



Ohjelmistosuunnittelu ja ohjelmistokehitysprojektin hallinta

Projektipäiväkirja, Lasse Simonen

Tiistai 10.06.2025:

Projekti potkaistaan käyntiin. Projektin vaatimusten esittely Ohjelmistosuunnittelu ja ohjelmistokehitysprojektin Teams-kanavalla.

Olin alustavasti pohtinut SpaceWar -pelin kehitystä kokonaisvaltaisena projektina, mutta vaatimusten määrittelyn myötä pelin refaktorointi front- ja backendiin kuulostaa paljon soveliaammalta projektilta. Siinäkin on ehkä työtä aivan tarpeeksi työtä aloittelevalle amatöörille. Gemini pilkkoi projektia alustavasti osiin teams-keskustelun aikana, ja se olisi ehkä aivan tehtävissä vaikka vastaan tulee unohdettuja ja uusiakin asioita.

Samalla se avaisi ovet moninpeliä varten, todennäköisesti se vaadittaisiin jos ikinä koskaan haluaisin siirtyä Itch.io:sta Steamiin, ja nähtävästi Itch.iokin kuitenkin hyväksyy toisaalla hostatut pelit, joiden frontend vain sattuu olemaan itch.io:ssa. Kuulostaa oikein hyvältä. Hommiin siis.

Aloitin projektisuunnitelman työstämisen.

Keskiviikko 11.06.2025

Jatkoin projektisuunnitelman työstöä.

Torstai 12.06.2025

Hinkattu projektisuunnitelmaa edelleen. Työ on edistynyt aika hyvin. Suunnitelma alkaa vaikuttaa aika tiukalta, mutta suunnitellulta. Se ikään kuin avautuu tässä suunnitelman työstämisen myötä teoriassa tehtävissä olevaksi proggikseksi. Aika jännä transformaatio.

Testaus ja johdonmukainen Gittailu yms jännittävät. Olen enemmänkin kaikki legot lattialle ja leikitään -tyyppi, mutta tähän vaatii ihan oikeasti johdonmukaisuutta suunnitelmassa pitäytymistä. Mutta askel kerrallaan.

Lähetetty projektisuunnitelma Jarkolle välicheckiä varten, josko ehtisi tarkastaa ennen lomille lähtöään.

Perjantai 13.6.2025

Aloitettu Näyttötöydokumentaatiokin jo nyt heti. En keksinyt miten ja missä esittelisin draw.io:lla tehdyt kaaviot, joten päätin, että tuo Projektisuunnitelma toimii uuden Näyttötöydokumentaation pohjana. Siinä olen vihdoin saanut muotoilut Wordissa kohdilleen ja tuo pelkistetty, kevyesti akateeminen tyyli miellyttää.

Wordin muut valmiit pohjat ovat ehkä vähän turhan värikkäitä ja ennen kaikkea liian erilaisia projektisuunnitelmadokumenttiin nähden, niin että tuo vanhan dokumentin uusi elämä on vain järkevin vaihtoehto. Design pysyy samana, ja voin suoraan tuoda / pitää projektisuunnitelman samassa dokumentissa, mutta laajennan vain otsikointia ja otsikkorakenteen muotoilua hieman. Nyt H1 -> OSA A, B, C jne, ja H2 -> Johdanto, määritelmät, kohderyhmä, jne jne. Toimii aika hyvin.

Tehty arkkitehtuurikaavio, käyttötapauskaavio, ja luokkakaavio. Yllärihidasta vielä. Tehty sanallisia selityksiä kaavioille ja tuotu näyttötöydokumenttiin.

Kaikkiaan aika näppärän näköinen vaikka itse sanonkin.

Jarkolta peukut projektisuunnitelmalle. "Suunnitelma ok, eikun hommiin." Ja oli itseiassa "todella hyvin laadittu projektisuunnitelma" woopwoop!

Tiistai 17.06.2025

Yllättävä työkeikka eilen maanantaina vei projektin hieman jälkijunaan heti alussa. Tuli tunne, että projekti etenee valon nopeudella: vasta viikko takana, mutta olen jo kuukauden jäljessä.

No, tässä saatiin kiinni paljon tänään. Näyttötyödokumentointiin saatu Osa B hyvälle mallille. Luotu osiot Järjestelmän arkkitehtuuri kaavioineen, käyttötapauskaavio esittelyineen, käyttöliittymäsuunnitelma rautalankakuvineen (päävalikko, pelin päänäkyvä, planeettavalikko, ryhmävalikko), luotu Luokkakaavio esittelyineen, Sekvenssikaavio (aluksen rakentaminen) esittelyineen, Aktiviteettikaavio tekoälyn toimintavuorosta.

Lisäksi luotu kansiorakenne, yhdistetty repositio Gitiin, tehty ensimmäinen commit alkuperäisellä yhden tiedoston koodilla. Tästä on hyvä jatkaa, ja lopulta taidammekin olla Gantt-kaavion aikataulussa, kun huomenna alan työstää backendiä. Ei huono. 😊

Keskiviikko 18.06.2025

Tänään kai saatu aika paljonkin aikaan, vaikka ei oikein siltä vielä tunnukaan. Luotu MongoDB, saatu skeemat aikaiseksi, irrotettu tekoäly omaksi AIController.js:ksi, alustettu GameManager.js tilaan, että se alustaa pelin ja luo mm. pelin tekoälyt ja pelaajat. Luotu server.js:n alku, testsocket.js ja test.http, client.js ja uusi index.html. Socket.IO saatu toimimaan...

