

Programų sistemų architektūros projektas

Komandos pavadinimas: Such team

Nariai:

- Mark Ganusevič
- Oleg Smelov

Sistemos apibūdinimas

Pavadinimas

Chasing Past

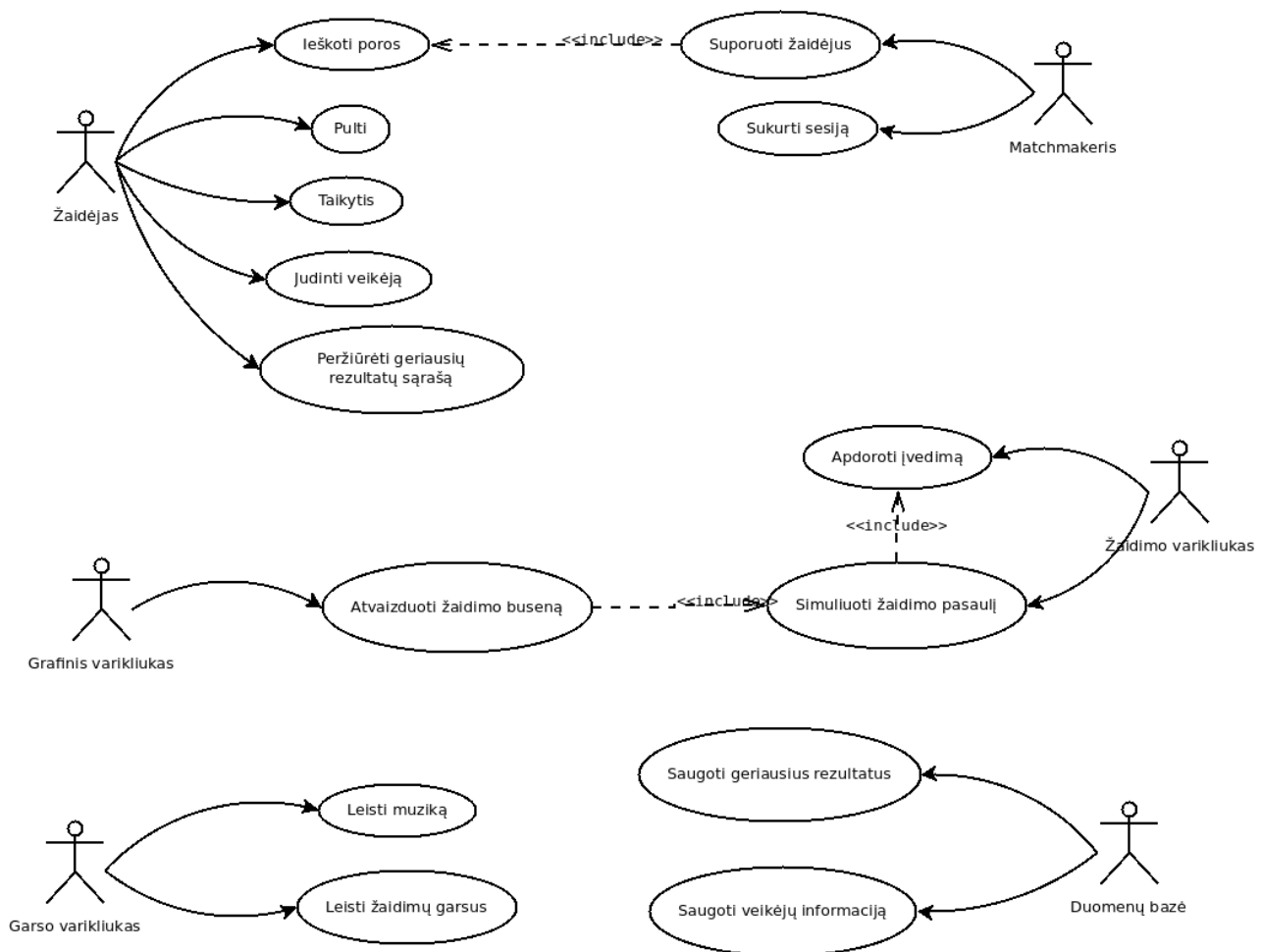
Tikslas

Sukurti naršyklinį kelių žaidėjų žaidimą, simuliuojantį kovą kardais.

