



Hahmon kehityssivun suunnittelu ja toteutus: kokemuksia web-ohjelmoinnista ja tiimityöstä etäympäristössä

Juha Jokinen

2025 Laurea



Laurea-ammattikorkeakoulu

Hahmon kehityssivun suunnittelu ja toteutus: kokemuksia web-ohjelmoinnista ja tiimityöstä etäympäristössä

Juha Jokinen
Tietojenkäsittely
Opinnäytetyö

Juha Jokinen

Hahmon Kehityssivun Suunnittelu ja Toteutus: Kokemuksia Web-ohjelmoinnista ja Tiimityöstä Etäympäristössä

Vuosi

2025

Sivumäärä

47

Opinnäytetyö tehtiin Psyche's Royale Gaming ry:ssä toteutetun ALT Zone -mobiilipelin verkkosivuprojektista. Työ tarjosi mahdollisuuden kehittää osaamista ja työtahokkuutta ohjelmoinnissa sekä projektinhallinnallisissa menetelmissä. Opinnäytetyö antaa toimeksiantajalle syvämpään näkökulmaan harjoittelijan työskentelyprosessista ja tiimityöstä etätyöprojektissa.

Kehittämistehtävänä oli hahmojen kehityssivun suunnittelu ja toteutus hyödyntäen moderneja teknologioita, joita ovat React, TypeScript, Next.js ja FSD-arkkitehtuuri. Työssä analysoitiin oppimisprosessia uuden teknologian, kuten Figma sekä projektinhallinnallisen ja viestinnällisten menetelmien parissa toimimista etätyöskentelyssä.

Viitekehiksenä toimivat web-ohjelmointiprojektin erityispiirteet, frontend-arkkitehtuurin jäsentämiseen FSD-arkkitehtuurilla, tiimityö etätyöympäristössä ja TypeScript-syntaksin omaksuminen.

Menetelmällisinä ratkaisuna käytettiin tehtävien kirjaamista ja viikoittaista analysointia, missä tarkasteltiin tehtyjä ratkaisuja, saavutettuja tuloksia sekä kehityskohteita. Työ painotui käytännönläheiseen oppimiseen ja jatkuvaan palautteen hyödyntämiseen.

Keskeisiä tuloksia olivat hahmojen kehitys - verkkosivun toteutus ja merkittävä oppimiskäyrä ohjelmoinnissa sekä työkalujen käytössä. Tulosten arviointi osoitti, että kyettiin omaksumaan nopeasti uusi teknologia ja etätyössä tapahtuva viestinnän kulttuuri. Työ loi vahvan pohjan ammatilliselle kehittämiselle.

Asiasanat: web-ohjelmointi, TypeScript, FSD-arkkitehtuuri, Next.js, Figma käyttö

Juha Jokinen

Design and implementation of the character development page: experiences in web development and teamwork in a remote environment

Year

2025

Pages

47

The thesis was conducted as part of the ALT Zone mobile game website project implemented at Psyche's Royale Gaming ry. The work provided an opportunity to develop expertise and efficiency in programming as well as project management methods. The thesis offers the employer a deep insight into the trainee's work process and teamwork in a remote work project.

The development task involved designing and implementing a character development page using modern technologies, including React, TypeScript, Next.js, and the FSD architecture. The learning process of new technologies such as Figma, as well as working with project management and communication methods in a remote work environment, was analyzed in the study.

The framework was based on the characteristics of web development projects, structuring frontend architecture with the FSD model, teamwork in a remote work environment, and the adoption of TypeScript syntax.

As a methodological approach, task documentation and weekly analysis were utilized to review implemented solutions, achieved results, and areas for improvement. The focus of the work was on practical learning and continuous feedback utilization.

The key outcomes included the implementation of the hero development webpage and a significant learning curve in programming and tool usage. The evaluation of the results demonstrated the ability to quickly adopt new technologies and adapt to remote work communication culture. The work laid a strong foundation for professional development.

Keywords: web development, TypeScript, FSD architecture, Next.js, Figma usage

Sisällys

1	Johdanto	6
1.1	Keskeiset ammattikäsitteet	6
2	Nykytilanne	8
2.1	Sidosryhmät	9
2.2	Vuorovaikutustaidot	10
2.3	Kehittäminen	10
2.4	Tavoitteet	10
3	Päiväkirjaraportointi	10
3.1	Seurantaviikko 1	11
3.2	Seurantaviikko 2	16
3.3	Seurantaviikko 3	19
3.4	Seurantaviikko 4	23
3.5	Seurantaviikko 5	27
3.6	Seurantaviikko 6	32
3.7	Seurantaviikko 7	36
3.8	Seurantaviikko 8	41
4	Yhteenveto ja pohdinta	44
	Lähteet	46
	Kuvat	47

1 Johdanto

Tässä opinnäytetyössä seurataan web-ohjelmistokehittäjän arkea vapaaehtoistyövoimin toteuttavassa verkkosivustossa. TPRG - Psyche's Royale Gaming ry. toteuttaa mobiilipeliä ja siihen verkkosivustoa harjoittelijoiden ja vapaaehtoisten työpanoksella. Yhteisö koostuu 90 henkilöstä, jossa on graafikoita, käyttöliittymäsuunnittelijoita, pelisuunnittelijoita ja ohjelmiojia. Ohjelmointi koostuu kolmesta ryhmästä: palvelin-, verkkosivu- ja peliohjelmiojista. Verkkosivuja on toteuttamassa noin kymmenen henkilöä. Päiväkirjamuotoinen opinnäytetyö kirjoitetaan aikavälillä 10.10.2024 - 10.1.2025. Opinnäytetyöhön liittyvän päiväkirjan osuus on 10 viikkoa. Päiväkirjamuotoisen opinnäytetyön raportointi tapahtuu päivittäisellä työtehtävien kuvaamisella ja viikoittaisella raportilla, jossa analysoidaan kulunutta viikkoa.

Työtehtävät koostuvat sekä itsenäisestä työskentelystä että ryhmätöistä, ja niitä voi valita oman osaamisen mukaan. Ryhmätyöaiheen määrittää vastaava ohjelmoija, joka tuntee harjoittelijoiden osaamistason. Minulle annettiin haastavampi tehtävä kahden hengen ryhmässä, ja työskentelymme kattaa sekä suunnittelun että toteutuksen. Ohjelmointityö tehdään itsenäisesti, mutta Discordin kautta voi olla yhteydessä muihin tiimin jäseniin ja keskustella teknisistä ratkaisuksista. Verkkosivujen kehittäjiltä odotetaan Node.js- ja React.js-osaamista sekä hyvää ymmärrystä verkkosivujen arkkitehtuurista.

1.1 Keskeiset ammattikäsitteet

Frontend tarkoittaa palvelin - asiakas arkkitehtuurissa asiakasta eli käytännössä verkkoselain osuutta verkkopalvelussa (W3Schools 2024a). Tyypillisesti ammattikielessä client-sanaa ei käännetä, vaan se on käsitteen nimi sellaisenaan.

JavaScript (JS) on kevyt, tulkattu (tai just-in-time-käännetty) ohjelmointikieli. Vaikka se tunnetaan parhaiten verkkosivujen skriptikielenä, sitä käytetään myös monissa muissa ympäristöissä, kuten Node.js:ssä, Apache CouchDB:ssä ja Adobe Acrobatissa. JavaScript on prototyyppipohjainen, moniparadigmaattinen, yksisäikeinen ja dynaaminen kieli, joka tukee oliopohjaisia, imperatiivisia ja deklaraatiivisia (esim. funktionaalinen ohjelmointi) tyylejä. (MDN 2024a.)

TypeScript on JavaScript, jossa on tyyppien syntaksi. TypeScript on vahvasti tyyppitetty ohjelmointikieli, joka pohjautuu JavaScriptiin ja tarjoaa parempia työkaluja kaikenkokoisiin projekteihin. (TypeScript 2024.)

React on JavaScript-kirjasto käyttöliittymien rakentamiseen. Reactia käytetään yksisivuisten sovellusten (single-page applications) luomiseen. React mahdollistaa uudelleenkäytettävien käyttöliittymäkomponenttien luomisen. (W3School 2024b.)

React Hook. Hookit mahdollistavat erilaisten React-ominaisuuksien käytön komponenteissasi. Voit käyttää joko sisäänrakennettuja Hookeja tai yhdistellä niitä luodaksesi omia. (React 2024a.)

FSD - Feature-Sliced Design. (FSD) on arkkitehtoninen menetelmä front-end-sovellusten jäsentämiseen. Yksinkertaisesti sanottuna se on kokoelma sääntöjä ja käytäntöjä koodin järjestämiseen. Tämän menetelmän päätarkoitus on tehdä projektista helpommin ymmärrettävä ja vakaampi jatkuvasti muuttuvien liiketoimintavaatimusten edessä. (FSD 2024.)

Figma on monipuolinen yhteiskäyttöinen verkkosovellus käyttöliittymäsuunnitteluun. Sillä pystyy muun muassa generoimaan CSS-tyylitiedostoja HTML-sivun käyttöön. (Figma 2005.)

Git. Nykyään Git on ylivoimaisesti laajimmin käytetty moderni versionhallintajärjestelmä. Lukemattomat ohjelmistoprojektit, niin kaupalliset kuin avoimen lähdekoodin projektit, käyttävät Gitia versionhallintaan. Lisäksi Git toimii erinomaisesti monilla käyttöjärjestelmillä ja integroiduilla kehitysympäristöillä (IDE). Gitin hajautettu arkkitehtuuri tekee siitä esimerkin hajautetusta versionhallintajärjestelmästä. Gitissä jokaisella kehittäjällä on oma työversio, joka sisältää täydellisen historian kaikista muutoksista. (Atlassian 2024.)

GitHub helpottaa kehittäjiä jakamaan kooditiedostoja ja tekemään yhteistyötä muiden kehittäjien kanssa avoimen lähdekoodin projekteissa. GitHub toimii myös sosiaalisen verkostoitumisen alustana, jossa kehittäjät voivat avoimesti verkostoitua, tehdä yhteistyötä ja esitellä töitään. (Juviler 2021.)

Branch. Branchillä viitataan git versiohaaraan (git 2025a).

Commit. git commit - komento tallentaa muutokset Git-repositoryyn luoden uuden commitin eli version tai tilannevedoksen (git 2025b).

Push. git push -komento siirtää paikalliset Git-repositorion muutokset (commits) etäpalvelimellä sijaitsevaan repositoryyn (git 2025c).

Merge tarkoittaa kahden versiohallinnallisen haaran yhdistämistä yhdeksi. Tämä mahdollistaa useiden kehittäjien yhteistyön ilman, että heidän työnsä tallentuu päällekkäin.

Pull request. Pyyntö Githubissa yhdistää sivuhaara päähaaraan. Siinä on nimettyjä henkilöitä tarkastamassa koodia. Ja se näyttää muokatut koodirivit koostettuna. Pull request muodostaa syötteen, jossa voi keskustella omista muutoksista. Vastaava ohjelmoija voi kommentoida koodia ja se näkyy omassa kehitysympäristössä.

Renderöidä. Renderöinti tarkoittaa verkkosivukontekstissa HTML sivun luomista. HTML sivu luodaan yleensä javascript-koodilla. Prosessissa tietokone luo näkyvän osan verkkosivusta

näytölle. Renderöinti voi tapahtua palvelimella, verkkoselaimessa tai osa sivusta voi renderöityä palvelimella ja loput selaimessa.

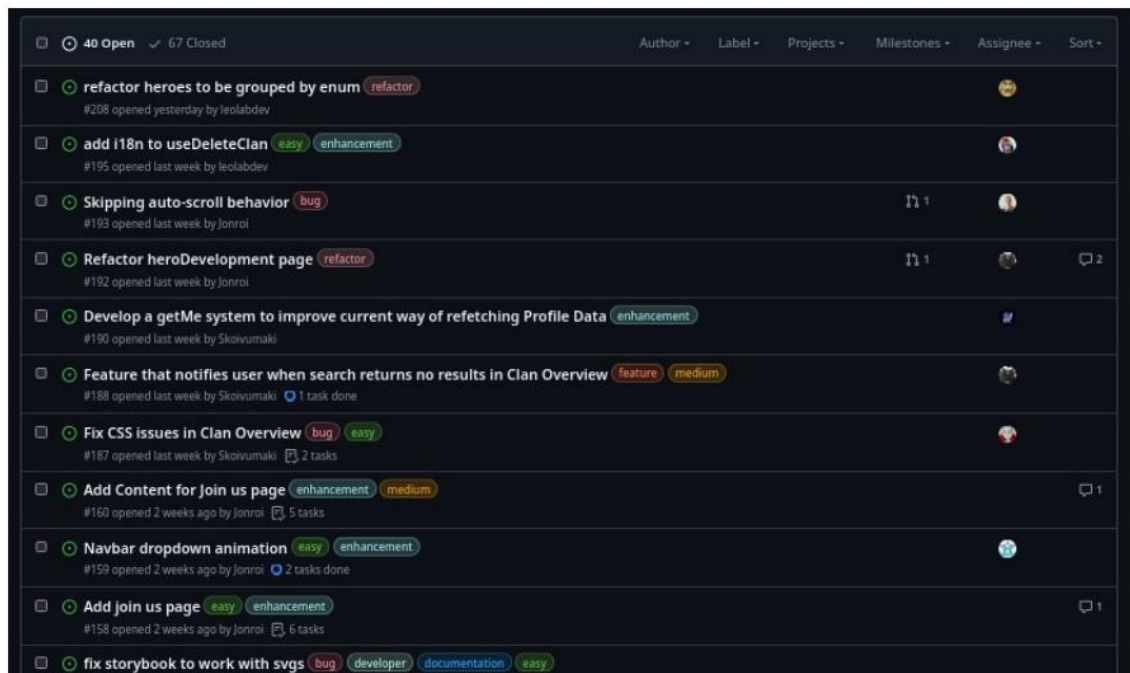
Discord on pikaviestintä- ja VoIP-sosiaalinen alusta, joka mahdollistaa viestinnän äänipuheluiden, videopuhelujen, tekstiviestien ja median kautta. (Discord 2025.)

Kovakoodaus (engl. hardcoding) tarkoittaa ohjelmoinnissa sitä, että tiettyjä arvoja, kuten parametreja, asetuksia tai käyttäjätietoja, asetetaan suoraan ohjelmakoodiin, sen sijaan että ne saataisiin ulkoisista lähteistä, kuten konfiguraatiotiedostoista tai tietokannoista. Tyylit

2 Nykytilanne

Nykyisessä työssäni toimin harjoittelijana verkkosivujen kehittäjänä. Minulla on noin vuoden verran kokemusta React.js:stä harrastuneisuuden kautta sekä vaadittavaa osaamista vastaava koulutus Laureasta. Olen opiskellut it-tradenomin koulutusohjelmaa web-ohjelmointipainotteisesti. Olen suorittanut avoimen yliopiston Full Stack - Websovelluskehityskurssin, joka vastaa sisällöltään vaadittavia teknologioita. Vastaava ohjelmoija mainitsi kyseisestä kurssista, että se takaa hyvät valmiudet työtehtävään.

Työtehtäviä voi valita vapaasti Githubista issue-välilehdeltä lukuunottamatta ryhmätyöaihetta, joka annettiin verkkosivujen kehittäjien palaverissa (kuva 1.) Työtehtävät koostuvat suunnittelusta, toteutuksesta, dokumentoinnista ja testaamisesta. Myös ryhmätyö, joka on laajempi, pilkotaan issueiksi Figma suunnitelman ja verkkosivujen kuvaruutusuunnitelman sekä verkkosivupalavereissa saadun tiedon perusteella. Kaikki työtehtävät menevät issueiden kautta, koska Githubissa issueen voidaan liittää tekijä ja branch. Tämä näkyy sitten myös Githubin projektiosiossa, jossa kullekin projektille on sprint backlog tyyppinen jaettu työtila, jossa voi seurata issueiden vaiheita ja tarkastella issueita. Kun issue on valmis, se pushataan Githubiin ja luodaan pull request, jolloin toteutettu koodi tulee nähtäville ja arvioitavaksi vastaaville ohjelmoijille. Pull requestissa näkyy tehdyt muutokset ja syöte, jossa voi keskustella muutoksista. Vastaavat ohjelmoijat tekevät tarvittaessa change requestin, jos jotakin korjattavaa on. Sitten, kun pull request on kaikkien hyväksymä se mergetään lähdekoodiin ja branch poistetaan. Suunniteltaessa verkkosivukokonaisuutta työkaluna käytetään Figmaa. Figma toimii verkossa ja siinä on jaettu näkymä, jotta useampi voi osallistua yhtä aikaa suunnitteluun. Tekninen toteutus tapahtuu itsenäisesti. Toteutukseen kuitenkin liittyy tuotoksen esittely ryhmässä ja palautteeseen reagointi. Dokumentointi tehdään ohjelmakoodiin JSdock-muotoisena.



Kuva 1 Työtehtäviä GitHubissa.

Typescript syntaksin ymmärtäminen on hyödyksi projektissa, koska projekti on toteutettu typescriptillä, mutta hyvä javascriptin tuntemus riittää, että pääsee alkuun typescriptissä. Itse olen ensimmäistä kertaa tekemisissä typescriptin kanssa tässä projektissa. Onneksi kehitysympäristö tukee sitä niin hyvin, ettei se anna tehdä virheellistä typescriptiä. Myös next.js:n ja FSD-arkkitehtuurin ymmärtäminen ovat projektin kannalta olennaisia. Projektin kannalta on hyvä osata käyttää Figmaa. Figman käytöstä ei ole vielä mitään kokemusta, mutta sitä kehitetty pakostikin ryhmätyön yhteydessä.

