PMF Cirilica.wmf

INSTITUT ZA MATEMATIKU i INFORMATIKU

PRIRODNO – MATEMATIČKI FAKULTET

UNIVERZITET U KRAGUJEVCU

SEMINARSKI RAD

LORA

Profesor Student

dr Ana Kaplarević – Mališić Milica Živković

Sadržaj

[1. Uvod 3](#_Toc459746178)

[1.1. Opis igre 3](#_Toc459746179)

[1.2. Ekran za igru 4](#_Toc459746180)

[2. Softversko rešenje 7](#_Toc459746181)

[2.1. IGRAČ 9](#_Toc459746182)

[2.2. IGRA 10](#_Toc459746183)

[2.2.1. Sva srca 10](#_Toc459746184)

[2.2.2. Maksimum 11](#_Toc459746185)

[2.2.3. Dame NE 11](#_Toc459746186)

[2.2.4. Žaca tref 11](#_Toc459746187)

[2.2.5. Šesta ruka i kralj srce 11](#_Toc459746188)

[2.2.6. Minimum 11](#_Toc459746189)

[2.3. ENGINE 12](#_Toc459746190)

[2.4. GUI 12](#_Toc459746191)

[2.5. OSTALE KLASE 13](#_Toc459746192)

[2.5.1. Karta 13](#_Toc459746193)

[2.5.2. Špil 13](#_Toc459746194)

[2.5.3. Znak 13](#_Toc459746195)

[2.5.4. Naziv igara 13](#_Toc459746196)

[2.5.5. Gui karta 13](#_Toc459746197)

[2.5.6. Test 13](#_Toc459746198)

[3. Literatura 14](#_Toc459746199)

1. Uvod

Ovaj seminarski rad je dodatak aplikaciji koja je napravljena iz predmera Objektno – orjentisano programiranje. Aplikacija je pisana u programskom jeziku Java i korišćene su tehnologije koje prate ovaj jezik. Odrađena aplikacija predstavlja model kartaške igre “LORA”.

* 1. Opis igre

Lora je kartaška igra za četiri igrača. Igra se sa 32 karte, jačine od 7, 8, 9, 10, B, Q, K do A, pri čemu je sedmica najslabija, a As (Kec) najjača karta. Karte su raspoređene u četiri boje: pik , karo , herc i tref.

Igra se sastoji od 6 različitih igara, pri čemu svaki igrač bira da se odigra svaka igra jednom. U jednoj partiji ima 24 igara. Igrači biraju igre naizmenično, od igara koje su im preostale. Pobednik partije je igrač koji ima najmanje poena na kraju 24-te igre.

Kod svih igara, svi igrači bacaju po jednu kartu i onaj koji ima najjacu dobija određen broj poena. Jača karta može da nosi slabiju samo ako je iste boje. **Igre:**

https://l.undeliver.com/games/game0.png

Minimum

Cilj igre Minimum je odnetu što manje ruku, po mogućnosti ni jednu. Svaka odneta ruka donosi -1 poen. Još jednom napomenaČ cilj u igri Lore je imati što manje poena.

https://l.undeliver.com/games/game1.png

Maksimum

Cilj igre Maksimum je odneti što više ruku, po mogućnosti sve. Svaka odneta ruka donosi umanjuje poene igrača za jedan.

https://l.undeliver.com/games/game2.png

Srce sve ili ništa

Cilj igre Srce sve ili ništa je odneti sva srca ili, ako to nije moguće, onda odneti što manje. Svako odneto srce donosi +1 poen, osim u slučaju kada igrač odnese svih 8 srca – tada dobija -8 poen. Dakle, ili sva srca ili igrati tako da protivnici rasporede srca između sebe. Ne dozvoliti da neki od protivnika odnese svih 8 srca.

https://l.undeliver.com/games/game3.png

Dame NE

Cilj igre Dame Ne je odneti što manje dama (Q). Svaka odneta dama donosi +2 poena.

https://l.undeliver.com/games/game4.png

Kralj srce i šesta ruka

Cilj igre Kralj srce i šesta ruka je ne nositi kralja (K) srce i ne nositi šestu ruku po redu. Šesta ruka se igra u trenutku kada igrač ima još 3 karte i treba da baci jednu od njih. Igrač koji nosi kralja srce dobija +4 poena, a igrač koji nosi šestu ruku dobija +4 poena.

https://l.undeliver.com/games/game5.png

Žaca tref

Cilj igre Žaca tref je ne nositi žandara u trefu. Igrač koji ga odnese dobija +8 poena.