Kaikenkaikkiaan nyt sellaisessa tilassa, että tietokanta kättelee paikallista serveriä. Saatu hello world läpi molempiin suuntiin. Saatu aikaan diff-viestit pelitilan eroavaisuuksista palvelimelta tietokantaan ja pelaajalle.

Onhan tässä. Tietokannan koot kasvavat aika nopeasti tähtien määrän johdosta, mutta tähän etsitään ratkaisua paaljon myöhemmin. En oikein enää edes tiedä mitä kaikkea olen saanut tänään aikaan, mutta lienen aikataulussa edelleen. Tai jopa edellä.

1. Käyttöliittymän "Start Game" -nappi tekee fetch-kutsun backendin /api/games/new -reittiin.
2. Asiakassovellus vastaanottaa onnistuneen vastauksen, joka sisältää uuden pelin gameId:n.
3. Tämän ID:n avulla asiakas liittyy oikeaan pelihuoneeseen Socket.IO:n join_game-tapahtumalla.
4. Palvelin vastaa tähän lähettämällä initial_state-tapahtuman, joka sisältää koko pelimaailman datan.
 - **Datapohjainen maailman renderöinti:** Tämä oli suurin saavutus. scene.js-moduuli ei enää itse arvo pelimaailmaa, vaan sen buildSceneFromState-funktio vastaanottaa palvelimelta saadun datan ja rakentaa sen perusteella Three.js-scenen – jokainen tähti ja alus luodaan nyt tietokannasta peräisin olevan tiedon pohjalta.
 - **Visuaalisen ilmeen palauttaminen:** Vanhan monoliitin yksityiskohtainen renderöintilogiikka (mm. tähtien värit omistajan mukaan, hehkuefektit, kotiplaneettojen erottuminen) siirrettiin onnistuneesti uuteen scene.js-tiedostoon. Peli näyttää nyt visuaalisesti jälleen tutulta ja toimivalta.
 - **Pohja reaaliaikaisille päivityksille:** client.js-tiedostoon luotiin socket.on('game_diff', ...) -käsittelijä, joka on nyt valmis vastaanottamaan ja välittämään palvelimelta tulevia pienempiä pelitilan päivityksiä scene.js:lle.

Päivän yhteenveto ja fiilis:

Tämän päivän työn tuloksena projektissa on nyt toimiva perusrakenne, jossa backend simuloi peliä ja frontend visualisoi sen tilan. Monet käyttöliittymän toiminnot saatiin tänään kytkettyä myös komentojen lähettämiseen, ja arkkitehtuurin vaikein osuus on nyt todennetusti toiminnassa. Tunne siitä, kun näkee backendin luoman maailman ilmestyvän ruudulle selaimessa, oli erittäin palkitseva ja todisti, että suunnitteluvaihe ja backendin rakennus on tehty oikein.

Nyt tuntuu siltä, että pahimmat itkut on ehkä itketty. "TEORIASSA" loppu on vain toiminnallisuuden siirtämistä ja ihan tavallista, puhdasta hikeä. Ehkä tässä ehtii vielä juhannustakin juhlia <3.

Perjantai 20.6.2025

Eilisen suuren optimismin jälkeen tänään palattiin syvälle juoksuautoihin. Päivä oli klassinen esimerkki ohjelmistokehityksen vuoristoradasta: yksi lisätty ominaisuus

paljasti kaksi piilevää, perustavanlaatuisia ongelmaa. Tästä huolimatta saatiin kuitenkin valtavasti uutta toiminnallisuutta peliin.

Päivän aikana toteutettiin pelin kannalta keskeisiä mekaniikkoja:

- Alusten liikkuminen tähtien välillä ja tähtien valloitusjärjestelmä.
- Planeetoilla tehtävät ostokset ja rakennusjonojen hallinta.
- Pelin nopeussäätimet ja kaksi erilaista taukotilaa: välilyönnin "pikatauko" ja ESC-näppäimen paluu päävalikkoon, josta voi jatkaa tai aloittaa uuden pelin.

Mutta sitten iski se kuuluisa todellisuus. Huomasin, että kun pelaaja sulki selaimen, serverille jäi kummitelemaan vanhoja pelisessioita. Nämä "zombie-pelit" jatkoivat eloaan ja resurssien keräämistä tyhjiydessä, mikä olisi pitkällä aikavälillä kestämaton ja kallis vika.

Tätä jäljitettiin Geminin kanssa tuntikausia. Kokeilimme monimutkaisia ajastimia ja elinkaarimalleja, mutta mikään ei tuntunut toimivan luotettavasti. Vika oli syvemmillä ja samalla yksinkertaisempi kuin osasimme aavistaa. Monen yrityksen ja erehdyksen – sekä useamman pahoittelun – jälkeen syy viimein löytyi: vika ei ollutkaan monimutkaisissa ajastimissa, vaan yhdessä ainoassa sanassa: Socket.IO:n disconnect-tapahtuma oli ajettava disconnecting-tapahtumana. Kun tämä muutos tehtiin, serveri alkoi siivota vanhat pelit pois välittömästi ja luotettavasti. Voitonriemu oli valtava!

Samassa rytkäksä otettiin käyttöön express-session, mikä on arkkitehtuurin kannalta valtava harppaus. Serveri ei ole enää yhden pelin "hiekkalaatikko", vaan se pystyy nyt palvelemaan useita pelaajia samanaikaisesti, jokaisella ollen oma, erillinen pelinsä.

Tänään ei ehkä saatu niin paljon uutta ja kiiltävää kuin eilen, mutta konepellin alla tehtiin perustavanlaatuisia ja kriittisiä korjauksia. Nyt järjestelmä on vakaampi kuin koskaan. Tästä voi hyvillä mielin lähteä juhannuksen viettoon.

