Univerzitet u Beogradu
Elektrotehnički fakultet

Razvoj vizuelnog softverskog simulatora Enigma mašine

DIPLOMSKI RAD

Kandidat:
Mitić Natalija 2017/0085

Profesor:
prof. dr Stanisavljević Žarko

Beograd
avgust 2021.

Sadržaj

1. UVOD ... 1

2. ENIGMA MAŠINA ... 3

2.1 ISTORIJAT ... 3
2.2 PRINCIP RADA ... 3
2.2.1 Osnovne karakteristike ... 4
2.2.2 Tastatura i lampice ... 5
2.2.3 Rotor ... 5
2.2.4 Reflektor ... 8
2.2.5 Priključna ploča ... 8
2.2.6 Šifrovanje i dešifrovanje ... 10
2.3 VERZIONISANJE ... 11
2.3.1 Enigma I ... 11
2.4 ANALIZA SIGURNOSTI ... 12
2.5 RAZBIJANJE ENIGME ... 12
2.5.1 Uparivanje originalne i šifrovane poruke ... 13

3. IMPLEMENTACIJA *SEAL* SIMULATORA ... 15

3.1 KORIŠĆENE TEHNOLOGIJE ... 15
3.2 FUNKCIONALNI ZAHTEVI ... 16
3.3 DIZAJN APLIKACIJE ... 16
3.4 DIZAJN VIZUELIZACIJE ALGORITMA ... 16
3.5 PREGLED PAKETA ... 17
3.5.1 Paket rs.ac.bg.etf.mn170085d ... 18
3.5.2 Paket enigma ... 18
3.5.3 Paket gui ... 20

4. NAČIN KORIŠĆENJA *SEAL* SIMULATORA ... 23

4.1 KORISNIČKI MENI ... 24
4.1.1 File ... 24
4.1.2 Mode ... 24
4.1.3 Settings ... 25
4.1.4 Reset Rotors ... 25
4.2 REŽIMI RADA ... 25
4.2.1 Textbox ... 26
4.2.2 Simulation ... 28
4.2.3 Keyboard ... 29
4.3 PODEŠAVANJA MAŠINE ... 31

5. ZAKLJUČAK ... 33

LITERATURA ... 35

SPISAK SLIKA ... 37

SPISAK TABELA ... 38

1. UVOD

U današnjem svetu, isprepletanom mrežnim kablovima, elektronska pošta, beskontaktna plaćanja, kriptovalute uveliko predstavljaju svakodnevicu. Međutim, ništa od navedenog ne bi bilo zamislivo bez enkripcije.

Enkripcija poseduje istoriju dugu hiljadama godina i u svom osnovnom obliku predstavlja proces (algoritam) pretvaranja originalne poruke u šifrovanu, time obezbeđujući tajnost informacija [1]. Njen značaj oduvek je bio prepoznat od strane vojske, pa su samim tim osvajanja i ratovi bili najveća motivacija za razvoj jačih algoritama šifrovanja. Takođe, razvoj enkripcije usko je vezan uz razvoj matematike, ali i statistike, lingvistike, pa i elektrotehnike i računarstva. Samim tim, zbog naglog razvoja tehnike, ali i ratnog stanja, do ogromne revolucije dolazi nakon Prvog svetskog rata. Shvativši ograničene kapacitete ručnog šifrovanja i dešifrovanja rađa se ideja za pravljenjem sistema, tačnije mašine, u tu svrhu. Počinje razvoj elektromehaničke, rotor mašine zadužene za šifrovanje i dešifrovanje poruka – Enigma mašine [2].

Tokom Drugog svetskog rata, primarni kriptosistem svih nemačkih vojnih jedinica bila je upravo Enigma. Mašina je omogućavala brzo šifrovanje, a u isto vreme posedovala ogroman broj kombinacija podešavanja, što je izuzetno otežavalo analizu i razbijanje algoritma enkripcije. Iako se smatralo da je Enigma neprobojna, zbog svojih nedostataka u dizajnu, ali i u samom korišćenju, pri kraju rata algoritam je uspešno razbijen. Pored svoje istorijske važnosti, algoritam Enigma mašine predstavljao je odskočnu dasku za dalji razvoj modernih, današnjih digitalnih, sistema šifrovanja. Samim tim, razumevanje ovog algoritma, kao i analiza njegove kriptografske složenosti, ali i slabosti i nedostataka, od ključne je važnosti.

Radi bržeg i jasnijeg razumevanja algoritma prirodno se nameće vizuelizacija istog, na primer putem simulatora. Takav simulator mora biti jednostavan za korišćenje, intuitivan i pre svega verodostojan, odnosno kompatibilan sa Enigma mašinom. Sistem treba jasno da oslikava rad algoritma, uz mogućnost šifrovanja i dešifrovanja teksta. U tu svrhu razvijen je *SEAL*[1] simulator, koji će biti tema ovog rada. Simulator je namenjen prvenstveno korisnicima koji žele da se upoznaju sa radom algoritma Enigma mašine i kao takav pogodan je za studente koji pohađaju neki kurs kriptografije ili zaštite podataka (npr. kurs *Zaštite podataka* na Elektrotehničkom fakultetu u Beogradu).

[1] Naziv simulatora predstavlja akronim za ***S**imple **E**nimga M**a**chine Simu**l**ator* što se sa engleskog može prevesti kao *jednostavan simulator Enigma mašine*

Nastanak Enigma mašine, njen istorijski značaj, kao i dalji razvoj i verzionisanje mašina predstavlja se u drugoj glavi ovog rada. Takođe, detaljno se opisuje način funkcionisanja pojedinačnih elemenata mašine uz odgovarajuće primere i dijagrame. Na kraju glave daje se analiza sigurnosti algoritma i ukazuje se na mane i slabosti istog.

Treća glava predstavlja tehničku specifikaciju simulatora u kojoj se napominju korišćene tehnologije prilikom razvoja aplikacije. Ujedno, prikazuju se funkcionalni zahtevi sistema i opisuje njegov celokupni implementacioni dizajn. Radi boljeg razumevanja arhitekture aplikacije koriste se dijagrami klasa i paketa.

U četvroj glavi daje se nedvosmisleno korisničko uputsvo za aplikaciju. Detaljno se predstavljaju različiti režimi rada koje simulator podržava, kao i način na koji se aplikacija koristi. Takođe, ovde se napominju sva ograničenja simulatora.

U petoj, ujedno i poslednjoj glavi daje se zaključak, kao i rezime svega urađenog. Na samom kraju, iznose se uočeni nedostaci i potencijalna poboljšanja sistema.

2. ENIGMA MAŠINA

Na početku ove glave biće ukratko predstavljen istorijat Enigma mašine. Potom, temeljno će biti opisane karakteristike mašine, kao i princip rada pojedinačnih delova iste. Na samom kraju, biće data analiza sigurnosti sa osvrtom na mane algoritma Enigma mašine.

2.1 Istorijat

Krajem Prvog svetskog rata, nemački inženjer elektrotehnike Artur Šerbijus, prepoznavši značaj sigurne razmene informacija, započinje razvoj Enigma mašine s namerom da se ista proizvodi, usavršava i koristi za šifrovanje, odnosno očuvanje tajnosti, poruka. [3]

Kako je nemačka vojska odbila prvu verziju Enigme, Artur se posvetio pravljenju mašine u komercijalne svrhe. Krajem dvadesetih godina XX veka pojavljuju se prvi primerci pomenute mašine pod nazivom Model A [4]. U daljem periodu, kroz novije modele mašina biva konstantno unapređivana, što za posledicu ima probuđivanje zainteresovanosti kod nemačke vojske. Nemačka armija (nem. *Deutsches Heer*) kao i nemačka mornarica (nem. *Kriegsmarine*) su preuzele kasnije modele Enigme uz dodatne izmene i dorade, prilagođavajući ih svojim potrebama. Ove modifikacije su astronomski uvećavale broj mogućih podešavanja mašine i time otežavale razbijanje iste.

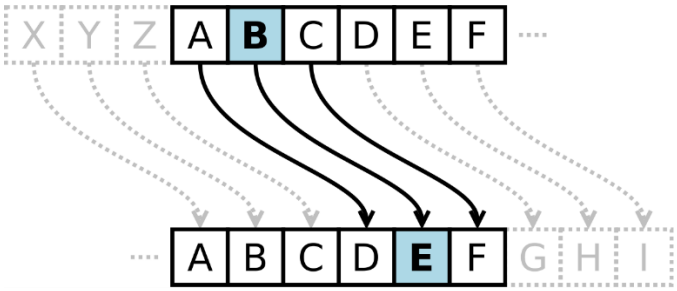
Tokom Drugog svetskog rata Enigma mašina je predstavljala standardni vid zaštite komunikacije među nemačkim vojnim jedinicama, ujedno pružajući jedan od najjačih kriptografskih šifarnika u to vreme. Međutim, kao i sve što čovek napravi, ni Enigma nije bila savršena, te je imala svoje nedostatke, time omogućivši suprotnoj strani (Saveznicima) da razbije ovu šifru.

2.2 Princip rada

Mehanizam rada Enigma mašine zasnovan je na prostom principu zamene (engl. *substitution*), što podrazumeva preslikavanje slova azbuke u neko drugo slovo iste azbuke. Preciznije, Enigma predstavlja polialfabetsku šifru zamene pošto koristi nekoliko različitih šifara zamene.

Klasičan predstavnik supstitucionog šifrovanja jeste Cezarova šifra [5]. Cezarova šifra ulazno slovo mapira u za k mesta udaljeno slovo u azbuci, pri čemu broj k predstavlja ključ šifrovanja.

Na slici 1 može se videti Cezarova šifra za ključ $k = 3$.



Slika 1 - Cezarova šifra

Najveća mana ovakvog algoritma jeste da algoritam čuva frekvenciju simbola. Naime, ukoliko se u nekom tekstu slovo A javlja *p* puta, a isto to slovo šifruje slovom R, tada će šifrovani tekst sadržati slovo R takođe *p* puta. Analizom dovoljno dugačkog šifrovanog teksta, uz poznavanje jezika na kom je pisana originalna poruka, kao i učestanosti pojavljivanja odgovarajućih simbola, lako se može izvršiti napad i dešifrovati poruka.

Iako je Enigma zasnovana na jednostavnom principu zamene, zbog svojih karakteristika, mnogobrojnih konfiguracija, kao i specifičnog načina rada pružala je svojevremeno izuzetno jak mehanizam zaštite.

2.2.1 Osnovne karakteristike

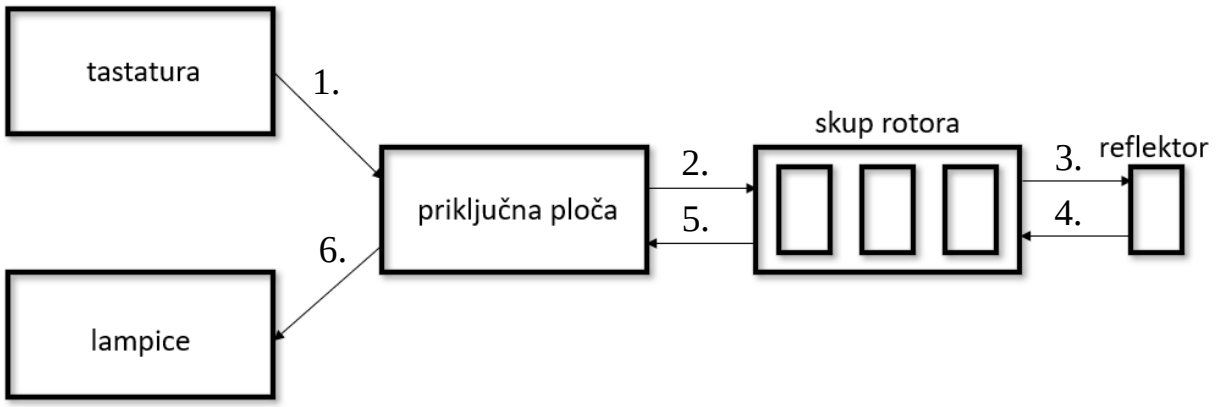
Enigma je mašina koja po svom izgledu podseća na pisaće mašine, spada u rotor mašine i sastoji se od mehaničkog i električnog dela. [6]

Mehanički deo mašine obuhvata:

- tastaturu od 26 slova
- priključnu ploču
- rotirajuće valjke (rotore)
- reflektor (posebna vrsta rotora)
- skup od 26 lampica – svaka lampica predstavlja jedno slovo

Takođe, postojali su dodatni elementi koji su mogli da se koriste, jedan od tih dodataka bio je *Schreibmax* [7] – mali štampač koji bi se ugradio umesto lampica omogućavajući štampanje šifrovane poruke na papir.

Mašinom se upravlja mehanički, pri čemu električni signal prolazi kroz žice i različite mehaničke delove. Jedan ciklus šifrovanja slova od ulaza (dugme tastature) do izlaza (lampica) mogao bi se predstaviti dijagramom na slici 2.



Slika 2 - Ciklus šifrovanja na Enigma mašini

Pritiskom određenog slova na tastaturi generiše se električni signal koji otpočinje svoj put kroz mašinu prolazeći redom kroz priključnu ploču, rotore i reflektor. Zatim, vraćajući se kroz rotore i priključnu ploču, električni signal stiže do lampica obasjavajući šifrovano slovo.