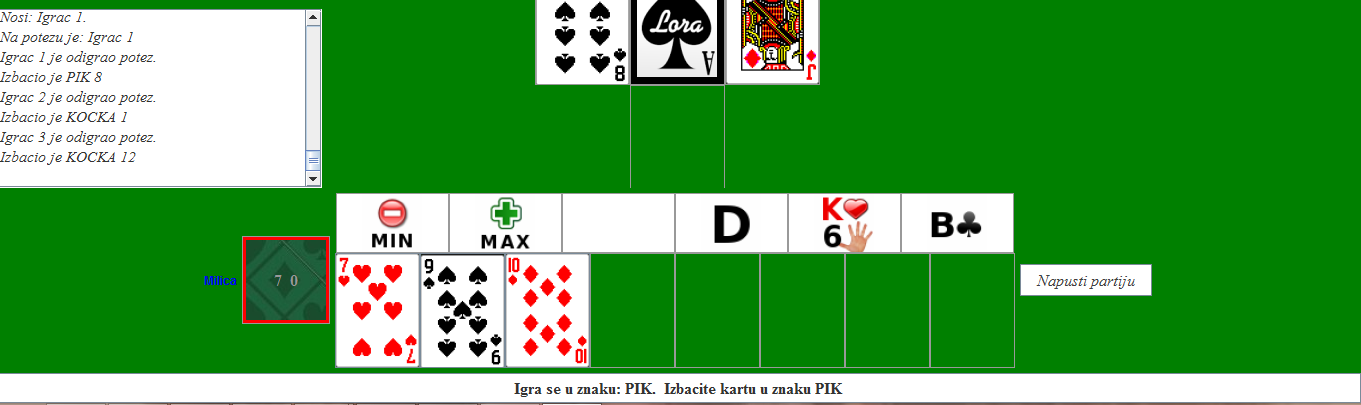
* 1. Ekran za igru

Na početku igrač unosi svoje ime i time igra može da počne.



Slika 1 – Pocetak igre

Način igre je jednostavan. Kada se podele karte, ukoliko je igrač na redu da bira igru, treba da klikne na sličicu igre koju želi da odabere i nakon toga je on na potezu da baci kartu. Karta se baca klikom na istu. Svaki naredni igrač koji je na potezu prati znak izbačene karte, te u koliko korisnik klikne na kartu koja nije u zadatom znaku u donjem uglu prozora pojaviće se obavestenje da potez nije ispravno odigran, a ukoliko nema ni jednu kartu u tom znaku izbacuje po slobodnom izboru bilo koju kartu, ali se ta karta na talonu ne računa pri traženju najveće karte sa talona, već se računaju samo one koje su u zadatom znaku.



Slika 2 – Obaveštenje o izbacivanju karte u pogrešnom znaku

Pored imena igrača nalazi se polje sa dva broja. Broj desno je trenutni broj poena igrača u tekućoj partiji. Broj levo pokazuje broj poena igrača u tekućoj igri. Sa leve strane ekrana nalaze se informacije o trenutnoj igri koja se igra, tačnije ko bira igru, koja igra je izabrana, koje su karte izbačene, kako bi korisnik aplikacije mogao u svakom trenutku da proveri prethodne karte i olakša sebi igu, kao i koji igrač je imao najjaču kartu na talonu. U gornjem desnom uglu prozora stoji ikona igre koja se trenutno igra kao i njen kratak opis.

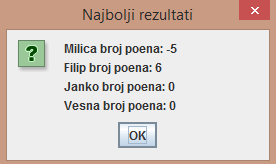
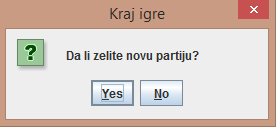


Slika 3 – Partija u toku

U gornjem uglu prozora nalazi se sat koji otkucava igračima vreme potrebno za odigravanje poteza, s tim što je korisniku aplikacije vreme neograničeno. Korisnik aplikacije u svakom trenutku može odustati od partije klikom na dugme „Napusti partiju“ i time se partija završava, računa se trenuti broj poena (u koliko je korisnik prekinuo partiju u togu igranja igre, broj poena stečen u toj igri neće biti uračunat) i sačinjava se rang lista na osnovu broja poena. Takođe sačinjava se i tabela najboljih rezultata prijavljenih igrača. Na kraju igrač može da izabere da li želi da odigra novu partiju.



