

Lie to me

Članovi tima:

Danilo Vulović 15965

Dušan Zamahajev 15967

Sadržaj

Opis igre	3
Uloge	3
Cilj igre	3
Odvijanje igre	3
Funkcionalni zahtevi.....	3
Arhitekturni zahtevi	4
Nefunkcionalni zahtevi.....	4
Arhitekturni dizajn	4
Strukturni pogled	4
Bihevioralni pogled	5
Alokacioni pogled.....	6
Arhitekturni obrasci	6
Specifikacija biblioteka i programskih okvira.....	6

Opis igre

„Lie to me“ je interaktivna igra između 7 do 15 igrača izrađena po uzoru na „Town of Salem“ igru. Na početku igre igrači nasumično dobijaju uloge. Ove uloge se mogu raspodeliti u tri grupe: Townsfolk, Mafia i Neutrals.

Uloge

Townsfolk: Bodyguard, Doctor, Investigator, Jailor, Lookout, Medium, Mayor, Retributionists, Sheriff, Spy, Transporter, Vampire Hunter, Veteran, Vigilante, Crusader, Tracker, Trapper, Psychic.

Mafia: Blackmailer, Consigliere, Consort, Disguise, Forger, Framer, Godfather, Janitor, Mafioso, Hypnotist, Ambusher

Neutral: Arsonist, Executioner, Guardian Angel, Jester, Juggernaut, Pirat, Plaguebearer, Serial Killer, Werewolf, Witch, Vampire, Survivor.

Cilj igre

Ako je igrač u grupi Townsfolk, igrač treba ubiti sve pripadnike grupe Mafia pre nego što ga ubiju. U čemu je fora? Igrač ne zna ko je građanin a ko je zlikovac. Ako je igrač u grupi Mafia, on mora tajno da ubije sve građane pod velom noći, a da pritom ne bude uhvaćen. Do kraja igre se dolazi ukoliko su ubijeni svi igrači iz grupe Townsfolk ili Mafia, dok je pobednik grupa koja i dalje ima preživjele.

Igrači u grupi Neutrals, imaju specijalne ciljeve koji su definisani na osnovu njihove uloge.

Odvijanje igre

Igra se odvija kroz tri sukcesivne, ciklične faze: „Night“, „Discussion and voting“ i „Defense and judgement“.

Tokom faze „Night“, većina uloga može koristiti njihove sposobnosti. „Serial Killer“ može ubiti nekoga, „Doctor“ može izlečiti napadnutu osobu, i tako dalje.

Tokom faze „Discussion and voting“, svi prisutni igrači diskutuju o tome ko je osumnjičeni. Kada glasanje krene, glas većine odlučuje o tome ko ide na suđenje.

Tokom faze „Defense and judgement“, ukoliko je neko optužen, optuženi ima šansu da se odbrani. Nakon toga svi igrači glasaju da li su „za“, „protiv“ ili „uzdržani“ da optuženi bude osuđen. Glasovi su javni. Ako ima više glasove „za“ nego „protiv“, osuđeni ide na lomaču. Nakon ove faze, ponovo sledi faza „Night“.

Funkcionalni zahtevi

1. Povezivanje klijenta i servera.
2. Klijentu je omogućeno kreiranje sobe.
3. Klijentu je omogućeno pridruživanje već postojećim sobama.
4. Server omogućava kreiranje sobe i povezivanje 7-15 igrača.
5. Server omogućava početak igre.
6. Server vodi računa o toku igre, potezima igrača i ishodima poteza.
7. Igrač/klijent je obavešten o trenutnom stanju igre.
8. Logika za implementaciju igre.
9. Vođenje evidencije i skladištenje podataka.

Arhitekturni zahtevi

- Potrebno je omogućiti pamćenje velike količine podataka.
- Šema baze podataka je skrivena.
- Sistem će biti dostupan 7 dana nedeljno, 24 sata dnevno.

