

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 364

MREŽNA KARTAŠKA IGRA NA PLATFORMI WINDOWS

Dominik Matijaca

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 364

MREŽNA KARTAŠKA IGRA NA PLATFORMI WINDOWS

Dominik Matijaca

Zagreb, lipanj 2022.

Zagreb, 11. ožujka 2022.

ZAVRŠNI ZADATAK br. 364

Pristupnik: **Dominik Matijaca (0036524568)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: prof. dr. sc. Krešimir Fertalj

Zadatak: **Mrežna kartaška igra na platformi Windows**

Opis zadatka:

Razviti aplikaciju za kartašku igru Belot za više igrača putem mreže. Klijentski dio implementirati na platformi Windows koristeći radni okvir Avalonia. Implementirati funkcionalnosti dijeljenja, zvanja, igranja i bodovanja prema pravilima za varijante igre s različitim brojem igrača. Također ugraditi upravljanje korisničkim računima, povezivanje igrača te snimanje i pregled igara.

Rok za predaju rada: 10. lipnja 2022.

Sadržaj

Uvod	1
1. Analiza problemskog područja	2
1.1. Playtoy Belot	2
1.2. Bela Online (bela.fun)	4
1.3. Facebook BeloT	5
1.4. Zaključak analize	7
2. Specifikacija zahtjeva	8
2.1. Funkcionalni zahtjevi	8
2.2. Nefunkcionalni zahtjevi	8
3. Arhitektura sustava	9
3.1. Poslužitelj	9
3.1.1. Korisnički računi	9
3.1.2. Logika igre	11
3.1.3. Mrežna komunikacija	13
3.2. Klijent	14
3.2.1. Mrežna komunikacija	14
3.2.2. Modeli	15
3.2.3. Grafičko sučelje	16
3.3. Jezik mrežne komunikacije	19
3.3.1. Primjer komunikacije	22
4. Funkcionalnosti aplikacije	23
4.1. Prijava korisnika u sustav	23
4.2. Pregled dostupnih soba	24
4.2.1. Stvaranje vlastite sobe	25
4.3. Čekanje ostalih igrača u sobi	26

4.4.	Partija u tijeku.....	27
4.4.1.	Određivanje aduta.....	28
4.4.2.	Licitiranje zvanja	29
4.4.3.	Kartanje	32
4.5.	Ostale i općenite funkcionalnosti	35
4.5.1.	Postavke.....	35
4.5.2.	Korisnički profili	36
4.5.3.	Gledanje partija.....	37
5.	Korištene tehnologije i alati.....	39
5.1.	.NET 6	39
5.2.	Avalonia	39
5.3.	Humanizer	39
5.4.	NetCoreAudio.....	39
5.5.	Visual Studio 2022	40
5.6.	Git i GitHub Desktop.....	40
	Zaključak	41
	Literatura	42
	Sažetak.....	43
	Summary.....	44

Uvod

Kartaška igra Belot popularna je u sjevernoj Hrvatskoj i među mladim i među starijim generacijama. U doba pandemije i nemogućnosti kontakta uživo s najmilijim prijateljima kartašima, teško je pronaći prikladan način za zadovoljiti potrebu za kartanjem. Postojeće popularne aplikacije ruše se radi preopterećenja, dok ostale manje poznate aplikacije nisu dovoljno prikladne. Sasvim je logično onda mladom studentu FER-a iskoristiti priliku pa napisati aplikaciju za privatno kartanje s kolegama i usput naučiti o izradi mrežne aplikacije iz temelja.

Tema ovog završnog rada je izrada aplikacije za kartašku igru Belot putem mreže imena Aleb. Upotrijebljena mrežna arhitektura jednostavan je odnos poslužitelja i više klijenata. Na početku završnog rada definirani su funkcionalni i nefunkcionalni zahtjevi kakvi se očekuju od jedne takve moderne aplikacije. U nadolazećim poglavljima opisana je arhitektura sustava i funkcionalnosti izrađene aplikacije te su navedene tehnologije korištene pri izradi aplikacije.

1. Analiza problemskog područja

Prije početka izrade vlastitog programskog rješenja, prvo je potrebno proučiti postojeća rješenja koja su javno dostupna na tržištu. Na hrvatskom tržištu postoji malen broj aplikacija koji implementiraju kartašku igru Belot, pa ćemo izdvojiti tri popularne aplikacije: Playtoy Belot, Bela Online, i Facebook BeloT. Iznijeti ćemo neke osnovne podatke i analizirati funkcionalnosti koje pružaju ove aplikacije.

1.1. Playtoy Belot

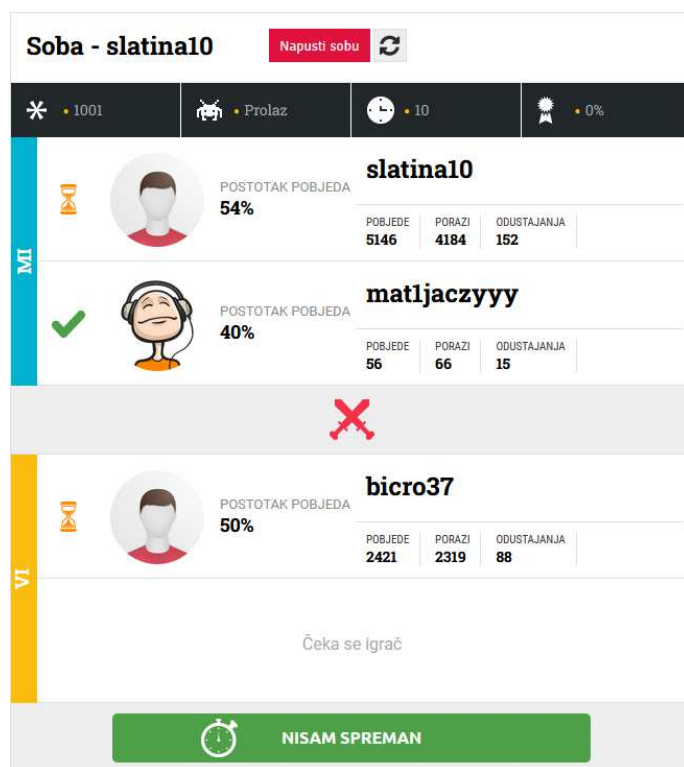
Playtoy Belot [1] uvjerljivo je najpopularnija aplikacija za kartanje Belota. Aplikacija je dostupna samo unutar web-preglednika i bilježi preko 87 milijuna odigranih partija u malo više od desetljeća svog postojanja. Inicijalno je bila implementirana u Adobe Flashu, no tijekom 2016. godine prolazi kroz reimplementaciju u HTML5. Dostupna je na web-adresi <https://playtoy.tportal.hr/>.

Uz Belot, Playtoy nudi i razne druge kartaške i društvene igre, ali u ovoj analizi ćemo se usredotočiti samo na Belot. Na slici 1.1 prikazan je ulaz u igru Belot na naslovnoj stranici aplikacije.



Slika 1.1 Ulaz u igru Playtoy Belot

Za korištenje aplikacije je obavezna registracija korisničkog profila. Aplikacija korisnicima nudi mogućnost kreiranja sobe (Slika 1.2) s podesivom privatnosti, parametrima igre i filterima igrača. U sobu je moguće direktno pozivati prijatelje ili općenite korisnike iz popisa svih trenutno aktivnih igrača.



Slika 1.2 Prikaz sobe u aplikacije Playtoy Belot

Nakon što se četiri igrača okupe unutar sobe, igra može početi (Slika 1.3). U igri su implementirana standardna domaća pravila igre bez pretjeranih odstupanja. Omogućeno je dopisivanje igrača u trenutnoj sobi tijekom čekanja za početak i tijekom igre. Aplikacija podržava zabavne izazove vezane uz događanja u igri. Također prati i broj pobjeda, poraza i odustajanja korisnika (Slika 1.4).



Slika 1.3 Prikaz igre u aplikaciji Playtoy Belot

Multiplayer izazovi (BETA)

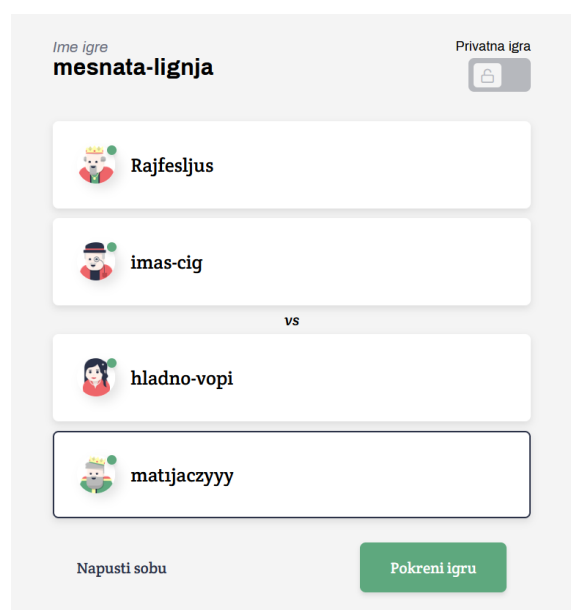


Slika 1.4 Statistike korisničkog profila u aplikaciji Playtoy Belot

1.2. Bela Online (bela.fun)

Bela Online [2] je najmlađa aplikacija za kartanje Belota, nastala u vrijeme početka pandemije koronavirusa. Aplikacija je dostupna unutar web-preglednika te na platformama iOS i Android. Na Google Play trgovini bilježi preko 10 tisuća instalacija. Aplikacija je primarno usredotočena na iskustvo igranja na mobilnim uređajima s modernim dizajnom.

Za korištenje aplikacije nije potrebno stvoriti korisnički račun, te ne pamti korisničke podatke. Pod lako izmjenjivim korisničkim imenom moguće je stvoriti sobu (Slika 1.5) s podesivim parametrima igre. U sobe se mogu dodati i zamjenski igrači kojima upravlja računalo umjesto nedostajućih igrača.



Slika 1.5 Prikaz sobe u aplikaciji Bela Online

Nakon što se četiri igrača okupe unutar sobe, igra može početi (Slika 1.6). U igri su implementirana standardna domaća pravila [5], s jednom iznimkom: igrač ne treba izabrati karte za zvanje dodatnih bodova, već igra samostalno računa zvanja. Korisniku je dana mogućnost objave ili skrivanja zvanja. Omogućeno je izražavanje među igračima pomoću preddefiniranih reakcija baziranih na emotikonima.



Slika 1.6 Prikaz igre u aplikaciji Bela Online

1.3. Facebook BeloT

Facebook BeloT [3] je aplikacija koja je do ukidanja Adobe Flasha bila dostupna na društvenoj mreži Facebook. Kako danas aplikacija više nije dostupna, ovdje nećemo ulaziti u detaljnu analizu već ćemo samo spomenuti neke karakteristike po kojima odstupa od ranije spomenutih rješenja.



Slika 1.7 Prikaz glavne stranice u aplikaciji Facebook BeloT [4]

Zahvaljujući uskoj integraciji s korisničkim profilom na Facebooku, povezivanje s prijateljima u igru je jako jednostavno. Aplikacija nudi sustav osvajanja zlatnika koji se koristi kao ulog u igri. Vanjski gledatelji mogu uživo promatrati igre. U aplikaciji se mogu održavati i turniri.



Slika 1.8 Prikaz igre u aplikaciji Facebook BeloT

1.4. Zaključak analize

Nakon analize tri postojeće aplikacije, možemo utvrditi postojanje nekoliko osnovnih zajedničkih funkcionalnosti:

- korisnički računi s praćenjem statistike
- sustav soba za okupljanje igrača prije početka igre
- zajednička osnovna pravila kartaške igre Belot
- komuniciranje između igrača direktno u igri.

Aplikaciju Bela Online možemo dodatno istaknuti jer nudi najbolji izgled sučelja, prateći moderni minimalistički dizajn koji je pregledan i nije prenapučen.

2. Specifikacija zahtjeva

Prije korištenja aplikacije nužno je otvoriti korisnički račun, stoga aplikacija ima samo jednu vrstu korisnika – registrirani korisnici. Kako bi korisnik mogao pristupiti sljedećim funkcionalnostima mora prethodno obaviti prijavu u aplikaciju.

2.1. Funkcionalni zahtjevi

- registracija i prijava u aplikaciju
- stvaranje soba s podesivom privatnosti i parametrima igre
- potpuno implementirana sva osnovna pravila kartaške igre Belot s četiri igrača [5]
 - iznimka osnovnim pravilima: mogućnost zvanja „na talon“ za posljednjeg igrača
 - pregled svih karata u prethodnom krugu igre („što je tko imao“)
- praćenje vlastite statistike za korisnika
- snimanje i pregled prethodno odigranih igara korisnika
- gledanje tuđe igre u tijeku uživo.

2.2. Nefunkcionalni zahtjevi

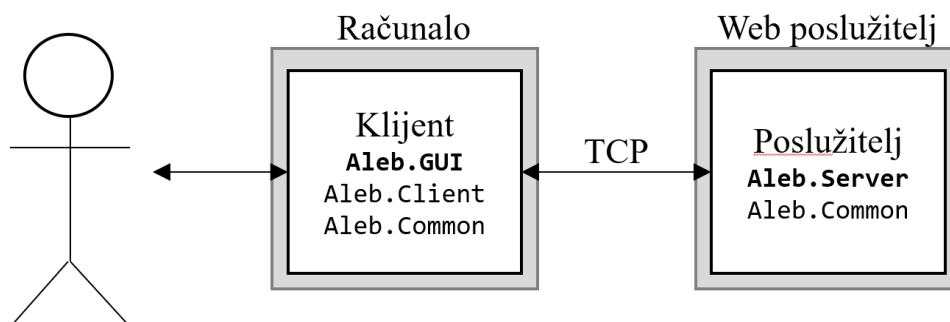
Aplikacija mora ispravno raditi pri istovremenom korištenju više korisnika i više soba bez pretjeranog čekanja kako bi se osiguralo nesmetano kretanje igre u stvarnom vremenu.

Aplikacija mora biti otporna na pogreške u slučaju gubitka i ponovne uspostave mrežne veze te u predviđenom korištenju (ne smije se zaključati u stanju iz kojeg ne može napredovati). Aplikacija mora omogućiti ponovno povezivanje na igru u slučaju prekinute veze.

Grafičko sučelje aplikacije mora biti pregledno, jednostavno i intuitivno za korištenje kako bi svi korisnici mogli nesmetano koristiti aplikaciju. Grafičko sučelje se mora prilagoditi raznim širinama i visinama (responzivno) uz mogućnost prikaza preko cijelog zaslona.

3. Arhitektura sustava

Arhitektura sustava je izrađena iz temelja za potrebe aplikacije. Na površini, aplikacija je podijeljena na dvije glavne izvršne komponente prikazane na slici 3.1: **glavni poslužitelj** (`Aleb.Server`) i **korisničko grafičko sučelje** (`Aleb.GUI`).



