broj predstavlja jedan od četiri moguća smjera za kretanje. Slika 3.17 prikazuje skriptu jednog od NPC-ova u kojoj možemo mijenjati brzinu kretanja, vrijeme trajanja istog te vrijeme stajanja.

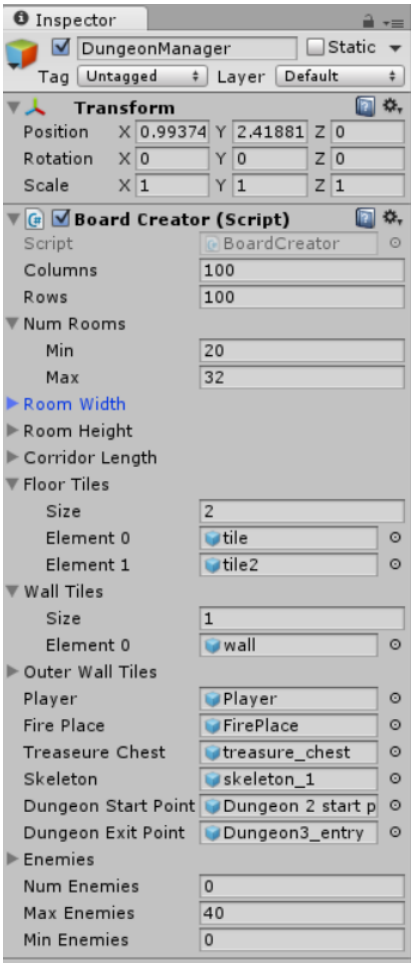
Sl.3.17 Skripta za kretanje NPC-ova

3.5. Proceduralno generirane razine

Proceduralno generirane razine predstavljaju tip razina koji se „slučajno“ generira unutar određenih parametara svaki puta kada se ista učita. To znači da će se prilikom svakog pristupa razini ista drugačije izgledati, odnosno biti će drugačiji raspored prostorija, njihov broj i veličina. Broj i vrsta neprijatelja su također svaki puta drugačiji, što pruža korisniku određenu vrstu svježine jer svaki puta ima osjećaj kao da igra drugu razinu.

Postoje razni načini kako proceduralno generirati razine i područja unutar igre, te je to u isto vrijeme umjetnost, ali i znanost. U ovoj igri se koristi jednostavan način za generiranje tamnica unutar igre unutar kojih se nalaze neprijatelji. U razinama koje se želi proceduralno generirati dodaje se objekt sa skriptom koja upravlja generiranjem svih elemenata koje želimo unutar scene. Objekt sa pripadajućom skriptom je vidljiv na slici 3.18. Prva stvar koja je definirana je grid. Pomoću njega se određuje prostor u kojemu se radi generiranje objekata. U ovom slučaju njegova je veličina 100x100 tile-ova. U inspektoru se također mogu mijenjati razne vrijednosti kao što je broj prostorija u trenutnoj razini. Postavljaju se minimalan i maksimalan broj prostorija te se slučajnim odabirom prilikom pokretanja razine generira broj prostorija između minimalnog i maksimalnog broja. Ista stvar se primjenjuje i za širinu i visinu prostorija te dužinu hodnika koji povezuju prostorije. Razlog za korištenje ovog pristupa je što veća raznolikost generiranih prostorija i hodnika. Također je omogućeno generiranje jedne prostorije preko druge što rezultira većom raznolikošću oblika prostorija koje bi u suprotnom bile sve pravokutnog oblika, dok se ovakvim pristupom mogu dobiti zanimljivi oblici prostorija. Skripta također sadrži dva polja objekata koji predstavljaju „Floor“ i koji predstavljaju „Wall“. Osim njih, sadrži i razne druge objekte kao što su neprijatelji, škrinje sa blagom itd., koji će se generirati unutar prostorija i hodnika. Na početku je kreiran „Enum“ sa dvije moguće vrijednosti(„Floor“ i „Wall“). Prilikom izvođenja skripte prvotno se kreiraju jedna prostorija i jedan hodnik koji vodi iz iste. Zatim se kreiraju sve ostale prostorije i hodnici. Prostorije se

uvijek generiraju od donjeg lijevog tile-a. Najvažnije je odrediti x i y koordinate tog tile-a. Hodnici služe za povezivanje prostorija, a smjer u kojemu se izlazi iz prostorije (istok, jug, sjever ili zapad) se također slučajno odabire. Prilikom definiranja početne i krajnje koordinate hodnika provjerava se da li bi hodnik izašao van grid-a. U tom slučaju njegova duljina se smanjuje. Svaka prostorija, osim prve, ima hodnik koji ulazi u nju te se prilikom generiranja hodnika koji izlazi iz te prostorije provjerava smjer prošlog hodnika te se zabranjuje da hodnik ide u suprotnom smjeru. Razlog tomu je želja za što većim grananjem i pokrivanjem što većeg područja unutar definiranog grid-a.