2.1 Sidosryhmät

Koska peli on vasta kehitysvaiheessa ja sitä kehittää vapaaehtoistyönä yhdistys, sillä ei ole muita kuin sisäisiä sidosryhmiä. Työryhmädokumentista löytyy lukuisia kategorioita ja Discordissa on runsaasti ryhmiä, sillä tekijöitä on taiteesta tekniikkaan. Lisäksi on oikeakielisyydestä vastaava taho ja käsikirjoittajia sekä pelisuunnittelijoita. Listaan tässä omaa työtäni koskevat sidosryhmät:

- verkkosivu
- serveri
- vastaavat ohjelmoijat

- UI/UX design

- taiteellinen johtaja.

2.2 Vuorovaikutustaidot

Vuorovaikutus tapahtuu kokonaan Discordissa. Discordissa voi lähettää yksityisiä viestejä tai viestittää ryhmälle. Esimerkiksi kerätessäni palautetta viestitän verkkosivu-ryhmälle. Pienemmissä käytännön asioissa laitan yksityisviestiä. Palaverit tapahtuvat ääni ja video streamin avulla. Haasteena on ollut, että olen samaan aikaan aktiivisena kuin muut, sillä vuorokausiryhtämini on epänormaali. Viestintä tapahtuu suomen kielellä, mutta dokumentointi on englanniksi.

2.3 Kehittäminen

Minulle annettiin ryhmätyöaiheeksi verkkosivuston "hahmojen kehittäminen" -osio, jonka suunnittelua ja toteutusta dokumentoin tässä päiväkirjamuotoisessa opinnäytetyössä. Tavoitteena on toteuttaa kyseinen verkkosivuosio sekä kehittää omaa työtehokkuutta ja menetelmiä jatkuvan raportoinnin ja analyysin avulla. Pysin myös löytämään uusia ja parempia työkaluja tai tapoja tehostaa työskentelyäni.

2.4 Tavoitteet

Tämän opinnäytetyön tavoitteena on dokumentoida työharjoitteluni ja toimia referenssinä itseni kehittämisessä. Harjoittelun keskiössä on oman osaamisen vahvistaminen web-ohjelmoijana sekä syvemmän ymmärryksen hankkiminen FSD-arkkitehtuurista.

Harjoittelun aikana saan palautetta jokaisesta palautuksesta, ja jos tuotokseni rikkoo arkkitehtuurin määritelmiä, minun on korjattava virheet. Päiväkirjan pitäminen auttaa jäsentämään työskentelyäni, alleviivaamaan tekemiäni päätöksiä ja tehostamaan oppimista virheistä.

Tavoitteena on laatia raportti, joka auttaa minua tunnistamaan vahvuuteni ja kehityskohteeni valitsemallani uralla. Lisäksi pyrin suunnittelemaan ja toteuttamaan verkkosivuston "Hahmojen kehittäminen" -osion.

3 Päiväkirjaraportointi

Tässä luvussa on opinnäytetyön raportointi. Se koostuu viikon tavoitteista, päiväkohtaisesta raportista ja viikon yhteenvedosta. Kaikkiaan on raportoitu kahdeksan seurantaviikkoa.

3.1 Seurantaviikko 1

Viikon tavoitteena on saada ryhmätyötä eteenpäin. Ryhmätyöaiheena on hahmon kehittäminen, joka kattaa verkkosivun suunnittelun ja toteutuksen. Tavoitteena on tehdä layoutista version tiimille esikatseltavaksi. Tehtävänä on Figma-työkalun opiskelu ja sisällöntuottaminen. Tavoitteena olisi, että pääsisi toteuttamaan verkkosivua, mutta kuitenkin hyvä suunnitelma on ensisijaisen tärkeä

Maanantai 28.10.2024

Päivän tavoitteena on saada yhteistyö toimimaan liittyen ryhmätyöhön ja saada palaveri aikaiseksi projektista vastaavan ohjelmoijan ja ryhmätyöparin kanssa sekä päästä alkuun suunnittelutyössä.

Aloitin työpäivän korjaamalla vian, joka aiheutti piirtohäiriön taulukon reunassa mobiililaitteella. Olin jo aikaisemmin korjannut sen, mutta olin unohtanut tehdä Github pull requestin siitä. Kyseinen branch oli jo niin vanha, että jouduin luomaan uuden branchin. Korjasin sitten virheen siihen ja tein pull requestin.

Minulla oli ollut ongelmia löytää harjoitteluparia nimellä Discordista, mutta sain sitten selvitettyä hänen käyttäjätunnuksen ja sain lähetettyä kaveripyynnön. Toimin sitten välikätenä kun yritettiin sovittaa aikatauluja palaveriin. Opiskelin siinä ohessa projektin arkkitehtuuria. Siinä on omat haasteensa, kuinka pilkkoa toiminnallisuus FSD-arkkitehtuuriin. Toiminnallisuus on nimittäin melko hajalla, että koodia on kohtalaisen haastava tulkita, jotta saa kokonaiskuuvan. Ensimmäinen palaveriajankohta ehdotus ei sopinut harjoitustyöparille. Tämä ehdotti kello 16, joka sitten sopi kaikille. Aloin valmistautumaan palaveriin. Kävin Google-dokumentteja läpi ja kokeilin pelata peliä saadakseni käsityksen hahmon tiedoista. Verkkosivuversiosta pelistä en saanut käyntiin, kun iän validointi ei mennyt läpi. Kännykällä en onnistunut kirjautumaan verkkosivulla luomallani tilillä peliin, joten kirjauduin vieraskäyttäjänä. Peli vaikutti varsin keskeneräiseltä. Palaverista heräsi lisää kysymyksiä. Tuntuu, että en käsitä logiikkaa. Lähetin viestiä aiheeseen liittyen. Katsoin 45 min pitkän tutoriaalivideon Figma työkalun käyttöön.

Tiistai 29.10.2024

Tein koko päivän Figma työkalulla suunnitelmaa. Otin tehtäväkseni suunnitella työpöytäversion hahmon kehitys -sivusta. Aikaa kului myös viestimiseen. Aloitin tekemällä framen, joka vastaa käyttöliittymänäkymää. Pyrin saamaan sen resoluutioltaan vastaamaan yhden suhdetyhteen pikselimäärältään koko ruudun auki olevaa verkkoselainta. Myöhemmin huomasin, kun zoomasin ulos ja näin muiden suunnitelmia, että koko oli aika satunnaista. Siellä saattoi olla esimerkiksi mobiilinäkymä, jonka leveys on 2000 pikseliä. Tärkein tehtävä kuitenkin Figmalla

on projektissamme saada yksimielisyys käyttöliittymäelementeistä. Asetin framelle taustakuvaksi verkkosivun käyttämän taustakuvan ja jäljensin Figma suorakaide- ja tekstityökälulla navigointipalkin vertaamalla auki olevaa verkkosivua Figma näkymään. Seuraavaksi alkoi itse sisällön suunnittelu, joka oli vähän haastavampaa, kun verkkosivun kuvaruutusuunnitelma on sen verran suurpiirteinen. Piti mennä olettamuksella mitä kohta hahmon default-tiedot tarkoittavat. Tein sitten suunnitelman, jossa on kuva ja pylväsdiagrammi hahmon ominaisuuksista. Laitoin suunnitelmasta ruudunkaappauksen taiteelliselle johtajalle

ja tämä totesi suunnitelman hyväksi. Minua häiritsi yksi toiminnallisuus kuvaruutusuunnitelmassa. Aikani viestittyä huomasin, että en ymmärrä toiminnallisuutta. Selvisi, että olin oletanut, että kirjautuessa hahmon kehittäminen näyttää pelaajan hahmogallerian hahmojen tietoja, mutta tarkoitus olikin näyttää samat tiedot kaikille. Eli hahmojen kyvyt ovat lähtötasoissa, joka on kaikille sama ja sitten laskurilla voi suunnitella niiden päivityksiä, niin että valitaan lähtötaso. Keskustellessani aiheesta taiteellisen johtajan kanssa hän ehdotti, että ehdotan ominaisuutta nettisivutiimille. Sain vastauksen, että sisäänkirjautuminen tuo General Data Protection Regulation (GDPR) vaatimukset mukanaan eikä aiheeseen perehtyneitä ole tiimissä. Kuitenkin sisäänkirjautuminen toimii, joten kyllä siellä ainakin access token liikkuu. Kuvassa 2 on hahmotelmaa perustuen kuvaruutusuunnitelmaan. Päätin sitten jatkaa suunnittelua sillä periaatteella, että kaikki tiedot ovat samat kaikille. Kuvaruutusuunnitelmassa lukee ”leveleiden kehitystaulukko näkyville”, joten aloin selvittämään, mitä se mahtaa tarkoittaa. Referenssinä on online taulukko. Tutkin sitä jonkin aikaa ja, kun aloin tekemään sen pohjalta taulukkoa hahmojen kehittäminen -sivulle, online taulukon sisältö oli täysin muuttunut. Joku muokkasi sitä samanaikaisesti. Päätin odottaa seuraavaan päivään taulukon osalta, jos vaikka referenssinä oleva online taulukko olisi valmiimpi.



Kuva 2 Hahmon oletustiedot.

Keskiviikko 30.10.2024

Päivä alkoi koko tiimin palaverilla. Palaveri kesti melkein kaksi tuntia. Siellä ei juurikaan tullut verkkosivuihin liittyvää muutamaa yksittäistä poikkeusta lukuun ottamatta. Lisäksi ilmoitettiin uusista discord kanavista, josta osa koski minua. Palaverissa lähinnä käytiin pelipuolen asioita ja testausta peliin liittyen.

Palaverin jälkeen tarkistin gmail-sähköpostin. Gmail oli ollut jonkin aikaa pimennossa, kun olin ollut keskittyneempi Laurean sähköpostiin. Huomasin kauhukseni, että minulta oli jäänyt huomioimatta github-ilmoituksia. Kahteen pull requestiin oli tullut korjausehdotuksia. Yksi oli build-virhe, jota minun kehitysympäristössä ei havainnut. Olin ylikirjoittanut yhden komponentin export lausekkeen uudemmalla versiolla, kun -deprecated-kansiossa oleva koodi kutsuu vanhempaa versiota. Tämä oli varsin helppo korjata. Toinen oli pelin klaanilistan taulukon reunan paksuus, jonka olin asettanut suhteelliseksi yhden pikselin leveyden sijaan, koska siinä oli ongelmia mobiililaitteilla. Nyt reuna näkyi 4k näytöllä liian paksuna. Korjausehdotuksessa oli ehdotus, miten se kannattaisi tehdä, joten korjasin sen suoraan ehdotuksen mukaisesti. Testasin sen ja vein sen versionhallintaan. Ongelmana oli, että minulla ei ole mahdollisuutta testata 4k-näytöllä, mutta korjaus vaikutti toimintavarmalta.

Korjausten jälkeen jatkoin Figma työtä. Minulle koitui haasteeksi suunnitelmassa esiintynyt piirakkamalli, kun siinä esiintyvät ominaisuudet ovat eri yksikössä eivätkä keskenään vertailukelpoisia. Discordissa viestiteltyäni selvisi, että piirakkamalli kuvaa kyseisten ominaisuuksien

tasoja, mikä nyt tuntuu itsestään selvyydeltä. Saatuani ahaa-elämyksen, tein Figma suunnitelman valmiiksi. Samalla harjoitustyöpari aloitteli mobiiliversion suunnittelua. Laitoin Figma suunnitelmasta viestiä taiteelliselle johtajalle hyväksyntää tai kehitysehdotuksia varten tai mahdollisten väärinkäsitysten oikaisemiseksi, jotta tehtävää saadaan eteenpäin. Lisäksi harjoitustyöpari hyötyi kommentteista tehdessään mobiiliversiota, kun ollaan yksimielisiä sisällöstä.

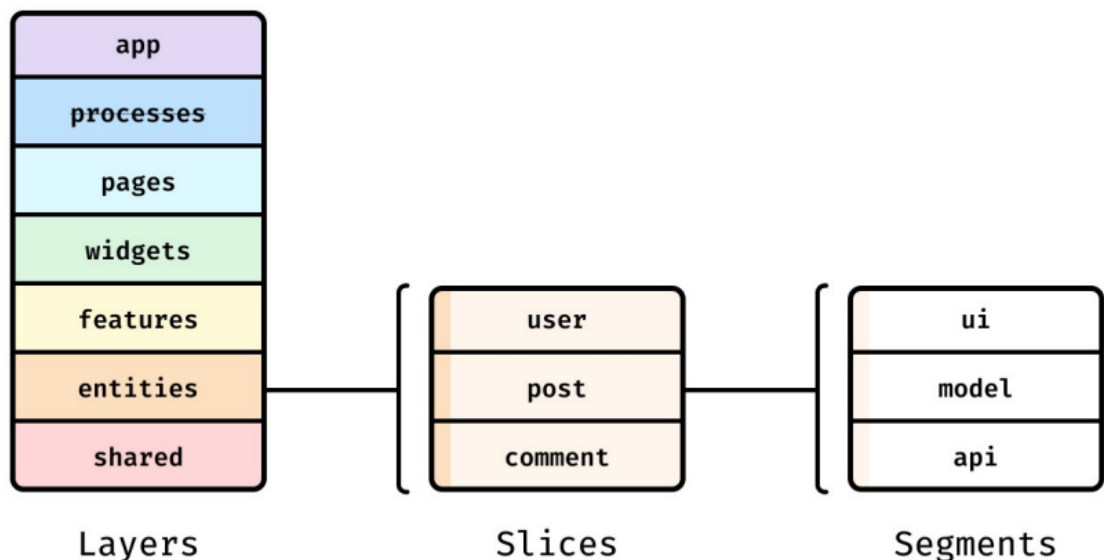
Torstai 31.10.2024

Aloitin työpäivän tutkimalla kuinka sivu Sankarit (Sankarit tarkoittavat pelihahmoja) ovat toteutettu, jotta voin tarvittaessa kehittää kysymyksiä nettisivu-palaveriin liittyen hahmojen (eli pelihahmojen) kehitys sivuun. Sivun data on kovakoodattu front-endiin. Nettisivupalaveri reilu puolitoista tuntia. Palaverissa selvisi ihan oleellisia asioita. Hahmojen pelikehitys on ihan kesken eikä serveripuolen kehittäjien kanssa ole ollut mitään keskustelua aiheesta, niin tulinkin siihen tulokseen, että huomenna, kun aloitan hahmojen kehitys - sivun ohjelmoinnin, koodataan joitakin hahmoja tarvittavalla datalla front-endiin, niin pääsen toteuttamaan käyttöliittymäelementtejä. Lisäksi palaveri auttoi korjaamaan käsityksen hahmon kykyjen kehityspotentiaalin ideasta, mikä on oleellista suunniteltaessa toimintaa ja datatyyppisiä ajatellen. Loppupäivä meni Github pull requestin parissa. Tein muutoksia navbar animaatio - koodiin, niin että collapse-tila pysyy selaimen muistissa, jos painaa refresh- tai back-nappia. Sitten eräs ohjelmistokehittäjä pyysi synkronoimaan minun local-branchin, niin ilmaantui virheitä, koska päivitettyt funktiokutsut eivät päivittyneet tekemiini uusiin versioihin ohjelmakooditiedostoista. Korjattua ne huomasin VSCode problems välilehdellä mystisiä ongelmia, joita ei aikaisemmin ollut eikä ne suoraan liittynyt tekemiini muutoksiin. Ongelmien määrä vaihteli ajonaikaisesti, kun testiympäristö oli käynnissä. Tutkin navigointipalkin osien tyyppisiä ja en oikein saanut kiinni, mistä ongelmat voisivat johtua. Ilmeisesti jossakin vaiheessa, kun järjestelmä rakentaa verkkosivua se syöttää null-arvoisia elementtejä tai väärän tyyppisiä objekteja ja merkijoniarvoja, jotka ei toimi ongelmakohtassa, kun koodi vaatii elementText nimistä avainta objektilta. Laitoin sitten ongelmakohtaan ehdon sisälle, että arvo ei saa olla null tai undefined ja sen pitää olla objekti, jolla on elementText ominaisuus. Aloin katsomaan vielä asiaa, kun kirjoitin tätä raporttia, niin huomasin, että minulla ei ollut @ts-ignore määrettä ongelmakohtassa vertailtaessa synkronoitua koodia omiin versioihin. Eli tällä määreellä typescript kääntäjä oli pakotettu jättämään huomiotta ongelmakohdat. Jää nähtäväksi tuleeko asiaan liittyen korjausehdotuksia pull request syötteessä. Kyllä siellä näkyy jo, että muutoksia on pyydetty, mutta ei ole kuvattu mitä muutoksia. Seuraavalla kerralla laitan synkronoidun koodin näkymään esille samasta kohtaa, jos työstän eri versiota tiedostosta. Asia jonka unohdin palaverin jälkeen oli päivittää työryhmä online dokumenttiin omat tiedot.

Perjantai 1.11.2024

Aloitin päivän tutkimalla Githubissa navigointipalkki-collapse animaatio branchia. Siellä oli edelleen ”pyydetään muutoksia” eikä mitään kommenttia. Tosin siellä oli linkki kommenttiin, jossa luki katso kommentit. Kommenteissa oli kohta issuen määrittelystä, jossa mainittiin, että pitäisi esitellä komponentille parametri, jolla collapse- ja expand tiloja hallitaan. Minun toteutuksessa se oli tehty erillisellä tilakomponentilla. Mietin, että onko se sitten toteutustavasta kiinni, että pull request ei mene läpi. Käyttämäni tapa ei ollut monimutkaisuudeltaan mitenkään eriarvoinen ehdotettuun ja siinä selvisi itse asiassa vähemmällä määrällä koodirivejä.