Slika 3.1 Dijagram ugradnje

Jako velik dio mrežne komunikacije implementiran je u zajedničkoj biblioteci `Aleb.Common` koju koriste i poslužitelj i klijent, pa možemo na nju gledati kao komponentu „ljepila“ između poslužitelja i klijenta. Komunikacija se odvija direktno putem otvorenog TCP kanala, a jezik komunikacije je također dizajniran iz temelja kako bi ispunio potrebe aplikacije.

3.1. Poslužitelj

Gotovo sve radnje koje korisnik izvršava u aplikaciji događaju se na poslužitelju. Poslužitelj ima dvije glavne zadaće: vođenje i snimanje korisničkih računa i podataka povezanih s njima, te vođenje potpune logike same kartaške igre i koordiniranje igre među korisnicima.

3.1.1. Korisnički računi

Jedna instanca razreda `User` predstavlja jednog registriranog korisnika. Svi podatci nekog korisnika su zapisani u njegovoj instanci, uključujući korisničko ime, lozinku, statistike, i povijest partija. Ako je korisnik trenutno povezan na poslužitelj, onda će putem te instance biti moguće direktno komunicirati s korisnikom tijekom igre. Poslužitelj održava *pool* svih registriranih korisnika u memoriji, uključujući i trenutno nepovezane korisnike.

Statistike korisnika su jednostavne varijable u razredu `User`, dok je povijest partija zapisana kao popis instanci razreda `RecordedMatch` od kojih svaka sadrži naziv sobe, korisnička imena četiri korisnika koji su odigrali partiju, krajnje stanje bodova, datum i vrijeme odigrane partije, i zapisanu svu mrežnu komunikaciju koja predstavlja tijek igre iz perspektive neutralnog gledatelja.

Svaka promjena podatka na instanci korisnika ili dodavanje novoregistriranog korisnika u *pool* pokreće snimanje sadržaja *poola* u binarnu datoteku na disku za što je zadužen statički razred `Persistent`. Taj razred je također zadužen za čitanje te datoteke pri pokretanju poslužitelja kako bi se rekonstruirao glavni *pool* korisnika u memoriji. Sustav je zamišljen tako da je poslužitelj uvijek dostupan, a korisnici se na njega povezuju po potrebi, tako da se učestala ponovna pokretanja poslužitelja ne bi smjela događati.

Razredi `User`, `RecordedMatch` i `Persistent` prikazani su sljedećom slikom.



Slika 3.2 Dijagram razreda `User`, `RecordedMatch` i `Persistent`

3.1.2. Logika igre

Sva logika igre implementirana je na poslužitelju. Kada klijenti žele izvršiti neku radnju u igri, poslužitelju šalju zahtjev za željenom radnjom. Poslužitelj provjerava radnju i ako je legalna, izvršava ju i svim upućenim korisnicima (primjerice, ostalim igračima u igri) šalje obavijest o izvršenoj radnji. Time se cijela igra odvija na poslužitelju, a klijenti se ponašaju kao nekakvi „daljinski upravljači“ s uvidom u aktualnu situaciju.

Ovdje nećemo detaljno ulaziti u pravila igre već ćemo ih prikazati kasnije kroz funkcionalnosti aplikacije. Razred `Room` predstavlja jednu sobu u kojoj se može odvijati partija. Sobe stvaraju korisnici, i mogu im se pridruživati i izlaziti iz njih dok se ne skupe četiri korisnika koja žele međusobno odigrati partiju. Tada se može započeti partija koju predstavlja razred `Game`.

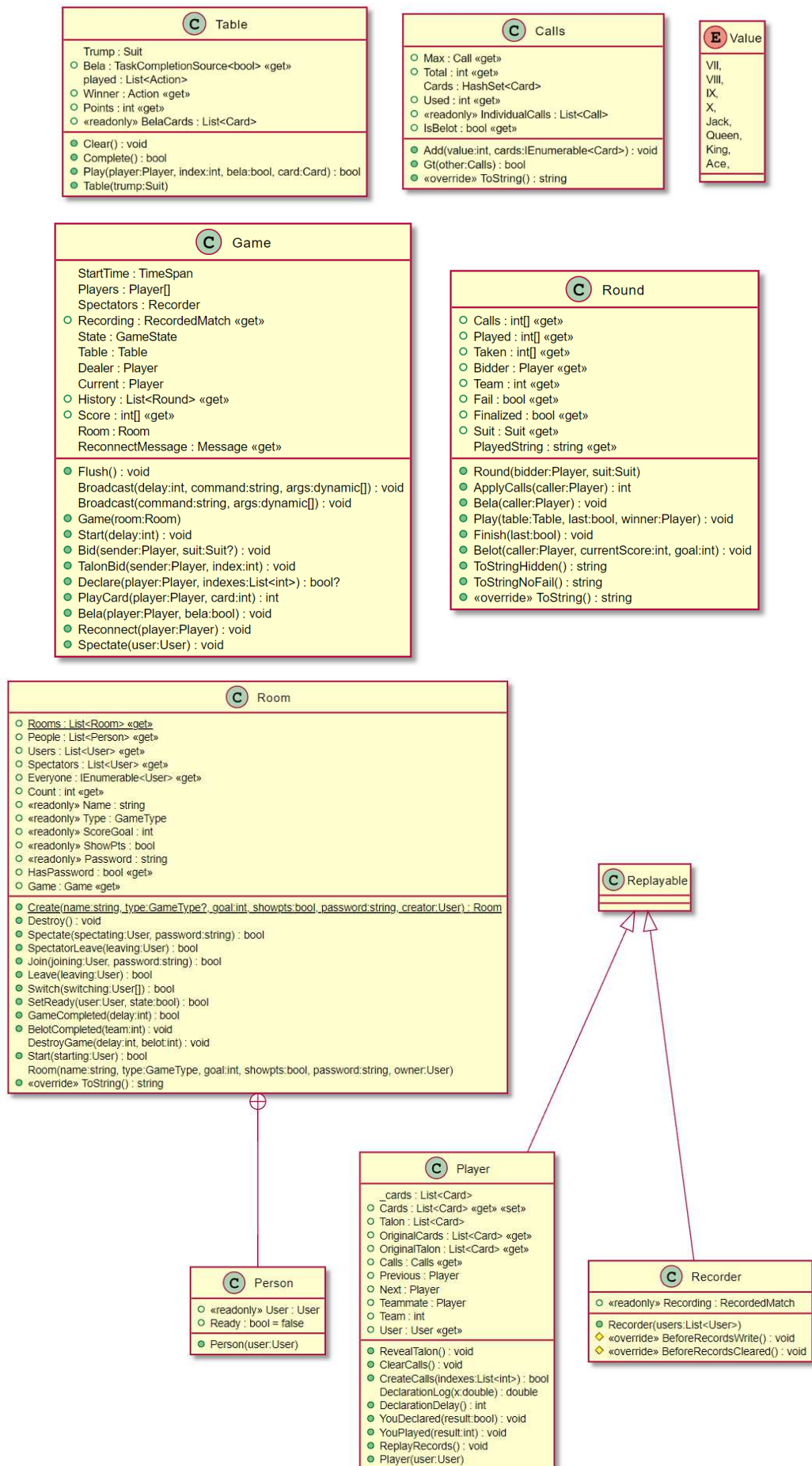
Korisnici se promoviraju u igrače koje predstavlja razred `Player` čija je glavna svrha pamtiti mrežnu komunikaciju u slučaju gubitka veze i ponovnog spajanja korisnika, koje se vrši tako da se jednostavno ponovno pošalju sve mrežne poruke od početka trenutne runde. Slično postoji razred `Recorder` koji snima igru iz neutralne perspektive. Služi za prijenos igre gledateljima i snimanje potpunog tijeka igre. Unutar partije `Game`, pravila i tijek igre implementirani su uz pomoć razreda `Card`, `Calls`, `Table` i `Round`.

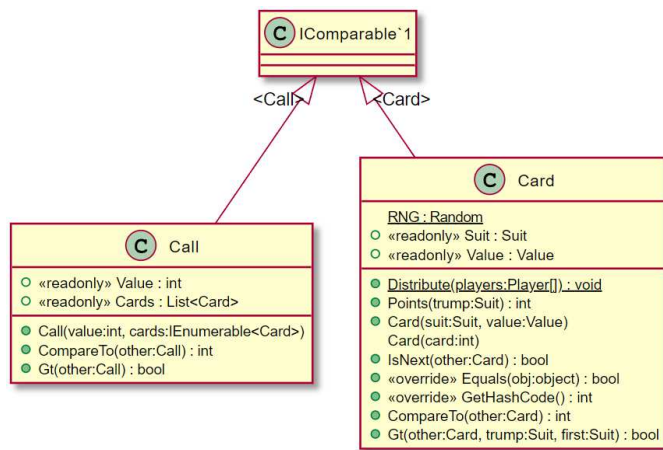
Razred `Card` predstavlja jednu kartu koja se naći igraču u ruci, talonu ili na stolu. Karte se dijele tako da se od poredanog skupa od 32 karte bira slučajna permutacija s 32 radnje zamjene mjesta karata. Takva permutacija se slijedno dijeli igračima.

Razred `Calls` predstavlja skup zvanja koje je jedan igrač odlučio licitirati i može se direktno prirodno uspoređivati među igračima.

Razred `Table` predstavlja stol na kojeg igrači bacaju karte i on vrši provjere ispravnosti boje i jačine bačene karte. Na kraju jednog kruga, stol određuje pobjednika i vrijednost štiha i zapisuje bodove u razred `Round`. Taj razred prati trenutno i završno stanje bodova jedne runde igre.

Razredi `Room`, `Game`, `Player`, `Recorder`, `Card`, `Calls`, `Table` i `Round` prikazani su sljedećom slikom.



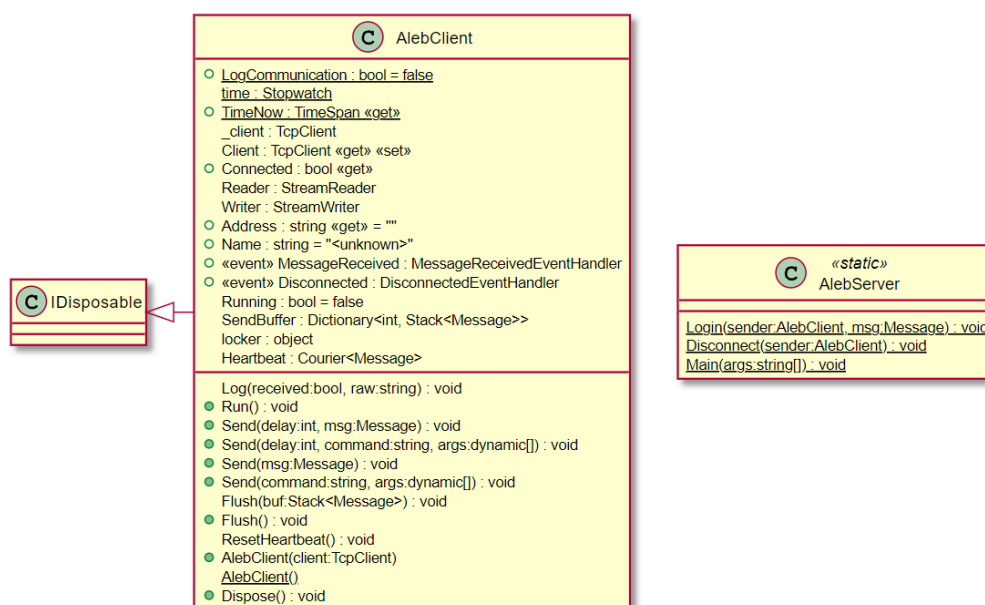


Slika 3.3 Dijagrami razreda Room, Game, Player, Recorder, Card, Calls, Table i Round

3.1.3. Mrežna komunikacija

Poslužitelj pri pokretanju koristi statički razred `AlebServer` kako bi počeo slušati dolazne veze korisnika. Klijenti u trenutku povezivanja stvaraju instancu razreda `AlebClient` i koriste ju da se povežu na poslužitelj. Kada poslužitelj prihvati vezu, on stvara instancu `AlebClient` za sebe kako bi mu služila za direktnu komunikaciju s tim korisnikom. Nakon što se korisnik prijavi u sustav, ta instanca `AlebClient` se zapisuje u njegovu pripadajuću instancu `User` kako bi sva buduća komunikacija mogla ići direktno putem njene metode `Received`.

Razred `User` prikazan je ranije na slici 3.2, a razredi `AlebServer` i `AlebClient` prikazani su sljedećom slikom.



Slika 3.4 Dijagrami razreda AlebServer i AlebClient

3.2. Klijent

Aplikacija je pisana imajući u vidu potencijalne višestruke implementacije klijenata (buduća mobilna ili web-verzija) pa su odvojeni općeniti razredi za klijente koji se sastoje od mrežnih specifičnosti i modela podataka.

Takav klijent ustvari je minimalna podloga koja implementaciji grafičkog sučelja znatno pojednostavljuje proces komunikacije s poslužiteljem i dohvaćanja podataka s poslužitelja.

3.2.1. Mrežna komunikacija

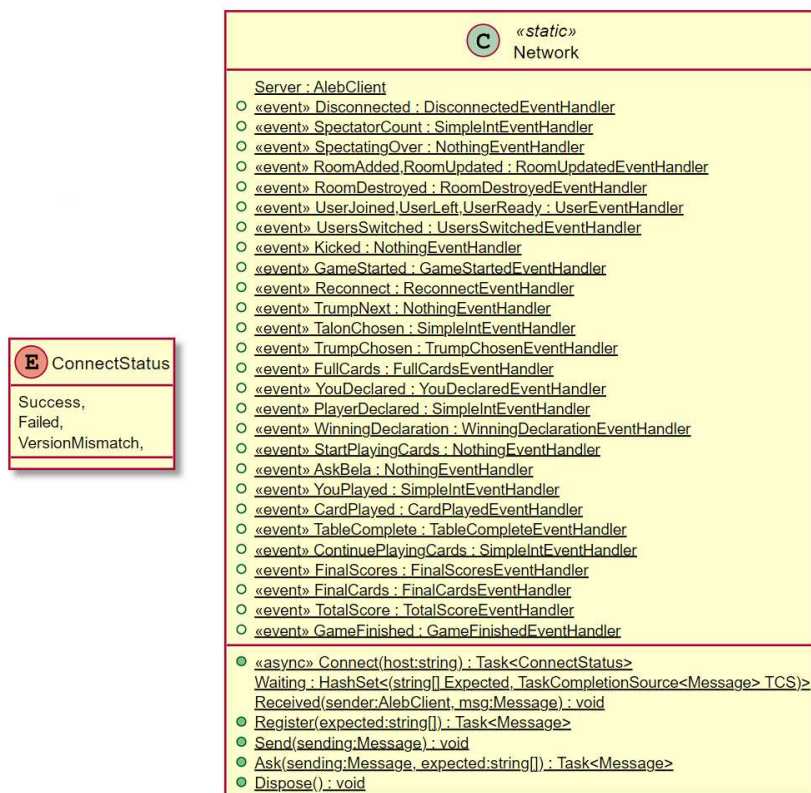
Neovisno o grafičkom sučelju u klijentu su implementirani statički razredi `Network` i `Requests` kako bi implementacija grafičkog sučelja što jednostavnije obavljala komunikaciju s poslužiteljem.