Nefunkcionalni zahtevi

- Skalabilnost
- Performanse
- Održljivost
- Evolutivnost
- Pouzdanost
- Lakoća postavljanja

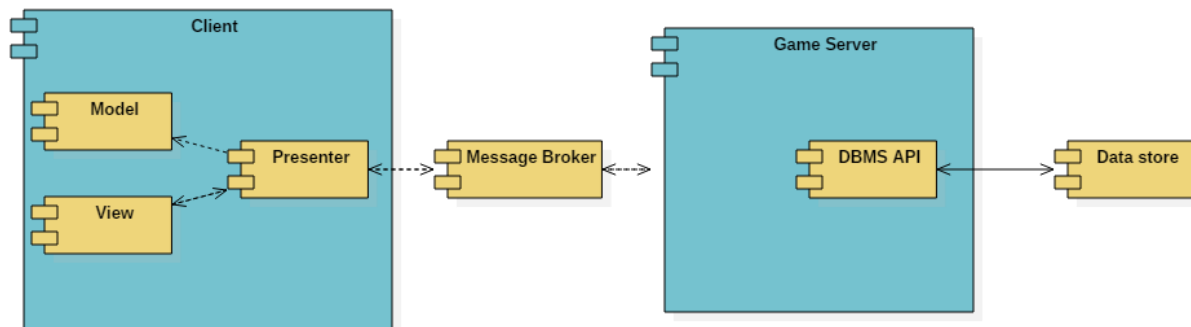
Arhitekturni dizajn

Strukturni pogled

Prikazuje strukturu sistema navodeći komponente sistema.

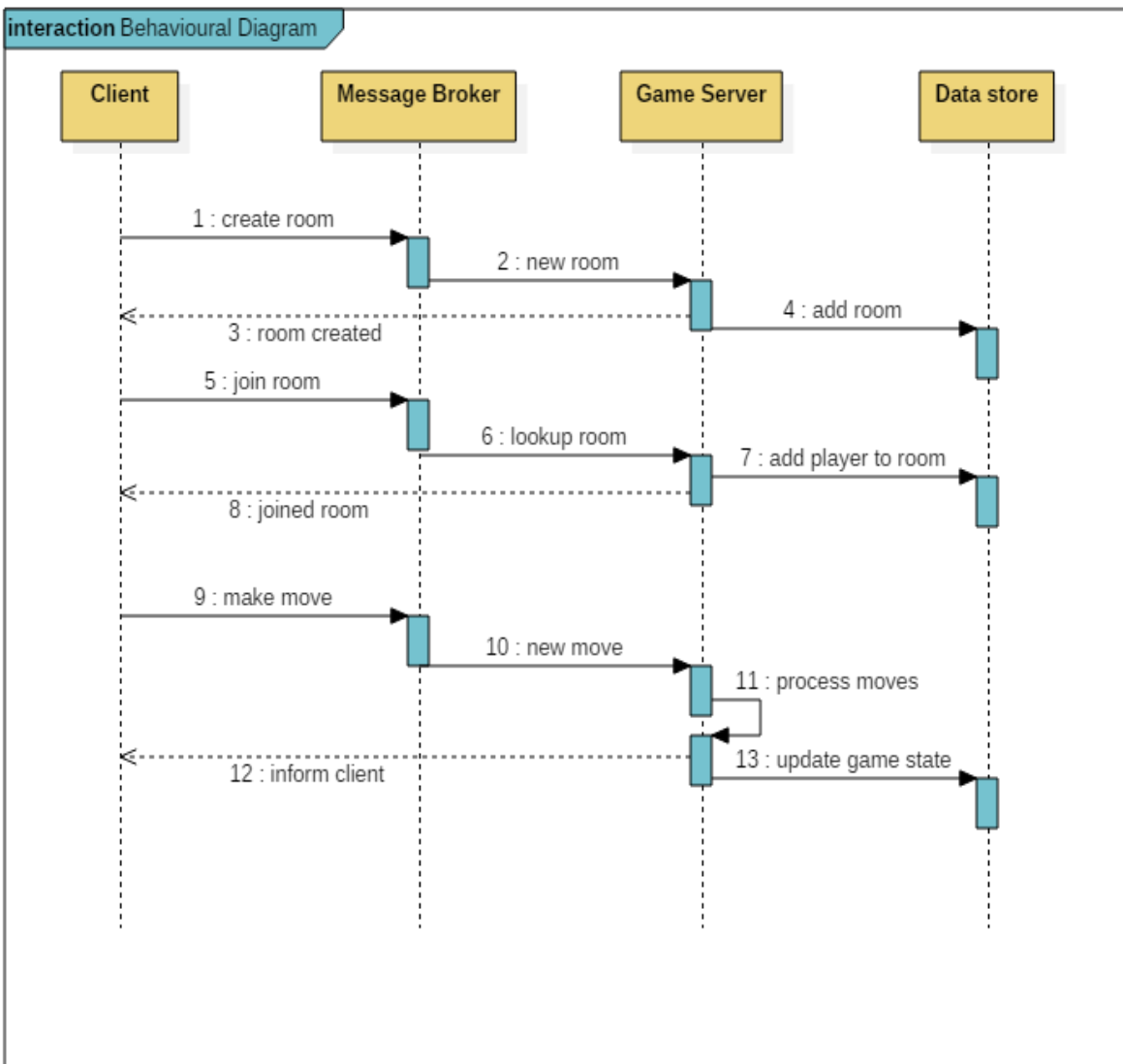
Klijent ima unutrašnju strukturu zasnovanu na MVP (Model-View-Presenter) projektnom obrascu.