Siirryin sitten uuden www-sivuosison ”hahmon kehittäminen” pariin. Yritin alkuun selvittää, että miten nykyinen kehitysversio reitittyy nykyiseen versioonsa, joka on sivu, jossa lukee: ”työn alla”. Aikani selailin navigointipalkin rakennetta ja kävin läpi FSD arkkitehtuurin tasoja. Siinä on seitsemän kansiota, joita kutsutaan tasoiksi. Kultakin tasolta on erilainen pääsyoikeus muihin kansioihin. Toiminnallisuus on hajautettu näihin kansioihin sen mukaan, kuinka geneeristä ohjelmakoodi on. Hierarkiassa alinta kansiota voi käyttää kaikki sen yläpuolella olevat kansiot. Selailin aikani koodia seuraamalla import määreitä. Minulle ei selvinnyt, missä konkreettisesti määritellään reitityspolut navigointivalikon linkkeihin, mutta huomasin kuitenkin, että rakenteesta löytyi HeroDevelopment-kansiot ja tämä riittää tiedoksi, mihin uutta ohjelmakoodia tulee tuottaa. Selvitin myös, mihin sivujen tekstit tulee laittaa, kun niitä kutsuttiin funktion avaimilla komennossa ”rakenna sivu”. Tekstit tulevat shared/i18n/locales-kansion alle maakohtaisiin kansioihin (kuva 3).



Kuva 3 FSD arkkitehtuuri (FSD 2024).

Selvitelyäni aikani arkkitehtuuria ryhdyin määrittelemään tyyppejä ja kirjoittamaan dummy-dataa entities/Hero/model/heroes.ts tiedostoon (kuva 3). Tästä voin hakea datan

kehitysvaiheessa olevaan verkkosivuun. Hahmojen pelikehitys on vielä kesken ja kyseistä dataa ei ole vielä haettavissa palvelimelta. Yritin selvittää mitenkä uusi branch nimetään liityen uuteen sivukokonaisuuteen, mutta se pitäisi ensin pilkkoa issueiksi, koska projektin ja versionhallinnassa seurataan issueita. Se on sellainen tehtävä, josta täytyy neuvotella harjoitustyöparin ja vastaavan ohjelmoijan kanssa. Tämä jäänee maanantaille.

Viikon 1 yhteenveto

Ehkä käsitykseni FSD arkkitehtuurista on parantunut. Eli FSD arkkitehtuurissa on 7 tasoa, kusakin tasossa on viipaleita "slices", joissa on sitten segmenttejä esimerkiksi model, ui, api (malli, käyttöliittymä, palvelinrajapinta) (FSD 2024). Tasot ovat kuvan mukaisessa järjestyksessä arkkitehtuurissa ja tasolla on pääsy vain alempiin tasoihin. Se mikä alkuun hämäsi, on se että kehitysympäristö laittaa kansiot aakkosjärjestykseen eikä arkkitehtuurin mukaiseen järjestykseen. Laitoin sitten monitoriin postit lapun, josta voi tarkistaa järjestyksen. Noudattamalla FSD arkkitehtuuria, syntyy helpommin ylläpidettävää koodia, mutta koodista tulee vaikeaselkoisempaa, mutta haasteellisuus on osaltani tervetullutta.

Olen päässyt kuluneen viikon aikana käsitykseen pelihahmojen kehittämismallista. Taiteellinen johtaja mahdollisesti keskittyy enemmän omaan näkökulmaansa ja asia ei todellakaan tullut kerralla selväksi. Viikkopalaverissa selvisi paljon sekä olen selvittänyt asiaa discordin chatissa eri ihmisiltä.

Ongelmana on ollut ymmärtää joitain käyttöliittymäelementtejä pohjautuen käsitykseen datan rakenteesta ja ideaa hahmon kehittämisestä. Asiat olen selvittänyt viestinnällä. Lisäksi git pull request on jäänyt tarkistus "review" tilaan roikkumaan. Olen muokannut ratkaisua ahkerasti kommenttien perusteella, mutta sen tila ei ole muuttunut. Tosin en ole nähnyt perjantaina vastaavia ohjelmoijia aktiivisena discordissa.

Viikon keskeisin oivallus oli se, että pelihahmo pystyy kehittämään mitä tahansa ominaisuutta niin paljon kuin kehityspisteitä on yhteensä jäljellä eikä tiettyä määrää kutakin ominaisuutta, kuten olin alkuun ymmärtänyt.

3.2 Seurantaviikko 2

Viikon tavoitteena on saada uuden verkkosivun Hahmojen Kehittäminen tuotanto käyntiin, mikä tarkoittaa issueiden ja Github branchin luomista. Kuitenkin vielä edellinen projekti on työn alla, joten tarkoitus olisi saada tämä navigointipalkin animaatio pois päiväjärjestyksestä. Tarkoitus olisi kehittää hyvän käytännön mukaista ohjelmointitaitoa ja ottaa opiksi vastaavien ohjelmoijien neuvoista.

Maanantai 4.11.2024

Navigointipalkin animaatio - branchiin oli tullut muutospyyntö. Olin animoinut käyttäen Reactin Effectiä ja javascriptin setTimeout-funktiota. Sain discord viestissä linkatut artikkelit referenssiksi, miksi näitä ei pitäisi käyttää. Reactin useEffect pakottaa komponentin renderoimaan uudelleen renderöinnin lopuksi ja nostattaa turhien renderöintien määrää. Settimeout-funktiokutsut saattavat tehdä skriptistä funktiokutsupainajaisen, mitä on vaikea ylläpitää ja se on virhealtista. Vastaava ohjelmoija oli toteuttanut työpöytäversion navigointipalkin animaatiosta. Siinä ei ollut kaikkia fadeout-animaatioita, joita olin toteuttanut tällä virhealttiimmalla koodilla, joten oletan ettei minun tarvitse toteuttaa niitä mobiiliversioonkaan. Tämä päivä meni tutustuesssa vastaavan ohjelmoijan koodiin ja alustavaan toteutukseen mobiiliversiosta. Sain illan päätteeksi toimivan version, kun jatkoin CSS:n tutkimista vielä illalla. Lisäksi minulla oli juoksevia asioita hoidettavana sekä opinnäytetyöpalaveri. Huomenna siistin koodista tarpeettoman pois, sillä kopioin työpöytäversiosta CSS:ää aika reippaasti. Osa saattaa olla tarpeetonta. Koodin tulee olla siistiä, eikä mitään turhaa pidä jättää lojumaan.

Tiistai 5.11.2024

Aloitin päivän korjaamalla CSS:ää navigointipalkin animaatioon. Selvittelin sitten versiointiin liittyviä asioita ja tein korjaukset, jotta sain tekemäni koodin oikeisiin tiedostoihin. Sain vihdoin työnnettyä finaaliversiosta Githubiin ja suoritin toiminnon "re-request reviewer", jotta vastaava ohjelmoija tajuaa katsoa Githubista työni tulosta. Odotellessani tuomiota aloin hahmottelemaan/testailemaan HeroDevelopment sivua. Täydensin ensin tiedot HeroManageriin, jotta pääsin tyyppivirheistä, jotka tulivat uusista tyyppimäärittelyistäni. Sitten toin hahmotaulukon HeroDevelopmentPage. Taulokassa on vaan se vika, että se on Record tyyppiä ja siinä on lista ryhmiä, joiden alla on hahmot, joita kaipaamaan hahmojen kehitys - alavetoalivalkossa. Kokeilin reduce funktiolla luoda hahmotaulukkoa, mutta jostain syystä se antaa tyhjän taulukon, vaikka konsolitulosteet näyttävät, että funktio käy hahmotaulukoita läpi. Huomasin, että Navigointipalkin animaatio - issue oli hyväksytty ja mergetty dev branchiin, mutta se oli poikunut kaksi uutta issueta. Ensinnäkin pitäisi luoda testitiedostot jest ja react-testing-libraryja varten ja toiseksi pitäisi luoda dokumentaatiot Jdock ja storybook tyyliin. Nämä kaikki ovat lähes tuntemattomia minulle. Aloin perehtymään storybookiin. Kesti hetken ennen kuin sain edes storybook näkymän auki. Kävin ensin muutaman tutoriaalivideon läpi. Huomasin, että navigointipalkin olemassaoleva stories ei toimi. Pistin merkille, että testaaminen oli aikaisempi issue ja siirryin tutkimaan olemassaolevia testitiedostoja. Tästä on hyvä jatkaa seuraavana päivänä. Luultavasti nämä uudet issuet vievät seuraavan päivän.

Keskiviikko 6.11.2024

Päivä alkoi koko tiimin palaverilla. Tämä tarkoittaa siis ihan kaikkia, sillä huomenna minulla on nettisivuryhmän palaveri, joka on enemmän minua koskettavaa. Palaverissa märehdittiin

pitkään Github oikeuksista, kuten personal access tokenin käytöstä. Palaverin jälkeen aloin kokeilemaan @testing-library/react kirjastoa. Siinä tulee ilmeisesti jest kirjasto mukana, kun sitä ei tarvitse erikseen importata. Alkuun oli vaikeuksia saada redux kontekstia, koska navigointipalkki testataan irrallisena komponenttina, mikä aiheuttaa suunnattomasti vaikeuksia, koska ei sitä ole tarkoitettu toimimaan yksinään. Sain vinkin vastaavalta ohjelmoijalta, mistä tiedostosta katsoa referenssiä mock-funktioihin. Tällä pääsin siitä, että olisi pitänyt asentaa redux-mock-store paketti. Siis pääsin user permissions redux tila ongelmasta, mutta tätä seurasi liuta muita ongelmia liittyen mock-funktioihin ja komponentin riippuvuuksiin. Vähän kuin testaisi tiilitalon toiminnallisuutta yksittäisestä tiilestä joutuen simuloimaan tiilitalon siihen ympärille. Tähän meni koko päivä ja todennäköisesti seuraavakin.

Torstai 7.11.2024

Päivä alkoi nettisivutiimin palaverilla. Nähtävästi kukaan ei ole vielä aloittanut ryhmätyön ohjelmointia. Suunnitelmia on Figmassa. Otin asian esille, että voisi alkaa tekemään issueita oman ryhmätyön suunnitelmasta. Vastuu issueiden tekemisestä delegoitui vastaavalle ohjelmoijalle. Tosin minun piti sitä varten purkaa suunnitelma dokumentiksi, josta selviää täsmällisemmin toiminnallisuus ja komponentit. Toiminnallisuutta kirjoittaessa mieleeni tuli lisäominaisuus, mutta en ottanut sitä nyt esille ettei suunnitelma mene sekavaksi. Se olisi kuitenkin itse verkkosivun ominaisuus, kun vastaava ohjelmoija määräisi, että ensin tehdään sivun komponentit valmiiksi, joita voidaan testata Storybookissa. Storybookissa voi testata komponentteja yksittäisinä komponentteina ja käyttöliittymä sallii parametrien muokkaamisen näkyvässä olevalle komponentille. Tänään tuli tutustuttua Storybookiin, kun issueina oli lisätä Storybook ja JSDock dokumentaatiot. Sain itse asiassa Storynook tiedostot valmiiksi ennen päivän päättymistä. Viimeistelin sitä ennen komponenttien testauksen pull requestia varten ja tein pull requestin. Komponenttien testaus - branch ei suostunut buildautumaan. Ilmeisesti vika on dev-branchissa, sillä virhe tulee koodista, johon en ole tehnyt muutoksia. Asensin Github Desktop sovelluksen palaverin aikana. Virallista repositoryä sille ei löytynyt Debian jakeluun, joten latasin suoraan asennuspaketin. Linuxin kanssa asennus usein menee säätämiseksi. Eli oma versionhallinta yksinkertaistuu, kun on projektin virallinen työkalu. Navbar collapse issuesta poikineista issueista jäi enää Jdock hoitamatta.

Perjantai 8.11.2024

Aloitin päivän muokkaamalla StoryBook tiedostoja. Päätin kuitenkin, että en tuo Navbar dokumentaatioon Collapsed ja Fixed konteksteja mukaan vaikka siitä sitten puuttuu toiminnallisuutta. Kysymys on kuitenkin yksittäisen komponentin dokumentaatiosta, niin olisi hämmäntävää renderata storybookissa muitakin komponentteja. Lisäksi laitoin merkille, kuinka eräässä toisessa komponentissa oli Dock tiedostot. Selvittelin asiaa, niin Storybookin saa automaattisesti generoimaan dokumentaatiota. Tähän asti pääsin suht helposti, mutta miten lisätään

kuvaukset parametreille, vaati hieman enemmän tutustumista StoryBook dokumentaatioon. Saatuaani StoryBook tiedostot viimein valmiiksi ryhdyin laatimaan JSDock kommentteja. Havaituin chatGPT:n tekevän parempia käännöksiä englannin kielelle, kuin Google kääntäjän, joten käytin sitä. ChatGPT ymmärtää kontekstin toisin kuin Google kääntäjä. Tumma ulkoasu chatGPT:ssä näyttää lähes samalta, kuin Discord, niin onnistuin kysymään hölmöyksiä vastaavasta ohjelmoijalta. Onneksi siitä saa poistettua lähettämänsä viestin. Sain vihdoinkin tehtyä GitHub pull requestin issueelle, jossa oli tehtävänä StoryBook ja JSDock. Testaus issue siirtyi pitotilaan, kun vastaava ohjelmoija sanoi, että navigointipalkin komponentteja täytyy refactoroida ja vasta sen jälkeen testata. Otin sitten uuden issuen työn alle. Tämä liittyy Hahmojen Kehitys - sivuun, sillä kyseessä on pylväsdiagrammi. Pylväsdiagrammi skaalaa suurimman arvon koko y-akselin pituudelle.

Sain suhteellisen nopeasti version, joka näyttää hyvältä, mutta koodiin jäi paljon tekemistä. Olin nopeasti testailut CSS:ää inline tyyppisenä. Tämä pitäisi kirjoittaa mieluummin SCSS-tiedostoon. Lisäksi vastaava ohjelmoija ehdotti, että toteutan automaattisen kielikäynnöksen valmiiksi tähän komponenttiin. Päivä alkaa olemaan täynnä. Jätän nämä maanantaille.

Viikon 2 yhteenveto

Kuluneen viikon aikana olen joutunut selvittämään ja omaksunut StoryBook konseptin. Storybook on kehitysympäristö ja käyttöliittymäkomponenttikirjasto uudelleenkäytettävien käyttöliittymäkomponenttien rakentamiseen, testaamiseen ja dokumentoimiseen verkkosovelluksissa. Sen avulla kehittäjät voivat eristää ja kehittää käyttöliittymäkomponentteja hiekkalaatikkoympäristössä ilman, että koko sovellusta tarvitsee ajaa. Opin kirjoittamaan Storybookin käyttämiä stories tiedostoja. Lisäksi olen selvittellyt ja oppinut testausta @testing-library/react-kirjastolla. Se ei ollut mikään helpoin tehtävä, koska testiskriptin antamat virheilmoitukset eivät ole kovinkaan informatiivisia. Lisäksi opin testaamisesta sen, että CSS visibility arvoa ei voi testata kunnollisesti testauskirjaston toBeVisible-funktiolla, joten testaustehtävä menee uusiksi. Lisäksi olen kohdannut haasteita ja oppinut, että setTimeout- ja useEffect-funktioita tulisi välttää React-koodissa.

3.3 Seurantaviikko 3

Viikon tavoitteena on vakinaistaa StoryBook ympäristö omana kehitysympäristössään. Se vaatii ehdottomasti vielä dokumentaatioon perehtymistä. Lisäksi tavoitteena on opiskella testausta. Sain viime viikolla palautetta, että testikirjaston waitFor-funktio ei toimi kunnolla ja, että olin testannut navigointipalkki-animaatiota viallisella testillä. Eli testauksessa on vielä opittavaa ja se tulee lähinnä tiimin palautteena.

Maanantai 11.11.2024

Aloitin päivän käymällä läpi ja siistimällä pylvädiagrammin koodia. Discordissa UI/UX-suunnittelija oli kommentoinut, että varjot ovat selkeärajaisia. Tyyli oli juuri sitä, mitä hän oli ajatellut. Myös taiteellinen johtaja oli peukuttanut kuvaa, joten finaaliversio on valmis. Seuraavaksi ryhdyin siirtämään tyylimäärittelyjä omaan tiedostoon. Saatuani tämän valmiiksi ajattelin tehdä storybook tiedoston. Vaihdoin GitHub Desktopissa branchiin, jossa olin tehnyt Storybookin. Kun vaihdoin takaisin, niin kaikki uusi koodi oli poistettu. Tuntui äärimmäisen turhauttavalta, kun aloin miettimään pylvädiagrammin uudelleen koodaamista. Aloin tutkimaan kehitysympäristön source control - osiota. Löysin sieltä osion "staches" ja löysin sieltä poistetun koodin. Tein sitten uudet saman nimiset tyhjät tiedostot ja kopioin koodin niihin. Tilanne oli pelastettu. Seuraavaksi aloin tekemään storybook - tiedostoa. Löysin aikaisemman koodini GitHubin pull requestin alta, sillä en enää uskaltanut vaihtaa branchia Github Desktopista ennen, kun olen pushannut koodin. Sain tehtyä Storybook tiedoston. Nähtävästi useClientTranslation ei toimi StoryBookissa, joten listan alkiot olivat siinä muuttujien nimiä.

Tiistai 12.11.2024

Tarkoitus oli tehdä vain JSDock pylvädiagrammiin ja tehdä git pull request, mutta tutkimalla koodia ja testaamalla, löytyi aina jotain parannettavaa. Loppujen lopuksi päädyin aina samaan ongelmaan, koska tausta elementin leveys ei skaalaudu fit-content määreellä, minun pitää tietää paljonko poistan leveyttä per pylvä, jotta koordinaatiston vasemmalle puolelle ei jäisi liikaa tyhjää tilaa. Olin tehnyt kokomääritykset niin kryptisesti, joten menin taakse päin suunnittelussa. Tarvitsin tietoa heti ensimmäisessä elementissä, jonka määrittelin myöhemmin pohjautuen fontti kokoon, jota olin säätänyt siinä matkan varrella. Jouduin pysähtymään miettimään, että missä kaikkialla olen viitannut aikaisemman elementin fonttikokoon ja muokannut sitä sekä mihin kaikkiin elementteihin vaikuttaa, jos muutan ylempänä hierarkiassa määrittelyä. Ongelmaksi tuli, kun säädän y-akselin fonttikokoa dynaamisesti, saada sen leveys selville. Tulin siihen tulokseen, että se on suurin piirtein 1.2 kertaa fonttikoko, kun y-akselin maksimiarvo on alle 10 ja 1.6 kertaa fonttikoko, kun se on yli 10. Sain tähän kulutettua koko päivän, kun alkoi väsyttää, niin tein viimein pull requestin ja lähetin koodin tarkastettavaksi.