Maanantai 23.06.2025

Maanantai juhannuksen jälkeen. Viikonloppu meni tunkatessa peliä, kun ei lopettaakaan oikein osannut, eikä nyt tärkeämpääkään ollut. Neljän päivän lähes tauoton koodaaminen on takana, ja aivot tuntuvat samanaikaisesti ylikierroksilla käyvältä prosessorilta ja täysin tyhjäksi puristetulta massalta. Viikonloppu, jonka piti olla juhannuksen juhlaa, muuttuikin syväksi sukellukseksi client-server-arkkitehtuurin sieluelämään. Toisin kuin perjantaina vielä oletin, tämä ei ollut suoraviivainen tehtävälista; tämä oli epätoivon ja oivallusten tarina. Uskoakseni peli on nyt kuitenkin vakaampi kuin perjantaina, ja monta siinä piilyyttä vikaa korjattu.

Kaikki alkoi, kuten niin usein, yhdestä ainoasta ärsyttävästä ongelmasta: viiveestä. Kun laivastoni saapui vihollisen hallitsemalle planeetalle, mitään ei tapahtunut. Kului 5, jopa 10 piinaavaa sekuntia, kunnes serveri viimein laski taistelun lopputuloksen ja lähetti sen clientille. Illuusio reaaliaikaisesta sodankäynnistä oli täysin Rikki ja epätoivo alkoi hiipiä pintaan.

Ensimmäinen yritys ja välitön umpikuja

Ensimmäinen, naiivi ajatukseni oli yksinkertainen: jos tieto ei kulje tarpeeksi nopeasti, lähetetään sitä useammin ja aggressiivisemmin. Muokkasin koodia lähettämään diff-päivityksiä ja tallennuskäskyjä lähes jatkuvalla syötöllä, toivoen että saisin taistelun lopputuloksen jotenkin puristettua nopeammin läpi.

Lopputulos oli katastrofi. Serveri alkoi kaatuilla välillä MongoDB:n parallelSaving-virheisiin ja 10x -nopeudella se kaatui lähes välittömästi. Yrittäessäni pakottaa nopeutta olin luonut järjestelmään kilpa-ajotilanteen, jossa samaa dokumenttia yritettiin päivittää useasta paikasta samanaikaisesti.

Välttämätön kiertotie: Taustalla tallennus

Lauantai kului tämän uuden, itse aiheutetun ongelman parissa. Minun oli pakko perääntyä ja miettiä koko tallennuslogiikka uusiksi. Syntyi `_saveInBackground()`-funktio `GameManager.js`-tiedostoon.

Tämä irrotti tietokantaan kirjoittamisen kokonaan pelin pääloopista. Järjestelmä vakautui välittömästi. Kaatumiset loppuivat ja peliä pystyi nyt ajamaan jopa 10x-nopeudella ilman ongelmia. Se oli itselle valtava tekninen voitto, mutta se oli voitto väärässä taistelussa. Alkuperäinen viiveongelman pysyi.

Aivot narikkaan ja tekoälyt apuun

Sunnuntain päätin pyhittää tämän ongelman selättämiseen. Käännyin modernin ohjelmoijan viimeisen oljenkorren puoleen: avasin rinnakkain ChatGPT:n ja Clauden. Aloin syöttää heille koodinpätkiäni ja heitellä heidän ideoitaan toisilleen, yrittäen seuloa hallusinaatioista käyttökelpoisia ajatuksen siemeniä.

Ja sitten se tapahtui. Keskellä keskustelua GameManagerin loopista, ChatGPT tokaisi jotain $N \times M$ -kompleksisuudesta. Se huomautti, että tapani tarkistaa taistelutilanteet oli tehoton. Olin itse sokeutunut sille, koska olin testannut peliä vain pienellä skaalalla. Tekoäly näki metsän puilta ja paljasti piilevän aikapommin.

Vaikka tämä ei suoraan liittynyt viiveongelmaan, tajusin sen olevan kriittinen. Ryhdyimme tekoälyjen kanssa välittömästi purkamaan ongelmaa. Lopputuloksena oli uusi, optimoitu logiikka, joka vähensi laskentatehon tarvetta dramaattisesti.

Laskelma: N*M-optimoinnin todellinen vaikutus

Ennen kuin mennään tarinassa eteenpäin, on syytä pysähtyä tähän hetkeksi. Lasketaan, kuinka suuri parannus tämä oli. Tässä karkea arvio:

- **Oletukset:** 150 tähteä, 300 alusta pelissä. Pelin logiikka pyörii serverillä yhden tickin sekunnissa (TICK_MS = 1000).
- **Vanha metodi (pahimmillaan):** Joka tick, looppi käy läpi kaikki 150 tähteä. Jokaisen tähden kohdalla se suodattaa KAIKISTA 300 aluksesta ne, jotka ovat kyseisellä tähdellä. Tämä vaati noin $150 * 300 = 45\ 000$ tarkistusoperaatiota joka sekunti vain alusten ryhmittelyyn. Vaikka 300 alusta tässä pelissä on yhdelle pelaajalle aika paljon, niin 4:llä pelaajalla se on itse asiassa aika maltillinen luku. Prototyypissä loppupelissä pelkästään pelaajalla on usein reilu 200 alusta.
- **Uusi metodi:** Joka tick, looppi käy läpi kaikki 300 alusta kerran ja sijoittaa ne Map-rakenteeseen tähden perusteella. Tämä vaatii **300 operaatiota**. Sen jälkeen se käy läpi vain ne tähdet, joilla oikeasti on aluksia (esim. 50 tähteä). Yhteensä noin $300 + 50 = 350$ operaatiota.