Razred `Network` direktno rukuje komunikacijom s poslužiteljem, sluša obavijesti od poslužitelja, pokreće događaje, i implementira općenito slanje poruka i upita poslužitelju. Događaji su poruke koje poslužitelj šalje direktno klijentima i ne moraju nužno biti direktan odgovor na nekakav upit od istog klijenta.

Razred `Requests` implementira sve poruke i upite koje klijent može samostalno zatražiti od poslužitelja. Također implementira i čekanje na sve valjane odgovore na zatražene upite i njihovo tumačenje.

Razredi `Network` i `Requests` prikazani su sljedećom slikom.

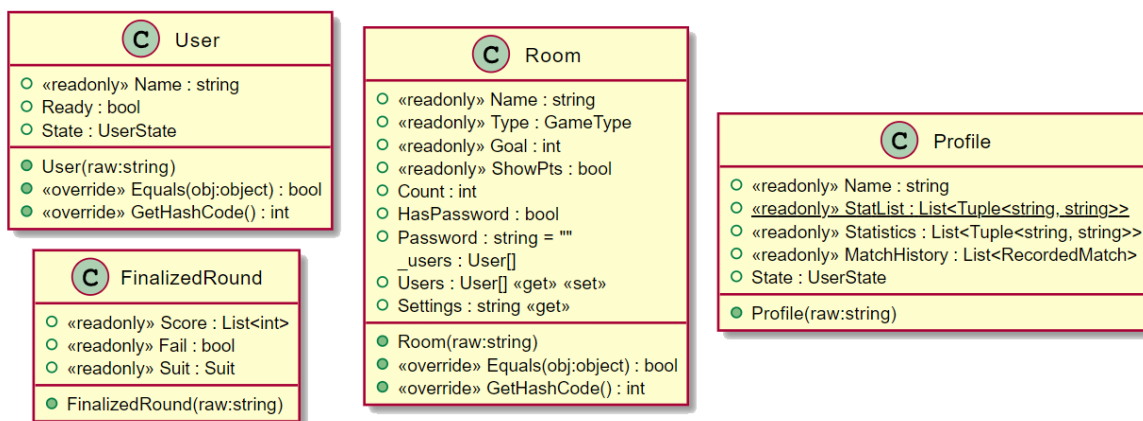




Slika 3.5 Dijagrami razreda Network i Requests

3.2.2. Modeli

Neovisno o grafičkom sučelju u klijentu su implementirani razredi – modeli User, Room, FinalizedRound i Profile (struktura prikazana na slici 3.6). Model User predstavlja podatke vezane uz korisnika koji se nalazi u sobi. Model Room predstavlja podatke vezane uz sobu. Model FinalizedRound predstavlja jednu dovršenu rundu unutar partije. Model Profile predstavlja podatke vezane uz pregled profila korisnika što uključuje statistike i povijest partija.



Slika 3.6 Dijagrami razreda User, Room, FinalizedRound i Profile

3.2.3. Grafičko sučelje

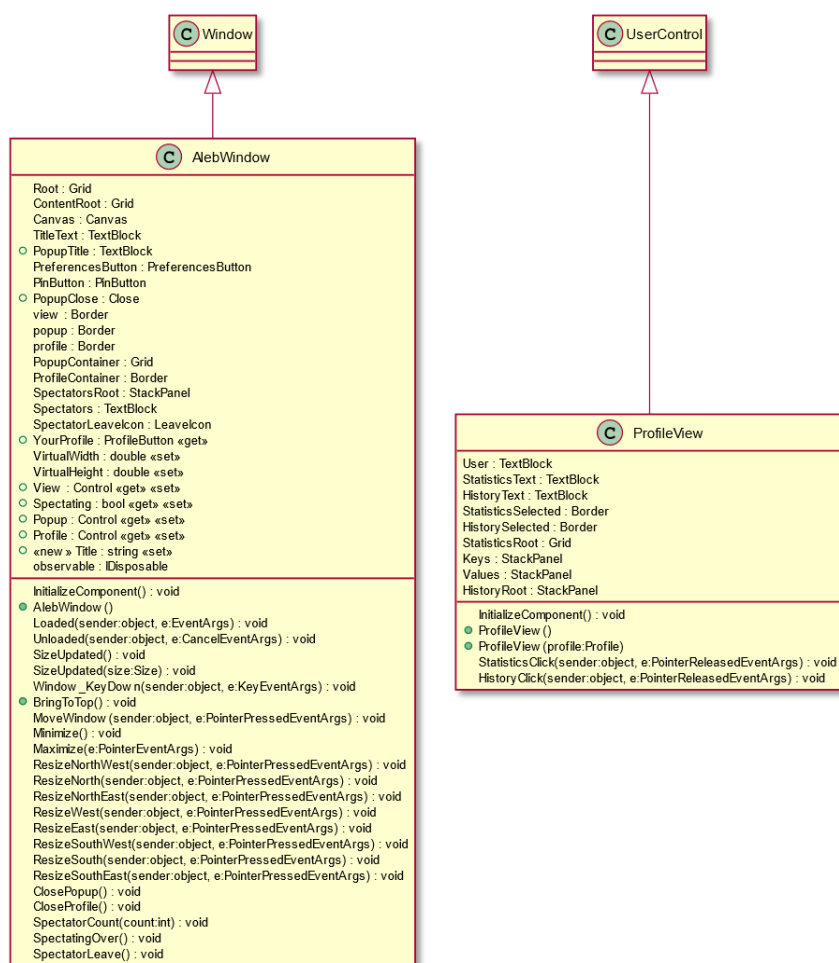
Implementirano grafičko sučelje koristi radni okvir Avalonia i primarno je namijenjeno za platformu Windows, ali se može brzo prilagoditi za ostale operacijske sustave stolnih računala.

Aplikacija je potpuno implementirana na istom prozoru, jednoj instanci razreda AlebWindow. On na sebi sadrži nekoliko predviđenih prostora na koje se postavljaju određeni prikazi: View, Popup, i Profile.

Prostor je općeniti dio prozora u kojeg se mogu proizvoljno učitavati razni pogledi i kontrole, a nudi prednost zajedničke implementacije okruženja u kojem se nalazi (primjerice, obrub skočnog prozora i tipka za njegovo zatvaranje kod Popup prostora).

Profile prostor koji se širi s desne strane aplikacije predviđen je isključivo za komponentu ProfileView.

Razredi AlebWindow i ProfileView prikazani su sljedećom slikom.

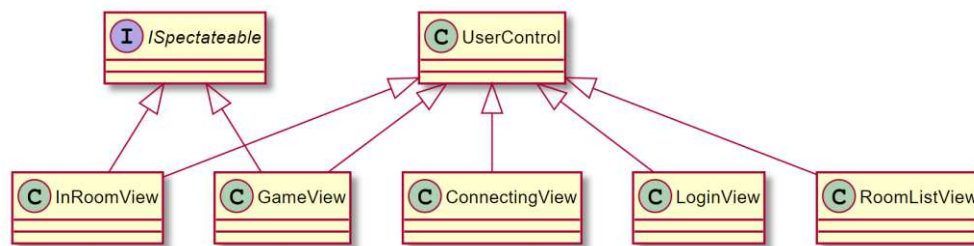


Slika 3.7 Dijagram razreda AlebWindow

3.2.3.1 Prostor View

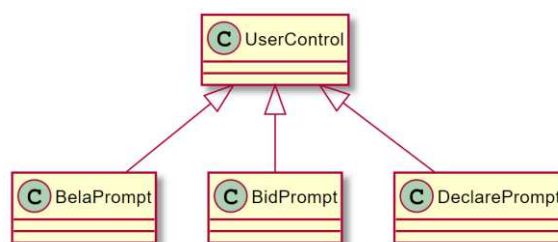
Prostor View je glavni prostor za prikaz sadržaja u aplikaciji. Za njega su previđeni sljedeći Viewovi: `ConnectingView`, `LoginView`, `RoomListView`, `InRoomView`, i `GameView`.

`ConnectingView` prikazan je pri pokretanju aplikacije tijekom pokušaja povezivanja s poslužiteljem. U slučaju greške prikazuje grešku i nudi mogućnost ponovnog pokušaja, a u slučaju uspješne veze nastavlja dalje na `LoginView`. `LoginView` prikazuje formu za prijavu korisnika. `RoomListView` prikazuje popis dostupnih soba i korisniku nudi mogućnost stvaranja vlastite sobe. `InRoomView` prikazuje korisnika u sobi s ostalim korisnicima koji čekaju početak partije. `GameView` prikazuje partiju u tijeku.



Slika 3.8 Dijagram svih Viewova

`GameView` dodatno u sredini ima predviđen prostor za Prompt koji je predviđen za razne upite koji se mogu pojaviti tijekom partije. Za taj prostor predviđeni su `BidPrompt` (upit odabiranja aduta), `DeclarePrompt` (upit licitiranja zvanja) i `BelaPrompt` (upit zvanja bele).

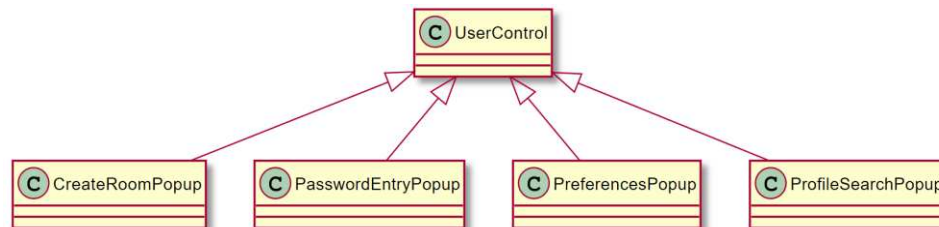


Slika 3.9 Dijagram svih Promptova

3.2.3.2 Prostor Popup

Prostor Popup je glavni prostor za prikaz formi i upita u obliku lažnog skočnog prozora. Za njega su previđeni sljedeći Popupi: `PreferencesPopup`, `CreateRoomPopup`, `PasswordEntryPopup` i `ProfileSearchPopup`.

PreferencesPopup predstavlja skočni prozor s postavkama klijentske aplikacije. CreateRoomPopup predstavlja formu za podešavanje sobe prilikom njenog stvaranja. PasswordEntryPopup predstavlja formu za unos lozinke pri pokušaju ulaska u sobu zaštićenu lozinkom. ProfileSearchPopup predstavlja formu za unos korisničkog imena čiji profil treba dohvatiti.

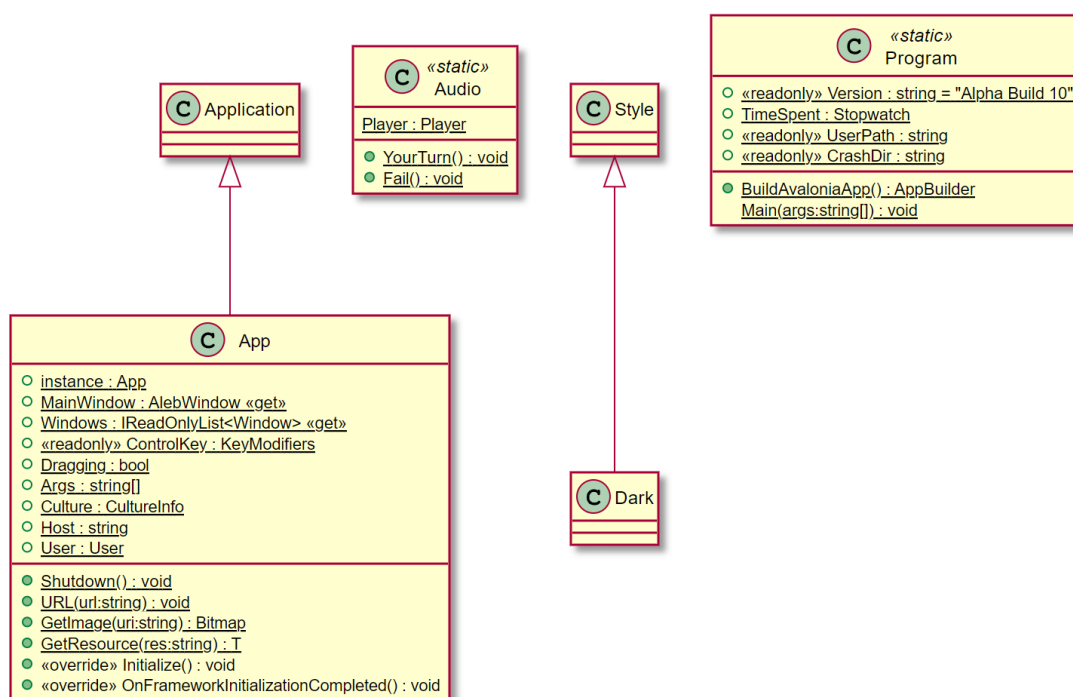


Slika 3.10 Dijagram svih Popupa

3.2.3.3 Ostale osnovne komponente

Razred Program zadužen je za pokretanje aplikacije i postavljanje zapisivanja grešaka. Razred App zadužen je za pokretanje radnog okvira Avalonia i određene statičke globalne varijable i metode. Razred Audio zadužen je za reprodukciju zvučnih obavijesti za dolazak na red u partiji i efekt pada pomoću biblioteke NetCoreAudio. U razredu Dark definirana je tema boja koja se koristi u aplikaciji.

Razredi Program, App, Audio i Dark prikazani su sljedećom slikom.

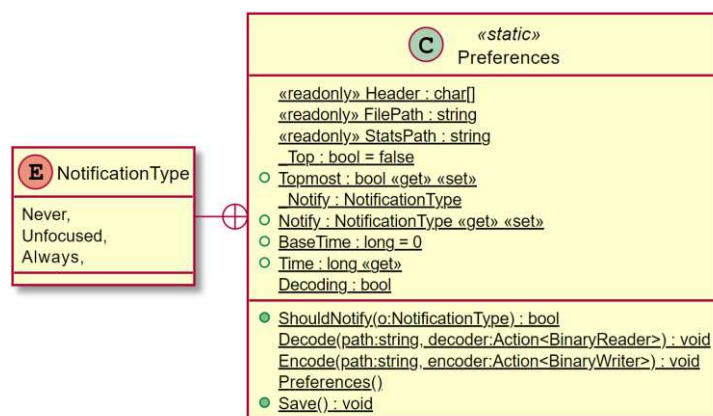


Slika 3.11 Dijagram razreda Program, App, Audio i Dark

3.2.3.4 Postavke

Razred `Preferences` sadrži postavke lokalne klijentu. Slično kao što poslužitelj sprema *pool* korisnika, tako i aplikacija u binarnu datoteku sprema postavke kada ih korisnik promijeni te ih ponovno učitava pri pokretanju aplikacije.

Razred `Preferences` prikazan je sljedećom slikom.