Keskiviikko 13.11.2024

Vaikka sain tehtyä aamutunneilla pull requestin, jatkoin saman leveyden skaalaus ongelman parissa. Pyörittelin sitä päässä nukkumaan mennessä. Avattuani tyylimäärittelytiedoston koikeilin muokata justify-content arvoa, mutta sillä ei ollut efektiä. Sitten tajusin kokeilla laitaa yhden flex-elementin leveyden asettamista arvoon 100%, jolloin se käyttää kaiken vapaana olevan tilan. Lähtee tyhjät pois. Sitä oikeastaan kaipasinkin. Ehdin commitoimaan ja pushaamaan, kun huomasin, ettei asetukset toimi Firefoxissa kunnolla. Elementit menevät marginaalin yli. Odottelin iltapäivään asti, että sain palautetta pylvädiagrammista. Palautetta

tulikin sitten ihan kiitettävästi. Esimerkiksi minun pitää refaktoroida koodi paremman luettavaksi, sillä minulla olikin aika kryptisiä sotkuja siellä. Itse asiassa aloitinkin jo sitä hommaa. Sain kommentoidun kohdan korjattua. Tässä VSCode kehitysympäristössä näkee heti kommentit, kun vastaava ohjelmoija on niitä lisännyt. Lisäksi näin kommentin stats-tyypistä. Siitä tulee sitten ihan perustavalaatuinen remontti. Tämä jääköön huomiselle. Taaskin minä harhaudu jumittelemaan sen CSS:n kanssa. Tein muutoksia, mutta en tiedä onko siitä edes apua. Ainakin minä opin jotain koodista.

Torstai 14.11.2024

Aamupäivästä sain tehtyä kaikki vaaditut muutokset pylväsdiagrammin koodiin lukuunottamatta storybook osuutta. Tein sekavasta JSX koodista oman funktion, joka palauttaa JSX:ää, if - else - rakenteella kommentoituna kuten muutospyyntöissä edellytettiin ja tulihan siitä paljon selkeämpi. Seuraavaksi tein tyyppeihin vaaditut muutokset eli stats on taulukko Stat objekteja yhden objektin, jossa statsit kenttinä, sijaan. Minua vain askarrutti tyyppimäärittelyssä index-typestä herjaava virheilmoitus, mutta asia olikin niin yksinkertainen, josta pääsee oroon käyttämällä object[field] sijaan object.field viittausta. Aikaisemmassa toteutuksessa en siihen pystynyt, koska field arvo oli tuntematon, mutta tässä toteutuksessa ei tullut sitä ongelmaa. Loppupäivä meni storybookin kontrollin määrittelyyn. Vastaava ohjelmoija oli sanonut, etten saa kovakoodata arvoja. Storybook ensisijaisesti muotoilee kontrollit alustusarvoista. Kävin huomattavan määrän dokumentaatiota läpi, chattasin kolmen ihmisen kanssa ja laitoin ehdotuksia vastaavalle ohjelmoijalle. Päädyttiin kuitenkin siihen, että kovakoodaan arvot viidelle statsille. Sain päivän päätteeksi pushattua korjatun version nähtäville Githubin pull requestiin.

Perjantai 15.11.2024

Otin seuraavan issuen liittyen hahmojen kehitys sivuun työn alle, vaikka pylväsdiagrammin pull requestissa lukee edelleen ”1 change requested” ja ”merging blocked”, kun en nyt oikein tiedä mitä sillä muutoksella tarkoitetaan vai onko se siitä kiinni, että kaikki ei ole ehtinyt tarkistaa koodia eli minun muutoksia ei ole vielä hyväksytty. En viitsinyt toimeettomana jäädä odottelemaan, niin otin seuraavan tehtävän vastaan Liitin issuen itseeni, jotta issuelistaa se lailevat näkevät, että se on minulla työn alla, loin uuden branchin githubissa, vaihdoin local branchin uuteen branchiin. Kaikki vielä tähän mennessä hyvin, mutta kun aloin asentamaan puuttuvaa pakettia @fortawesome/font-awesome-svg-core, niin ongelmat alkoivat (ks. kuva 4). Eipä aikaisemmin ole tullut moista ongelmaa vastaan. Olen joka kerta joutunut asentamaan tuon paketin uudelle branchille, jotta verkkosivu saadaan toimintaan. Vastaava ohjelmoija ei osannut auttaa. Ehdotti, että poistan paketin package.lock.json tiedostosta, mutta se ei auttanut. Ongelma on siinäkin mielessä erikoinen, kun pystyn muuten kyllä asentelemaan paketteja, mutta juuri tuota en pysty. En sitten mitenkään saanut asennettua pakettia,

niin kyhäsin kehitysympäristön aikaisemmasta versiokansiosta kopioimalla node_modules kansion (jossa on puuttuva paketti asennettuna) uuteen branchiin ja asentamalla sitten siitä puuttuvat paketit sitä mukaan, kun virheilmoitus antaa ymmärtää, että paketti puuttuu, kunnes ei enää virheilmoituksia tullut. Sitten jouduin kopioimaan keskeneräiset ja uudet kooditiedostot uuden branchin toiseen versioon. Aika epäkäytännöllistä, jos tämä ongelma jää pysyväksi, niin on aina sama rullanssi edessä. Sitten pääsin uuden komponentin AttributesPricing kimppuun. Ongelmallista on, ettei hahmojen pelisuunnittelu ole vielä niin pitkällä, että löytyisi hinnoittelua kaavailtuun maksimileveliin 24 asti. Sain kuitenkin päivän päätteeksi komponentista toimivan version tosin ilman CSS muotoilua.

```

home_user@host:~/Documents/GitHub/Altzone-WebPages/frontend-next-migration$ npm install fortawesome/fontawesome-svg-core
npm error code 128
npm error An unknown git error occurred
npm error command git --no-replace-objects ls-remote ssh://git@github.com:fortawesome/fontawesome-svg-core.git
npm error Warning: Permanently added the ECDSA host key for IP address '140.82.121.3' to the list of known hosts.
npm error git@github.com: Permission denied (publickey).
npm error fatal: Could not read from remote repository.
npm error
npm error Please make sure you have the correct access rights
npm error and the repository exists.
npm error A complete log of this run can be found in: /home/home_user/.npm/_logs/2024-11-15T13_16_14_943Z-debug-0.log
home_user@host:~/Documents/GitHub/Altzone-WebPages/frontend-next-migration$ git branch
branch
dev
juha/docs/258-update-navbar-to-have-jsdocs-and-storybook
juha/feature/266-add-bar-chart-component
* juha/feature/267-add-attributes-pricing-component
juha/test/259-update-navbar-to-have-jest-and-react-testing-library-tests
main
home_user@host:~/Documents/GitHub/Altzone-WebPages/frontend-next-migration$

```

Kuva 4 Ongelma kehitysympäristön pystyttämisessä uuteen branchiin.

Viikkoyhteenveto

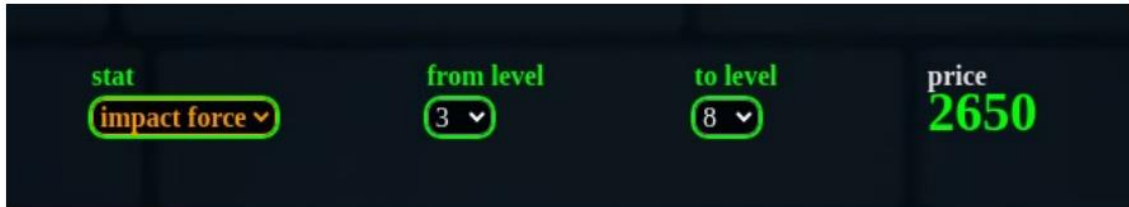
Navigaatiopalkkianimaatio oli kuitenkin hyväksytty, joten testaukseen syventyminen tapausta koskien jäi väliin. Storybookiin tuli syvällisempi tutkimus, kun yritin toteuttaa mahdotonta. Opin myös kantapään kautta Github Desktop ohjelman versionhallintaa. Kuluneen viikon aikana on tullut selville, että testaukseen ja storybookiin kuluu yhtä paljon aikaa kuin itse koodin luomiseen. Se testaus tuntuu jokseenkin turhalta, kun siinä todetaan vaan asioita, jotka on jo testattu koodin kehittämisen yhteydessä verkkoselaimella, mutta kai se on pakko siihenkin tuhlata aikaa, kun niin vaaditaan. Ehkä se kuitenkin hyvä luoda ne testitiedostot, kun ne voi ajaa kaikki kerralla ja kun ne testaavat yksittäisiä komponentteja niin näkee helposti mitä hajoaa, jos jokin riippuvuussuhteista muuttuu.

3.4 Seurantaviikko 4

Viikon tavoitteena on päästä itse hahmojen kehitys sivun komponenttien lisäksi varsinaisen sivun toteutukseen. Tarkoitus olisi myös luoda riittävän laadukasta koodia, jotta uudet komponentit menevät läpi korkeat laatuvaatimukset. Olisi hyvä oppia pääsemään useEffect-hookin käytöstä eroon, koska se rikkoo React paradigmaa.

Maanantai 18.11.2024

Muokkasin AttributesPricing-komponenttia selkeyden vuoksi yhdeksi komponentiksi, kun aikaisemmin se koostui kahdesta komponentista, jossa oli erikseen Price-komponentti, joka piti sisällään laskulogiikan. Tarkoitus tässä komponentijaossa oli, että olisin selvinnyt ilman Reactin useEffect-hooksia. Kuitenkin useStates-hooksit eivät riittäneet ja tilat päivittyivät epäsynkroonisissa, niin päädyin käyttämään useEffectiä. Sain tyylimäärittelyinkin tehtyä (kuva 5) vastaamaan suurin piirtein Figma suunnitelmaa lisäpiirteinä statsin nimen väri väristandardin mukaisesti. Testitiedoston tekemisessä oli haasteita. En löytänyt esimerkkiä, jossa olisi testi tietyille valitulle select - option valinnalle. Löytyi vain esimerkki getByRole("option", {name:"zebra"}).selected, mikä ei toimi tapauksessani, jossa on 3 select - option rakennetta, joten luovuin sen testaamisesta. Testasin vain, että tulee oikea hinta tietyille valinnoille.



Kuva 5 AttributesPricing-komponentti tyylimäärittelyllä.

Tiistai 19.11.2024

Jatkoin AttributesPricing-komponentin Storybook-tiedoston määrittelyllä. Komponentti on varsin yksinkertainen, joten sain aika suoraviivaisesti tiedoston kasaan muuttamalla pylväsdiagrammin stories-tiedostoa. Päätin kovakoodata hahmon statsin tiedot, koska se olisi aika työlästä täyttää kaikki käsin storybook näkymästä ja sitäpaitsi pylväsdiagrammista opittuna objektit tarvitsevat placeholderin, joten datan kovakoodaus on välttämätöntä. Tein sitten JSDockin ja pull requestin. Commitissa tuli vaan ongelmia, koska precommitin lint ei onnistunut, koska olin joutunut kasaamaan kehitysympäristön leikkaa - liimaa tekniikalla ja puuttuvat prettier-moduulit eivät asentuneet oikein. Jouduin taas kopioimaan tyhjään branchiin tekemäni koodin, jotta sain sen versionhallintaan. Kyselin sitten Discordissa, että mikä pylväsdiagrammin pull requestissa on, kun siellä näkyi muutospyyntö, vaikka kaikki pyydetyt muutokset ovat tehty. Sitten tuli ihan uusi ehdotus, että se voisi toimia ilman kovakoodattuja

leveys ja korkeus arvoja. Koko pylväsdiagrammin rakentaminen perustuu niille arvoille. Ei ole mahdollista pelkillä CSS määreillä rakentaa y-koordinaatin asteikkoa dynaamisesti nolasta maksimi leveliin, jos en saa tietoa korkeudesta JSX-koodiin. Päätin sitten kokeilla korkeuden määrittämistä referenssielementin `offsetHeight` arvosta `useEffect` hookissa, vaikka siitä varmaan tulee sanomista, mutta en alustavasti löytänyt parempaakaan ratkaisua googlaamalla.

Keskiviikko 20.11.2024

Päivä alkoi koko tiimin palaverilla. Oli ihan hyödyllistä olla paikalla. Hahmojen kehittämiseen liittyvä laskulogiikka muuttui täysin. Edes vastaavat ohjelmoijat eivät olleet perillä, miten levelit kehittyvät ja hinnat toimivat. Eli `AttributesPricing`-komponentti menee remonttiin, kunhan saan pylväsdiagrammikomponentin valmiiksi mikä vaikuttaa absurdilta, kun ohjeet ja vaatimukset muuttuvat kehitystyön aikana. Aikaisemmin oli päätetty, että pylväsdiagrammin koko määräytyy leveydestä ja muoto pysyy samana. Nyt vastaava ohjelmoija antoi koodia ja kommentoi: katso tästä. Kyseisessä koodissa muoto määräytyy leveydestä ja korkeudesta, joten koko logiikka menee uusiksi. Ilmeisesti tarkoitus olisi, etten käyttäisi `useEffect`ä. Olen käyttänyt dynaamisiin muotoilun parametrien selvittämiseen leveyden pikseliarvoa, jos sitä ei sallita, niin pylväsdiagrammista tulee paljon vähemmän responsiivinen. Lisäksi en tiedä kuinka tarkan y-koordinaatiston esittäminen on mahdollista. Esimerkkikoodin diagrammi on kuvassa 7 ja tämänhetkinen toteutus on kuvassa 6. Kuvasta 7 näkee, että y-akselin arvot ovat vähän mitä sattuu ja vähän missä sattuu. En tiedä tällä hetkellä pystynkö korkeuden prosentiarvoina toteuttamaan kuvan 6 koordinaatistoa kuvan 7 toteutukseen.



Kuva 6 Oma toteutus.



Kuva 7 Esimerkkikoodin toteutus pylväsdiagrammista.

Torstai 21.11.2024

Käytin aamupäivän pylväsdiagrammin koodin järjestelemiseen. Inline CSS omiin luokkiinsa ja muuhun koodin siivoamiseen. Päätin luopua dynaamisista muotoiluparametreista ja käytin kiinteää fonttikokoa ja border-radius pyöristyksen arvoja. Dynaamisuus olisi vaatinut enemmän javascriptiä. Vastaava ohjelmoija sanoi, että pyritään minimoimaan javascript. Tämä ratkaisu ei kestä skaalaamista niin hyvin, kuin oma versio ja y-akselin tarkkuus ei ole samaa luokkaa. Käytännössä kaikki koodi meni roskikseen ja tein esimerkkikoodin päälle. Muokkasin vain sen verran, että se näyttää suurimmilta osin samalta. Sitten olikin perinteinen verkkosivupalaveri. Siinä oli puhetta näyttötyöstä, joka tuli minulle yllätyksenä. Sain sitten pushattua pylväsdiagrammin pull requestiin ja vaihdoin branchin GithubDesktopilla AttributePricing branchiin. Sinne oli ilmaantunut paljon mielenkiintoista koodia. Kaksi eri versiota. Tutkin sitä koodia jonkin aikaa. Kysyin sitten vastaavalta ohjelmoijalta, joka oli luonut sitä koodia, että mitä minun tulee tehdä sen pull requestin suhteen. Kommentoiti, että voisin jättää sen sellaiseksi ja tehdä issueen siitä. Issueen laatiminen kesti sitten iltamyöhään, kun toinen vastaava ohjelmoija otti osaa siihen. Itse asiassa issueiden laatiminen on kuulema ollut yleensä vastaavan ohjelmoijan vastuulla, niin kommentoin sitten vastaavan ohjelmoijan muotoilua issues-tani, että päästiin yhteisymmärrykseen.

Perjantai 22.11.2024

Kuulin vastaavalta ohjelmoijalta, että toinen oli kommentoinut kuvalla, jossa teksti menee layout reunan yli pylväsdiagrammia. En tiedä oliko se sitten vastaavien ohjelmoijien välisestä keskustelusta, kun en löytänyt pull requestista sitä kommenttia, mutta kuulin asiasta toiselta. Päätin sitten korjata asian. Keskustelussa tämän vastaavan ohjelmoijan kanssa tuli esille CSS:n container query ja otin selvää ominaisuudesta. Varsin hyvä ominaisuus tilanteeseeni.

Siitä saa yksikön käyttöön, joka kuvaa prosenttia kontainerin leveyttä tai korkeutta. Juuri tällä saa dynaamisen fonttikoon ja sitä kautta muutkin parametrit. Tein sitten container queryja juurielementin kuvasuhteesta ja muotoilin näin sisällön ja hyödynsin elementin korkeuteen skaalautuvia fontteja. Sain responsiivisen lopputuloksen, joka pysyi layout rajojen sisällä. Seuraavaksi tiedustelin, haluaako vastaava ohjelmoija tehdä tekemäni issuen, kun keran aloittikin AttributesPricing-elementin toteutusta. Sain sen tehtävän itselleni, kunhan vaan teen siitä oman version. Eli tehtävänä oli suunnitella, miten toteutetaan päivitetty laskulogiikka. Se ei paljoa muuttunut. Leveli skaalat drop down valikoissa muuttuivat enemmän. Minulla oli vaikeuksia saada Storybookissa muutettuja data arvoja näkymään käyttöliittymässä tai näkymään oikein. Jatkoin koodin tutkimista vielä perjantaiyöhön. Lisäsin konsolitulostuksia komponenttiin ja sen blokkeihin ja sain eristettyä ongelmakohdan. Sain sen korjattua niin, että haen saman statsin stats taulukosta, joka on komponentin parametrina, niin saan päivitetyn version statsista.

