

**IMPLEMENTACIJA SIMULATORA ZA GAZEPOINT EYE TRACKER KAMERU****GAZEPOINT EYE TRACKER CAMERA SIMULATOR IMPLEMENTATION**Nenad Gligorov, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSTVO I AUTOMATIKA**

**Kratak sadržaj** – Eye tracking je senzorska, hardverska tehnologija koja omogućava računarima ili drugim uređajima praćenje korisničkog pogleda. Eye tracker kamera može da detektuje prisustvo, pažnju i fokus korisnika. Omogućava jedinstven uvid u razumevanje ljudskog faktora prilikom korišćenja računara i olakšava izradu što prirodnijeg korisničkog interfejsa za širok spektar digitalnih uređaja. Takođe, mogućnost upravljanja računarom uz pomoć očiju predstavlja potencijalno olakšanje za korisnike kojima je onemogućen govor ili korišćenje ruku. Ovakav tip kamere je veoma skup, te je zbog toga nepristupačan većini populacije. Potencijalno rešenje ovog problema može da bude Gazepoint Eye Tracker simulator. U radu je detaljno opisan postupak implementacije softvera, način njegovog rada, kao i princip određivanja simuliranog korisničkog pogleda. Za izradu simulatora korišćen je radni okvir Angular, koji se odnosi na prezentacijski sloj aplikacije, dok će se za samu biznis logiku pobrinuti radni okvir Spring Boot u čijoj se osnovi nalazi programski jezik Java.

**Ključne reči:** eye-tracker, Spring Boot, Angular, Java, regioni od interesa, Markovljevi lanci, slučajni procesi, HTML, CSS

**Abstract** – Eye tracking is a sensor, hardware technology which allows computers or other devices to track user viewing point. Eye tracker camera can detect presence, attention and focus of a user. Also, it represents a potential relief for visually impaired users or users unable to use their hands. This type of camera is very expensive, thus being unavailable to most of user population. Potentially, Gazepoint Eye Tracker simulator could solve this problem. Detailed process of implementation of the software, as well as the principle of determination of user viewing point is explained in this paper. Angular framework was used for the implementation of the front end of the application, while Spring Boot, whose core language is Java, was used to handle the bussiness logic.

**Keywords:** eye-tracker, Spring Boot, Angular, Java, regions of interest, Markov chains, random processes, HTML, CSS

**1. UVOD**

Eye tracking je proces merenja kako tačke pogleda, tako i pomeraja oka u odnosu na glavu. Eye tracker je uređaj koji se koristi za merenje pozicije očiju i njihovog pomeraja. Najviše se koriste u istraživanjima vezanim za vizuelne sisteme, psihologiju, psiholingvistiku, marketing, kao ulazni uređaj pri interakciji čoveka sa računarom. Sve više ih koriste aplikacije za rehabilitaciju i pomoć hendikepiranim osobama.

S obzirom na njegovu nepristupačnost, koja proizilazi prvenstveno iz visoke cene, a koja u proseku iznosi oko 17500 dolara, stvara se potreba za izradom softverskog simulatora koji bi bio sposoban da u velikoj meri pokrije funkcionalnosti stvarnog uređaja. Konkretno, ideja za implementaciju ovakvog softvera nastala je prilikom implementacije Web platforme za ocenjivanje znanja uz praćenje pokreta oka [7], kada Gazepoint Eye tracker kamera [8] nije bila dostupna prilikom ulaska u fazu testiranja.

**2. PROGRAMSKI JEZIK JAVA**

Istorija Jave [1] počinje 1991. Godine, inicijalnim projektom pod nazivom “Java language project”. U projektu su učestvovali Džejs Gosling (eng. James Gosling), Majk Šeridan (eng. Mike Sheridan) i Patrik Noton (eng. Patrick Naughton). Ovaj tim inženjera ubrzo je nazvan “Zelenim timom” (eng. Green Team).