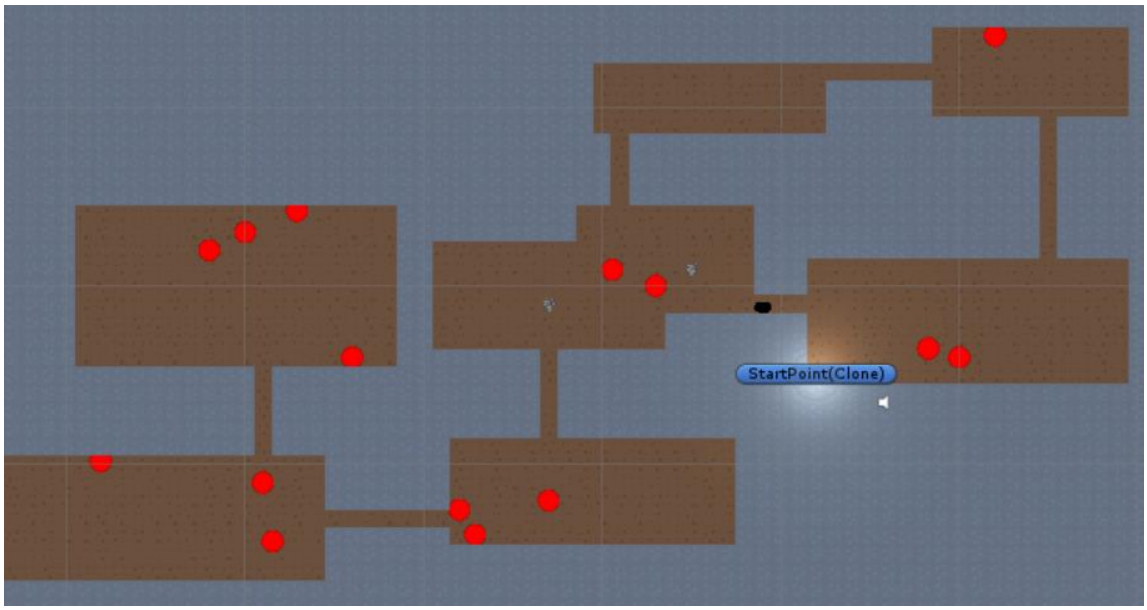


Sl. 3.18 BoardCreator skripta koja upravlja generiranjem razina

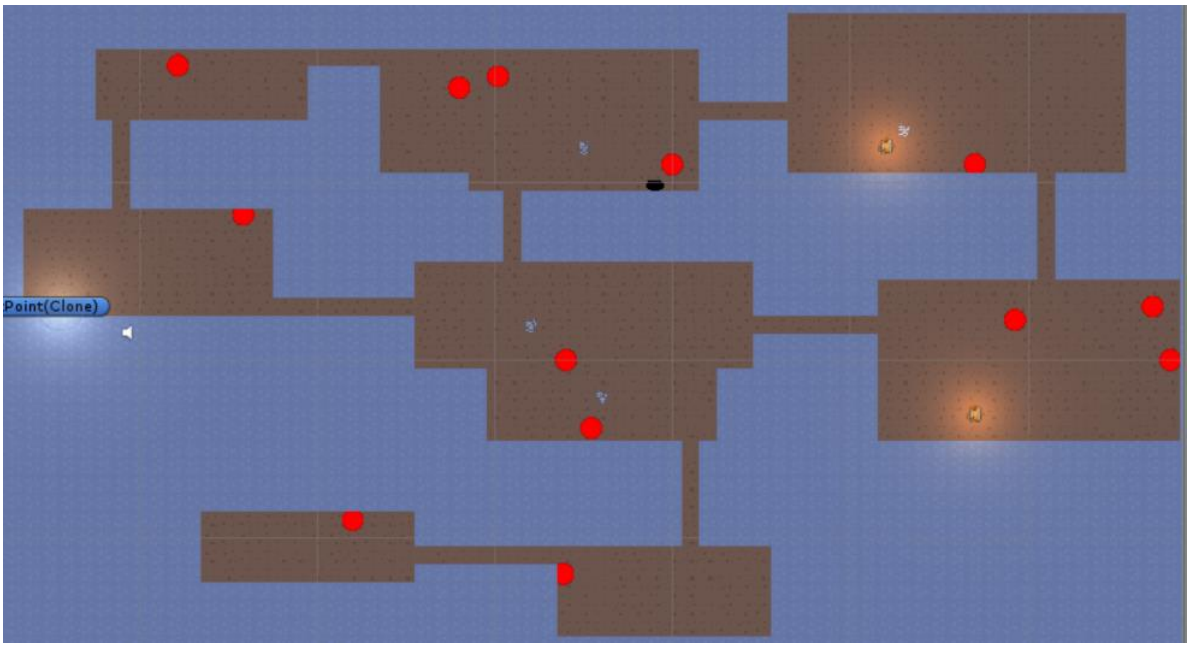
Nakon postavljanja početnih koordinata prostorije i hodnika, krećemo od tih koordinata i pomičemo se za jedan po širini i visini (x i y os) za prostoriju odnosno po duljini za hodnike. Prilikom toga svaku koordinatu postavljamo da je tipa „Floor“, a sve ostale da su tipa „Wall“. Zatim se na svaku koordinatu koja ima vrijednost „Floor“ stvara objekt koji predstavlja tlo te za

koordinatu koja ima vrijednost „Wall“ stvara objekt koji predstavlja zid. Tlo i zidovi se stvaraju slučajnim odabirom iz polja objekata radi što veće raznolikosti razina.

Sl. 3.19 Proceduralno generirana razina



Sl. 3.20 Proceduralno generirana razina

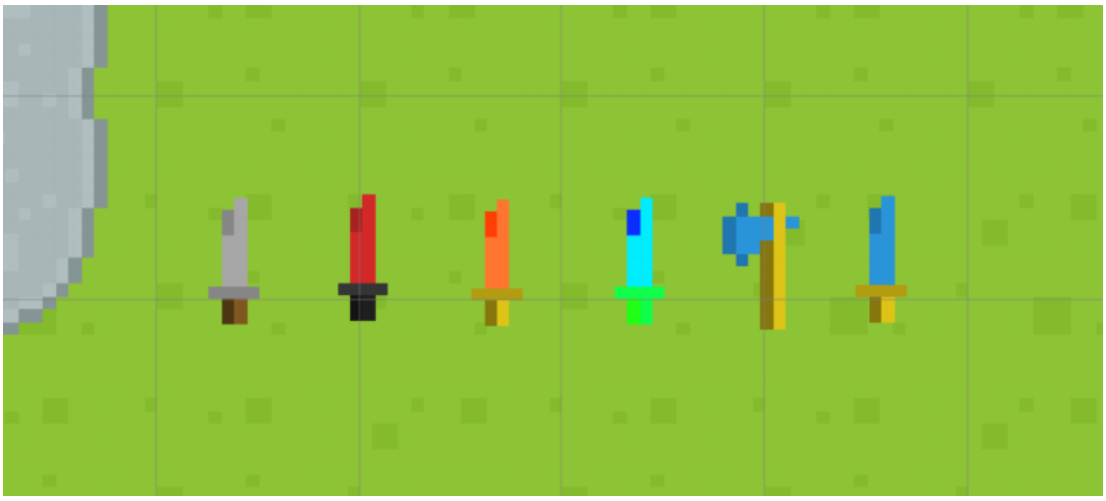


Sl. 3.21 Proceduralno generirana razina

Slike 3.19, 3.20 i 3.21 prikazuju proceduralno generirane razine koje sadrže različiti broj prostorija. Prostorije unutar razina su različitih dimenzija sa različitim brojem neprijatelja. Prilikom kreiranja prostorije slučajnim odabirom se određuje broj neprijatelja, te hoće li ista sadržavati škrinju sa blagom, put za odlazak u iduću razinu ili neki detalj. Neprijatelji se, isto kao i objekti koji predstavljaju tlo i zidove, nalaze unutar polja objekata. Unutar jedne prostorije može se pojaviti više vrsta neprijatelja.

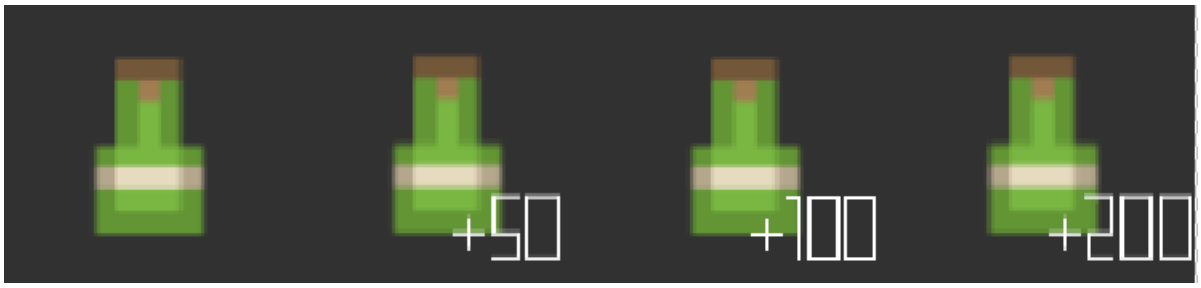
3.6. Predmeti koji se prikupljaju

Kao što je spomenuto ranije, postoje razni predmeti koje glavni igrač može pokupiti i spremiti u popis predmeta kojima raspolaže ili kupiti te koristiti tijekom igre. Postoje tri glavne vrste predmeta: oružje, napitci te novac. Svaki od objekata sadrži collider komponentu koja omogućuje provjeru je li glavni lik došao u kontakt sa predmetom kako bi ga prikupili. Sam način prikupljanja i način na koji se predmeti koriste biti će objašnjen u idućem potpoglavlju. Postoji nekoliko različitih oružja u igri koje igrač može skupiti/kupiti, a prikazana su na slici 3.22. Razlika među oružjem, osim u izgledu, je u količini štete koju nanosi pojedino oružje te u cijeni prilikom kupnje.



Sl. 3.22 Oružja u igri

Kao i oružje napitci se također mogu skupiti prilikom uništenja neprijatelja ili kupiti od NPC-a. Postoji nekoliko različitih napitaka koji se razlikuju po količini zdravlja glavnoga lika koji mogu regenerirati (slika 3.23).



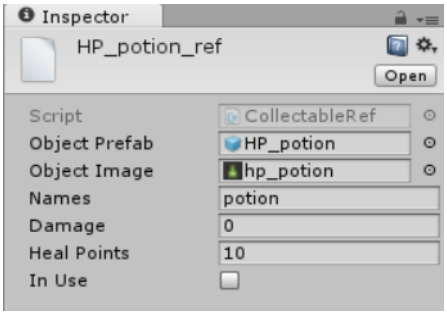
Sl. 3.23 Napitci u igri

Zadnji predmet koji se može skupiti je novac koji, kao što je rečeno u prijašnjem potpoglavlju, se stvara prilikom uništenja neprijatelja. Objekt koji predstavlja novac sadrži skriptu u kojoj je definirano da se prilikom stvaranja objekta slučajnim odabirom(između maksimalne i minimalne vrijednosti) određuje iznos novca koji će glavni lik dobiti prilikom skupljanja istog (Slika 3.24).

