

## Igra zmijice (Projektan zadatak)

# Predmet CS324 - Skripting jezici

Profesori:

Dr. Igor Franc

Asistent:

Katarina Banovic Student: Nikola Nesovanovic 4284

# Sadrzaj:

## 1.Uvod……………………………………………………………………

## 2.Dijagrami?

## 3.Opis koriscenih tehnologija………………………………………….

## 3.1. React………………………………………………………….

## 3.2. NodeJS……………………………………………………….

## 3.3. Node Package Manager…………………………………....

## 4.Struktura projekta…………………………………………………….

## 5.Pokretanje projekta…………………………………………………..

## 6.Zakljucak……………………………………………………………....

## 7.Reference……………………………………………………………..

1.Uvod

Cilj projektnog zadatka ‘Igra zmijice’ je da se napravi klon klasicne ‘snakes’ igrice za internet pretrazivace, koja je napravljena pomocu tehnologija koji omogucavaju dalji development i integraciju sa lakocom. Posto je korisnicki interfejs radjen pomocu react-a, igricu je moguce lako prilagoditi za android i ios platforme, koristeci react-native tehnologije.

**NodeJS** – NodeJS je programski jezik pisan u JavaScript-u koji je korišćen kao rešenje za serverski deo našeg projekta. Koristi Google-ov V8 JavaScript engine kao način interpetiranja. Open source je i takodje je napravljen da funckioniše na svakoj platformi.

**React** - React je javascript biblioteka koja se koristi za dinamicno i interaktivno razvijanje korisnickih interfejsa za veb pretrazivace.

??????dodaj js i cmder i jos par stvari???????? Package json???

2.

3. Opis koriscenih tehnologija:

## 3.1. React:

React je javascript biblioteka koja se koristi za dinamicno i interaktivno razvijanje korisnickih interfejsa za veb pretrazivace.

## 3.2. NodeJS:

NodeJS je programski jezik pisan u JavaScript-u koji je korišćen kao rešenje za serverski deo našeg projekta. Koristi Google-ov V8 JavaScript engine kao način interpetiranja. Open source je i takodje je napravljen da funckioniše na svakoj platformi.

## 3.3. Node Package Manager:



Node Package Manager je podrazumevani menadžer paketa za JavaScript runtime environment NodeJS. Od 0.6.3 verzije NodeJS-a, Node Package Manager je automatski instaliran sa okruženjem i da bi se danas koristio dovoljno je samo instalirati NodeJS okruženje. Node Package Manager je potpuno napisan u JavaScript-u i razvijen je od strane Isaac Z. Schlueter-a. Prvo izdanje Node Package Manager-a je objavljeno 12.1.2006, dok je poslednje stabilno izdanje objavljeno 6.5.2016 godine. Sve Node Package Manager komande je moguće pokrenuti iz terminala ili Command Prompt-a u zavisnosti od operativnog sistema korisnika.

4. Struktura projekta:

## 4.1. Opis strukture direktorijuma i fajlova:

Struktura foldera i fajlova je jako bitna za projekat kada na njemu rade vise osoba ili se development na projektu nastavlja posle duzeg vremena. Node package manager sam odvaja sve NodeJS module u node\_modules folder, kao i odvaja sve korisnicke fajlove u public i src foldere (public za fajlove koji su vidljivi krajnjem korisniku i src za ‘back-end’ fajlove).

### 4.1.1. Node\_modules direktorijum:

Node package manager pri pokretanju npm i komande kreira preko 1000 foldera sa svim mogucim bibliotekama i modulima koje je moguce koristiti sa NodeJS-om.??????????????????????????????????????????????????????????????????????????????????????

### 4.1.2. Public direktorijum:

Public direktorijum ne sadrzi javascript, css ili slicne korisnicke fajlove, od kojih se sastoji glavni deo nase aplikacije. U njemu se nalaze samo krajnji fajlovi koji su dostupni krajnjem korisniku.

**Index.html** je prosto template koji sadrzi html kod, koji sadrzi meta informacije o nacinu prikazivanja aplikacije na stranici, kao i ‘root div’ u koji cemo smestati sav react generisan kod.

**Manifest.json** je fajl koji sadrzi informacije o aplikaciji poput imena, ikone kao i osnovnih nacina prikaza.

**Favicon.ico** je ikonica aplikacije tj stranice.

### 4.1.3. Src direktorijum:

Src direktorijum sadrzi sve skripte koje kontrolisu rad aplikacije.

**useInterval.js** je fajl koji sadrzi specijalni ‘hook’ koji je razvio Dan Abramov. ‘Hook’-ovi su specijalne komande nalik event listener-ima. Oni vezuju promenljivu sa specijalnom funkcijom, koja menja vrednost promenljive pri njenom pozivu i usput pozivaju ponovni render react elemenata. ‘useInterval’ je hook koji se poziva posle odredjenog intervala vremena i u ovom programu ga koristima kako bi omogucili da se ‘gameLoop’ funkcija poziva posle konstantom odredjenog vremenskog perioda.

**Index.js** prosto vezuje react elemente generisane **app.js** fajlom, sa root div elementom koji se nalazi u **index.html** fajlu.

**Constants.js** je fajl koji sadrzi konstante koje kontrolisu pocetnu poziciju zmije i jabuke, kolicinu i velicinu tajlova i brzinu zmije. Takodje ovde se nalaze pomocne konstante poput ‘key code’-ova strelica kao i vektora direkcije koji odgovaraju pravcu strelice.

**App.js** je fajl u kome se nalazi glavni deo koda.

5. Pokretanje projekta:

6. Zakljucak:

7. Reference: