

Dokumentaatio – Doctor Games

1. Tavoite

Lähdin tekemään peliä, koska Phaserin käyttö oli mukavaa ja pelin ympärille saa kasattua kiinnostavaa tietoa esimerkiksi high-scoreista. Tavoitteena oli, että peliä pystyisi pelaamaan ja käyttäjien tulokset tallentuisivat tietokantaan, mikäli käyttäjä tekisi uuden ennätyksen. Lisäsin peliin elementtejä BBC:n Doctor Who -sarjasta, jota tykkään seurata.

2. Työn kulku

Aloitin tekemällä peliä Phaserillä, käyttäen apuna Phaserin nettisivua (<http://phaser.io/tutorials/making-your-first-phaser-3-game>) ja loin mieleiseni pelin. Sain käytettyä tähän yllättävän paljon aikaa ja en päässyt ihan tavoitteeseeni, että hahmo voisi ampumalla tuhota Dalekeja, vaan käytännössä niitä joutuu vain väistelemään kerätessä Sonic:eja ja Dalekeja. Tarkoituksena on kerätä mahdollisimman paljon pisteitä keräämällä Sonic:eja ennen kuin Dalekit tuhoavat ne ja väistellä pommeja.

Tämän jälkeen aloin luomaan runkoa sivulle ja tietokannalla käyttämällä Express Local Library tutoriaalia pohjana. Tämä osoittautui kuitenkin huonoksi ratkaisuksi ja ylhäältä alaspäin rakentamiseen sain käytettyä paljon aikaa ja lopulta päädyinkin käyttämään pelkkää "skeletonia" ja jättämään tietokannan käytön pois. Samalla yksinkertaistin tiedostorakennetta ja en enää yrittänyt tehdä erillisiä controllereita ja route-tiedostoja eri osille sivustoa. Jälkeenpäin ajateltuna olisi vaan pitänyt alkaa rakentaa alhaalta päin alusta asti yrittämättä oikoa, olisin varmaan säästänyt aikaa ja aika monilta ongelmilta.

Loin uudet view-t ja laitoin ne reitittymään kaikki yhden tiedoston (routeController) kautta. Kaunistelin pug-view:ja käyttämällä Materializea. Tämän jälkeen aloitin Vuen käyttämisen toteuttamisen Vue:lla ja loin vastaavat js-tiedostot eri view:lle, jotta pystyin liikuttelemaan käyttäjä ja pistedataa score-listan luomiseen. ScoreBoardin toteutus Vuella on vielä aika alkeellinen, käytännössä muunnelma kauppalistasta mutta toimii kuitenkin

Käytin välimuistia ja JSON:ia käyttäjän parhaan tuloksen tallentamiseen highScore-muuttujaan. LocalStorageen tallentuu viimeisin pistemäärä, korkein saavutettu pistemäärä ja pistelistaan syötetyt tulokset.

3. Ohjelman rakenne

- Pääsivu eli peli (/)
- Submit-sivu käyttäjän tietojen tallentamiseen, kun tehdään uusi ennätys
- Scoreboard-sivu, jossa voi tarkastella pelaajien tuloksia listasta

Pääsivulla oleva peli linkittyy doctorGame.js tiedostoon, jossa on Phaser peli. Sekä Submit että scoreBoard.pug käyttäjät scoreBoard.js -tiedostoa tiedon liikuttelemiseen Vuella. Submit-sivulla Vue näyttää siihen mennessä parhaan pistemäärän ja käyttäjä voi tallentaa oman nimimerkkinsä ja pistemääränsä.

4. Toteutetut toiminnallisuudet

Responsiivinen ulkoasu	5
JSONnin (/XML:n) käyttö tiedon liikuttelussa	3
Front-controllerin käyttö (esim. Vue)	5
Välimuistin käyttö	2
Canvas-elementin käyttö	3
Kattava dokumentaatio	5
Tavoiteltu pistemäärä 20	

5. Reflektio

Opin tämän kurssin ja työn aikana todella paljon uutta ja en osannut sivuainetaustalta arvioida oikein kuinka kuormittava tämä kurssi tulisi olemaan. Välillä oli kaikkien eri tekniikoiden kanssa pää aika pyörällä mutta uskon, että kurssista jäi hyvä kuva, tuntuma ja ymmärrys siitä miten nettisivut toimivat ja miten niiden rakentaminen tapahtuu. Kurssin suorittaminen intensiivisemmin periodin aikana olisi kyllä helpoittanut asiaa aika paljon mutta tulipahan taisteltua tässä lopussa sitten eri palojen kanssa kunnolla.