Lopputulos: Laskentatehon tarve väheni noin 45 000 operaatiosta ~350 operaatioon per tick. Parannus on yli 99%.

Vaikka en ollut sitä itse huomannut, ChatGPT pelasti tässä projektini täydelliseltä hyytymiseltä myöhemmässä vaiheessa.

Lopullinen oivallus: Ongelma ei ollut kellossa, vaan päässäni

Mutta edes tämän massiivisen optimoinnin jälkeen... viive oli yhä siinä. Turhautuneena heitin hanskat tiskiinkin ja päätin vain debugata tapahtumaketjun sekunti sekunnilta kielimallien kanssa. Vahvistimme, että prosessi oli jo niin nopea kuin se voi olla:

1. Clientin fysiikkamoottori siirtää aluksen visuaalisesti perille.
2. Serveri vahvistaa tickin aikana, että alus on matemaattisesti perillä.
3. Serveri ratkaisee taistelun samalla tickillä.
4. Serveri lähettää lopputuloksen clientille.

Lokien perusteella ei ollut mitään mistä nipistää. Olin kokeillut käydä taistelut eri putkien kautta, mutta viive pysyi aina vaan. Taistelun kulussa ja sen logiikassa on ehkä edelleen perustavanlaatuisen virhe jota en vain ole vielä löytänyt, mutta ainakin löysin henkisen korjauksen.

Jos serveri suorittaa taistelun sillä samalla tickillä kun millä se on saanut tiedon

taistelusta, ja jos tieto taistelun osapuolista väistämättä vaatii vahvistuksen tietokannasta, niin ehkä en vaan pysty korjaamaan tilannetta. Kaatuuko koko projekti? Pelaaminen tuntuu "wonkyttä" eikä pelaaja saa välitöntä palautetta tilanteesta. Monoliittiprototyypissä palaute oli alusten välitön räjähtäminen. Nyt räjähtäminen vaatii vahvistuksia, se taas vie aikaa, eikä sitä voinut kiertää. Tai en selkeästi ainakaan osannut kiertää sitä. Kyllähän noita on reaaliaikaisia sujuviakin pelejä, mutta omani ei sellainen näköjään ollut.

Mutta entä jos viihdyttäisin pelaajaa sillä välin, kun taistelukierrosten tarkistukset tehdään? Grafiikka tapahtuu pelaajan selaimessa ja se voidaan laukaista heti vaikka vahvistuksia asioille ei olisi vielä saatu.

Syntyi CombatEffectGroup: graafinen taistelun pauhu tähden ympärillä sillä välin kun tieto kulkee edestakaisin ja prosessit käydään läpi. Se on pelaajalle välitön palaute, vaikkei lopputulos vielä ratkennutkaan. Ja vot! Peli TUNTUI heti erilaiselta. Se oli jälleen nopea, reagoiva ja tyydyttävä. Siinä oli taas ripaus sitä alkuperäisen monoliitin taikaa.

Tämä viikonloppu opetti aika paljon. Se opetti, että joskus oikea ratkaisu ei ole tekninen optimointi, vaan käyttäjäkokemuksen ymmärtäminen. Se opetti, että joskus on pakko rakentaa vakaa perusta (taustalla tallennus) ennen kuin voi korjata sen päällä olevan ongelman. Ja se opetti, että joskus paras sparrauskumppani voi olla tekoäly, joka näkee sen, mille itse on tullut sokeaksi. Olo on uupunut, mutta samalla voittamaton.

Tiistai 24.6.2025

Päivän päätavoitteena oli hioa tekoälyn logiikka huippuunsa. Tavoitteena oli saada se kehittämään valtakuntaansa kokonaisvaltaisesti ja orgaanisesti, samaan tapaan kuin alkuperäisessä monoliittiprototyypissä. Tämä onnistui erinomaisesti: tekoäly tekee nyt älykkäämpiä päätöksiä niin talouden, rakentamisen kuin laivaston koostumuksenkin suhteen. Se oli merkittävä edistysaskel, mutta menestys toi mukanaan uuden, hiipivän ongelman.

Kun laitoin neljä tekoälyä taistelemaan toisiaan vastaan seuratakseen pelin kehittymistä, huomasin jotain häiritsevää. Alun sulavan 60 ruudun sekuntivauhdin jälkeen ruudunpäivitys alkoi yllättäen takkuilla. FPS tippui hitaasti mutta vääjäämättä, eikä se enää palautunut entiselleen. Peli muuttui raskaaksi ja epämiellyttäväksi pelata.

Päivä muuttui systemaattiseksi vianetsinnäksi. Palasimme jälleen kerran siivousjärjestelmien ja CombatEffectGroup-logiikan kimppeun, mutta syytä ei tuntunut löytyvän. Jotta ongelman todellisuus ja laajuus saataisiin todennettua, rakensin peliin uudet debug-työkalut:

- **F3** tuo näkyviin suorituskykymonitorin.

- **F4** tulostaa konsoliin yksityiskohtaista dataa scenen tilasta.
- **F5** käynnistää manuaalisen muistinsiivouksen.

Näiden työkalujen avulla pystyin todentamaan, että FPS:n tippuminen ei ollut kuvitelmaa. Vihdoin iltamyöhään, Chromen DevTools-profilointityökalujen avulla, syylliset löytyivät. Lähtöoletuksena seurasin ensin CPU usagea, sekä JS HeapSize -käyriä. Kova CPU-usage -käyrä kertoisi optimoimattomasta koodista, ja JS Heap Size -käyrän loputon kasvaminen toimimattomasta siivousjärjestelmästä ja muistivuodosta.

Heap Size vaikutti terveeltä, ikään kuin sydänkäyrältä tai hitaalta sahalaitakuviolta missä Heap Size nousi tasaisesti, kunnes laski peruslukemiin - ja alkoi nousun uudelleen. Pikaisella seurannalla se pysyi kuitenkin symmetrisenä mikä kertoi, että siivousjärjestelmä toimi kuten sen kuuluu.

CPU sen sijaan huiteli 100% käyttölukemissa. Tämä vaati lisäselvitystä, että mitkä prosessit venyivät ja johtavat jopa uuden pelitietokoneen kyykkäämiseen. Kaksi asiaa nousi selvästi esiin raskaina prosesseina:

1. **Bokeh-efekti:** Jälkikäsitteleyefekti, joka sumentaa taustaa, oli odotettua raskaampi näin monen objektin kanssa.
2. **Jatkuva alusten luonti:** Jokainen uusi alus loi täysin uuden 3D-objektin, geometrian ja materiaalin. Satojen alusten myötä tämä alkoi kuormittaa muistia ja suoritinta valtavasti.

Päivä ei siis päättynyt vielä korjaukseen, mutta se päättyi diagnoosiin. Nyt tiedän tarkalleen, mitkä ovat suurimmat suorituskykyjarru. Koska projektisuunnitelma vaatii vähintään tasaista 30 FPS:n ruudunpäivitystä, näiden korjaaminen on seuraava prioriteetti. Huomenna alkaa jahti näiden kahden pullonkaulan optimoimiseksi.

Keskiviikko 25.06.2025: Suorituskyvyn metsästys ja instansoinnin vallankumous

Eilisen diagnoosin pohjalta päivä omistettiin kahden suurimman suorituskykyjarrun eliminomiselle. Edistys oli merkittävää ja konkreettista. Ensimmäinen, nopea korjaus oli luopua raskaasta Bokeh-jälkikäsitteleyefektistä, joka yksinään vapautti huomattavasti näytönohjaimen resursseja.

Päivän päätyö ja todellinen mullistus oli siirtyminen alusten renderöinnissä instansointiin (InstancedMesh). Tämän massiivisen refaktoroinnin myötä peli ei enää luo jokaiselle alukselle omaa, raskasta 3D-objektia. Sen sijaan, että sadat tai tuhannet

1. **Yhden tekoälyn pelissä** valloituksia ja planeettapäivityksiä tapahtui harvoin, joten tämä "muistivuoto" oli hidasta eikä juuri vaikuttanut suorituskykyyn.
2. **Usean tekoälyn sodassa** planeetat vaihtoivat omistajaa jatkuvasti. Jokainen valloitus laukaisi bugisen päivityksen, mikä johti tuhansien ylimääräisten objektien kertymiseen sceneen ja lopulta pelin hyytymiseen. Ongelma siis liittyi taisteluihin, mutta syy ei ollut taisteluefekteissä vaan niiden seurauksena tapahtuvassa planeettadatan päivityksessä.

Tämä oli klassinen esimerkki vaikeasti havaittavasta bugista, jossa yksittäinen pieni virhe kertautuessaan aiheuttaa katastrofaalisen suorituskykyongelman. Löytö alleviivasi systemaattisen debuggauksen ja oireiden (hidastuminen taisteluiden aikana), ei vain oletettujen syiden (taisteluefektit), tutkimisen tärkeyttä.

Kun tämä virhe korjattiin ja puolustusrenkaat instansoitiin, pelin suorituskyky teki valtavan harppauksen. Nyt peli pyörii neljälläkin tekoälyllä ja 250 planeetalla vakaasti ja kevyesti, suorittimen kuorman pysyessä 35-45 % tuntumassa. Suurimmat tunnetut suorituskykyongelmat on nyt ratkaistu.

Iltapäivä: Tekoälyn strateginen uudistus

Aamupäivän massiivisen suorituskykyparannuksen jälkeen iltapäivä voitiin omistaa pelin strategisen syvyyden parantamiseen, erityisesti tekoälyn osalta.

Tekoälyohjain (AIController.js) koki täydellisen uudistuksen. Merkittävin lisäys on **strateginen laivastonkeräyslogiikka**. Sen sijaan, että tekoäly lähettäisi aluksia hyökkäykseen heti niiden valmistuttua, se osaa nyt arvioida vahvasti puolustettuja kohteita ja koota joukkonsa ennalta määrättyyn "kokoontumispisteeseen" massiivista, koordinoitua hyökkäystä varten.

Lisäksi tekoäly analysoi nyt aktiivisesti vihollislaivastojen koostumusta ja **mukauttaa omaa laivanrakennustaan vastatakseen uhkaan**. Jos vihollinen panostaa hävittäjiin, tekoäly alkaa tuottaa niitä vastaan tehokkaita destroyereita, jos destroyereita, niin cruisereita, jos cruisereita, niin fightereita. Alusten toiminta taistelussa on kivi-paperisakset -tyyppinen, joten rakennusjonon tulisi huomioida tämä. Ja näin se nyt tekee, enää loppupeli ei ole pekkien cruisereiden pelikenttä.

Myös aamupäivällä toteutettua instansointijärjestelmää hiottiin edelleen. Kun yksikkö tuhoutuu, sen käyttämä paikka instanssipuskurissa vapautetaan ja **kierrätetään uudelle yksikölle**. Tämä parantaa pelin pitkän aikavälin vakautta ja tehokkuutta.

Päivä oli kokonaisuudessaan yksi projektin tuotteliaimmista. Kriittisen suorituskykyongelman ratkaiseminen avasi oven merkittäville pelillisille parannoille, jotka tekevät pelikokemuksesta paitsi sulavan, myös haastavamman ja strategisemmän.

