VIncit ennakkotehtävä 2019
Dokumentaatio
Joonas Ryynänen
050 404 3908
joonas.ryynanen1@gmail.com
jonestpg.github.io

Sovelluksen url: https://thawing-wildwood-25973.herokuapp.com
Repositorion url: https://github.com/JonesTPG/nappipeli

Yleistä:

Ennakkotehtävänä piti toteuttaa moninpelinä pelattava “nappipeli”, jossa yhdistetyt pelaajat painavat ruudulla näkyvää nappia ja näin kasvattavat taustalla pyörivän laskurin arvoa. Joka 100, 200 ja 500 painallus ovat palkinnon arvoisia, ja se pelaaja kuka painaa nappia oikeaan aikaan voittaa tämän palkinnon. Pelaajat näkevät, kuinka monta painallusta seuraavaan palkintoon tarvitaan.

Toteutus:

palvelin: Päätin toteuttaa sovelluksen palvelinpuolen Node.js:llä, ja käyttää sovelluksen reaaliaikaisuuden varmistamiseen socket.io-kirjastoa. Socket.io-kirjaston avulla palvelin ja client keskustelevat toistensa kanssa websocket-yhteyden kautta (ainakin silloin kun se vain on mahdollista), mikä mahdollistaa nopean ja edestakaisen tiedonsiirron. Jos sovellusta olisi lähtenyt toteuttamaan normaaleilla HTTP GET ja POST -kutsuilla, ei sovellus olisi ollut reaaliaikainen ja muutenkin kutsuja olisi pitänyt tehdä jatkuvasti.

Palvelimen logiikka on loppujen lopuksi melko yksinkertainen. Palvelin kuuntelee jatkuvasti websocketin “connect” ja “disconnect” -tapahtumia, jonka perusteella palvelimella säilytettävää pelaajalistaa päivitetään. Kun pelaajalista on päivittynyt, lähetetään clienteille tieto siitä, ja näin pelaajien ruudussa näkyy aina oikein pelaavien käyttäjien määrä.

Näiden lisäksi palvelin kuuntelee tapahtumaa, jonka client lähettää silloin kun käyttäjä on painanut nappia. Palvelin tarkistaa, tuliko painalluksella voitto ja mikäli tuli niin voittajan tiedot tallennetaan.

client: Sovelluksen client-side on toteutettu React.js-kirjastolla, jossa lisämausteena Materialize.css-tyylikirjasto. Käyttöliittymä on jaoteltu useisiin pieniin komponentteihin, niin kuin React.js:n filosofiaan kuuluukin. Lisäksi avattava websocket-yhteys avataan clientillä vain kerran react-sovelluksen juuressa, ja sitten kyseinen yhteys “passataan” alaspäin niille komponenteille jotka tapahtumia kuuntelevat ja lähettävät palvelimelle.

Asentaminen/ajaminen:

Sovellus on julkaistu toimivana versiona suoraan herokuun (https://thawing-wildwood-25973.herokuapp.com), jossa peliä siis voi käydä pelaamassa. Mikäli sovellusta haluaa kokeilla lokaalisti, voi kloonata git-repositorion ja käydä muuttamassa reactin juuresta websocket-connection herokun osoitteesta localhostin osoitteeseen. (`http://localhost:5000`). Tämän jälkeen tulee sekä client- että palvelinpuoli asentaa komennolla npm install, ja lisäksi client puoli buildata komennolla npm run build. Sitten sovelluksen voi käynnistää localhostin porttiin 5000 komennolla node index.js.

Oppimiskokemus:

Ennakkotehtävän vaikeustaso oli mielestäni sopiva, ja tehtävän tekeminen oli mielenkiintoista. Vaikka en ns. reaaliaikaisia sovelluksia ennen ollutkaan tehnyt, niin tajusin heti että normaaleilla HTTP API-kutsuilla ei tällaista sovellusta ole järkevää tehdä. Olin lisäksi kuullut socket.io-kirjastosta, ja dokumentaatiota selatessa huomasin, että tällä kirjastolla sovelluksen toteutus käy suhteellisen kivuttomasti.

Pääpainoni sovellusta tehdessä oli saada siitä ensin ns. MVP (minimum viable product), jossa kaikki päätoiminnallisuudet toimivat. Lisäksi painotin sitä, että teen koodista mahdollisimman luettavaa ja ymmärrettävää varsinkin client-sidella. React-sovelluksissa on tärkeää, että komponentit pelaavat järkevästi yhteen, muuten voi helposti tulla tilanne jossa päivitysten/muutosten tekeminen ei onnistu kun sovelluksen rakenne on liian sekava.

Puutteet/parannukset:

-palvelinpuolen toteutuksessa ei tarpeeksi error handlingia
-käyttöliittymä voisi olla modernimpikin
-käyttäjän on mahdollista huijata sovelluksessa, jos hän simuloi napin painallusta sopivalla websocket-tapahtumalla. Tämän voisi estää palvelinpuolella sillä, että tarkastaisi milloin käyttäjä viimeksi on painanut nappia. Tuo aikaraja tosin pitäisi olla aika pieni (n. 100ms), koska hiirtä pystyy halutessaan rämpyttää aika nopeasti.