Slika 3.12 Dijagram razreda `Preferences`

3.3. Jezik mrežne komunikacije

Sva komunikacija između poslužitelja i klijenta odvija se slanjem poruka u TCP vezi. Statički razred `Protocol` definira određene konstante koje se tiču mrežne komunikacije. Iako se poruke šalju u obliku stringa, razred `Message` služi kako bi predstavljao jednu poruku. Odvojeni dijelovi poruke odvojeni su znakom točka-zarez (`.,;`). Razred `Courier` služi za slanje poruke s vremenskom odgodom i često se koristi od strane poslužitelja tijekom partije.

Razredi `Protocol`, `Message` i `Courier` prikazani su sljedećom slikom.



Slika 3.13 Dijagram razreda `Protocol`, `Message` i `Courier`

Iako je s ovakvim sustavom moguće poslati bilo kakvu poruku, postoji određen skup poruka koji se upotrebljava u aplikaciji opisan u Tablici 3.1.

Poruka	Pošiljatelj	Opis poruke	Eksplícitan odgovor na
Version	Poslužitelj	Upit o verziji jezika	
Login	Klijent	Pokušaj prijave korisnika	
LoginResult	Poslužitelj	Uspjeh ili neuspjeh prijave	Login
GetRoomList	Klijent	Korisnik želi popis soba	
RoomList	Poslužitelj	Popis soba	GetRoomList
RoomAdded	Poslužitelj	Soba dodana na popis soba	
RoomUpdated	Poslužitelj	Soba s popisa soba ažurirana	
RoomDestroyed	Poslužitelj	Soba izbrisana s popisa soba	
CreateRoom	Klijent	Korisnik želi stvoriti sobu	
RoomCreated	Poslužitelj	Uspješno stvaranje sobe	CreateRoom
RoomFailed	Poslužitelj	Neuspješno stvaranje sobe	CreateRoom
JoinRoom	Klijent	Korisnik se želi pridružiti sobi	
RoomJoined	Poslužitelj	Uspješno pridruživanje u sobu	JoinRoom
RoomJoinFailed	Poslužitelj	Neuspješno pridruživanje u sobu	JoinRoom
SpectateRoom	Klijent	Korisnik želi gledati sobu	
SpectateSuccess	Poslužitelj	Uspješan početak gledanja sobe	SpectateRoom
SpectateFailed	Poslužitelj	Neuspješan početak gledanja sobe	SpectateRoom
SpectatorCount	Poslužitelj	Ažuriran broj gledatelja	
SpectatorLeave	Klijent	Korisnik želi prestati gledati sobu	
LeaveRoom	Klijent	Korisnik želi izaći iz sobe	
UserJoined	Poslužitelj	Drugi korisnik je ušao u sobu	
UserLeft	Poslužitelj	Drugi korisnik je izašao iz sobe	
SetReady	Klijent	Korisnik želi promijeniti spremnost u sobi	
UserReady	Poslužitelj	Drugi korisnik je promijenio spremnost u sobi	
SwitchUsers	Klijent	Vlasnik sobe želi zamijeniti mjesta korisnika u sobi	
UsersSwitched	Poslužitelj	Mjesta korisnika u sobi su zamijenjena	
KickUser	Klijent	Vlasnik sobe želi izbaciti korisnika iz sobe	
Kicked	Poslužitelj	Izbačeni ste iz sobe	
StartGame	Klijent	Vlasnik sobe želi započeti partiju	

GameStarted	Poslužitelj	Početak partije (ili runde)	
Reconnecting	Klijent	Korisnik se želi ponovno povezati u partiju	
Reconnect	Poslužitelj	Podatci za ponovnu izgradnju partije	
TrumpNext	Poslužitelj	Igrač na potezu je rekao dalje na određivanju aduta	
Bid	Klijent	Igrač želi izabrati aduta	
TrumpChosen	Poslužitelj	Adut je izabran	
TalonBid	Klijent	Igrač želi zvati na talon	
TalonChosen	Poslužitelj	Adut je izabran talonom	
FullCards	Poslužitelj	Otkrivanje talona igraču	
Declare	Klijent	Igrač želi licitirati zvanja	
YouDeclared	Poslužitelj	Uspjeh ili neuspjeh licitiranja zvanja	Declare
PlayerDeclared	Poslužitelj	Igrač je licitirao zvanja	
WinningDeclaration	Poslužitelj	Pobjedna zvanja	
StartPlayingCards	Poslužitelj	Početak kartanja	
PlayCard	Klijent	Igrač želi baciti kartu	
AskBela	Poslužitelj	Upit želi li igrač baciti kartu	
Bela	Klijent	Želim ili ne želim baciti kartu	AskBela
YouPlayed	Poslužitelj	Uspjeh ili neuspjeh bacanja karte	PlayCard
CardPlayed	Poslužitelj	Igrač je bacio kartu	
TableComplete	Poslužitelj	Svi su bacili karte u krugu, podatci o bodovima	
ContinuePlayingCards	Poslužitelj	Nastavak kartanja	
FinalScores	Poslužitelj	Krajnji bodovi runde	
FinalCards	Poslužitelj	Karte svih igrača za prikaz "u prethodnoj rundi"	
TotalScore	Poslužitelj	Sveukupni zbroj bodova	
GameFinished	Poslužitelj	Partija dovršena, pobjednička ekipa	
UserProfile	Klijent	Korisnik želi vidjeti pregled profila drugog korisnika	
UserProfileSuccess	Poslužitelj	Uspješno dohvaćanje profila	UserProfile
UserProfileFailed	Poslužitelj	Neuspješno dohvaćanje profila	UserProfile
Heartbeat	Oboje	Održavanje povezanosti veze	

Tablica 3.1 Popis svih valjanih poruka koje se upotrebljavaju u aplikaciji

3.3.1. Primjer komunikacije

U nastavku je prikazan primjer komunikacije iz perspektive klijenta pri čemu su neki retci obrisani radi skraćanja i preglednosti.

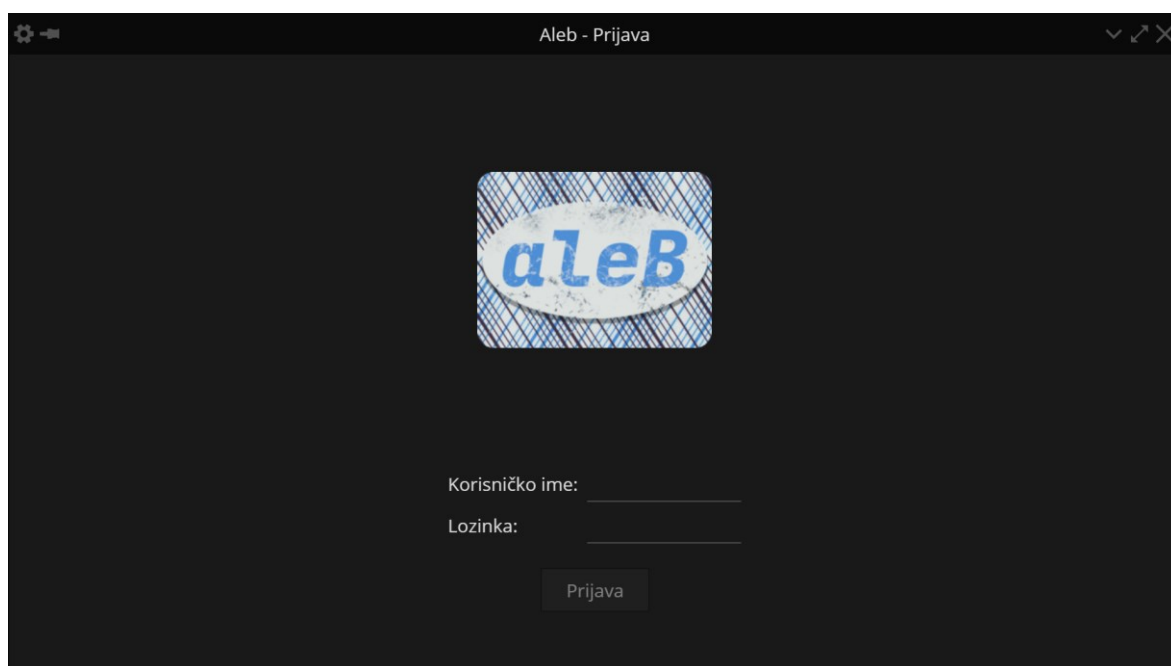
```
0:00.1103875 [RECV] > Version;9
0:10.1781196 [SEND] > Login;mat1;qweasdzxc
0:10.1997077 [RECV] > LoginResult;Idle
0:10.2152394 [SEND] > GetRoomList
0:10.2307923 [RECV] > RoomList
0:23.6108050 [SEND] > CreateRoom;FERovci;Prolaz;1001;True;
0:23.6188175 [RECV] > SpectatorCount;0
0:23.6194888 [RECV] > RoomCreated;FERovci,Prolaz,1001,True,False,1,mat1
0:28.4147641 [RECV] > UserJoined;ivan
...
0:35.8092133 [RECV] > UserReady;ivan;True
0:36.4110994 [SEND] > SetReady;True
0:36.4137374 [RECV] > UserReady;mat1;True
0:40.5069905 [SEND] > StartGame
0:40.5767518 [RECV] > GameStarted;3;1|10|12|18|19|23
0:57.9234883 [SEND] > Bid;
0:57.9260468 [RECV] > TrumpNext
0:57.9263593 [RECV] > FullCards;1|10|12|17|18|19|23|25
1:01.6148003 [RECV] > TrumpChosen;Hearts
1:05.5918373 [SEND] > Declare;3|4|5
1:05.5995145 [RECV] > YouDeclared;True
1:05.6000036 [RECV] > PlayerDeclared;20
...
1:12.1970857 [RECV] > WinningDeclaration;3;20;29|30|31;
1:15.1449312 [RECV] > StartPlayingCards
1:18.1287587 [SEND] > PlayCard;7
1:18.1324349 [RECV] > YouPlayed;7
1:18.1334095 [RECV] > CardPlayed;25;False
1:19.1431633 [RECV] > CardPlayed;28;False
...
2:11.3988987 [RECV] > CardPlayed;14;False
2:11.3990660 [RECV] > TableComplete;0,20;40,122,Hearts,False
...
2:11.4069059 [RECV] > FinalCards;0;1|10|12|18|19|23;17|25
2:13.4087453 [RECV] > FinalScores;40,142,Hearts,False;1
2:14.4097212 [RECV] > TotalScore;40,142,Hearts,False;40|142
```

4. Funkcionalnosti aplikacije

Aplikaciju ćemo podijeliti na 4 osnovna prikaza, prijava korisnika u sustav, pregled dostupnih soba, čekanje ostalih igrača u sobi i partija u tijeku. Pomoću ovakve podjele aplikacije možemo prikazati sve funkcionalnosti aplikacije.

4.1. Prijava korisnika u sustav

Nakon pokretanja aplikacije i uspješnog povezivanja klijenta na poslužitelj, korisniku je vidljiva forma za prijavu u sustav (Slika 4.1).

The image shows a login window titled "AleB - Prijava". At the top center is a logo consisting of a blue oval with the text "aLeB" in white, set against a background of blue and white diagonal stripes. Below the logo are two input fields: "Korisničko ime:" followed by a text box, and "Lozinka:" followed by a text box. Below these fields is a button labeled "Prijava". The window has a dark gray background and standard window controls (minimize, maximize, close) in the top right corner.

Slika 4.1 Forma za prijavu korisnika

Ako korisnik već ima korisnički račun, dovoljno je samo unijeti svoje korisničko ime i tajnu lozinku kako bi se prijavio u sustav. Aplikacija dalje prelazi u pregled dostupnih soba.

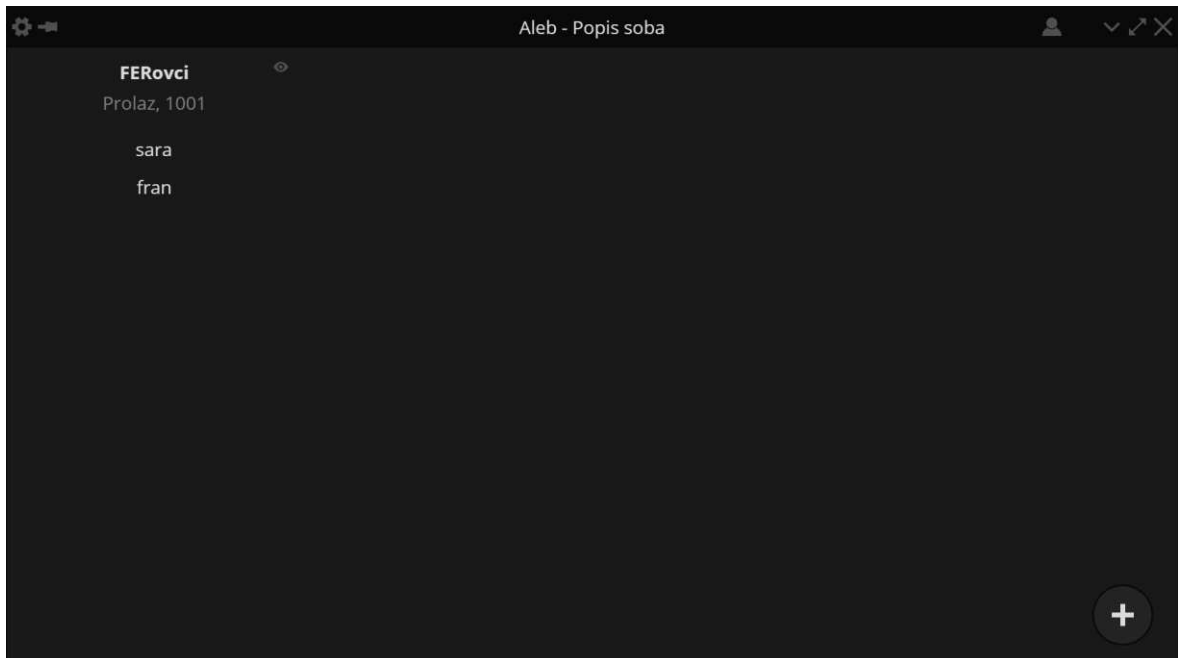
Ako korisnik nema postojeći korisnički račun, jednostavno unosi svoje željeno korisničko ime i lozinku direktno, na način kao da takav njegov račun već postoji. Ako željeno korisničko ime nije zauzeto, poslužitelj će automatski stvoriti novi račun za korisnika i tako provesti registraciju. Korisnik će tada odmah biti prijavljen u sustav.

4.2. Pregled dostupnih soba

Nakon uspješne prijave, korisniku je vidljiv popis postojećih soba (Slike 4.2 i 4.3), uz opciju da stvori vlastitu sobu.