Perjantai 27.06.2025

Tämän päivän työ keskittyi pelikokemuksen hiomiseen ja useiden tärkeiden, pelaajalle näkyvien ominaisuuksien viimeistelyyn. Eilisten massiivisten suorituskykyoptimointien jälkeen oli palkitsevaa lisätä toiminnallisuuksia, jotka tekevät pelistä ammattimaisemman ja ennen kaikkea pelattavamman. Päivän aikana saatiin valmiiksi useita keskeisiä parannuksia.

Päivän keskeiset tekniset toteutukset:

- **Ryhmien luonti ja pikanäppäimet:** Peliin toteutettiin strategiapeleistä tuttu kontrolliryhmäjärjestelmä. Pelaaja voi nyt luoda ryhmiä valitsemistaan aluksista CTRL + [0-9] -näppäinyhdistelmällä ja valita kyseisen ryhmän nopeasti painamalla vastaavaa numeronäppäintä.
- **Kameran automaattinen kohdistus:** Pelikokemusta parannettiin merkittävästi lisäämällä kaksi uutta kameran kohdistustoimintoa:
 - Tähdien tuplaklikkaaminen keskittää kameran nyt sulavasti kyseiseen tähtijärjestelmään.
 - Kontrolliryhmän pikanäppäimen tuplapainallus valitsee ryhmän ja kohdistaa kameran sen sijaintiin.
- **Parannettu visuaalinen palaute:** Käyttöliittymän selkeyttä ja intuitiivisuutta parannettiin:
 - Valitun planeetan ympärille ilmestyy nyt selkeä kohdistinrenkas ("targeting reticule").
 - Tähtien ja niiden välisten starlane-yhteyksien hehkuefektit reagoivat nyt hiiren kursoriin, korostuen kun kursori viedään tähden päälle.
- **Taistelun ja tekoälyn tasapainotus:** Pelimekaniikkaa hiottiin edelleen. Taistelujärjestelmää tarkistettiin ja tasapainotettiin reilummaksi. Lisäksi tekoälyn uhka-arviointia korjattiin niin, että se osaa nyt paremmin arvioida planetaarisen puolustuksen (PD) voimakkuutta ennen hyökkäyspäätöksen tekemistä.
- **Slipstream-ominaisuuden viimeistely:** Ominaisuus on nyt toiminnallisesti valmis. Animaatiologiikkaa korjattiin niin, että se vastaa täysin serverin laskentaa, ja laivueet liikkuvat nyt visuaalisesti täydellisessä synkronissa.
- **Puolustusrenkaiden korjaus:** Aiemmin havaitut ongelmat planetaaristen puolustusrenkaiden muodostumisessa korjattiin. Renkaat ilmestyvät ja poistuvat nyt luotettavasti planeetan puolustustason mukaisesti.

Päivän yhteenveto ja fiilis:

Nämä ovat juuri niitä ominaisuuksia, jotka muuttavat teknisen demon oikeaksi peliksi. Mahdollisuus hallita yksiköitä pikanäppäimillä ja saada selkeää visuaalista palautetta tekee valtavan eron. Projekti tuntuu kokonaisemmalta ja viimeistellymmältä kuin koskaan aiemmin. Hyvillä mielin viikonlopun viettoon.

Lauantai 28.06.2025: Viimeiset silaukset ja ansaittu virstanpylväs

Tänään saavutettiin merkittävä virstanpylväs. Intensiivisen kehitysvaiheen jälkeen oli aika lisätä viimeiset, tunnelmaa luovat elementit ja jopa kurkottaa projektisuunnitelman "nice to have" -laatikkoon. Päivä oli täynnä onnistumisen tunteita, kun peli alkoi tuntua ja kuulostaa täysin valmiilta kokonaisuudelta.

Päivän aikana viimeisteltiin kaksi tärkeää osa-aluetta:

- **Galactic Hub -ominaisuus:** Projektisuunnitelman MoSCoW-priorisoinnin Could-tason toive toteutettiin. Peliin lisättiin uusi, kallis loppupelin rakennus, Galactic Hub, joka luo dynaamisen starlane-verkoston kaikkien sen omistavien tähtien välille. Tämä lisää peliin uuden strategisen ulottuvuuden ja palkitsee pitkälle edenneen pelaajan.
- **Äänimaailman palautus:** Pelin sielu palautettiin toteuttamalla äänimaailma täsmälleen alkuperäistä monoliitti.html-versiota vastaavaksi. Rauhoittava taustadrone, napakka klikkausääni ja hienovarainen hover-efekti ovat nyt kaikki osa uutta arkkitehtuuria, viimeistellen pelikokemuksen ja tuoden siihen tuttua tunnelmaa.

Päivän yhteenveto ja fiilis:

Refaktoroinnin osalta projekti on nyt valmis. Kaikki alkuperäisessä prototyypissä olleet ominaisuudet on siirretty onnistuneesti uuteen, vankkaan client-server-arkkitehtuuriin, ja päälle on rakennettu lukuisia parannuksia ja täysin uusia ominaisuuksia.

Loppupäivä onkin pyhitetty sille, mille pelit on tehty: pelaamiselle. On täydellinen hetki istahtaa alas, kaataa lasi viiniä ja nauttia ensimmäisestä aidosta pelisessiosta.