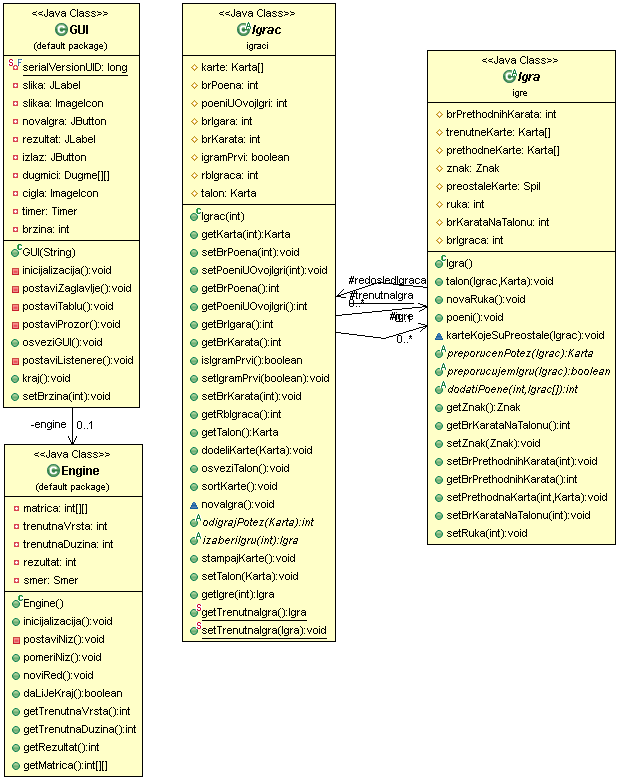
Slika 5 - Kraj partije

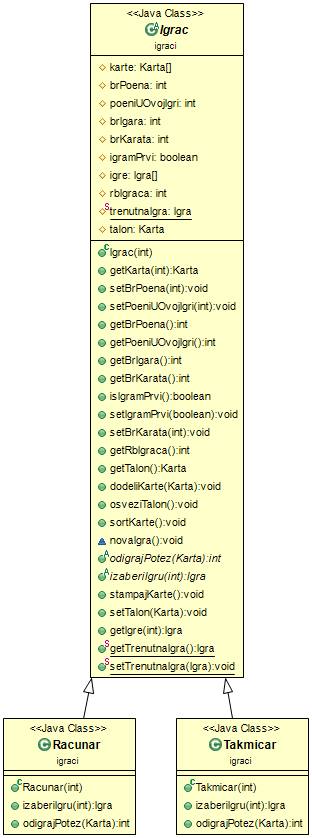
1. Softversko rešenje

Osnovne klase koje su korišćene u izradi ovog projekta su Engine i Gui, a pored njih klasa Igrač, kao apstraktna klasa koju nasleđuju klase Takmičar i Računar, i klasa Igra, takođe apstraktna klasa, koju nasleđuju klase svh tipova igara: „Maksimum“ (Max) , „Minimum“ (Min), „Sva srca“ , „Žaca tref“, „Kralj srce i 6-ta ruka“ i „Dame“.

Suština problema izrade ove aplikasije sastoji se u prividnoj veštačkoj inteligenciji. Zapravo, pošto je u igru uključeno 4 igrača od kojih je jedan korisnik aplikacije, preostala 3 igrača moraju biti kompletno definisana u smislu da je potrebno definisati univerzalno igranje kartaške igre Lora. Igrač, koji nije korisnik aplikacije, treba sam da izabere najpogodniji potez za trenutno izabranu igru, kao i da, u trenutku kada je njegov red, odabere igru na najbolji način na osnovu karata koje su mu podeljene. Čitava izrada problema obuhvaćena je metodama i podacima dveju klasa: klase *Igrac* i klase *Igra*, koje klasa *Engine* obrađuje, povezuje i zatim, kada se kompletna suština logickog problema završi, predaje podatke klasi *GUI* koja nam obezbeđuje izgled aplikacije.



Slika 6 - UML dijagram najbitnijih klasa



Slika 6 – UML dijagram klase Igrač i njenih podklasa

* 1. IGRAČ

Klasa *Igrac* je apstraktna klasa i nju nasleđuju klasa *Racunar* koja predstavlja igrača koji nije korisnik aplikacije i klasa *Takmicar* koja predsavlja korisnika aplikacije. Ove dve podklase klase *Igrac* imaju sebi svojstvene implementacije metoda *odigrajPotez(Karta)* i *izaberiIgru(int)*.

Ono što svaki igrač ima od podataka je: niz karata koje mu se dodeljuje tokom svake igre koje se funkciom *sortKarte()* karte sortiraju u opadajući poredak redom znakovima tref, herc, pik, karo, niz igara koje su mu dostupne da izabere, svoj redni broj koji ga određuje u rang listi, broj poena u trenutnoj igri, kao i ukupan broj poena u trenutnoj partiji, *boolean* promenljivu *igramPrvi* koja te postavljena na *true* onda kada igrač bira igru te prvi izbacuje kartu i onda kada je njegova karta najjača na talonu od karata koje se računaju, tačnije koje su u traženom znaku, promenljiva *talon* je tipa *Karta* i ona pamti kartu koju je igrač izbacio na talon. Specijalna promenljiva *trenutnaIgra* koja je tipa Igra je statička pomenljiva i nju dele svi igrači. Ona čuva podatak o trenutno izabranoj igri, koje su karte trenutno na talonu.

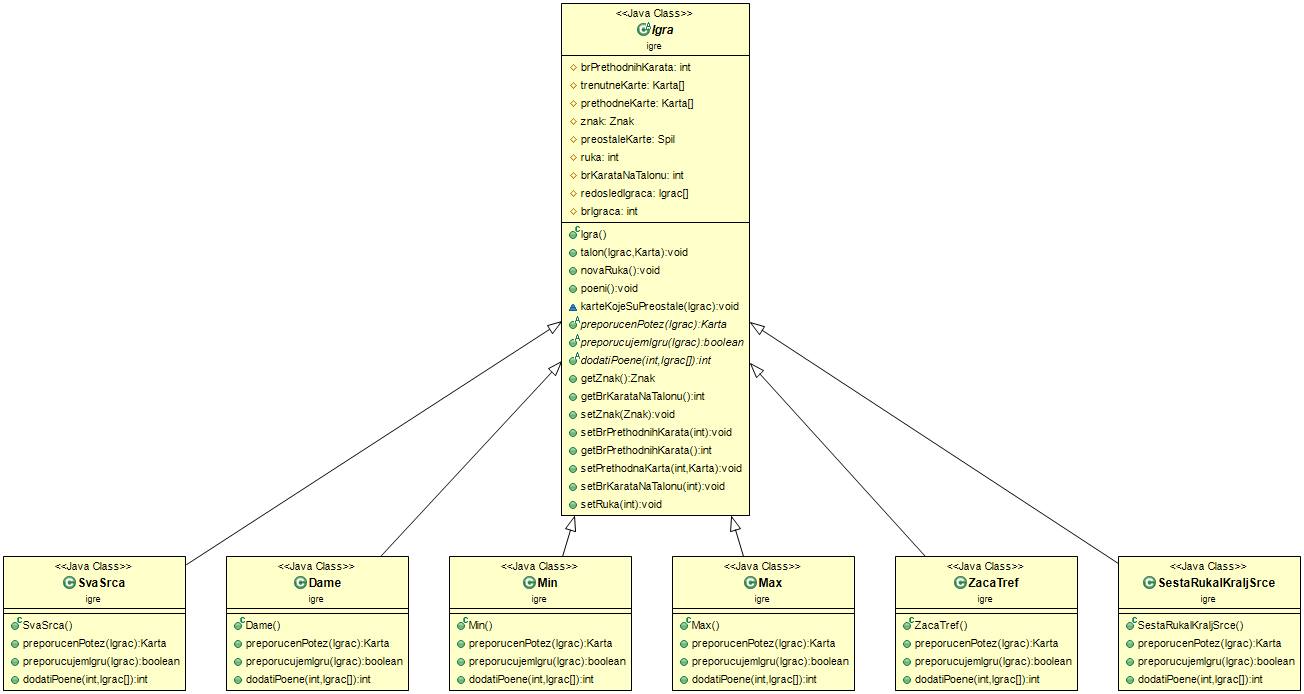
Dve bitne metode koje smo pomenuli *odigrajPotez(Karta)* i *izaberiIgru(int)*. Implementacie ovih metoda u klasi *Takmicar* predstavlja upisivanje podataka koje korisnik zadaje, npr: koju je kartu izbacio na talon ili koju je igru odabrao. Implementacije metoda u klasi *Racunar* su složenije.

Suština metoda *izaberiIgru(int)* klase *Racunar* je da igraču izabere najpogodniju igru na osnovu njegovih karata. Svaka igrač poseduje metod *preporucijemIgru(Igrac)* koji vraća *boolean*, tj. vraća *true* ako podeljene karte igraču ispunjavaju uslov ili, ako je poslednja igra u nizu igara igrača, u suprotnom vraća *false*. Metod *izaberiIgru()* prolazi kroz niz igara koje su poređane redom po težini da se uslovi ispune u opadajućem poretku *SvaSrca, Max, Dame, ZacaTref, SestaRukaIKraljSrce, Min* i bira najpogodniju igru.

Metod *odigrajPotez(Karta)* klase *Racunar* određuje najpovoljniju kartu koju igrač treba da izbaci pozivanjem metoda *preporuciPotez(Karta)* trenutne igre koja je odabrana. Za svaku igru je definisana posebna implementacija ovog metoda, a svaki od njih vraća kartu koja ispunjava uslove kojim je metod definisan.

Ovim podacima i metodama opisana je nadklasa Igrač.

* 1. IGRA



Slika 6 - UML dijagram klase Igra kao i njenih podklasa

Klasa *Igrac* je takođe apstraktna klasa koju nasleđuju klase svih igara navedenih u UML-u. Ova klasa sadrži sve potrebne podatke o trenutnoj igri kao što su: znak trenutne ruke, karte koje se trenutno nalaze na talonu, igrače koji su izbacivali karte i pamti sve prethodne karte koje su prošle tokom igre i metodom *karteKojeSuPreostale(Igrac)* dobija karte koje su uključene u igru odstranjivanjem karata koje su prošle i onih koje se nalaze u ruci igrača.

Dve bitne metode su *preporucujemPotez(Igrac)* i *preporucijemIgru(Igrac)* ukratko su opisane u prethodnom poglavlju.

Detaljni opis svake igre pojedinačno:

* + 1. Sva srca

Igra Sva srca je prva igra u nizu igara. Cilj igre je pokupiti sve karte u znaku srce ili ne pokupiti ni jednu. Ukoliko igrač sakupi sva srca, osvaja -8 poena u suprotnom, svako odneto srce donosi 1 poen. Da bi igrač odneo sva srca potrebno je da ima najjace karte u znaku srce, tačnije potrebno je da ima 1 srce, K srce ili Q srce i makar još jednu kartu u znaku srce. Naravno ovo nije siguran uslov da će osvojiti -8 poena, ali je dovoljan da igrač pokuša i izabere igru.

Kada je igrač odabrao igru potrebno je postaviti određene uslove kako bi igrač odigrao potez. Posmatramo dve vrste slučajeva: igrač prvi izbacuje kartu ili igrač prati znak sa talona i sva srca su u igri ili neka srca nedostaju. U slučaju kada igrač prvi izbacuje kartu na talon i sva srca su u igri igrač prvo proverava da li ima odgovarajuće karte da pokupi sva srca i dobije -8 poena. Ukoliko je taj uslov zadovoljen on izbacuje najjaču kartu u znaku srce ili u koliko ima kombinaciju 1 srce Q srce izbaciće prvo Q srce. U suprotnom igrač ce pokušati da se podvuče sa najmanjom kartom. Ukoliko igrač prvi izbacuje kartu i neko srce nedostaje, igrač proverava da li je on pokupio sva srca koja nedostaju i ako je tako nastavlja da izbacuje najveću kartu u znaku srce ili ako ih vise nema izbacuje najveću kartu koju ima u ruci. Ako igrač nije pokupio sva srca koja su prošla on se podvlači izbacivanjem najmanje karte od karata koje ima. Slično se dešava i kada igrač prati znak talona. On prvo proverava da li njegove karte zadovoljavaju uslov da ponese sva srca i u zavisnosti od toga izbacuje najveću kartu u tom znaku ili se podvači. Kada igrač treba da isprati znak talona, a nema karte u tom znaku tada on u zavisnosti od toga da li planira da pokupi sva srca ili da izbegne skupljanje srca, izbacuje najmanju kartu u onom znaku u kome ima najmanje karata ili izbacuje najveću kartu u onom znaku u kome ima najmanje karata.

* + 1. Maksimum

U igri Maksimum cilj je da igrač odnese što više ruku. Svaka odneta ruka donosi -1 poen. Potreban uslov da igrač odabere ovu igru je da može da osvoji najmanje -3 poena. Tačnije, potrebno je da igrač ima 3 najjače karte kako bi izabrao igru. Kada je odabere igrač najjačim kartama uzima talon sve dok u ruci ima najjače karte u igri, u slučaju da nema izbacuje najmanju kartu i čeka da neka od njegovih karata bude najjača karta u igri.

* + 1. Dame NE

Cilj igre Dame NE je da se ne pokupi ni jedna dama (Q), svaka odneta dama donosi 2 poena. Igrač bira igru u koliko nema ni jednu damu, nema samo karte veće od dame (Q) u nekom znaku ili ako ima damu u nekom znaku potrebno je da ima i 2 manje karte u tom znaku. Kada je igra Dame NE odabrana igrač se podvlači svaki put kad je to moguće i izbacuje najmanje karte u koliko prvi treba da odigra potez, a kada dođe u situaciju da nema karata u znaku koji se igra on proverava da li ima damu, ako ima izbacuje je u suprotnom izbacuje kartu koja mu je najmanje potrebna.

* + 1. Žaca tref

Igra Žaca tref donosi 8 poena igraču koji odnese žandara u trefu. Igrač da bi izabrao ovu igru potrebno je da u slučaju ako ima žandara (J) u trefu ima barem još 2 karte manje od žandara (J), ima u nekom znaku samo jednu kartu manju od 11 kako bi mogao da se podvuče ili mu nedostaje neki znak (nema ni jednu kartu u nekom znaku) , a ima kartu da se podvuče, u suprotnom (nema žandara u trefu) ni u jednom znaku nema veću kartu od žandara ili ako ima ima i pored nje još 2 manje da se podvuče. Kada je igra odabrana, igrač se trudi da izbacuje što manje karte, a ako nema kartu u znaku talona, a ima žandara u trefu izbacuje ga.

* + 1. Šesta ruka i kralj srce

U igri Šesta ruka i kralj srce je poenta ne odneti kralja srce i ne odneti šestu ruku, oba donose po 4 poena. Igrač na isti način bira igru kao i u igri Žandar tref samo što je u pitanju karta kralj srce. A na potpuno sličan način i igra poteze, s' tim što ne vodi računa o karti žandar tref, već o karti kralj srce.

* + 1. Minimum

Igra Minimum je poslednja igra u nizu igara svakog igrača. Ako je neka igra poslednja igra u nizu i ni jedan uslov od prethodnih igara nije ispunjen, ta igra je podrazumevana igra i ona će automatski biti odabrana bez ikakvih uslova. Kada je igra Minimum odabrana igrač izbacuje najmanje karte preostale u igri koje ima u ruci.

* 1. ENGINE

Klasa *Engine* je jedina klasa koja direktno komunicira sa klasom GUI. Iz tog razloga se u klasi Engine nalaze sve funkcionalnosti i kompletna logika igrice. U ovoj klasi su zapisane najbitniji podaci kako bi se igrica odvijala pravilno. Vodi računa o tome ko je trenutno na potezu, da li je na potezu računar ili takmičar, tj. korisnik aplikacije *( koIgra()* i *next()* ), kada je igra završena ( *krajIgre()* ) ili kada je partija završena ( *kraj()* ) i ko je pobednik ( *pobednik()* ), o broju poena igrača *( brojPoena()* ) i o nagradama koje igrač moze da osvoji ( *nagrada()* ), deli karte ( *podeliKarta()* ), bira igru ( *birajIgru( NazivIgara )*) , odigrava potez ( *odigrajPotez( int )* ) i sve to povezuje u jednu celinu.

* 1. GUI

Klasa *GUI*-a nasleđuje klasu *JFrame* te na ekranu grafički prikazuje izgled aplikacije*.* Klasa *GUI* zapravo oslikava sve ono što se dešava u *Engine* klasi i prateći instrukcije klase *Engine* prikazuje grafičke elemenate. Neki od elemenata koji su korišćeni su: *JPanel, JScrollPane, JOptionPane, JTextArea, Timer.* Instanciranjem klase *GUI* pravi se nit koja pokreće prozor aplikacije, a sam izgled tog prozora definiše se u metodi *izgled()*.

Prozor je podeljen na 5 dela standardnim *BorderLayout*-om. Centralni deo predstavlja talon igre. Na levoj strani se nalazi *Igrac1* i niz njegovih igara i broj poena, kao i *JtextArea* polje. Na gornjoj strani je *Igrac2* sa svojim nizom igara i brojem poena, ikona tekuće igre kao i njen kratak opis. Na desnoj strani nalazi se *Igrac3* i njegov niz igara i broj poena. Na donjoj strani ekrana nalaze se karte korisnika aplikacije, njegov niz igara, broj poena i dugme za izlazak iz partije.

Metod *init()* uvek dovodi igriču na početno stanje. Metod *osveziTalon()* služi za osvežavanje prikaza trenutnih karata sa talona na ekranu. Metod *osveziMojeKarte()* osvežava ikone karata koje su dodeljene korisniku aplikacije. Metod *actionPerformedKarta(ActionEvent)* odigrava potez koji je korisik odabrao i pokreće *Timer* pomoću klase *javax.swing.Timer.* *Timer* poziva metodu *odigrajPotez()* kako bi preostala 3 računara odigrala svoje poteze pozivajući metode klase *Engine*. Ove dve metode *actionPerformedKarta(ActionEvent) i odigrajPotez()* po potrebi pozivaju metod *krajIgre()* kada ih o tome obavesti *Engine.* Metod *krajIgre()* pokreće *Timer* ako je red na računara da odabere igru, a kada je korisnik na potezu očekuje događaj klika na neku od ikonica igara koje su korisniku dostupne. U koliko korisnik nema više igara *Engine* javlja *GUI*-u da pozove metod *kraj()* i obavesti korisnika da je partija završena, prikazuje rang listu i najbolje rezultate svih igrača.

* 1. OSTALE KLASE
     1. Karta

Klasom *Karta* opisani su podaci karte: znak i broj.

* + 1. Špil

U klasi Spil definiše se niz od trideset **i** dve kate koji predstavlja špil karata. Postoji **i** definicija metoda *promesajKarte()* koji random bira jednu po jednu kartu iz špila dok ne ostane ni jedna **i** formira novi promešani špil karata iz kojeg se redom dele karte igračima.

* + 1. Znak

Klasa *Znak* je numerički podatak **i** njom su opisani postojeći znakovi karte: *tref, srce, pik* ***i*** *kocka*.

* + 1. Naziv igara

Klasa *NazivIgara* je numerički podatak i njom su opisani nazivi svih postojećih igara*: Max, Min, SvaSrca, Dame, ZacaTref i SestaRukaIKraljSrce.*

* + 1. Gui karta

Klasa *GuiKarta* je podklasa klase *JButton* u njoj se pamti indeks dugmeta u nekom nizu.

* + 1. Test

*Test* klasa sadrži glavni metod *main(String[])* u kome se instancira klasa *G****UI***.

1. Literatura
2. <http://imi.pmf.kg.ac.rs/moodle/course/view.php?id=33>
3. <http://stackoverflow.com/questions/tagged/java>
4. <https://apps.facebook.com/loraonline/?fb_source=bookmark&ref=bookmarks&count=0&fb_bmpos=_0>