**RAZVOJ RAČUNALNIH IGARA**

ZAVRŠNI PROJEKT

**ZMIJA**

Student: Karlo Sever

Profesor: Nenad Breslauer, v. pred.

Sadržaj

[1. Opis igre 3](#_Toc45546974)

[2. Postupak izrade 4](#_Toc45546975)

[2.1 Postavljanje scene s vizualnim elementima 4](#_Toc45546976)

[2.2 Programski kod 9](#_Toc45546977)

[3. Zaključak 12](#_Toc45546978)

# Opis igre

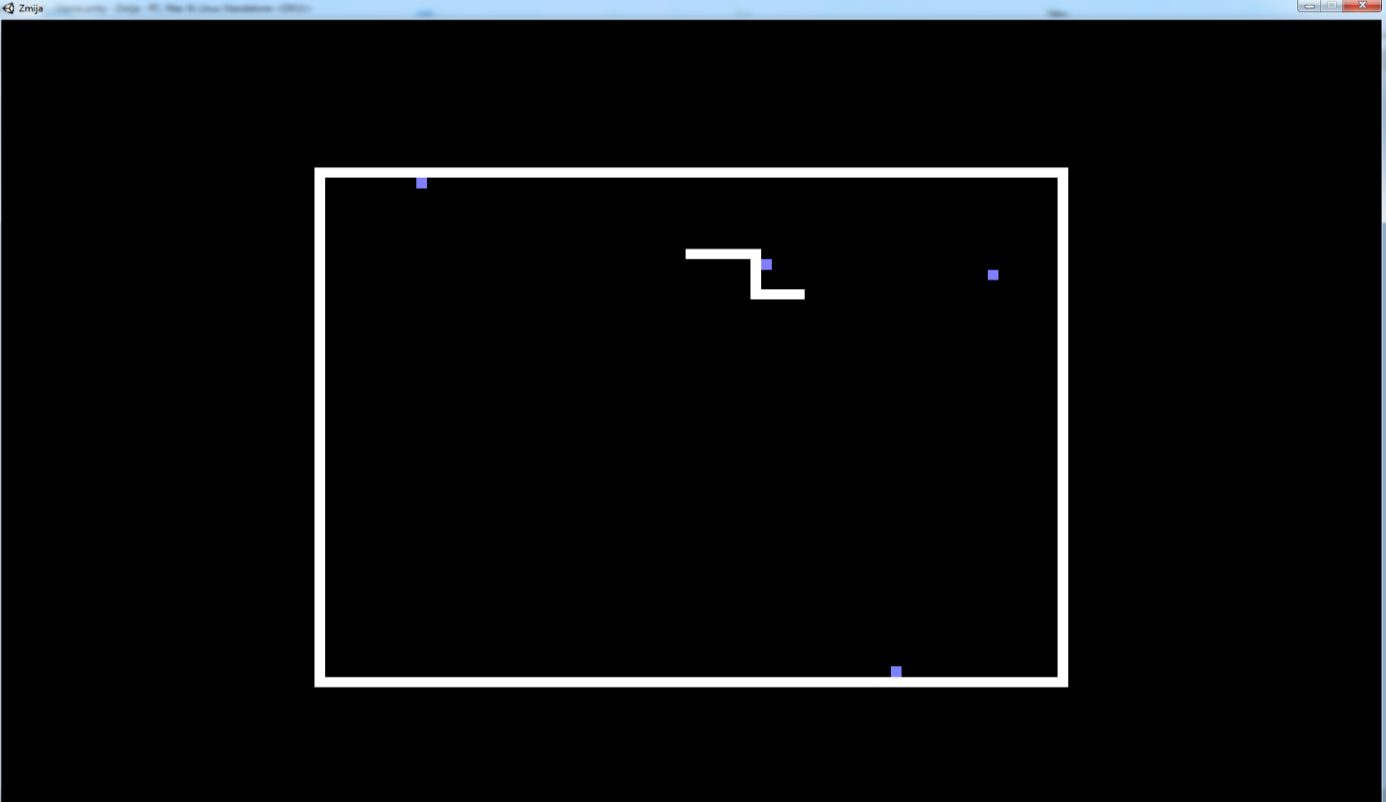
Naslov: Zmija

Platforma: PC

Žanr: 2D arkada

Ciljana publika: 5-12 god.