2.2.2 Tastatura i lampice

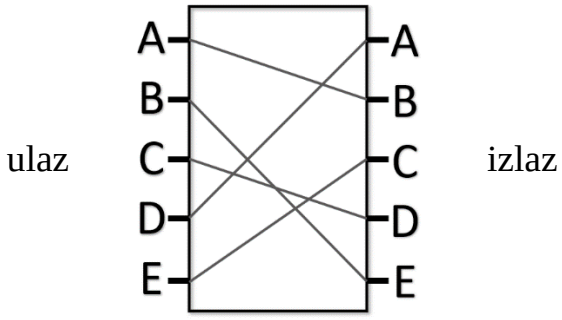
Tastatura se sastoji od 26 dugmića preko kojih se dostavlja ulaz poruke za šifrovanje. Svako dugme predstavlja jedno slovo alfabeta, pri čemu se velika i mala slova ne razlikuju.

Lampica takođe ima 26 i poput dugmića tastature, svaka lampica odgovara jednom slovu alfabeta. Izlaz šifrovanja predstavlja odgovarajuća lampica (slovo) koja bi zasvetlela.

2.2.3 Rotor

Rotor je primarni deo Enigme i kao takav je odgovoran za šifrovanje pojedinačnog slova – konkretno, preslikavanje jednog slova u drugo. Sastoji se od dva prstena, unutrašnjeg (ulaznog) i spoljašnjeg (izlaznog), koji sadrže po 26 priključaka (pinova) koji su međusobno povezani žicama kroz koje prolazi električni signal, time formirajući mapu šifrovanja slova (tkzv. ožičenja). [8]

Primer ovakve mape, odnosno rotora, može se videti na slici 3. Radi pojednostavljenog prikaza za azbuku je uzet skup {A, B, C, D, E}, a rotor odgovara preslikavanju ABCDE –> BEDAC.



Slika 3 - Pojednostavljen rotor

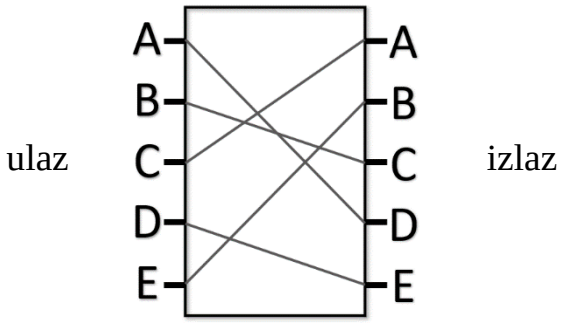
Postoji nekoliko vrsta rotora, pri čemu je svaka vrsta imenovana rimskim brojem (I, II, itd.). Različite vrste rotora pružaju različita unutrašnja preslikavanja. Broj različitih vrsta rotora, kao i ukupan broj rotora koji se koristi u mašini tokom šifrovanja zavisi od verzije mašine na kojoj se poruka kriptuje.

Treba naglasiti da rotor sam po sebi predstavlja jednostavan vid šifrovanja metodom supstitucije. Međutim, kriptografska jačina mašine leži u korišćenju nekoliko (uglavnom tri ili četiri) serijski vezanih rotora – konektori spoljnjeg (izlaznog) prstena rotora naležu na konektore unutrašnjeg (ulaznog) prstena sledećeg rotora u nizu. Samim tim, šema preslikavanja se znatno komplikuje.

2.2.3.1 Podešavanje prstena

Rotor je moguće postaviti u mašinu u jednu od 26 mogućih startnih pozicija. Ove pozicije zapravo predstavljaju (cirkularno) rotiranje preslikavanja naniže u odnosu na podrazumevanu poziciju.

Prethodno prikazan, pojednostavljen rotor se podrazumevano nalazi u prvoj, odnosno A, poziciji. Smeštanjem uprošćenog rotora u drugu, odnosno B, poziciju uočava se sledeća konfiguracija ABCDE –> DCAEB.

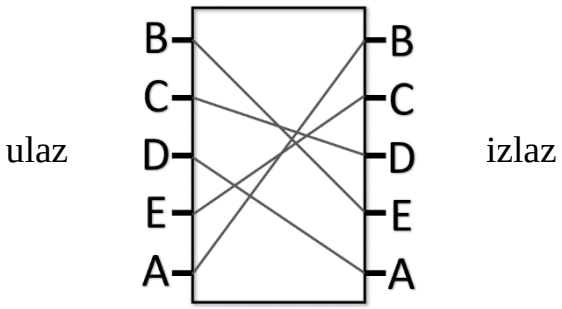


Slika 4 - Pojednostavljeno podešavanje prstena

2.2.3.2 Navijanje rotora

Dodatno, nakon ubacivanja rotora u mašinu, moguće je njegovo navijanje, odnosno ručno okretanje, na bilo koje slovo alfabeta. Navijanje predstavlja (cirkularno) rotiranje priključaka (sa sve mapiranjem) naviše. Podrazumevano, rotor je navijen na poziciju 1, odnosno A.

Prvobitno prikazan, pojednostavljen rotor u slučaju navijanja na poziciju 2, odnosno B, rezultuje u preslikavanje BCDEA –> EDACB. Treba primetiti da su mapiranja ulaz-izlaz zapravo ostali neizmenjeni, međutim u slučaju korišćenja vezanih rotora dolazi do promene kada se sa jednog rotora prelazi na drugi – upravo zbog izmenjenog redosleda ulaza/izlaza.



Slika 5 - Pojednostavljeno navijanje rotora

2.2.3.3 Rotacija

Za rotaciju rotora su odgovorni karakteristični zarezi koji se mogu naći na spoljnom prstenu rotora, te oni i diktiraju trenutak rotacije. Takođe, bitno je napomenuti da se rotacija dešava neposredno pre započinjanja šifrovanja pritisnutog slova.

Prvi rotor, onaj koji se nalazi neposredno nakon tastature, rotira prilikom svakog pritiska tastera. Svaki sledeći rotor rotira kada njegov prethodnik izrotira ceo ciklus, iliti kada dospe do zareza. Većina vrsta rotora poseduje samo jedan zarez stoga se rotacija javlja sa periodom dužine korišćene azbuke.

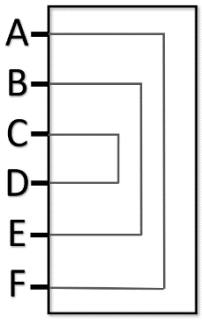
Primera radi, neka prvi rotor ima zarez na poziciji slova Y i neka se slovo Q mapira na slovo X. Pod pretpostavkom da je trenutna pozicija rotora na poziciji Q, tokom narednog prelaza, sa Q na R, nailazimo na zarez X na Y i tom prilikom dolazi do rotacije rotora sledbenika. Analogno se izvode rotacije i narednih rotora u nizu.

U cilju izbegavanja očuvanja frekvencije ulaznih simbola prilikom svakog pritisnutog tastera dolazi do rotacije najmanje jednog rotora za jednu poziciju/slovo. Drugim rečima, u potpunosti je moguće da se za ulaznu sekvencu AAAAAA dobije šifrovani tekst QYHJMZ. Treba primetiti da se identični karakteri nesmetano šifruju u različite karaktere što samim tim dodatno otežava razbijanje šifre.

2.2.4 Reflektor

Reflektor predstavlja posebnu vrstu rotora koja pruža upareno preslikavanje. Takođe, i za reflektor važi da postoje različite vrste sa različitim, unutrašnjim povezivanjima. Međutim, za razliku od rotora koji se imenuju rimskim brojevima, različite vrste reflektora se imenuju slovima abecede.

Na narednoj slici može se videti primer jednog uprošćenog reflektora za azbuku {A, B, C, D, E, F} koji odgovara preslikavanju ABCDEF –> FEDCBA. Treba uočiti da pošto važi A –> F, onda direktno važi i F –> A – reflektor obezbeđuje simetričnost.



Slika 6 - Pojednostavljen reflektor

U zavisnosti od verzije mašine koja se koristi, reflektor je moguće postaviti u neku startnu poziciju (navijanje reflektora) i može rotirati tokom šifrovanja po istom principu kao što to čine rotori.

2.2.5 Priključna ploča

Priključna ploča se sastoji od 26 ulaza, pri čemu svaki ulaz predstavlja jedno slovo alfabeta. Spajanjem dva ulaza kablom dobija se efekat zamene pre nego što električni signal dospe do rotora (Slika 2 tačka 2.), kao i u povratku kada napusti rotore (Slika 2 tačka 5.).

Primera radi, ukoliko su spojeni ulazi slova A i B, pritiskom slova A, električni signal se preusmerava na slovo B neposredno pre ulaska u prvi rotor, identično preusmeravanje se izvršava i tokom povratne putanje nakon izlaska iz rotora. Budući da priključna ploča poseduje svojstvo simetričnosti, ujedno važi preusmeravanje A –> B, kao i B –> A.

Treba napomenuti da spajanje ulaza nije obavezno već opciono i da u slučaju nespojenog ulaza konkretno slovo neće imati zamenu – odnosno, predstavljaće samo sebe.

2.2.5.1 Kombinatorna analiza priključne ploče

Kao što je već pomenuto, postoji 26 ulaza, samim tim postoji i najviše 13 kablova kojima se ovi ulazi mogu spojiti. Posledično tome, ukupan broj kombinacija (*N*) zavisi od broja spojenih parova ulaza, iliti iskorišćenih kablova (*n*).

Za nekih *n* kablova potrebno je izabrati $2 * n$ ulaza, što se može učiniti na $C_{2n}^{26} = \binom{26}{2n}$ načina.

Postavlja se pitanje na koliko različitih načina se preko tih *n* kablova mogu povezati izabrani ulazi. Za prvi kabl potrebno je izabrati 2 ulaza od $2 * n$, što se može učiniti na $\binom{2n}{2}$ načina. Za naredni kabl, biraju se 2 ulaza od preostalih $2 * n - 2$ ulaza, što se može učiniti na $\binom{2n-2}{2}$ načina. Za *n*-ti kabl, biraju se 2 ulaza od preostalih $2 * n - 2 * (n - 1) = 2$. Dobija se:

$$\binom{2n}{2} * \binom{2n-2}{2} * \dots * \binom{2}{2} = \frac{(2n)!}{2! * (2n-2)!} * \frac{(2n-2)!}{2! * (2n-4)!} * \dots * 1 = \frac{(2n)!}{2^n}$$

Međutim, važno je istaći da se kod Enigma mašine ne razlikuju kablovi, niti redosled biranja parova ulaza. Drugim rečima, kombinacija udruženih parova A-B, C-D je istovetna kombinaciji udruženih parova C-D, A-B. Stoga, potrebno je dobijeni broj kombinacija prilagoditi na sledeći način:

$$\frac{(2n)!}{2^n} * \frac{1}{n!}$$

Kombinovanjem ovog novodobijenog broja i broja načina biranja $2 * n$ ulaza dobija se ukupan broj kombinacija *N* u zavisnosti od broja parova *n*:

$$N = \binom{26}{2n} * \frac{(2n)!}{2^n} = \frac{26!}{(2n)! * (26-2n)!} * \frac{(2n)!}{2^n * n!}$$

$$N = \frac{26!}{n! * (26-2n)! * 2^n}$$

Tabela 1 pokazuje broj kombinacija *N* u zavisnosti od konkretnog broja povezanih parova *n*:

Broj parova *n*	**Broj kombinacija *N***
0	1
1	325
2	44,850
3	3,453,450
4	164,038,875
5	5,019,589,575
6	100,391,791,500

7	1,305,093,289,500
8	10,767,019,638,375
9	53,835,098,191,875
10	150,738,274,937,250
11	205,552,193,096,250
12	102,776,096,548,125
13	7,905,853,580,625
Total	532,985,208,200,576

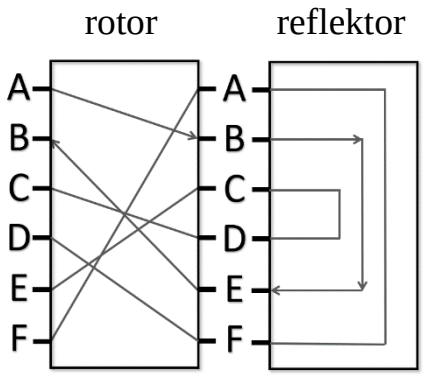
Tabela 1 - Broj dodatih kombinacija preko priključne ploče

Priključna ploča postala je neizostavni deo vojnih Enigma mašina zbog nesumnjivog povećanja kriptografske jačine uvođenjem ogromnog broja kombinacija. Uprkos mogućnosti promenljivog broja parova mašine su uglavnom koristile $n = 10$ povezanih parova, što je ujedno pružalo skoro najveći broj kombinacija.

2.2.6 Šifrovanje i dešifrovanje

Korišćenje reflektora, iliti uparenog preslikavanja, čini Enigmu samo-recipročnom. Stoga, ukoliko bi se na proizvoljnoj Enigma mašini proizvoljne startne konfiguracije X šifrovala poruka, takva šifrovana poruka može se lako, na isti način, dešifrovati bilo kojom identičnom (ili kompatibilnom) mašinom uz prethodno podešavanje mašine na konfiguraciju X. Nije potreban nikakav dodatan algoritam za dešiforvanje.