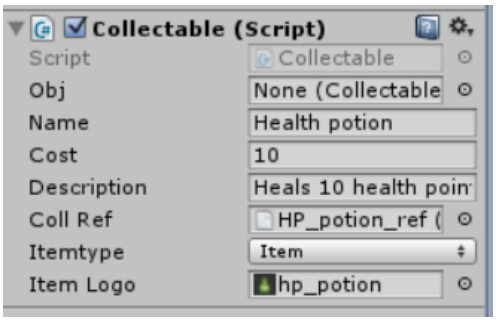
Sl. 3.24 Novac stvoren nakon uništenja neprijatelja

3.7. Sustav za korištenja predmeta, spremnici podataka i sustav za brzo putovanje

Jedna od glavnih karakteristika RPG igara je sustav korištenja prikupljenih predmeta, koji omogućuje prikupljanje predmeta i njihovo korištenje ovisno o potrebi. Prilikom izrade ovog sustava u igri ključni su bili objekti koji sadrže informacije (engl. scriptable objekti), odnosno spremnici podataka. Ovi objekti su vrsta objekata koji mogu sadržavati različite podatke. Razlog zbog kojega se koriste u ovom projektu je taj što je prilikom dodavanja nekog predmeta u popis prikupljenih predmeta iz scene potrebno obrisati odnosno uništiti predmet u sceni. Ako se ne koriste ovakvi objekti prilikom brisanja/uništavanja objekta iz scene se briše i kopija objekta koja je dodana u popis prikupljenih predmeta glavnoga lika. Na slici 3.25 je vidljiv primjer jednoga od spremnika podataka. Kao što je vidljivo, isti sadrži razne podatke gdje je najvažniji podatak objekt koji predstavlja sa svim njegovim komponentama. Svaka vrsta objekta spomenuta ranije sadrži skriptu pod nazivom „Collectable“ (Slika 3.26), u kojoj je definiran spremnik podataka koji predstavlja taj objekt. Svaki objekt je povezan sa odgovarajućim spremnikom podataka i obratno, što omogućuje da se prilikom dodavanja objekta u popis dodaje objekt koji sadrži podatke umjesto da se dodaje objekt iz scene tako da se sada isti može obrisati.

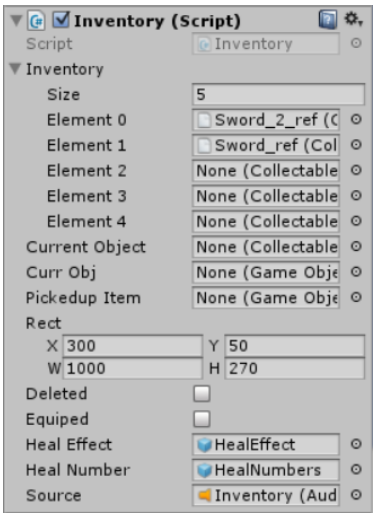


Sl. 3.25 Primjer spremnika podataka



Sl. 3.26 Collectable skripta

Nakon kreiranja objekta i pripadajućeg spremnika podataka potrebno je napraviti objekt koji će predstavljati popis prikupljenih predmeta. Najbitnija komponenta je skripta u kojoj je definirana lista objekata koji predstavljaju spremnike podataka, te se prilikom korištenja nekog od predmeta iz popisa prikupljenih predmeta instancira (stvara) objekt čije podatke sadrži odgovarajući spremnik podataka.



Sl. 3.27 Skripta za upravljanje popisom prikupljenih predmeta

Sl. 3.28 Izgled popisa u igri

Slika 3.28 u crvenom pravokutniku prikazuje sve objekte koji se trenutno nalaze u listi spremnika podataka te koje glavni lik može koristiti pritiskom nekoga od njih. Unutar plavog kvadrata se nalazi tipka čije korištenje omogućava stvaranje portala za povratak glavnoga lika, odnosno brzo putovanje (učitavanje scene) u grad u kojemu počinje igra sa svim predmetima, novcem, iskustvom koje je dosad prikupio. Sustav za brzo putovanje služi korisniku da se opskrbi boljim oružjem ili za kupovinu dodatnih napitaka za zdravlje. Slika 3.29 prikazuje izgled portala. Međutim, ako se sustav za brzo putovanje koristi za izlazak iz tamnice, prilikom povratka tamnica se ponovno proceduralno generira što znači da se i neprijatelji ponovno generiraju te raspored prostorija nije isti kao prilikom povratka u početni grad.

Sl. 3.29 Portal za povratak u glavni grad

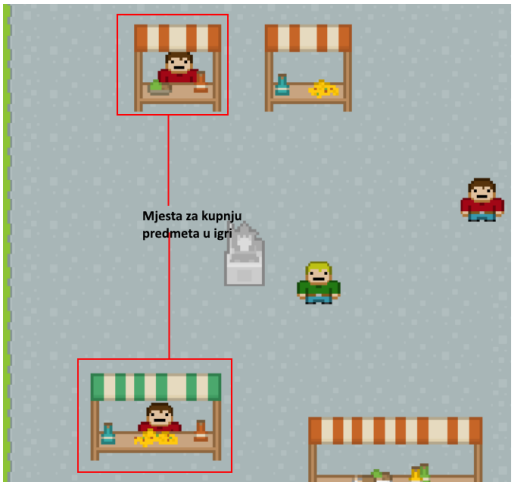
Kako korisnik ne bi morao ponovno prelaziti igru do onoga dijela gdje je iskoristio sustav za brzo putovanje nazad na početak, prilikom korištenja portala automatski se stvara dodatni portal u centru glavnoga grada prikazan na slici 3.30. Isti se pojavljuje uvijek na istomu mjestu.

Uporabom tog portala korisnik se vraća na mjesto odakle je prvotno iskoristio brzo putovanje. Prilikom brzog putovanja u početni grad ime trenutne scene se sprema u varijablu, te se prilikom korištenja portala u centru grada vraćamo u scenu čije se ime nalazi u spomenutoj varijabli.

Sl. 3.30 Dodatni portal

3.8. Kupovanje predmeta

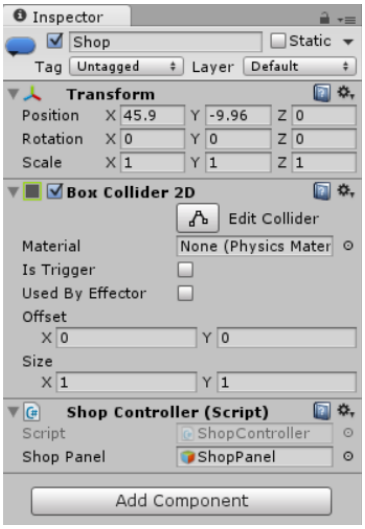
Već je ranije spomenuta mogućnost kupovine predmeta u igri. Postoje dvije lokacije na kojima je omogućena kupovina, odnosno trgovine (Slika 3.31).