Slika 4.2 Prazan popis soba

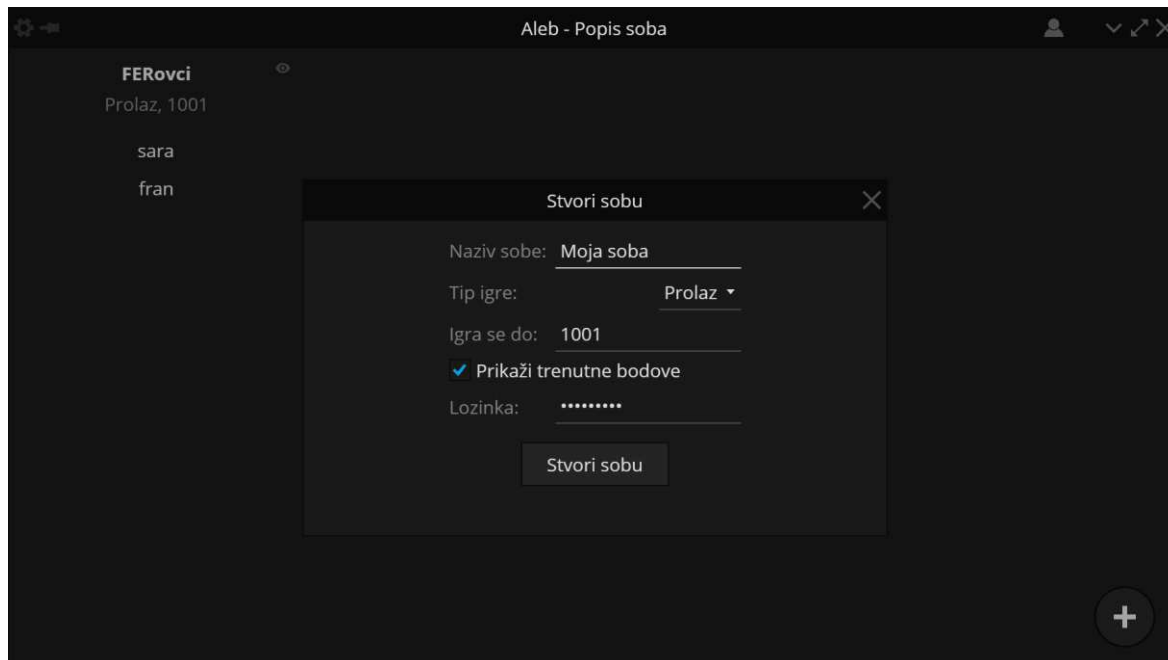


Slika 4.3 Popis soba s dostupnom sobom

Korisnik može kliknuti na sobu kako bi u nju ušao. Ako su u sobi već četiri igrača, onda je soba puna i korisnik ne može ući u takvu sobu kao igrač. Moguće se pridružiti gledanju takve sobe klikom na ikonicu oka u gornjem desnom kutu sobe, ali više o tome kasnije.

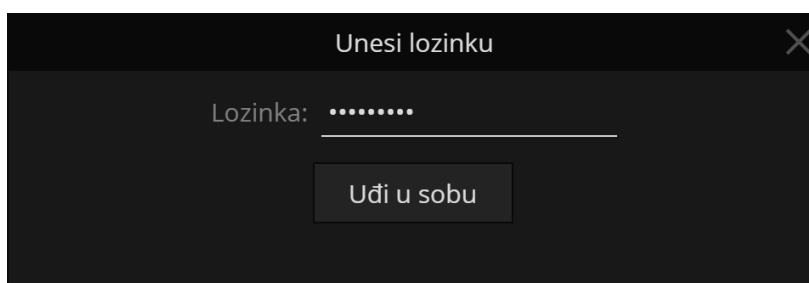
4.2.1. Stvaranje vlastite sobe

Klikom na tipku u obliku plusa u donjem desnom kutu, korisnik može započeti proces stvaranja vlastite sobe. Otvara se forma za stvaranje sobe (Slika 4.4):

The screenshot shows a game interface with a dark theme. In the top left, there's a settings icon and a user profile icon. The top bar displays 'Aleb - Popis soba'. On the left side, there's a list of users: 'FERovci', 'Prolaz, 1001', 'sara', and 'fran'. A modal dialog box titled 'Stvari sobu' is open in the center. It contains the following fields: 'Naziv sobe:' with the value 'Moja soba', 'Tip igre:' with a dropdown menu showing 'Prolaz', 'Igra se do:' with the value '1001', a checked checkbox for 'Prikaži trenutne bodove', and a password field 'Lozinka:' with masked characters. A 'Stvari sobu' button is at the bottom of the dialog. In the bottom right corner of the main interface, there is a circular button with a plus sign.

Slika 4.4 Forma za stvaranje sobe

Korisnik može proizvoljno izabrati naziv sobe i dovoljan broj bodova za pobjedu. Moguće je sakriti trenutni broj bodova u rundi, što znači da igrači onda trebaju pamtititi bodove dok idu ako ih žele znati. Sobu je također moguće zaštititi jednokratnom lozinkom koju će ostali korisnici morati unijeti prilikom ulaska u sobu (Slika 4.5).

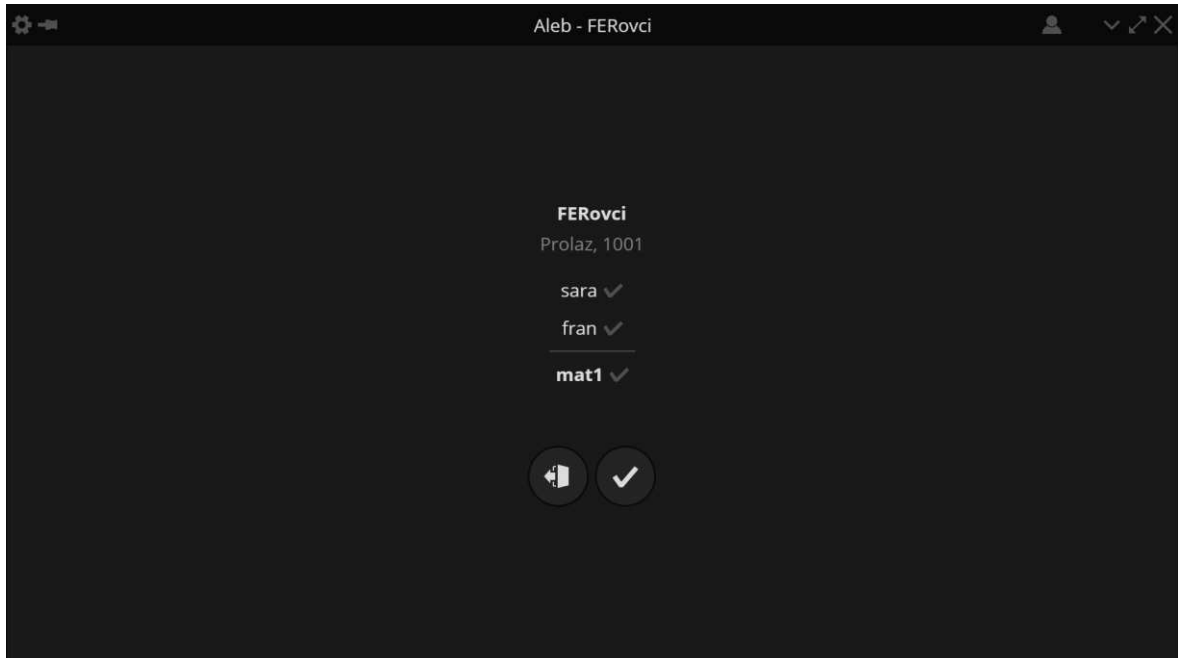
The screenshot shows a modal dialog box titled 'Unesi lozinku'. It contains a password field labeled 'Lozinka:' with masked characters. Below the field is a button labeled 'Uđi u sobu'.

Slika 4.5 Upit unosa lozinke za zaštićenu sobu

Za tip igre moguće je iz padajućeg izbornika izabrati „Prolaz“ ili „Dosta“. U slučaju igre na prolaz igra završava tek nakon što je neki kartaški par sakupio dovoljan broj bodova na kraju runde, dok u slučaju igre na dosta igra završava čim kartaški par sakupi dovoljan broj bodova tijekom runde. Primijetite da je u slučaju igre na prolaz moguće sakupiti dovoljan broj bodova tijekom runde ali ih izgubiti na kraju runde, primjerice padom.

4.3. Čekanje ostalih igrača u sobi

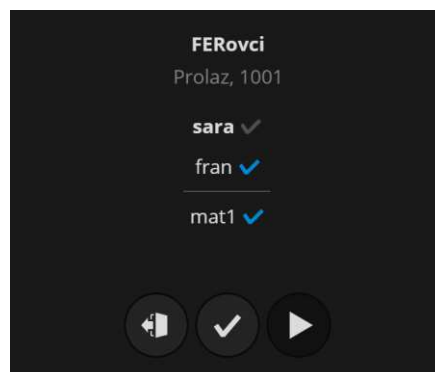
Nakon ulaska u sobu, korisniku je vidljiv prikaz sobe (Slika 4.6) dok čeka da sva četiri igrača pristupe sobi.



Slika 4.6 Prikaz korisnika koji se nalazi u sobi

Vlasnik sobe uvijek je prvi igrač u sobi. Korisnik ima mogućnost izlaska iz sobe i mijenjanja svoje oznake spremnosti. Kartaški parovi odvojeni su crtom – dakle prva dva igrača čine prvi kartaški par u partiji, a posljednja dva igrača čine drugi kartaški par. Soba se raspušta i briše s popisa soba kada svi igrači iz nje izađu.

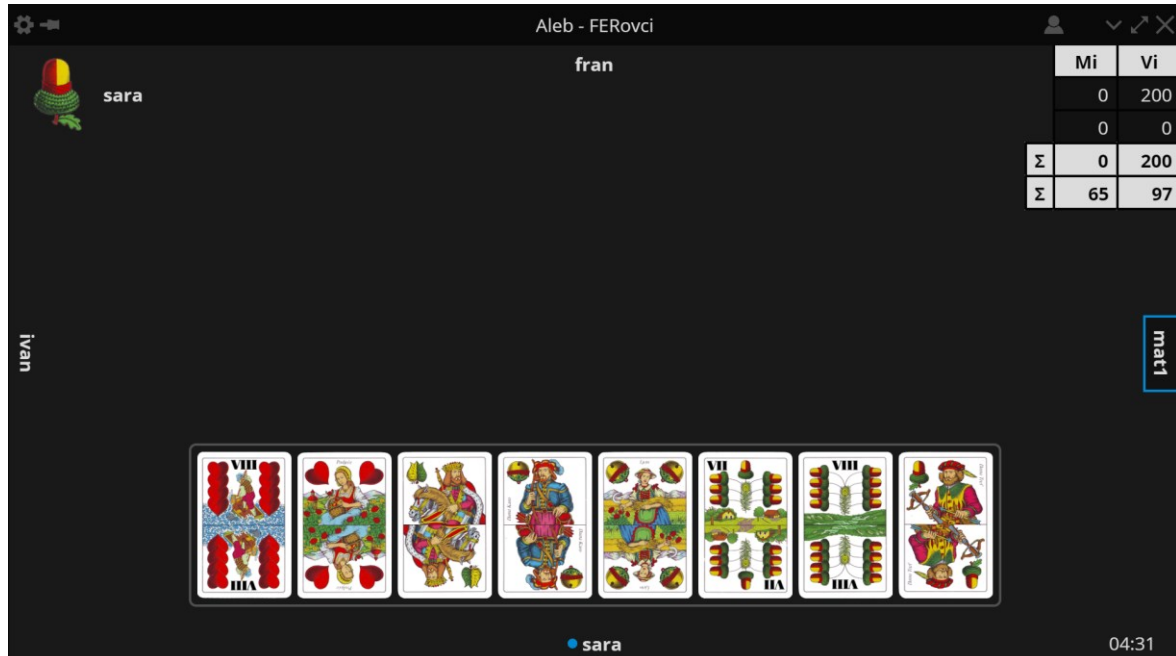
Prikaz je nešto drukčiji za vlasnika sobe (Slika 4.7). Vlasnik sobe jedini može započeti partiju, a sva četiri igrača moraju biti spremna kako bi to mogao učiniti. Također jedini ima mogućnost mijenjanja rasporeda igrača u sobi (tko je s kim kartaški par) te izbacivanja iz sobe ostalih igrača u sobi.



Slika 4.7 Prikaz sobe za vlasnika sobe

4.4. Partija u tijeku

Nakon što su se okupila četiri igrača u sobi, svi označili da su spremni za početak partije i vlasnik sobe je pokrenuo igru, svi igrači prelaze na prikaz partije u tijeku (Slika 4.8).



Slika 4.8 Prikaz partije u tijeku

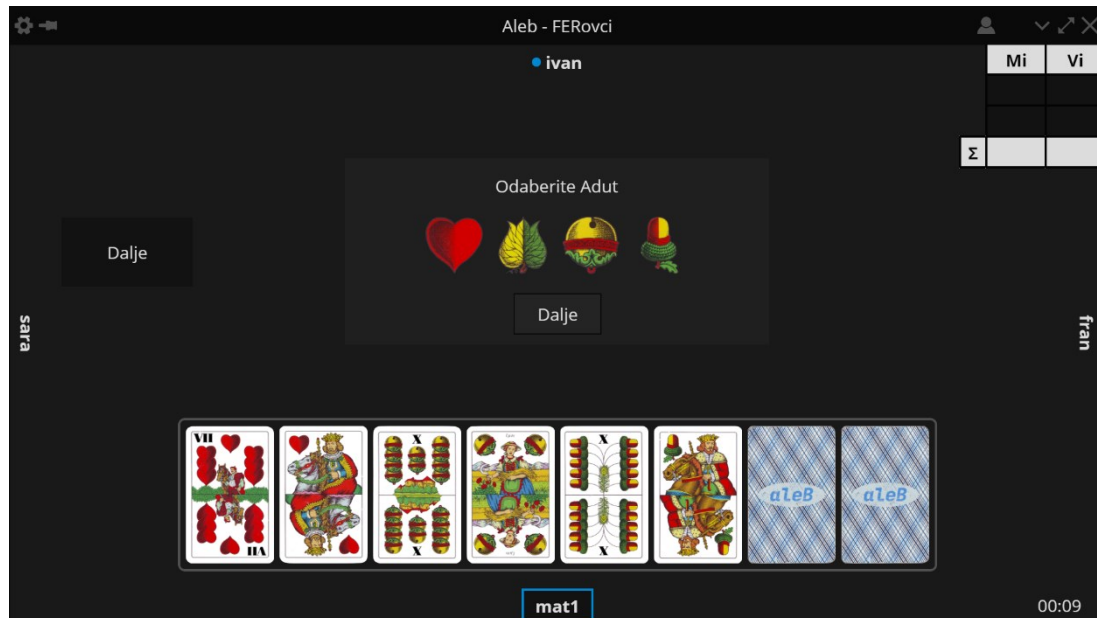
Jedan kartaški par su dva nasuprotna igrača. Korisnik je uvijek donji igrač, nasuprot njega je njegov partner, a s lijeve i desne strane je protivnički kartaški par. Plavi pravokutnik koji okružuje ime igrača označava da je taj igrač na potezu, tj. čeka se da odigraju svoj potez. Plava točkica pored imena označava djelatitelja karata i zadnjeg igrača po redu u krugu, tzv. „igrača na musu“. U donjem desnom kutu prikazano je trenutno proteklo vrijeme partije.