Slika 7 ilustruje šifrovanje slova A u slovo B pomoću jednog rotora i jednog reflektora. Putanja koja se dobije jeste A –> B –> E –> B.



Slika 7 – Pojednostavljen primer šifrovanja

Može se primetiti da se ista putanja dobija nevezano od toga da li je pritisnuto slovo A ili slovo B – to i jeste osobina samo-recipročnosti. Jedina razlika je što se u slučaju dešifrovanja putanja prati u suprotnom smeru. Slovo B se dešifruje po putanji B –> E –> B –> A.

2.3 Verzionisanje

Tokom istorije postojalo je nekoliko verzija Enigma mašine [9], kako komercijalnih tako i vojnih. One bi se uglavnom razlikovale po broju i vrsti korišćenih valjaka, kao i po tome da li je određeni element (npr. priključna ploča) ugrađen u mašinu.

Prvi, ujedno i komercijalni, modeli mašine, Model A i Model B, upotrebljavali su dva rotora i reflektor, dok je Model C uveo treći rotor. Nijedan od ovih početnih modela nije imao ugrađenu priključnu ploču. [8]

2.3.1 Enigma I

Prva Enigma korišćena od strane nemačke vojske (nem. *Wehrmacht*) bila je Enigma I. Ova mašina se sastojala od:

- pet rotora (tipovi I-V), od kojih su se tri nalazila u mašini
- tri reflektora (tipovi A-C), od kojih je jedan bio u mašini
- priključne ploče

Karakteristično za Enigmu I jeste da reflektor ima statičku konfiguraciju, iliti ne rotira i ne može se smeštati u različite startne pozicije.

Detaljan prikaz karakteristika Enigma I mašine dat je u tabeli 2. U prvoj koloni, rimskim brojevima označene su vrste rotora, dok su slovima abecede označene vrste reflektora. Polja druge kolone opisuju konkretna mapiranja (izlaze) za abecedu (ulaz).

Valjak	ABCDEFGHIJKLMNOPQRSTUVWXYZ	Zarez	Trenutno slovo
I	EKMFLGDQVZNTOWYHXUSPAIBRCJ	Y	Q
II	AJDKSIRUXBLHWTMCQGZNPYFVOE	M	E
III	BDFHJLCPRTXVZNYEIWGAKMUSQO	D	V
IV	ESOVPZJAYQUIRHXLNFTGKDCMWB	R	J
V	VZBRGITYUPSDNHLXAWMJQOFECK	H	Z
A	EJMZALYXVBWFCRQUONTSPIKHGD		
B	YRUHQSLDPXNGOKMIEBFZCWVJAT		
C	FVPJIAOYEDRZXWGCTKUQSBNMHL		

Tabela 2 - Karakteristike mašine Enigma I

Nakon Engime I, nemačka mornarica razvila je nove mašine pod nazivom M1, M2, M3. Nove mašine su preuzele karakteristike Enigme I, pa su shodno tome sa njom i kompatibilne. Povrh toga uvedeni su novi rotori – preciznije, mašine su imale na raspolaganju šest, sedam i osam rotora, respektivno, od kojih su tri bila u upotrebi tokom šifrovanja.

Po uzoru na Enigmu I razvile su se i mnoge druge Enigma mašine, čineći Enigmu I osnovom Enigma mašina nemačke vojske. Upravo u tome leži značaj ovog konkretnog modela i stoga će se dalji rad, kao i sam simulator, ticati isključivo te verzije mašine.

2.4 Analiza sigurnosti

Kao što je prethodno napomenuto, Enigma I poseduje pet različitih rotora, pri čemu se tri rotora koriste u datom trenutku.

$$P_{aranzmanRotora} = 5 * 4 * 3 = 60$$

Svaki od rotora moguće je naviti, kao i podesiti prsten, na 26 različitih pozicija.

$$P_{konfiguracijaRotora} = 26^3 * 26^3 = 308{,}915{,}776$$

Enigma I poseduje tri različita reflektora, pri čemu se samo jedan koristi u datom trenutku.

$$P_{aranzmanReflektora} = 3$$

U slučaju korišćenja priključne ploče sa 10 parova, dobija se:

$$P_{prikljucnaPloca} = 150{,}738{,}274{,}937{,}250$$

Kombinovanjem svih izračunatih kombinacija, dobija se ukupan broj mogućih inicijalnih konfiguracija mašine:

$$P = P_{aranzmanRotora} * P_{konfiguracijaRotora} * P_{aranzmanReflektora} * P_{prikljucnaPloca}$$

$$P \approx 8{,}3 * 10^{24}$$

2.5 Razbijanje Enigme

Kako se Enigma u jednom periodu prodavala komercijalno, proces analize rada i mana mašine od strane drugih država započet je znatno pre Drugog svetskog rata. Tom prilikom, autor ističe uložene napore poljskog matematičara Marjana Rejevskog [10] koji je među prvima pokušao da odgonetne način rada mašine. Primenjujući matematičke metode u kriptoanalizi tokom svoje analize mnogobrojnih presretnutih poruka uspeo je da reši problem otkrivanja ožičenja rotora, odnosno da rekonstruiše unutrašnja povezivanja rotora. Međutim, konstantna unapređivanja mašine doprinosila su vrtoglavom porastu broja kombinacija šifrovanja i time otežavala razbijanje šifre, kao i delimično invalidirala prethodne analize.

Uprkos enormnom broju mogućih kombinacija, Enigma mašina poseduje par karakterističnih slabosti:

- simetrično šifrovanje – Zbog samo-recipročnosti ukoliko se slovo A šifruje u X, tada se i slovo X šifruje u A.

- Nijedno slovo alfabeta ne može se šifrovati samim sobom. Ovakav nedostatak uzrokuje reflektor koji uvek uparuje različita slova alfabeta. Samim tim, električni putevi koji vode do i od reflektora, stoga i ulazno i izlazno slovo, se uvek razlikuju.

Glavni način razbijanja šifre zasnivao se na principu odbacivanja brojnih nevalidnih podešavanja mašine pozivajući se na prethodno navedene slabosti. Upravo na taj način je Enigma mašina uspešno razbijena tokom Drugog svetskog rata. [11]

2.5.1 Uparivanje originalne i šifrovane poruke

Pored nedostataka u samom algoritmu, razbijanju Enigme doprineo je i stil komunikacije nemačke vojske. Preciznije, Nemci su često prenosili ustaljene izraze poput *ništa za izvestiti* (nem. *keine besonderen Ereignisse*). Samim tim, bilo je moguće naslutiti barem neku reč originalnog teksta (engl. *plaintext*) i izvršiti napad na šifru (engl. *known-plaintext attack*). Za presretnutu poruku za koju se sumnja da sadrži konkretnu frazu potrebno je utvrditi na kojoj poziciji se ta fraza nalazi. [12]

Primera radi, ukoliko je presretnuta poruka

UAENFVRLBZPWMEPMIHFSRJXFMJKWRAXQEZ

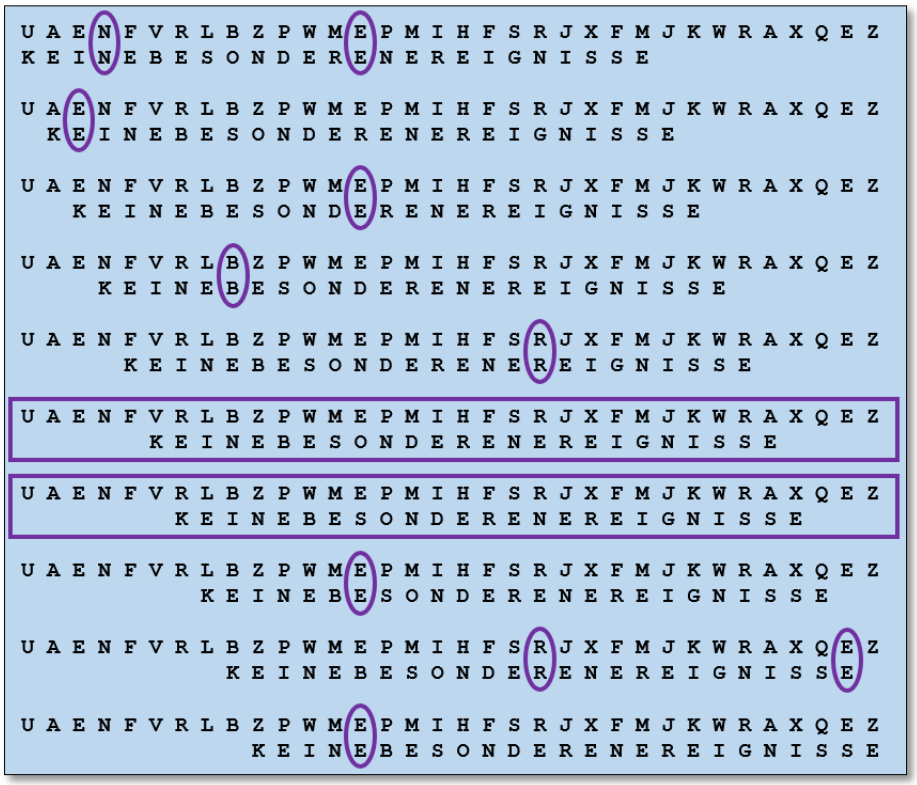
za frazu

KEINEBESONDERENEREIGNISSE

postoji deset mogućih pozicija – od kojih osam nije validno (Slika 8 - Moguće pozicije poznate fraze) i može se odbaciti – validne pozicije markirane su pravougaonikom, dok su elipsom markirana nevalidna šifrovanja.

Gorepomenuti pristup razbijanju šifre često se koristio kao prvi korak, upravo zbog jednostavnog načina eliminacije nevalidnih stanja i podešavanja mašine.

Treba primetiti da za razbijanje Enigme, ali i bilo kog drugog sistema, podjednako utiču mane u dizajnu koliko i sam način korišćenja kriptografskog sistema. Drugim rečima, kroz (neadekvatnu) upotrebu sistema povećava se prostor za eksploatisanje slabosti algoritma.



```
U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
K E I N E B E S O N D E R E N E R E I G N I S S E

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
  K E I N E B E S O N D E R E N E R E I G N I S S E

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
    K E I N E B E S O N D E R E N E R E I G N I S S E

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
      K E I N E B E S O N D E R E N E R E I G N I S S E

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
        K E I N E B E S O N D E R E N E R E I G N I S S E

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
          K E I N E B E S O N D E R E N E R E I G N I S S E

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
            K E I N E B E S O N D E R E N E R E I G N I S S E

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
              K E I N E B E S O N D E R E N E R E I G N I S S E

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
                K E I N E B E S O N D E R E N E R E I G N I S S E

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
                  K E I N E B E S O N D E R E N E R E I G N I S S E
```

Slika 8 - Moguće pozicije poznate fraze

3. IMPLEMENTACIJA *SEAL* SIMULATORA

U ovoj glavi biće predstavljeni implementacioni zahtevi simulatora sa osvrtom na pristup i način rešavanja glavnih problema. Takođe, biće opisane i korišćene tehnologije prilikom implementacije, kao i sama struktura simulatora.

3.1 Korišćene tehnologije

Za razvoj simulatora korišćen je programski jezik *Java* [13], preciznije *SDK 16* [14], čineći simulator desktop aplikacijom. Grafičko-korisnički interfejs (engl. *GUI*) realizovan je pomoću *JavaFX* [15] softverske platforme pri čemu je korišćena verzija 16, kao i pomoću alata *Scene Builder* [16]. Alat *Maven* [17] korišćen je za automatizaciju sastavljanja projekta, pošto je u novijim verzijama *SDK*-a paket *JavaFX* izbačen i potrebno ga je eksplicitno uvezati. Tom prilikom podešeno je i uvezivanje odgovarajućih *JavaFX* grafičkih zavisnosti za podršku kako *Windows* tako i *Linux* i mac*OS* operativnih sistema. Takođe, realizovano je pravljenje izvršnog *JAR* fajla, u koji se pakuju sve potrebne zavisnosti. Samim tim, simulator se praktično može pokretati na skoro svim operativnim sistemima uz preduslov da je odgovarajuća *Java* instalirana.

Kako je glavna primena simulatora zapravo vizuelizacija, ona se mogla realizovati i pomoću *web* tehnologija (*HTML, CSS, JS*). Međutim, zbog nedostatka potrebe da korisnik bude na mreži i u komunikaciji sa serverom, kao i zbog jednostavnosti kreiranja *GUI*-a putem *drag & drop*[2] opcije *JavaFX Scene Builder*-a, prednost je imala upravo *JavaFX* platforma.

Kao jedan od konkurenata *JavaFX* platforme izdvaja se i njen stariji brat – alat *Swing* [18]. Uzimajući u obzir konzistentnost sa *MVC* arhitekturom[3], moderan dizajn, kao i podršku za stilizovanje i animacije, *JavaFX* platforma je za ove potrebe neminovno pogodnija u odnosu na *Swing*.