„Zmija“ je klon klasične igre koja potječe iz 70-ih te je kroz godine prošla mnoge verzije i platforme. Mehanika igre je kroz godine ostala manje-više ista. Igrač upravlja sa zmijom koja se kreće unutar četiri zida te je cilj proći kroz čim više objekata unutar zidova prilikom kojih se tijelo zmije produžuje i time zauzima sve više mjesta unutar prostora. Težina se progresivno povećava s čim više objekata zmija ima interakciju. Ono što otežava igraču sakupljanje tih objekata je da nije dozvoljena kolizija sa zidom ili vlastitim repom zmije.

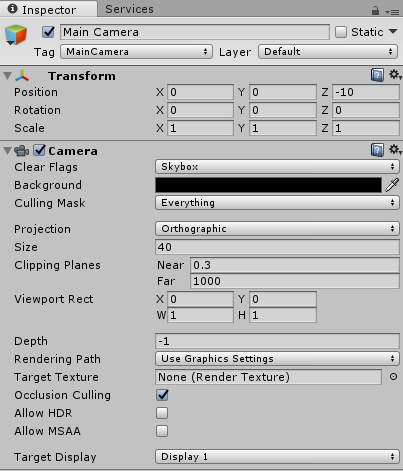


Slika 1. Izgled igre, izvor: autor

# Postupak izrade

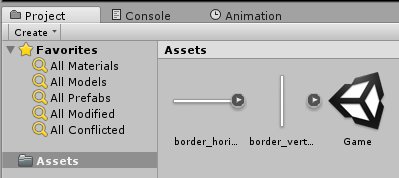
## 2.1 Postavljanje scene s vizualnim elementima

Za izradu igre koristio se Unity 5.6.0f3 te je izrađena u 2D formatu. Za početak potrebno je namjestiti svojstva pozadine i pozicioniranje glavne kamere



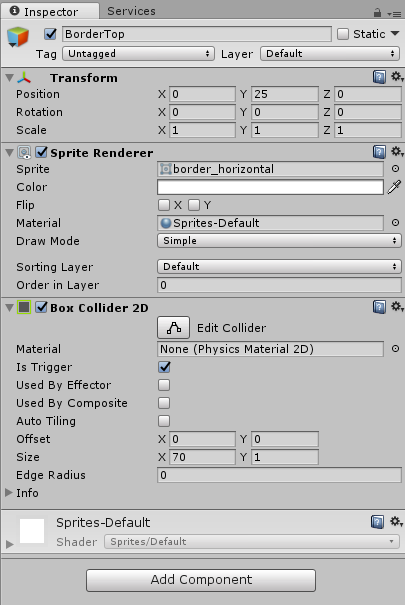
Slika 2. Svojstva kamere i pozadine, izvor: autor

Zatim je potrebno dodati „zidove“ odnosno granice kud se zmija može kretati. Potrebno je izraditi sliku zida, jednu koja je vertikalna i jednu horizontalnu. Nije potrebno ručno izraditi 4 slike pošto se mogu kopirati. Prvo se dovuku slike u područje Assets.



Slika 3. Sadržaj projekta koji sastoji scenu i slike zidova, izvor:autor

Nakon što slike ubacimo u Unity scenu potrebno je namjestiti postavke zidova. Bitno je namjestiti istu veličinu kao i kopija ali sa suprotnim vrijednostima pozicije.



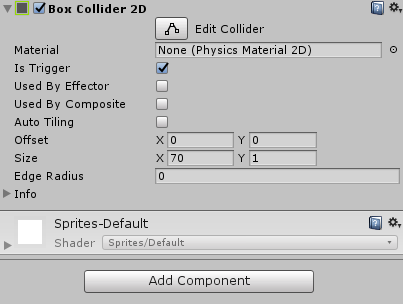
Slika 4. Postavke zida, izvor: autor

U hijerarhiji će se imena zidova preimenovati kako je prikazano na sljedećoj slici.