Dalykinis aprašymas

Žaidimas turi veikti naujausiose naršyklėse ir neturi reikalauti nieko papildomai instaliuoti. Žaidėjai pereina į žaidimo puslapį, kur yra ieškomas jiems priešininkas (kitas žaidėjas). Suradus priešininką, prasideda žaidimas. Žaidėjai kontroliuoja žmogeliuką, kuris gali bėgioti po jam išskirtą lauką bei atakuoti kitą žaidėją kardu. Sėkmingai atakuotas žaidėjas miršta, jį atakavęs žaidėjas gauna vieną tašką, žaidimas yra restartuojamas, prasideda kitas lygis. Pirmas žaidėjas, surinkęs 10 taškų, laimi dvikovą.

Use case diagrama

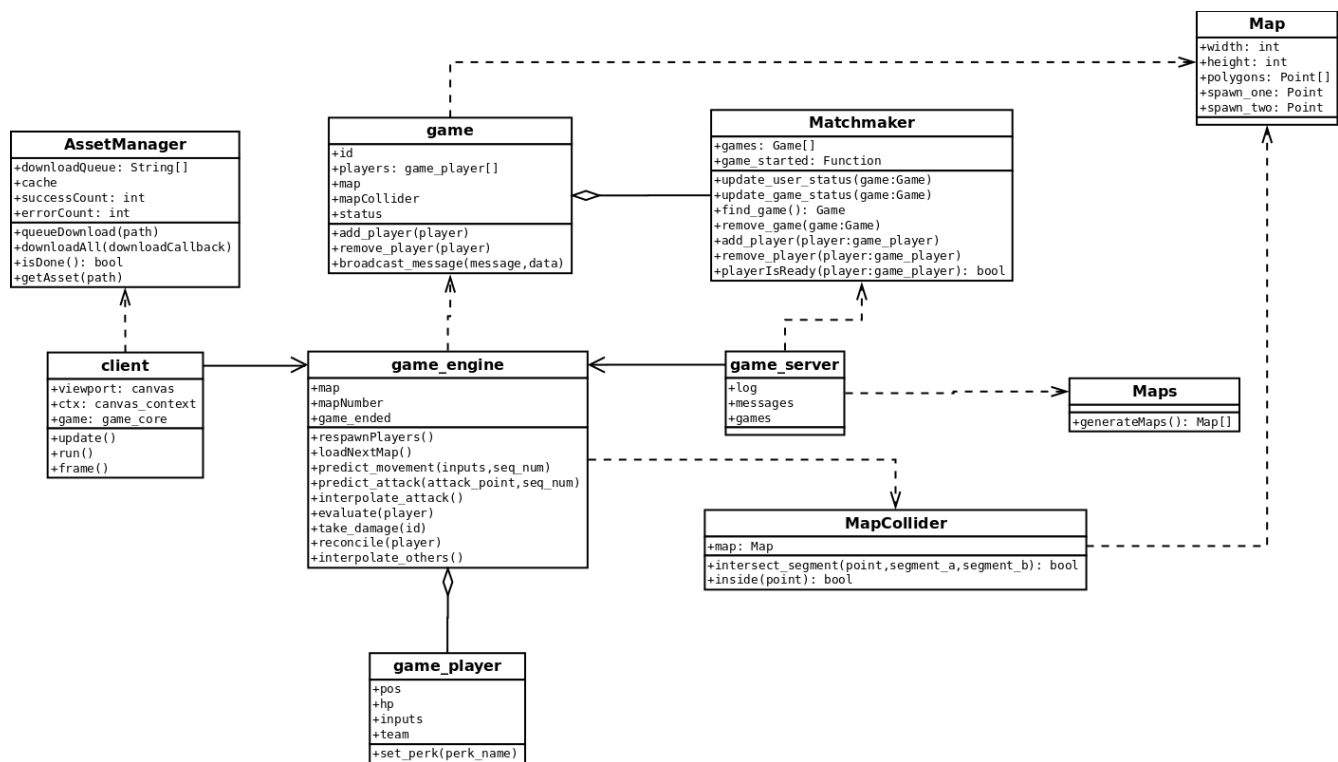


Aktorius	Aprašymas
Žaidėjas	Aktoriui įkrovus puslapį, jam yra ieškoma pora, jei nesurandama, jis įtraukiamas į laukiančiųjų sąrašą. Suporavus du žaidėjus jie gauna galimybę kontroliuoti savo veikėją: judinti, pulti, taikytis.
Matchmakeris	Atsakingas už žaidėjų suporavimą bei žaidimų sesijų kūrimą.
Grafinis varikliukas	Atsako už žaidimo pasaulio atvaizdavimą vartotojo naršyklėje.
Žaidimo varikliukas	Reaguoja į vartotojo įvedimą (puolimas, taikymasis, judėjimas) ir simuliuoja žaidimo pasaulį.
Garso varikliukas	Naudojamas muzikai ir įvairiems žaidimo garsams leisti.