[2] prevlačenje i ispuštanje – gest pokazivačkog uređaja kojim korisnik bira virtuelni objekat tako što ga *ugrabi* i prevuče na drugu lokaciju

[3] razdvajanje aplikacionog softvera na komponente: model (*Model*), prikaz (*View*) i kontroler (*Controller*)

3.2 Funkcionalni zahtevi

Prilikom izrade simulatora, nameće se nekoliko nužnih funkcionalnosti koje je potrebno realizovati:

- (de)šifrovanje teksta
- (de)šifrovanje pojedinačnog karaktera uz vizuelizaciju rada algoritma
- podešavanje mašine (pojedinačnih rotora, reflektora, priključne ploče)

Pored ovih osnovnih funkcionalnosti, simulator je dopunjen i sledećim dodatnim funkcionalnostima, koje za cilj imaju olakšavanje upotrebe simulatora:

- uvoz i izvoz teksta koji treba (de)šifrovati
- resetovanje rotora na početnu konfiguraciju
- režim rada koji simulira korišćenje Enigme – korisnik stiče utisak da kuca po pravoj mašini

3.3 Dizajn aplikacije

Radi ispunjavanja prethodno navedenih zahteva aplikacije prirodno se nameću tri različita režima rada, dok je sam izgled aplikacije podeljen u nekoliko prozora:

- *Textbox* (tkzv. standardni) režim rada – omogućava (de)šifrovanje teksta, kao i uvoz ulaznog i izvoz izlaznog teksta
- *Simulation* režim rada – omogućava (de)šifrovanje pojedinačnih karaktera uz istovremenu vizuelizaciju rada algoritma
- *Keyboard* režim rada – omogućava (de)šifrovanje pojedniačnih karaktera uz pružanje vizuelnog osećaja kucanja po Enigma mašini
- *Settings* prozor – omogućava podešavanje svih delova mašine (priključna ploča, rotori, reflektor)

Za detaljnije informacije o pojedinačnim prozorima, kao i o njihovom korišćenju, pogledati glavu 4.

3.4 Dizajn vizuelizacije algoritma

Kako bi se predstavio način rada algoritma potrebno je prikazati:

- rad pojedinačnih delova mašine,
    - Svaki deo mašine predstavljen je kao pravougaonik sa ulaznom i izlaznom azbukom, pri čemu postoje linije (ožičenja) koje povezuju pomenute azbuke. Na ovaj način, brzo i lako se mogu uočiti unutrašnja mapiranja svakog elementa.

Dosledno tome, ovakva predstava elemenata mašine korišćena je i u primerima u poglavlju 2.2.

- Prilikom kriptovanja karaktera potrebno je drugačijom bojom istaći konkretna mapiranja korišćena tokom šifrovanja u odnosu na ostala unutrašnja povezivanja. Dodatno, na slici 2 može se primetiti da se putanja šifrovanja prelama u reflektoru. Stoga, ona se može podeliti na ulazni (putanja do reflektora) i izlazni (putanja od reflektora) deo. Ova dva dela putanje predstavljena su različitim bojama radi boljeg uočavanja redosleda prolaska kroz elemente mašine. Takođe, dodatnim simbolima <<< i >>> označen je početak (originalno slovo) i kraj (šifrovano slovo) šifrovanja, respektivno.

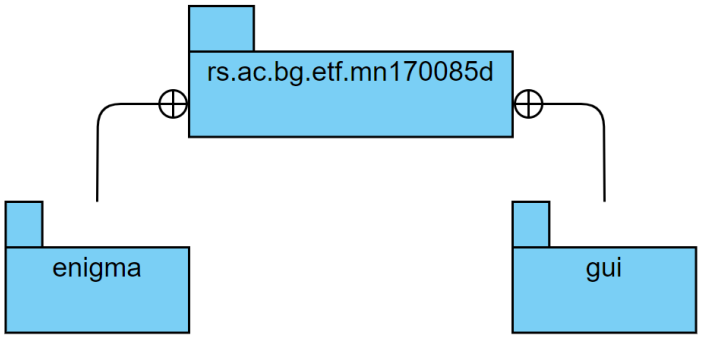
- aktuelno stanje rotora,
  - Radi jednostavnosti, ali i verodostojnosti prikaz trenutnog stanja rotora realizovan je po uzoru na Enigma mašinu (sferni prozor sa naznačenim stanjem/slovom za svaki rotor).
- ulazno i izlazno slovo šifrovanja.
  - Kako simulacija služi za šifrovanje manje količine teksta prirodno se nameće korišćenje tekstualne labele, pri čemu se ostavlja mogućnost horizontalnog skrolovanja u slučaju dužeg teksta.

3.5 Pregled paketa

Kompletna aplikacija može se podeliti na dva dela:

- deo zadužen za implementaciju algoritma Enigma mašine
- deo zadužen za sam dizajn, odnosno vizuelizaciju.

Shodno tome, dijagram paketa sistema (engl. *package diagram*) izgleda poput slike 9.



Slika 9 - Dijagram paketa aplikacije

Mogu se uočiti sledeći paketi:

- *rs.ac.bg.etf.mn170085d* – glavni paket aplikacije koji služi kao ulazna tačka
- *enigma* –paket zadužen za rad algoritma Enigma mašine

- *gui* – paket zadužen za sva potrebna iscrtavanja aplikacije

Treba napomenuti da su na dijagramu prikazani samo paketi koji su napravljeni od strane autora. Svakako da aplikacija koristi i gorepomenuti *JavaFX* paket, kao i standardni *java.util* paket, međutim, oni su samo uvezeni, bez dodatih implementacija.

3.5.1 Paket rs.ac.bg.etf.mn170085d

Paket *rs.ac.bg.etf.mn170085d* predstavlja glavni paket aplikacije i kao takav služi kao ulazna tačka programa. Odgovaran je za pokretanje *GUI* aplikacije i definisanje svih globalnih vrednosti, podešavanja i stilova.

3.5.2 Paket enigma

Paket *enigma* sadrži celokupnu logiku potrebnu za rad istoimenog algoritma, kao i dodatne pomoćne metode potrebne za grafički prikaz algoritma.

Kompozicija osnovnih delova mašine, pri čemu svaki od njih ima sopstvenu odgovornost pri šifrovanju, a združeno formiraju Enigmu, prirodno diktira granularnu implementaciju algoritma i poštovanje *SOLID* principa o samo jednoj odgovornosti[4] (engl. *single responsibility principle*) [19]. Na ovaj način, izbegava se problem monolitnih klasa, povećava se čitljivost programa i omogućava lakše održavanje sistema.

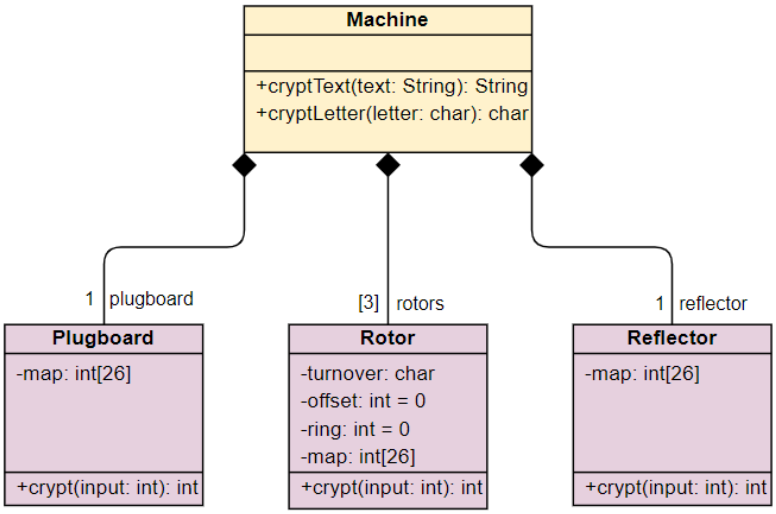
Osnovni delovi mašine, u ovom slučaju klase paketa *enigma* su:

- *Plugboard* – priključna ploča zadužena za inicijalno, zbog simetričnosti algoritma i finalno, preslikavanje karaktera,
- *Rotor* – pored svojih karakteristika ima informaciju i o mapi mapiranja uz pomoć koje obavlja kriptovanje,
- *Reflector* – poput rotora i reflektor ima svoju mapu preslikavanja.

Kao što je već rečeno, prethodno nabrojani delovi čine samu mašinu (atributi klase *Machine*) koja uz pomoć njih implementira logiku šifrovanja.

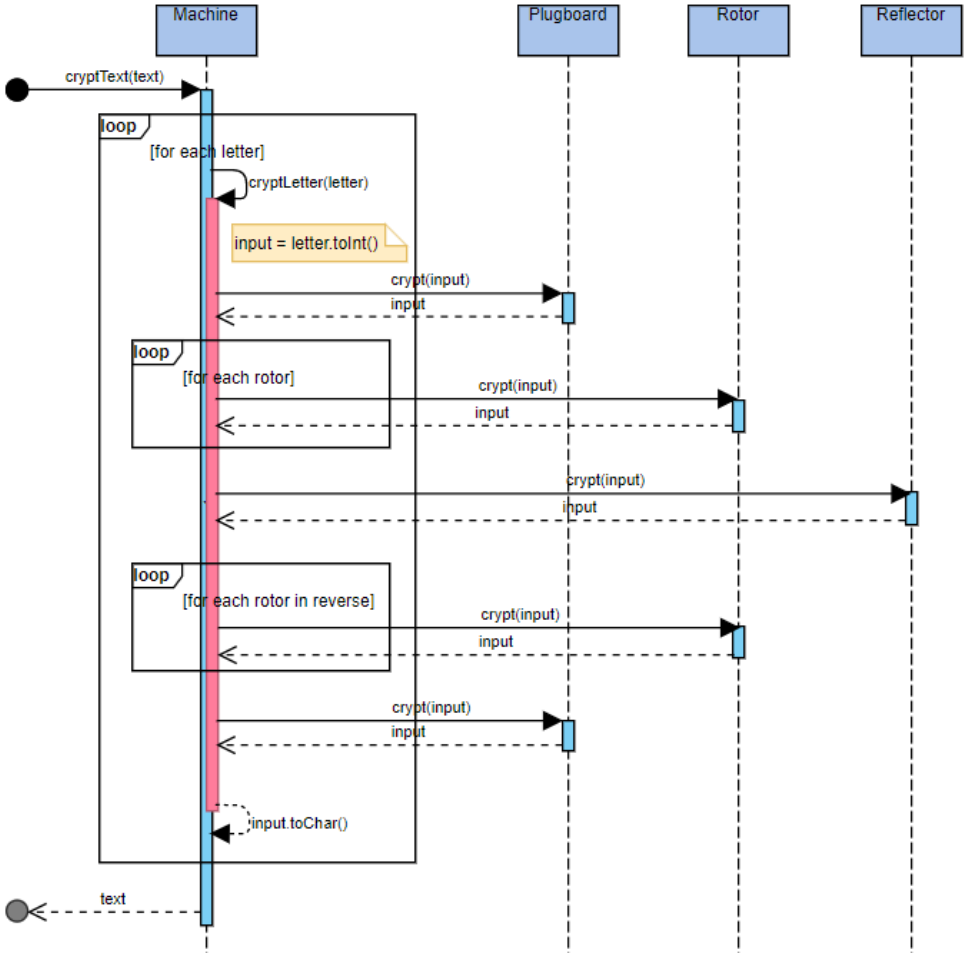
Pojednostavljen dijagram klasa (engl. *class diagram*) paketa *enigma* može se videti na slici 10.

[4] Objekat treba da ima samo jedan fokus, odnosno klasa treba da ima samo jedan razlog za izmenu.



Slika 10 - Pojednostavljen dijagram klasa paketa *enigma*

Na slici 11 prikazan je dijagram sekvence (engl. *sequence diagram*) u slučaju šifrovanja proizvoljnog teksta. Tom prilikom, za svako slovo teksta pozivaju se redom funkcije šifrovanja za priključnu ploču, sva tri rotora i reflektor, a potom za rotore u obrnutom poretku i još jednom za priključnu ploču – kao što je i prikazno na slici 2.



Slika 11 - Dijagram sekvenci za šifrovanje teksta

3.5.3 Paket gui

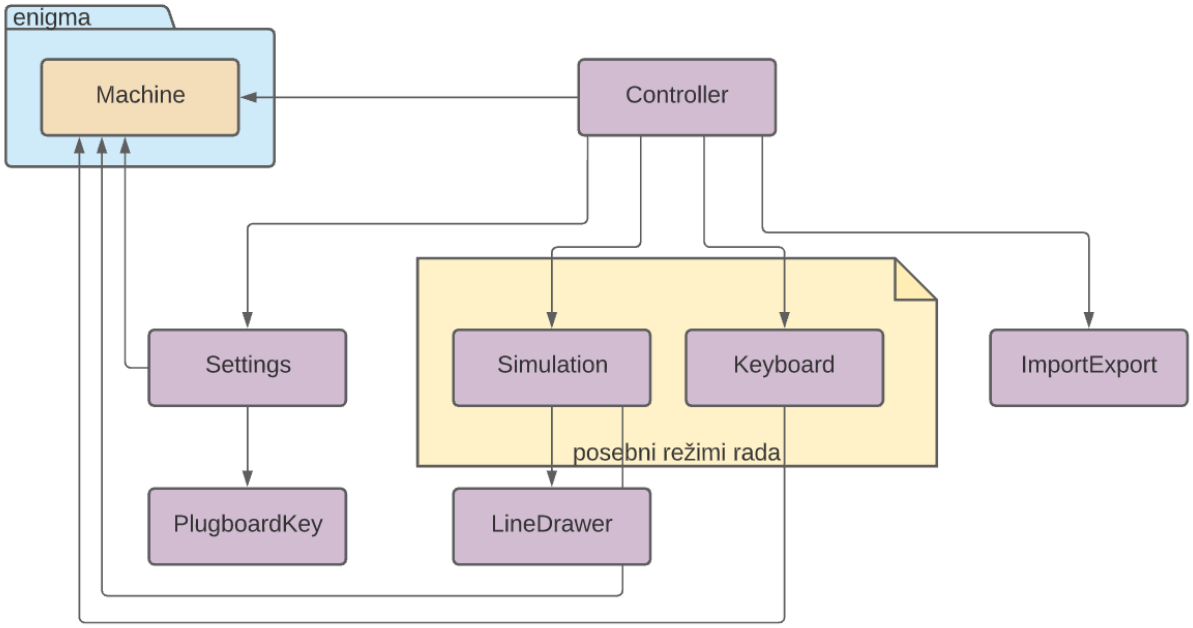
Paket *gui* sadrži celokupnu logiku potrebnu za iscrtavanje i rad aplikacije, kao i za vizuelno prikazivanje rada algoritma.

Kako je za prikazivanje algoritma nužno i poznavati sam algoritam paket *gui* je čvrsto spregnut sa paketom *enigma* kroz klasu *Machine*. Naime, mnoge klase paketa *gui* poseduju asocijativnu vezu ka klasi *Machine* kako bi adekvatno, u skladu sa podešenom mašinom, rukovale aplikacijom i osvežavale prikaz.

Za glavnu klasu ovog paketa izdvaja se klasa *Controller* koja ujedno predstavlja i srž simulatora. Približnije, dužnost *Controller*-a jeste obrada ulaznih, odnosno korisničkih, akcija i izvršavanje aplikativne logike uz potrebne izmene grafičkog prikaza. Samim tim, *Controller* upravlja pojedinačnim delovima (tkzv. prozorima) pokrenute aplikacije, kao što su:

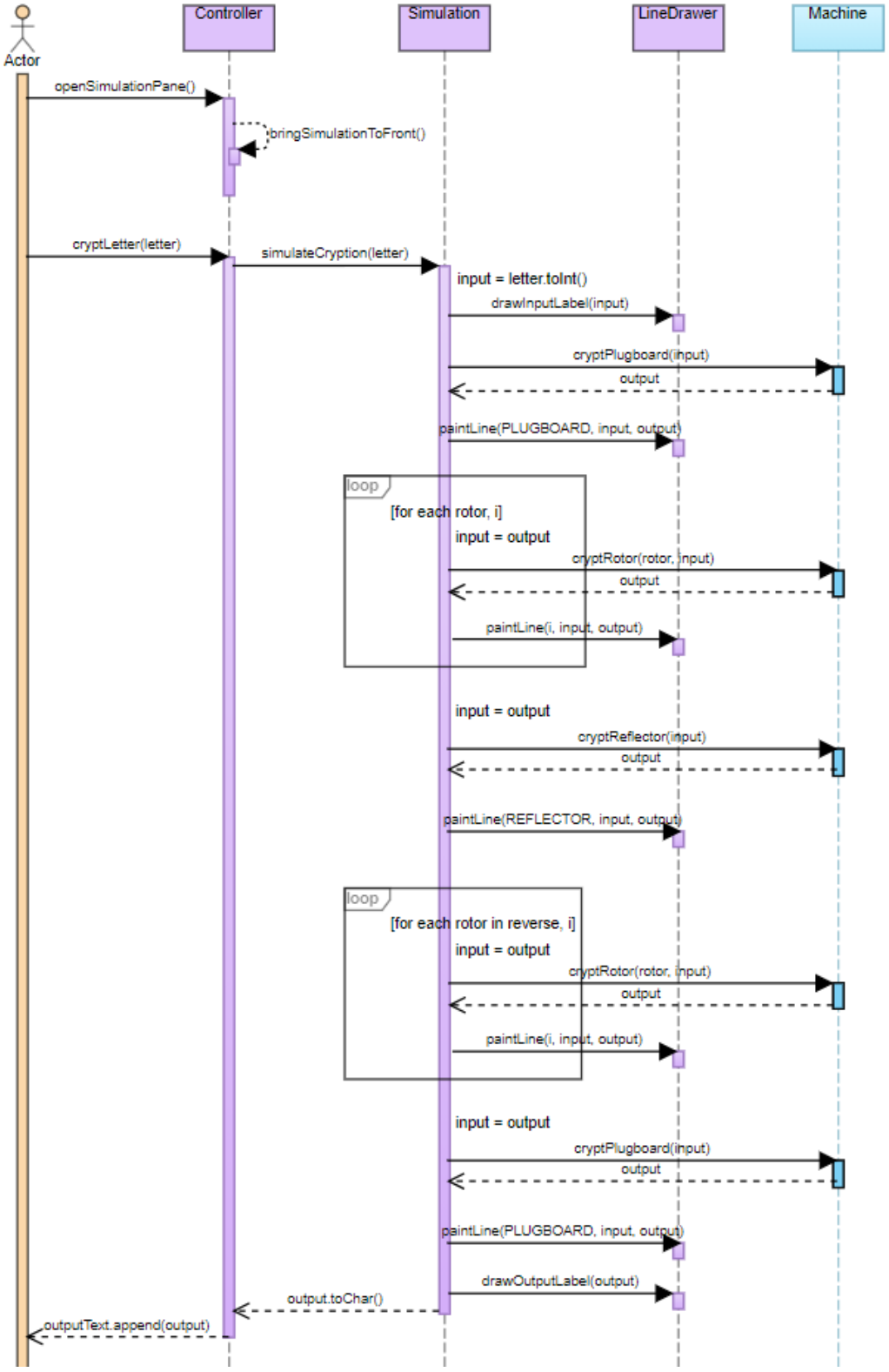
- *Simulation*
  - Klasa je zadužena za grafički prikaz samog algoritma i pri tome koristi pomoćnu klasu *LineDrawer* čija je odgovornost iscrtavanje linija, odnosno električnih puteva (tkzv. unutrašnjih mapiranja), pojedinačnih elemenata Enigme. Mapiranja se crtaju inicijalno, u zavisnosti od podešavanja mašine, kao i prilikom šifrovanja karaktera, kada se putanja šifrovanja istaknuto markira u odnosu na ostala povezivanja.
- *Keyboard*
  - Klasa je zadužena za grafički prikaz simulacije kucanja po Enigmi.
- *Settings*
  - Klasa je zadužena za postavljanje podešavanja rotora, reflektora, kao i priključne ploče, pri čemu pomoćna klasa *PlugboardKey* služi za uparivanje ulaza priključne ploče i njihovo markiranje.
- *ImportExport*
  - Pomoćna klasa uz pomoć koje se realizuje uvoz i izvoz *.txt* fajlova u standardnom (*textbox*) režimu rada.

Uprošćen dijagram klasa (engl. *class diagram*) paketa *gui* može se videti na slici 12.



Slika 12 - Pojednostavljen dijagram klasa paketa *gui*

Primer komunikacije klasa tokom simulacijskog šifrovanja jednog karaktera može se videti na dijagramu sekvenci na slici 13. Naime, klasa *Controller* delegira akciju šifrovanja na klasu *Simulation* koja pomoću klase *Machine* obavlja šifrovanje korak po korak i uporedo pomoću klase *LineDrawer* svaki korak iscrtava na prozoru. Takođe, treba napomenuti da je klasa *Machine* ovom prilikom proširena pomoćnim funkcijama koje obezbeđuju parcijalno šifrovanje pozivajući metode konkretnih delova mašine (npr. *cryptPlugboard* poziva metodu *crypt* klase *Plugboard*). Međutim, ovi dodatni pozivi od strane klase *Machine* ka ostalim delovima mašine nisu obuhvaćeni dijagramom na slici 13 kako se šema ne bi dodatno komplikovala.

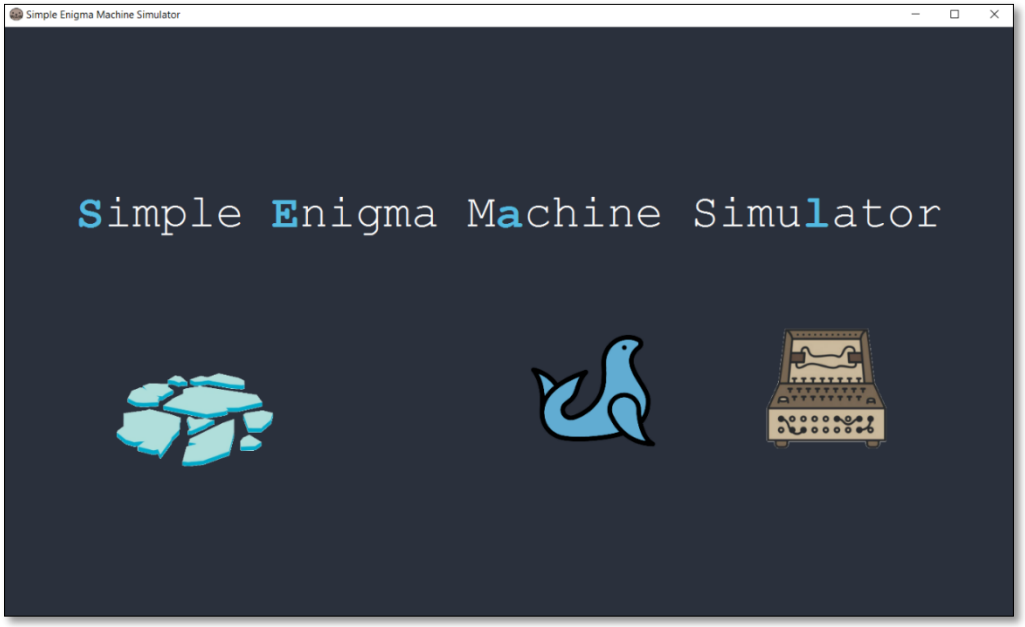


Slika 13 - Dijagram sekvenci za simulacijsko šifrovanje

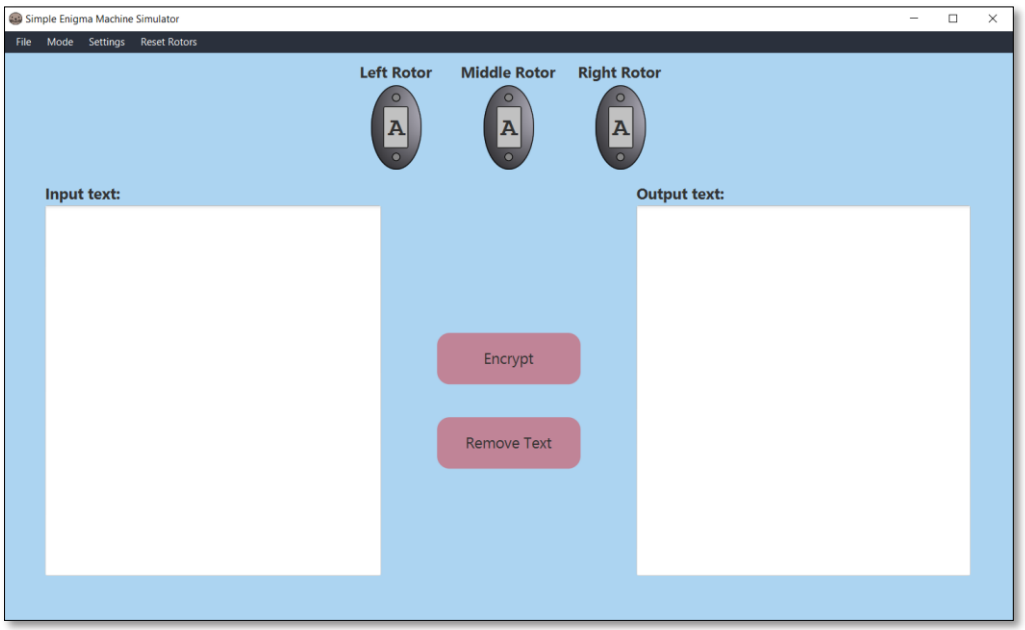
4. NAČIN KORIŠĆENJA *SEAL* SIMULATORA

Pun naziv simulatora je ***S**imple **E**nigma M**a**chine Simu**l**ator*, a njegov akronim *SEAL*. Prirodno se nameće da maskota aplikacije bude foka (engl. *seal*).

Prilikom pokretanja aplikacije korisniku se otvara ekran sa animacijom učitavanja (Slika 14), a nakon toga se otvara početni ekran (Slika 15).



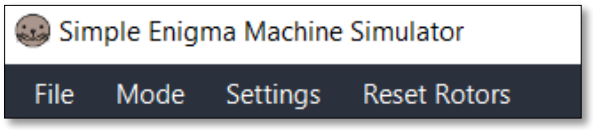
Slika 14 - Ekran sa animacijom učitavanja



Slika 15 - Početni ekran

4.1 Korisnički meni

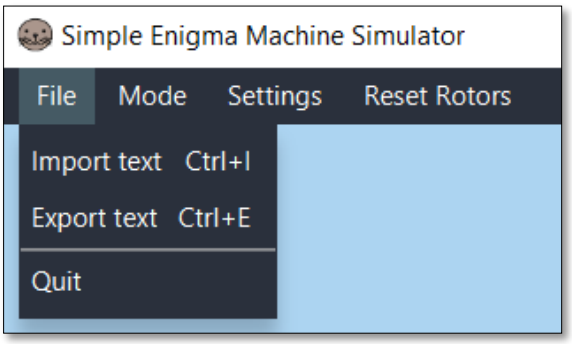
U gornjem delu prozora simulatora uočava se korisnički meni (Slika 16) uz pomoć kojeg je moguće pristupiti svim prozorima i opcijama simulatora. Tokom korišćenja simulatora korisnički meni konstantno je pristupačan.



