VISOKA ŠKOLA STRUKOVNIH STUDIJA

ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

INTERNET TEHNOLOGIJE

**Razvoj “game engine-a” u Angular-u**

Završni rad

Mentor: Student:

Dr. Nenad Kojić, dipl inž Uroš Ćirić 50/15

Beograd, 2019.

VISOKA ŠKOLA STRUKOVNIH STUDIJA

ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

INTERNET TEHNOLOGIJE

Predmet: Web programiranje

Tema: **Razvoj “game engine-a” u Angular-u**

Ocena \_\_\_ ( \_\_\_\_\_\_\_\_\_ )

Članovi komisije:

\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Sadržaj

[1 Sadržaj 3](#_Toc8773362)

[2 Uvod 4](#_Toc8773363)

[3 Radno okruženje 5](#_Toc8773364)

[4 Organizacija 6](#_Toc8773365)

[4.1 Engine 7](#_Toc8773366)

[4.2 Screen 9](#_Toc8773367)

[4.3 Entities 10](#_Toc8773368)

[4.4 Collision 11](#_Toc8773369)

[4.5 Controls 11](#_Toc8773370)

[4.6 Sound 11](#_Toc8773371)

[4.7 UI 12](#_Toc8773372)

[4.8 Entity 12](#_Toc8773373)

# Uvod

Kada se spomene razvoj igara, programski jezici koji padaju na pamet su C++, C#, Java, Python i njihove varijacije, uglavnom korišćeni u određenom razvojnom okruženju za pravljenje igrica („game engine“).

Game engine u sebi sadrži suštinsku logiku funkcionalnosti kao što su prikaz 2D i 3D grafike, fiziku, detekciju sudaranja, zvuk, animaciju, veštačku inteligenciju... Neki od popularnijih engine-a kao što su Unreal i Unity zadovoljavaju potrebu razvijanja većine igara. Danas su retki slučajevi gde pravljenje neke igre zahteva pisanje sopstvenog engine-a, takvi slučajevi bi zahtevali veći razvojni tim.

Ovaj projekat sam odlučio da pišem u TypeScript-u - proširenju JavaScript-a, a za okruženje sam odabrao Angular. Za prikaz 2D grafike će biti zaslužan HTM5 Canvas. Od malobrojnih engin-a za browser igre odlučio sam se da ne koristim nijedan, već u osnovi, da napišem svoj.

Ideja jeste korišćenje manje zastupljenih tehnologija i gotovih kodova kako bi se bolje razumela osnova i suština pri radu na sličnim projektima.

# Radno okruženje

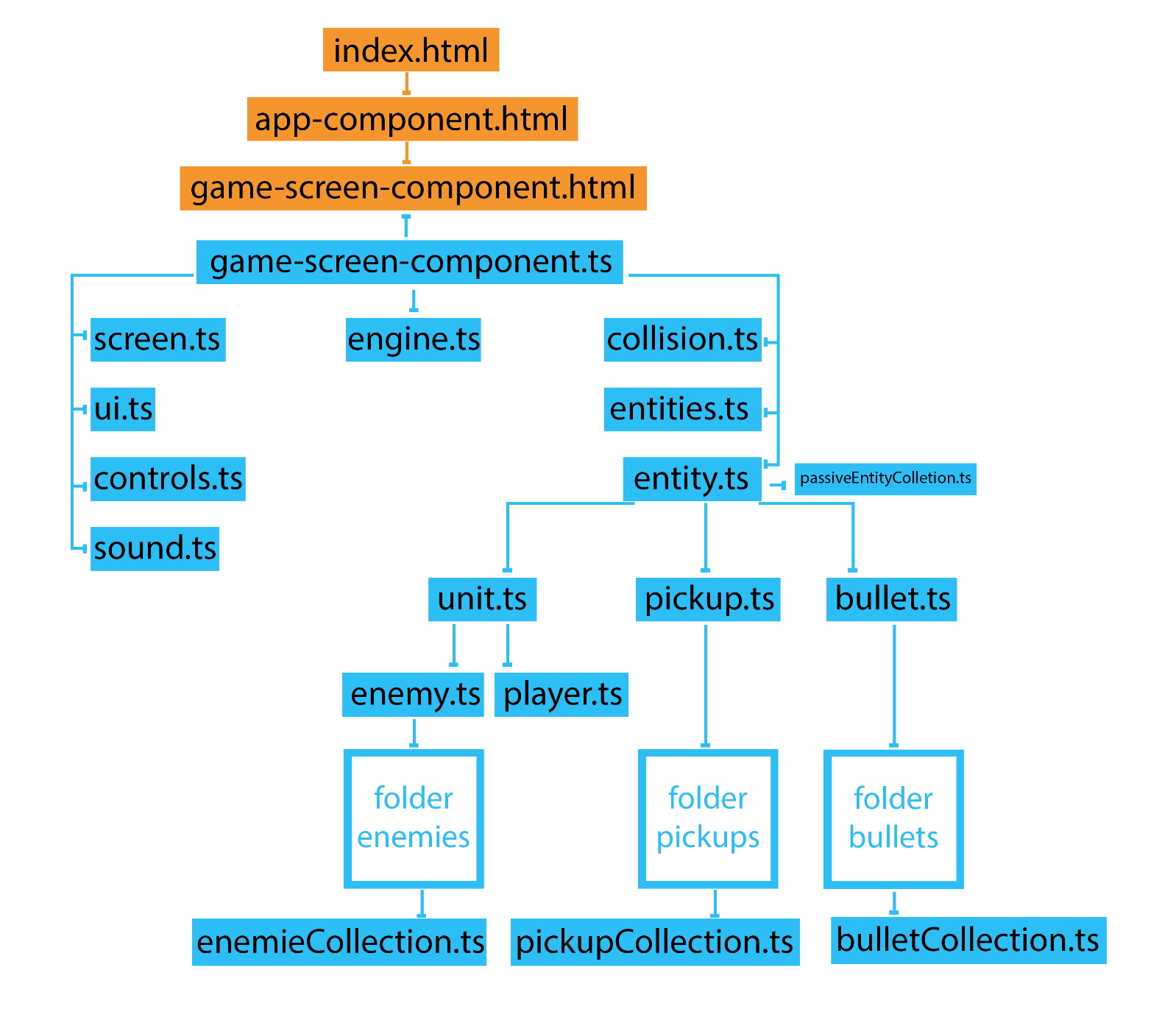
**Pri razvoju igrice korišćeno je razvojno okruženje:**

