

WWW-sovellukset

Harjoitustyödokumentti

Alexi Ruokoniemi 0452334

Sisällysluettelo

1. Aihe	3
2. Toteutus	3
2.1 Repository	3
2.1 Front-End	4
2.2 Back-End	5
2.3 Tietokanta	6
3. Testaus	6
4. Vaatimukset/pistemäärä	7

1. Aihe

Työn aiheeksi valikoitui esimerkkiainetta 3 mukaillen Phaser-pelin ja sen sivuston luonti. Peli on ylhäältä kuvattu autoilupeli, jossa tarkoituksena on ajaa määrätty reitti mahdollisimman nopeasti. Tarkemmat kuvaukset pelin mekaniikoista löytyvät sivustolta.

Aihevalinnan taustalla oli se, että Phaser-viikkotehtävän teko osoittautui miellyttäväksi. Työn tarkoituksena oli luoda kaikin puolin paranneltu versio tuosta viikkotehtävänä luodusta pelistä. Koska kilpailulliset pelit ovat hauskoja, piti peliin tehdä myös multiplayer ja näiden pelien tuloksiin perustuva rating-systeemi.

2. Toteutus

2.1 Repository

Repositoryn rakenne on seuraavanlainen:

Kansio	Sisältö
bootstrap	Bootstrap 4:n vaatimat JS-skriptit, CSS-tiedostot
components	AngularJS:n controller+view parit kansioittain. Servicet/factory omassa kansiossa. Juurikansiossa Angular-moduuli ja applikaation root-controller.
CSS	CSS

game	assets-kansiossa pelissä käytetyt spritet, spritesheetit states-kansiossa pelin tilat omissa tiedostoissaan. Juurikansiossa apuskripti multiplayer-serverin kanssa tiedon välittämiseen.
js	Kaikki käytetyt JS-frameworkit, kirjastot.
misc	Tietokannan luonnissa käytetyt MySQL-käskyt.
node_modules	Node.JS moduulit.
php	PHP-kontrollerit omassa kansiossaan. Front controller + apuskriptit juurikansiossa.

2.1 Front-End

Sivuston selainpuolella on käytetty AngularJS:ää. Sivusto on jaettu erilaisiin komponentteihin joilla on omat controllerinsa ja näkymänsä. Sivusto on toteutettu SPA (Single Page Application)-periaatteella. Tämän helpottamiseksi on käytetty ui-router -laajennosta, joka mahdollistaa esim. näkymien periyttämisen ja URL:n päivittämisen eri tiloissa. Serviceissa on toteutettu ajax-kutsuja, joita voidaan tarvita useimmissa controllereissa. Pelaajan rating-käppyrän piirtämiseen tilastosivulla on käytetty Chart.js -laajennosta, joka piirtää annetun datan canvas-elementtiin.

Bootstrap v4 käytetty luomaan ulkoasusta responsiivinen. Kustomoidut css-säännöt löytyvät tiedostosta /css/custom.css. Sivustolla on implementoitu muutamia CSS3-transitioita painikkeisiin ja navigaatioelementteihin.

Itse pelin luomiseen on käytetty Phaseria. Peli on luotu kiinteän kokoiseksi, eikä siten ole responsiivinen. Phaser ei tarjoa mahdollisuuksia muuttaa tätä jälkeen päin, eikä peliä ole tosin muutenkaan tarkoitettu pelattavaksi pienemmillä laitteilla. Pelissä käytettyjen Internetissä vapaaseen käyttöön jaettujen sprite-kuvien tekijät on mainittu sivuston Credits-välilehdellä. Muut käytetyt graafiset elementit on luotu itse.

2.2 Back-End

Palvelinpuolella tiedon hakemiseen ja varastointiin tietokantaan ja käyttäjän autentikointiin on käytetty vanilla PHP:ta. PHP skriptit toteutettu Front-Controller periaattella: FrontController.php ottaa vastaan POST-pyynnössä saadun actionin, yrittää etsiä tätä vastaavan controllerin ja delegoi tehtävän eteenpäin controllerille.