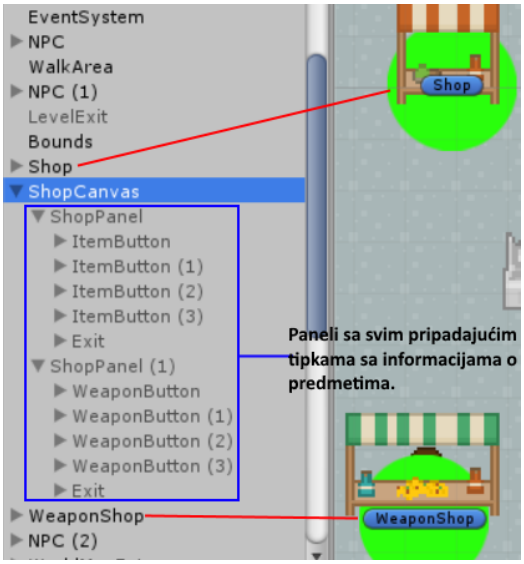
Sl. 3.31 Lokacije za kupnju predmeta u igri

Na mjestima gdje se nalaze trgovine su dodani objekti koji sadrže dvije komponente. Prvo što sadrže je collider koji služi za provjeru je li glavni lik došao u kontakt sa trgovinom te skriptu

pomoću koje se prikazuju svi predmeti koje igrač može kupiti. U skripti je definirano da se prilikom kontakta collider-a trgovine sa collider-om glavnoga lika aktivira panel, odnosno dio korisničkog sučelja koji je inače neaktivan, sa svim informacija o predmetima u trgovini.



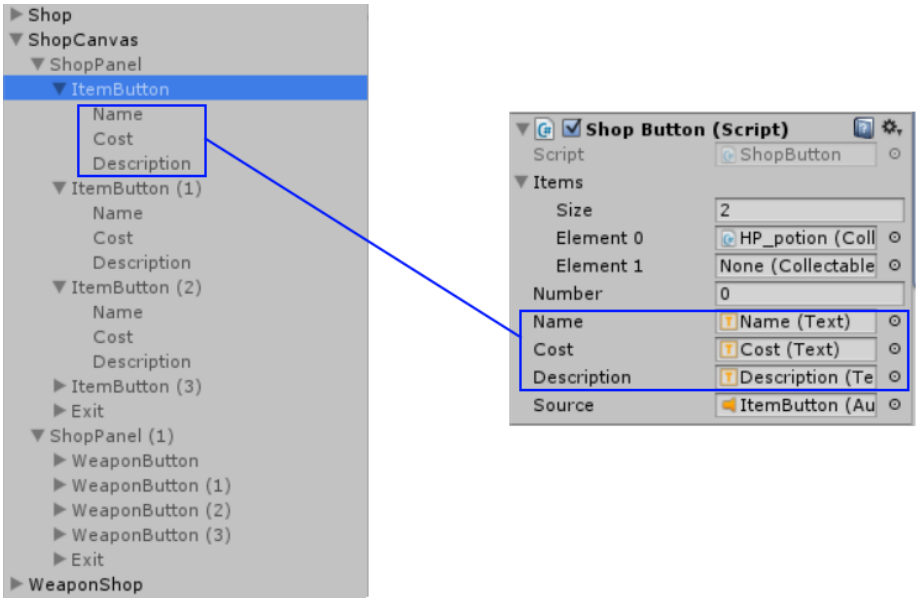
Sl. 3.32 Objekti trgovina sa svojim komponentama



Sl. 3.33 Paneli koji su prvotno neaktivni

Kao što je vidljivo na slici 3.33, postoje dva panela, po jedan za svaku trgovinu. Paneli se sastoje od nekolicine tipki sa informacijama o predmetima koje možemo kupiti. Tipke sadrže po tri tekstualna objekta za naziv predmeta, opis te njegovu cijenu. Uz osnovne komponente koje sadrži svaka tipka kreirana u Unity-u svaka od tih tipki sadrži i skriptu „Shop Button“. U ovoj tipki definirane su tekstualne varijable koje su povezane sa odgovarajućim tekstualnim objektom te se pomoću njih vrijednosti tih varijabli mijenjaju aktiviranjem panela. U skripti je definirana i

lista predmeta u kojoj se nalazi predmet ili više njih koji se prilikom pritiska tipke kupuju. Pošto svaki od predmeta koji se može pokupiti ili kupiti (kao što je vidljivo na slici 3.26) sadrži informacije o sebi kao što su cijena, naziv, opis, ovi se podatci zapisuju u tekstualne objekte svake tipke prilikom aktivacije panela. Prikaz koji predstavlja povezanost tekstualnih objekata tipke i varijabli u skripti je na slici 3.34.