Duomenų bazės	Atsako už duomenų (geriausių rezultatų ir veikėjų informacijos) saugojimą.
---------------	--

Funkcija	Aprašymas
Ieškoti poros	Kai žaidėjas įjungia žaidimą, yra siunčiama užklausa į serverį rasti tam žaidėjui porą (kitą žaidėją).
Pulti	Kai žaidėjas spaudžia pelės mygtuką, jis puola kitą žaidėją. Jei pataiko į kitą žaidėją, jis praranda gyvybes.
Taikytis	Priklausomai nuo pelės pozicijos žaidėjo atžvilgiu yra skaičiuojama puolimo kryptis.
Judinti veikėją	Žaidėją galima judinti naudojant strelytes arba WASD mygtukus.
Peržiūrėti geriausių rezultatų sąrašą	Mygtukas “See highest scores”, kuris leidžia pamatyti geriausių rezultatų sąrašą.
Suporuoti žaidėjus	Matchmakeris suporuoja du žaidėjus kartu.
Sukurti sesiją	Suporuotiems žaidėjams yra sukuriamas atskiras žaidimo pasaulis.
Apdoroti įvedimą	Žaidimo variklius gauna įvedimą (kas ir kada buvo paspausta).
Simuliuoti žaidimo pasaulį	Pagal gautą įvedimą yra priimami sprendimai, apskaičiuojamos žaidimų pozicijos, jų būklė.
Atvaizduoti žaidimo būseną	Naršyklė piešia žaidimo pasaulį.
Leisti muziką	Naršyklė leidžia muziką.
Leisti žaidimo garsus	Tam tikrais momentais yra paleidžiami garso efektai, pavyzdžiui, puolant ar bėgant.
Saugoti geriausių rezultatus	Geriausių rezultatų sąrašas yra saugomas serverio duomenų bazėje.
Saugoti veikėjų informaciją	DB saugo veikėjų informaciją (prisijungimai, atributai, pavadinimai).

Klasių diagrama



Naudojamos technologijos (node.js) leidžia mums naudoti dalį parašyto kodo ir naršyklėje, ir serveryje. Šiuo atveju kai kurios klasės naudojamos kaip kliente, taip ir serveryje.

client klasė veikia tik naršyklėje ir atsako už komunikavimą su serveriu, žaidimo atvaizdavimą naršyklėje.

game_server klasė veikia tik serveryje, atsakinga už šiuo metu žaidžiamus žaidimus, apdoroja klientų užklausas.

game_engine atsakinga už vieno žaidimo būseną, saugo praėjusį žaidimo laiką, apdoroja pagrindinę žaidimo logiką, fiziką, žaidėjų susikirtimus, apdoroja įvedimą.

game_player saugo vieno žaidėjo būseną, jo poziciją, dydį, įvestas komandas ir panašiai.

game saugo informaciją apie žaidimą.

AssetManager atsako už žaidimo failų (paveikslukai, garsas ir t.t.) įkrovimą.

Matchmaker atsako už žaidėjų suporavimą.

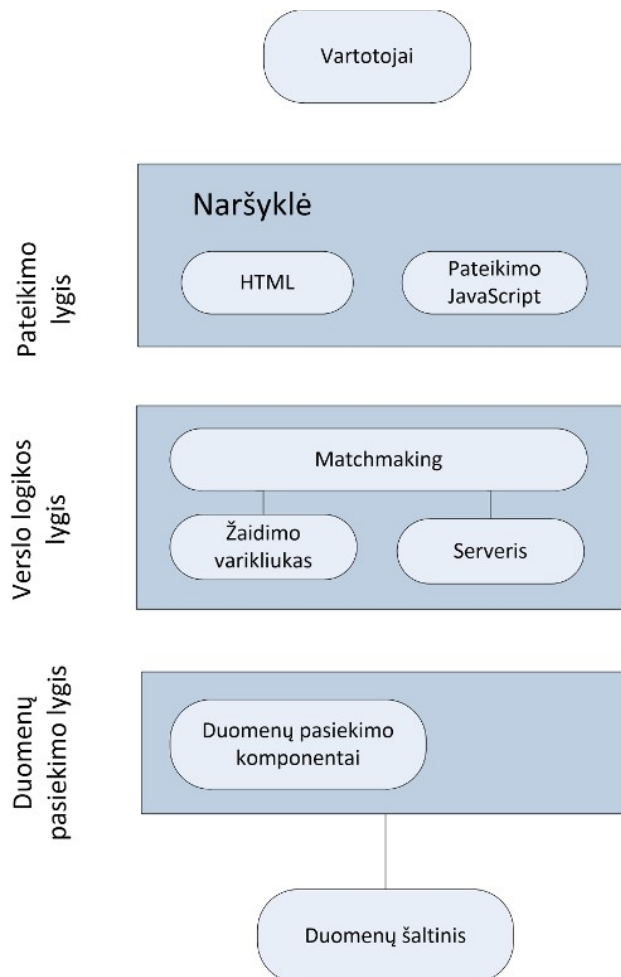
Map saugo informaciją apie žemėlapi.