Tämä on luonteva ja ansaittu hetki pysähtyä ja todeta, että koodi on valmis. Projektin ensimmäinen, raskain osa on takana. Seuraava vaihe, parin viikon ansaitun tauon jälkeen, onkin sitten selkeä: kommenttien siistiminen, dokumentaation viimeistely vastaamaan toteutunutta arkkitehtuuria, testien kirjoittaminen ja ajaminen, serverin pystytys ja lopulta kaiken paketointi osaksi näyttötyötä.

Torstai 10.07.2025

Parin viikon "talkoolomalla" on ollut kolme vapaapäivää, jotka olen viettänyt projektin parissa. Näiden kolmen vapaapäivän aikana on saatu aikaan merkittäviä edistysaskeleita. Suurin voitto on ollut toimivan pelipalvelimen pystyttäminen render.com-palveluun ja pelin onnistunut integrointi osaksi CV-sivujani osoitteessa www.lassesimonen.fi/spaceWar/. Pelin pelaaminen interwebissä sai sukat pyörimään jaloissa; refaktorointiprojekti on oikeasti onnistunut.

Ainoa nykyinen haaste on Renderin ilmaisversion 15 minuutin aikakatkaisu, joka katkaisee pelin. Harkitsen vakavasti palvelun päivittämistä maksulliseen Starter-pakettiin (7 €/kk) tämän rajoituksen poistamiseksi.

Toinen suuri urakka on ollut pelin koodin kokonaisvaltainen kommentointi JSDoc-muotoon. Työ on ollut puuduttavaa, mutta samalla erittäin tarpeellista. Huomasin, että olin jo ehtinyt unohtaa monia pelin toiminnallisuuksien yksityiskohtia. Kommentointiprosessi on toiminut erinomaisena kertauksena ja muistuttanut mieleen pieniä, mutta tärkeitä asioita pelin logiikasta. Nyt kommentoitavana on enää scene.js, jonka jälkeen tämä massiivinen dokumentaatiovaihe on valmis. Se on tosin pelin suurin tiedosto, joten vielä on urakkaa jäljellä. Oli kuitenkin hyvä päätös käyttää nämä harvat vapaapäivät tähän, sillä se selkeytti koodia huomattavasti tulevaa ylläpitoa ja jatkokehitystä varten vielä ennen kuin ehdin unohtaa aivan kaiken.

Maanantai 14.07.2025: Elävä tutoriaali ja tarinankerronnan syventäminen

Viimeiset pari päivää ovat kuluneet intensiivisesti uuden ominaisuuden parissa: pelin sisäisen tutoriaalin. Alun perin ajatuksena oli luoda vain perusohjeet pelin mekaniikoista, mutta projekti on ottanut merkittäviä askeleita kohti kokonaisvaltaisempaa tarinankerrontaa ja maailman elävöittämistä.

Keskeiseksi osaksi on muodostunut dynaaminen järjestelmä, jossa neuvonantajat Elara ja Valerius kommentoivat aktiivisesti pelin tapahtumia. Sen sijaan, että tutoriaali olisi vain staattinen ohjekirja, se on nyt reaktiivinen ja tarinallinen elementti, joka syventää pelikokemusta. Neuvonantajat reagoivat nyt pelaajan saavutuksiin ja strategiaan valintoihin suhteessa tekoälyvastustajaan.

Tämä tekee pelimaailmasta huomattavasti elävämmän. Neuvonantajat voivat esimerkiksi:

- Kehua pelaajaa, jos tämä onnistuu valloittamaan merkittävän osan galaksista ilman yhtään tappiota.
- Ihmetellä, kun tekoälyt ajautuvat sotaan keskenään.

- Käydä dialogia pelaajan taloudellisesta tilanteesta, varoittaen esimerkiksi liian suurista laivaston ylläpitokuluista.
- Antaa strategisia neuvoja, kuten kehotuksen iskeä vihollisen kehittyneisiin kaivosmaailmoihin.

Tämä on osoittautunut yllättävän antoisaksi vaiheeksi. Vaikka alun perin kyse oli vain opastuksesta, järjestelmästä on kasvanut tehokas työkalu pelin tarinan ja tunnelman syventämiseen. Hahmoista on tullut persoonallisia ja heidän kommenttinsa tuovat peliin uuden, reaktiivisen tason. Samalla luotiin myös "Skip Tutorial" -toiminto, joka kunnioittaa kokeneempien pelaajien aikaa.

Tämä menee ehdottomasti projektisuunnitelman ulkopuolelle, mutta on jotain, mikä peliin on ollut tarkoitus lisätä jo pitkään. Olkoon se MoScoW:n nice to have osa 2.

Tiistai 15.07.2025: Dynaamisen tarinankerronnan viimeistely ja toteutus

Eilisen pohdinnan ja suunnittelun jälkeen tänään keskityttiin täysillä dynaamisen tutoriaali- ja tarinajärjestelmän toteuttamiseen. Työ oli antoisaa, ja lopputulos ylitti jopa omat odotukseni. Järjestelmä, joka alkoi yksinkertaisena opastusmekanismina, on nyt pelin elävä ja hengittävä sydän.