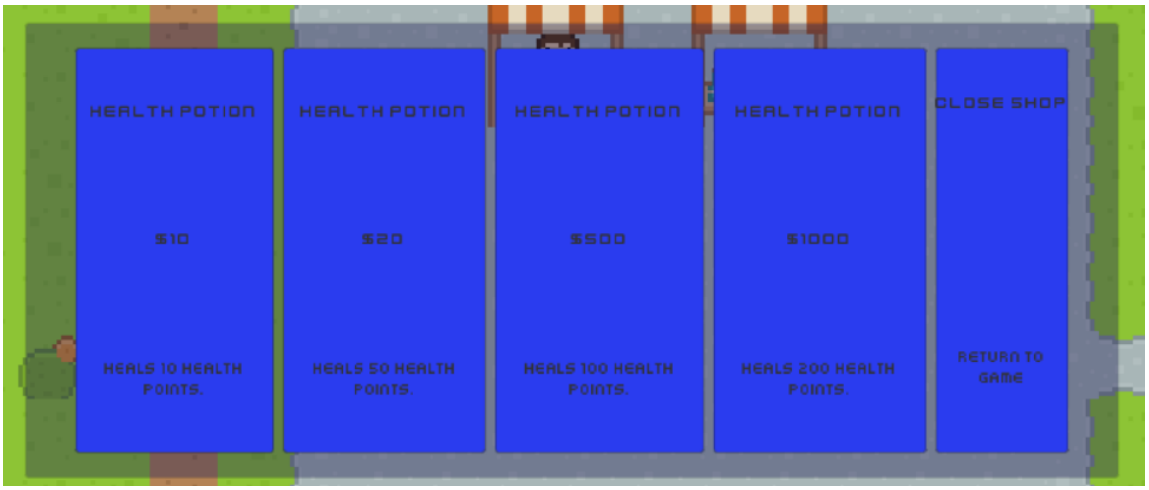


Sl. 3.34 Povezanost tekstualnih objekata te varijabli skripte

Izgleda panela je prikazan na slici 3.35. Prikazana je trgovina u kojoj je omogućena kupnja raznog oružja. Kod svakog oružja, uz naziv, prikazana je i cijena te opis, odnosno koliko dodatne štete glavni lik prilikom napada nanosi neprijateljima.

Sl. 3.35 Izgled panela trgovine oružjem

Pritiskom na jedno od ovih oružja u igri se provjerava je li skupljena količina novca dovoljna za kupnju određenog oružja te ako je iznos veći ili jednak cijeni predmet se dodaje u popis prikupljenih predmeta glavnoga lika i spreman je za korištenje. Prikaz iznosa prikupljenoga novca te ostalih podataka biti će objašnjen te prikazan u daljnjem tekstu.



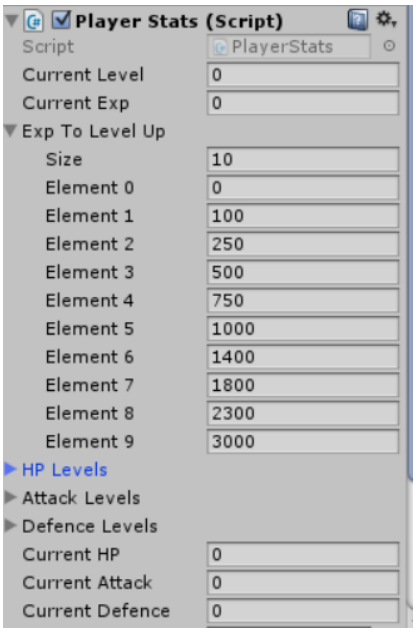
Sl. 3.36 Izgled panela trgovine napitcima za zdravlje

3.9. HUD

HUD (engl. Head Up Display) su pokazivači u igrama koji nam pokazuju i govore brojne informacije. U ovoj igri pomoću njega korisnik zna koliko životnih bodova ima, koliko je razvijen glavni lik, koliko novaca posjeduje, koliko iskustvenih bodova mu treba kako bi došao do nove razine u razvoju lika te naposljetku popis prikupljenih predmeta koje posjeduje glavni lik. Sam sustav za korištenje prikupljenih predmeta je pojašnjen u nekom od ranijih potpoglavlja. Uz sve te podatke HUD također sadrži i mini mapu, odnosno vrstu karte koja prikazuje određene objekte u neposrednoj blizini glavnoga lika. Prilikom igre svi su ovi podatci vidljivi i prikazani su na slici 3.37 te će u daljnjem tekstu biti pojašnjeni.

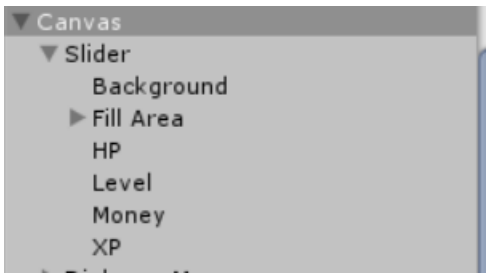
Sl.3.37 HUD igre

Unutar crnoga pravokutnika nalaze se razni podatci. Zelenim slovima su ispisani trenutni životni bodovi glavnoga lika od trenutno mogućih. Iznad brojčanoga ispisa nalazi se i klizač pomoću kojega se grafički prikazuje trenutno stanje zdravlja. Zelena boja u klizaču prikazuje količinu zdravlja dok crvena prikazuje koliko nedostaje od mogućeg ukupnog broja životnih bodova. U ovom slučaju je to 40 od 50. Žuti tekst prikazuje trenutnu količinu novca koju je glavni lik prikupio uništavanjem neprijatelja. Način povećanja trenutnog zdravlja te način na koji korisnik dolazi do novaca i čemu isti služi biti će objašnjen u nekom od idućih potpoglavlja. Crveni tekst prikazuje trenutnu razinu razvoja lika dok tamno crveni tekst pokazuje koliko je iskustvenih bodova potrebno da bi glavni lik otišao na novu razinu (u ovom slučaju je potrebno 100 bodova) odnosno kako bi se unaprijedio. Ovo je vrlo bitno za svaku RPG igru jer u njima se glavni lik uništavanjem neprijatelja razvija, tj. postaje sve jači. Slika 3.38 prikazuje skriptu koja provjera trenutno stanje skupljenih bodova te koliko će se glavni lik unaprijediti ako zadovolji kriterij. U skripti se pomoću definiranog polja cijelih brojeva provjerava je li trenutni broj bodova na određenoj razini dovoljan za povećanje iste.

