



Toni Kivelä

Reaaliaikainen pelitulosten integrointi HTML-sivulle

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Insinöörityö

25.4.2022

Tiivistelmä

Tekijä: Toni Kivelä
Otsikko: Reaaliaikainen pelitulosten integrointi HTML-sivulle
Sivumäärä: 35 sivua
Aika: 25.4.2022

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Tieto- ja viestintätekniikka
Ammatillinen pääaine: Pelisovellukset
Ohjaaja: Lehtori Antti Laiho

Insinööriyössä kartoitettiin CS:GO-pelin reaaliaikaisen integraation mahdollisuuksia, joita voidaan hyödyntää verkkosivuilla muodostamaan reaaliaikaista pelidataa seuraajien nähtävälle, jotta myös ne ottelut, joissa ei ole suoratoistoa saatavilla, saavat seurantapalvelun. Insinööriyössä tarkasteltiin myös projektityöskentelyn kautta tulleita tilaajan uusia tarpeita, joita ei aloituksessa tullut esille, jolloin ne toteutetaan erikseen omana työnään. Insinööriyö luo pohjan reaaliaikaisten tulosten tuomiselle verkkosivulle.

Insinööriyössä perehdyttiin erilaisiin toteutusvaihtoehtoihin, joissa käytetään Valven Counter-Strike: Global Offensive Game State -integraatiota. Vaihtoehtoja oli muutamia, ja jokainen oli toteutettu hieman eri tarkoituksiin, kuin mikä oli tarve reaaliaikaisessa integraatiossa. Suoraan reaaliaikaisen integraation toteutusta HTML-sivulle tulosten näyttämiseksi reaaliaikaisesti ei ollut löydettävissä.

Reaaliaikaisessa integraatiossa toteutettiin CS:GO-peliin tulosten keräämiseen tuote, joka tuottaa reaaliaikaista dataa pelipalvelimelta HTML-tiedostoon. Toimeksiantajana oli yhdistys, joka järjestää yritysten sisäisiä kilpailullisia e-urheiluturnauksia eri peleissä.

Insinööriyön tuloksena syntyi valmis reaaliaikainen integraatio, josta muodostuu HTML-sivu. Pelien tulokset tulevat reaaliaikaisesti sivulle nähtäväksi. Työn pohjalta voidaan jatkokehittää tulospalvelua laajentumaan esittämään enemmän pelidataa peleistä, ja jopa pelaajakohtaista dataa voidaan reaaliaikaisesti tuloksiin tuoda näkyviin. Toteutuksen pohjalta voidaan lähteä jatkokehittämään yhdistyksen omaa reaaliaikaisesta integraatiota verkkosivulle, johon data kerätään ja yhdistetään peleistä saatavaa reaaliaikaista dataa suoraan pelipalvelimelta verkkosivuille.

Avainsanat: reaaliaikainen integraatio, game state -integraatio, integraatio

Abstract

Author: Toni Kivelä
Title: Live score integration from game to HTML page
Number of Pages: 35 pages
Date: 25th April 2022

Degree: Bachelor of Engineering
Degree Programme: Information and Communications Technology
Professional Major: Game applications
Instructor: Antti Laiho, Senior Lecturer

The thesis explored the possibilities of live integration of the CS: GO game, which can be utilized on the website to generate real-time game data for followers to view, so that even those matches that do not have streaming are available to the tracking service itself. The thesis also reviews the client's new needs that have arisen through project work, which were not raised at the beginning, in which case they are implemented separately as their own work. The thesis creates the basis for bringing live results to your website.

Various implementation options using Valve's Counter-Strike: Global Offensive Game State Integration are reviewed in the thesis. There are a number of alternatives, each implemented for slightly different purposes than the need for live integration, a direct live integration implementation to display the results in real time was not available.

In real-time integration, a product that collects real-time data from the game server to an HTML file was implemented to collect CS: GO game results in real time. The client was Kanaliiga Ry, which organizes intra-company e-sport game tournaments in various games.

The result of this study was a finished live integration, which forms an html page. The results of the games will be displayed on the page in real time. Based on the work, the results service can be further developed to expand to present more game data from games, and even player-specific data can be displayed in live results. From the bottom of the implementation, it is possible to further develop Kanaliiga's own live integration into the website, where data is collected and real-time data from games is combined directly from the game servers to the website.

Keywords: Live integration, Game State Integration, Integration

Sisällys

Lyhenteet

1	Johdanto	1
2	Kilpailullinen elektroninen urheilu yritysten välillä	1
2.1	Kanaliigan historiaa	4
2.2	CS:GO-pelin historia	6
2.3	Valve Counter-Strike: Global Offensive Game State Integration - työkalu	7
2.4	LinuxGSM-palvelinratkaisu	8
2.5	Kanaliigan integraatiotarve	11
3	Reaaliaikaisen pelitilanteen integraatio verkkosivuille	12
3.1	Integraatiolla haettavien tietojen tavoite	12
3.2	Integraation eri toteutusvaihtoehdot	12
4	Reaaliaikaisen CS:GO-pelitilanteen integraatio verkkosivulle	14
4.1	Integraation tavoitteet	14
4.2	Integraation toteutus	15
4.3	Integraation tiedostojen siirto	16
4.4	Integraation HTML-sivun koostaminen	17
5	CS:GO- reaaliaikaisen pelitilanteen integraation lopputulos	19
5.1	Analyysi reaaliaikaisen pelitilanteen integraatiosta	23
5.2	Integraation reaaliaikaisen pelitilanteen testaus	25
5.3	Reaaliaikaisen pelitilanteen integraation seuraavat kehitysvaiheet	27
6	Yhteenveto	30
	Lähteet	32

1 Johdanto

Insinööriyössä tehdään integraatio Counter Strike: Global Offensive -pelin ja Kanaliiga Ry:n verkkosivujen välille. Integraation jälkeen pelejä voidaan seurata reaaliaikaisina tuloksina suoraan Kanaliiga Ry:n verkkosivuilta. Työ sisältää tiedon noutamisen pelipalvelimen kautta saatavista tiedoista, joista muodostetaan ihmisen luettavaa tiedostoa integraation avulla. Saadusta datasta muodostetaan HTML-sivuja, joita voidaan yhdistää suoraan yhdeksi verkkosivuksi myöhemmässä vaiheessa. Integraatio mahdollistaa kaikille kiinnostuneille oman tai muun yrityksen joukkueen pelien seuraamisen reaaliaikaisesti suoraan Kanaliiga Ry:n verkkosivujen kautta. Insinööriyössä käytetään omaa palvelinta, jossa on Ubuntu-käyttöjärjestelmän päällä toimiva LinuxGSM-pelipalvelin, joka mahdollistaa Counter Strike: Global Offensive -pelin järjestämisen paikallisesti ja julkisessa verkossa.

Kanaliigalla on käytössä tulospalveluita Kanaliiga Statistics -sivustolla, jossa voidaan nähdä pelien tuloksia eri lajeissa, mutta nämä tulokset eivät tällä hetkellä päivity reaaliaikaisesti peleistä, jolloin tulosten seuranta tapahtuu vasta, kun pelit on jo pelattu. Sivustolta voidaan seurata pelien sisältämää dataa sarja-, joukkue- ja pelaajatasoilla, ja nähtävissä on myös pelien historiallista dataa aiemmilta kausilta. [1.]

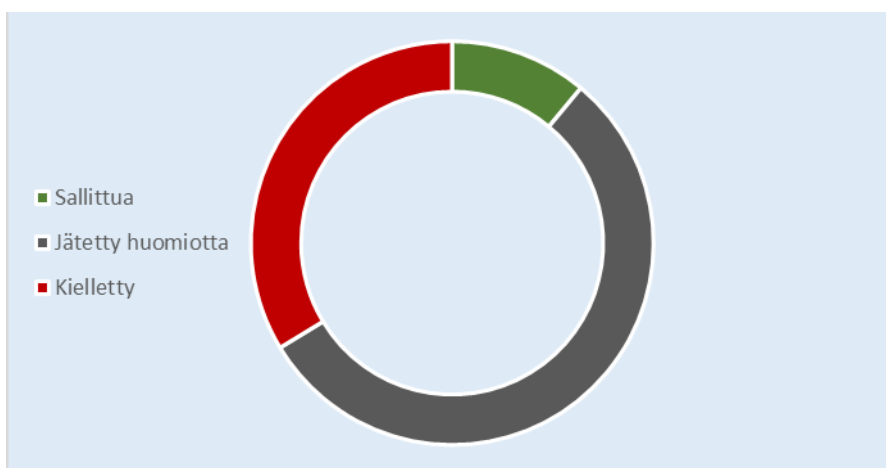
2 Kilpailullinen elektroninen urheilu yritysten välillä

Kilpailullinen elektroninen urheilu on viime vuosina alkanut nousta esille, kun sukupolvet, jotka ovat aloittaneet verkkopelaamisen, ovat nyt työelämässä vahvasti kiinni. Mahdollisuudet tapahtumien järjestämiseen verkossa ovat parantuneet, kun kehitys on vienyt verkkoteknologiaa ja muuta teknologiaa eteenpäin nopeasti siitä, missä oltiin n. 40 vuotta sitten.

Yritykset ovat vapaaehtoisvoimin tehneet erilaisia harrasteryhmiä, joissa työyhteisön näkökulmasta haetaan sitoutumista ryhmään ja sitä kautta yritykseen.

Yrityksen kannalta tällaista toimintaa kannattaa tukea erilaisten harrastetilojen, toimintamaksujen tai muiden maksujen kattamisella, jotta kaikilla on yrityksessä mahdollisuus osallistua vapaa-ajalla tapahtuviin ryhmäytymistä edistäviin tapahtumiin. Hyöty syntyy siitä, kun yksilöt tuntevat kuuluvansa isompaan ryhmään, jossa olennaisin ja sitovin osa on yritys, jossa he kaikki työskentelevät.

Keskimäärin 34 % eri aloista on kieltänyt pelaamisen töissä, 11 % on sallinut pelaamisen työajalla, ja 55 %:ssa työaloista tämä vaihtoehto on jätetty huomioimatta (kuva 1). Pelaaminen on osassa työpaikoista kiellettyä työajalla, osassa sallittua, ja joissain työpaikoissa tämä mahdollisuus on jätetty huomioimatta kokonaan. Keskimäärin eri aloilla yleissuhtautuminen pelaamiseen on neutraali. [2, s. 33.]