* Visual Studio Code

**Programski jezici koji su korišćeno za razvoj:**

* HTML5
* CSS3
* Sass
* TypeScript
* Angular 6.1.5

# Organizacija



U folderima enemies, pickups i bullets, nalaze se podklase tih roditelja (enemy1.ts, enemy2.ts, enemy3.ts…). Većina linija predstavlja uzajamnu parent-child vezu.

* 1. Engine

Engine je zaslužan za ključne operacije kao što su pokretanje i pauziranje igre, učitavanje i promenu nivoa.

Treba napomenuti da je struktura nivoa zapisana u odvojenim JSON fajlovima. Ta struktura sadrži:

* Naziv nivoa
* Poruku igraču koja se ispisuje pri učitavanju nivoa
* Pozadinu nivoa (više o tome u nastavku)
* Muziku koja se pušta dok se odvija igra
* Talase protivnika sa informacijama o tome gde, kad i u kom redosledu da se prikažu

Ovde se taj JSON fajl obrađuje i pretvara u objekat. Poštuje redosled odvijanja događaja.

U ovom fajlu se takođe nalazi detekcija aktivnosti korisnika i trenutno vreme u milisekundama – u zavisnosti od potreba, nekad se koristi to vreme, a nekad brojanje pomenjenih frejmova.

Praćenje aktivnosti korisnika je neophodno uslednovih novih izmena internet browsera gde neophodna njegova interakcija sa sajtom kako bi mogli da mu se putem JavaScript-a puste video snimci i zvukovi.

Primer strukture nivoa:

{

"title": "Level 1",

"subtitle": "Get ready!",

"background": [

{"src" : "river.png", "speed": 20, "width":288, "height": 512},

{"src" : "", "speed": 3, "height": 100, "width": 300, "size":50, "margin":0, "place": "right", "delay": 0}

],

"music":"soundtrack\_pop\_is\_just\_a\_matter\_of\_preference.mp3",

"waves":

[

{

"pause" : 2000,

"requireClear" : true,

"enemies":

[

{"id":1,"pause":1000, "pos": ["A14", "B14", "J14"]},

{"id":1,"pause":0, "pos": ["A5", "B5", "F5"]},

{"id":1, "pos": ["A23","B23", "F23"]}

],

"pickups":

[ ]

},

{

"pause" : 2000,

"requireClear" : true,

"enemies":

[

{"id":1,"pause":1000, "pos": ["A14", "B14", "J14"]},

{"id":1, "pos": ["A10", "B10", "F10"]},

{"id":1, "pos": ["A6","B6", "C6"]},

{"id":1, "pos": ["A18", "B18", "F18"]},

{"id":1, "pos": ["A22","B22", "C22"]}

],

"pickups":

[{"id":2,"pos": "a14", "pause":0}]

},

{

"pause" : 2000,

"requireClear" : true,

"enemies":

[

{"id":2,"pause":1000, "pos": ["A14", "B14", "J14"]}

],

"pickups": [ ]