U gornjem desnom kutu prikazana je tablica stanja bodova. Prvi stupac „Mi“ označava korisnikov kartaški par, dok stupac „Vi“ označava protivnički kartaški par. U prvom retku prikazana su zvanja svakog kartaškog para, u drugom retku prikazano je trenutno stanje bodova u rundi, u trećem retku prikazan je ukupni zbroj trenutne runde, a u četvrtom retku prikazan je ukupni zbroj svih dovršenih rundi. Pregled dovršenih rundi otkriva se prelaskom miša preko tablice stanja bodova.

Jedna runda se sastoji od tri glavna dijela (faze igre): određivanje aduta, licitiranje zvanja, i kartanje. Nakon što su sve tri faze dovršene, računaju se krajnji bodovi te runde i runda se dodaje na popis dovršenih rundi. Partija se sastoji od više rundi, onoliko koliko je potrebno dok određeni kartaški par ne sakupi dovoljan broj bodova za pobjedu (najčešće 1001 bod).

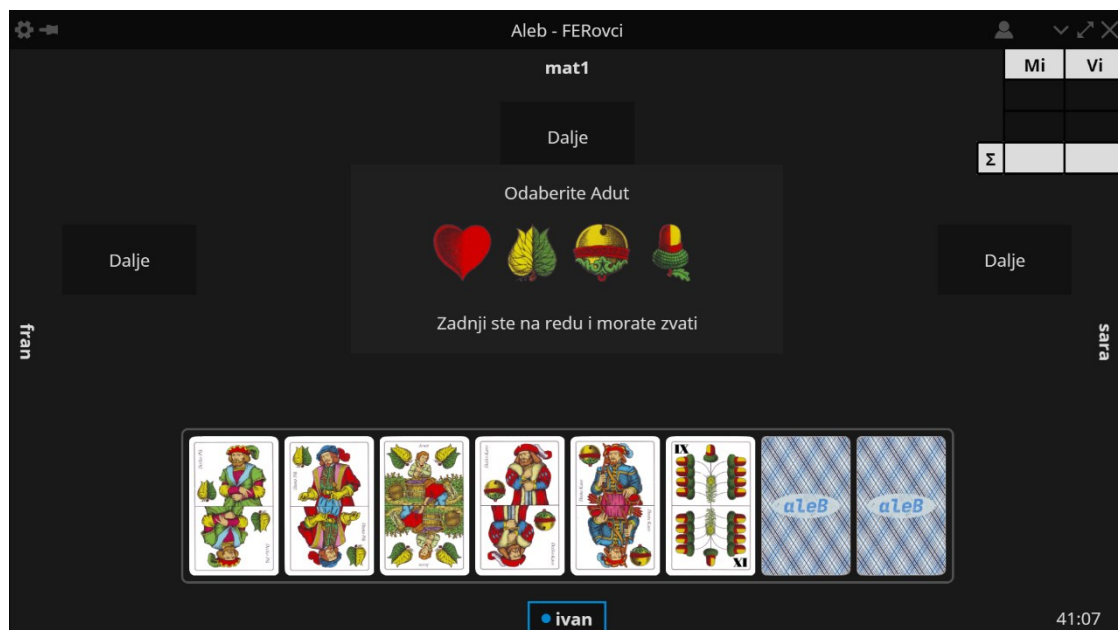
4.4.1. Određivanje aduta

Prva faza runde je određivanje aduta. Počevši od igrača desno od djelatelja pa dalje u smjeru obrnuto kazaljke na satu igrači imaju opciju odrediti između četiri aduta (redom: herc, pik, karo, tref) sa znanjem o šest karata koje drže u ruci.



Slika 4.9 Upit o odabiru aduta

Na sredini prozora otvara se upit (Slika 4.9) gdje igrač može izabrati jednu od četiri boja za odrediti ili će reći „Dalje“. Preostale dvije karte (tzv. „talon“) otkrivaju se nekom igraču nakon što je rekao dalje, ili je boju izabrao netko od prethodnih igrača.



Slika 4.10 Igrač koji je djelatelj mora izabrati boju

4.4.1.1 Djelitelj (igrač na musu)

Ako svi igrači kažu dalje, igrač koji je djelitelj mora izabrati boju (Slika 4.10). Iznimno, djelitelj može izabrati neku od dvije karte iz talona, te klikom na nju „zvati na talon“. Tada je on izabrao boju te karte kao svoj izbor, i zato mora pokazati tu kartu svim ostalim igračima (Slika 4.11).

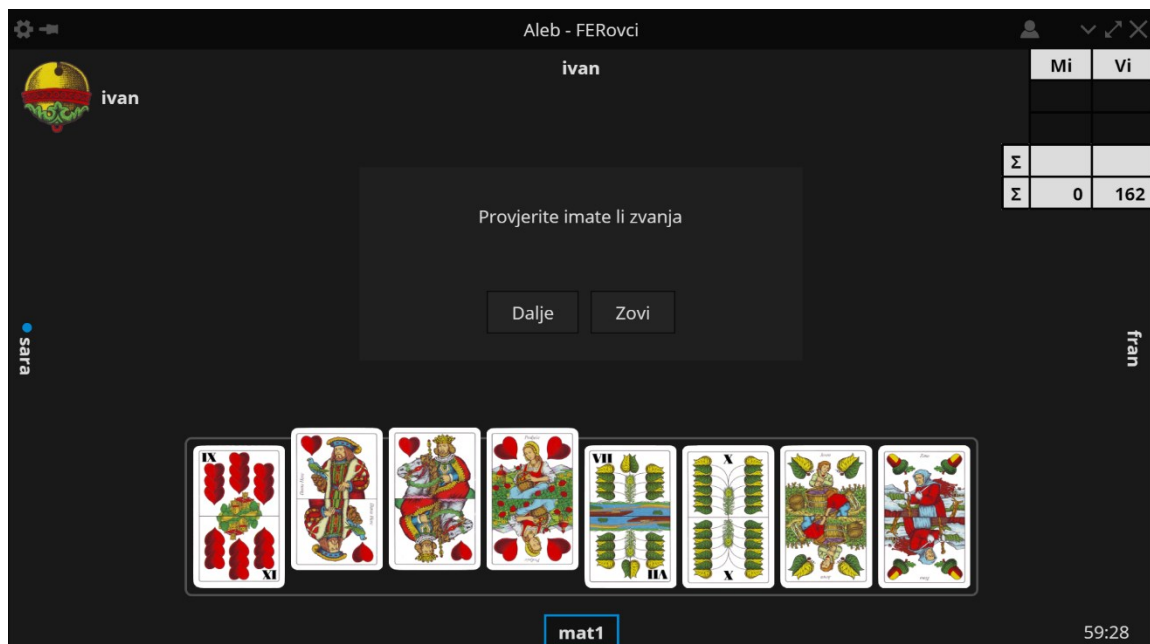


Slika 4.11 Igrač djelitelj je zvao na talon

Nakon što je adut izabran, izabrani adut do kraja runde je prikazan u gornjem lijevom kutu prozora uz korisničko ime osobe koja ga je izabrala. Runda prelazi u fazu licitiranja zvanja.

4.4.2. Licitiranje zvanja

Slično kao kod određivanja aduta, počevši od igrača desno od djelitelja pa dalje u smjeru obrnuto kazaljke na satu igrači imaju opciju licitirati zvanja iz svih osam karata.



Slika 4.12 Upit o licitiranju zvanja, s izabranih 20 zvanja do asa u hercu

Na sredini prozora otvara se upit (Slika 4.12) gdje igrač može izabrati koje karte iz svoje ruke želi uključiti u svoje zvanje. Ako je to zvanje valjano, onda ih igrač može zvati. Ako igrač nema zvanja ili ih iz nekog razloga ne želi zvati, onda može jednostavno izabrati opciju „Dalje“.

Osim bele koja se može zvati jedino tijekom faze kartanja, postoje 3 vrste zvanja: niz karata u istoj boji, četiri karte iste vrijednosti, i belot (svih osam karata iste boje). U slučaju jednakih zvanja, prednost ima igrač koji je prije na redu u igri. Drugi igrač iz kartaškog para također može pridodati svoje zvanje iako možda ima niže od protivnika, ako ovaj prvi ipak ima najjače zvanje od svih igrača.

4.4.2.1 Niz karata u istoj boji

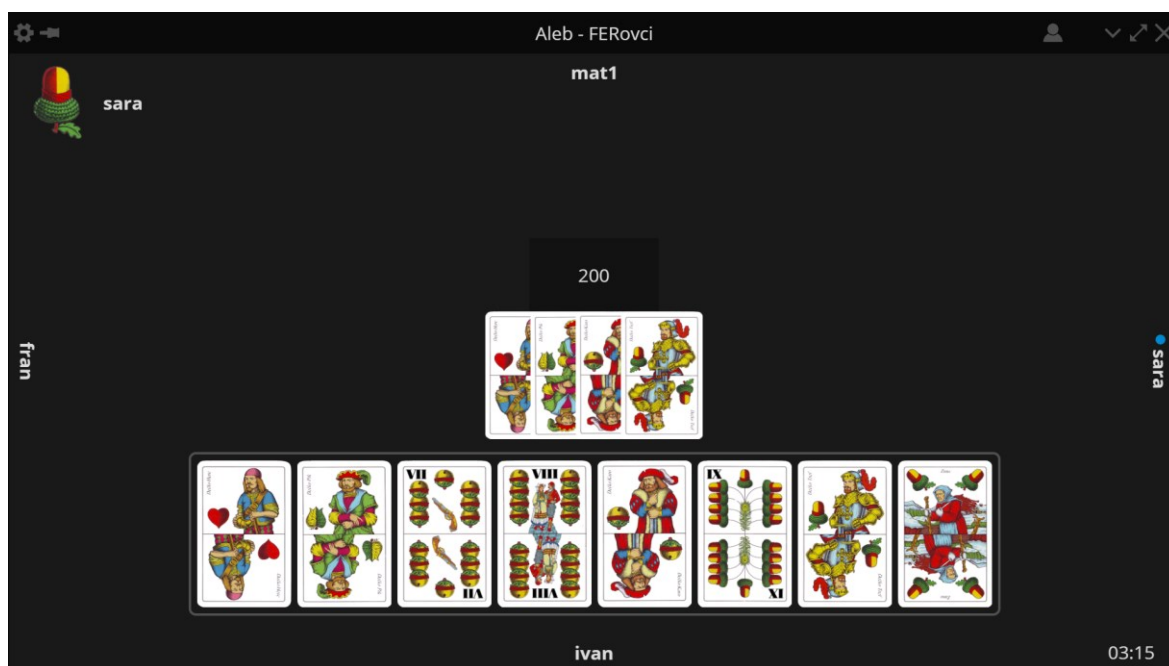


Slika 4.13 Osvojenih 100 zvanja do dame u piku

Poredak karata za zvanje nizom je sljedeći: sedmica, osmica, devetka, desetka, dečko, dama, kralj, as. Ako igrač primijeti niz karata u tom poretku (Slika 4.13), može zvati jedno od sljedećih zvanja:

- 20 – tri karte u nizu
- 50 – četiri karte u nizu
- 100 – pet ili više karata u nizu.

4.4.2.2 Četiri karte iste vrijednosti



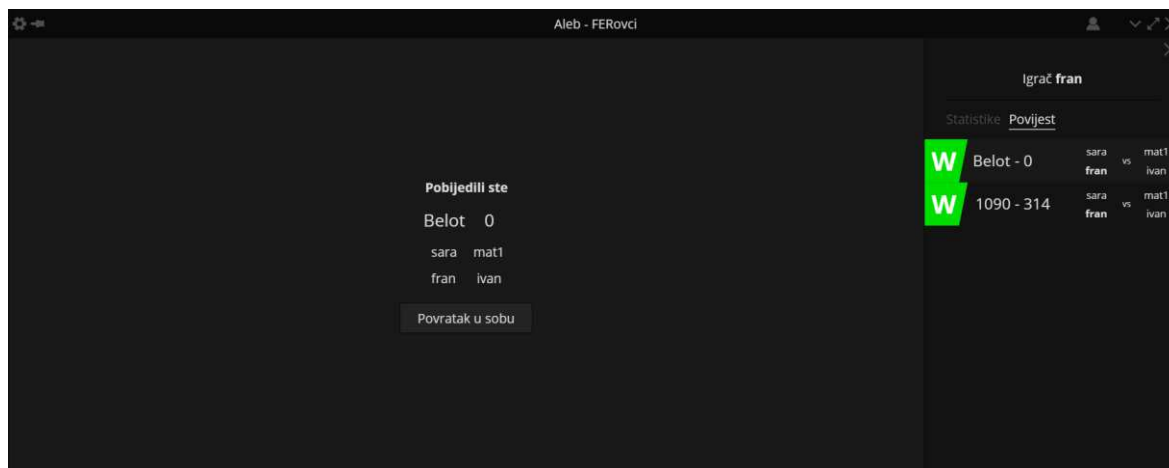
Slika 4.14 Osvojenih 200 zvanja s 4 dečka

Ako igrač primijeti četiri karte iste vrijednosti (Slika 4.14), može zvati jedno od sljedećih zvanja:

- 100 – četiri desetke, dame, kralja ili asa
- 150 – četiri devetke
- 200 – četiri dečka.

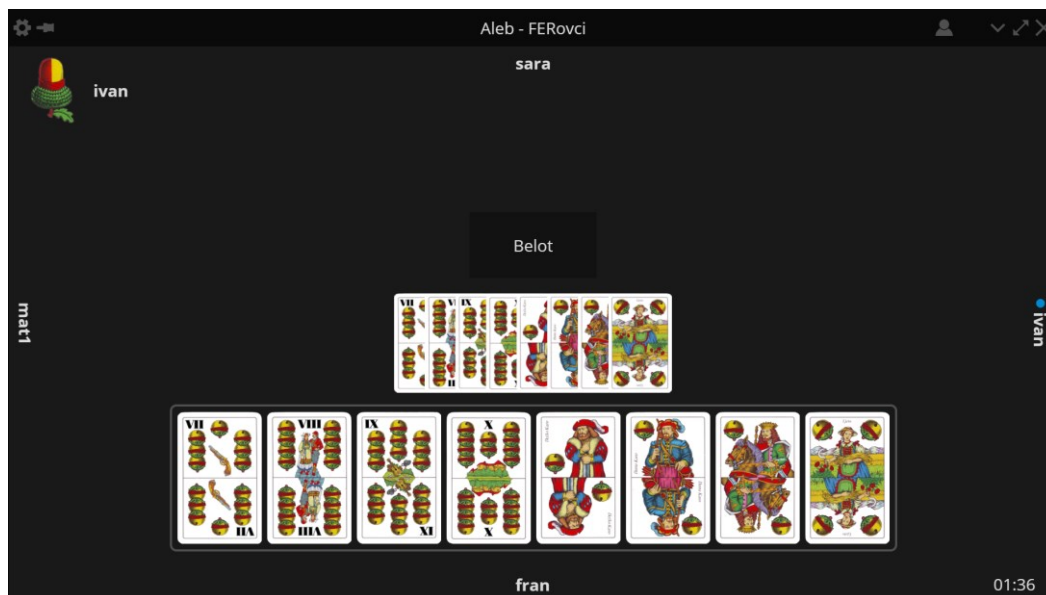
Primijetite da četiri sedmice i četiri osmice ne donose nikakvo zvanje. Ovakva zvanja uvijek su jača od prethodno opisanog niza karata.