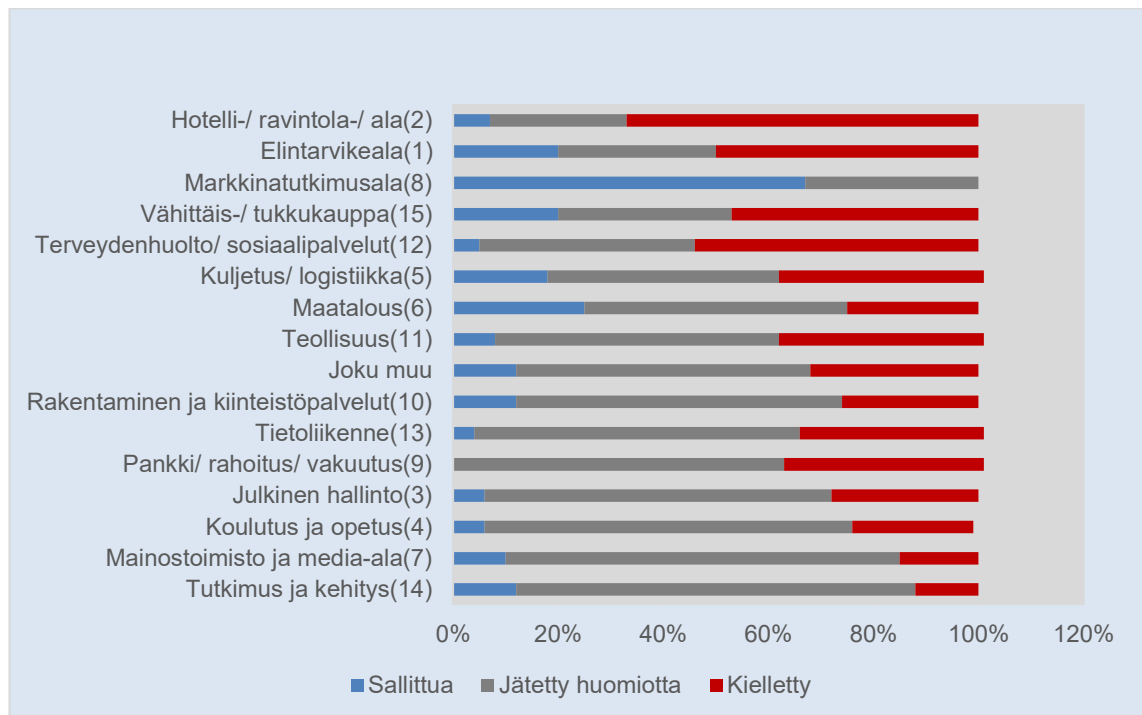
Kuva 1: Työnantajien suhtautuminen työpaikalla pelaamiseen [2, s. 33].

Kuvassa 2 on nähtävissä, että pelaamiseen suhtaudutaan eri aloilla eri tavoilla: kielteisimminkin asiaan suhtaudutaan hotelli- ja ravintola-alalla, jossa 67 %:ssa paikoista töissä pelaaminen on kiellettyä, mutta sielläkin 7 %:ssa työpaikoista tämä on sallittua. Kielteisimminkin pelaamiseen suhtautuvat seuraavat alat:

- hotelli- ja ravintola-ala (67 %:ssa työpaikoista kiellettyä)
- terveydenhuolto- ja sosiaalipalvelut (54 %:ssa työpaikoista kiellettyä)
- elintarvikeala (50 %:ssa työpaikoista kiellettyä)

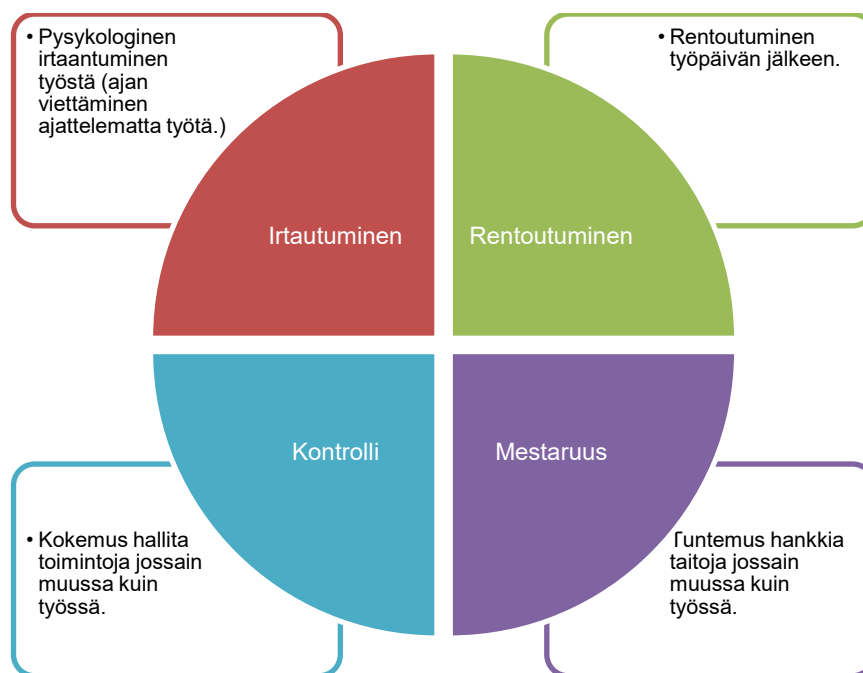
- vähittäis- ja tukkukauppa (47 %:ssa työpaikoista kiellettyä)

Markkinatutkimusala on ainoa, joka erottuu joukosta selkeästi ja on sallinut pelaamisen 67 %:ssa työpaikoista. Muilla aloilla pelaaminen töissä on jätetty huomioimatta, ja suhtautuminen pelaamiseen työpaikoilla on neutraalia tai kielteisen neutraalia. [2, s. 33.]



Kuva 2: Työnantajien suhtautuminen työpaikalla pelaamiseen toimialoittain [2, s. 33].

Tutkimuksissa on todettu, että työperäisestä stressistä palautumiseen tarvitaan neljää asiaa palautumisprosessissa (kuva 3). Palautumisen kannalta on korostettu erityisesti urheilun ja liikunnan kaltaisten aktiviteettien olevan erityisen hyödyllisiä. Nykyään työntekijät tekevät pidempiä päiviä ja raportoivat lisääntyneestä työstressistä. Elektroniset laitteet ovat integroituneet tehokkaasti jokaisen arkielämään, ja kun joukkueurheilun sisällyttäminen jo valmiiksi kiireisiin päiviin on haastavaa, on todettu, että pelaaminen voi vaikuttaa positiivisesti palautumiseen. Tutkimuksen mukaan myös digitaalisella pelaamisella näytti olevan kumulatiivinen myönteinen vaikutus palautumiseen. [3.]



Kuva 3. Työperäisestä stressistä palautuminen [3].

2.1 Kanaliigan historiaa

Kanaliiga Ry on perustettu syksyllä 2018, jolloin perustajajäsenet olivat havainneet, ettei yrityssarjoista löydy oma sarjaa PC- tai konsolipelaamiseen. Pelaaminen yrityssarjoissa aloitettiin vuonna 2018 pelillä PlayerUnknown's BATTLEGROUNDS, josta pelikategoria yrityssarjassa on lähtenyt laajentumaan mm. Dota 2- ja Counter-Strike: Global Offensive -peleihin. Kanaliiga Ry toimii vapaaehtoisvoimin, ja se kuuntelee yhteisön ja yhteistyökumppaneiden palautteita oman toiminnan kehittämisessä. Kanaliigalla on oma Discord-kanava, jota kautta kaikki viestintä pääasiallisesti kulkee pelaajille. [4.]

Kanaliiga Ry:llä on omat verkkosivut, joilla on yleistä tietoa tulevista ja menneistä tapahtumista. Lisäksi on wiki-sivusto, jossa on ohjeita, sääntöjä ja muuta tarpeellista tietoa pelaajille ja yrityksille, jotka haluavat osallistua peliin. Pääasiainen viestintä tulevien turnauksien osalta tapahtuu Discordin kautta, jossa tiedotetaan tulevista kausista, puhutaan ja sovitaan turnauksien pelejä, yms.

Kanaliigan yhdistyksen toimihenkilöt on esitelty kuvassa 4, josta näkee organisaation henkilöstön vuonna 2021.



Kuva 4: Kanaliiga yhdistyksen toimihenkilöt vuonna 2021 [4].

Kanaliigan elektronisen urheilun liiga tavoittaa yli 15 000 pelaajaa kuukausittain, ja yleisin asiakassegmentti ja yleisin (80 %) asiakas on 25–40-vuotias mies, joka

- työskentelee IT-sektorilla (60 %)
- on korkeasti koulutettu (75 %)

- ansaitsee yli 40 000 € vuodessa bruttona (65 %) [5].

Kanaliigan toiminta tuo näkyvyyttä erityisesti IT-alan osaajien parissa yrityksille, jotka osallistuvat sponsoreina tai joukkueina kisoihin. Jokainen osallistuja edustaa omaa yritystään, ja yksi ehdoista osallistumiselle on olla osallistuvan yrityksen palkkalistoilla, niin että pystyy tarvittaessa osoittamaan, että osallistuja (työntekijä) saa elantonsa kyseisen yrityksen kautta [6].

2.2 CS:GO-pelin historia

Counter-Strike: Global Offensive on verkossa pelattava ensimmäisen persoonan ammutamoninpeli, joka on osa Counter-Strike-pelisarjaa. Siihen kuuluu myös alun perin Half-Lifen mukana vuonna 1999 ilmaisena lisäosana julkaistu Counter-Strike (tunnetaan myös nimillä Counter-Strike 1.6 ja Half-Life Counter-Strike) [7].