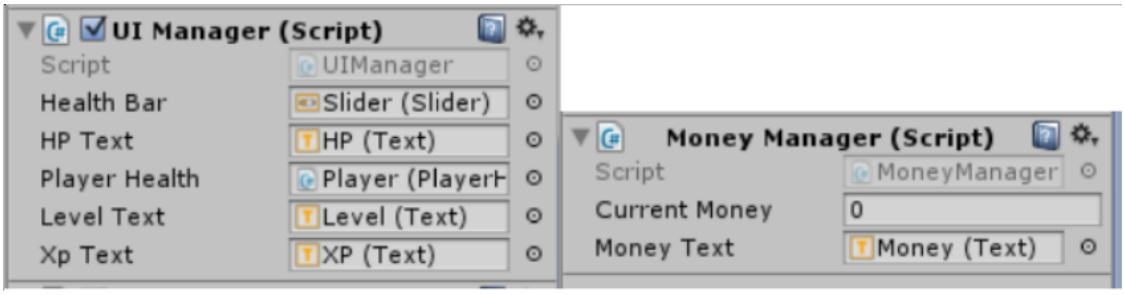


Sl. 3.38 Skripta koja određuje razvoj lika

Kada je prikupljeno dovoljno bodova za unaprjeđenje, glavnom liku se povećava maksimalni mogući broj životnih bodova. Povrh toga, povećava se i napad, odnosno koliko štete nanosi neprijateljima (ovom broju se zbraja i dodatna šteta koju pridonose oružja) te se smanjuje šteta koju mu nanose neprijatelji. Sva tekstualna polja koja čine HUD igre su dio platna. Platno u Unity Engine-u služi za prikazivanje svih elemenata korisničkoga sučelja čega je HUD dio. Navedeno je vidljivo na slici 3.39 dok se na slici 3.40 nalaze dvije skripte koje su dodane kanvasu i služe za ispisivanje svih podataka na ekran.



Sl. 3.39 Kanvas unutar igre sa svim dijelovima korisničkoga sučelja

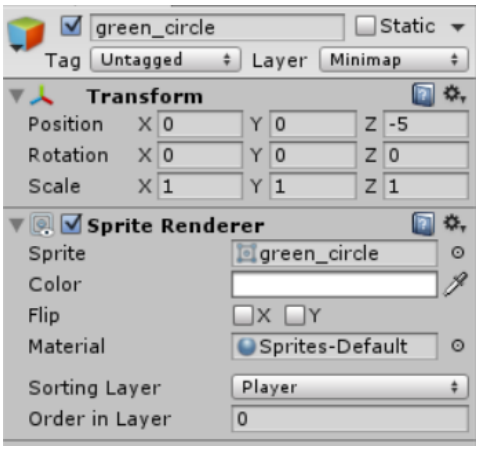


Sl. 3.40 Skripte za upravljanje HUD-om

Unutar crvenog pravokutnika na slici 3.37 nalazi se minijaturna karta koja prikazuje neprijatelje, NPC-ove, mjesta za prijelaz u drugu scenu te trgovine u neposrednoj blizini glavnoga lika. Za potrebe mini mape kreirana je dodatna kamera koja prati glavnoga lika te prikazuje samo objekte koji se nalaze na određenom sloju u igri. Svojstvo kamere pod nazivom „Culling Mask“ određuje koje slojeve će prikazivati odnosno objekte sa kojega sloja. U ovom slučaju radi se o samo jednom sloju pod nazivom „Minimap“.

Sl. 3.41 Kamera koja prikazuje objekte na mini mapi

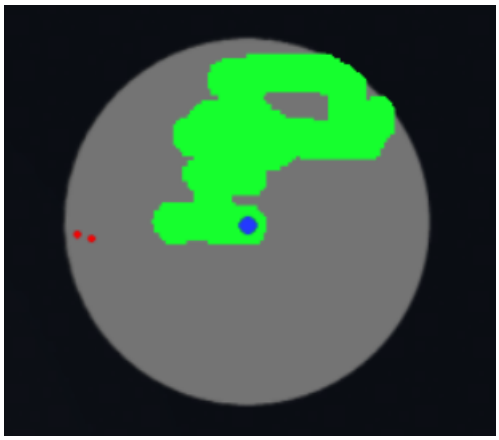
Objekti koji se nalaze na tom sloju su obični krugovi u raznim bojama koje određuju tip objekta (neprijatelj, NPC, trgovina itd.). Kao što je vidljivo na slici 3.42, sloj („Layer“) na kojemu se nalazi objekt je jednak onomu koji kamera prikazuje. Obojani krugovi nisu vidljivi tijekom igranja osim na mini mapi.



Sl. 3.42 Jedan od objekata na sloju koji prikazuje kamera

Sl. 3.43 Obojeni krugovi koje prikazuje mini mapa

Također, prilikom istraživanja tamnica unutar igre površina koju je prešao glavni lik se prikazuje unutar mini mape tako što se taj dio oboji zelenom bojom. Razlog tomu je da tamnice ponekad znaju biti konfuzne te kako se igrač ne bi vrtio u krug ovako može vidjeti koje dijelove je već posjetio i istražio. Prikaz djelomično istražene tamnice je vidljiv na slici 3.44.



Sl. 3.44 Zelena boja prikazuje prijeđene dijelove tamnice

4. ZAKLJUČAK

Za izradu 2D igre pomoću Unity game engine-a potrebno je poznavati osnove objektno orijentiranog programiranja te poznavati C# programski jezik. Uz programski dio izrade diplomskog rada potrebno je i znati koristiti neki od mnogobrojnih programa za uređivanje slika, u ovom slučaju se radi o Paint.NET programu, te program za kreiranje 2D razina Tiled Map Editor.

