



KARI MARTTILA

# **Unityllä toteutetun VR-pelin etäohjauksen suunnittelu ja toteutus**

TIETOJENKÄSITTELYN TUTKINTO-OHJELMA  
2022

## TIIVISTELMÄ

Marttila, Kari: Unityllä toteutetun VR-pelin etäohjauksen suunnittelu ja toteutus  
Opinnäytetyö, AMK  
Tutkinto-ohjelma: Tietojenkäsittely  
Helmikuu 2022  
Sivumäärä: 63

Opinnäytetyön tavoitteena oli kehittää menetelmä VR-pelin aloitusparametrien säätämiseen VR-lasien ulkopuolisesta laitteesta.

Työn toimeksiantaja oli Satakunnan ammattikorkeakoulun (SAMK), Tampereen ammattikorkeakoulun (TAMK), Tampereen yliopiston (TAU) sekä Metropolia-ammattikorkeakoulun yhteinen hanke: Verkostoyhteistyöllä vauhtia tekoälypohjaisten virtuaaliteknologioiden saavutettavuuteen kuntoutuksessa.

Työssä suunniteltiin ja toteutettiin selainpohjainen etäohjainsovellus SAMK:ssa aikaisemmin kehitettyyn kuntoutukseen suunniteltuun Whack-a-fysio-peliin.

Työn taustalla oli kuntoutuksen ammattilaisten toive siitä, että pelin potilaan mukaan yksilöityjä parametreja pystyisi ohjaamaan ilman VR-lasien vaihtoa ohjattavalta ohjaajalle. Lisäksi etäohjauksen hyöty on, että ohjaajan ja ohjattavan ei tarvitse olla samassa tilassa.

## Abstract

Marttila, Kari: Developing Remote Application for Unity-based VR-game

Bachelor's thesis

Degree programme: Business Information Systems

February 2023

Number of pages: 63

The goal of this thesis was to develop a procedure to control the starting parameters of VR-game from remote device.

The client of this thesis was the co-operation project between Satakunta university of Applied Sciences (SAMK), Tampere University of Applied Sciences (TAMK), Tampere University (TAU) and Metropolia University of Applied Sciences called Collaborations with Networks through artificial intelligence-based virtual technologies for rehabilitation.

In this thesis the web based remote controller application was developed for game called Whack-a-fysio. The game was previously developed by SAMK.

The background for this thesis was the desire of the rehabilitation professionals to have rehabilitation game parameters controlled individually without needing to dress the VR-headset. In addition, the benefit of the remotely controlled VR-game is that the instructor and the player do not need to be in the same place.

# SISÄLLYS

1	Johdanto .....	6
2	Toimeksiantajan esittely .....	7
3	Etäohjauksen merkitys ja hyödyt .....	8
4	Whack-a-Fysio-pelin esittely .....	10
4.1	Pelin idea .....	10
4.2	Pelissä olevat parametrien säätimet .....	10
4.3	Pisteenlasku .....	12
5	Sovelluskokonaisuuden rakenne .....	14
6	Etäohjaussovelluksen Suunnittelu .....	16
6.1	Sovelluksen suunnittelu .....	16
6.1.1	Tarvittavat komponentit .....	16
6.1.2	Ohjelman asettelun (layoutin) suunnittelu .....	16
7	Etäohjainsovelluksen toteutus .....	18
7.1	Sovelluksen toiminta .....	18
7.2	HTML .....	20
7.2.1	HTML DOM .....	21
7.2.2	HTML Etäohjainsovelluksessa .....	21
7.3	CSS .....	22
7.3.1	CSS Etäohjainsovelluksessa .....	22
7.4	JavaScript .....	22
7.4.1	JavaScript etäohjainsovelluksessa .....	23
7.5	PHP .....	24
7.5.1	PHP Etäohjainsovelluksessa .....	25
7.6	JSON .....	26
7.6.1	JSON Etäohjainsovelluksessa .....	27
8	Whack-a-fysio-peliin tehtävät muutokset .....	30
8.1	Unity ja Whack-a-fysio .....	30
8.2	Parametrien lataaminen palvelimelta .....	31
8.3	Parametrien sijoittaminen Whack-a-fysion käyttöliittymään .....	36
9	Jatkokehitysjatatuksia .....	38
10	Yhteenveto .....	39
	LÄHTEET .....	40
	LIITTEET .....	42

## SYMBOLI- JA LYHENNELUETTELO

Apache HTTP Server	Avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma
C#	Olioperustainen vahvasti tyyplitetty ohjelmointikieli
CSS	Cascading Style Sheets, suom. verkkosivun tyylimäärittelyt
HTML	Hypertext Markup Language, suom. hypertekstin merkintäkieli
HTTP	<i>Hypertext Transfer Protocol</i> , suom. hypertekstin siirtoprotokolla
JavaScript	Verkkoympäristöissä käytettävä ohjelmointikieli
PHP	Hypertext Preprocessor, suom. hypertekstin esikäsittelijä
Unity	Unity Technologiesin kehittämä pelimoottori
VR	Virtual Reality, suom. virtuaalitodellisuus

## 1 JOHDANTO

Tämän opinnäytetyön aihe on Unityllä toteutetun VR-pelin etäohjauksen suunnittelu ja toteutus. Opinnäytetyössä esitetään ja toteutetaan yksi tapa, jolla ohjain voidaan toteuttaa.

Työssä tehtiin etäohjaus Satakunnan ammattikorkeakoululla yhteistyössä Fyysiotähti Oy:n kanssa aiemmin luotuun kuntoutukseen tarkoitettuun ”Whack-a-fysio”-peliin.

Opinnäytetyön toimeksiantaja on Satakunnan ammattikorkeakoulun (SAMK), Tampereen ammattikorkeakoulun (TAMK), Tampereen yliopiston (TAU) ja Metropolia-ammattikorkeakoulun yhteishanke: ”Verkostoyhteistyöllä vauhtia tekoälypohjaisten virtuaalitekniologioiden saavutettavuuteen kuntoutuksessa”.

Opinnäytetyössä kehitettyyn etäohjaimen käyttöön käytettiin HTML, JavaScript ja PHP- webkehityskieliä. Sovellus sijoitettiin palvelimelle, josta peli voi hakea asetustietoja. Pelin Unity-projektia muokattiin niin, että siihen luotiin C#-ohjelmakooditiedosto, joka lataa palvelimella olevan parametrin pelin käytettäväksi. Lisäksi Unity-pelimoottorissa pelin käyttöliittymään tehtiin tarvittavat muutokset palvelimelta haettavan datan hyödyntämiseksi.

## 2 TOIMEKSIANTAJAN ESITTELY

Hanke, joka toimii tämän opinnäytetyön toimeksiantajana, on nimeltään: ”Verkostoyhteistyöllä vauhtia tekoälypohjaisten virtuaaliteknologioiden saavutettavuuteen kuntoutuksessa”. Hanketta rahoittaa opetus- ja kulttuuriministeriö.

Hankkeeseen osallistuvat tahot ovat Satakunnan ammattikorkeakoulu (SAMK), Tampereen ammattikorkeakoulu (TAMK), Tampereen yliopisto (TAU) sekä Metropolia-ammattikorkeakoulu.

Hankkeen kohderyhmiä ovat: kuntoutusalan ammattilaiset, opiskelijat sekä nuoret kuntoutujat. Erityisesti huomio kiinnittyy haavoittuvassa asemassa oleviin kohderyhmiin. Näitä ovat mm. toimintakyvyltään rajoittuneet henkilöt, osatyökykyiset, sekä ammatillista koulutusta vaille olevat nuoret.

Hankkeen tavoite on luoda toimiva yhteistyöverkosto oppilaitosten ja niissä toimivien asiantuntijoiden välille. Tavoite on myös kerätä tietoa ja kokemuksia tekoälypohjaisten virtuaaliteknologioiden hyödyistä kuntoutuksessa, sekä varmistaa niiden saavutettavuus. Lisäksi hankkeen tavoite on innovoida, tutkia ja tuottaa lisää tekoälypohjaisen virtuaaliteknologian osaamista kuntoutuksen ammattilaisille ja kuntoutujille opiskelu-, työ- ja toimintakyvyn edistämiseen. Hanke on käynnissä 1.1.2020 ja 31.12.2022 välisenä aikana. (Virtuaalikuntoutusverkosto ,2022).

### 3 ETÄOHJAUKSEN MERKITYS JA HYÖDYT

Viimevuosina on kuntoutuksessa erilaiset sähköiset palvelut lisääntyneet huomattavasti. Näillä palveluilla on mahdollista lisätä ihmisten mahdollisuutta huolehtia omasta terveydestään ja hyvinvoinnistaan. Virtuaalitodellisuutta voidaan hyvin käyttää kuntoutuksessa, koska siinä on monia muuttujia, joita terapeutti voi säätää. Virtuaalinen kuntoutus vaatii terapeutin ohjausta, valvontaa, muokkausta ja tulkintaa. Virtuaalitodellisuudessa tapahtuvia harjoitteita voidaan pitää terapiana silloin kun sitä ohjaa ammattilainen ja sillä on selkeä tavoite. Virtuaalinen kuntoutus mahdollistaa, että terapeutin jatkuvaa valvontaa ei välttämättä tarvita. Tämä mahdollistaa kustannustehokkaan kuntoutuksen. Virtuaalikuntoutuksen avulla voidaan lisätä motivaatiota, toistoja ja kertoja. Lisäksi kuntoutuja voi saada palautetta omasta toiminnastaan. (Heiskanen T. ym. 2016).