Counter-Strike sarjan vaiheet:

- Ensimmäinen itsenäinen oma versionsa Counter-Strike-pelistä julkaistiin vuonna 2000 [7].
- Vuoden 2000 jälkeen julkaistiin vuonna 2004 Counter-Strike: Condition Zero (kehittäjä Valve Corporation ja julkaisija Sierra Entertainment) [8].
- Counter-Strike: Condition Zeron jälkeen julkaistiin vuonna 2004 Counter-Strike: Source (kehittäjä ja julkaisija Valve Corporation) [9].
- Counter-Strike: Sourcen jälkeen julkaistiin nykyinen ja uusin sarjan peli Counter-Strike: Global Offensive vuonna 2012 (kehittäjät Valve Corporation ja Hidden Path Entertainment ja julkaisija Valve Corporation) [10].

2.3 Valve Counter-Strike: Global Offensive Game State Integration -työkalu

Counter-Strike: Global Offensive Game State Integration on työkalu, jonka avulla tuotekehittäjien on mahdollista saada ulos pelin tietoja eri tavoilla käyttämällä HTTP POST endpointin kautta saatua JSON-sanomaa. Käyttämällä Counter-Strike: Global Offensive Game State Integration -työkalua on mahdollista saada pelistä ulos reaaliajassa mm. tietoa pelin senhetkisestä tulostilanteesta, pelaajien tietoja ja monia muita tietoja [11].

Esimerkkikoodissa 1 on konfigurointimalli, miten pelaajan omat tiedot voi poimia hyödyntäen Counter-Strike: Global Offensive Game State -integraatiota [11].

```
"Console Sample v.1"
```

```
{
  "uri" "http://127.0.0.1:3000" // palvelinn osoite ja portti
  "timeout" "5.0" // tauko x ajan kuluttua
  "buffer" "0.1"
  "throttle" "0.5"
  "heartbeat" "60.0"
  "auth"
  {
    "token" "CCWJu64ZV3JHDT8hZc" // token session autentikointiin
  }
  "output"
  {
    "precision_time" "3"
    "precision_position" "1"
    "precision_vector" "3"
  }
  "data"
  {
    "provider" "1" // yleinen info palvelinistä: game
    name, appid, client steamid, etc.
    "map" "1" // kartta, pelimoodi, ja pelin ti-
    lanne ('warmup', 'intermission', 'gameover', 'live') ja pisteet
    "round" "1" // kierroksen tilanne ('freezetime',
    'over', 'live'), pommin status ('planted', 'exploded', 'defused'), ja
    kierroksen voittaja (jos on)
    "player_id" "1" // pelaajan nimi, joukkueen tägi,
    tarkkailijan tiedot ja joukkue
    "player_state" "1" // pelaajan status tällä kierrok-
    sella, kuten elämäpisteet, panssarin pisteet ja tapot, jne.
    "player_weapons" "1" // pelaajalla käytössä olevat aseet
    "player_match_stats" "1" // Pelaajan tulokset tässä pelisse
    kuten tapot, avustukset, pisteet, kuolemat ja MPVs (Most Valuable
    Player = kierroksen arvokkain pelaaja)
  }
}
```

Esimerkkikoodi 1. Konfigurointimalli.

Tallentamalla `gamestate_integration_consolesample.cfg`-kooditiedoston oikeaan kansioon saa ulos pelaajan tiedot tämänhetkisestä pelistä, jota hän on pelaamassa valitulla palvelimella [11].

Kutsun tekemiseen löytyy useita valmiita pohjia, joita voisi hyödyntää integraation tekemiseen eri ohjelmointikielillä, kuten C#, Java, TypeScript, NodeJS/JavaScript, Python, Go jne. [11].

Cfg-tiedosto on konfigurointitiedosto, jolla voidaan Counter-Strike: Global Offensive Game State Integration -työkalulla valita, mitä tietoja pelin palvelimelta halutaan hakea. Tiedostoon syötetään ennalta asetettuja parametrejä, joilla valitaan mukaan halutut tiedot. Toki pitää huomioida, että jotkin parametrit ovat pakollisia, jotta integraatio toimii.

Counter-Strike: Global Offensive Game State Integration -työkaluun syötettyjen parametrien perusteella peli muodostaa halutut tiedot omaan tiedostoon, jota voidaan hyödyntää 3. osapuolen työkaluja käyttäen tai kuten insinööri-työssä tehdään, hyödyntäen itse rakennettua työkalua. Suoraan tällaisenaan sillä ei saa muodostettua haluttua verkkosivulle tulevaa reaaliaikaista tulospalvelua, mutta käyttäen tätä tietoa voidaan tulkita taustalla olevia kommentoja, jotta voidaan muodostaa verkkosivuilla näkyvä reaaliaikainen tulospalvelu.

2.4 LinuxGSM-palvelinratkaisu

LinuxGSM on avoimen lähdekoodin palvelinratkaisu, joka tukee 117:ää erilaista peliä. LinuxGSM on kirjoitettu Bash Script -kielellä, ja se käyttää MIT-lisenssiä. LinuxGSM toimii Linux-järjestelmissä. [12.]

Asennusohjeissa annetaan tarkat ohjeet, mitä tarvitaan, jotta pelipalvelinta voidaan käyttää Ubuntussa. Esimerkkikoodin 2 komento pitää syöttää komentoriville (englanniksi tutummin kutsuttu nimillä "terminal" ja "CLI"), jotta Ubuntu

pystyy pyörittämään 32-bittistä ohjelmistoa myös, ja siitä löytyvät kaikki tarvittavat muut ohjelmat [12].

```
sudo dpkg --add-architecture i386; sudo apt update; sudo apt install  
curl wget file tar bzip2 gzip unzip bsdmainutils python util-linux ca-  
certificates binutils bc jq tmux netcat lib32gcc1 lib32stdc++6  
libsdl2-2.0-0:i386 steamcmd
```

Esimerkkikoodi 2. Tarvittavien ajurien ja ohjelmien asennus.

Komentorivillä ajetaan läpi esimerkkikoodi 3, jolla päivitetään arkkitehtuuri [12].

```
sudo dpkg --add-architecture i386;
```

Esimerkkikoodi 3. Arkkitehtuurin päivitys.

Esimerkkikoodin 4 mukaisella komentorivillä järjestelmä päivitetään [12].

```
sudo apt update;
```

Esimerkkikoodi 4. Järjestelmän päivitys.

Esimerkkikoodin 5 komentorivillä asennetaan curl, wget, file, tar, bzip2, gzip, unzip, bsdmainutils, python, util-linux, ca-certificates, binutils, bc, jq, tmux, netcat, lib32gcc1, lib32stdc++6, libsdl2-2.0-0:i386 ja steamcmd. Kaikki nämä ovat omia kirjastojaan, joita tarvitaan, jotta csgoserver ja Steam CMD voidaan asentaa Ubuntuun.

```
sudo apt install curl wget file tar bzip2 gzip unzip bsdmainutils python  
util-linux ca-certificates binutils bc jq tmux netcat lib32gcc1  
lib32stdc++6 libsdl2-2.0-0:i386 steamcmd
```

Esimerkkikoodi 5. Asennetaan tarvittavat paketit.

Syntaksi on esimerkkikoodin 6 mukainen sudo apt install -komennolla.

```
sudo apt install <package1> <package2> ... <package(i)>
```

Esimerkkikoodi 6. Syntaksi sudo apt install -komennolle [14].

Kun käytetään file-komentoa, syntaksi on esimerkkikoodin 7 mukainen.

```
sudo apt install <package1> <package2> ... <package(i)> file <option1>
.. <option(i)> <file1> .. <file(i)>
```

Esimerkkikoodi 7. Syntaksi useamman asennuspaketin asentaminen [14].

Komentoriviin syötetään esimerkkikoodin 8 asennuskomento.

```
sudo apt install curl wget file tar bzip2 gzip unzip bsdmainutils py-
thon util-linux ca-certificates binutils bc jq tmux netcat lib32gcc1
lib32stdc++6 libsdl2-2.0-0:i386 steamcmd
```

Esimerkkikoodi 8. Asennuskomento tarvittaville ohjelmille.

Ensin asennetaan curl- ja wget-kirjastot. Tämän jälkeen asennetaan Steam CMD, jonka tiedostoina hyväksytään useampi eri tiedoston formaatti.

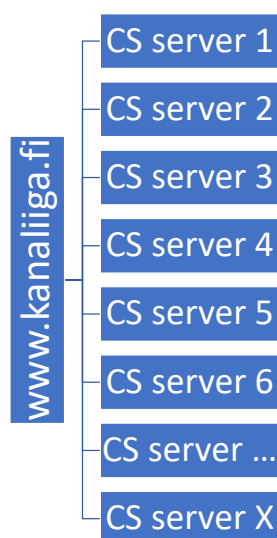
curl	Komentorivityökalu, jolla voidaan siirtää dataa useita verkko-protokollia käyttäen [15].
wget	Komentorivityökalu, jolla voidaan noutaa tietoja verkkopalvelimilta [16].

LinuxGSM:n asennukseen löytyy hyvät sivustot, mutta osa ohjeista on vanhentunutta tai liian yleispätevää (eri peleillä toimii hieman eri tavalla), mikä aiheutti omat haasteensa asennuksessa. Aluksi asennus tehtiin kahdella erillisellä tietokoneella, joissa oli molemmissa Ubuntu asennettuna, jotta nähtiin, että toiminnallisuus toimi myös eri koneissa ja saatiin kerättyä dataa, josta näki, miten tiedostoja voisi myöhemmin yhdistää yhdeksi verkkosivuksi. Myöhemmin nämä

kaksi konetta korvattiin palvelimella, johon asennettiin virtuaaliympäristö, jolla pystyi jakamaan yhden palvelimen useammaksi palvelimeksi.

2.5 Kanaliigan integraatiotarve

Kanaliigan yritysliigassa pelataan useita eri pelejä päivittäin turnauksien aikana, ja kaikkia ei ole aina mahdollista live-lähetyksenä striimata Kanaliigan Twitch-kanavien kautta, minkä vuoksi on tarvetta tuottaa Kanaliigan sivuille reaaliaikaisia ottelutilastoja sitä mukaa, kuin pelejä pelataan. Tämä tulee tarkoittamaan useamman eri palvelimen sisäisiä integraatioita (kuva 8), joista muodostetaan tiedostoja, joista voidaan muodostaa verkkosivulle reaaliaikaista pelidataa eri palvelimilta yhdistämällä tämä tiet. Sovittiin, että projektin aikana tästä muodostuu html-sivusto, ja mahdolliset jatkokehitykset toteutetaan omina projekteinaan.