Viikon yhteenveto

Tuli opittua, ettei Reactin Effectiä tule käyttää. Olin vain niin tottunut sen käyttöön omissa projekteissa, mutta olen nyt omaksunut tämän. Jos ulkoista järjestelmää ei ole mukana (esimerkiksi, jos haluat päivittää komponentin tilaa, kun tietyt propsit tai tila muuttuvat), et todennäköisesti tarvitse Effectiä. Tarpeettomien Effectien poistaminen tekee koodistasi helpommin ymmärrettävää, nopeammin suoritettavaa ja vähemmän virhealtista. (React 2024b.)

Opin myös vastaavan ohjelmoijan koodista useMemo ja useCallback hookieiden käyttöä. useMemoa käytetään komponentin ylimmällä tasolla laskennan välimuistittamiseen uudelleenrenderöintien välillä (react.dev 2024c). useCallback on React Hook, joka välimuistittaa funktiomäärittelyn uudelleenrenderöintien välillä (React 2024d). Vastaava ohjelmoija oli paketonut kaiken JSX:n ulkopuolisen koodin näihin, joten noudatin samaa menettelyä omissa modifikaatioissa. Näin koodista tulee optimoidumpaa, kun pystyy riippuvuustaulukossa määrittelemään, mitkä tilat tai muuttujat aiheuttavat uudelleenlaskennan tai määrittelyn.

Uutena asiana myös tuli CSS container queries. Tämä CSS kikka oli ratkaisu kohtaamiini haasteisiin pylväsdiagrammin toteutuksessa. Container queryn avulla voi soveltaa tyylejä elementtiin sen sisältävän containerin koon perusteella. Jos esimerkiksi containerin ympärivälissä kontekstissa on vähemmän tilaa, on mahdollista piilottaa tiettyjä elementtejä tai käyttää pienempiä fontteja. (MDN 2024b.)

3.5 Seurantaviikko 5

Tavoitteena on kirjoittaa puhdasta React koodia välttämällä paradigman rikkomuksia. Päästääkseni tavoitteeseen aion hyödyntää react.dev/learn materiaalia. Vaikka siinä onkin kohtalaisesti luettavaa, niin pidemmällä tähtäimellä säästän aikaa. Tavoitteisiin kuuluu myös hahmojen kehitys sivun implementaation aloittaminen. Olisi hyödyllistä päästä toteuttamaan sitä, kun olen jumittanut sen sivun komponenteissa, niin pääsisi oppimaan suuremman kokonaisuuden toteuttamista.

Maanantai 25.11.2024

Löysin `AttributesPricing`-komponentin koodista testaamalla vielä loogisen virheen hinnan laskukaavasta. Siinä oli otettu huomioon kehitysleveli, vaikka se on jo kertaalleen huomioitu `level range` -laskelmassa. Korjasin tämän. Tutkin vielä ajatuksella tuotosta, niin huomasin turhia riippuvuuksia, jotka siivosin pois. Lisäksi minua häiritsi, että koodissa oli toistoa. Tein oman funktion toistosta `setDropdowns`. Niin pääsin eroon ja tein `pull requestin`. Laitoin nyt viestiä vastaavalle ohjelmoijalle, että siellä on koodia esikatseltavaksi. Vielä tuli pylväsdiagrammiin korhauspyyntö. Mietin hetken, miten saan numeroelementin leveyden oikeaksi. Koikeilin javascriptia. Koodi näytti hieman purkkaviritelmältä. Pysähdyin miettimään, että saisi kohan sen puhtaasti CSS:llä ja niinhän sen sai `fit-content` määreellä. Minun on selvästi otettava CSS lähtöisempi lähestymistapa kehitykseen. `AttributesPricing`-komponentin `pull requestiin` oli ilmestynyt konflikti, kun pylväsdiagrammi oli vihdoinkin `mergetty`. Olin käyttänyt saman nimistä käännöstiedostoa, kun niissä on päällekkäisyyttä. Uudelleennimesin sitten tiedostot ja `pushasin pull requestiin`. Edelleen sinne jäi virhe `buildauksesta`. En tiedä mitä sen kanssa tehdä, kun se ei mielestäni johdu tekemistäni muutoksista

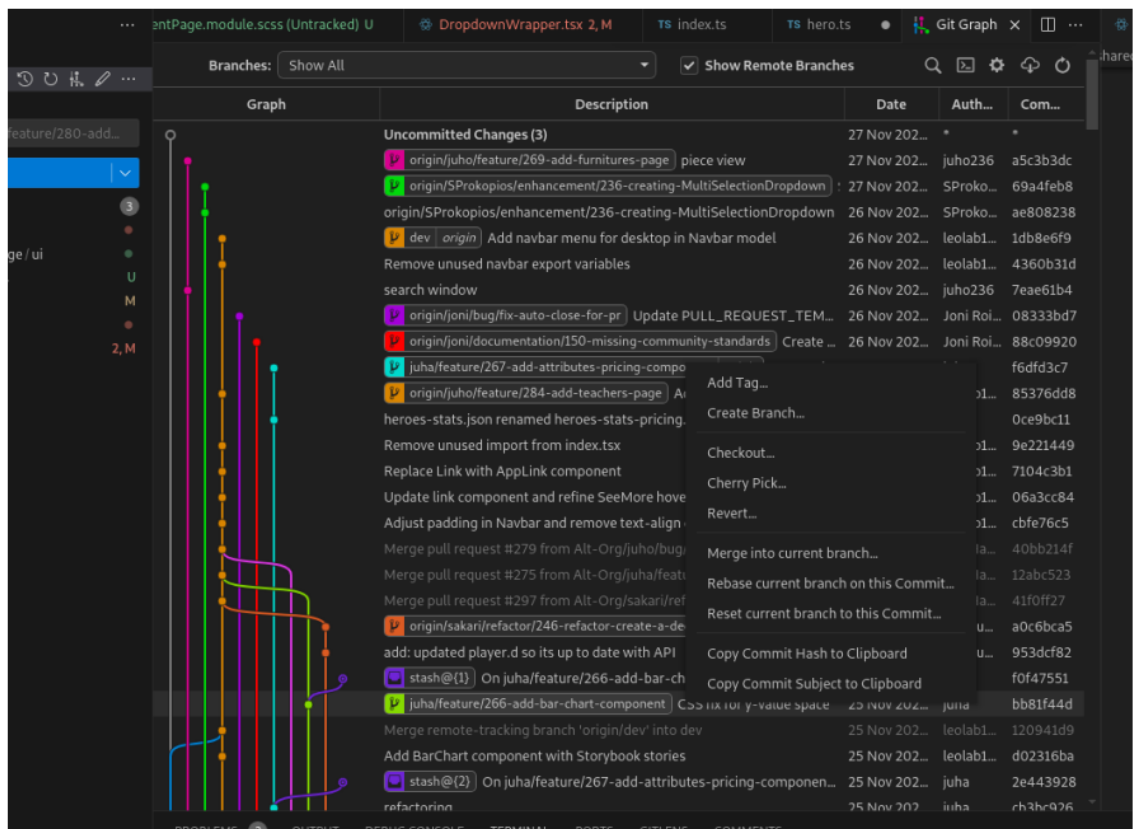
Tiistai 26.11.2024

Aloittelin `HeroDevelopmentPage`-komponentin tekoa. Olin saanut vinkin, että voisin käyttää `HeroPage` sivulla käytettyä `alasetovalikko` hyväkseni. Tutkin, miten se `alasetovalikko` toimii. Se käyttää `NavMenuWithDropdowns`-komponenttia. Tämä on lähtöjään huono, koska minun ei ollut tarkoitus tehdä kuin yksi sivu. En tarvitse navigaatiota eri sivujen välille. Tutkin pitkään `NavMenuWithDropdowns`-komponenttia. Se kutsuu itseään rekursiivisesti. En tahtonut löytää aluksi, missä `alasetovalikkojen` alkiot ovat toteutettuna. Sitten tutkin `DropdownWrapperia`, jota se käyttää. Sain vihdoinkin testitulostetta valinnan tekstin rinnalle. Ongelmana oli, että sitä käytetään reitittämiseen. Aloin sitten tutkimaan missä `HeroPagen` reititys on toteutettu. `Heroes` sivun toteutus on niin hajallaan; on `preparedPages`-, `features`-, `entities`- ja `app-sliceissa` jauttuna erilaisiin `containereihin`, `managereihin`, `segmentteihin`. Ei tahtonut sekään helpolla löytyä. Siis reititysten pitäisi olla `app` kansiossa `kansiohierarkiana`. Löysin sieltä `[slug]` nimisen kansion. Ilmeisesti `DefaultAppRoutesProps` syö sen hahmon nimen `url:stä` ja sitä kautta annetaan parametrina `HeroPage`-komponentille. Aloin kuitenkin miettimään, että on vähän

kömpelöä tehdä sivu, jossa on ainoastaan alavetovalikko ja joukko sivuja sen valinnoille. Päätin muokata DropDownWrapperiä. Lisäsin sinne takaisinkutsun painallukselle ja näin sain HeroDevelopmentPagen tilan muuttumaan valitun hahmon mukaan. Seuraavana ongelmana oli alavetovalikon sulkeminen. Sitä ei oltu toteutettu, koska reititykset ohjaavat eri sivulle, mikä aiheuttaa uudelleen renderöinnin. Silloin ei tarvitse sulkea valikkoa erikseen. Lisäksi, kun alavetovalikossa on rekursiivisesti alavetovalikkoja, sen tilan "isOpen" manipuloiminen sulkee vaan sisimmän alavetovalikon. Se tila pitäisi toteuttaa kontekstina tai Redux store tilana. Taas karkasi käsistä oman sivun tekeminen. Löysin kuitenkin niksini, miten saan alavetovalikon uudelleen renderautumaan, jolloin se tulee suljettuna näkyviin. Tästä varmaan tulee sanomista, mutta koitan nyt aluksi edistyä omassa aihealueessa.

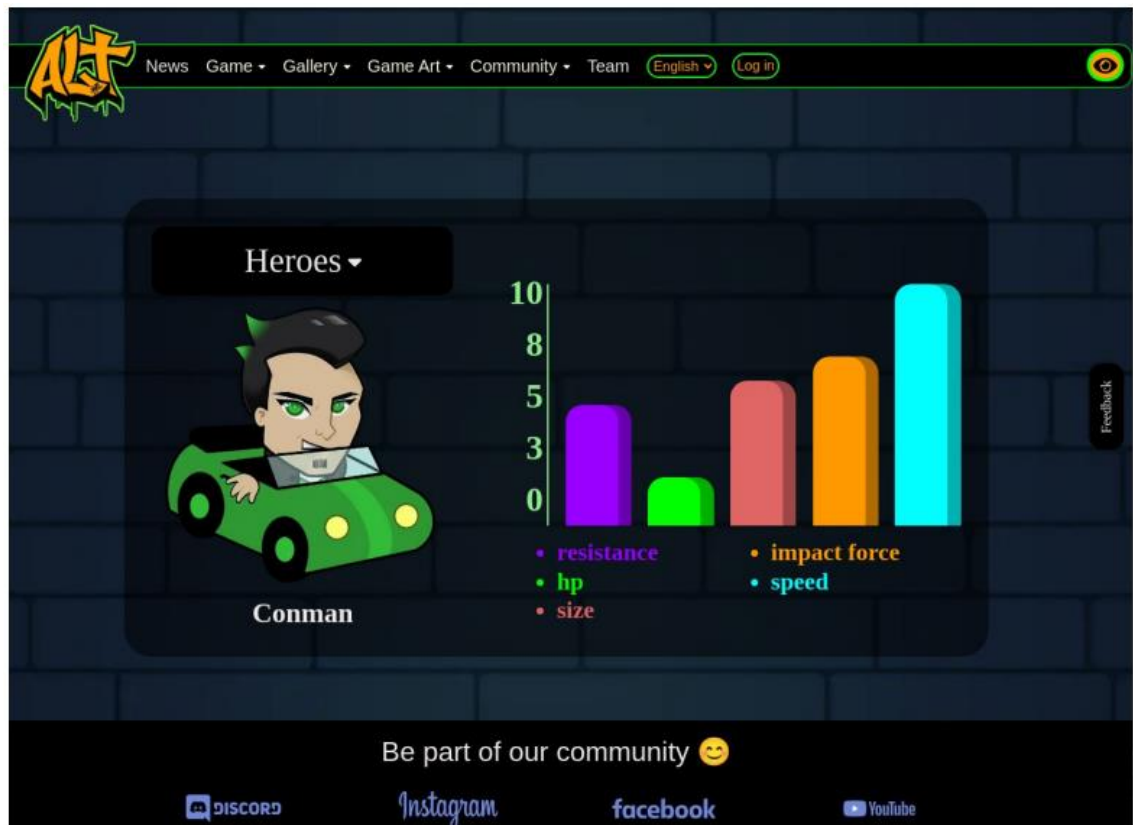
Keskiviikko 27.11.2024

Päivä alkoi koko tiimin palaverilla. Siellä tuli pelkkää Unity (Unity on pelikehitysympäristö) asiaa kaksi tuntia. Ei laisinkaan verkkosivujen kehityksestä, mutta se piti kuunnella, jos sieltä olisi tullut jotain tärkeää. Jatkoisin sen ongelman, miksi en saa uusimpia komponentteja versiohallinnasta. Vastaava ohjelmoija vinkkasi, että projekti pitää päivittää välillä. Olin luullut, että fetch origin - toiminto tuo aina tuoreimman tavaran, mutta ei ilmeisesti uusia tiedostoja. Minun Linux version Github Desktopista puuttuu update-branch nappi, jonka löysin googlaamalla ja siinä viitattiin Windows ympäristöön. En löytänyt myöskään VSCodium (kehitysympäristö) update optiota (ks kuva 8).



Kuva 8 VSCodiumin versionhallintaa.

Löyty kyllä varsin monenlaista näkymää versiohistoriaan. Vastaava ohjelmoija viittas WebStormin kehitysympäristöön kuvalla, jossa oli update ominaisuus. Asensin sitten WebStormin. Olin kyllä kokeillut dev branchin päivitystä VSCodiumissa, mutta en onnistunut commitoimaan mystisen "unknown console error. Exit code 1" virheen takia precommitissa, joten tekemäni koodi olisi kadonnut vaihtamalla branchia Github Desktopilla. Siispä asensin uuden kehitysympäristön. Vastaava ohjelmoija on kyllä kehunnut sitä. Sieltä löysin update project - napin. Kävi kuitenkin ilmi, että olin ilmeisesti tehnyt uuden branchin väärin main branchiin eikä dev branchiin, joten siinä ei ollut mitään päivitettävää. Mutta siirryin kuitenkin käyttämään WebStormia. Sain muutaman Github operaation jälkeen versionhallinnan ojentukseen ja lisäsin BarChart-komponentin hahmojen kehitys sivulle (kuva 9).



Kuva 9 Tämänhetkinen versio hahmojen kehitys - sivusta.

Torstai 28.11.2024

AttributePricing-komponentilla olisi käyttöä tässä vaiheessa, mutta pull request ei ole edistynyt. En tiedä johtuuko se siitä, että se ei onnistu buildaamaan ja vastaava ohjelmoija ei sen takia ole vaivautunut katsomaan sitä. Buildauksessa näkyy 2 virhettä, jotka ovat vastaavan ohjelmoijan tekemässä koodissa. Vastaava ohjelmoija pyysi jättämään hänen tekemän koodin sinne omaksi versiokseen, joten en mahda mitään tilanteelle. Pitäisikö siitä sitten tehdä uusi branch, joka suostuu buildaamaan, mutta sitten pitäisi tehdä uusi issue. Muotoilin viestin vastaavalle ohjelmoijalle, jossa mainitsin asiasta ja kysyin, että pitäisikö hahmojen kehitys sivun loput toiminnallisuus sisällyttämään AttributesPricing-komponenttiin, koska ne riippuu kyseisen komponentin tilasta. Hän mainitsi, että koodin puhtaudesta vastaava ohjelmoija todennäköisesti pyytää sitten kuitenkin erottamaan ne toisistaan. Muotoilin koodin puhtaudesta vastaavalle ohjelmoijalle viestin, että pitäisikö tehdä komponentti AttributeDetails, jota AttributePricing käyttää tai sitten pitää valintojen tilat ylemmässä komponentissa ja säätää niitä AttributePricing-komponentissa propsien kautta. Odotellessa vastausta tein HeroDeveloperPage-komponentista responsiivisen hyödyntämällä projektin omaa hookia useSizes. Katsoin referenssiä hahmosivun koodista. En ollut saanut päivän päätteeksi vastausta viestiin. Huomenna täytyy varmaan ottaa uusi issue työn alle, jos vastausta ei ala kuulumaan, jotta olisi tekemistä.

Perjantai 29.11.2024

Vastaava ohjelmoija oli vastannut viestiin ja peukuttanut ideaa, että AttributesPricing säättää HeroDevelopmentPage-komponentin tiloja propsien kautta. Vastaavalta ohjelmoijalta oli tullut vielä korjauspyyntö AttributesPricing-komponenttiin, että tekisin apuluokan laskentaa ja hakuja varten. Ilmeisesti vastaava ohjelmoija edellyttää, ettei komponenteissa ole mitään alemman tason toteutusta, vaan pelkkiä funktiokutsuja, joiden toteutus on muualla. Tämä tarkoittaa, että HeroDevelopmentPage-komponentti tulee muuttumaan, jotta se pääsee seuraavaksi läpi, sillä siinä on alavetovalikon toteutusta. Se oli mielestäni pakollista, jotta saan siihen alavetovalikkoon text-valinta link-valinnan sijaan ja callback funktion asettamaan sivulle hahmo tila, sillä päätin tehdä sivun tilalla enkä reitityksellä. Vähän epäilyttää, että vastaava ohjelmoija tulee vielä tämän muotoseikan takia vaatimaan sivun toteutusta reitityksellä, jotta koodi näyttäisi puhtaammalta, mikä olisi ristiriidassa sen tiedon kanssa, että React olisi erityisesti single page appeja varten. Tämä vaatisi koodin totaalista remonttia ja lukuisten uusien kooditiedostojen tekemistä. Tein sitten AttributesPricingHelper luokan, jonne siirsin funktioiden toteutusta, vaikka osalla funktioita tulikin vaan yksi rivi koodia. Vastaava ohjelmoija oli korjannut AttributesPricing-komponentin pull requestin build errorin sallimalla .eslintrc.json tiedostossa enemmän.