Slika 16 - Korisnički meni

4.1.1 File

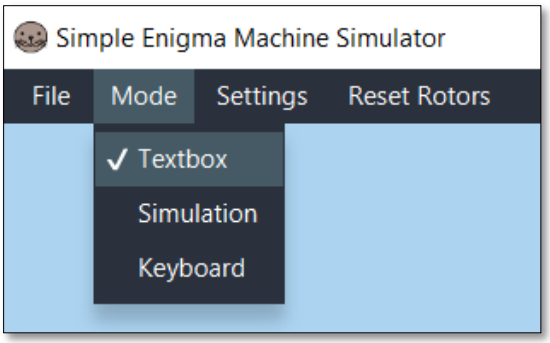
Prva stavka korisničkog menija *File* (Slika 17) omogućava uvoz i izvoz teksta (detaljno opisano u poglavlju 4.2.1.1), kao i gašenje aplikacije na dugme *Quit*.



Slika 17 - Korisnički meni – *File*

4.1.2 Mode

Druga stavka korisničkog menija *Mode* (Slika 18) omogućava prelazak u različite režime rada simulatora (vidi poglavlje 4.2). Trenutno izabrani režim rada označen je ikonicom ✓.



Slika 18 - Korisnički meni – *Mode*

4.1.3 *Settings*

Pritiskom na dugme *Settings* korisniku se otvara prozor namenjen za konfiguraciju mašine (vidi poglavlje 4.3).

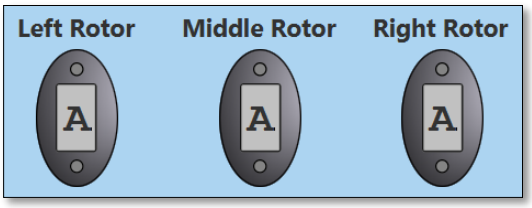
4.1.4 *Reset Rotors*

Dugme *Reset Rotors* vraća rotore mašine na početna podešavanja – ona koja su izabrana u *Settings* prozoru.

4.2 Režimi rada

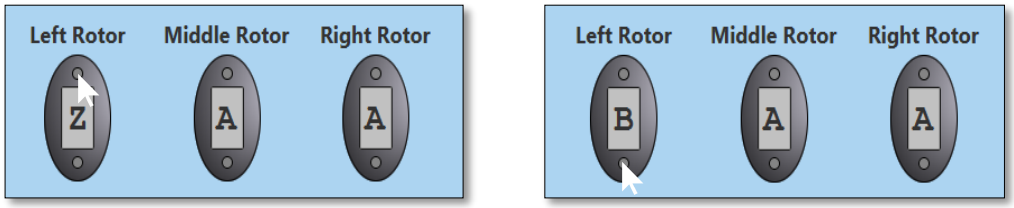
Simulator nudi korisniku na raspolaganju tri različita režima rada kojima je moguće pristupiti putem korisničkog menija.

Zajedničko za sve režime rada jeste da dele istu mašinu pa samim tim dele i prikaz rotora (Slika 19), odnosno njihove trenutne pozicije. Inicijalno, svi rotori su podešeni na prvu poziciju, poziciju A.



Slika 19 - Prikaz rotora

Prikaz rotora nije samo informativnog karaktera, već ujedno pruža mogućnost navijanja rotora pritiskom na kružiće iznad i ispod odgovarajućeg prozora rotora (Slika 20). Pritiskom na kružić iznad rotor se navija cirklurano unazad (A -> Z, B -> A, itd.), dok se pritiskom na kružić ispod rotor navija cirkularno u suprotnom smeru, iliti unapred (A -> B, B -> C, itd.).



Slika 20 - Navijanje rotora unazad i unapred, respektivno

4.2.1 Textbox

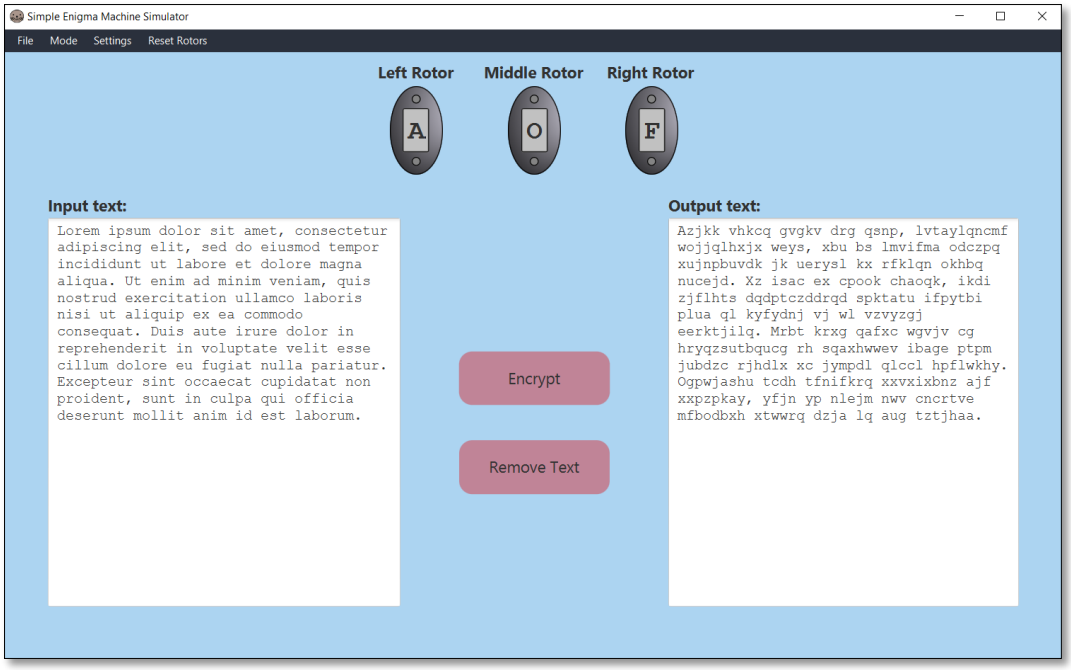
Standardni režim rada, iliti *textbox*, ujedno je i inicijalni režim rada koji se prikazuje odmah po pokretanju aplikacije (Slika 15). Ovaj režim rada pogodan je za (de)šifrovanje veće količine teksta.

Ulazni tekst se unosi u polje sa leve strane prozora aplikacije. Pritiskom na dugme *Encrypt* tekst se šifruje i pojavljuje u polju sa desne strane prozora. Tokom šifrovanja i prikaz rotora se ažurira na trenutne pozicije. Pritiskom na dugme *Remove Text* uklanja se tekst iz ulaznog kao i iz izlaznog polja.

Treba napomenuti da karakteri koji nisu u sklopu azbuke Enigme (iliti 26 slova abecede) bivaju prepisani i pri tome ne dolazi do rotacije rotora. Drugim rečima, tekst *ħ!* biće šifrovan u *ħ!* bez ikakve izvršene rotacije rotora – tačnije, nevalidan unos se ne šifruje.

Nakon šifrovanja teksta ažurirano stanje rotora se koristi za naredno šifrovanje – stanje rotora ne biva resetovano, ali se to može učiniti pritiskom na dugme *Reset Rotors* u korisničkom meniju. Takođe, svaki naknadni unos ulaznog teksta, ili njegova izmena, rezultuje u brisanje izlaznog i ponovno šifrovanje celokupnog ulaznog teksta.

Primer šifrovanja jednog teksta, kao i trenutne pozicije rotora, može se videti na slici 21.

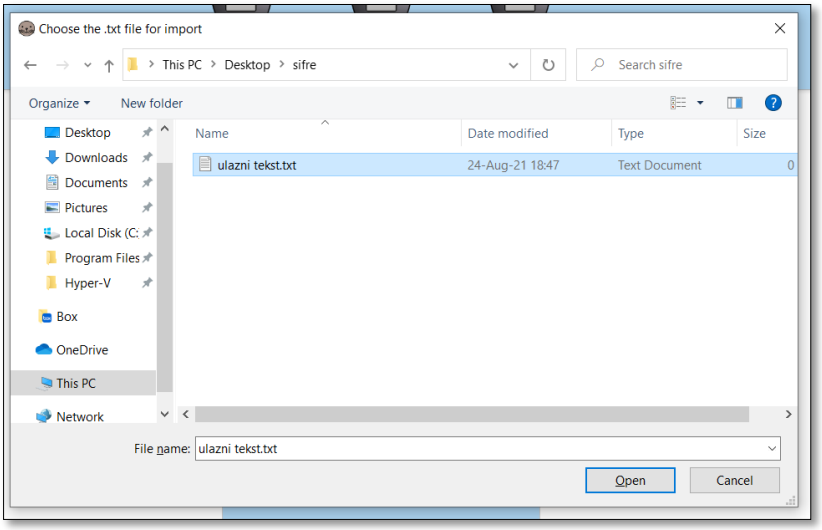


Slika 21 - Režim rada *Textbox* - primer šifrovanja

4.2.1.1 Uvoz i izvoz teksta

Radi pojednostavljenog unosa teksta omogućen je uvoz ulaznog teksta iz *.txt* fajla. Ovoj opciji pristupa se ili iz korisničkog menija *File -> Import text* ili korišćenjem prečice na tastaturi *Ctrl*

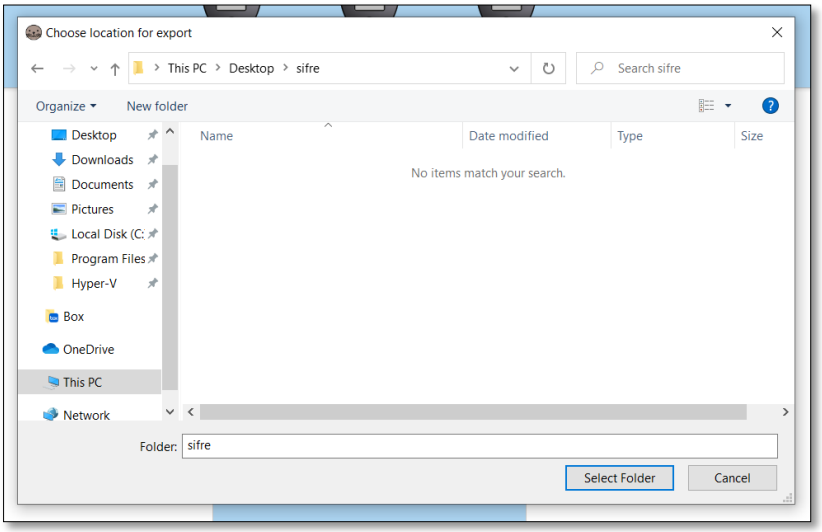
+ *I*. Tom prilikom, otvara se novi prozor pomoću kojeg se može izabrati pomenuti ulazni fajl (Slika 22).



Slika 22 - Prozor za uvoz teksta

Dodatno, omogućen je i izvoz izlaznog teksta, takođe u *.txt* formatu. Analogno opciji uvoza i ovoj opciji se pristupa bilo iz korisničkog menija *File -> Export text* ili korišćenjem prečice na tastaturi *Ctrl + E*. Putem novootvorenog prozora (Slika 23) bira se lokacija na kojoj će se sačuvati izlazni tekst.

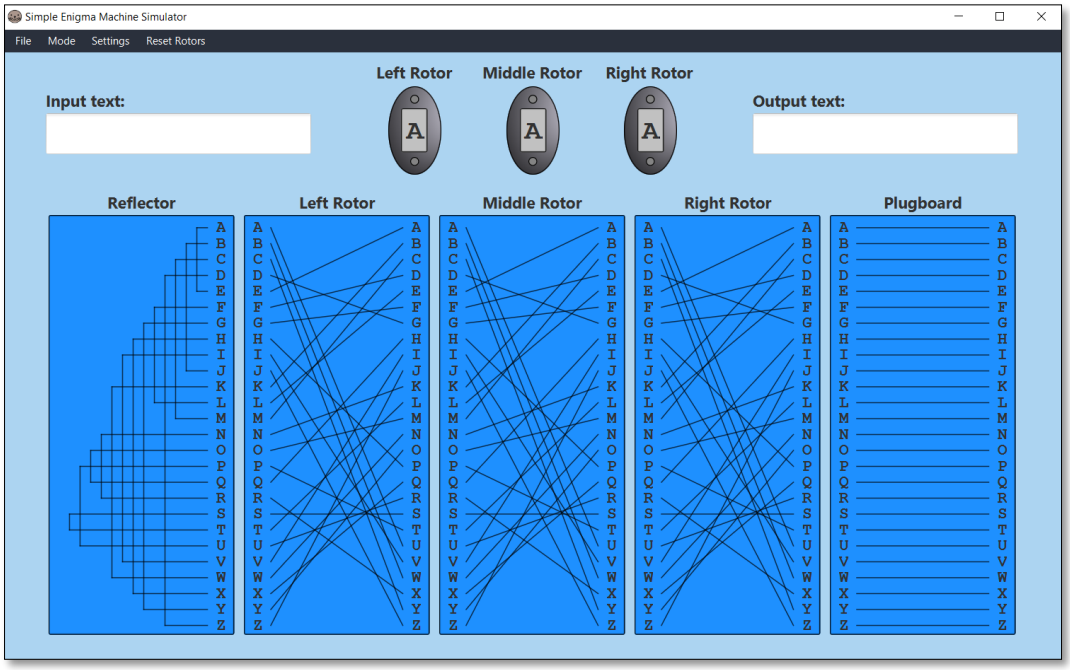
Napominje se da izvoz praznog izlaznog teksta nije moguć i da pokušaj takve akcije ignoriše, kao i da se naziv izlaznog teksta formira automatski po formatu *exportedFileBySEAL_* + %timestamp% + *.txt* (npr. *exportedFileBySEAL_1629823561430.txt*).