Kanaliigalla on useita peli servereitä, joista on tavoite muodostaa useista eri servereistä kootusti dataa, jota voidaan käyttää live tuloksien visuaalisen esittämiseen kanaliigan kotisivuilla.

Kuva 8. Kanaliigan palvelinrakenne.

3 Reaaliaikaisen pelitilanteen integraatio verkkosivuille

3.1 Integraatiolla haettavien tietojen tavoite

Pelin dataan tehdään integraatio, josta voidaan muodostaa verkkosivu tai verkkosivulle sopivaa datatiedostoa, josta pelin etenemistä voidaan seurata verkkosivulla reaaliaikaisesti. Haasteita datan keräämiseen tulee, kun on monia palvelimia, joista pelejä voidaan pelata ja näin ollen data voi olla uutta tai vanhaa koko ajan palvelimella.

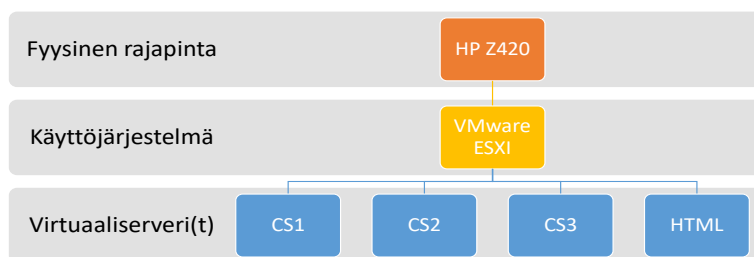
Tiedosto, josta pelin dataa noudetaan, muodostaa pelitietoja koko pelattavan pelin ajan. Pelin data muodostuu Valven itse kehittämään tekstitiedostoon, joka hieman mukailee muotoilultaan json-tiedostoja, mutta lopulta ei ole puhdasta sellaista, jotta siitä voisi muodostaa suoraan haluttua tietoa.

Tiedoston muodostamiseen käytetään palvelimella luotavaa Bash-skriptiä, joka muodostaa tuloksista HTML-sivustolle sopivan elementin, joka voidaan implementoida suoraan palvelimilta yhteen sivustoon omana integraationaan.

3.2 Integraation eri toteutusvaihtoehdot

Integraation toteutuksena käytetään Linux GSM:ää toteuttamaan pelipalvelinpaketti, koska myös tilaaja käyttää tätä samaa pakettia omilla palvelimillaan. Koska suoraa pääsyä ei ole tarjolla palvelimilta etänä, toteutuksen kannalta hankittiin käytetty HP Z420 -palvelin, johon asennettiin pohjalle VMwaren ESXI-ohjelmisto, jonka sisään tehdään useita virtuaalipalvelimia, joista voidaan muodostaa testidataa. Itse toteutuksen tuotantoon viennin hoitavat Kanaliigan omat IT-asiantuntijat tämän projektin jälkeen. Toteutuksessa kuitenkin testiympäristön rakentamiseksi pyritään mukailemaan mahdollisimman paljon tilaajan olemassa olevia ympäristöjä, jotta toteutus on mahdollisimman helppoa viedä myös tuotantoon Kanaliigan omille palvelimille.

Palvelininfrastruktuuri koostuu fyysisesti palvelimesta, käyttöjärjestelmästä ja virtuaalipalvelimista (kuva 9).



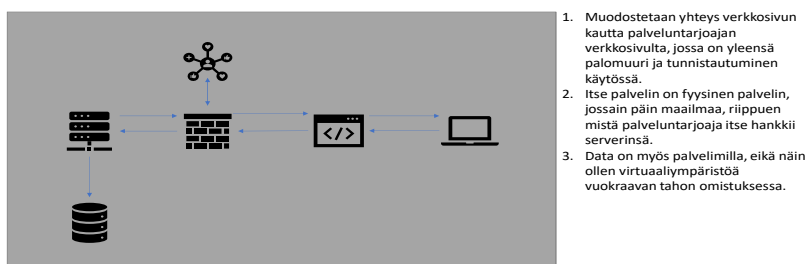
Kuva 9. Testausympäristön infrastruktuuri.

Ympäristön fyysinen rajapinta on HP Z420 -palvelin, jonka laitteisto koostuu seuraavista komponenteista:

Valmistaja:	Hewlett-Packard
Malli:	HP Z420 Workstation
CPU:	8 ydin, Intel Xeon E5-2665 2,40GHz
Muisti:	8 GB
Tallennuskapasiteetti:	4 x 1 Tb HDD

Toteutukseen olisi voinut myös harkita käytettävien vuokrattavien virtuaaliympäristöjen, mutta tässä tapauksessa käytetyn palvelimen hinta oli sen verran edullinen, ettei perusteluja ollut lähteä vuokraamaan kalliimmalla virtuaalipalvelinta. Lisäksi oman palvelimen käyttäminen tarkoittaa myös, että omistaa täysin sen, mitä palvelimella tehdään, eikä se ole 3. osapuolien sopimusteknisten tai teknisten rajoitusten vuoksi jonkun toisen omistamaa dataa.

Kuvassa 10 kuvataan yhteyden avaaminen 3. osapuolen virtuaalipalvelimiin yleisellä tasolla.



1. Muodostetaan yhteys verkkosivun kautta palveluntarjoajan verkkosivulta, jossa on yleensä palomuuuri ja tunnistautuminen käytössä.
2. Itse palvelin on fyysinen palvelin, jossain päin maailmaa, riippuen mistä palveluntarjoaja itse hankkii serverinsä.
3. Data on myös palvelimilla, eikä näin ollen virtuaaliympäristöä vuokraavan tahon omistuksessa.

Kuva 10. Kuvaus 3. osapuolen virtuaalipalvelimen yhteyden muodostamisesta.

Kun muodostetaan yhteyttä VMwaren ESXI-järjestelmään, se voidaan toteuttaa etänä miltä tahansa koneelta, joka on samassa verkossa kuin palvelin on. Verkon ulospäin avaaminen ei ole osa tätä projektia, joten tietoturvasyistä myös tämä pidetään näin ja esittely tehdään etänä johtuen myös osittain vallitsevasta koronatilanteesta.

Virtuaalipalvelimille asennetaan Ubuntu-käyttöjärjestelmä, koska Linux GSM -asennuspaketit on suunniteltu tälle käyttöjärjestelmälle ja tässäkin pyritään olemaan mahdollisimman lähellä tilaajan käyttämää infraa, jotta asennus tilaajan palvelimille on mahdollisimman mutkatonta.

4 Reaaliaikaisen CS:GO-pelitalanteen integraatio verkkosivulle

4.1 Integraation tavoitteet

Insinööriyöprojektin tavoitteena oli rakentaa reaaliaikainen tulosten integrointi pelistä verkkosivulle, josta tuloksia voidaan seurata reaaliaikaisesti. Toteutuksessa käytettiin luvussa 3 esiteltyä tutkimusmateriaalia kehityksessä, jotta saatiin aikaiseksi haluttu lopputulos, joka on kehitettävissä lisää myös tulevaisuudessa.

Projektissa keskityttiin ensimmäisenä toteuttamaan toimiva yhteys pelin ja palvelimen välille, jotta dataa saatiin ulos pelistä palvelimelle. Kun yhteys oli muodostettu, voitiin alkaa rakentaa datan ympärille integraatiota palvelimella, jossa testausta voitiin tehdä pelin pyöriessä taustalla tietokoneella. Jotta rakennusta voitiin tehdä, pelin piti olla käynnissä ja vähintään yhden pelaajan pelissä mukana ja loput pelaajat voitiin korvata boteilla, jotta dataa eri pelaajista saatiin esille.

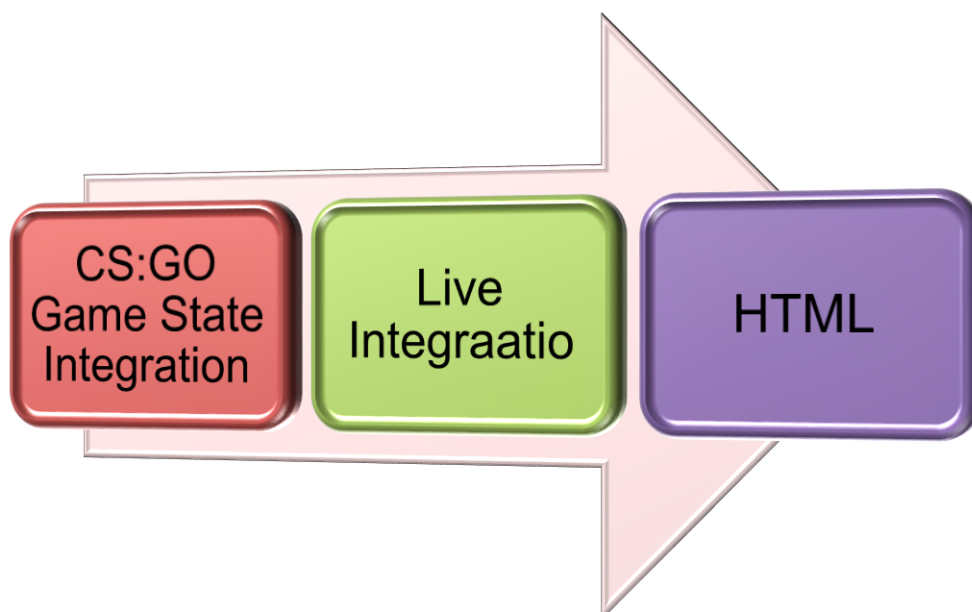
Pelissä itsessään on erilaisia pelimuotoja, mutta tämä toteutus koski kilpailullista 5v5 -ottelua. Myös asevelipelimuodossa tämä integraatio toimisi, mutta pelaajakohtaisen datan osalta toteutusta tulisi hieman muuttaa. Pelaajakohtainen data ei kuulunut osaksi tämän projektin toteutusta.

Kun dataa saatiin ulos, lähdettiin projektissa toteuttamaan verkkosivua datasta. Verkkosivulle data muodostui reaaliaikaisesti, joten tavoite saavutettiin projektin aikana ja reaaliaikainen datan integrointi verkkosivuksi oli toteutettu.

4.2 Integraation toteutus

Reaaliaikaisen integraation avulla tuodaan tulokset palvelimella pelattavasta pelistä suoraan verkkosivuksi visuaalisesti näkyviin html -sivuna. Ennen reaaliaikaista integraatiota tuloksia on voinut seurata reaaliaikaisesti vain suorista lähetyksistä, mutta jokaiseen peliin ei riitä selostajaa tai striimaajia, jotta jokaisen joukkueen pelit saataisiin reaaliaikaisesti ulos joka pelipäivä. Kun pelin käynnistää, alkaa kertyä dataa Valven Game State -integraation tiedostoon, joka aina ylikirjoitetaan, kun seuraava peli alkaa palvelimella ja reaaliaikainen integraatio pitää myös huolen, ettei tällaista tietojen katoamista tapahdu palvelimella.

Kuvassa 11 kuvataan, miten reaaliaikainen-pelitalteen integraatio tapahtuu ylätasolla, jossa ensin CS:GO Game State Integration -työkalu muodostaa pelistä dataa reaaliaikaisesti palvelimella, josta data poimitaan reaaliaikaisella integraatiolla talteen ja siitä muodostetaan verkkosivu.



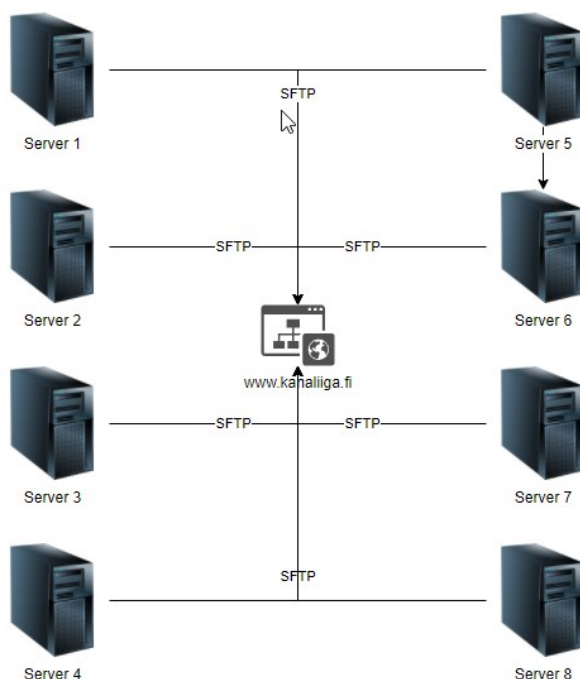
Kuva 11. Reaaliaikaisen pelitilanteen integraatio.

Kun palvelin käynnistetään, käynnistyskomentoon lisätään oma komento, joka käynnistää samalla myös reaaliaikaisen integraation, joka alkaa muodostaa pelistä HTML-sivulle sopivaa elementtiä, jonka voi käyttää sellaisenaan tai lisätä omana sivunaan näkyviin Kanaliigan verkkosivuille.

Reaaliaikaisen integraation avulla myös muiden yritysten pelaajien tai kannattajien on mahdollista seurata yritysten välistä kilpailua. Tällaiselle on kysyntää, ja integraatio tuo vastauksen tähän tarpeeseen.

4.3 Integraation tiedostojen siirto

Kuvassa 12 on kuvattu ylätasolla, miten tiedostojen siirtäminen tapahtuu SFTP-yhteyden avulla, jos lopulta muodostetaan dataa Kanaliigan verkkosivuille.



Kuva 12. Integraatiodokumenttien siirto.

Tiedostojen siirto toteutetaan SFTP-tiedostosiirtona palvelimilta koostetusti palvelimella, josta voidaan HTML-sivu ladata suoraan Kanaliigan sivulle. Tähän projektiin ei itsessään sisälly tiedostonsiirron rakentamista, vaan keskitytään enemmänkin toteuttamaan integraatio valmiiksi ja tarjotaan suunnitelma, jolla toteutusta voidaan jatkaa tiedostojensiirron osalta omana projektina.

Palvelimia voidaan lisätä käytännössä rajattomasti, ja itse tiedoston yhteen muodostaminen voidaan toteuttaa suoraan omalla palvelimella, johon tiedostot siirretään ja josta data siirretään verkkosivulle. Jos on teknisesti mahdollista, voidaan integraatio toteuttaa myös suoraan SFTP:lta verkkosivun kansioon, josta esim. JavaScriptillä voidaan muodostaa datasta visuaalisesti reaaliaikaisia tuloksia verkkosivulle.

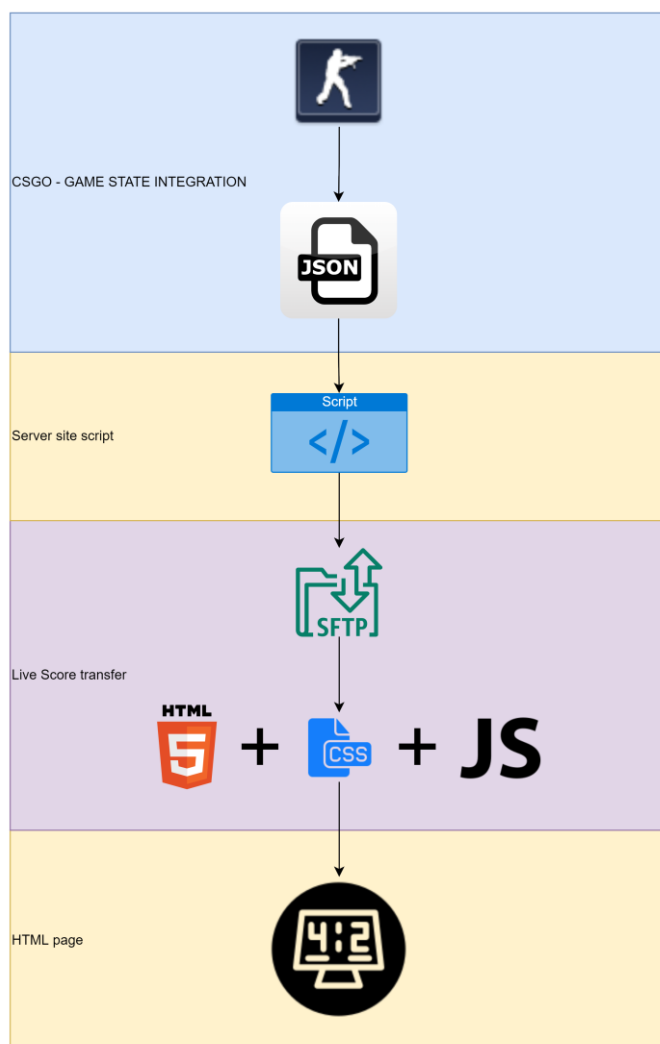
4.4 Integraation HTML-sivun koostaminen

HTML-sivusto koostetaan pelistä saatavasta datasta, josta kerätään halutut kohdat näkyviin ja tuotetaan niistä skriptillä haluttu data ulos reaaliaikaisesti.

Tällöin skripti juoksee koko pelin ajan taustalla automaattisesti halutulla ajastuksella, joka voi olla esimerkiksi 5 sekuntia tai vaikka pidempikin. Skripti itsessään ei kuormita palvelinta, koska käytettävissä olevia resursseja ei juurikaan käytetä palvelimelta.

Palvelimelta tuotetaan ulos HTML-sivusto tai -elementti, jota voi käyttää suoraan sellaisenaan verkkosivulla, josta pelaajien kannatusjoukot voivat seurata reaaliaikaisesti tuloksia verkkosivujen kautta.

Kuvassa 13 näytetään datan liikkuminen integraation eri tasoilla, joita on CS:GO Game State Integration -työkalu, joka kerää datan JSON-tiedoston kaltaiseksi ja josta palvelimella oleva skripti muodostaa HTML:ää, CSS:ää ja JS:ää hyödyntäen verkkosivuksi dataa reaaliaikaisesti.



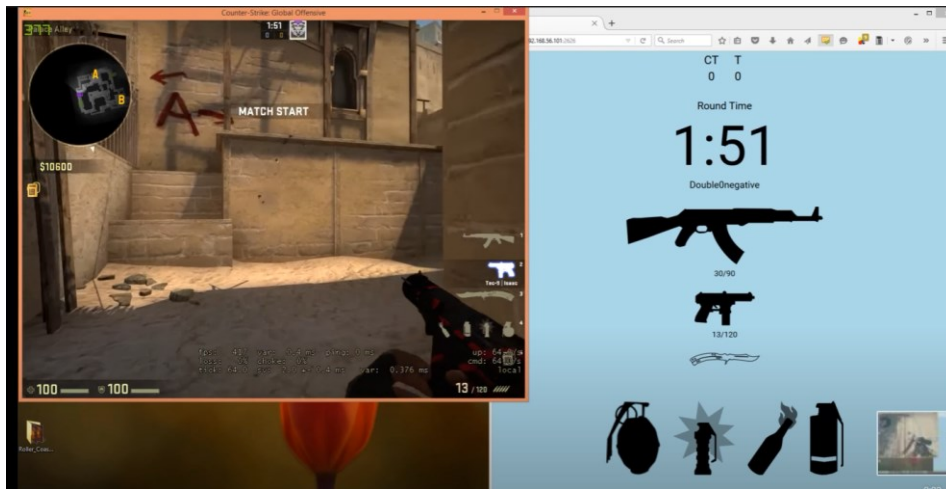
Kuva 13. Datan siirtyminen verkkosivulle.

5 CS:GO- reaaliaikaisen pelitilanteen integraation lopputulos

Kun lähdettiin tutkimaan toteutusta, alkuehdotus oli lähteä käyttämään ratkaisua, joka löytyy jo GitHubin kautta. Kun tätä toteutusta tutkittiin, tuloksena oli, että se vaatisi teknisen käyttäjän tai botin tai vastaavan käyttöä, jotta voisi käyttää ratkaisua, jossa pitää olla pelin sisällä, jotta reaaliaikaistulosta saadaan ulos pelistä [17].