Viikon yhteenveto

Projektitiimin kanta nähtävästi koodin puhtaudesta on erilainen, kuin react.dev:in pitämällä komponentit puhtaana - ohjeistuksen, jossa mainitaan, että puhtaat funktiot eivät muuta muuttujia, jotka ovat funktion ulkopuolella, eivätkä objekteja, jotka on luotu ennen funktion kutsua, mikä tämä tekee niistä epäpuhtaita (React 2024e). Toisin sanoen ehdotukseni, että AttributesPricing käyttää komponenttia AttributeDetails olisi puhtaampi ratkaisu kuin, että AttributesPricing säättää isäkomponentin tiloja, joka sitten muodostaa tietoja liittyen AttributesPricing-komponentissa tehtyihin valintoihin, mutta vastaavaa ohjelmoijaa pitää kunnioittaa projektissa. Hän sanelee säännöt.

Uutena asiana tuli Webstorm, vaikka yhä käytän VSCodea vielä vanhoissa brancheissa, koska hakemistopolut ovat sen muistissa. WebStormista on saatavilla ilmaisversio ei kaupallisiin projekteihin. Eli sopii mainiosti opiskelijalle. WebStorm on JetBrainsin kehittämä suosittu integroidun kehitysympäristön (IDE) työkalu erityisesti JavaScript-, TypeScript- ja muihin web-kehitykseen liittyviin teknologioihin. Se on suunniteltu tehostamaan web-sovellusten kehitystä tarjoamalla monipuoliset työkalut, kuten älykäs koodieditori, virheiden tarkistus, automaattinen koodin korjaus sekä saumaton tuki useille kehitysfraameworkeille.

Mitä myös olen huomannut kuluvan viikon aikana, että pull requesteihin reagoidaan aika hitaasti, mikä on hidastanut kehitystyötäni. Githubin pull request sivulla ei jostain syystä aina ole näkyvillä re-request reviewer nappi. Olen kyllä maininnut discordissa, kun olen saanut

jonkin asian valmiiksi, mutta tämä ei ehkä riitä. Täytynee jatkossa mainita pull request eksplisiittisesti viestissä, jos silloin vastaava ohjelmoija ymmärtäisi edistää asiaa.

3.6 Seurantaviikko 6

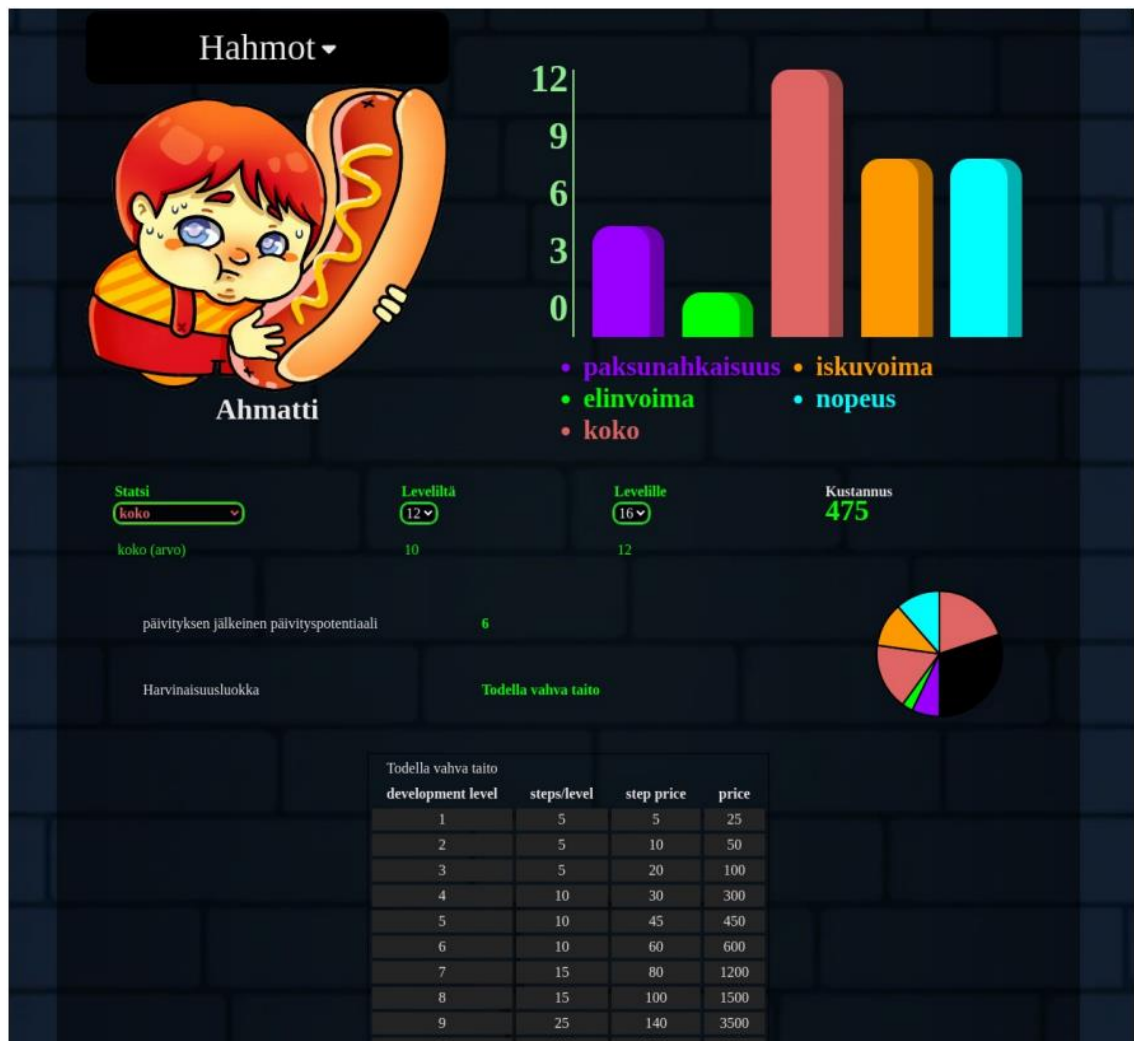
Tavoitteena on saada Hero Development sivun kaikki toiminnallisuus valmiiksi. Tai ainakin koodin puhtaus - pull request - update kierteeseen. Koska ei ole vielä tiedossa kuinka perustavalaatuksia muutoksia sivuun tullaan vaatimaan, ei sen valmistumisajankohtaa pysty tarkalleen sanomaan. Luultavasti muutoksia tullaan vaatimaan NavMenuWithDropDowns-komponenttiin, jotta se saadaan suljettua valinnan jälkeen oikeapoisemmin kuin tällä hetkellä setKey metodilla tai sitten tämän takia tullaan vaatimaan Hero Development - sivun jakamista sivuiksi ja ohjaamalla reitittämistä NavMenuWithDropDowns-komponentista.

Maanantai 2.12.2024

Laitoin vastaavalle ohjelmoijalle viestiä, että katsoo AttributesPricing-komponentin pull requestin. Enää tuli vain pieni korjausvaatimus. Minun tuli siirtää tyyppimäärittely asianmukaiseen tiedostoon. Tein sen, laitoin viestiä ja vastaava ohjelmoija merges juha/feature/267-add-attributes-pricing-component branchin dev branchiin. Enää piti saada uusi koodi hero development page - branchiin. En onnistunut update project - toiminnolla. Olin asiasta vastaavaan ohjelmoijaan yhteydessä ja hänkään ei onnistunut siinä jostain syystä. Löysin sitten toiminnon WebStormista, jossa pystyn päivittämään dev branchin taustalla, vaikka olen toisessa branchissa. Mutta, kun kyseinen branch perustuu dev branchiin, saan ehkä päivitetyn koodin sitä kautta. Vertaamalla dev branchiä hero development page branchiin huomasin, että dev branch oli päivittynyt. Löysin versionhallinta toiminnon Merge 'dev' into 'juha/feature/280-addHeroDevelopmentPage' ja sain AttributesPricing-komponentin koodin vihdoinkin käyttööni. Seuraavaksi minun piti alkaa muokkaa AttributesPricing-komponenttia toimivaksi hero development - sivun kanssa. Jotenkin tuntuu kömpelöltä, että joudun antamaan tolkkottomasti propseja AttributesPricing-komponentille, jotta saan pidettyä tilat HeroDevelopmentPage-komponentissa, kun nuo asiat olisi voinut pitää komponentin sisällä. Kaiken lisäksi attributes pricing - komponentin tapahtumakäsittelijöihin tulee vielä lisää tilapäiväytyksiä liittyen HeroDevelopmentPage-komponenttiin ja propsilista tulee vielä kasvamaan tuosta. Arvelluttaa, että mahtakohan vastaava ohjelmoija kuitenkaan loppujen lopuksi olla tyytyväinen tähän ratkaisuun. Itse olisin pitänyt alavetovalikon toteutuksen HeroDevelopmentPage-komponentissa, koska niiden tapahtumakäsittelijät ohjaavat kaikkea HeroDevelopmentPage-komponentissa, vaikka toteutuksesta tulisikin silloin melko massiivinen.

Tiistai 3.12.2024

Minun piti saada päivitys potentiaali päivityksen jälkeen - arvo hahmolle. Tein sitä varten tilan HeroDevelopmentPage-komponenttiin ja annoin AttributesPricing-komponentille propsina setUpgradePotential funktion. Yritin käyttää sitä AttributesPricing-komponentin toLevel-alasvetovalikon tapahtumakäsittelijässä. Sain virheellisiä arvoja. Lisäsin console.log-tulostuksia ja huomasin, että defaultLevel ei päivity handleToLevelChange funktiossa. Tajusin sitten laittaa riippuvuustaulukkoon muuttujan selectedStat, koska funktio on kääritty useCallback funktioon, sen määrittely päivittyy vain riippuvuustaulukon herättelemänä. Sain sitten oikeat arvot testitulostukseen. Sitten piti tehdä käännöstyötä kielivalintoja varten. Lisäsin loputkin puuttuvat elementit sivuun (Kuva 10).



Kuva 10 Hahmojen kehitys sivu vaaditulla sisällöllä.

Ehkä hahmon kuva ja pylväsdiagrammi voisivat olla sirompia, jos vertaa Figma suunnitelmaan. Muuten näyttää aika valmiilta. Toki koodia pitää vielä siistiä. Mm. @ts-ignore määreestä pitää päästä eroon. Tämä sivuuttaa typescript valitukset. Iän ikuisia valituksia. Nyt se valittaa, että muuttujan arvo ei voi olla undefined, mutta kun sillä ei yksinkertaisesti ole arvoa ennen kuin

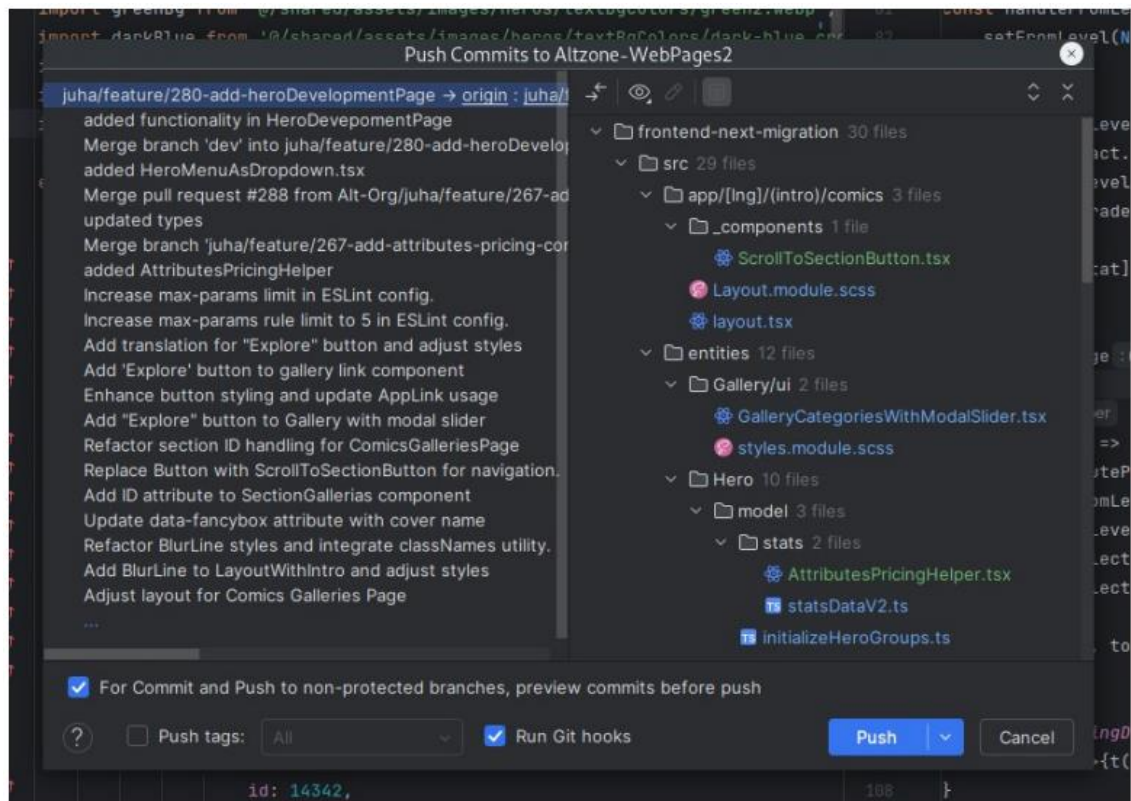
se alustetaan ja sitä ei käytetä sitä ennen. Saa nähdä, minkälaisia purkka - klemmari virityksiä senkin eteen pitää vielä tehdä.

Keskiviikko 4.12.2024

Päivä alkoi koko tiimin palaverilla. Siellä tuli verkkosivujen kehitykseen liittyen jonkin verran asiaa väreistä. Ei käytettäisi täysin mustaa tai täysin valkoista. Samalla havaitsin, mistä nuo tyylimäärittelyt löytyvät Figmassa. Palaverin jälkeen refaktoroin koodia ja testailin HeroDevelopmentPage-komponenttia. Piirakkamallissa oli virhe arvojen laskennassa. Korjasin sen. Ja oin initializeHeroGroups koodin kahteen tiedostoon. Laitoin stats datan erilleen, kun muuten tiedostokoko olisi ollut liian suuri koodin tyylimäärittelysääntöjen kannalta. Kehitysympäristö huomautti asiasta jo aiemmin, mutta asia oli jäänyt roikkumaan. Lisäksi laitoin HeroDevelopmentPage-komponentin tietorakenteita useMemo hookiin, jotta koodi olisi optimoidumpaa. Huomasin virheitä ja varoituksia HerodevelopmentPage-komponentissa. Nähtävästi HeroDevelopmentPage-komponentin tilojen säätö AttributesPricing-komponentissa tuottaa ongelmia. Tulee muun moassa varoitus: ei voi päivittää komponenttia HeroDevelopmentPage, kun renderaa AttributesPricing-komponenttia. Löytääksesi pahan setState kutsun AttributesPricing-komponentin sisällä, seuraa virheilmoituksia. Siis paha asetus tapahtuu, kun AttributesPricing lukee stats-propsin. En nyt tämän päivän aikana ratkaissut ongelmaa. Ehkä minun kannattaa tehdä pull request, jos vaikka tulee niin radikaaleja muutosehdotuksia, että ongelmatilanne poistuu.

Torstai 5.12.2024

Päivä alkoi nettisivupalaverilla. Palaverissa tuli ilmi, että minun pitää tehdä toimeksianto kielenkääntävävastuulliselle Githubin project-tilaan. En kyllä löytänyt kyseistä projektia, kun katselin Githubin projekteja. Täytyy yrittää löytää discordista linkki siihen. Vaikuttaa, että olisin tehnyt enemmänkin muokkauksia. Kuvan tiedostot eivät ole minun muokkauksia. Yritin pushata hahmojen kehitys sivua. Ilmestyi outo varoitus (kuva 11). Kysyin neuvoa vastaavalta ohjelmoijalta. Hän sanoi, että pushaa vaan, kun se on omassa branchissa, niin isompaa vahinkoa ei pääse sattumaan. Kun pushasin, niin tuli ilmoitus "pushed 63 commits". Minulla oli vaan muutama committi. Outoudet varmaan johtuvat siitä, kun en saanut AttributesPricing-komponenttia päivitettyä, päivitin taustalla dev branchin ja mergesin sen hahmojen kehitys branchiin. Tultiin siihen ratkaisuun, että teen uuden branchin ja kopioin muokkaukset siihen. Kävin vielä issueta läpi ennen kuin teen pull requestin. Huomasin, että olen unohtanut testaamisen. Kaikki käyttämäni komponentit ovat testattu erikseen jo kehitysvaiheessa paitsi tietenkin dropdown, joka ei ole minun tekemä komponentti. Sainkin muutaman tunnin kulumaan siihen, miten saan DropDownWrapper-komponentille välitettyä test-id:n. Minun piti luoda komponentille uusi propsi dataTestId, niin sain välitettyä NavMenuWithDropdowns-komponentista test-id:n ja sain vihdoin testin tehtyä.