Jotta VR-pelin näkymä hyödyttäisi juuri tiettyä kuntoutujaa tulee ennen pelin aloittamista tehdä pelin asetuksiin kuntoutujalle yksilöidyt asetukset. VR-laitteilla kuntoutuspelien kohdalla on ongelmana koettu, että kuntoutukseen tarkoitettujen sovellusten asetukset kuntoutujalle pitää asettaa virtuaalimaailmassa. Kuntoutuksen ammattilaisen pitää henkilökohtaisesti olla paikalla asettamassa pelin parametrit kuntoutujan mukaan sopivaksi, tai vaihtoehtoisesti neuvoa kuntoutujaa, kuinka parametrit tulisi asettaa. Tämä saattaa olla hankalaa, koska kuntoutujia on eri tasoisia niin liikkuvuuden, kuin tietoteknisen osaamisen suhteen. Käytännössä ohjaajan on puettava VR-lasit päähän, tehtävä tarvittavat muutokset, jonka jälkeen lasit puetaan ohjattavalle. Tässä vaiheessa pelin näkymä saattaa olla siirtynyt ohjattavan ulottumattomiin tai muuten väärään paikkaan. (Holappa, N, Merilampi, S. 2022)

Etäohjauksella on myös mahdollista ohjata kuntoutujaa täysin eri maantieteellisessä sijainnissa. Etäohjaus lisää kuntoutuksen saavutettavuutta niin matkustuksen kuluja ja matkustamiseen käytetyn ajan vähentyessä. Joillekin kuntoutusryhmille saattaa kotona suoritettava kuntoutus, ilman toisen ihmisen



fyysistä läsnäoloa tutussa ympäristössä olla parempi vaihtoehto, kuin vieraassa paikassa tapahtuva ohjaus. (Pot-Kolder R ym. 2020)

## 4 WHACK-A-FYSIO-PELIN ESITTELY

Whack-a-fysio-peli on Satakunnan ammattikorkeakoulun ja Fysiotähti Oy:n yhdessä kehitetty VR-peli. Pelin idea perustuu whack-a-mole peliklassikkoon.

Peli on luotu kuntoutuksen ammattilaisten työkaluksi kuntoutukseen. (Merilampi, S. ,2021)

### 4.1 Pelin idea

Virtuaalimaailmassa pelaajan edessä on seinä, jossa on 42 kuutiota ruudukossa. Pelaajan tavoite on osua kuutioihin VR-lasien käsiohjaimiin liitetyillä pampuilla (Kuva 1.), sitä mukaa, kun kuutiot syttyvät eri värisinä. Pamput ovat eri kädessä eriväriset. Vasemman käden pamppu on sininen ja sillä on tarkoitus osua sinisenä syttyviin kuutioihin. Oikean käden pamppu on punainen ja sillä on tarkoitus osua punaisena syttyviin kuutioihin. Mikäli kuution väri on purppura saa pelaaja osua kuutioon kummalla kädellä tahansa. Pelin muutettavia asetuksia ovat kuutioiden mahdollinen väri, kuutioiden syttymisnopeus sekä pelin kesto osumina -ja aikana.



Kuva 1. Kuvakaappaus whack-a-fysiosta

### 4.2 Pelissä olevat parametrien säätimet

Pelin alussa olevista säätimistä voidaan vaihtaa pelin asetuksia. Säädettäviä parametrejä ovat: Kuutioiden kätisyys (Kuva 2.), pelin kesto (Kuva 2.), syttymisen aikaväli (Kuva 2.), pelin kesto osumissa (Kuva 3.) sekä kuutioiden koko

(Kuva 3.). Lisäksi pelissä voi valita onko VR-lasien ohjaimiin liittyvä laserosoitin käytössä (Kuva 2.).

Kuutioiden käteisyysäädintä (Kuva 1.) ohjataan niin, että peliohjaimiin liitetyillä pampuilla kosketaan säätimessä olevan 6x7 ruudukon elementtejä. Koskettaessa yksittäiseen elementtiin pampulla vaihtaa se väriä. Väri muuttuu sitä mukaan, kumman käden pampulla kuutiota osutaan. Mikäli elementtiä koskee molemmilla pampuilla, muuttuu sen väri violetiksi.

Säätimissä on kolme eri numeroarvojen valintaa. Näitä ovat kesto (Kuva 2.), aikaväli (Kuva 2.) ja kuinka monta osumaa peli kestää (Kuva 3.). Näiden säätimien numeroarvojen skaalat on esitetty taulukossa 1.

Kuutioiden koon muuttamisen säätimessä (Kuva 3.) on viisi eri kokovaihtoehtoa. Laserosoittimen säätimessä on vaihtoehtoina joko päällä tai pois päältä.



Kuva 2. Kuvakaappaus 1. säätimestä



Kuva 3. Kuvakaappaus 2. säätimestä

Taulukko 1. Numerosäädinten skaalat

Säädin	Minimi	Maksimi	Porras
Kesto	1	9	1
Aikaväli	1	9	1
Pelin kesto osumina	0	$\infty$	10

#### 4.3 Pisteenkalkulaatio

Peli päättyy, kun peliin määritelty ajallinen kesto tai osuttujen osumien määrä tulee täyteen. Pelin päättyessä ilmestyy pelaajan eteen ruutu, joka kertoo joitakin tunnuslukuja pelatusta pelistä. Nämä tunnusluvut ovat reaktioaika, pelin kesto, pisteet, viimeisin combo ja paras combo. Reaktioaika kertoo kahden osuman välisen keskimääräisen ajan. Pelin kesto ilmaisee, kuinka kauan peliä pelattiin. Pisteet kertovat kuinka moneen palikkaan pelaaja kaiken kaikkiaan osui. Viimeisin combo kertoo, kuinka moneen palikkaan pelaaja osui ennen kuin peli loppui. Paras combo ilmaisee, kuinka moneen palikkaan pelaaja osui

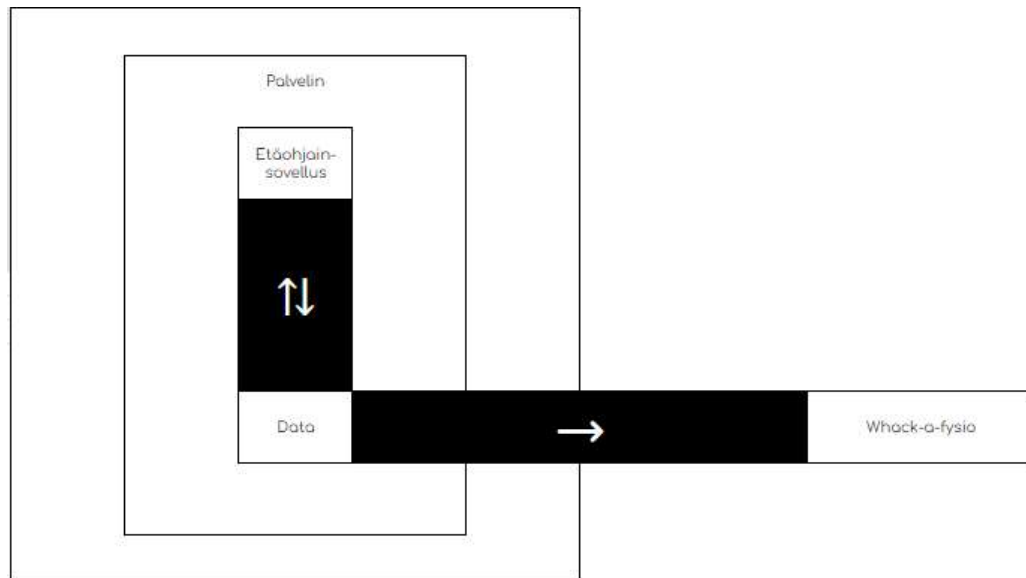
peräkkäin parhaimmillaan. Pelin lopuksi ilmestynvä pistetaulu on esitetty kuvassa 4.



Kuva 4. Pelin pistetaulu

## 5 SOVELLUSKOKONAISUUDEN RAKENNE

Sovelluskokonaisuus (Kuva 5.) koostuu palvelimella sijaitsevasta etäohjaimesta, etäohjaimen palvelimelle tallentamasta parametridatasta sekä itse Whack-a-fysio pelistä.



Kuva 5. Järjestelmän rakenne

Palvelimen käyttöjärjestelmäksi valikoitui Debian Linux-käyttöjärjestelmä, koska sitä pidetään yleisesti vakaana ja luotettavana palvelinkäyttöjärjestelmänä. Se on yksi vanhimmista Linux-käyttöjärjestelmistä ja näin ollen myös yksi suosituimmista, parhaiten dokumentoiduista ja helpoimmista käyttää. Tämän ohjelmakokonaisuuden kannalta ei ole suurta merkitystä minkä Linux-käyttöjärjestelmän palvelimelle asentaa. (Pontikis, C. 2013).

Palvelimelle asennettiin Apachen Software Foundation avoimeen lähdekoodiin pohjautuva HTTP-palvelinohjelma Apache HTTP Server, joka mahdollistaa tiedostojen jaon verkon yli HTTP-protokollalla. (Apache. 2022). HTTP (lyhenne sanoista Hypertext Transfer Protocol) on tiedonsiirtoprotokolla, jota käytetään palvelinten ja asiakasohjelman väliseen tiedonsiirtoon. Protokolla perustuu siihen, että asiakasohjelma lähettää pyynnön palvelimelle. Palvelin vastaa lähettämällä sopivan vastauksen, kuten HTML-sivun, kuvan tai ääntä.

Tässä ohjelmakokonaisuudessa, palvelin lähettää pelissä tarvittavia parametritietoja itse pelille. (MDN Web Docs 2022).

## 6 ETÄOHJAUSSOVELLUKSEN SUUNNITTELU

### 6.1 Sovelluksen suunnittelu

Sovelluksen toteutuksen lähtökohdaksi otettiin pelissä jo olemassa oleva käyttöliittymä. Toteutukseen vaikutti itse pelissä ja etäohjaussovelluksessa olevien säätöjen erilaisuus. Etäohjainta käytetään joko hiirellä tai kosketusnäytöllä. VR-maailmassa peliä ohjataan VR-lasien ohjaimilla.