Zanimljivo je napomenuti da je Java prvobitno dizajnirana u svrhe interaktivne televizije. Ideja je bila napraviti jezik pogodan za digitalne uređaje kao što su moderni televizori i set-top box uređaji. Kako će se kasnije ispostaviti, sama ideja je bila pre više napredna za koncepte digitalne kablovske televizije devedesetih godina prošlog veka. Java je tako našla svoju primenu u web programiranju, da bi je nedugo zatim kupila firma Netscape.

Programski jezik Java [2] je platformski nezavistan, u skladu sa write once, run anywhere (WORA) filozofijom. Ovo podrazumeva da jednom napisan Java kod na bilo kojoj platformi, odnosno operativnom sistemu, kasnije može biti pokrenut na bilo kom drugom operativnom sistemu.

Java je potpuno objektno orijentisana. Svaki element je klasa, njen deo ili objekat neke klase. Koristi se za razvijanje desktop i mobilnih aplikacija, big data processing, embedded sisteme, web aplikacije. Nasleđivanje, enkapsulacija, polimorfizam i apstrakcija – osnovni koncepti Jave, kao i svakog drugog objektno-

**NAPOMENA:**

**Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinać, vanr. prof.**

orijentisanog jezika, u velikoj meri su olakšali implementaciju ovog simulatora.

### 3. SPRING FRAMEWORK I SPRING BOOT

Spring Boot [3] olakšava implementaciju stand-alone, production grade aplikacija zasnovanih na Spring platformi [4]. Moguće je koristiti Spring Boot za kreiranje Java aplikacija koje potom mogu biti pokrenute standardnim java -jar naredbama ili izvežene (eng. deployed) u war formatu.

Podržava rad sa Maven i Gradle alatima za izgradnju aplikacije. Takođe, omogućava rad sa više servletskih kontejnera, kao što su:

- Tomcat 9.0 (4.0 servletska verzija),
- Jetty 9.4 (3.1 servletska verzija),
- Undertow 2.0 (4.0 servletska verzija).

Sveukupno, Spring Boot značajno olakšava izradu kompleksnih enterprise aplikacije, bile one monolitne ili zasnovane na mikroservisima koristeći samo POJO-e (plain old Java object). Na taj način se izbegava potreba za korišćenjem EJB kontejnera, ali se i dalje ostavlja opcija za korišćenje nekog od gorepomenutih robusnih servletskih kontejnera.

Modularna organizacija omogućava usredsređivanje samo na one kole i pakete na kojima se trenutno radi, nezavisno od njihovog ukupnog broja.

Posebno je pogodan za implementaciju skalabilnih aplikacija. Podržava rad sa XML formatom podataka, kao i slanje tih istih podataka preko SOAP protokola, dok je Spring-ova web platforma dizajnirana po MVC šablonu.

Za razvijanje Gazepoint Eye tracker simulatora je ipak korisnija bila činjenica da je u Spring Boot-u, kao i u Spring-u, podržan rad sa REST servisima, kao i rad sa Web Socket-ima.

Takođe, dostupan je veliki broj API-a za povezivanje sa bazom podataka koji omogućavaju i umnogome olakšavaju code-first pristup projektovanju baze, pravljenje upita nad bazom, kao i upisivanje podataka u istu.

Ne treba izostaviti ni API za security, koji, iako komplikovan za inicijalno podešavanje, kasnije pruža veliku fleksibilnost pri zaštiti korisničkog pristupa REST endpoint-ima.

### 4. ANGULAR

Angular [5] predstavlja okruženje za razvijanje aplikacija na klijentskoj strani, odnosno frontend aplikacija. Baziran je na typescript-u. Prvi put se pojavio kao AngularJS 2010. godine. Prvobitno je razvijen kao biblioteka javascript-a, da bi 2014. počeo razvoj Angulara 2. Ovo je prva verzija razvijena u typescript-u. Za potrebe ovog projekta korišćen je Angular 8.

Najbitnija karakteristika bila bi mogućnost razvijanja single-page aplikacija. To znači da se cela aplikacija učitava na jednoj web stranici. Prilikom učitavanja novih

ili promene podataka nema potrebe za ponovnim učitavanjem cele stranice. Samo onaj deo stranice koji se izmenio se ponovo učitava. Ovakav način rada drastično ubrzava rad, jer se izbegava učitavanje redundantnog sadržaja.