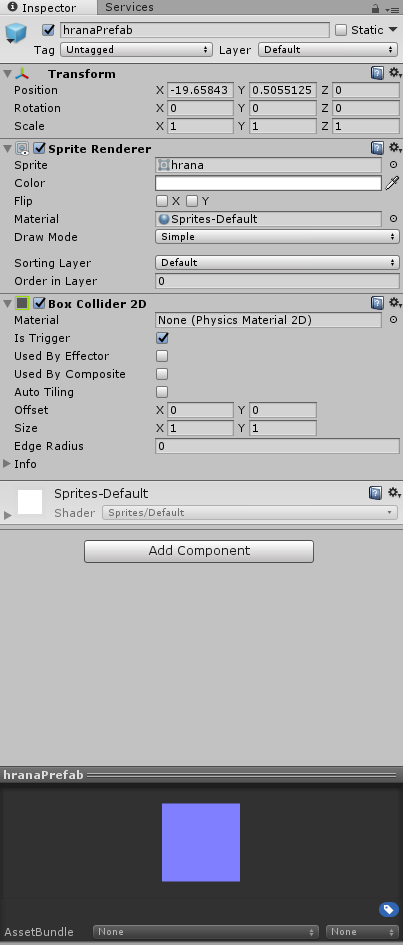
Slika 5. Imena zidova, izvor: autor

Dodavanje samo slika zidova nije dovoljno da zadrži zmiju unutar prostora već je potrebno dodati element koji daje fizička svojstva zidovima. Na svaki zid se dodaje komponenta „Box Collider 2D“.



Slika 6. Box Collider 2D svojstva, izvor: autor

Potrebno je dodati i objekt koji će služiti kao hrana zmiji te će se stvarati uzastopno nakon određeni broj sekundi. Kao prikaz tog objekta koristit će se slika malog plavog kvadrata. Osim toga imat će slična svojstva kao i zid jedino što će se kod kolizije uništiti. U postavkama će se također dodati Box Collider 2D ali sa „Is Trigger“ funkcijom.



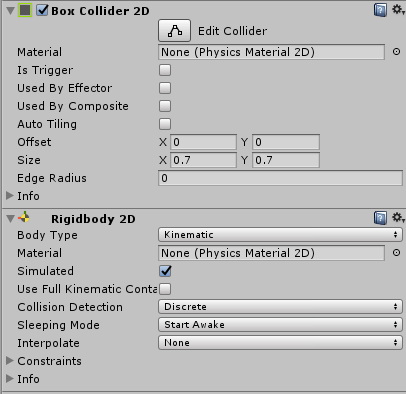
Slika 7. Postavke objekta hrane, izvor: autor

Od objekta hrane trebamo napraviti Prefab što znači da ćemo taj element više puta pozivati tj. stručnije rečeno „Instantiate“. Iz hijerarhije se dovuće u Assets prozor.

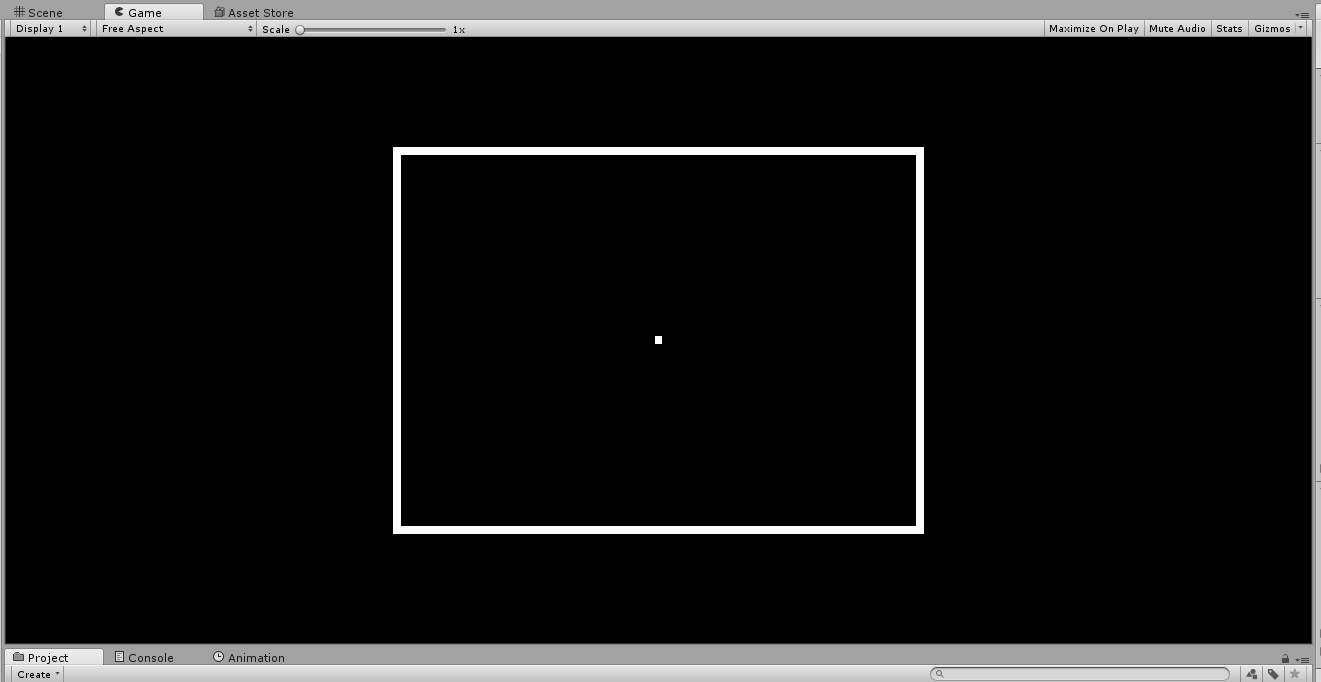


Slika 8. Assets, izvor: autor

Kao i kod prijašnjih elemenata za zmiju će se koristiti slika kvadrata samo u bijeloj boji. Te će njen objekt biti pod nazivom „Glava“ u hijerarhiji. Također je potrebno dodati Box Collider 2D jer će reagirati u doticaju s ostalim objektima. Osim toga pošto se zmija kreće, takva tijela moraju sadžavati Rigidbody komponentu. U Box Collider elementu smo veličinu na 0.7 da zmija nije u koliziji sam sa sobom te je i gravitacija na 0 jer će inače zmija pasti na dno. Još će se objekt glave skopirati kao Prefab jer će se dodavati na kraj zmije kao element repa.



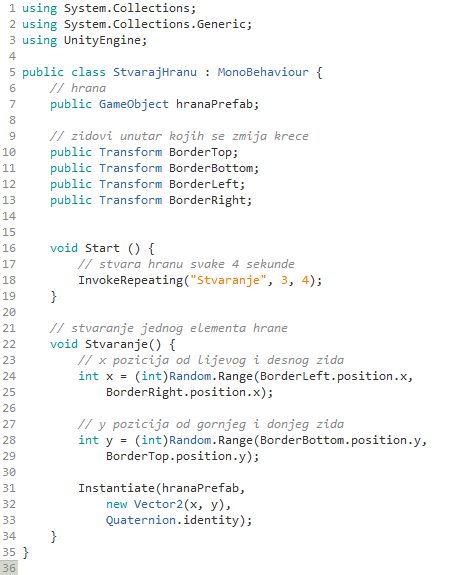
Slika 9. Svojstva glave od zmije, izvor: autor



Slika 10. Trenutni izgled scene sa zidovima i glavom zmije, izvor: autor

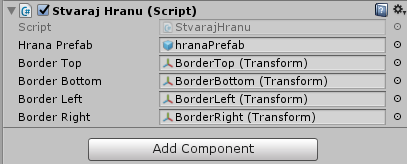
## Programski kod

Trebat ćemo skriptu koja će stvarati hranu za zmiju periodično. Za to koristimo MonoDevelop program unutar Unity-a koji se piše u C# jeziku. U hijerarhiji označimo Main Camera i dodamo novu komponentu New Script pod nazivom „StvarajHranu“. Dalje je prikazani programski kod te je pojašnjeno čemu koji blok kodova koristi.