Node.JS + Express + Socket.io yhdistelmää on käytetty moninpelin mahdollistamiseen pelissä. Palvelin yhdistää kaksi samaan aikaan peliä hakevaa pelaajaa samaan peliin. Pelin aikana palvelin välittää pelaajilta saadut tiedot esim. sijainnista (paikkaxy, kulma, aika jolloin tieto saatu) ja sen hetkisestä tilasta (romuna, maalissa) toisilleen. Yritetty ottaa myös huomioon pelistä poistuminen: pelit ilmoitetaan automaattisesti 10min alkamisen jälkeen, jolloin poistuvalla pelaajalle lisätään enimmissään tasapeli. Palvelin-pelaaja implementaatio on huijaamista ajatellen naiivi: palvelin vain välittää käyttäjiltä tulleen tiedon eteenpäin toiselle käyttäjälle.

2.3 Tietokanta

MySQL-tietokanta, joka sisältää neljä taulua.

Taulu	Rivin sisältö
user	Käyttäjän autentikaatiotiedot (salasana hashattu+suolattu). Käyttäjäkohtaiset pelitiedot kaikilta ajoilta.
spgame	Yksittäisen yksinpeli-tilassa pelatun pelin tiedot sisään kirjautuneelle käyttäjälle.
mpgame	Yksittäisen moninpeli-tilassa pelatun pelin tiedot sisään kirjautuneille käyttäjille.
ratinghistory	Käyttäjän rating-luku ajanhetkellä x.

Tietokantaan on luotu kaksi triggeriä: ratinghistory-tauluun uuden rivin lisääminen kun user-tauluun lisätään rivi (käyttäjä rekisteröityy, lisätään alkuarvo) uuden rivin lisääminen kun user-tilussa oleva rating-sarake muuttuu (käyttäjä pelannut pelin moninpeli-tilassa).

3. Testaus

1. Jotta multiplayer-ominaisuudet toimivat, syötä Cloud9:n ubuntu-terminaaliin komento `node server.js`
2. Valitse Run Project
3. Avaa sivusto Apache:n ilmoittamasta sivustosta, tai linkistä <https://www.sov-htyo-a-ruokoniemi.c9users.io/>

Multiplayerin testaamiseksi voi sivuston avata kahdella selaimen välilehdellä.

Palvelimen koodiin on tarkoituksella jätetty mahdollisuus pelata itseään vastaan.

4. Vaatimukset/pistemäärä

Ominaisuus/vaatus	Pisteet	Perustelut
Responsiivinen ulkoasu	5	Hyödynnetty Bootstrap:n gridejä luomaan responsiivinen ulkoasu.
Tietokannan käyttö tietojen tallennukseen	5	Käyttäjien tiedot sekä pelin erilaiset tilastotiedot tallennetaan MySQL-tietokantaan.
Käyttäjän autentikointi	5	Tallennettut salasanat hashataan ja suolataan PHP:n tarjoamilla funktioilla. Käyttäjä autentikoidaan sisäänkirjautumisen yhteydessä.
Front-controller	3	Palvelinpuolella PHP toteutettu Front-Controller periaatteella.
MVC-mallin mukainen sivusto	5	Selainpuolella käytetty AngularJS.
JSONin käyttö tiedon liikuttelussa/tallennuksessa	3	JSON:ia käytetään tiedon liikuttamiseen Angularin ja PHP-palvelimen välillä.
jQueryn / Reactin / AngularJS:n käyttö	3	Käytetty AngularJS.
Canvas-elementin käyttö	3	Canvas-elementtejä käytetään Phaser-pelissä sekä tilasto-osiossa graafien piirtelyyn Chart.js-kirjaston avulla.
Sivusto ei sisällä yhtään kuvaa, vaan kaikki grafiikka on tehty dynaamisesti CSS:llä ja muilla tekniikoilla.	1	Sivustolla ei kuvia pl. pelin spritet. Grafiikka luotu CSS-gradienilla ja muilla effekteillä.
Ajaj (/ajax) -ohjelmoinnin hyödyntäminen	3	Ajaxia käytetään aina (http-funktio) kun otetaan yhteys PHP-palvelimeen. Esim. leaderboards-sivu, statistics-sivu.
(Lähes) reaaliaikainen kommunikointi AJAX:lla esim. lentely- tai toimintapeli-tyyliseen projektiin tai chat-tyyliseen projektiin (WebSocket +3 pts päälle, mutta en lähtisi	6	Ei toteutettu AJAX:lla vaan NodeJS+Express+Socket.io combolla, mutta kelvanee(?) Socket.io:sta laskettu samat pisteet kuin WebSocket:n käytöstä.

WebSoketteja aivan kylmiltään tekemään)		
	= 42	