Kuva 11 Outouksia versionhallinnassa.

Viikkoyhteenvedo

Opin päivittämään projekti tavalla, joka rikkoo versiohallinnan. Sain kuulema täysin uudenlaisen ongelman aikaiseksi. On se sitten monimutkainen asia, kun ei voi mennä mutkattomasti. Ongelma saatiin korjattua kloonamalla dev branch uuteen kansioon ja tekemällä HeroDevelopmentPagea varten uusi branch. Tämä oli jo kolmas branch kyseiselle issuelle. Opin myös nettisivupalaverissa uuden käytänteen, jossa oikeakielisyyssasiat tekee joku muu. Pitää vaan löytää projektisivu Githubista, jota en ole löytänyt ja sinne tehdä toimeksianto. En pahemmin oppinut uusia käytäntöjä koodaamiseen kuluvalle viikolla, kun en saanut pull requestia tehtyä. Se on oikeastaan pääasiallinen tapa saada vuorovaikutusta aikaiseksi ja kehittävää keskustelua koodin toteutuksesta. Tosin kyllä opin DropDownWrapperin ja NavMenuWith-Dropdowns välisestä suhteesta, kun piti saada testi id välitettyä koko hierarkian läpi. Itse testi oli HeroMenuAsDropdowns-komponentille. Huomaa, että alavetovalikkoa on ollut toteuttamassa sekalainen joukko.

Ongelmia, joita kohtasin kuluneella viikolla gitin lisäksi on ratkaisematon varoitus (kuva 12). Vaikka kyseessä on vain varoitus, todennäköisesti se otetaan tosissaan. Tämä saattaa johtua epäpuhtaasta ratkaisusta AttributesPricing-komponentissa. Mainitsinkin jo asiasta viime viikon yhteenvedossa. Saa nähdä, miten asiat tulee muuttumaan pull requestin jälkeen, joka on tarkoitus julkaista maanantaina.

```

2 [Fast Refresh] rebuilding hot-reloader-client.js:162
Warning: Cannot update a component
('HeroDevelopmentPage') while rendering a different component
('AttributesPricing3'). To locate the bad setState() call inside
'AttributesPricing3', follow the stack trace as described in
https://reactjs.org/link/setstate-in-render Error Component Stack
  at AttributesPricing3 (AttributesPricingV3.tsx:32:5)
  at div (<anonymous>)
  at main (<anonymous>)
  at HeroDevelopmentPage (HeroDevelopmentPage.tsx:17:37)
  at Suspense (<anonymous>)
  at InnerLayoutRouter (layout-router.js:241:11)
  at RedirectErrorBoundary (redirect-boundary.js:72:9)
  at RedirectBoundary (redirect-boundary.js:80:11)
  at NotFoundBoundary (not-found-boundary.js:62:11)
  at Suspense (<anonymous>)
  at LoadingBoundary (layout-router.js:338:11)
  at ErrorBoundary (error-boundary.js:130:11)
  at InnerScrollAndFocusHandler (layout-router.js:152:9)
  at ScrollAndFocusHandler (layout-router.js:227:11)
  at RenderFromTemplateContext (render-from-template-context.js:15:44)
  at OuterLayoutRouter (layout-router.js:348:11)
  at InnerLayoutRouter (layout-router.js:241:11)
  at RedirectErrorBoundary (redirect-boundary.js:72:9)
  at RedirectBoundary (redirect-boundary.js:80:11)
  at NotFoundBoundary (not-found-boundary.js:62:11)
  at LoadingBoundary (layout-router.js:338:11)
  at ErrorBoundary (error-boundary.js:130:11)
  at InnerScrollAndFocusHandler (layout-router.js:152:9)
  at ScrollAndFocusHandler (layout-router.js:227:11)
  at RenderFromTemplateContext (render-from-template-context.js:15:44)
  at OuterLayoutRouter (layout-router.js:348:11)
  at div (<anonymous>)
  at div (<anonymous>)
  at div (<anonymous>)
  at InnerLayoutRouter (layout-router.js:241:11)

```

Kuva 12 Olisiko tämä merkki huonosta arkkitehtuurista?

3.7 Seurantaviikko 7

Tavoitteena olisi saada hyvä vuoropuhelu käyntiin hahmojen kehitys sivusta, kun olen tähän mennessä tehnyt siinä kaiken oman intuition mukaan lukuun ottamatta arkkitehtuuria, jonka ymmärsin oikeaksi yhden peukutuksen perusteella kommenttiini. Vastaava ohjelmoija on vähäsanainen persoona, joten peukku kommentissa on tulkittava hyväksynnäksi. Laitan maanantaina pull requestin vireille ja saan toivottavasti tämän vuoropuhelun vireille pian, koska en sitä itsekään hyväksyisi sellaisenaan tuotantoon.

Maanantai 9.12.20024

Aloitin päivän tutkimalla uuden hero-development-page branchin koodia, että olisiko se valmis pull requestiin. Näytti ihan hyvältä, joten commitoin, pushasin ja tein pull requestin.

Tälläkin kertaa tuli huomautus ”For Commit and Push non-protected branches, preview commits before push”, mutta tällä kertaa tiedostoluettelossa oli aidosti minun tekemiä muokkauksia. Pull request yrittää buildaa projektin ja tekemäni pull request hälyytti build-virhettä. Virheitä tuli muun muassa BarChart-komponentin stories ja test - tiedostoista, joihin oli unohtunut vanha muuttujan nimi. Vaihdoin value nimen defaultLevel nimeksi. Lisäksi tuli virhettä HeroDevelopmentPage-komponentin propseista. Sivuni ei tarvitse propsia, mutta app layerin _getPage:n createPagen funktioparametri buildPage vaatii sinne jotakin propseja, joten muokkasin sivua niin, että sille annetaan otsikko siellä ja tätä kirjoittaessa tajusin, että ehkä muukin tekstisisältö oli varmaan tarkoitus antaa siellä. Korjaan huomenna. Saatuani pull requestin buildamaan laadin Alt-Org/Projects/Kirjoittajat - projektiin issueen kehityssivujen käännöksistä (kuva 13).

Hero Development Page #76

Open Alt-Org/Altzone Public

juhaj77 opened 1 hour ago

Eli tarvittaisiin käännöksiä sivulla [kehityspisteiden hinnat](#) esiintyvälle termeille.

1. Termille stats pitäisi keksiä sana, joka on suomea. Olisiko englanninkielinen vastine stat?
2. Käännökset termeille: TODELLA VAHVA TAITO, VAHVA TAITO, AIKA VAHVA TAITO, KESKITASOA, AIKA HEIKKO TAITO, HEIKKO TAITO, TODELLA HEIKKO TAITO
3. Ovatko nämä oikein: SUOJAUS / RESISTANCE, ELINVOIMA / HP, KOKO / SIZE, ISKUVOIMA / STRIKE, NOPEUS / SPEED? Niitä on nähtävästi jo muokattu kehityspisteiden hinnat spreadsheetiin sillä resistance oli aiemmin paksunahkaisuus.
4. Ilmaisut ja käännökset asioille: a) päivityksen jälkeinen päivityspotentiaali b) Harvinaisuustuokka
5. oikeakieliset ilmaisut ja käännökset kehetyssivuhintataulukon sarakkeiden otsikoille:
Leveli Askelia/Leveli Askel Hinta Hinta

Hahmon kehitys

Hahmot ▾

Ahmatti

Stats: paksunahkaisuus ▾

Levelillä: 5 ▾

Kustannus: 0

Assignees: No one - Assign yourself

Labels: No labels

Projects

KIRJOITTAJAT

Status: HOMMAT kirjoittajille ▾

Priority: Choose an option

Size: Choose an option

Estimate: Enter number...

Iteration: Choose an iteration

Start date: No date

End date: No date

Milestone: No milestone

Development: Create a branch for this issue or link a pull request.

Notifications: Customize

Unsubscribe

You're receiving notifications because you're subscribed to this thread.

Kuva 13 Issue kirjoittajille.

Piti käyttää aikaa siihen, mistä olen termit ottanut ja mille asioille tarvitsen oikeakielisyyttä ja käännöksiä. Kaikelle tekstisisällölle tarvitsen kirjoittajien päätöksen, sillä sivuilla on sanastoja, joka ei ole mitään kieltä esimerkiksi ”stats”.

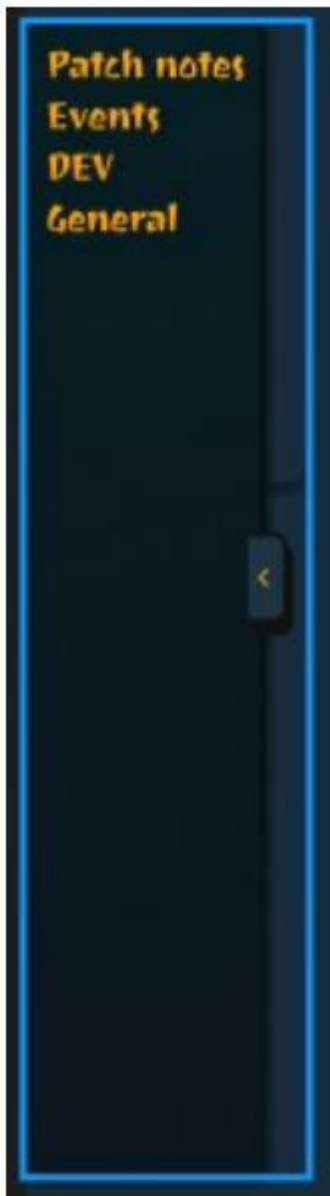
Tiistai 10.12.2024

Vastaavalta ohjelmoijalta oli tullut kommenttia pull requestiin liittyen import määreisiin. Minun ilmeisesti pitäisi importtaa ne ensin index-tiedostoon, josta sitten exporttaan. Näin kyseisten importtien hakemistopolku yksinkertaistuu. Vastaava ohjelmoija laittoi myös viestiä,

että toteutan hahmon valinta alavetovalikon sivupalkki tyyppisenä. Esimerkkikoodissa oli käytetty `LayoutWithSidebars`-komponenttia, jota käytin. Tulos ei kuitenkaan ollut sellainen, jota vastaava ohjelmoija tarkoitti. Vastaava ohjelmoija oli viitannut `ClansNavMenuAsSidebar.tsx` tiedostoon, jota minun projektista ei löytynyt vaikka päivitin projektin. Mahtaakohan tästä tulla taas versionhallinnallinen painajainen, jos kyseinen komponentti on mergetty projektiin branchinini tekemisen jälkeen? Sain kuitenkin päivän päätteeksi toteutuksen `LayoutWithSidebars`-komponentilla. Huomiseksi jäi import määreiden korjaaminen ja `ClansNavMenuAsSidebar.tsx` koodin löytäminen.

Keskiviikko 11.12.2024

Korjasin import määreet ja niihin liittyvät export määreet vastaamaan FSD:n muodollisuuksia. Kokeilin eri asetuksia `HeroMenuAsDropDown`-komponentille. Vastaava ohjelmoija oli kuitenkin sitä mieltä, että en käyttäisi alavetovalikkoa sivupalkissa, vaan tekisin ihan listan siihen. Hyvä etten ehtinyt poistaa `HeroMenuAsDropDown`-komponentin koodia, sillä selvisi vähän myöhemmin, että sitä käytettäisiin kuitenkin tabletti ja mobiilikoossa. Kuvassa 14 on vastaavan ohjelmoijan kaipaama ulkoasu sivupalkille. Tein `HeroMenu`-komponentin, jossa on `collapse/expand`-ominaisuus, mutta elementin koko ei vastaa todellista sivupalkin kokoa. Luulen, että joutuisin muokkaamaan `LayoutWithSidebars`-komponenttia, jos siihen sivupalkkiin halutaan kuvan 14 ominaisuudet. Tiedustelin vastaavalta ohjelmoijalta `ClansNavMenuAsSidebar` koodia, koska vastaava ohjelmoija oli pyytänyt käyttämään sitä referenssinä. Kävi ilmi ettei sitä koodia ole vielä olemassa, jolloin päädyin `HeroMenu`-komponentin toteutukseen. Luulen, että minulla on vielä edessä ulkoasun muuttaminen tuohon oranssiin väriin, jota esiintyy figma suunnitelmissa. Komponentit, kuten navbar ovat vielä vanhassa vihreässä teemassa.



Kuva 14 Vastaavan ohjelmoijan tarkoittama ulkoasu sivupalkille.

Torstai 12.12.2024

Vastaava ohjelmoija kommentoi, että voisinko sivulla olla oletuksena joku hahmo valittuna. Aloin toteuttamaan ominaisuutta, niin törmäsin haasteeseen, että missä välissä alustan stat-sien värit, kun aiemmin se on tehty hahmon valinnan yhteydessä. Nyt pylväsdiagrammi ei saa värejä oletushahmolle. Tämä johtuu siitä, etten viitsinyt kovakoodata värejä joka hahmon joka statsille, kun ne ovat hahmoille samat samoissa statseissa. Nyt asetan hahmon setState funktiolla. Jouduin pilkkomaan alustusfunktion pienemmiksi funktioiksi ja järjestelemään ne uudelleen setState funktioiden kanssa, sillä en voi käyttää funktiota ennen kuin se on määritelty ja en myöskään voi käyttää tilaa hookin riippuvuustaulukossa ennen kuin se on määritelty. Jossain vaiheessa tuli virheilmoituksia loputtomassa loopissa. Sain haasteen hoidettua

eriyttämällä muista alustuksista värien asettamisen ja kutsuin sitä listan ensimmäiselle hahmolle ennen hahmon setState funktiota. Hahmolistan sain heroManager-luokkafunktiolla.

```
const [hero, setHero] = useState<Hero>(initializeHeroColors(heroManager.getAllHeroes()[0]));
```

Loppupäivän tein CSS tyylimäärittelyitä. Korjasin myös hahmodatasta bugin. Pushasin sitten koodin pull requestiin ja tein sinne kommentin. Mahdollisesti tullaan vaatimaan CSS transitiota sivupalkin collapse/expand toiminnolle, jota teen sitä huomenna varmuudenvuoksi odotellessa reaktiota pull requestiin.

Perjantai 13.12.2004

Aloitin päivän korjaamalla HeroMenu-komponentin kutistettua ulkonäköä. Siinä oli hölmösti reunat suorakaiteen ympärillä, jonka sisällä oli laajenna-nuoli. Korjasin sen niin, että kutistetuissa tilassa näkyy pelkkä laajenna nuoli. Aloin seuraavaksi tekemään animaatiota eli CSS transitiota valikon leveydelle. Jouduin muuttamaan JSX:n rakennetta, kun näkyvyyttä ohjattiin JSX:ssä suoraan visible tilalla, kun CSS transitiossa se tapahtuu muuttamalla CSS luokkia. Saatuaani sen valmiiksi huomasin viestit vastaavalta ohjelmoijalta. Vastaava ohjelmoija oli toivonut, että kun valikko kutistetaan, se samalla vapauttaa tilaa sivun sisällölle. Eli jouduin ottamaan collapse tilan HeroMenusta ja laittamaan sen HeroDevelopmentPageen, jotta voin ohjata LayoutWithSidebars-komponenttia. Meni aikaa, kun ihmettelin miksei CSS:n min-width määre vaikuta. Huomasin minulle vieraan CSS määreen flex-bases, joka tarkoitti käytännössä elementin leveyttä myöskin. Sain sitten collapse toiminnon toimimaan niin, että se vapauttaa tilaa LayoutWithSidebars-komponentissa pääsivulle. Aika hyväksi todettu tapa on määritellä eri värisiä reunoja eri elementeille, niin saa käsityksen mitä pitää säätää layouttia muokattaessa. Vielä jäi ärsyttävä x-vierityspalkki Chrome selaimessa kutistettu-tilassa. Sain sen pois määrittämällä overflow-x:hidden collapse tilassa.

Viikon yhteenveto

Olin viikonloppuna yhteydessä vastaavan ohjelmoijan kanssa. Hän oli sitä mieltä, että lopullinen hero development - sivun koodi tulisi toteuttaa reitityksenä. Eli jokainen hero olisi oma sivu. Kuitenkin seuraavaan buildiin mennään nykyisellä toteutuksella, sillä buildilla on aika-taulu lopettavia harjoittelijoita varten 18.12. Tämä tarkoittaa totaalista remonttia koodissa. Syyksi hän ilmoitti, että se on mukavampi käyttää, kun voi esimerkiksi lähettää linkkinä sivun ja hakukoneoptimointi toimii. Vastaava ohjelmoija myös mainitsi, että next.js sotkee kuviota. Eli täytynee aloittaa sivun suunnittelu tyhjältä pöydältä uudestaan.

Itse viikko on sujunut odotusten mukaan. Olen saanut pull requestista interaktiivisen prosessin odotetulla tavalla. En varsinaisesti ole oppinut uutta tällä viikolla, kuin että next.js vaikuttaa ratkaisevasti komponenttiarkkitehtuuriin projektin vaaliman koodaustyylin kannalta (Next.js

2024). Koodin arkkitehtuuri joutuu vasaran alle. En tällä hetkellä tätä tekstiä kirjoittaessa tiedä vielä miten hero development - sivu kannattaisi toteuttaa. Vaikuttaisi vähän kömpelöltä tehdä joka herolle oma sivu, joka on kuitenkin sama. Täytyy kuitenkin kunnioittaa vastaavan ohjelmoijan toiveita ja pakottaa koodi siihen muotoon.