4.4.2.3 Belot



Slika 4.15 Partija je odmah prekinuta radi belota

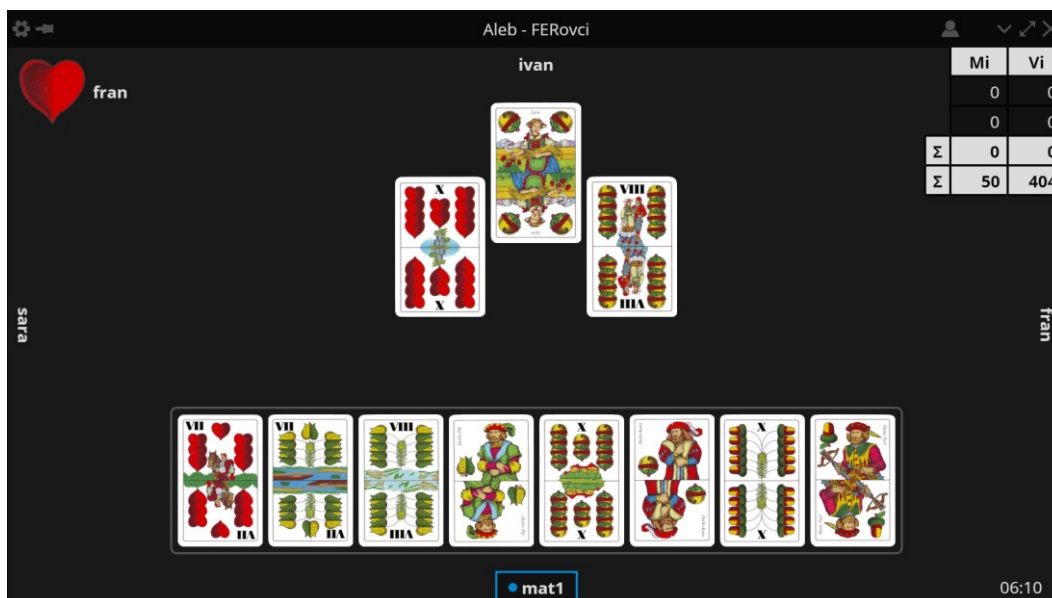
Ako igrač ima svih osam karata iste boje, može zvati belot (Slika 4.16). Belotom se cijela partija odmah prekida (Slika 4.15) i pobjeđuje kartaški par koji je zvao belot.



Slika 4.16 Osvojen belot u kari

4.4.3. Kartanje

Nakon zvanja igra prelazi u posljednju fazu kartanja (Slika 4.17).



Slika 4.17 Proces kartanja

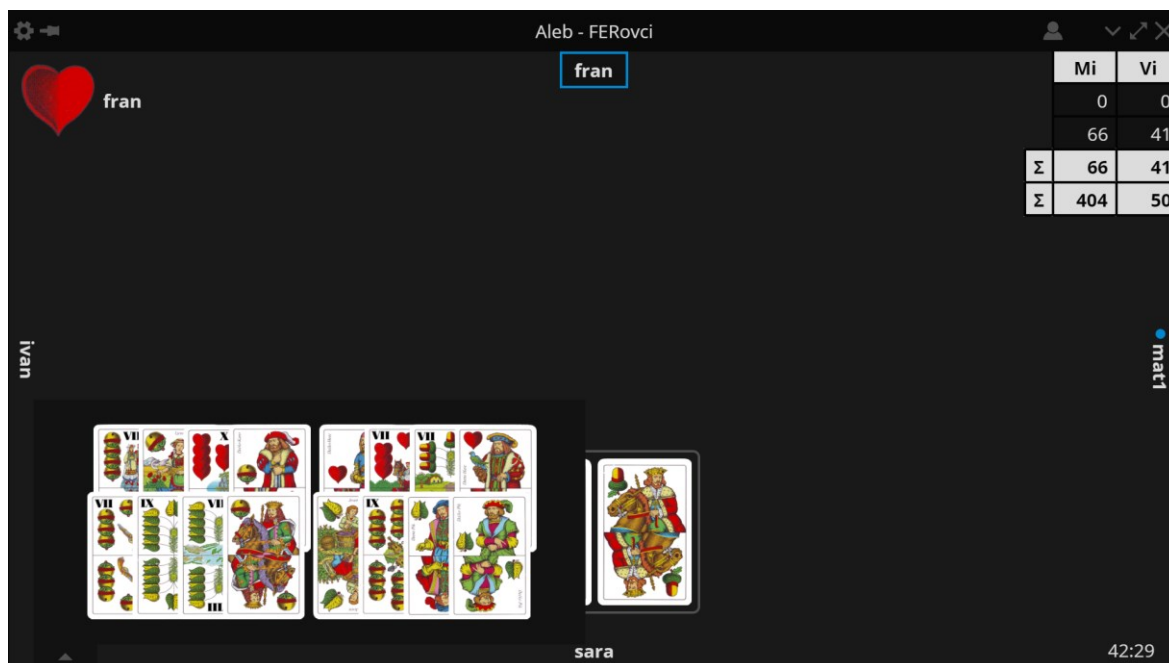
Prvi igrač baca bilo koju kartu iz ruke i određuje boju na stolu. Ostali igrači prate u smjeru suprotno kazaljke na satu. Obavezno je pratiti boju („odgovoriti“) i baciti jaču ako je ima (pravilo „ibera“). U istoj boji, jača je ona karta koja ima veću vrijednost. Vrijednosti karata raspoređene su prema Tablici 4.1.

	Uobičajena vrijednost	Vrijednost u adutu
Sedmica	0	0
Osmica	0	0
Devetka	0	14
Desetka	10	10
Dečko	2	20
Dama	3	3
Kralj	4	4
As	11	11

Tablica 4.1 Vrijednosti karata

Tako je osnovna vrijednost igre bez nadodanih zvanja jednaka 162.

U slučaju nemogućnosti praćenja boje, prvo baca adute, a onda ostale karte. Aplikacija će upozoriti igrača ako je pokušao baciti neispravnu kartu. Igrač s najjačom kartom na stolu osvaja štih i bodove za svoj kartaški par i dalje nastavlja igru. Svaki kartaški par može pratiti koje štihove su osvojili prelaskom miša preko tipke u donjem lijevom kutu (Slika 4.18).



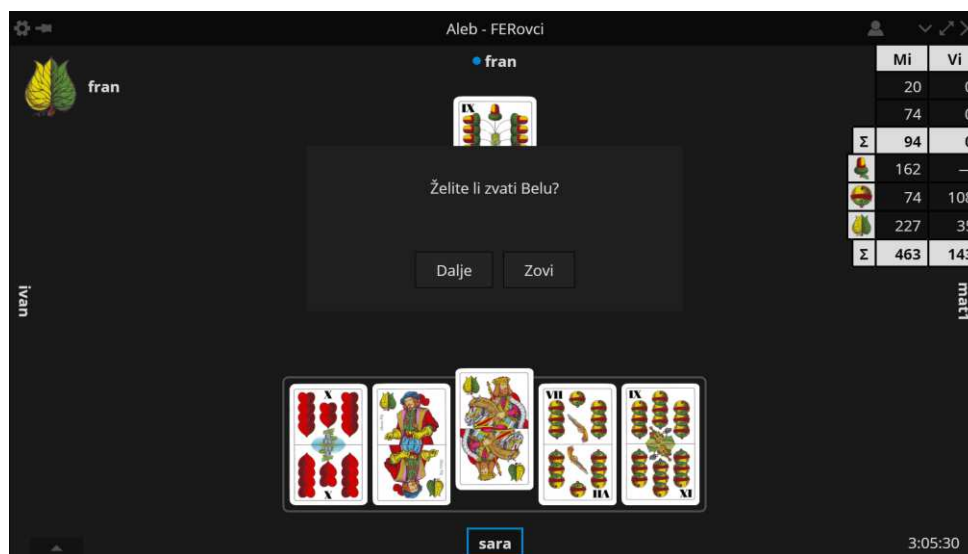
Slika 4.18 Prikaz osvojenih štihova u trenutnoj rundi

4.4.3.1 Dodatni bodovi

Pored zvanja, jedan od načina za dodati na ukupnu vrijednost igre je 10 bodova kartaškom paru koji je osvojio posljednji štih.

Kažemo da se dogodila tzv. „štiglja“ ako je jedan kartaški par pokupio sve štihove, za što su nagrađeni s 90 bodova. Primijetite da je moguće da jedan kartaški par osvoji štih koji ne vrijedi ni jedan bod, u kojem slučaju protivnički kartaški par ipak nije osvojio štiglju.

Zvanje bele moguće je napraviti kada igrač ima i damu i kralja u boji aduta. Vrijedi 20 bodova i pribraja se neovisno o prethodno izvedenim pravilima o zvanjima. Kada igrač odluči baciti tu kartu, na prozoru se pojavljuje upit (Slika 4.19) koji pita korisnika želi li zvati belu ili reći dalje.



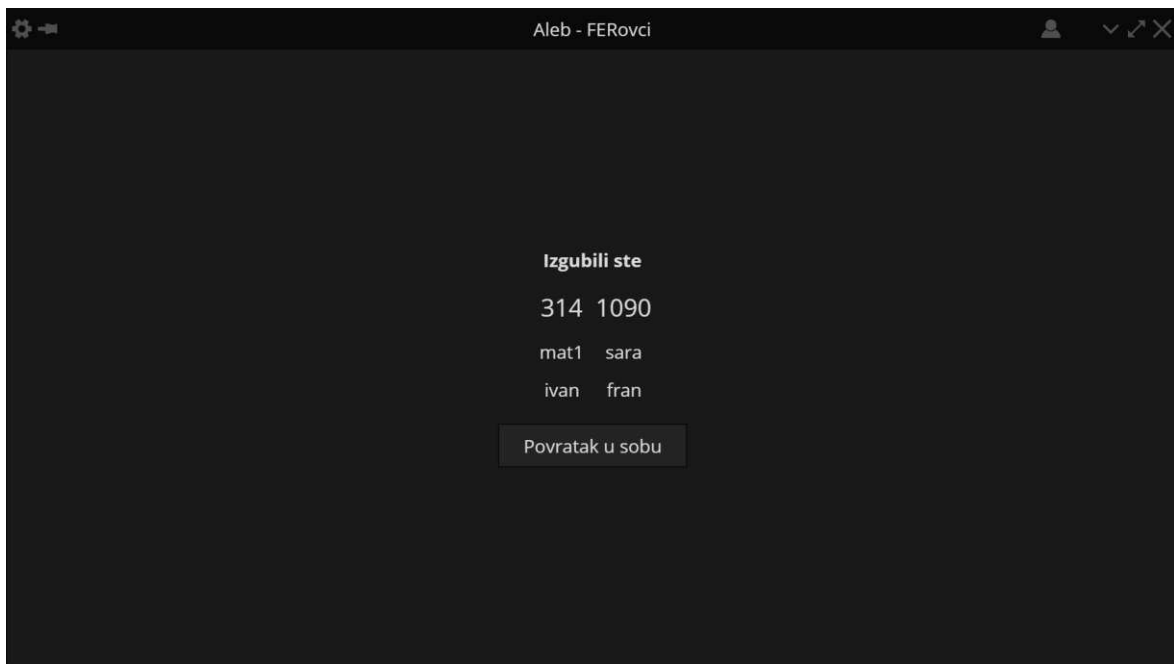
Slika 4.19 Upit o zvanju bele

4.4.3.2 Završetak runde

Kartaški par koji je birao aduta mora sakupiti strogo više od pola ukupne vrijednosti igre kako bi prošao. Ako ne uspiju sakupiti toliko bodova, smatra se da su pali i ukupna vrijednost igre se zapisuje protivničkom kartaškom paru. Pad se u tablici dovršenih rundi bilježi crticom.

Nakon toga je runda dovršena, novi djelatelj postaje onaj desno od prethodnog, i igra prelazi u novu rundu u početnoj fazi određivanja aduta. U novoj rundi, svim igračima omogućen je pregled karata iz prethodne runde svih igrača klikom na željenog igrača.

Kada kartaški par sakupi dovoljan broj bodova za pobjedu cijele partije, umjesto prelaska na novu rundu igra se prekida i prikazan je krajnji rezultat partije (Slika 4.20). Korisniku je omogućen povratak u sobu kao i prije početka partije. Partija je tada upisana u povijest partija na profilu igrača sudionika.

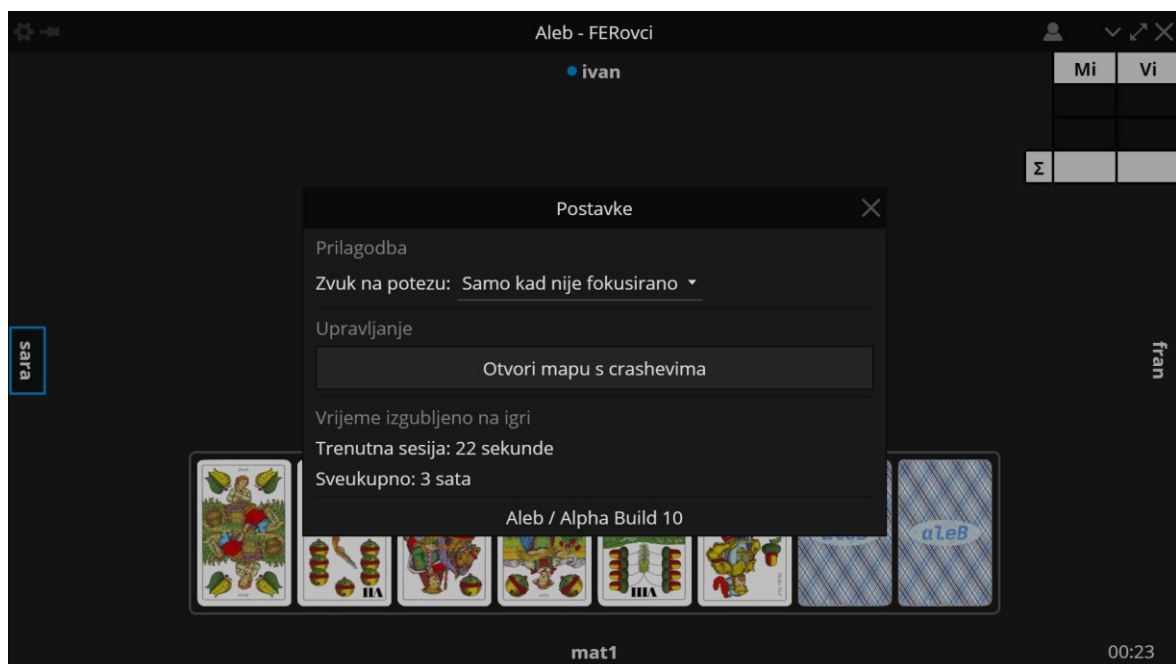


Slika 4.20 Krajnji rezultat partije s tipkom za povratak u sobu

4.5. Ostale i općenite funkcionalnosti

4.5.1. Postavke

Klikom na ikonicu zupčanika koja je stalno vidljiva u gornjem lijevom kutu prozora otvara se skočni prozor s postavkama (Slika 4.21) koji korisniku omogućava prilagodbu određenih klijentskih aspekata aplikacije:



Slika 4.21 Skočni prozor s postavkama

Korisnik može konfigurirati da se zvučna obavijest kada je igrač na potezu ne reproducira nikad, da se reproducira samo ako prozor nije fokusiran, ili da se uvijek reproducira. Također je moguće otvoriti skrivenu mapu s *crash* logovima u slučaju da se aplikacija neočekivano sruši pa bi te datoteke mogle pomoći programerima da poprave grešku.