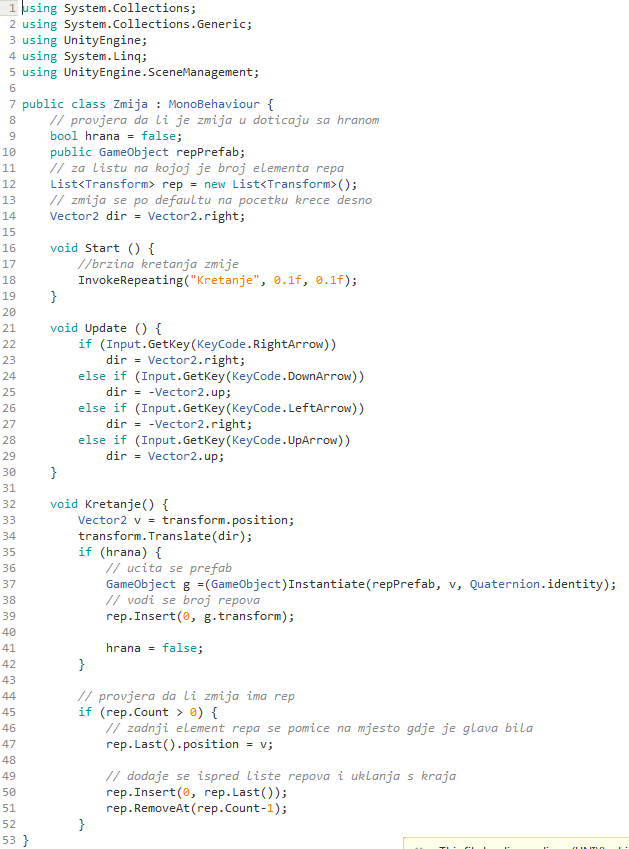
Slika 11. Programski kod za stvaranje hrane.

Ako označimo u hijerarhiji Main Camera u Inspectoru vidimo da po skriptom nudi varijable koje trebamo ubaciti unutra. Iz hijerarhije dovućemo sva četiri border elementa u odgovarajuća polja i hranaPrefab iz assets-a u Prefab polje. Na slici ispod je prikazano kako moraju izgledati polja od skripte.



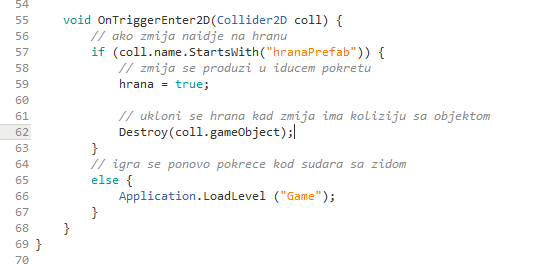
Slika 12. Elementi skripte, izvor: autor

Idući korak moramo označiti element „Glava“ iz hijerarhije te dodati novu komponentu New Script pod nazivom „Zmija“. Svrha tog programskog koda je kako će se zmija kretati, tj. kojom brzinom će se kretat te će sadržavati kod za produljivanje repa zmije. Također sadrži naredbe u slučaju da zmija dođe u koloiziju sa zidom te se pri koliziji igra vraća na početno stanje.



Slika 13. Kod za kretanje zmije i dodavanje repa na kraj zmije, izvor: autor

Na kraju programskog koda u skripti „Zmija“ je kod što se tiče kolizije zmije sa hranom i zidovima. Kad zmija dođe u doticaj sa hranom taj objekt hrane se uništava. U slučaju da zmija dotakne zid igra se resetira na početak.



Slika 14. Kod za koliziju s objektima, izvor: autor

# Zaključak

Igra zmija je jedonstavnije naravi za izradu te danas ima mnogo verzija na istu ideju sa različitim izazovima i preprekama. U povijest ulazi kao jedna od klasičnih arkadnih igara preko mnoštvo platforma i puno generacija ljudi. Na ranim mobitelima krajem 90-ih, točnije na modelu Nokia poslužila je kao odličan način razbijanje dosade. Osim single player načina igranja postoji i multiplayer te većina web stranica posvećene igricima sadrže i više verzija klasične igre zmije.