3.8 Seurantaviikko 8

Viikon tavoitteena on saada refaktoroitua koodi niin, että Hero Development - sivu toimii reitityksellä. Työharjoittelu on kirjattu jatkuvan 10.1.2025 asti vaikka tässä opinnäytetyössä seurataan vain 8 viikkoa, joten saan varmasti sen toteutettua aikaresurssi mielessä, vaikka tässä työssä saattaa vaikuttaa, että tiukalle menee. Eli tarkoitus on perehtyä Heroes sivun toteutukseen ja katsoa siitä esimerkkiä omaan toteutukseen. Heroes sivussa on pääsivu ja yksittäisille Heroille alasivu. Tästä vaan heti herää kysymys, että miten toimin sen oletushahmon kanssa? Ilmeisesti minun pitäisi saada pääsivu suoraan uudelleenohjautumaan alasivuun. Ei ole vielä mitään käsitystä, miten toteuttaisin sen. Täytynee vaivata vastaavia ohjelmoijia aiheella.

Maanantai 16.12.2024

Sain viikonloppuna mergettyä dev branchin hero-development-page branchiini, kun sinne oli tullut uutta koodia liittyen alasvetovalikon active-väriin. Olin tutkinut koodia aikani ja tehnyt muutoksia, joita vastaava ohjelmoija oli toivonut ja kokeillut saada active-väriä toimimaan, niin vastaava ohjelmoija sanoi, että olisi lisää aiheeseen liittyvää koodia mergetty. Jouduin sitten tekemään konfliktien välttämiseksi roll back kaikkiin muokkauksiini, jotta voi tehdä git pull ilman konflikteja. Siinä meni kaikki aamupäivän työ, mutta ei se onneks määrällisesti ollut paljoa. Kaikesta huolimatta en vaan saanut active-propsia toimimaan. Active-väri oli toteutettu NavMenuWithDropdowns-komponentissa, mutta tapa jolla käytän komponenttia ei toimi element.active asetuksella. NavMenuWithDropdowns-komponentti kutsuu DropdownWrapper-komponenttia, jonne se asettaa propsina elementtilistan sisäkkäiselle alasvetovalikolle. Jouduin sitten kuitenkin implementoimaan active-värin DropdownWrapper-komponenttiin. Sitten tein loppupäivän CSS hienosäätöä vastaavan ohjelmoijan pyyntöjä noudatellen. Tein lisäksi yhden rakenteellisen muutoksen JSX:ään ja sain paremman hallinnan alasvetovalikon elementin leveydelle.

Tiistai 17.12.2024

Deathline lähestyy seuraavaan buildiin. Pitäisi olla 18.12. mennessä. Vastaavalta ohjelmoijalta oli tullut kommentti, että hahmokuvien pitäisi olla yhtä suuria. Kuvakoko vaihtelee, kun selaa HeroMenua ja sivun sisältö pomppii. Korjasin ongelman. Huomasin lisäksi kirjoitusvirheen käännostiedostossa. Korjasin senkin. Minun piti tehdä uusi hotfix branch näitä korjauksia varten, kun branch juha/feature/280-add-hero-development-page oli mergetty dev branchiin,

joka oli sitten mergetty main branchiin ja juha/feature/280-add-hero-development-page -branch oli poistettu. Nimeämiskäytännön mukaan branchin nimessä pitää olla issunumero, mutta sain nyt erityisluvalla tehdä branchin ilman issueta, kun homma oli niin pieni, että is-suen laatiminen olisi ollut isompi homma. Sitten oli tullut kommenttia sivupalkkiin ilmestyvästä x-akselin suuntaisesta vierityspalkista, joka ilmestyy, kun sivua kavennetaan säätämällä selainikkunaa. Siitä pääsee eroon 1) asettamalla aside-elementille overflow: hidden, jolloin osa heromenun oikeaa laitaa leikkaantuu pois 2) asettamalla heromenulle leveyden % - yksikössä, jolloin osa heromenun sisällöstä jää piiloon tai 3) asettamalla flex-basis ja/tai flex-shrink arvoja, jolloin se rikkoo säännönmukaisuuden eri sivujen välillä.

Keskiviikko 18.12.2024

Päivä alkoi koko tiimin palaverilla. Siellä oli yleistä höpinää pelin nappuloista ja joku, jonka mikrofoni on niin heikkolaatuinen, etten saanut mitään selvää, oli paljon äänessä. Vastaava ohjelmoija kommentoi postaamaani videota reititetyn hero-developer sivun toiminnasta, että teen jotakin väärin. Sitten hän postasi linkin ja vinkkejä. Eli pitäisi kuitenkin tehdä toteutus ilman oletushahmoa. Eilinen työ menee osittain hukkaan, kun speksit muuttuu, mutta oli itse toiminnallisuudessakin vikaa, kun hahmoa valitessa koko sivu lasketaan uudestaan. Myöhemmin keksin, että se oli osin <a> - elementin käytöstä, kun olisi pitänyt käyttää AppLink-komponenttia. Enkä ollut myöskään käyttänyt next.js:n layout komponenttia tehokkaasti hyödyksi (Next.js 2024b). Etenemisestään oli puuttuva issue. Toinen vastaava ohjelmoija oli sanonut, että issuen tekeminen on vastaavien ohjelmoijien vastuulla. Kuitenkin tämä vastaava ohjelmoija sanoi, että tee vaan issue. Tein sitten issuen ja branchin ja pääsin virallisesti tekemään urakkaa. Siirsin vinkin mukaan LayoutWithSidebars-komponentin next.js layout.tsx tiedostoon. Alkuun oli ongelmia. Kun server-komponentissa ei voi käyttää tilaa. Muokkasin sitten sen käyttämät komponentit tilattomaksi kunnes huomasin, että voin käyttää "use client" määrettä. Tällöin sivu muodostuu selaimessa. En tiedä tuleeko tästä sanomista, kun layout on oletusarvoisesti server-komponentti. En muuten keksi, miten saan sivupalkin-collapse tilan hallittua. Voisin tietenkin katsoa, onko sitä käytetty misään muussa layout-komponentissa. Ryhdyin sitten muuttamaan komponenttia takas tilalliseksi.

Torstai 19.12.2024

Päivä alkoi nettisivupalaverilla. UI/UX henkilöllä oli paljon sanottavaa. Muummoassa, että hahmon kehittäminen tulisi toisen sivun toiminnoksi. Myöskään UI/UX persoona ei ollut ymmärtänyt sivun ideaa. Eli olisi varmaan hyvä mainita, että tämä on tällainen geneerinen työkalu hintojen tutkimiseen eikä assosioidu pelaajan hahmoprofiileihin, muuten kuin lähtötasoleveleillä, jotka ovat kaikille samat. Myös johtajalta tuli kommenttia kuvakoon vaihtelusta, joka on jo korjattu, mutta ei näy buildissa, jota katsottiin. Lisäksi tuli kommenttia alavetovalikoiden pienuudesta. Lisäksi tuli edelliseen liittyen, että hahmomenuvalikko on aivan liian

pieni ja tyyli voisi olla yhteneväinen. Minun näytöllä näyttää ihan riittävän isolta ja olen noudattanut tarkkaan vastaavan ohjelmoijan toiveita tyyliissä.

Testailin reititettyä sivua ja huomasin ongelman kielivalinnan pysyvyydessä. Kieli vaihtuu itsestään toiseksi, kun klikkailee hahmoja. Mietin, että ongelma on uusi ja muualla layout-komponentti on client-komponenttina. Muutin sen taas server-komponentiksi, mutta ongelma pysyi. Tuli kaatuminen (Kuva 15), kun klikkailin kahden eri hahmon välillä edestakaisin ja kieli vaihtui samalla.

```
▶ Uncaught Error: SEGMENT MISMATCH
  at serverPatchReducer (server-patch-reducer.js:46:19)
  at clientReducer (router-reducer.js:29:67)
  at eval (use-reducer-with-devtools.js:63:21)
  at updateReducerImpl (react-dom.development.js:11510:11)
  at updateReducer (react-dom.development.js:11356:10)
  at Object.useReducer (react-dom.development.js:13317:16)
  at useReducer (react.development.js:1747:21)
  at useReducerWithReduxDevtoolsImpl (use-reducer-with-devtools.js:102:53)
  at Router (app-router.js:175:169)
  at renderWithHooks (react-dom.development.js:10930:18)
  at updateFunctionComponent (react-dom.development.js:16072:20)
  at beginWork$1 (react-dom.development.js:18268:16)
  at HTMLUnknownElement.callCallback (react-dom.development.js:20367:14)
  at Object.invokeGuardedCallbackImpl (react-dom.development.js:20416:16)
  at invokeGuardedCallback (react-dom.development.js:20491:29)
  at beginWork (react-dom.development.js:26669:7)
  at performUnitOfWork (react-dom.development.js:25493:12)
  at workLoopSync (react-dom.development.js:25209:5)
  at renderRootSync (react-dom.development.js:25164:7)
  at recoverFromConcurrentError (react-dom.development.js:24381:20)
  at performConcurrentWorkOnRoot (react-dom.development.js:24326:26)
  at workLoop (scheduler.development.js:261:34)
  at flushWork (scheduler.development.js:230:14)
  at MessagePort.performWorkUntilDeadline (scheduler.development.js:534:21)
```

Kuva 15 Kieliongelma näkyy olevan kehitysympäristössä.

Virheilmoituspinosta huomaa, että polku menee pääsääntöisesti development-komponenttien läpi. Tästä päättelin, että vika on kehitysympäristössä. Itse asiassa dev branchiin mergetyllä kuvagallerialla on myös vastaavanlaisia ongelmia ja se löytyy buildistä, joten testasin alt-zone.fi verkkosivua ja siellä ei ole mitään ongelmaa. Kulutin sitten päivän ainoastaan kehitysympäristössä ilmenevän ongelman metsästämiseen.

Perjantai 20.12.2024

Nyt kun pugit on selvitetty, suoritin koodin refaktoroinnin. Minulla oli ongelma valitun hahmon active-väriin kanssa, kun käytin tilamuuttujaa säilömään sen. Kielivalinnan yhteydessä sivu

latautuu uudestaan ja tila katoaa. Päätin hyödyntää url:ää, kun kerran hahmon "slug" näkyy siinä. Slugia hyväksikäyttämällä yleisesti selvitetään hahmon tiedot. Koodissa pääsee käsiksi url:ään next.js:n hookilla usePathname. Sain ihan toimivan ratkaisun. Oli siis reitityksellä toteuttamisesta jotakin hyötyä koodissakin. Tein muutokset CSS:n, mitä torstain palaverissa tuli. Vaihdoin myös fontin yhtenäiseksi käyttämällä SCSS muuttujaa -font-family-main. Siivoin tarpeettomaksi jääneet tiedostot pois. Poistin myös koodia, jonka olin säästänyt kommentoimalla sen, kun olin varma etten tarvitse sitä enää. Lisäsin myös aloitussivun, jossa on infoteksti hahmojen kehityssivun tarkoituksesta. Uudelleenorganisoin käännöstiedostoja johdonmukaisemmaksi. Sain oikeastaan kaiken valmiiksi. Tein git commitin ja pushasin koodin. Enää jäi git pull request tekemättä siitä refactorointi-issuesta, mutta sillä tuskin on kiire, kun kaikki on lomalla.

Viikon yhteenveto.

Pääsin kärryille next.js:n app-routerin toiminnasta ja server / client - komponenttien ideologiasta. Eli next.js suorittaa palvelinpäädyn renderöintiä ja saadakseni käyttäjän toimintaan liittyviä tilamuuttujia, pitää käyttää "use client" määrettä, jolloin komponentti esitellään client-side - komponenttina. Tällöin se renderöityy selaimessa. Opin myös hyödyntämään route-polkua koodissa tilamuuttujan tavoin käyttämällä next.js:n usePathname hookia. Sain toteutettua refaktoroinnin. Lisäksi tein ominaisuuden alasvetovalikkoon, joka sulkee sen valinnan jälkeen. En tiedä halutaanko sitä, joten en commitoinut sitä vielä. Tämä viimeinen seurantaviikko on ollut varsin tuottelias loppuaan kohden.

4 Yhteenveto ja pohdinta

Olen oppinut kirjoittamaan puhdasta React-koodia. Tai ainakin huomattavasti lähempänä sitä kuin aloittaessa harjoittelua. Lisäksi pääsin tutustumaan FSD-arkkitehtuuriin käytännön kautta. Itse asiassa FSD-arkkitehtuuri yhdistettynä next.js projektirakenteeseen voisi olla täsmällisempi kuvaus, sillä next.js:n kansiorakenne on osittain sama app-router kansiorakenteen takia. Oli varsin rakentavaa perehtyä FSD-arkkitehtuuriin toteutettuun typescript koodiin, sillä kyseiseen arkkitehtuuriin toteutettu koodi on vaikeaselkoisempaa kuin klassiseen perus react projekti rakenteeseen toteutettu. Siinäkin tuli hyvää harjoitusta ja oivalluksia, kun toteutin hahmon kehitys sivun ensin single page - sovelluksena, jonka sitten refaktoroin useammaksi sivuksi toimimaan reitityksellä. Itse asiassa tässä vaiheessa vasta oivalsin syvällisemmin next.js:n ideologiaa. Olen laajentanut tietoperustaa parhaiten next.js ja react.dev sivustojen kautta. React.dev sivustoon olisi kannattanut tutustua jo aikaisemmin, mutta pidin suorittamieni kurssien oppimateriaaleja riittävänä. Jos haluaa kirjoittaa puhdasta React koodia, niin edellämainittu lähde on ihan välttämätön yleissivistävä kokonaisuus. Opinnäytetyön raportointi on auttanut hahmottamaan omaa ajankäyttöä ja kompastuskiviä, joista mainittakoon

versionhallinnalliset probleemat, mutta voin nyt sanoa, että git ja Github ovat enemmän van-
kalla pohjalla, kuin aikaisemmin. Toimeksiantaja sai hyötyä enimmäkseen harjoittelusta,
jossa toteutin hahmojen kehityssivun (kuva 16) ja muita issueita. Lisäksi edesautoin pelin ta-
sojen hinnoittelulogiikkaa, koska se oli keskeneräistä ja vaadittua sivun toiminnan kannalta.
Hyöty opinnäytetyöstä kohdistui harjoitteluun ajankäytön optimoinnissa. Koska opinnäytetyö
on päiväkirjatyypinen opinnäytetyö, opinnäytetyöstä toimeksiantaja voi tutustua haasteisiin
ja edistykseen harjoittelijan näkökulmasta harjoittelijavetoisessa projektissaan ja näin mah-
dollisesti tehdä harjoittelijoiden käytöstä vieläkin tuottoisampaa projektissaan.



Kuva 16 Valmis tuotos. Mobiilikoossa hahmolistausta tulee alaspäinvalittava. Sivun asettelu on responsiivinen.

Lähteet

Atlassian 2024. What is Git?. Viitattu 25.11.2024. <https://www.atlassian.com/git/tutorials/what-is-git>

Figma 2025. Think bigger, Build Faster. Viitattu 22.1.2025. <https://www.figma.com/>

FSD 2024. Overview. Viitattu 25.11.2024. <https://feature-sliced.design/docs/get-started/overview>

git 2025a. Reference. Viitattu 22.1.2025. <https://git-scm.com/docs/git-branch>

git 2025b. Reference. Viitattu 22.1.2025. <https://git-scm.com/docs/git-commit>

git 2025c. Reference. Viitattu 22.1.2025. <https://git-scm.com/docs/git-push>

Juviler, Jamie 2021. What Is GitHub? (And What Is It Used For?). Viitattu 25.11.2024. <https://blog.hubspot.com/website/what-is-github-used-for#what-github>

MDN 2024a. JavaScript. Viitattu 25.11.2024. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

MDN 2024b. CSS container queries. Viitattu 25.11.2024. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_containment/Container_queries

Next.js 2024. next.js. Viitattu 15.12.20024. <https://nextjs.org/>

Next.js 2024b. Layout.js. Viitattu 18.12.2024. <https://nextjs.org/docs/app/api-reference/file-conventions/layout>

React 2024a. Built-in React Hooks. Viitattu 25.11.2024. <https://react.dev/reference/react/hooks>

React 2024b. You Might Not Need an Effect. Viitattu 24.11.2024. <https://react.dev/learn/you-might-not-need-an-effect>

React 2024c. useMemo. Viitattu 24.11.2024. <https://react.dev/reference/react/useMemo>

React 2024d. useCallback. Viitattu 24.11.2024. <https://react.dev/reference/react/useCallback>

React 2024e. Keeping Components Pure. Viitattu 1.12.2024. <https://react.dev/learn/keeping-components-pure>

TypeScript 2024. TypeScript is JavaScript with syntax for types. Viitattu 25.11.2024. <https://www.typescriptlang.org/>

W3School 2024a. What is a Front-End Developer? Viitattu 24.11.2024. https://www.w3schools.com/whatis/whatis_frontenddev.asp

W3School 2024b. React Tutorial. Viitattu 24.11.2024. <https://www.w3schools.com/react/>

Kuvat

Kuva 1 Työtehtäviä GitHubissa.	9
Kuva 2 Hahmon oletustiedot.	13
Kuva 3 FSD arkkitehtuuri (FSD 2024).....	15
Kuva 4 Ongelma kehitysympäristön pystyttämässä uuteen branchiin.	22
Kuva 5 AttributesPricing-komponentti tyylimäärittelyllä.	23
Kuva 6 Oma toteutus.	24
Kuva 7 Esimerkkikoodin toteutus pylväsdiagrammista.....	25
Kuva 8 VSCodiumin versionhallintaa.	29
Kuva 9 Tämänhetkinen versio hahmojen kehitys - sivusta.	30
Kuva 10 Hahmojen kehitys sivu vaaditulla sisällöllä.	33
Kuva 11 Outouksia versionhallinnassa.	35
Kuva 12 Olisiko tämä merkki huonosta arkkitehtuurista?	36
Kuva 13 Issue kirjoittajille.	37
Kuva 14 Vastaavan ohjelmoijan tarkoittama ulkoasu sivupalkille.	39
Kuva 15 Kieliongelma näkyy olevan kehitysympäristössä.....	43
Kuva 16 Valmis tuotos. Mobiilikoossa hahmolistasta tulee alasettovalikko. Sivun asettelultaan responsiivinen.	45