}

]

}

* 1. Screen

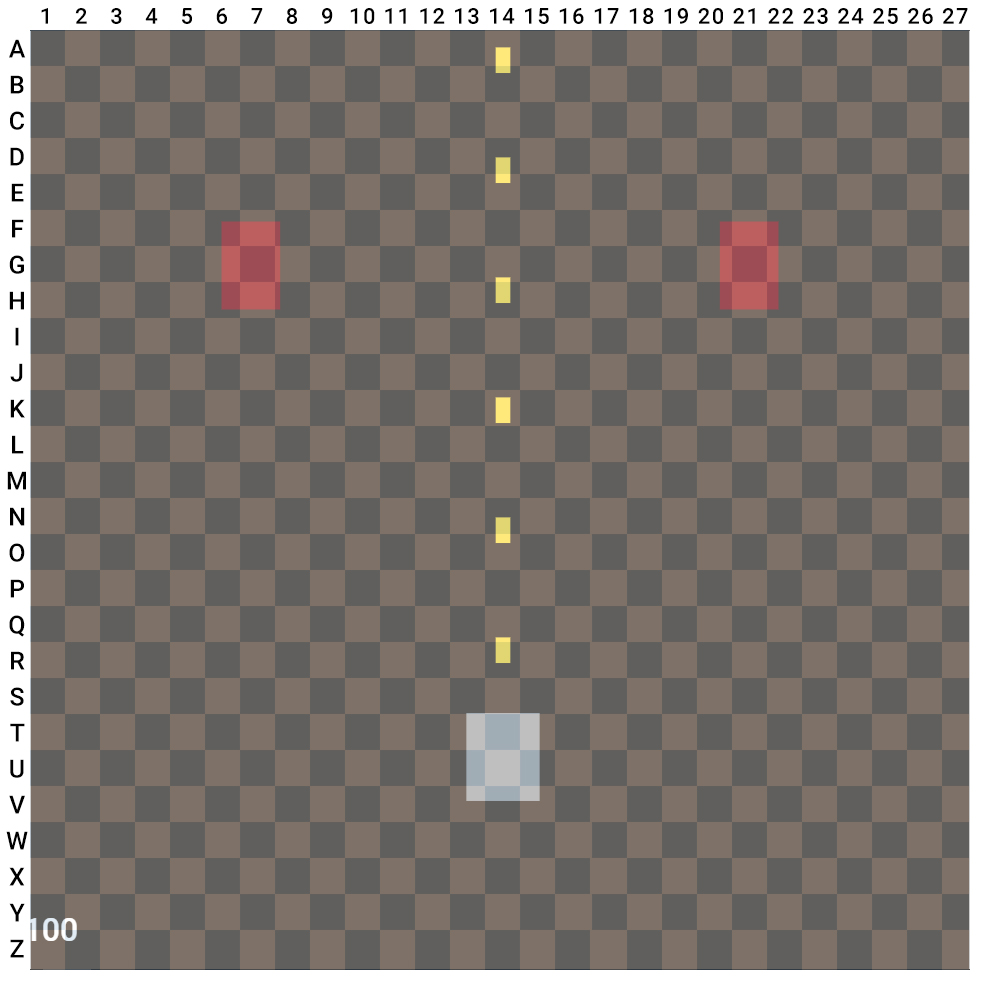
Ovde se nalazi sva logika zaslužna za vizuelni prikaz dešavanja. Ta logika podrazumeva sledeće:

## Upravljanje canvas-om

U ovom enginu postoje 3 canvas-a: po jedan za igrača, protivnike i municiju. Validan pristup bi bio i korišćenje jednog canvas-a, ali po maloj memorijskoj ceni, dobija se lakša kontrola. Pri učitavanju igre, oni se inicijalizuju. U slučaju promene dimenzije ekrana, canvas se prilagođava tako da su mu širina i dužina jednaki i da zauzima najviše mogućeg prostora.

Svaki canvas je zaslužan da renderuje – iscrtava svoje određene entitete. Kako većina monitora radi na 60 Hz, tako se i svaki canvas prazni i iscrtava 60 puta u sekundi.

## “Spawnovanje” – stvaranje protivnika i entiteta



Kako bi se omogućilo lakše stvaranje entiteta u koordinatnom sistemu, X i Y ose su podeljene na slova i brojeve, nalik obrnutoj šahovskoj tabli.

gridPos(**M14**) bi nam vratio poziciju blizu sredine, **B25** bi vratio gornji desni ugao, **X4** donji levi itd...

## Upravljanje pozadinskom slikom

Pozadina ovde se ne iscrtava u canvas-u već putem CSS-a i sastoji se iz više slojeva koji se kreću različitom brzinom kako bi dali efekat dubine prostora – “parallax”.