#### 6.1.1 Tarvittavat komponentit

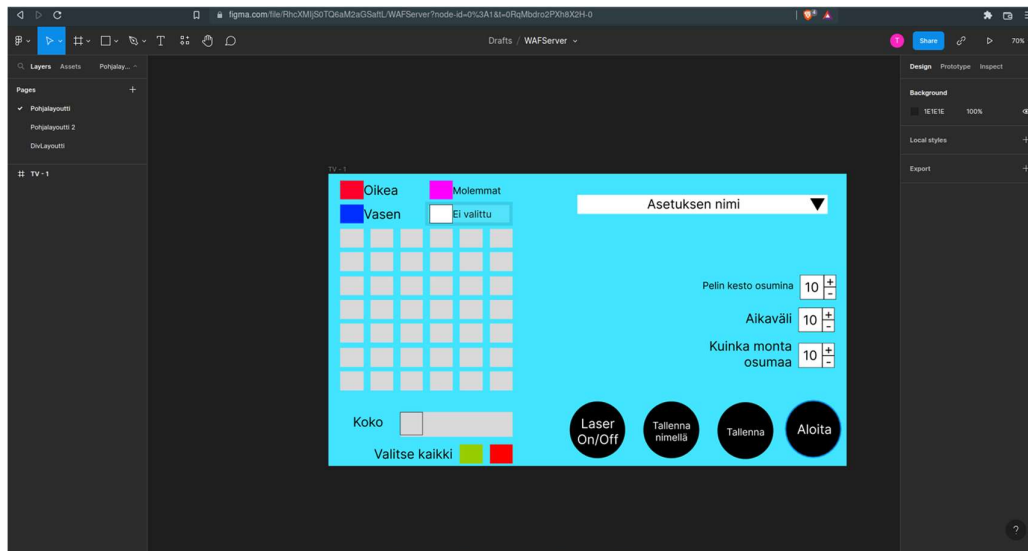
Ohjainsovellukseen tarvittavat säädinkomponentit saatiin pelin olemassa olevista käyttöliittymän elementeistä, nämä on käyty läpi kappaleessa: "4.2 Pelissä olevat parametrien säätimet". Ohjelmaan tarvittiin elementit 42 kuution värin muuttamiseksi, kuutioiden koon muuttamiseksi, numerosäätimet pelin keston, lyötävien kuutioelementtien ilmestymisaikaan ja siihen miten monta lyöntiä peli kestää, sekä laserohjaimen päälle ja poiskytkemiseksi painike. Ohjelmaan tarvittiin myös elementit, jolla voidaan päästä käsiksi tallennettuihin asetuksiin ja näihin tarvittavat tallennus- ja poistopainikkeet.

#### 6.1.2 Ohjelman asettelun (layoutin) suunnittelu

Ohjelman asettelun suunnittelussa on lähtökohtana pyritty siihen, että se skaalautuu tietokoneen 16:9 kuvasuhteen näytölle (Kuva 6.), sekä matkapuhelimen tai tabletin näytölle (Kuva 7.). Tämän ohjelman vaatimuksena ei ollut erityisen monimutkainen asettelu. Yksinkertaisen asettelun suunnitteluun sopii varsin hyvin esimerkiksi Figma-websovellus.

Figma (Kuva 4.) on Figma, Inc.- yhtiön kehittämä verkkosovellus, jota voidaan käyttää useamman käyttäjän yhteistyöprojekteihin. Se on kehitetty erityisesti käyttöliittymän ja käyttäjäkokemuksen suunnitteluun soveltuvaksi työkaluksi. (Figma n.d.)





Kuva 6. Kuva Figman käyttöliittymästä ja 16:9 kuvasuhteen näytölle tehdystä asettelusta.



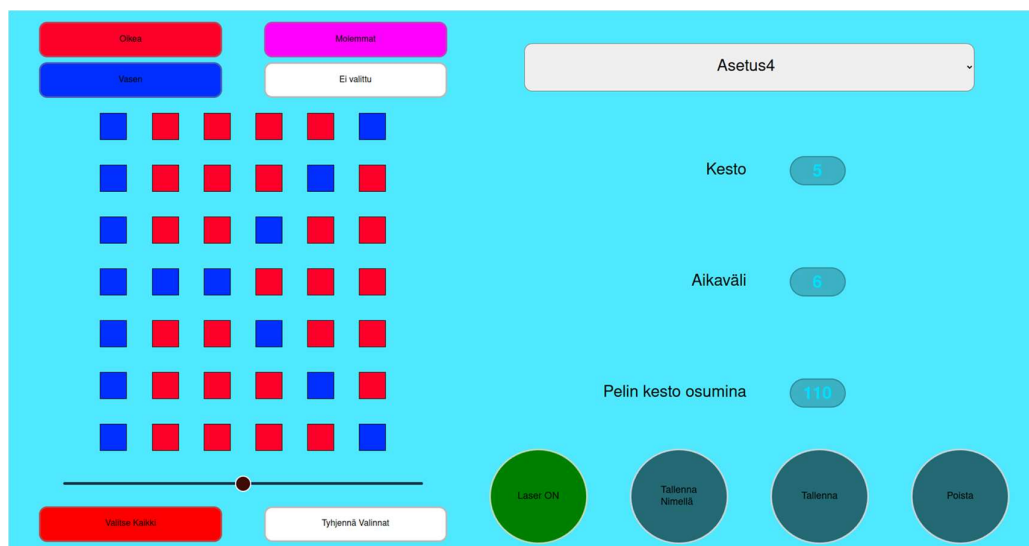
Kuva 7. Suunnitelma etäohjaussovelluksen asetteluksi 9:16 näytölle

## 7 ETÄOHJAINSOVELLUKSEN TOTEUTUS

Etäohjainsovelluksen tavoitteena oli, että sen käyttöliittymän avulla voidaan luoda JSON-muotoinen datatiedosto, joka sisältää pelin kannalta oleelliset parametrit. Tähän käytettiin HTML-, JavaScript- ja PHP-webkehityskieliä. Lisäksi ohjelma sisältää tyylimäärityitä sisältäviä CSS-kielen tiedostoja.

### 7.1 Sovelluksen toiminta

Sovellus toimii niin, että sen käynnistyessä käyttäjälle avautuu kaikki pelissä tarvittavat säätimet (Kuva 8.).



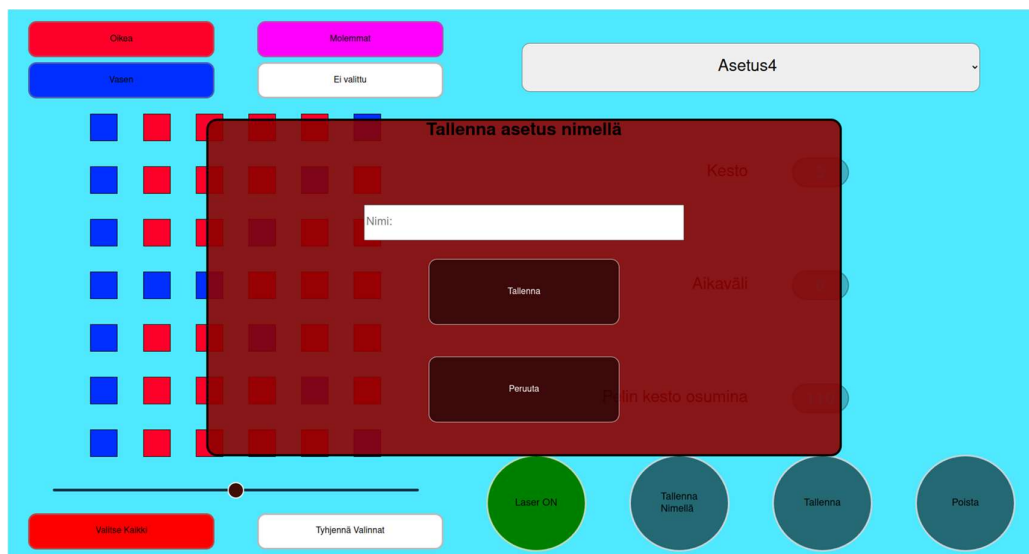
Kuva 8. Sovelluksen aloitusnäky

Käyttäjä voi värinapeista painamalla valita minkä väriseksi haluaa pelin 42 kuutiota muuttua. Värit ovat pelin mukaiset, jossa punainen tarkoittaa oikeaa pamppua, sininen vasenta, purppura molempia ja valkoisella ilmaista, ettei kuutio ole käytössä.

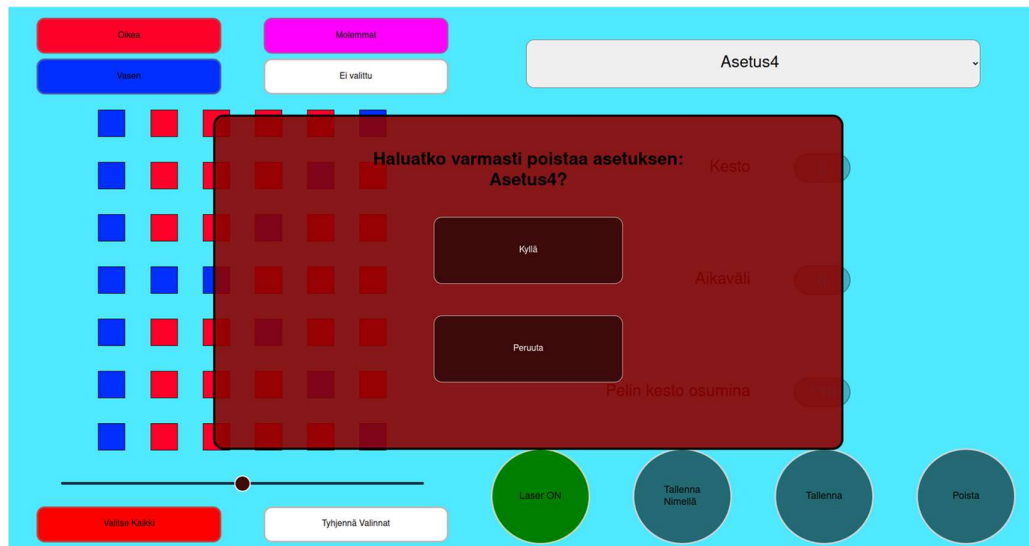
Lisäksi sovelluksessa on liukukytkin, josta kuutioiden kokoa voidaan säätää. Sovelluksessa on myös pikavalinnat, joilla voidaan kaikkien kuutioiden väri muuttaa kerralla ja vastaavasti tyhjentää kuutioiden valinnat. Lisäksi sovelluksessa on pudotusvalikko, josta käyttäjä näkee mitä asetuksia palvelimelle on

tallennettu, sekä voi valita mitä asetusta haluaa muokata tai poistaa. Sovelluksessa on tarvittavat numerovalitsimet kestolle, aikavälille ja osumien määrälle. Laser On/Off painikkeesta saa säädettyä halutaanko peliin käyttöön VR-ohjainten laserohjaus.

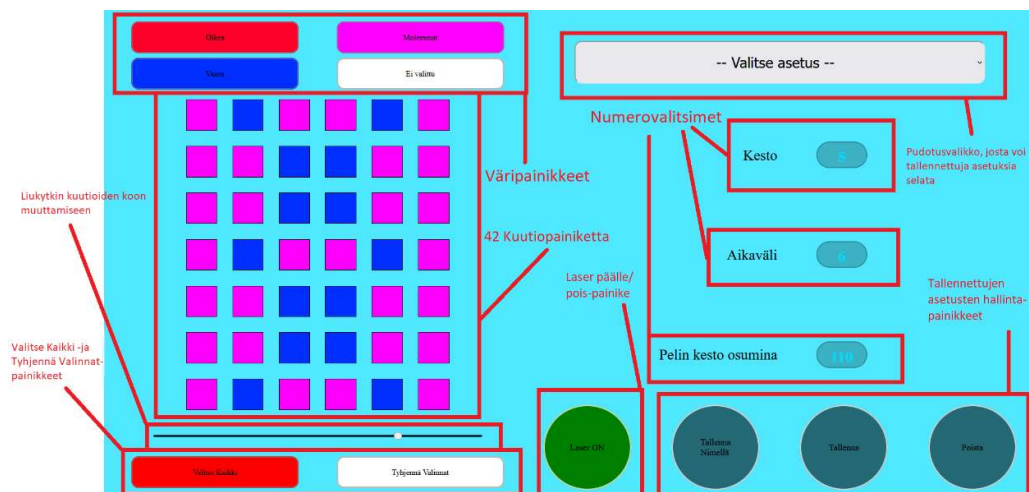
Tallenna nimellä- painikkeella voidaan tallentaa ruudulla näkyvissä oleva asetus uudella nimellä. Painiketta painamalla käyttäjälle avautuu näkymä johon käyttäjä voi syöttää uuden asetuksen nimen tai perua tallennuksen (Kuva 9.). Tallenna-painikkeella voidaan korvata pudotusvalikosta valitun asetuksen tilalle ruudulla näkyvät asetukset. Poista-napilla voidaan poistaa pudotusvalikosta valittu asetus. Painiketta painamalla avautuu käyttäjälle laatikko, jossa sijaitsevilla painikkeilla käyttäjän on mahdollista vahvistaa asetuksen poisto tai peruuttaa se (Kuva 10.). Elementtien sijoittelu on esitetty kuvassa 11.



Kuva 9. Näkymä käyttäjän painettua: "Tallenna nimellä"-painiketta.



Kuva 10. Näkymä käyttäjän painettua "Poista"-Painiketta.



Kuva 11. Elementtien sijoittelu.

## 7.2 HTML

HTML-kieli (Kuva 12.) on verkon ensisijainen merkintäkieli, joka määrittää miten kuvat, multimedia ja teksti näkyy verkkoselaimessa. HTML-kielen peruselementtejä on tagit ja elementit. Tavallisimmillaan elementit sisältävät aloitus- ja lopetustunnisteen. Lopetustunniste merkitään vinoviivalla. (w3schools.com n. d.,a)

```

<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>

```

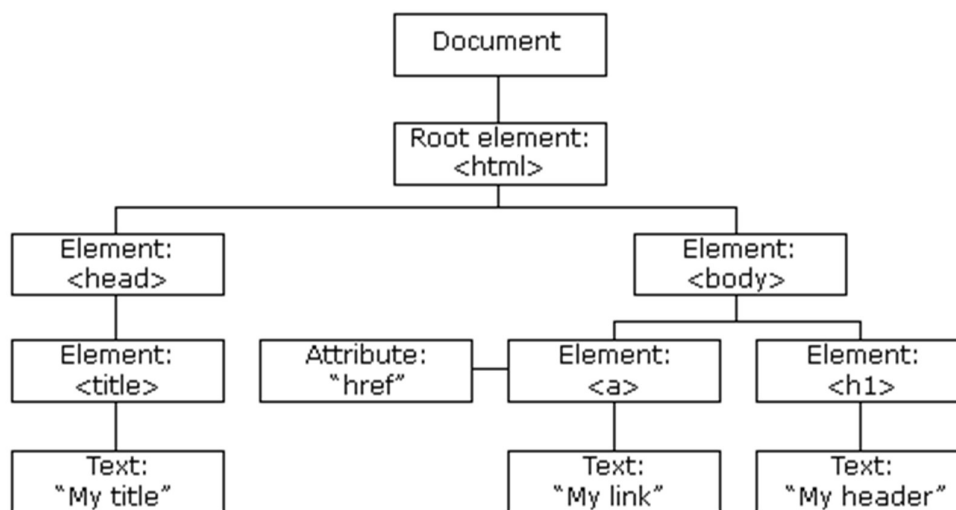
# This is a Heading

This is a paragraph.

Kuva 12. HTML-merkintäkielen syntaksia vasemmalla. Oikealla nähdään miltä se näyttää web-sivulla (w3schools.com n. d.,b)

## 7.2.1 HTML DOM

HTML DOM (Document object model) on verkkosivuston lataamisessa muodostuva puumainen rakenne (Kuva 13.), joka määrittelee, miten verkkosivusto näkyy selaimessa. Tämä rakenne voidaan luoda, joko kirjoittamalla se suoraan HTML-tiedostoon, tai esimerkiksi ohjelmoimalla se JavaScript-koodissa. (w3schools.com n. d.,b)



Kuva 13. HTML DOM-elementtien puurakenne. (w3schools.com n. d.,b)

## 7.2.2 HTML Etäohjainsovelluksessa

Etäohjainsovelluksessa HTML-tiedostoja (Liite 1.) on yksi: "index.html". Index.html on verkkosovelluksissa yleisesti oletussivun nimi, joka latautuu, kun

Apache-palvelimen URL-osoite selaimella valitaan. Tässä toteutuksessa HTML-tiedosto sisältää lähinnä määrittelyt tiedostoihin, joissa itse sisältö ja sisällön tyylimäärittelyt luodaan.

### 7.3 CSS

CSS (Cascading Style Sheets) on kieli, jolla voidaan määrittellä, kuinka verkkosivudokumentit näkyvät käyttäjälle. CSS-kieltä voidaan käyttää perusmuotoilujen kuten värien, kirjasimien, kirjasimen kokojen, elementtien kokojen jne. määrittelyyn. Esimerkkikuvassa (Kuva 14.) näkyy CSS-tiedoston syntaksia. Siinä on määritelty h1-nimisten elementtien kirjasimen väri punaiseksi (color: red) ja kirjasimen koko viisinkertaiseksi oletuskirjasimeen nähden (font-size: 5em). (Tutorialspoint n.d.)

```
h1 {  
  color: red;  
  font-size: 5em;  
}
```

Kuva 14. CSS-kielen syntaksia

#### 7.3.1 CSS Etäohjainsovelluksessa

Ohjelman CSS-määrittelyt on sijoitettu kolmeen CSS-tiedostoon. Tiedostoon: "./css/style.css" (Liite 2.) on kirjoitettu suurin osa ohjelman tyylimäärittelyistä. Ohjelman muut tyyli-tiedostot ovat: "./css/sliderstyle.css" (Liite 3.), jossa määritellään ohjelmassa käytettävän liukukyttimelementin tyyli. "./css/numbersstyle.css" (Liite 4.) -tiedostossa taas määritellään ohjelman kolmen numeroelementin käyttämät tyyli-muotoilut.

### 7.4 JavaScript

JavaScript on ohjelmointikieli, joka tunnetaan parhaiten dynaamisten ja interaktiivisten verkkosovellusten -ja sivujen luomiseen käytettävänä kielenä. Ja-

vaScript on nykyään maailman yleisin ohjelmointikieli. Sitä on käytössä nykyään 97 %:ssa kaikista verkkosivuista. Nykyaikaiset verkkoselaimet tunnistavat oletuksena JavaScriptiä.

Ennen kuin JavaScript kehitettiin, tehtiin verkkosivut ainoastaan pohjautuen HTML -ja CSS-kieliin. Ainoastaan näitä kieliä käyttämällä on kuitenkin mahdollista luoda staattisia verkkosivuja, jotka eivät ole käyttäjän kanssa interaktiivisia.

JavaScriptillä voidaan luoda esimerkiksi seuraavia elementtejä: Piilotettavat ja esiin vedettävät valikot, leijuntaefektit (tarkoittaa tilannetta, jossa kursori on elementin päällä), animaatiot, jne.

JavaScriptiin on saatavilla myös erilaisia kirjastoja eri tarpeisiin, näistä tunnetuimpia on mm. React, React Native ja Node.js. Tässä työssä käytetään vain JavaScriptin peruselementtejä, eikä mitään edellä mainituista tai muistakaan kirjastoista. (Meltzer, R. 2021)

#### 7.4.1 JavaScript etäohjainsovelluksessa

Sovelluksen kaikki näkyvät osat on ohjelmoitu JavaScriptillä. Ohjelmassa on JavaScript tiedostoja yhteensä 10 kappaletta. JavaScript tiedostojen (Liitteet 5.–14.) toiminta on esitetty taulukossa 2. Ohjelmassa JavaScriptin tarkoitus on sekä luoda käyttäjälle käyttöliittymä, että luoda käyttöliittymän pohjalta parametrit PHP-rajapinnalle.

Taulukko 2. Ohjelman JavaScript tiedostot ja kuvaus niiden toiminnasta.

<b>Tiedosto</b>	<b>Toiminta</b>
./js/createMainDivs.js	Koodi luo lohkotason elementtejä (div-elementit)
./js/createSecondaryDivs.js	Koodi luo lohkotason elementtejä (div-elementit)
./js/createThirdDivs.js	Koodi luo lohkotason elementtejä (div-elementit)

Tiedosto	Toiminta
./js/createFourthDivs.js	Koodi luo säädinelementit.
./js/slider.js	Liukusäätimen logiikka.
./js/numberInput.js	Numerosäätimien logiikka.
./js/buttonslogik.js	Nappien toimintalogiikka.
./js/createSaveObject.js	Koodi luo tallennettavan objektin säädinelementtien arvoista
./js/fetchFromJson.js	Haetaan olemassa olevat tallennusarvot ja asetetaan ne ohjelmaan
./js/windowresize.js	Tässä tiedostossa päivitetään ohjelman arvoja, mikäli selainikkunan kokoa muutetaan.

## 7.5 PHP

PHP (Hypertext Preprocessor) on ohjelmointikieli, joka soveltuu erityisen hyvin käytettäväksi web-kehitykseen HTML-kielen kanssa (Kuva 15.). PHP-kieli on palvelinpuolella suoritettava komentosarjakieli.

```

<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>

```

Kuva 15. PHP-koodia HTML-tiedostossa. Kuvan koodi tulostaa tekstin: "Hi, I'm a PHP script!" (php. n.d.).



Palvelinkielen, kuten PHP ja asiakaskielen, kuten JavaScript ero on siinä, että palvelinkielellä voidaan suorittaa komentoja itse palvelimella. Esimerkiksi tämän työn sovelluksessa PHP-kielellä luodaan, tai muokataan palvelimella sijaitsevaa json-päätteistä tekstitiedostoa. (php. n.d.).

PHP-koodi voidaan sijoittaa HTML-elementtien joukkoon sijoittamalla syntaksi alkutagin: "<?php" ja lopputagin: "?>" väliin. PHP-koodit voidaan sijoittaa myös omiin tiedostoihinsa, jotka voidaan suorittaa selaimessa avaamalla PHP-kooditiedoston URL-osoite. (php. n.d.). Tämän työn etäsovellusohjelmassa PHP-tiedostot avataan suoraan omista tiedostoistaan, eikä PHP-koodia ole sijoitettu suoraan HTML-koodiin.

### 7.5.1 PHP Etäohjainsovelluksessa

Sovelluksessa on kolme eri PHP-kooditiedostoa. Tiedoston "./php/deletepreset.php" (Kuva 16.) koodi poistaa yhden esiasetuksen, sen mukaan mikä indeksinumero koodiin on sovelluksen käyttöliittymältä tuotu.

```
1  <?php
2
3  $jsonString = file_get_contents('save.json');
4  $indexToDelete = $_GET['index'];
5  $data = json_decode($jsonString);
6
7  unset($data->presets[$indexToDelete]);
8  $data->presets = array_values($data->presets);
9
10 $newJsonString = json_encode($data, JSON_PRETTY_PRINT);
11 file_put_contents('save.json', $newJsonString);
12 header("Location: ../index.html");
13 ?>
```

Kuva 16. Kooditiedosto: "./php/deletepreset.php".

Tiedosto "./php/savejson.php" (Kuva 17.) tallentaa JSON-tiedostoon uuden arvon vanhojen lisäksi. Koodiin vietään ohjelmasta uusi tallennettava JSON-tyyppinen string parametri. Tämä objekti vietään jo olemassa olevien tallennusobjektien lisäksi.

```

1 <?php
2
3 $jsonString = file_get_contents('save.json');
4 $newPresetJsonString = $_GET['json'];
5
6 $data = json_decode($jsonString);
7 $dataToSave = json_decode($newPresetJsonString);
8 $data->presets[0] = $dataToSave;
9 array_push($data->presets , $dataToSave);
10 $newJsonString = json_encode($data, JSON_PRETTY_PRINT);
11 file_put_contents('save.json', $newJsonString);
12 header("Location: ../index.html");
13 ?>

```

Kuva 17. Kooditiedosto: ”./php/savejson.php”.

Tiedosto ”./php/replacejson.php” (Kuva 18.) tallentaa JSON-tiedostoon uuden arvon vanhan päälle. Koodiin viedään parametrinä tallennettavan string-objektin lisäksi korvattavan asetuksen indeksinumero.

```

1 <?php
2
3 $jsonString = file_get_contents('save.json');
4 $newPresetJsonString = $_GET['json'];
5 $indextoreplace = $_GET['index'];
6 $data = json_decode($jsonString);
7 $dataToSave = json_decode($newPresetJsonString);
8 $data->presets[0] = $dataToSave;
9 $data->presets[$indextoreplace] = $dataToSave;
10 $newJsonString = json_encode($data, JSON_PRETTY_PRINT);
11 file_put_contents('save.json', $newJsonString);
12 header("Location: ../index.html");
13 ?>

```

Kuva 18. Kooditiedosto: ”./php/replacejson.php”.

## 7.6 JSON

JSON (JavaScript Object Notation) on kevyt ja helposti ymmärrettävä tiedostomuoto, jonka tarkoitus on tallentaa ja välittää tietoja. JSON on täysin kielestä riippumaton, vaikka sen nimessä sana JavaScript esiintyykin. Se kuitenkin käyttää monen yleisen ohjelmointikielen käytänteitä. (Json. n.d.)

JSON-syntaksi koostuu avain/arvopareista. Avain on aina heittomerkkien välissä oleva string-tyyppinen muuttuja. Arvo voi olla tyypiltään string, numero, boolean, array tai objekti. Nämä ovat perus datatyyppisiä, jotka muodossa tai toisessa löytyvät kaikista nykyaikaisista ohjelmointikielistä. Avain/arvoparit

noudattavat tiettyä syntaksia, jossa arvo seuraa avainta. Avain/arvoparit erotellaan toisistaan pilkulla (.). (Squarespace. n.d.). Eri arvotyyppien esittäminen JSON-syntaksissa on esitetty kuvassa 19.

```

{ } //Empty JSON object

{
  "StringProperty": "StringValue",
  "NumberProperty": 10,
  "FloatProperty": 20.13,
  "BooleanProperty": true,
  "EmptyProperty": null
}

{
  "NestedObjectProperty": {
    "Name": "Neste Object"
  },
  "NestedArrayProperty": [10,20,true,40]
}

```

Kuva 19. JSON-arvotyytit.

### 7.6.1 JSON Etäohjainsovelluksessa

Etäohjainsovelluksessa on yksi JSON-tiedosto (Kuva 20), joka sisältää kaiken datan joka itse peliin siirretään. Tiedosto muuttuu sitä mukaan, kun sovellusta käytetään. Sovelluksen käyttämä JSON-tiedosto on muotoa, jossa tiedoston pääavain on nimeltään: "presets". "presets"-avaimen arvopari on Array-tyyppinen. Tämä array sisältää tallennettujen asetusten arvoja. Yksi tallennus, eli yksi edellä mainitun arrayn alkio koostuu seuraavista avaimista: presetname, blocks, size, duration, frequency, hits ja laser. Edellä mainittujen arvojen selitys on esitetty taulukossa 3.

Taulukko 3. JSON-tiedostossa olevien tallennuselementtien avaimet, niiden tarkoitukset ja niiden tyypit

Arvo	Tarkoitus	Tyyppi
presetname	Tallennuksen nimiarvo	String -tyyppinen muuttuja
blocks	Kätisyysäätimen kuutioiden arvot, jokainen merkki merkitsee kätisyysarvoa 42	42 kirjaiminen string-muuttuja

	kuution kohdalla. (l = vasen käsi, r = oikea käsi, b = molemmat kädet ja n = ei valintaa)	
size	Kuutioden koko	Numero, mahdolliset arvot 0, 25, 50, 75, 100
duration	Pelin kesto	Numero, arvot 1-9
frequency	Aikaväli	Numero, arvot 1-9
hits	Pelin kesto osumina	Numero, arvot 0-3990, kymmenen välein.
laser	Laserohjaimet pois tai päälle	Boolean

```

{
  "presets": [
    {
      "presetname": "Asetus1",
      "blocks": "llrnllllrllnllrlnnllrlnnllrlnnllrllnllrnl",
      "size": "100",
      "duration": "5",
      "frequency": "1",
      "hits": "70",
      "laser": "true"
    },
    {
      "presetname": "Asetus2",
      "blocks": "bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb",
      "size": "0",
      "duration": "4",
      "frequency": "1",
      "hits": "60",
      "laser": "true"
    },
    {
      "presetname": "Asetus3",
      "blocks": "rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr",
      "size": "25",
      "duration": "4",
      "frequency": "6",
      "hits": "110",
      "laser": "true"
    },
    {
      "presetname": "Asetus4",
      "blocks": "lrrrrllrrrlrllrllrrllrrrlrllrllrrll",
      "size": "50",
      "duration": "5",
      "frequency": "6",
      "hits": "110",
      "laser": "true"
    },
    {
      "presetname": "Asetus1",
      "blocks": "llrnllllrllnllrlnnllrlnnllrlnnllrllnllrnl",
      "size": "100",
      "duration": "5",
      "frequency": "1",
      "hits": "70",
      "laser": "true"
    }
  ]
}

```

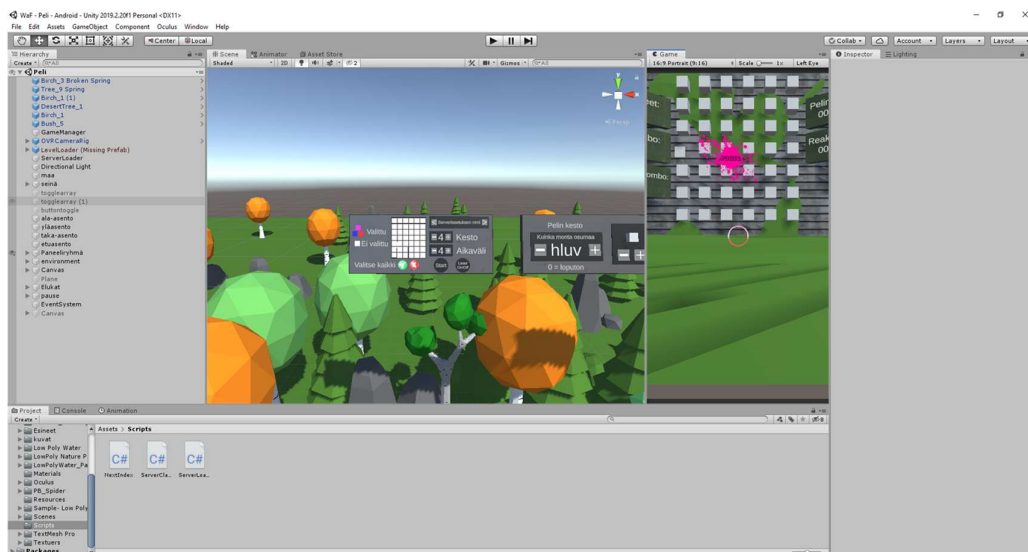
Kuva 20. Esimerkki ohjelman luomasta JSON-datasta.

## 8 WHACK-A-FYSIO-PELIIN TEHTÄVÄT MUUTOKSET

### 8.1 Unity ja Whack-a-fysio

Whack-a-fysio-peli on toteutettu Unity Technologiesin kehittämällä Unity-peli-moottorilla. Unitylle on olemassa versiot Windowsille, Mac -ja Linux-käyttöjärjestelmille. Unityllä voidaan luoda sovelluksia -ja animaatiota useille eri alustoille. Näitä alustoja on esimerkiksi: Windows, Mac, iOS, Android, Playstation ja useat eri VR-alustat. (Unity, 2022). Alun perin Whack-a-fysio on kehitetty Android-pohjaisille Oculus Quest VR-laseille. Peli toimii pienillä muutoksilla myös muilla Android-käyttöjärjestelmän VR-laseilla, kuten esimerkiksi ByteDance-yhtiön Pico-sarjan laseilla.

Unityn käyttöliittymä koostuu eri näkymistä (Kuva 21.). Käyttöliittymässä näitä näkymiä on aseteltu käyttäjän määritysten mukaan. Tyypillisiä näkymiä on esimerkiksi: hierarchy, scene, game, inspector, project ja console.



Kuva 21. Unityn käyttöliittymä ja esimerkki miten näkymäikkunat voidaan sijoittaa

Hierarchy-näkymässä näkyy projektissa käytetyt elementit. Näitä elementtejä voi olla esimerkiksi: 3D-objektit, valot ja kameraobjektit. Scene-näkymässä

käyttäjälle avautuu näkymä itse pelikohtauksesta, näkymässä voi liikkua vapaasti hiirellä navigoiden. Game näkymässä on esitetty kuva sovellukseen liitetystä kamerasta. Sen tarkoituksena on näyttää sitä kuvaa mikä näkyy valmiissa sovelluksessa. Inspector-näkymässä voidaan säätää pelin elementtien parametrejä. Inspector näkymään aukeaa tietoja niistä elementeistä mitä muissa näkymissä valitaan. Project-näkymässä näkyy kaikki peliprojektiin liittyvät tiedostot. Console-näkymään ilmestyy peliin liittyvät varoitukset ja virheilmoitukset. Lisäksi siihen ilmestyy pelin kooditiedostoissa määritellyt tulosteet.

Unityssä käytettävä ohjelmointikieli on Microsoftin .NET alustalle kehitetty C#. C#-kooditiedoston päätte on: ".cs". C# on olio-ohjelmointiin käytettävä kieli, jonka juuret ovat C-kieliperheen kielissä. Monet C#:lle tyypilliset piirteet ovat samoja, kuin esimerkiksi C-, C++-, Java- ja JavaScript-ohjelmointikielissä. C# on vahvasti tyyppitetty ohjelmointikieli, joka tarkoittaa, että jokaisella muuttujalla on oma tyyppinsä. Kielessä voidaan myös luoda uusia tyyppityksiä perustyyppihin pohjautuen. Näitä tyyppisiä kutsutaan luokiksi. (Sheldon, R. n.d.).

Whack-a-fysiossa on käytetty Unityn versiota 2019.2.20f1. Tämä versio on kirjoitushetkellä hieman vanhentunut ja uudempikin on saatavilla. Uuden version käyttäminen vaatisi kuitenkin esimerkiksi pelin Unity-projektissa käytettyjen VR-laseille tarkoitettujen kirjastojen uudelleen päivittämistä ja tämän takia muutamia toimintoja pitäisi luoda uudempien logiikoiden mukaan. Tämän työn asiasisällön kannalta se ei kuitenkaan ole olennaista. Tästä syystä peliprojektia ei lähdetty uudempien Unityn versioiden kanssa yhteensopivaksi päivittämään. Muutokset toimivat pääasiassa samoin myös uudemmilla Unityn versioilla.

## 8.2 Parametrien lataaminen palvelimelta

Datan lataamiseksi luotiin pelin hierarchy näkymään tyhjä peliobjekti ja sille annettiin nimi: "ServerLoader" (Kuva 22.). Lisäksi peliin luotiin C#-kooditiedosto, joka yhdistettiin Inspector-näkymässä tähän peliobjektiin. Kooditiedosto sai nimekseen: "ServerLoader.cs".



Kuva 22. ServerLoader peliobjekti ja siihen liitetty kooditiedosto ServerLoader.cs

Peliin luotiin kaksi C#-apuluokkaa, jotta data saataisiin palvelimelta tuotua ha-luttuun muotoon Unity-projektiin. Luokat tehtiin uuteen C#-tiedostoon nimeltään: "ServerClasses.cs". Tiedostossa on määritelty luokat yhdelle peliasetukselle, joka sisältää string-tyyppiset arvot: presetname ja blocks, int-tyyppiset arvot: size, duration, frequency ja hits, sekä bool-tyyppisen arvon laser. Näiden merkitykset ovat vastaavat kuin JSON-tiedoston määrittelyssä (taulukko 3.). Toinen luotu apuluokka on nimeltään PresetList, joka sisältää Preset luokan olioita (Kuva 23.). Luokat on määritelty System.Serializable- määrittelyllä. Määrittely mahdollistaa parametrien näkymisen Unityn inspector-näkymässä ja niiden käytön pelissä. (Unity Documentation. 2022a)



```

1
2   [System.Serializable]
3
4   1 reference
5   public class Preset
6   {
7       public string presetname;
8       public string blocks;
9       public int size;
10      public int duration;
11      public int frequency;
12      public int hits;
13      public bool laser;
14  }
15
16  [System.Serializable]
17
18  3 references
19  public class PresetList
20  {
21      public Preset[] presets;

```

Kuva 23. Luodut C#-luokat

ServerLoader.cs (Kuva 22.) tiedostossa on määritelty julkiset muuttujat PresetList-tyyppinen muuttuja list, string-tyyppinen muuttuja url ja int-tyyppinen muuttuja index. PresetList-muuttujaan tallennetaan palvelimelta tuleva data. Muuttujaan url asetetaan Unityn Inspector-ikkunassa palvelimella sijaitsevan JSON-tiedoston URL-osoite. Muuttujaa index käytetään kertomaan, mikä PresetList-muuttujan asetuksista on aktiivisena.

Unityn kooditiedostoissa metodi "void Start()" on oletusmetodi, joka suoritetaan aina kun ohjelma käynnistetään, mikäli kooditiedosto on yhdistetty johonkin peliobjektiin. ServerLoader.cs- koodissa Start()- metodissa suoritetaan StartCoroutine- metodi, jonka parametrinä on GetText-metodi ja sen suoritusparametrinä on JSON-tiedoston osoitteen sisältävä url-muuttuja.

StartCoroutine metodilla voidaan sen parametrimetodin suorittamista rytmittää. Tässä koodissa sen tarkoitus on varmistaa, että palvelin ehtii vastaamaan ennen kuin metodi suoritetaan loppuun. (Unity Documentation. 2022b)

Muuttuja list luodaan UnityEngine-luokkakirjastoon sisältyvän JsonUtility-luokan FromJson-metodilla, joka muuttaa palvelimelta tulevan JSON-muotoisen

datan C#-kielen muotoiseksi objektiksi. Koodissa (Kuva 24.) näkyvät vihreänväriset rivit ovat kommentoitu pois, koska niiden sisältämä koodi ei toimi sovelluksessa käytetyssä vanhemmassa Unityn versiossa. Uudempia versioita käyttäessä voidaan kommentin merkinä oleva "//"-merkintä poistaa. Muutoksella saadaan virheilmoitus console-näkymään, mikäli palvelimen tietoja ei syystä tai toisesta saada ladattua.

```

1  using System.Collections;
2  using UnityEditor.Experimental.SceneManagement;
3  using UnityEditor.Presets;
4  using UnityEngine;
5  using UnityEngine.Networking;
6
7  @ Unity Script | 36 references
8  public class ServerLoader : MonoBehaviour
9  {
10     public PresetList list = new PresetList();
11     public string url;
12     public int index;
13     @ Unity Message | 0 references
14     void Start()
15     {
16         StartCoroutine(GetText(url));
17     }
18     @ Unity Message | 0 references
19     void Update()
20     {
21     }
22     1 reference
23     IEnumerator GetText(string _url)
24     {
25         UnityWebRequest www = UnityWebRequest.Get(_url);
26
27         yield return www.SendWebRequest();
28
29         //if (www.result != UnityWebRequest.Result.Success)
30         //{
31             Debug.Log(www.error);
32         //}
33         //else
34         //{
35             Debug.Log(www.downloadHandler.text);
36             list = JsonUtility.FromJson<PresetList>(www.downloadHandler.text);
37         //}
38     }
39 }

```

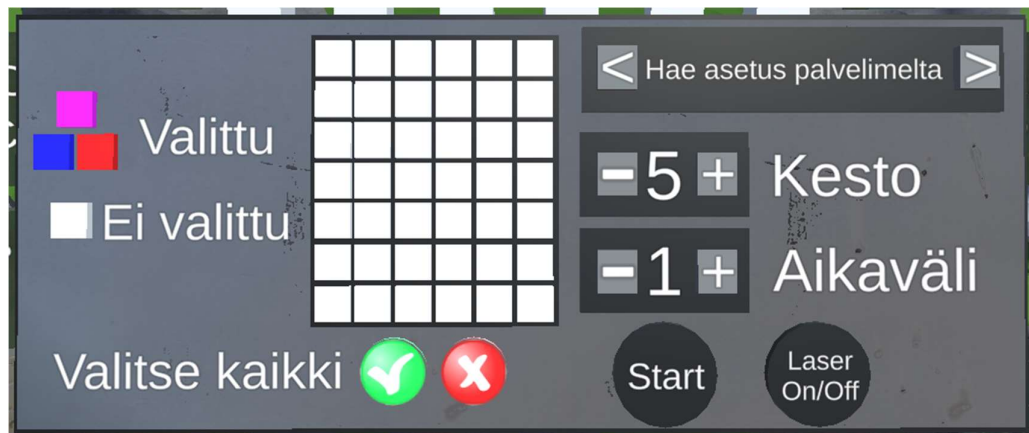
Kuva 24. ServerLoader.cs- kooditiedosto

Koodin suorittamisen tuloksena saadaan ServerLoader-peliobjektin inspector-näkymään ServerLoader.cs- objektin parametreihin ladattua palvelimen JSON-tiedoston sisältö (Kuva 25.). Nämä parametrit ovat tämän jälkeen Unityssä käytettävissä.



### 8.3 Parametrien sijoittaminen Whack-a-fysion käyttöliittymään

Serveriasetusten hakemiseksi oli pelin käyttöliittymään luotava palvelimelta tietojen hakemiseksi uusi kahden painikkeen paneeli (Kuva 26.), josta asetuksia voi selata. Paneeliin laitettiin kaksi painiketta, nuolet oikealle ja vasemmalle. Pelin logiikka toimii niin, että pelaajan käsissä olevilla pampuilla painikkeisiin koskemalla voidaan käyttöliittymän elementtejä aktivoida. Vasemmalla painikkeella selataan listan edellinen asetus ja vastaavasti oikealla painikkeella seuraava. Ennen kuin palvelinasetuksia selataan, on painikkeiden välissä väliaikainen teksti, joka kertoo, että painikkeella voidaan hakea asetuksia palvelimelta. Selattaessa listaa, muuttujan index arvo muuttuu ja haetun asetuksen nimi näkyy tekstinä. Lisäksi käyttöliittymän elementit muuttuvat asetuksen mukaiseksi (Kuva 27.)



Kuva 26. Pelin 1. säädin muutoksen jälkeen.



Kuva 27. Palvelimelta ladatut muutokset pelin molemmissa säätimissä

Muutokset on toteutettu pelissä kooditiedostossa NextIndex.cs (Liite 18.). Koodi on liitetty Unityssä vasemman -ja oikeaan painikeobjektiin inspector-nä-

kymässä. Palvelimelta tuleviin arvoihin voidaan viitata etsimällä projektista peliobjekti: "ServerLoader" ja siitä komponentti: "ServerLoader". Edellämainitusta komponentista saadaan parametrin haettua. Esimerkiksi haluttaessa hakea parametri size palvelimen kolmannesta asetuksesta, (indeksointi alkaa luvusta 0), se haetaan koodilla: "GameObject.Find("ServerLoader").GetComponent<ServerLoader>().list.presets[2].size". Koodi toimii niin, että vaihdettaessa palvelimelta tulevaa asetusta uuteen, vaihdetaan tarvittavat parametrit Unityn objekteihin.

## 9 JATKOKEHITYSAJATUKSIA

Tässä sovelluskokonaisuudessa VR-sovellus ottaa vain vastaan tietoja palvelimelta. Saattaisi olla kuitenkin tarve myös sille, että vastavuoroisesti palvelimelle lähetettäisiin tietoja itse pelistä. Tämän työn aiheena ollut peli on tehty kuntoutuksen tarpeisiin. Voisi olla mielekästä esimerkiksi, jos peli keräisi kuntoutujan liikkeistä dataa kuntoutuksen ammattilaisten käyttöön. Ohjaaja voisi tätä dataa hyödyntämällä suunnitella kuntoutujalle harjoitteita ja seurata kuntoutuksen edistymistä.

Mikäli edellä mainitun kaltainen järjestelmä toteutettaisiin ja näin ollen palvelimella olevan -ja sinne päin liikkuvan datan määrä merkittävästi kasvaisi, tulisi miettiä jotakin muuta palvelinratkaisua kuten ihan oikean tietokannan luontia JSON-tiedoston sijaan.

Itse etäohjaussovellus ei graafisessa mielessä ole kovin näyttävä, joten jos sovelluksen kehittämiseen olisi enemmän aikaa muokkaisin sen ulkonäköä. Lisäksi jotkut käytetyistä HTML-elementeistä toimivat huonosti kosketusnäytön kanssa, joten niiden osalta ohjelmassa olisi vielä hiottavaa. Ohjelman käyttöliittymän skaalautuvuuteen eri näyttökokoihin tulisi myös paneutua enemmän.

## 10 YHTEENVETO

Työssä oli tarkoitus luoda sovellus, jolla voidaan VR-lasien ulkopuolelta säätää, joitakin VR-peliin liittyviä parametrejä ja ladata näin lasien käyttäjälle peliasetuksia ohjelmasta. Lisäksi työn on tarkoitus esitellä konsepti, jolla VR-peleihin voidaan etäohjainsovellus luoda. Mielestäni työ vastaa edellä mainittuihin tavoitteisiin.

Tehdessäni työtä sain syvennettyä entisestään tietämystäni JavaScript-koodin kirjoittamisesta ja pelinkehittämisestä Unityllä. Lisäksi oli opettavaista tehdä muutoksia jonkun jo valmiiksi tekemään peliprojektiin ja lukea jonkun muun kirjoittamaa koodia.

## LÄHTEET

Apache. (2022) [Pääsivu]. Haettu 13.12.2022 osoitteesta <https://httpd.apache.org/>

Figma (n.d.). Haettu 4.12.2022 osoitteesta <https://www.figma.com/>

Heiskanen T, Hiekkala SH, Kaitaro T, Naamanka J, Salminen A-L, Stenberg J-H, Virtanen T, Vuononvirta T. (2016). Etäkuntoutus. Kela. Haettu 16.11.2022 osoitteesta [https://www.researchgate.net/publication/304797001\\_Etakuntoutus](https://www.researchgate.net/publication/304797001_Etakuntoutus)

Holappa, N, Merilampi, S. (2022). Teknologia työkaverina: yhteisiä kokemuksia hyvinvointi- ja terveysteknologioista Satakunta DigiHealth -ekosysteemissä. Haettu 13.12.2022 osoitteesta <https://satakuntadigihealth.fi/2021/03/03/kokemuksia-virtuaalitodellisuudesta-satakuntalaisessa-kuntoutuksessa/>

Json. (n.d.). Introducing JSON. Haettu 4.12.2022 osoitteesta <https://www.json.org/json-en.html>

MDN Web Docs. (2022). An overview of HTTP. Haettu 13.12.2022 osoitteesta <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

Meltzer, R. (2020). What is JavaScript used for? Haettu 4.12.2022 osoitteesta <https://www.lighthouse labs.ca/en/blog/what-is-javascript-used-for>

Merilampi, S. (2021). Kokemuksia virtuaalitodellisuudesta satakuntalaisessa kuntoutuksessa. Haettu 30.10.2022 osoitteesta <https://satakuntadigihealth.fi/2021/03/03/kokemuksia-virtuaalitodellisuudesta-satakuntalaisessa-kuntoutuksessa/>

Pontikis, C. (2013). Five Reasons to use Debian as a Server. Haettu 13.12.2022 osoitteesta <https://www.pontikis.net/blog/five-reasons-to-use-debian-as-a-server>

Pot-Kolder R, Veling W, Garaets C, Lokkerbol J, Smit F, Jongeneel A, Ising H, Van der Gaag M. (2020). Cost-Effectiveness of Virtual Reality Cognitive Behavioral Therapy for Psychosis: Health-Economic Evaluation Within a Randomized Controlled Trial. Haettu 9.1.2023 osoitteesta <https://www.jmir.org/2020/5/e17098/>

Sheldon, R. (n.d.). Definition C# (C-Sharp). Haettu 6.12.2022 osoitteesta <https://www.techtarget.com/whatis/definition/C-Sharp>

Squarespace. (n.d.). What is JSON? Haettu 4.12.2022 osoitteesta <https://developers.squarespace.com/what-is-json>

Tutorialspoint. (n.d.). CSS Tutorial. Haettu 4.12.2022 osoitteesta <https://www.tutorialspoint.com/css/index.htm>



Unity. (2022.) [Pääsivu]. Haettu 6.12.2022 osoitteesta <https://unity.com/>

Unity Documentation. (2022a). Serializable. Haettu 6.12.2022 osoitteesta <https://docs.unity3d.com/ScriptReference/Serializable.html>

Unity Documentation. (2022b). MonoBehaviour.StartCoroutine. Haettu 6.12.2022 osoitteesta <https://docs.unity3d.com/ScriptReference/MonoBehaviour.StartCoroutine.html>

Virtuaalikuntoutusverkosto (2022). Esittely. Haettu 14.11.2022 osoitteesta <https://projects.tuni.fi/virtuaalikuntoutusverkosto/esittely/>

php. (n.d.). What is PHP? Haettu 4.12.2022 osoitteesta <https://www.php.net/manual/en/intro-what-is.php>

w3schools. (n.d.,a). JavaScript HTML DOM. Haettu 16.11.2022 osoitteesta [https://www.w3schools.com/js/js\\_htmlDOM.asp](https://www.w3schools.com/js/js_htmlDOM.asp)

w3schools. (n.d.,b). HTML Tutorial. Haettu 4.12.2022 osoitteesta <https://www.w3schools.com/html/>

## LIITE 1: INDEX.HTML

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8"/>
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6
7   <link rel="stylesheet" href="./css/style.css"/>
8   <link rel="stylesheet" href="./css/sliderstyle.css"/>
9   <link rel="stylesheet" href="./css/numberstyle.css"/>
10  <link href="./js/createMainDivs.js"/>
11  <link href="./js/createSecondaryDivs.js"/>
12  <link href="./js/createThirdDivs.js"/>
13  <link href="./js/createFourthDivs.js"/>
14  <link href="./js/numberInput.js"/>
15  <link href="./js/buttonslogik.js"/>
16  <link href="./js/createSaveObject.js"/>
17  <link href="./js/fetchFromJson.js"/>
18  <link href="./js/windowresize.js"/>
19
20
21   <title>Whack a Fysio</title>
22 </head>
23 <body onresize="windowResize()" onfullscreenchange="windowResize()">
24 <script src="./js/createMainDivs.js"></script>
25 <script src="./js/createSecondaryDivs.js"></script>
26 <script src="./js/createThirdDivs.js"></script>
27 <script src="./js/createFourthDivs.js"></script>
28 <script src="./js/numberInput.js"></script>
29 <script src="./js/buttonslogik.js"></script>
30 <script src="./js/createSaveObject.js"></script>
31 <script src="./js/fetchFromJson.js"></script>
32 <script src="./js/windowresize.js"></script>
33 </body>
34 </html>
```

## LIITE 2: STYLE.CSS