Prikazano je i vrijeme potrošeno na igranje igre, koje se broji samo dok je prikaz partije vidljiv. Prikazana je i verzija aplikacije. Ikonica u obliku pribadače u gornjem lijevom kutu prozora mijenja „always-on-top“ stanje prozora.

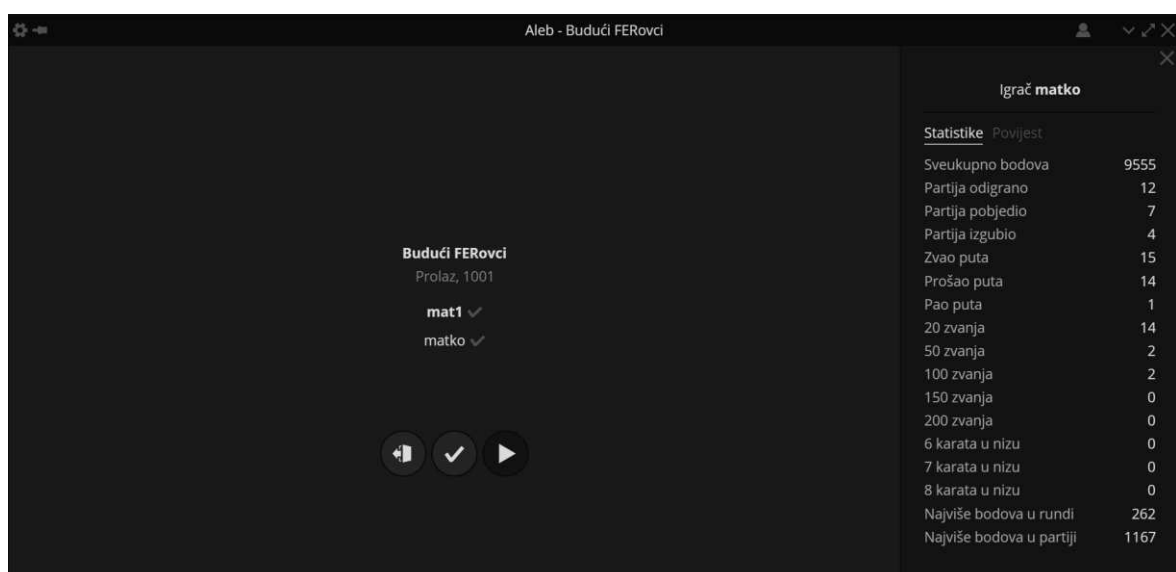
4.5.2. Korisnički profili

Na većini mjesta gdje se pojavljuju korisnička imena (pregled dostupnih soba, prikaz sobe, i tijekom partije) moguće je otvoriti pregled profila desnim klikom na prikazano korisničko ime i biranjem opcije „Profil“ (Slika 4.22):



Slika 4.22 Tipka za pregled profila

Na prikazu profila (Slika 4.23), dostupne su zanimljive statistike tog igrača i povijest njihovih partija (Slika 4.24).



Slika 4.23 Prikaz profila sa statistikama

Igrač mito			
Statistike		Povijest	
W	1087 - 791	mat1 mito	Kristalki vs Mata
L	853 - 1139	mat1 mito	Kristalki vs Mata
W	1026 - 752	mat1 mito	Kristalki vs Mata
W	1065 - 299	mat1 mito	matko vs Stjepan
L	550 - 1018	mat1 mito	matko vs Stjepan

Slika 4.24 Povijest partija na profilu igrača

Vlastitom profilu moguće je pristupiti klikom na ikonicu čovjeka koja je stalno vidljiva u gornjem desnom kutu prozora. Profilu proizvoljnog korisnika poznatog korisničkog imena moguće je pristupiti desnim klikom na istu ikonicu čovjeka. U formu koja se tada otvori (Slika 4.25) potrebno je unijeti traženo korisničko ime:

Pretraži igrača

Naziv igrača:

Pretraži

Slika 4.25 Pristup profilu proizvoljnog korisnika

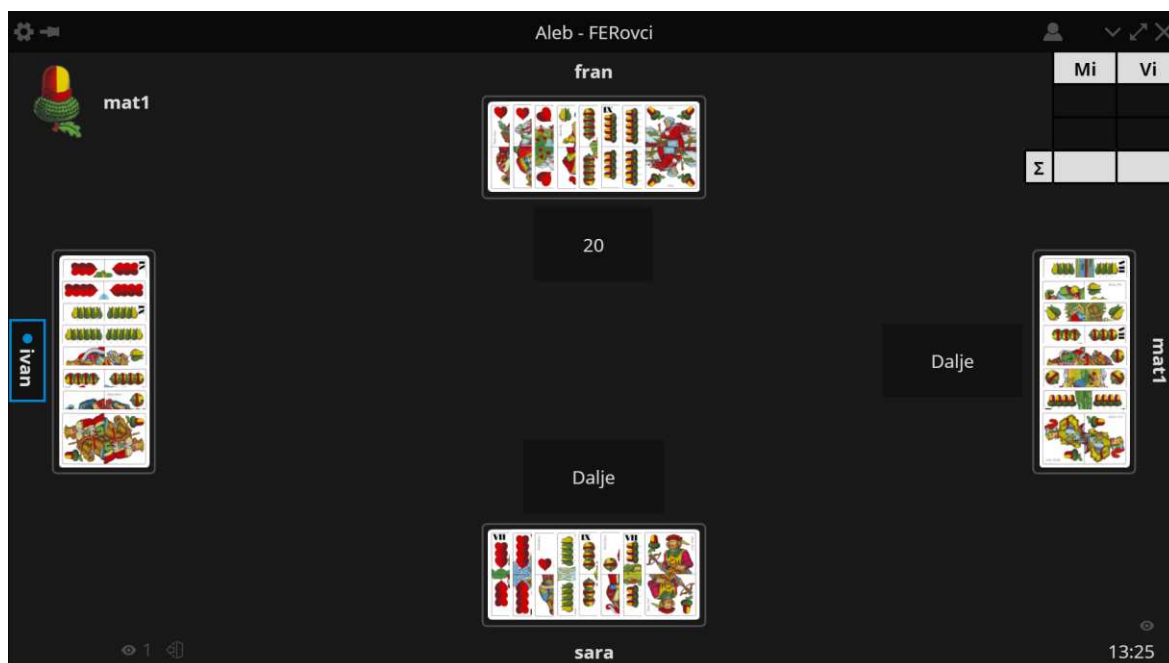
4.5.3. Gledanje partija

Korisnik može gledati tijek sobe uživo klikom na ikonicu oka (Slika 4.26) u gornjem desnom kutu sobe na prikazu pregleda dostupnih soba:



Slika 4.26 Ikonica za početak gledanja sobe

Korisnik tada postaje gledatelj sobe i vidi sve što igrači u sobi vide iz neutralne perspektive promatrača (Slika 4.27). U donjem lijevom kutu svim igračima i gledateljima je vidljiv broj gledatelja koji trenutno gledaju tu sobu. Gledatelji također pored tog broja imaju i tipku za izlaz iz gledanja sobe natrag na popis dostupnih soba. U donjem desnom kutu, gledateljima je dodatno dostupna ikonica u obliku oka koja mijenja jesu li sve karte igrača vidljive ili ne (Slika 4.28).



Slika 4.27 Gledanje igre s prikazanim kartama igrača



Slika 4.28 Gledanje igre sa skrivenim kartama igrača

5. Korištene tehnologije i alati

5.1. .NET 6

.NET [6] je platforma otvorenog koda za razvoj različitih vrsta aplikacija koju je razvio Microsoft. .NET aplikacije mogu se pisati u više jezika: C#, F# ili Visual Basic. Korišten je kao osnovna platforma za izradu aplikacije zajedno s jezikom C# 10. Iako .NET podržava više platformi (Windows, Linux i macOS), rad je namijenjen samo za Windows operacijski sustav, ali bi se lako mogao prenamijeniti za ostale podržane operacijske sustave.

5.2. Avalonia

Avalonia [7] je višepatformski okvir otvorenog koda za izradu grafičkog sučelja za aplikacije pisane u .NET platformi. Razvoj grafičkog sučelja u Avalonia okviru nije strogo vezan uz nijedan operacijski sustav. Smatra se nasljednikom pouzdane WPF platforme za nešto stariji .NET Framework. Koristi se za oblikovanje i prikaz svog grafičkog sučelja u aplikaciji za krajnjeg korisnika.

5.3. Humanizer

Humanizer [8] je biblioteka otvorenog koda za .NET čija je svrha pretvorba raznih tipova podataka poput datuma, raspona vremena i sličnog u ljudski čitljive ekvivalente. Koristi se za jednostavan prikaz istih na hrvatskom jeziku.

5.4. NetCoreAudio

NetCoreAudio [9] je minimalna biblioteka otvorenog koda za .NET koja omogućuje jako jednostavnu reprodukciju zvuka na bilo kojem operativnom sustavu. Korištena je za jednostavne zvučne obavijesti.

5.5. Visual Studio 2022

Visual Studio [10] je integrirano razvojno okruženje od Microsofta specijalizirano za rad s .NET platformom. Sadrži uređivač koda s podrškom za dopunjavanje koda, vizualni dizajner s pretpregledom grafičkog sučelja, i integrirani sustav za otklanjanje grešaka (debugger). Nudi potpuno upravljanje različitim komponentama aplikacije i njihovom integracijom. Inačica Visual Studio 2022 Enterprise korištena je za izradu aplikacije u potpunosti.

5.6. Git i GitHub Desktop

Git [11] je besplatan softver otvorenog koda za praćenje promjena na datotekama. Često se koristi za koordinaciju izvornog koda na projektu razvoja softvera. Korišten je tijekom razvoja aplikacije kako bi se olakšao proces održavanja izvornog koda. GitHub Desktop [12] je aplikacija koja omogućava brži, jednostavniji i pregledniji rad s Gitom kroz grafičko sučelje.

Zaključak

Cilj ovog završnog rada prvenstveno je bio naučiti mrežno programiranje iz temelja, ali i osigurati kolegama studentima platformu za neometano i sigurno kartanje Belota tijekom kriznih vremena. Razvijena aplikacija Aleb ima minimalno, jednostavno i intuitivno sučelje s nekim značajkama koje općenito ne nalazimo kod sličnih javno dostupnih aplikacija. Ipak, zbog vremenskih ograničenja manji dio željenih funkcionalnosti je ostao neimplementiran.

Pored toga, izrađena aplikacija definitivno ima prostora za daljnji razvoj i napredak. Prvenstveno, da aplikacija dosegne što veću ciljanu skupinu ljudi i bude lakše dostupna, korisničko sučelje trebalo bi ponovno biti implementirano unutar web-preglednika i/ili na mobilnim platformama. Funkcionalnosti koje bi bile malo više trivijalne za nadodati su predaja partije, reprodukcija snimljene partije, praćenje statistike i svjesnost djelatelja unutar sobe te slične manje prepravke.

Literatura

- [1] Playtoy Belot, <https://playtoy.tportal.hr/belot/gamelobby/124>, 11.4.2022.
- [2] Bela Online, <https://bela.fun/>, 11.4.2022.
- [3] Facebook BeloT, <https://belotportal.blogspot.com>, 11.4.2022.
- [4] Robert Odniković – BELOT Facebook, https://youtu.be/-Hcr_jluThk, 11.4.2022.
- [5] Osnovna pravila Belota, <https://www.eivanec.com/wp-content/uploads/2019/12/pravila-igre-bela-belot.pdf>, 11.4.2022.
- [6] .NET, <https://dotnet.microsoft.com/>, 5.6.2022.
- [7] Avalonia, <https://avaloniaui.net/>, 5.6.2022.
- [8] Humanizer, <https://github.com/Humanizr/Humanizer>, 5.6.2022.
- [9] NetCoreAudio, <https://github.com/mobiletechtracker/NetCoreAudio>, 5.6.2022.
- [10] Microsoft Visual Studio, <https://visualstudio.microsoft.com/>, 5.6.2022.
- [11] Git, <https://git-scm.com/>, 5.6.2022.
- [12] GitHub Desktop, <https://desktop.github.com/>, 5.6.2022.1

Sažetak

Naslov: Mrežna kartaška igra na platformi Windows

Sažetak:

U ovom radu opisana je izrada mrežne kartaške igre Belot imena Aleb. Na početku je odrađena analiza problemskog područja u kojoj je analizirano nekoliko sličnih aplikacija. Uz pomoć prikupljenih podataka iz analize i inovativnih ideja navedena je specifikacija zahtjeva. Nakon toga opisana je arhitektura sustava i iznesene su sve funkcionalnosti izrađene aplikacije uz opis korisničkog sučelja. Na kraju rada navedene su tehnologije korištene pri izradi i iznesen je zaključak.

Razvijena aplikacija ima implementirana sva pravila kartaške igre Belot, podržava više usporednih partija na istom poslužitelju, podržava razne varijacije igre, ima minimalno i responzivno korisničko sučelje, ima pregled karata završene runde ostalih igrača, prati statistike igrača i pregled prethodnih igara te ima mogućnost gledanja tuđe partije u tijeku uživo. Aplikacija je razvijena za operacijski sustav Windows na .NET platformi u radnom okviru Avalonia.

Ključne riječi: Windows aplikacija, Avalonia UI, .NET, C#, mreža, igra, multiplayer, Belot, Bela, kartanje, kartaška igra

Summary

Title: Online card game for Windows

Summary:

This paper describes the development of the online card game Belote named Aleb. At the beginning, an analysis of the problem area is performed, in which several similar applications are analysed. With the help of the collected data from the analysis and innovative ideas, the requirements are specified. After that, the system architecture is described and all the functionalities of the developed application are presented, along with a description of the user interface. At the end of the paper, the technologies used in development are listed and a conclusion is presented.

The developed application has implemented all the rules of the card game Belote, supports multiple simultaneous games on the same server, supports various game variations, has a minimal and responsive user interface, allows viewing hands of other players from completed rounds, tracks player statistics and allows viewing previous games, and allows watching someone else's game live. The application was developed for Windows on the .NET platform in the Avalonia UI framework.

Keywords: Windows application, Avalonia UI, .NET, C#, networking, game, multiplayer, Belote, Bela, playing cards, card game