Pozadinski slojevi se kreću konstantnom brzinom. Najniži sloj mora da pokriva celu pozadinu i njegov početak i kraj moraju biti vertikalni odrazi kako bi slika mogla se nastavlja bez cepkanja.

* 1. Entities

Entities sadrži funkcije inicijalizacije i pomeranja entiteta koje se izvršavaju promenom svakog frejma kako bi se svaki entitet konstantno pomerao.

## Inicijalizacija protivnika i ostalih entiteta

Protivnicima se pri inicijalizaciji zadaju barem dve koordinatne tačke npr. B i C i one predstavljaju početnu i krajnju tačku putanje. Na te dve tačke izračunava se i treća tačka A koja koja je na istoj udaljenosti od tačke B koliko je i tačka B, samo orijentisana u suprotnom smeru. Time se postiže da se protivnik ne stvori na sred kanvasa, već da “uleti” sa najbliže tačke van njega.

## Pomeranje entiteta

U svakom frejmu entiteti se pomeraju. Protivnici se pomeraju po zadatoj liniji i stanu kada dođu na kraj, metkovi se uglavnom pomeraju vertikalno nadole ili nagore, stacionarni entiteti se pomeraju vertikalno nadole.

Radi čuvanja memorije briše se svaki entitet koji se pomeri van ekrana.

Pri ovom pomeranju protivnici takođe napadaju u određenom vremenskom intervalu.

* 1. Collision

Logika za detekciju sudaranja. Svaki entitet je predstavljen u formi pravougaonika. Ovde se ispituje da li u datom trenutku ivica jednog entiteta ne preklapa ivicu drugog.

U slučaju da se ivice entiteta preklapaju npr metka i protivnika, metak se briše (ako je tako podešen), a protivnik gubi toliko životnih poena koliko metak nosi poena za napad. U slučaju da protivnik nema preostale životne poene, briše se i on.

Pri brisanju entiteta, moguće je puštati zvukove i određene animacije tog entiteta.

* 1. Controls

Kod kontrola, sluša se aktivnost tastature i miša. Pri pomeraju miša, avatar igrača ubrzava ka kursoru. Ubrzavanje ovde je neophodno, da se avatar nalazi konstantno na kursoru, teže bi bilo sprečiti varanje u igri sa metodom gde se igra pauzira, zatim se pomeri miš i igra se nastavi. Takođe postoji izbor korišćenja strelica na tastaturi.

Levim klikom se aktivira specijalni potez.

U slučaju da se igrač pomeri van ekrana, avatar će ostati na samoj ivici.

* 1. Sound

Funkcija koja pravi HTML5 audio elemente.

Inicijalno su svi zvuci utišani dok korisnik ne interaktuje sa sajtom, u suprotnom zbog novih pravila internet pretraživača, zvuci ne bi mogli da se puštaju.

Zvuci se mogu puštati neprekidno kao pozadinska muzika ili jednom kao posledica stvaranja ili sudaranja entiteta.

* 1. UI

UI ili User Interface je zaslužan za sve ostalo što se prikazuje korisniku, a ne pripada kanvasu:

* Prikaz uvodnih video klipova
* Prikaz završnog ekrana u zavisnosti da li igrač pobedio ili izgubio
* Prikaz menija za vreme pauze
* Prikaz i promena životnih poena, energetskih poena i poena za finalni rezultat.
  1. Entity

Entity je glavna klasa za svaki entitet što se iscrtava na ekranu. Ti entiteti uključuju samog avatara koga igrač kontroliše, protivnike, prijateljske i neprijateljske napade, entitete koje igrač može da pokupi prelaskom preko njih i pasivne entitete koji uglavnom postoje iz estetskih razloga.

Podklase od entiteta su:

1. Unit
   1. Player
   2. Enemy
2. Bullet
   1. FriendlyBullet
   2. EnemyBullet
3. Pickups

Konačno, ove podklase nasleđuju njihove podklase; primer: klasu Enemy nasleđuju klase Enemy1, Enemy2, Enemy3 itd....

Osobine koje sadrži svaki entitet su:

* Identifikacioni broj i naziv
* Stanje u kome je entitet (stvaranje, mirovanje, umiranje...)
* X i Y koordinatke gde se trenutno nalazi
* Visina i širina
* Brzina i ubrzanje
* Životni poeni i poeni za napad
* Niz zvukova koji se puštaju u prikladnom trenutku
* Podatke za animaciju
  + Spritesheet – niz frejmova za animaciju u jednoj slici
  + Visinu i širinu spritesheet-a

Svaki entitet u sebi ima funkcije za:

* Renderovanje ili crtanje
* Pomeranje
* Napadanje
* Animiranje
* Promene veličine u odnosu na ekran
* Puštanje zvuka