Double Negativen ratkaisu vaikutti hyvältä pohjalta, mutta siinä tuli useampi ongelma vastaan, minkä vuoksi tähän pohjaan ei lopulta päädytty (kuva 14). Lähtökohtaisesti Double Negativen pohja on tarkoitettu näyttämään katsottavan pelaajan tietojen reaaliaikaistulosten tuomiseen, jolloin muut jäsenet jäävät

pimentoon tuloksista, mikä voi rajoittaa tulevaisuuden vaihtoehtoja lähteä kehittämään lisää reaaliaikaista integraatiota uusin ominaisuuksin.



Kuva 14. CS:GO GameState Integration HUD -esimerkki [18].

Lopputulena tämä ratkaisu oli liian raskas, jos halutaan saada aikaiseksi reaaliaikainen integraatio, josta voitaisiin saada ulos dataa datana ilman ylimääräisiä botteja, teknisiä käyttäjiä, tms. Myös asiakas oli samaa mieltä asiasta, joten tutkimuksia jatkettiin, jotta löydettäisiin juuri oikea integraatoratkaisu Kanaaliin tarpeisiin.

Kuvassa 15 näkyy, minkälaista dataa pelistä saadaan ulos CS:GO Game State Integration -työkalun avulla, jota mm. LHM Manager HUD myös käyttää.



Kuva 15. LHM Manager HUD -esimerkki [19].

Muitakin vaihtoehtoja tutkittiin, kuten Lexogrine HUD Managerin ratkaisua ja miten se oli rakennettu, jos siitä olisi voinut katsoa pohjaa työlle. Kuten nimestä jo pystyi päättelemään, tämä tuote lähestyi asiaa enemmän striimaajille kohdennettuna työkaluna, jossa saa reaaliaikaista tietoa ulos kyllä itse pelistä ja todella kattavastikin, mutta tulokseen päästäkseen tuli jälleen olla kirjautuneena peliin sisälle käyttäjänä, jota kautta integraatio linkittyy peliin [19].

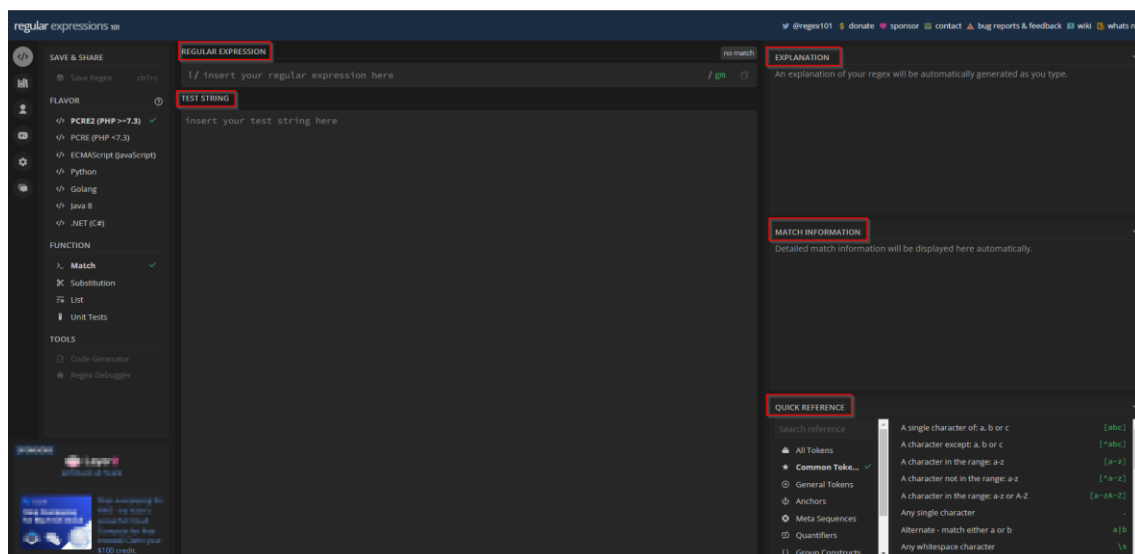
Valven dokumentaatiossa integraation osalta osa linkitetystä ratkaisusta oli joko vanhentuneita tai niitä ei ollut todennettu, joten syntyi epäily, ovatko ne toimivia ratkaisuja. Insinööriyössä tultiin tulokseen, että on parasta tutkia asiaa suoraan datalähteestä eli palvelimilta.

Kanaliigan kanssa käytiin keskustelua, voisiko insinööriyössä saada kehitystä varten pääsyä sen palvelimille, mutta tämä ei ollut mahdollista, joten työssä päädyttiin rakentamaan oma palvelin, jossa voisi toteuttaa demoympäristön. Kanaliigalta saatiin tietoa, mitä se on käyttänyt pohjana CS:GO-pelipalvelimen tekemiseen, ja insinööriyössä toteutettiin vastaava käyttäen Ubuntuja ja asentamalla sinne sisään LinuxGSM-pelipalvelin. Itse palvelimen pystyttäminen 0-budjetilla ei ollut haastavaa, koska käytössä oli ylimääräinen tietokone, jonka pystyi

valjastamaan tähän käyttöön. Haastava osuus tuli verkkoyhteyksien avaamisen kanssa ja porttien avauksen kanssa.

Itse palvelimen pystytys ja toimintakuntoon saaminen vei aikaa n. 3 kuukautta, jossa ajassa ehdittiin saada myös verkkoyhteydet toimimaan palvelimella ja ajettua useita testejä. Testien lopputuloksena tuli huomattua, että paras tapa lähestyä tätä integraatiota on kirjoittaa oma skripti, jota ajetaan mahdollisimman tiuhaan tahtiin, jotta tieto päivittyy HTML-sivulle reaaliaikaisesti. Työ aloitettiin tutkimalla ensin Valven tuottamaa dataa, joka on JSON-tiedostojen kaltaista, mutta ei tyylipuhdasta sellaista, jotta sitä voisi ajaa suoraan JSON-tiedostona ja siitä muodostaa verkkosivua. Tästä tuli uusi haaste, jossa lopputuloksena pääteltiin, että paras tapa saada haluttua dataa ulos on käyttää BashScriptiä, jossa haetaan oikeat kohdat käyttäen Regular Expressioineita, joilla osoitetaan skriptillä, mistä kohtaa tiedostoa halutaan hakea tietoa. Regular Expressionit ovat malleja, joita käytetään vastaamaan merkkijonojen merkkiyhdistelmiä [20].

Regular Expressioneihin löytyy useita hyviä sivustoja, joissa voi testata omia malleja, joilla pyritään paikallistamaan ja poimimaan haluttu tieto datasta. Mm. regex101.com-sivustolla voi suoraan kirjoittaa oman Regular Expressionin ja lisätä sisään halutun testimerkkijonon, josta voi sitten reaaliaikaisesti kokeilla eri yhdistelmiä, jotta haku kohdistuu vain haluttuihin osiin merkkijonossa (kuva 16). Regex101 tarjoaa myös pikakäyttöoppaan eli selityksen, mitä malli tekee ja mitä hakuun osuneiden kohtia halutaan käyttää [21].



Kuva 16. Regex101-sivusto testaamiseen [21].

Testiympäristöön tämän toteutuksen rakentaminen oli suhteellisen nopeaa, ja siinä pystyttiin katsomaan reaaliaikaisesti, miten data liikkuu, saadaanko poimitua haluttua dataa ja pystyttiin ajamaan muutoksia nopeasti skriptiin sisään. HTML-tiedoston tekeminen ja tyyli-tiedosto CSS myös muotoutui ajan myötä lopulliseen esittelymuotoonsa testauksien lomassa. Toteutus itsessään on nyt kehitettävissä myös eteenpäin ja mahdollistaa kaiken saatavilla olevan tiedon noutamisen reaaliaikaisesti suoraan pelipalvelimelta ilman turhia teknisiä käyttäjiä, botteja, tms. välissä.

5.1 Analyysi reaaliaikaisen pelitilanteen integraatiosta

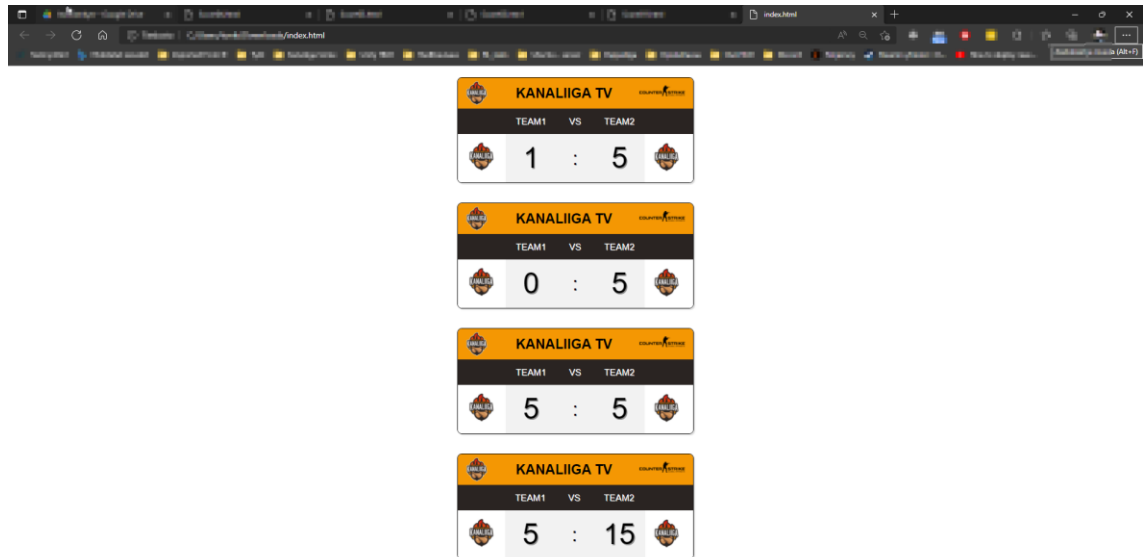
Reaaliaikaiseen integraatioon Kanaliiga halusi mukaan tuloksien lisäksi tiimien nimet ja tiimien logot näkyviin (kuva 17). Tämä oli mahdollista toteuttaa, mutta ilman pääsyä Kanaliigan palvelimiin käytettiin testijoukkueita Team1 ja Team2. Itse integraation toiminnallisuutta voitiin seurata kahdesta eri paikasta, pelipalvelimelta ja tietokoneelta (jossa itse peliä pelattiin.) Testaaminen testissä testidatalla oli validi valinta, kunhan käytettiin itse peliä, josta data syntyy saataville myös Kanaliigan palvelimille ja samoilla ohjelmilla. Näin päästiin hyvin lähelle todellista tilannetta, joka olisi kilpailun tilanteissa, kun pelataan Kanaliigan pelejä. Pelaajien sijaan käytettiin botteja, joista tulee samalla tavalla dataa ulos (pois lukien steamid, jolla voisi esim. noutaa pelaajan käyttämän kuvan osaksi

reaaliaikaisen integraatiototeutusta.) Vaikka toteutus tuli tehtyä hieman soveltaen ja tuotannon ympäristöä jäljentäen mahdollisimman tarkasti, lopputuloksena saatiin luotua testiympäristö, jonka tuloksia voidaan pitää luotettavina itse integraation toteutuksen kannalta.