Prilikom izrade pojavili su se brojni problemi i greške unutar samog programa. Pri osmišljavanju načina kako izraditi igru uvelike su pomogli materijali, kako video tako i tekstualni materijali pronađeni na internetu. Greške koje su se pojavljivale su također rješavane pomoću interneta ili eksperimentiranjem.

Igra je izrađena kao 2D RPG igra pomoću Unity game engine-a. Sama igra se sastoji od više scena u kojima je glavnom liku omogućeno kupovanje predmeta, skupljanje novaca te uništavanje neprijatelja. Razine u kojima se nalaze neprijatelji su proceduralno generirane što pruža veću raznolikost u igri. Postoje razni predmeti koje glavni lik može skupiti te se na kraju igre nalazi glavni neprijatelj čijim uništavanjem igra završava. Prilikom pokretanja igre na zaslonu se pojavljuje glavni izbornik koji sadrži tri tipke, a to su „Play“, „Help“ te „Controls“. Pritiskom tipke „Play“ započinje igra dok se pritiskom tipke „Help“ ispisuju informacije koje pomažu igraču prilikom igre. Naposljetku pritiskom tipke „Controls“ se prikazuju upute za kretanje te akcije kao što su napadanje, korištenje predmeta te odbacivanje istih. Pritiskom tipke „Quit Game“ korisnik napušta igru.

Postoje mnogobrojni dodatni sadržaji koji bi se mogli kreirati kako bi igra bila zabavnija i zanimljivija te kako bi igra bila duža. Ovdje se misli na dodavanje više vrsta neprijatelja, mogućnost spremanja trenutnog stanja, poboljšanje priče igre te veći broj predmeta. Osim toga, postoje mogućnosti implementiranja različitih vrsti napadanja i boljih animacija, kako glavnog lika tako i neprijatelja, ubacivanje raznolikije glazbe i zvukova prilikom izvođenja akcija te naposljetku izrada više razina sa više detalja.

5. LITERATURA

[1] https://unity3d.com/learn/tutorials

[2] https://docs.unity3d.com/Manual/index.html

[3] https://docs.unity3d.com/ScriptReference/

[4] https://en.wikipedia.org/wiki/Paint.NET

[5] http://doc.mapeditor.org/en/stable/

[6] Sue Blackman, Jenny Wang, „Unity for Absoulte Beginners, 1st Edition“

[7] Jon Skeet, „C# in Depth, 3rd Edition“

[8] http://pcg.wikidot.com/

[9] http://roguebasin.roguelikedevelopment.org/index.php?title=Main_Page

[10] https://unity3d.college/category/unity3d/

[11] Ian Griffiths, „Programming C# 5.0“

[12] https://www.gamasutra.com

6. SAŽETAK

6.1. Sažetak

Tema ovog diplomskog rada je izrada „2D RPG igre sa proceduralno generiranim razinama“. Za izradu igre je korišten Unity game engine, program za uređivanje slika Paint.NET i program za kreiranje 2D razina Tiled Map Editor. Kako bi se igra izradila potrebna su znanja o objekto orijentiranom programiranju jer je igra kreirana pomoću C# programskog jezika. Prilikom pokretanja igre pojavljuje se glavni izbornik u kojemu je moguće pritiskom određenih tipki klikom miša započeti prelazak igre, provjeriti upute za upravljanje glavnim likom, te pročitati informacije koje mogu pomoći prilikom igranja. Upravljanje glavnim likom se odvija pritiskom tipki na tipkovnici. Igra se sastoji od više razina od kojih su neke proceduralno generirane što pruža veću raznolikost tako da svako prelaženje bude drugačije. Prilikom prelaženja igra navodi korisnika što mu je sljedeći korak te naposljetku postoji borba protiv glavnog neprijatelja čijim uništenjem završava igra. Padom razine zdravlja glavnog lika na nula nastupa „Game Over“, odnosno kraj igre te se korisniku omogućava da ponovno pokuša prijeći igru.

6.2. Summary

The topic of this graduate thesis is creating a „2D RPG game with procedurally generated levels“. The programs used for creating this game are Unity game engine, image editing program Paint.NET and a program for creating 2D levels Tiled Map Editor. In order to design the game, a knowledge of object oriented programing is required, since the game is created using the C# programming language. When starting the application, the main menu appears, in which you can start the game, get instructions how to control the main character, as well as reading about information that can be of help while playing the game. By pressing the buttons on the keyboard, you can control the main character. The game consists of multiple different levels of which some are procedurally generated which provides a greater variety so every time the user plays it feels

different. While playing the game it leads the user so he knows what his next step should be. At the end there is a boss fight in which if the user defeats the enemy the game ends. If the health of the main character drops to zero the game ends and results in a “Game Over” screen and the user is given a chance to try again.

7. ŽIVOTOPIS

Matej Umiljanović rođen je 11.1.1994. godine u Našicama. Osnovnu školu je upisao 2000. godine te istu završava 2008. godine. Nakon osnovne škole upisuje srednju ekonomsku školu u srednjoj školi Isidora Kršnjavog u Našicama. Srednju školu završava 2012. godine te iste godine upisuje preddiplonski studij smjer računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija. 2015. godine završava preddiplomski studij te upisuje diplomski studij smjer informacijske i podatkovne znanosti. Član je nogometnog kluba NK „Martin“ koji se natječe u 3. županijskoj ligi. 2018. godine se zapošljava u tvrtki za razvoj software-a „Mono“ u Osijeku gdje je i trenutno zaposlen.