Päivän aikana koodattiin ja testattiin toimivaksi laaja joukko uusia, dynaamisia laukaisimia (triggers). Neuvonantajien ja jopa tekoälyn kyky kommentoida pelin tapahtumia on nyt huomattavasti aiemmin visioitua syvällisempi. Toteutettuja ominaisuuksia ovat muun muassa:

- **Pelaajan strategian analysointi:** Järjestelmä tunnistaa ja kommentoi nyt aktiivisesti pelaajan pelityyliä. Se antaa palautetta esimerkiksi aggressiivisesta laajentumisesta pienellä laivastolla ("Empire Sprawl") tai vahvasti puolustuksellisesta "kilpikonna"-taktiikasta.
- **Kontekstitietoinen sotatilanne:** Yksinkertaisen tappiolaskurin sijaan järjestelmä antaa nyt älykkäämpää palautetta. Se voi kehua pelaajaa, vaikka tämä kärsisi tappioita, jos kokonaissotatilanne on voitollinen ("Winning Attrition"). Vastaavasti se varoittaa, jos kulutussota on kääntymässä pelaajaa vastaan.
- **Laajennetut virstanpylväät:** Kehuja ja kommentteja annetaan nyt porrastetusti useissa eri vaiheissa, kuten galaksin valloituksen edetessä tai pelaajan rakentaessa laajan Galactic Hub -verkoston. Tämä palkitsee pelaajaa pitkäjänteisestä pelaamisesta.

- **Tekoälyn oma ääni:** Tekoäly ei ole enää hiljainen vastustaja. Se lähettää satunnaisia, kryptisiä viestejä ja voi jopa kommentoida pelaajan tappioita, mikä lisää sen persoonallisuutta ja uhkaavuutta.

Dialogia ja sen laukaisuehtoja hiottiin useaan otteeseen, jotta lopputulos olisi mahdollisimman luonnollinen ja immerstiivinen. Vaikka tämä ominaisuus onkin alkuperäisen projektisuunnitelman ulkopuolella, sen tuoma syvyys ja persoonallisuus peliin on korvaamaton. Se muuttaa staattisen pelilaudan eläväksi, reagoivaksi maailmaksi. Tämä oli ehdottomasti "nice to have" -osion arvoinen lisäys.

Torstai 17.07.2025:

Tänään saavutettiin jälleen merkittäviä virstanpylväitä, kun viimeisetkin suuret kokonaisuudet loksahivat paikoilleen ja projekti otti harppauksen kohti lopullista valmistumista.

Aiemmin aloitettu, mittava scene.js-tiedoston kommentointiurakka saatiin tänään päätökseen. Koko sovelluksen koodikanta on nyt perusteellisesti dokumentoitu JSDoc-muotoon, mikä oli yksi projektin keskeisistä tavoitteista ja selkeyttää jatkokehitystä huomattavasti.

Päivän toisena suurena saavutuksena peli julkaistiin onnistuneesti myös itch.io-alustalla (<https://pecatus.itch.io/spacewar/>). Tämä vaati pientä säätöä palvelimen CORS-asetuksiin, jotta itch.io:ssa isännöity frontend sai luotettavasti yhteyden Render.com-palvelimella pyörivään backendiin. Luotu tämän kunniaksi uusi itch-kuvake pelille:



Päivän kruunasi onnistumisen tunne, kun aiemmin ReferenceError-virheitä aiheuttanut tietokannan automaattinen siivousajo toimi ensimmäistä kertaa täydellisesti. Pienen, mutta kriittisen korjauksen jälkeen oli hienoa todistaa, kuinka palvelin tunnisti ja poisti kymmeniä vanhentuneita pelejä tietokannasta juuri kuten pitikin.

Dokumentaation ollessa valmis, pelin julkaistuna ja taustajärjestelmien todistetusti toimivina, projekti tuntuu nyt erittäin eheältä ja valmiilta. Seuraavat askeleet kohti näyttötyön viimeistelyä tuntuvat nyt selkeiltä.

Seuraavaksi palataan dokumentaation pariin, ja tarkastellaan, että näyttötyödokumentaation kaaviot ja väitteet ohjelmiston toimintaperiaatteista pitävät paikkansa uuteen todellisuuteen verraten. Sitten testauksia. Jotain viimeistelyä varmasti vielä vaaditaan, mutta sittenpä tuo alkaisi olla koossa.

Maanantai 21.07.2025:

Viikonloppu ja tämä päivä ovat kuluneet projektin viimeistelyn parissa, siirtyen luovasta koodauksesta järjestelmälliseen laadunvarmistukseen ja dokumentointiin. Pääpaino on ollut testauksessa ja näyttötyödokumentaation viimeisen osan, Osa C:n, viimeistelyssä.

Jest-testien pystyttäminen vaati hetken pohdintaa ja klassisten polkuvirheiden korjailua, mutta lopulta kaikki viisi vaadittua yksikkötestiä saatiin ajettua onnistuneesti läpi. Tämä vaihe tuntui hyvin erilaiselta kuin aiempi "kaikki legot lattialle ja leikitään" -kehitysvaihe – vähemmän luovaa ongelmanratkaisua ja enemmän systemaattista todentamista. Samalla se toi kuitenkin valtavan luottamuksen backend-koodin vakauteen ja oikeellisuuteen; nyt on mustaa valkoisella, että ydinlogiikka toimii kuten pitää.

Samalla olen työstänyt näyttötyödokumentaatiota valmiiksi luoden vaaditut raja-arvotaulukot ja hioen testauksen tuloksia ja analyysiä. Eryyisen antoisaa oli analysoida F3-debug-työkalulla kerättyä dataa suoraan Render.com-palvelimelta. Data vahvisti aiemmat epäilyt: peli täyttää asetetut suorituskykytavoitteet, mutta palvelimen suorituskyky muodostuu pullonkaulaksi äärimmäisissä stressitilanteissa. Tämän analysoiminen ja kirjaaminen tuntui ammattimaiselta ja tärkeältä osalta prosessia.

Projekti tuntuu nyt todella olevan paketissa. Kaikki suuret kokonaisuudet ovat valmiita ja testattuja. Enää huomina README-tiedoston kirjoitus, ja sitten kaikki suunnitelman mukaiset osiot ovat valmiita esiteltäväksi. Fiilis on helpottunut ja itsevarma.