Kuva 17. Reaaliaikaisen integraation pelituloksia.

Yhden palvelimen tekeminen kotikutoisin menetelmin ei ole ongelma, mutta useamman palvelimen tekeminen oli hieman haastavampaa varsinkin, kun pelien alkaminen palvelimilla tarkoittaa, että siellä pitää olla sisällä vähintään yksi ihmiskäyttäjä, jotta peli voi alkaa. Tässä kohtaa oli järkevintä kehityksen kannalta tehdä nyt reaaliaikainen integraatio toimimaan yhdellä palvelimella ja rakentaa valmiiksi myös pohja useamman palvelimen integraatioiden yhdistämiselle, kuten kuvassa 18 näkyy.



Kuva 18. Pelitulosten reaaliaikainen integraatio.

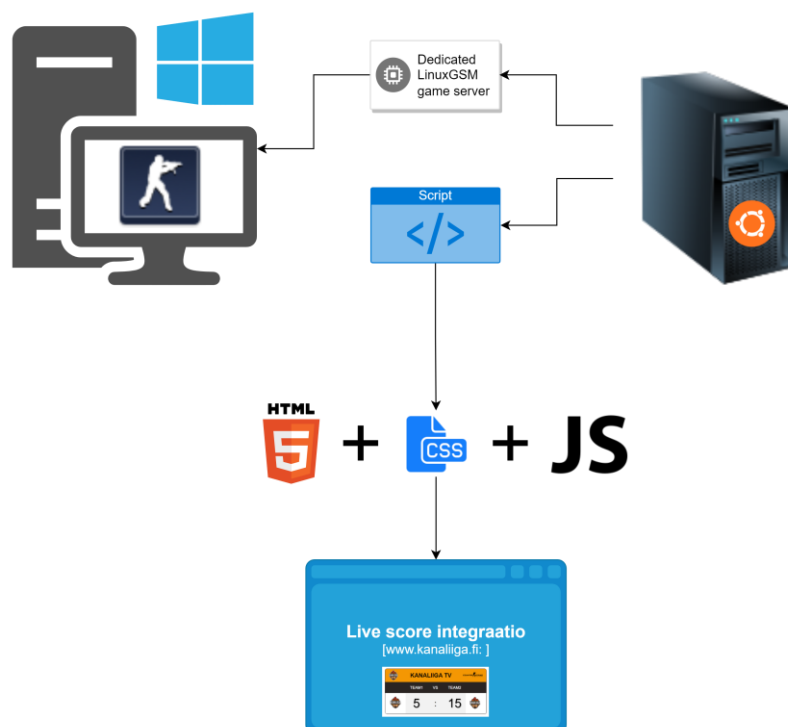
Tuloksena saatiin toimiva reaaliaikainen integraatio, joka ei sellaisenaan ole vielä valmis julkaistavaksi käyttöön, mutta luo kattavan ja vahvan pohja, jonka päälle voi lähteä jatkokehittämään ja rakentamaan Kanaliigan integraatiot omille verkkosivuille. Ratkaisussa on huomioitu myös tulevaisuuden kehityksen tarpeet, mikä takaa vapaan ja suoran pääsyn pelin dataan suoraan palvelimelle.

5.2 Integraation reaaliaikaisen pelitilanteen testaus

Reaaliaikaisen pelitilanteen integraation testaus toteutettiin käyttämällä yhtä palvelinta ja tietokonetta. Palvelimen käyttöjärjestelmä oli Ubuntu, jonka sisällä pyöri LinuxGSM-pelipalvelin. Tietokoneessa oli Windows 10 -järjestelmä, jossa oli Steam asennettuna, josta saa ladattua Counter Strike Global Offensive -pelin testaamista varten. Pelissä itsessään käynnistettiin peli botteja vastaan, ja käyttäjä siirtyy tarkkailijan rooliin.

Kun peli oli käynnissä, voitiin nähdä reaaliaikaisesti verkkosivun pelituloksen päivittyminen verkkosivuna, joka taas oli palvelimen sisällä, joten tulosten seuranta tapahtui palvelimelta, josta myös hallittiin peliä ja sen asetuksia. Peliin liit-
tykseen tietokoneen tuli olla osana paikallisverkkoa. Jotta yhteys saatiin auki,

tuli myös palomuuuri avata sisäisesti, jotta yhteys voitiin muodostaa palvelimeen (kuva 19).



Kuva 19. Reaaliaikaisen integraation testaus.

Testausta hieman hidasti verkkoyhteyksien avaaminen, joka ei ollut ihan yhtä sujuvaa kuin ajatuksen tasolla voisi kuvitella, mutta useamman yrityksen ja erehdyksen jälkeen yhteys alkoi toimia sisäisesti. Ulospäin myös yhteyden sai avattua, mutta se ei ollut tarpeellista integraation testauksen kannalta. Lähtöoletuksena on, että asiakas hallinnoi itse omat palvelimen verkkoyhteydet, palomuurit ja porttien avaukset, eivätkä ne kuuluneet osaksi tätä projektia.

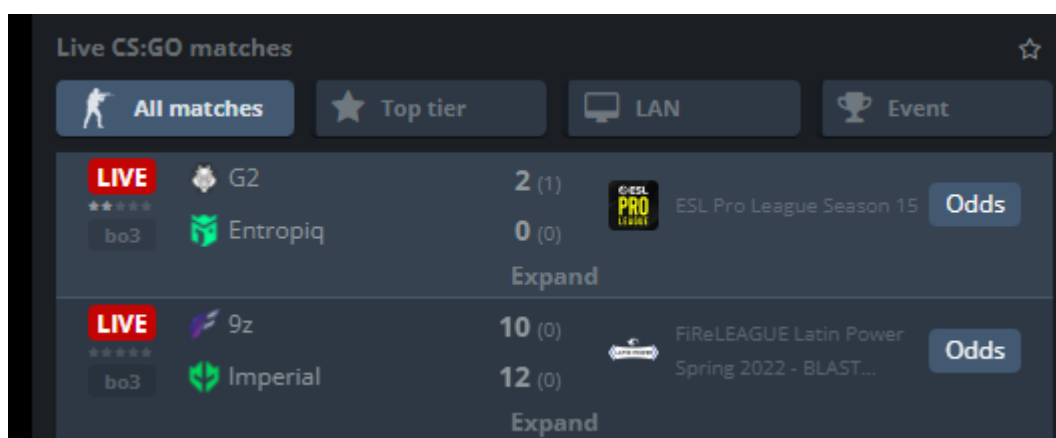
Kun peli käynnistyy, dataa alkaa muodostua uuteen tiedostoon, jolloin skripti alkaa kirjoittaa uuteen kansioon uudella nimellä HTML-sivua ja kerätä haluttua dataa pelistä. Kansion nimeämisen yksilöllisyyden takaamiseksi lisättiin aika-leima osaksi kansion nimeä, jolloin vältettiin tahattomat ylikirjoittautumiset. Niin kauan kuin peli pyöri, myös skripti ajoi dataa osaksi verkkosivua, jossa sivuston muotoilussa käytettiin CSS-tyylitiedostoa. Itse sivuston päivittämisen hoiti

JavaScript, jonka tehtävä oli päivittää sivua, tasaisin välein, jotta sivuston tulokset pysyivät reaaliaikaisena (näin käyttäjän ei tarvinnut itse päivittää sivua manuaalisesti).

5.3 Reaaliaikaisen pelitilanteen integraation seuraavat kehitysvaiheet

Kanaliigan puolesta tuli ehdotus, että seuraavaksi voitaisiin data alkaa muodostaa suoraan JSON-tiedostoon, josta verkkosivulle voidaan siirtää tieto JSON-tiedostosta JavaScriptiä hyödyntäen. Tämän lisäksi todettiin, että siinä tapauksessa voitaisiin muuntaa pelipalvelimen sisäinen JSON:n kaltainen tiedosto yhteensopivaksi JSON-tiedostoksi, josta voidaan lukea tietoja JavaScriptillä jatkossa. Tämä vähentäisi HTML-tiedostoon kirjoittamista, mutta mahdolliseen reaaliaikaiseen tuloksien seurantaan tämä tekninen toteutusmalli voisi aiheuttaa lisäviiveitä, koska tiedoston siirtona olisi tällä hetkellä ajatuksena käyttää SFTP-tiedostonsiirtopalvelimia, jolloin reaaliaikaisuus olisi osaksi riippuvainen siitä, kuinka usein tiedostoa päivitetään SFTP:lle tai SFTP:lta.

Kuvassa 20 näkyy, miten HLTV:n sivustolla näytetään reaaliaikaisena integraationa pelien tuloksia verkkosivulla.



Kuva 20. HLTV:n reaaliaikaiset pelitulokset [22].

Kuvassa 21 nähdään pelaajakohtaista dataa, josta voidaan poimia reaaliaikaisesti integraation avulla verkkosivulle pelikohtaisesti.

coreboard Normal Advanced (BETA)

R: 3 - dust2 1:2 1:47

Entropiq			\$	K	A	D	ADR
Krad	100	\$350	4	0	2	108.7	
El1an	100	\$0	1	1	2	78.3	
NickelBack	0	\$2600	1	1	3	58.0	
Lack1	0	\$100	1	0	3	34.0	
Forester	0	\$750	0	1	3	37.7	

☠ ☠

G2			\$	K	A	D	ADR
m0NESY	0	\$550	5	1	1	182.3	
NiKo	0	\$200	4	0	1	142.7	
huNter-	0	\$150	3	0	2	75.0	
Aleksib	0	\$100	1	0	1	33.3	
JACKZ	0	\$150	0	0	2	0.0	

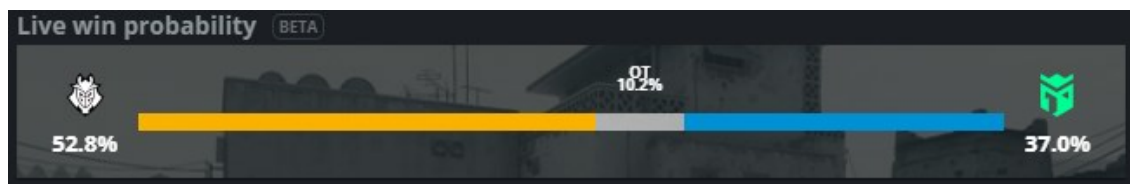
Kuva 21. HLTV:n reaaliaikainen pistetaulukko [22].

Kuvassa 22 näkyy pelaajien sijainti kartalla reaaliaikaisen integraation avulla, ja lisäksi on nähtävissä toimintalogi, josta näkee, mitä pelissä tapahtuu reaaliaikaisesti.



Kuva 22. HLTV:n reaaliaikainen kartta ja tulossyöte [22].

Kuvassa 23 näkyy päivittyä voittomahdellisuustieto visuaalisesti ja numeroina, ja se voi olla myös mahdollista tehdä reaaliaikaisen integraation avulla.



Kuva 23. HLTV:n pelin voittomahdellisuus [22].

Ideana nousi myös esille, että jos halutaan jatkojalostaa ajatusta niin, että tulokset tulisivat kapeammin esille, ja jos halutaan tarkempaa tietoa, peliä klikkaamalla pääsisi toiselle sivulle, jossa on mahdollista nähdä tarkemmin reaaliaikaisia tuloksia, kuten pelaajakohtaista tietoa, pommin purkua tai laittoa pelaajien toimesta, tiimien taloustilannetta, jne.

Malliesimerkkeinä nousivat esille HLTV:n reaaliaikaiset tulokset, joissa oli ensimmäisenä näkyvissä kuvan 20 mukaisesti lyhyesti reaaliaikaisia tuloksia useammasta pelistä, jotka ovat menossa. Kuvasta 21 alkaen taas näkyy, mitä kaikkea pystyy näkemään, kun klikkaa auki pelin, jota haluaa seurata. Pisteet

näkyvät yhdestä osiosta kuten pelissäkin, jossa on pelaajakohtaisesti suoritus-tietoja ja pisteitä. Karttakin päivittyi reaaliaikaisesti, jolloin voi katsoa suoraan, missä pelaajat ovat kuolleet kartalla ja lisäksi todennäköisyyden, kummalla on paremmat mahdollisuudet voittaa nykyisten tietojen perusteella [22].

Jatkon kehityksen kannalta työssä nähtiin kriittisenä, että palvelin ja pelialusta tulevat kehityksen ajaksi palveluntilaajalta, jotta aikaa säästyy itse varsinaiseen kehitystyöhön ja voidaan käyttää jo olemassa olevia infrastruktuureja ja toimivia verkko-ohjauksia, jotta ei menetä arvokasta kehityksen aikaa jo rakennettujen ratkaisujen uudelleen rakentamiseen.

6 Yhteenveto

Insinööriyöprojektissa keskityttiin ensin toteuttamaan testiympäristö, josta yhteydet tuli saada toimimaan, jotta voitiin alkaa kehittää toteutusta integraatiolle. Palvelimen pystytys tuli hieman ylimääräisenä työnä itse integraatioon nähden, mikä lisäsi hieman työtä itse projektissa ja vähensi myös käytettävää aikaa lopullisesta toteutuksesta. Palvelimen ja verkkoyhteyksien testaaminen vei projektityöstä huomattavan paljon aikaa, joten jatkokehityksen kannalta pidetään tärkeänä päästä rakentamaan jatkossa valmiissa palvelinympäristössä.

Projektissa toteutettiin reaaliaikainen integraatio palvelimen datasta, josta muodostettiin verkkosivuille reaaliaikaisesti tulostietoa pelistä, ja testaamisessa ei havaittu puutteita integraation toimivuuden osalta. Integraatio esiteltiin Kanaliiga Ry:lle, josta saadun palautteen perusteella todettiin, että tuotetta tulee jatkokehittää, jotta se saadaan osaksi Kanaliigan verkkosivujen tulospalveluja.

Projektissa saavutettiin haluttu tavoite, jossa luotiin reaaliaikainen integraatio, josta saadaan muodostettua dataa ulos. Integraatiota voidaan jatkojalostaa enemmän vastaamaan Kanaliigan tulevaisuuden tarpeita. Projektissa opittiin useita uusia asioita, kuten riittävien oikeuksien ja pääsyjen sopiminen ennen projektin aloitusta, jotta voidaan tehdä enemmän kehitystä ja vähemmän

palvelimen ja verkko-ohjauksien rakentamista. Projektista syntyy lisää työtä, joka jää Kanaliigan järjestettäväksi.

Projektista tullut tuote on valmis jatkokehitykseen ja sellaisenaan jo pystyy tuottamaan haluttua dataa verkkosivuna. Jotta tätä ratkaisua voi käyttää useamman palvelimen ja yhden verkkosivun kanssa, tulee miettiä yhteysrajapinnat palvelimen ja verkkosivun kannalta. Se voi olla suora päivitys palvelinkohtaisesti verkkosivulle, mutta tämä voi aiheuttaa päällekkäisyyksiä ja tietojen häviämistä tahattoman ylikirjoittamisen vuoksi, joten datan kerääminen yhteen paikkaan ensin voi olla hyvä lähtökohta, josta jatkaa tätä projektia.

Lähteet

- 1 Kanaliiga statistics. Verkkoaineisto. Kanaliiga Ry. <<https://stats.kanaliiga.fi/>>. Luettu 27.3.2022.
- 2 DIGITAALINEN PELAAMINEN TYÖHYVINVOINNIN EDISTÄMISESSÄ. Verkkoaineisto. Turun Yliopisto. <<https://www.utupub.fi/handle/10024/116164>>. Luettu 27.3.2022.
- 3 Digital Games and Mindfulness Apps: Comparison of Effects on Post Work Recovery. Verkkoaineisto. JMIR Publications. <<https://pubmed.ncbi.nlm.nih.gov/31322125/>>Luettu 27.3.2022.
- 4 Kanaliigan esittely. Verkkoaineisto. Kanaliiga Ry. <<https://kanaliiga.fi/info/>>. Luettu 3.4.2021.
- 5 Mikkola, Erkki. 2021. Perustajajäsen, Kanaliiga Ry. Espoo. Keskustelu 4.4.2021.
- 6 Kanaliiga Wiki – Kaikki eivät halua olla ilman dokumentaatiota. Verkkoaineisto. Kanaliiga Ry. <<https://wiki.kanaliiga.fi/>>. Luettu 3.4.2021.
- 7 Counter-Strike: Success Unlimited. Verkkoaineisto. Gameguru. <<https://www.gameguru.in/pc/2010/10/counter-strike-success-unlimited/>> Luettu 3.4.2021.
- 8 Counter-Strike: Condition Zero. Verkkoaineisto. Metacritic. <<https://www.metacritic.com/game/pc/counter-strike-condition-zero>>. Luettu 3.4.2021.
- 9 Counter-Strike: Source. Verkkoaineisto. Steam <https://store.steampowered.com/app/240/CounterStrike_Source/>. Luettu 3.4.2021.
- 10 Counter-Strike: Global Offensive. Verkkoaineisto. Steam. <https://store.steampowered.com/app/730/CounterStrike_Global_Offensive/>. Luettu 3.4.2021.
- 11 Counter-Strike: Global Offensive Game State Integration. Verkkoaineisto. Valve Software. <https://developer.valvesoftware.com/wiki/CounterStrike:_Global_Offensive_Game_State_Integration>. Luettu 3.4.2021.
- 12 Home. Verkkoaineisto. LinuxGSM. <<https://linuxgsm.com>>. Luettu 3.4.2021.
- 13 Deploy Counter-Strike: Global Offensive Game Servers. Verkkoaineisto. LinuxGSM. <<https://linuxgsm.com/lgsm/csgoserver/>>. Luettu 4.4.2021.
- 14 Using apt Commands in Linux [Complete Guide]. Verkkoaineisto. It's Foss. <<https://itsfoss.com/apt-command-guide/>>. Luettu 4.4.2021.

- 15 cURL. Verkkoaineisto. Curl. <<https://curl.se/docs/history.html>>. Luettu 4.4.2021.
- 16 Wget. Verkkoaineisto. Wikipedia. <<https://lists.gnu.org/archive/html/info-gnu/2021-09/msg00011.html>>. Luettu 4.4.2021.
- 17 Double0negative/CSGO-HUD 2015. Verkkoaineisto. Double0negative. <<https://github.com/Double0negative/CSGO-HUD>> Luettu 12.4.2021.
- 18 CSGO GameState Integration HUD example. Verkkoaineisto. YouTube. <https://www.youtube.com/watch?v=FM1-iapbEtc&ab_channel=Double-Negative> Katsottu 12.4.2021.
- 19 Home. Verkkoaineisto. Lexogrine LHM. <<https://lhm.gg/>> Luettu 12.4.2021.
- 20 Regular Expressions. Verkkoaineisto. Developer Mozilla. <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions> Luettu 12.3.2022.
- 21 Home. Verkkoaineisto. Regural expressions 101. <<https://regex101.com/>> Luettu 12.3.2022.
- 22 Matches. Verkkoaineisto. HLTV. <<https://www.hltv.org/matches/2354537/g2-vs-entropiq-esl-pro-league-season-15>> Luettu 12.3.2022.