Takođe, bitno je napomenuti da se ceo kod izvršava direktno u pretraživaču. Budući da su pretraživači u suštini javascript kompajleri, prilikom kompajliranja Angular aplikacije celokupan kod se mašinski prevodi iz typescript-a u javascript.

Angular podržava dva režima rada:

- razvojni režim – programiranje i debugging se vrše u typescript-u,
- produkciono režim – kod se trajno prevodi u javascript i postavlja na neki web server.

Bazira se na MVC šablonu. Veoma bitna osobina je i data-binding, koja olakšava rad sa podacima iz modela.

### 5. O APLIKACIJI

Gazepoint Eye tracker simulator je web aplikacija uz pomoć koje je moguće simulirati rad Eye tracker hardvera. Korisnički softver je moguće povezati sa simulatorom uz pomoć Web socket-a, preko porta 4242 (što je ujedno i konekciono port stvarnog uređaja).

Sama aplikacija je razdvojena u dva dela – prezentacioni sloj, koji je razvijen u Angular framework-u i sloj biznis logike, koji je implementiran uz pomoć Spring Boot platforme. Prezentacioni sloj simulatora je prikazan na slici 1.



Slika 1. Prezentacioni sloj simulatora

U gornjem levom uglu se nalazi dugme Connect. Pritiskom na ovo dugme izveštava se prezentacioni sloj o potrebi za povezivanjem sa Web socket-om na portu 4242. Na ovaj način se vrši povezivanje sa slojem biznis logike i omogućava slanje zahteva simulatoru, kao i primanje njegovih odgovora.

U tekstualnom bloku Send request se pišu zahtevi za slanje, u formatu srodnom XML-u, kao što je prikazano na slici 2.



Slika 2. Zahtev za slanje

U trenutnoj implementaciji omogućeni su sledeći zahtevi:

- `ENABLE_SEND_DATA` – get i set, podešavanje uslova za omogućavanje slanja podataka preko socket-a
- `ENABLE_SEND_COUNTER` – get i set, podešavanje uslova za omogućavanje slanja sistemskog brojača – pogodno za praćenje kontinuiteta primljenih odgovora
- `CALIBRATE_START` – get i set, simulira pokretanje kalibracije uređaja
- `CALIBRATE_TIMEOUT` – get i set, vrednost timeout flag-a
- `CALIBRATE_DELAY` – get i set, vrednost zakašnjenja kalibracije
- `SCREEN_SIZE` – get i set, dimenzije ekrana
- `ENABLE_SEND_POG_BEST` – set, podešavanje uslova za omogućavanje slanja koordinata korisničkog pogleda

Nakon poslatog zahteva, u tekstualnom bloku Answer se ispisuje odgovor servera. Tekstualni blok Server logs u realnom vremenu prikazuje serverske logove.

Kako je ovde reč o simulatoru, odnosno softverskoj komponenti, potrebno je zadati određen obrazac posmatranja simuliranog korisnika.

Celokupan ekran je podeljen u matricu regija od interesa, od kojih svaka predstavlja moguću poziciju korisničkog pogleda. Koristeći zadati obrazac, a uz pomoć lanaca Markova, vrši se proračun naredne regije interesa u koju će simulirani korisnik gledati. Naredna regija interesa može biti neka od susednih trenutnoj, ili se pogled može zadržati na istoj poziciji, kao što je prikazano prvom matricom na slici 3.

```

F (Xn+1 = Up) = x
F (Xn+1 = Right) = y
F (Xn+1 = Left) = z
F (Xn+1 = Down) = k
F (Xn+1 = Current) = j

[
  UL UC UR
  L C R
  DL DC DR
]
P = P(1) = [
  UL UC UR L C R DL DC DR
  UL * * * * *
  UC * * * * *
  UR * * * * *
  L * * * * *
  C x+z x+j x+y z j y k+z k+j k+y => (sum) = 3
  R * * * * *
  DL * * * * *
  DC * * * * *
  DR * * * * *
]
3x + 3y + 3z + 3k + 3j = 3 <=> (T1)