Klijent komunicira sa serverom preko Presenter-a koristeći Message Broker. Na game serveru se nalazi Api koji komunicira sa bazom podataka.



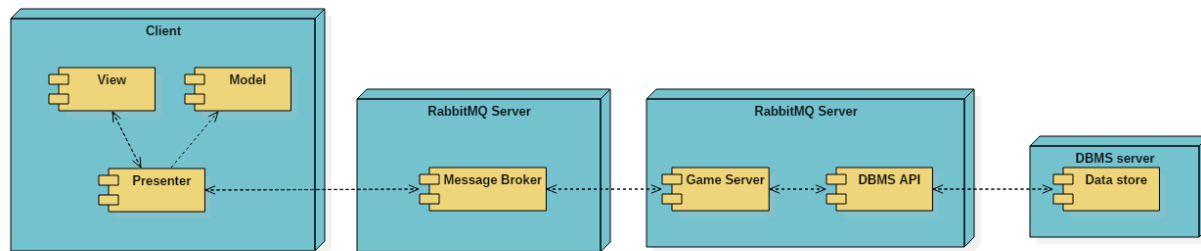
Bihevioralni pogled

Bihevioralni pogled opisuje interakciju između komponenti sistema prilikom povezivanja novog igrača na server. Igrač prvo, ima mogućnost kreiranja sobe ili pridruživanje već postojećoj sobi. Kada se u sobi nalazi dovoljno igrača, vlasnik sobe može započeti igru koja se odvija kroz niz faza koje se smenjuju ciklično.



Alokacioni pogled

Alokacioni pogled prikazuje raspored komponenti na čvorovima tj. na stvarnim računarima. Client se nalazi na desktop računar. Za komunikaciju sa serverom se koristi RabbitMQ server.



Arhitekturni obrasci

- **MVP(Model-View- Presenter)** - modifikacija MVC gde ne komuniciraju direktno View i Model.
- **Publish-Subscriber** - za komunikaciju između klijenta i server; ovaj obrazac je sadržan u Message Broker-u.
- **Modifikacija** u odnosu na standardni obrazac: funkcija Notify ne obaveštava sve klijente, već samo one koji su u istoj sobi.
- **Singleton** - kreiranje konekcije.
- **Repository** – za komunikaciju sa bazom

Specifikacija biblioteka i programskih okvira

- **.NET framework** - izrada klijentskog i serverskog dela
- **WCF framework** – framework za asinhronu komunikaciju između klijenta i servera
- **MSMQ** – message broker server.
- **RabbitMQ.NET** - message broker server.
- **Entity Framework** – ORM alat
- **MySQL DBMS** – server baze podataka
- **ASP.NET** – framework za izradu web API-ja