MapCollider naudojamas kolizijoms su žaidimo pasauliu. Ši klasė apskaičiuoja, ar taškas yra žaidimo

pasaulyje, kur galima būti, ar yra išorėje.

Maps užkrauna žemėlapius.

Sistemos architektūra



Architektūros stilius: klientas ir serveris.

Architektūros šablonas: model-view-controller.

Vartotojai. Žaidėjai, prisijungę per naršyklę prie mūsų žaidimo puslapio.

Pateikimo lygis (view). Į šitą sluoksnį įeina klientų naršyklėse rodomi HTML puslapiai ir vykdomas JavaScript kodas, susijęs su atvaizdavimu.

Verslo logikos lygis (controller). Į jį įeina žaidimo varikliukas, aprašantis visą žaidimo logiką bei žaidimo eigos simuliacijos taisykles, sesija, kuri suporuoja prisijungusius klientus ir sukuria jiems atskirus žaidimus.

Duomenų pasiekimo lygis (model). Į jį įeina klasės, skirtos dirbti su duomenimis, esančiais duomenų šaltinyje.

Duomenų šaltinis. Duomenų bazės serveris, mūsų atveju MongoDB.

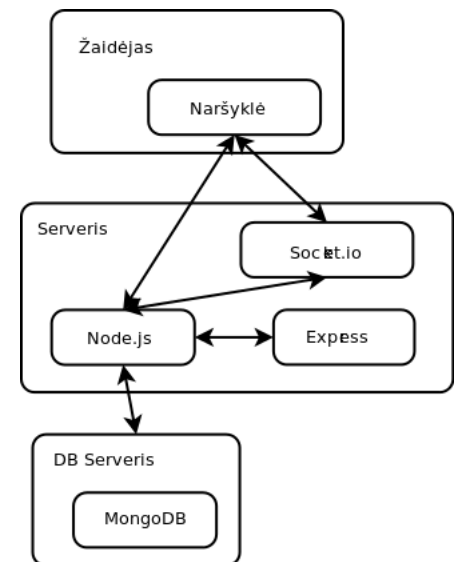
Naudojamos technologijos

Klientas: HTML5, JavaScript, WebSocket.

Serveris: Node.js, Express, Sockets.io; jei prireiks, MongoDB.

Planas antrai iteracijai

- Sukurti prototipą
 - Parengti techninį pagrindą
 - Node.js
 - Express
 - Socket.io
 - Parašyti žaidimo logikos variklį
 - judėjimo įvedimo apdorojimą
 - žaidėjų pozicijų apskaičiavimą
 - Parašyti žaidimų sesijų sukūrimą
 - žaidėjų suporavimą
 - sesijų išskyrimą
 - Vizualinė dalis
 - pozicijų atvaizdavimas ekrane



Planas trečiai iteracijai

- Patobulintas žaidimo variklis
 - Įvedimo apdorojimas
 - Puolimas
 - Taikymasis
 - Kollizijų su žaidimo objektais apdorojimas

- Puolimo ir taikymosi simuliacija
- Patobulintas grafinis varikliukas
 - Galimybė atvaizduoti paveikslukus (sprites)
 - Paveikslukų animacija

Planas ketvirtai iteracijai

- Paruošti ir pritaikyti žaidimo grafiką (reikalingus paveikslukus ir pan.)
- Sukurti žaidimo garso varikliuką
 - Paruošti žaidimo garsus ir muziką
- Žaidimo lygių kūrimas

Planas penktai iteracijai

- Žaidimo variklio patobulinimai
 - Pergalių skaičiavimas
 - Geriausių rezultatų sąrašas
 - Kintami žaidėjų atributai (stats) ir galimybės (abilities)
 - Žaidimo balansavimas

Papildomai sukurtos funkcijos

- Įrankis žaidimų žemėlapiams kurti
- Dinamiškas fonas (raudonas, kai lieka mažai gyvybių)
- Dinamiškas muzikos garsumas (garsiau, kai lieka mažai gyvybių)
- “Charge” sistema – žaidėjas gali atakuoti tik tam tikrą skaičių kartų, kol jam prireiks laukti kito žaidėjo atakos