Slika 23 - Prozor za izvoz teksta

4.2.2 *Simulation*

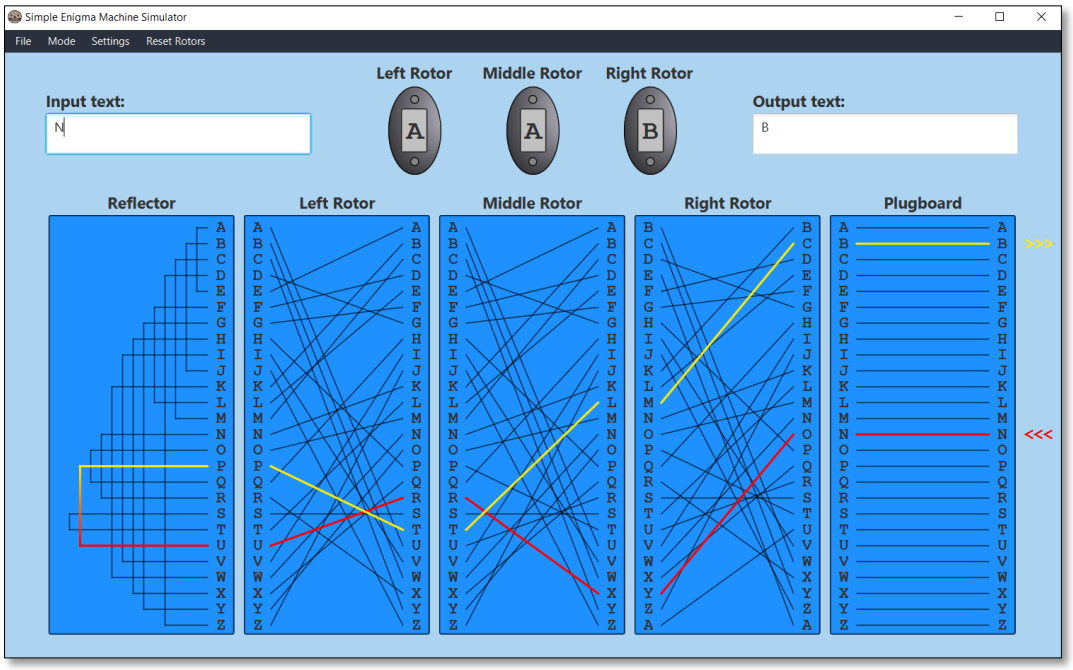
Simulacijski režim rada (Slika 24) omogućava korisniku da zaviri u unutrašnjost Enigma mašine i interaktivno stekne uvid u sam algoritam šifrovanja. U ovom režimu moguće je ispratiti unutrašnja povezivanja svih delova Enigma mašine (zdesna nalevo na slici, to su priključna ploča, rotori, reflektor). Konkretna unutrašnja povezivanja zavise od izabrane konfiguracije.



Slika 24 - Režim rada *Simulation*

Prilikom unosa pojedinačnog karaktera u polje sa leve strane, pored ažuriranog prikaza rotora i izlaznog šifrovanog karaktera sa desne strane, može se opaziti i način na koji je šifrovanje postignuto, iliti putanja kretanja električnog signala kod Enigma mašine. Ulazna putanja, označena crvenom bojom, počinje od priključne ploče i putuje kroz rotore do reflektora. Od reflektora, putanja, sada označena žutom bojom, se vraća do izlaza, prolazeći kroz rotore u suprotnom smeru. Za svaki naredni karakter putanja se prikladno ažurira. Primer jednog šifrovanja prikazuje slika 25.

Za razliku od *Textbox* režima rada u kome se ulazni tekst svaki put iznova šifruje (nakon pritiska dugmeta), u *Simulation* režimu rada otkucani karakter uzrokuje šifrovanje isključivo tog novododatog karaktera, dok se već šifrovani karakter neće ponovno šifrovati. Takođe, kriptovanje se vrši automatski, bez eksplicitnog dugmeta, čim se pritisne neko dugme tastature. U slučaju pritiska dugmeta tastature koje nije iz azbuke Enigme taj karakter se neće razmatrati, niti upisivati kao ulazni znak.



Slika 25 – Režim rada *Simulation* – primer šifrovanja

4.2.3 Keyboard

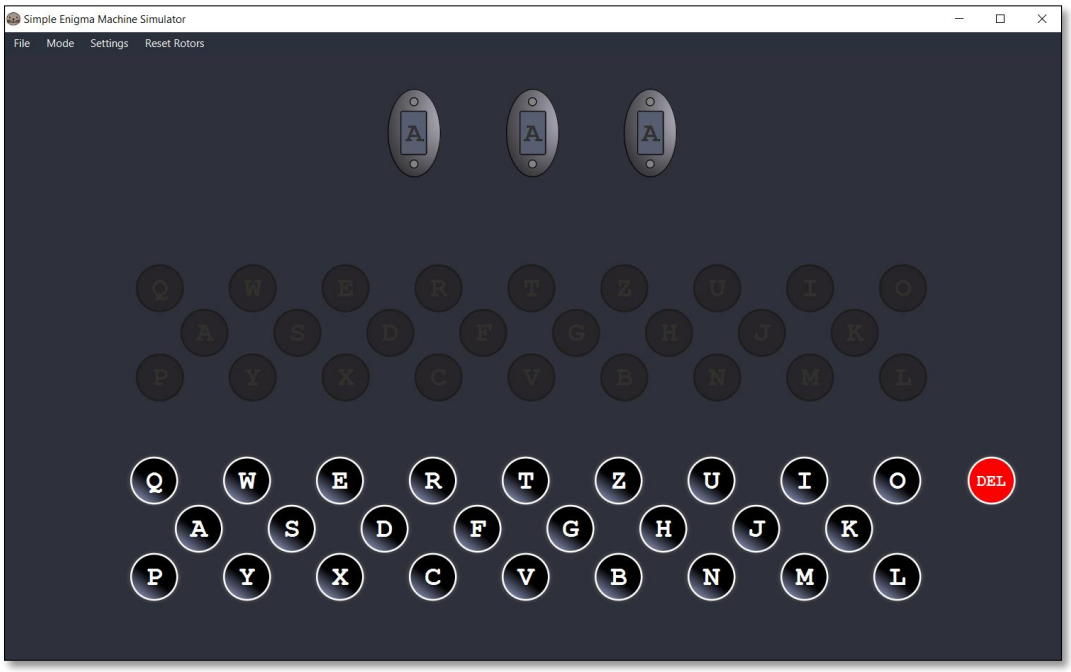
Režim rada *Keyboard* (Slika 26) pruža korisniku stvaran utisak kucanja po Enigma mašini.

Prozor je izdeljen na dva dela, u donjem delu se nalazi tastatura za kucanje ulaznog teksta, dok se u gornjem delu nalaze lampice koje predstavljaju izlaz šifrovanja. Na ekranu, iznad tastature prikazuje se ulazni (otkucani) tekst, dok se iznad lampica prikazuje izlazni (šifrovani) tekst.

Korisnik može tekst unositi na dva načina: kucajući po svojoj tastaturi ili mišem pritiskajući odgovarajući taster (slovo) na ekranu. Prilikom unosa karaktera, jedna od 26 lampica biva obasjana označavajući šifrovani karakter. Trake u kojima se prikazuju ulazni i izlazni tekst se automatski brišu kada se napune (kapacitet cca 80 karaktera). Istoriju kucanja moguće je obrisati i putem crvenog dugmeta *DEL* na ekranu, ili pritiskom dugmeta *backspace* na tastaturi.

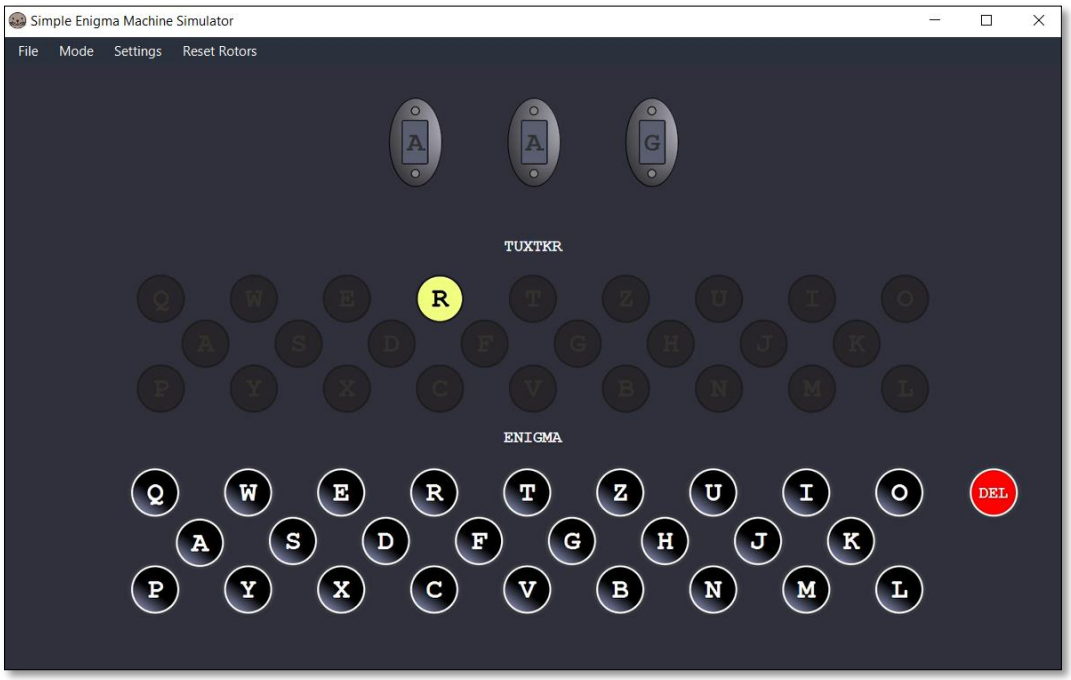
Napominje se da je opcija čuvanja istorije kucanja dodata radi lakšeg korišćenja simulatora – Enigma mašina nije posedovala pamćenje ulaznog ili šifrovanog teksta već bi se on zasebno, ručno beležio.

Kao i kod *Simulation* režima rada, ni *Keyboard* režim rada ne dozvoljava unos karaktera koji nisu iz azbuke Enigme – takav unos se ignoriše.



Slika 26 - Režim rada *Keyboard*

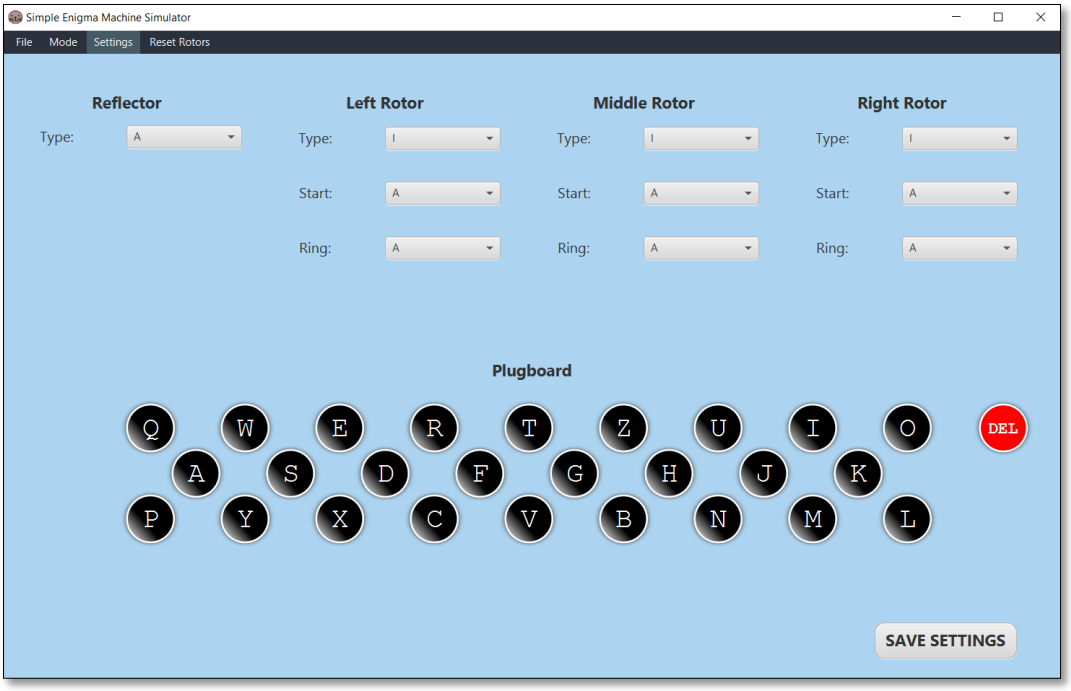
Primer jednog šifrovanja u *Keyboard* režimu rada prikazan je na slici 27. Može se uočiti da se poslednje otkucano slovo A šifrovalo u slovo R (sija lampica).