```

Slika 3 Delovi koda

Budući da se Markovljevim lancima računa verovatnoća prelaska pogleda sa bilo kog polja na bilo koje buduće, za proračun simulatora nije potrebno sračunavati sve vrednosti u drugoj prikazanoj matrici. Potrebno je pronaći

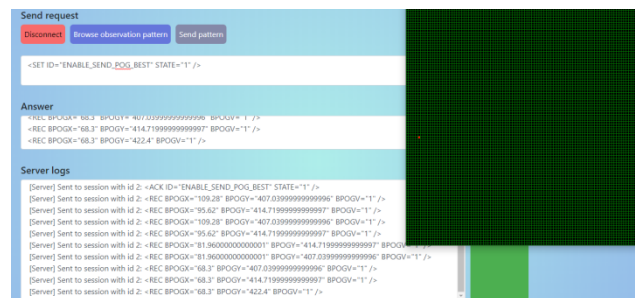
samo vrednosti koje se tiču trenutne gledane regije interesa. To u suštini znači da se računanje verovatnoće prelaska pogleda pojednostavljuje – više nije potrebno određivati celu Markovljevu matricu, već samo jedan njen red (sve vrednosti u ostalim redovima su na slici 3 popunjene zvezdicama).

Ovaj postupak značajno ubrzava proces računanja verovatnoća. Kako je poznata trenutna pozicija korisničkog pogleda, gubi se smisao računanja verovatnoća za sve ostale pozicije.

Takođe, potrebno je sračunati verovatnoće samo za jedan korak u budućnosti, imajući u vidu da simulator ima sposobnost praćenja trenutne situacije.

Pritiskom na dugme Browse observation pattern vrši se odabir obrasca korisničkog pogleda. Obrazac je potrebno zadati u JSON formatu. Dugme Send pattern omogućava slanje odabranog obrasca serveru.

Slanjem zahteva `ENABLE_SEND_POG_BEST` se otvara dodatni prozor koji modeluje matricu regija interesa. Crvenom bojom se u realnom vremenu obeležava trenutno posmatrana regija interesa. Izgled modelovane matrice je prikazan na slici 4.



Slika 4. Modelovana matrica

## 5. ZAKLJUČAK

Problem koji se rešavao u ovom radu tiče se implementacije softverskog simulatora Gazepoint eye tracker kamere. Opisan je način rada simulatora, kao i način odabira sledeće pozicije korisničkog pogleda korišćenjem slučajnih procesa.

Sledeći korak u unapređenju ovog simulatora je dodavanje novih zahteva ka serveru, kao i njihova obrada. Pored toga, moguća je i optimizacija algoritma za Markovljeve lance.

## 6. LITERATURA

- [1] Java programiranje- <https://www.programiz.com/java-programming> (posećeno 12. juna 2020)
- [2] Istorija Jave - <https://www.javatpoint.com/history-of-java> (posećeno 12. juna 2020)
- [3] P. Webb, D. Syer, J. Long, S. Nicoll, R. Winch, A. Wilkinson, M. Overdijk, C. Dupuis, S. Deleuze, M. Simons, V. Pavić, J. Bryant, M. Bhave, E. Melendez, S. Frederick „Spring Boot Reference Documentation“

[4] Spring Framework - <https://spring.io/> (posećeno 12. juna 2020)

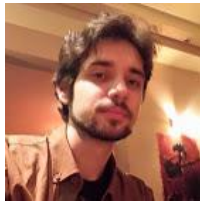
[5] Angular - <https://angular.io/> (posećeno 13. juna 2020.)

[6] T. N. Cornsweet, H. D. Crane “Accurate two-dimensional eye tracker using first and fourth Purkinje images”

[7] N. Gligorov „Web platforma za ocenjivanja znanja uz praćenje pokreta oka“

[8] Gazepoint – <https://www.gazept.com> (posećeno 19. avgusta 2020)

#### **Kratka biografija:**



**Nenad Gligorov** rođen je u Kraljevu 1996. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Primenjene računarske nauke i informatika odbranio je 2020. god.  
kontakt: [ngligorov@uns.ac.rs](mailto:ngligorov@uns.ac.rs)