Slika 27 - Režim rada *Keyboard* - primer šifrovanja

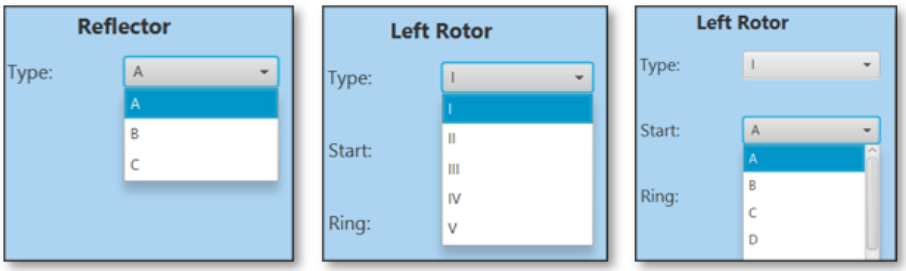
4.3 Podešavanja mašine

Mašina koja se koristi u simulatoru se konfiguriše pritiskom na dugme *Settings* iz korisničkog menija. Tom prilikom otvara se nov prozor čiji je izgled dat u nastavku – u gornjoj polovini podešavaju se valjci (rotori i reflektor), a u donjoj priključna ploča (engl. *plugboard*). Podrazumevana, inicijalna podešavanja mašine su takođe prikazana na slici 28 – sve opcije postavljene na prvi izbor, dok su ulazi priključne ploče razvezani.



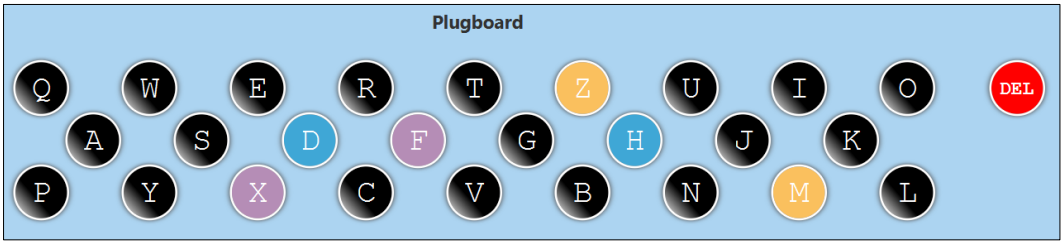
Slika 28 - *Settings* prozor

Simulator koristi Enigma I mašinu (vidi poglavlje 2.3.1) i u skladu sa tim moguće je postaviti tipove (engl. *type*) pojedinačnih rotora i reflektora, kao i početna navijanja (engl. *start*) i položaje prstena (engl. *ring*) pojedinačnih rotora. Podešavanja se biraju iz padajuće liste.



Slika 29 - Podešavanja reflektora i rotora

U donjoj polovini ekrana moguće je konfigurisati priključnu ploču, odnosno izabrati parove. Pritiskom na dva različita karaktera na ploči njihovi ulazi bivaju spojeni – simulator te ulaze boji istom bojom. Primera radi, slika 30 ukazuje na uparene ulaze D-H, F-X, M-Z.



Slika 30 - Priključna ploča - spajanje ulaza

Pritiskom na crveni taster *DEL* svi ulazi bivaju raspareni. Svakako, moguće je i pojedinačno raspariti ulaze – pritiskom na bilo koji upareni karakter isti biva rasparen.

Važno je napomenuti da izabrana podešavanja postaju validna tek nakon pritiska na dugme *Save Settings* koje se nalazi u donjem desnom uglu prozora.

5. ZAKLJUČAK

Enigma mašina predstavlja prekretnicu između stare, ručne enkripcije na papiru i moderne, digitalne enkripcije. Samim tim, od velike je važnosti razumeti način funkcionisanja ovog polialfabetskog algoritma, ali i njegove prednosti i mane. Stoga, tematika ovog rada jeste simulator Enigma mašine – konkretno, Enigma I. Sam simulator razvijen je prvenstveno kao pomoćno sredstvo prilikom razumevanja načina rada kriptografskog algoritma mašine i kao takav za ciljnu grupu favorizuje studente, ali i zaljubljenike u tajanstveni svet kriptografije i istorije.

Rad je započet osvrtom na motivaciju koja stoji iza razvoja aplikacije, kao i kratkom istorijskom retrospektivom mašine. Potom je temeljno objašnjena teorijska struktura algoritma, princip rada i analiza složenosti, samim tim i sigurnosti, istog. U narednim glavama opisana je arhitektura implementacije, dat je prikaz funkcionalnih zahteva i priloženo je korisničko uputstvo celokupnog simulatora.

Tokom razvoja alata, uočeni su određeni nedostaci, čije bi otklanjanje u budućnosti doprinelo kvalitetu korišćenja simulatora. Takođe, primećena su i konkretna stilska i funkcijska poboljšanja, koju su zajedno sa nedostacima data u nastavku:

- Trenutno je atraktivnost desktop aplikacija znatno manja u odnosu na mobilne ili web aplikacije. Radi bolje pristupačnosti simulator, u slučaju razvoja za mobilne uređaje moguće je koristiti projekat otvorenog koda (engl. *open-source*) *JavaFXPorts* održavan od strane *Gluon*-a [20], a čija je glavna namena prenošenje *JavaFX* koda na mobilne (*Android* i *iOS*) i ugrađene (engl. *embedded*) sisteme.
- Simulator funkcioniše po principu Enigme I. Jedno od proširenja bila bi podrška drugih verzija mašine, njihovo korišćenje kao i upoređivanje. Za ovakvu realizaciju potrebno je prilagoditi *GUI* (uvođenje opcije biranja mašine) i registrovati konfiguracije novih mašina, dok se logika algoritma ne bi pretežno menjala, eventualno pojedinačnih delova (npr. ukoliko reflektor više nije statičke konfiguracije). Moguće je i uvesti nov režim rada koji bi služio isključivo za komparaciju različitih mašina.
- U širem smislu, aplikaciju je moguće dalje razviti kao sistem za slanje tajnih poruka među korisnicima. Tom prilikom, idealno bi bilo preći na implementaciju *Web* aplikacije pomoću nekog jednostavanog, modernog i brzog *framework*-a kao što je *Spring* [21]. Vizuelizaciju algoritma, kao i sam *GUI*, najjednostavnije je realizovati posredstvom modernih *framework*-a (npr. *Angular* [22]) i biblioteka.
- Kako se danas pretežno sve aplikacije razvijaju sa minimum dve različite teme (npr. svetla i tamna), jedno od stilskih poboljšanja može biti uvođenje tamne teme (engl. *dark mode*). Tokom stilskog preobražaja, treba obratiti pažnju na kontrast boja, formatiranje teksta i sveobuhvatnu pristupačnost aplikacije (engl. *web accessibility*). [23]

- Tokom korišćenja simulatora ne emituje se nikakav zvuk. Potencijalno se mogu uvesti zvučni efekti prilikom kucanja teksta, rotacije valjaka i drugih korisničkih akcija, kako bi se sam utisak korisnika upotpunio.

LITERATURA

[1] „History of cryptography,“ Wikipedia, [Na mreži]. Available: https://en.wikipedia.org/wiki/History_of_cryptography. [Poslednji pristup 15. jul 2021.].

[2] S. Singh, The Code Book - How to make it, break it, hack it, crack it, Delacorte Press, 2001.

[3] S. Faint, The Enigma History and Mathematics, University of Waterloo, 1999.

[4] „Enigma A,“ Crypto Museum, [Na mreži]. Available: https://www.cryptomuseum.com/crypto/enigma/a/index.htm. [Poslednji pristup 20. jul 2021.].

[5] J. Andress, The Basics of Information Security, Syngress, 2014.

[6] W. Stallings, Cryptography and Network Security - Principles and Practice, Pearson, 2017.

[7] „Schreibmax,“ Crypto Museum, [Na mreži]. Available: https://www.cryptomuseum.com/crypto/enigma/schreibmax/index.htm. [Poslednji pristup 20 jul 2021.].

[8] H. Ulbricht, Die Chiffriermaschine ENIGMA, 2005.

[9] „Enigma family tree,“ Crypto Museum, [Na mreži]. Available: https://www.cryptomuseum.com/crypto/enigma/tree.htm. [Poslednji pristup 20. jul 2021.].

[10] J. Hubrich, Cracking Enigma, a (short) Summary of the past.

[11] G. Sullivan i F. Weierud, Breaking German Army Ciphers.

[12] E. Roberts, „The Enigma Machine,“ 2016. [Na mreži]. Available: http://stanford.edu/class/archive/cs/cs106a/cs106a.1164/handouts/29-TheEnigmaMachine.pdf. [Poslednji pristup 20. jul 2021.].

[13] Oracle, „Java,“ [Na mreži]. Available: https://www.oracle.com/java/. [Poslednji pristup 20. jul 2021.].

[14] Oracle, „Java SE Development Kit 16,“ [Na mreži]. Available: https://www.oracle.com/java/technologies/javase-jdk16-downloads.html. [Poslednji pristup 20. jul 2021.].

[15] OpenJFX, „JavaFX,“ [Na mreži]. Available: https://openjfx.io/. [Poslednji pristup 20 jul 2021.].

[16] Oracle, „JavaFX Scene Builder,“ [Na mreži]. Available: https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html. [Poslednji pristup 20. jul 2021.].

[17] „Maven,“ [Na mreži]. Available: https://maven.apache.org/. [Poslednji pristup 20. jul 2021.].

[18] Oracle, „Java Swing,“ [Na mreži]. Available: https://docs.oracle.com/javase/8/docs/api/index.html?javax/swing/package-summary.html. [Poslednji pristup 20. jul 2021.].

[19] S. Millington, „A solid guide to SOLID principles,“ [Na mreži]. Available: https://www.baeldung.com/solid-principles. [Poslednji pristup 20. jul 2021.].

[20] Gluon, „JavaFXPorts,“ [Na mreži]. Available: https://gluonhq.com/products/mobile/javafxports/. [Poslednji pristup 20. jul 2021.].

[21] Spring, „Spring Framework,“ [Na mreži]. Available: https://spring.io/projects/spring-framework. [Poslednji pristup 20. jul 2021.].

[22] Angular, „Angular framework,“ [Na mreži]. Available: https://angular.io/. [Poslednji pristup 20. jul 2021.].

[23] WebAIM, „Introduction to Web Accessibility,“ [Na mreži]. Available: https://webaim.org/intro/. [Poslednji pristup 20. jul 2021.].

[24] G. Ellsbury, „The Enigma Machine - Its construction, operation and complexity,“ [Na mreži]. Available: http://www.ellsbury.com/enigma2.htm. [Poslednji pristup 20. jul 2021.].

SPISAK SLIKA

Slika 1 - Cezarova šifra 4
Slika 2 - Ciklus šifrovanja na Enigma mašini 5
Slika 3 - Pojednostavljen rotor 6
Slika 4 - Pojednostavljeno podešavanje prstena 6
Slika 5 - Pojednostavljeno navijanje rotora 7
Slika 6 - Pojednostavljen reflektor 8
Slika 7 – Pojednostavljen primer šifrovanja 10
Slika 8 - Moguće pozicije poznate fraze 14
Slika 9 - Dijagram paketa aplikacije 17
Slika 10 - Pojednostavljen dijagram klasa paketa *enigma* 19
Slika 11 - Dijagram sekvenci za šifrovanje teksta 19
Slika 12 - Pojednostavljen dijagram klasa paketa *gui* 21
Slika 13 - Dijagram sekvenci za simulacijsko šifrovanje 22
Slika 14 - Ekran sa animacijom učitavanja 23
Slika 15 - Početni ekran 23
Slika 16 - Korisnički meni 24
Slika 17 - Korisnički meni – *File* 24
Slika 18 - Korisnički meni – *Mode* 24
Slika 19 - Prikaz rotora 25
Slika 20 - Navijanje rotora unazad i unapred, respektivno 25
Slika 21 - Režim rada *Textbox* - primer šifrovanja 26
Slika 22 - Prozor za uvoz teksta 27
Slika 23 - Prozor za izvoz teksta 27
Slika 24 - Režim rada *Simulation* 28
Slika 25 – Režim rada *Simulation* – primer šifrovanja 29
Slika 26 - Režim rada *Keyboard* 30
Slika 27 - Režim rada *Keyboard* - primer šifrovanja 30
Slika 28 - *Settings* prozor 31
Slika 29 - Podešavanja reflektora i rotora 31
Slika 30 - Priključna ploča - spajanje ulaza 32

SPISAK TABELA

Tabela 1 - Broj dodatih kombinacija preko priključne ploče 10
Tabela 2 - Karakteristike mašine Enigma I 11