```
1  * {
2    margin: 0px;
3  }
4
5  body {
6    background-color: #50e8ff;
7  }
8
9  .leftCenterDiv {
10   justify-content: center;
11   display: flex;
12 }
13
14 /*div {
15   border: 1px solid black;
16 }*/
17
18 .leftCenterDivs {
19   display: flex;
20   justify-content: center;
21   align-items: center;
22   flex-direction: column;
23 }
24
25
26 .leftCenterInnerDivs {
27   background-color: rgb(255, 255, 255);
28   border: 1px solid black;
29 }
30
31 .leftUpperDivs, .leftLowerDivs {
32   align-content: center;
33   align-items: center;
34   justify-content: center;
35   display: flex;
36 }
37
38
39 .colorDivs {
40   border: 3px solid rgba(129, 129, 129, 0.555);
41   border-radius: 15px;
42   display: flex;
43   justify-content: center;
44   align-items: center;
45   cursor: pointer;
46 }
47
48 .sliderContainer {
49   display: flex;
50   justify-content: center;
51   align-items: center;
52   flex-direction: column;
53 }
54
55 .dropDownContainer {
56   width: 100%;
57   height: 100%;
58   display: flex;
59   justify-content: center;
60   align-items: center;
61   flex-direction: column;
62 }
63
64 .dropDown {
65   width: 80%;
66   height: 50%;
67   font-size: 30px;
68   border-radius: 15px;
69   text-align: center;
70 }
71
```

```
72 .dropDown select {
73     width: 100%;
74     height: 100%;
75     font-size: 30px;
76     text-align: center;
77 }
78
79
80 .NumberDivs, .HeaderDivs {
81     width: 100%;
82     height: 100%;
83     display: flex;
84     align-items: center;
85 }
86
87 .NumberDivs {
88     display: flex;
89     justify-content: left;
90     align-items: center;
91     /*min-height: 100vh;*/
92 }
93 .HeaderDivs {
94     justify-content: right;
95     font-size: 30px;
96 }
97
98 .rightCenterDivs {
99     display: grid;
100    gap: 10px;
101    grid-template-columns: 1fr 1fr;
102 }
103
104 .rightLowerDivs {
105     display: flex;
106     justify-content: center;
107     align-items: center;
108     flex-direction: column;
109 }
110
111 .rightButton {
112     display: flex;
113     justify-content: center;
114     align-items: center;
115     flex-direction: column;
116     border-radius: 100%;
117     border: 3px solid rgba(255, 255, 255, 0.733);
118     background-color: rgba(0, 0, 0, 0.548);
119     font-size: large;
120     cursor: pointer;
121 }
122
123 .rightButton:hover {
124     background-color: rgba(0, 0, 0, 0.9);
125     border-color: rgba(255, 255, 255, 0.9);
126     color: white;
127 }
128
```

```
129 .savePresetDiv, .deletePresetDiv {
130   position: fixed;
131   top: 50%;
132   left: 50%;
133   border: 4px solid black;
134   background-color: white;
135   z-index: 10;
136   height: 60vh;
137   width: 60%;
138   margin-left: calc(60% / -2);
139   margin-top: calc(60vh / -2);
140   border-radius: 20px;
141   background-color: rgba(141, 0, 0, 0.9);
142   justify-content: center;|
143   align-items: center;
144   flex-direction: column;
145   display: none;
146 }
147
148 #savePresetName {
149   width: 50%;
150   height: 10%;
151   font-size: 130%;
152   display: flex;
153   justify-content: center;
154   align-items: center;
155   flex-direction: column;
156   margin: 3%;
157 }
158
159 h1 {
160   width: 50%;
161   height: 20%;
162   text-align: center;
163 }
164
165 .FormButton {
166   width: 30%;
167   height: 20%;
168   border-radius: 15px;
169   background-color: rgba(0, 0, 0, 0.548);
170   border: 1px solid rgba(255, 255, 255, 0.733);
171   color: white;
172   margin-bottom: 5%;
173   display: flex;
174   justify-content: center;
175   align-items: center;
176   flex-direction: column;
177 }
178
179 .FormButton:hover {
180   background-color: rgba(0, 0, 0, 0.9);
181   border-color: rgba(255, 255, 255, 0.9);
182   color: white;
183 }
184
185 #savePresetSubmitButton {
186   float: left;
187 }
188 #CancelButton1, #CancelButton2 {
189   float: right;
190 }
```

## LIITE 3: SLIDERSTYLE.CSS

```
1  input[type="range"] {
2    display: block;
3    -webkit-appearance: none;
4    background-color: #153347;
5    width: 80%;
6    height: 5px;
7    border-radius: 5px;
8    margin: 0 auto;
9    outline: 0;
10 }
11 input[type="range"]::-webkit-slider-thumb {
12   -webkit-appearance: none;
13   background-color: #3f0c06;
14   width: 30px;
15   height: 30px;
16   border-radius: 50%;
17   border: 2px solid white;
18   cursor: pointer;
19   transition: .3s ease-in-out;
20 }
21 input[type="range"]::-webkit-slider-thumb:hover {
22   background-color: white;
23   border: 2px solid #3f0c06;
24 }
25 input[type="range"]::-webkit-slider-thumb:active {
26   transform: scale(1.6);
27 }
```

## LIITE 4: NUMBERSTYLE.CSS

```

1  .container {
2      position: relative;
3      width: 100px;
4      height: 50px;
5      border-radius: 40px;
6      border: 2px solid rgba(0, 0, 0, 0.233);
7      background-color: rgba(0, 0, 0, 0.233);
8      margin-left: 70px;
9  }
10
11 .container:hover {
12     width: 180px;
13     background-color: rgba(0, 0, 0, 0.562);
14     border: 2px solid rgba(77, 77, 77, 1);
15     margin-left: 30px;
16 }
17
18 .container .next {
19     position: absolute;
20     top: 50%;
21     right: 30px;
22     width: 15px;
23     height: 15px;
24     border-top: 3px solid rgb(255, 255, 255);
25     border-left: 3px solid rgb(255, 255, 255);
26     border-radius: 9px 0 0 0;
27     z-index: 1;
28     transform: translateY(-50%) rotate(135deg);
29     cursor: pointer;
30     opacity: 0;
31     transition: 0.5s;
32 }
33
34 .container:hover .next {
35     opacity: 1;
36     right: 20px;
37 }
38
39 .container .prev {
40     position: absolute;
41     top: 50%;
42     left: 30px;
43     width: 15px;
44     height: 15px;
45     border-top: 3px solid rgb(255, 255, 255);
46     border-left: 3px solid rgb(255, 255, 255);
47     border-radius: 9px 0 0 0;
48     z-index: 1;
49     transform: translateY(-50%) rotate(315deg);
50     cursor: pointer;
51     opacity: 0;
52     transition: 0.5s;
53 }
54
55 .container:hover .prev {
56     opacity: 1;
57     left: 20px;
58 }
59
60 #box0 span, #box1 span, #box2 span {
61     position: absolute;
62     display: block;
63     width: 100%;
64     height: 100%;
65     text-align: center;
66     line-height: 58px;
67     display: none;
68     color: #00deff;
69     font-size: 32px;
70     font-weight: 700;
71     user-select: none;
72 }
73
74 #box0 span:nth-child(1), #box1 span:nth-child(1), #box2 span:nth-child(1) {
75     display: initial;
76 }

```

## LIITE 5: CREATEMAINDIVS.JS

```
1   let mainDiv = document.createElement("div");
2   let mainDivMargin = 20;
3   mainDiv.id = "mainDiv";
4   mainDiv.className = "mainDiv";
5   //mainDiv.style.width = window.innerWidth + "px";
6   mainDiv.style.display = "grid";
7   mainDiv.style.margin = mainDivMargin + "px";
8   mainDiv.style.gridTemplateColumns = "4fr 5fr";
9   document.body.appendChild(mainDiv);
10
11  let leftDiv = document.createElement("div");
12  leftDiv.id = "leftDiv";
13  leftDiv.className = "leftDiv";
14  //leftDiv.style.width = window.innerWidth / 2 + "px";
15  mainDiv.appendChild(leftDiv);
16
17  /*let centerDiv = document.createElement("div");
18  centerDiv.id = "centerDiv";
19  centerDiv.className = "centerDiv";
20  //centerDiv.style.width = window.innerWidth / 2 + "px";
21  centerDiv.style.backgroundColor = "green";
22  mainDiv.appendChild(centerDiv);*/
23
24  let rightDiv = document.createElement("div");
25  rightDiv.id = "rightDiv";
26  rightDiv.className = "rightDiv";
27  //rightDiv.style.width = window.innerWidth / 2 + "px";
28  mainDiv.appendChild(rightDiv);
```



## LIITE 6: CREATESECONDARYDIVS.JS

```
1 leftDiv.style.display = "grid";
2 leftDiv.style.gridTemplateRows = "2fr 9fr 1fr 1fr";
3
4 let leftUpperDiv = document.createElement("div");
5 leftUpperDiv.id = "leftUpperDiv";
6 leftUpperDiv.className = "leftUpperDiv";
7 leftUpperDiv.setAttribute("active_color", "white");
8 leftDiv.appendChild(leftUpperDiv);
9
10 let leftCenterDiv1 = document.createElement("div");
11 leftCenterDiv1.id = "leftCenterDiv1";
12 leftCenterDiv1.className = "leftCenterDiv";
13 leftDiv.appendChild(leftCenterDiv1);
14
15 let leftCenterDiv2 = document.createElement("div");
16 leftCenterDiv2.id = "leftCenterDiv2";
17 leftCenterDiv2.className = "leftCenterDiv";
18 leftDiv.appendChild(leftCenterDiv2);
19
20 let leftLowerDiv = document.createElement("div");
21 leftLowerDiv.id = "leftLowerDiv";
22 leftLowerDiv.className = "leftLowerDiv";
23 leftDiv.appendChild(leftLowerDiv);
24
25 rightDiv.style.display = "grid";
26 rightDiv.style.gridTemplateRows = "2fr 7fr 2fr";
27
28 let rightUpperDiv = document.createElement("div");
29 rightUpperDiv.id = "rightUpperDiv";
30 rightUpperDiv.className = "rightUpperDiv";
31 rightDiv.appendChild(rightUpperDiv);
32
33 let rightCenterDiv = document.createElement("div");
34 rightCenterDiv.id = "rightCenterDiv";
35 rightCenterDiv.className = "rightCenterDiv";
36 rightDiv.appendChild(rightCenterDiv);
37
38 let rightLowerDiv = document.createElement("div");
39 rightLowerDiv.id = "rightLowerDiv";
40 rightLowerDiv.className = "rightLowerDiv";
41 rightDiv.appendChild(rightLowerDiv);
```

## LIITE 7: SLIDER.JS

```
1 var elem = document.querySelector('input[type="range"]');
2
3 var rangeValue = function(){
4     var newValue = elem.value;
5     var target = document.querySelector('.value');
6     target.innerHTML = newValue;
7 }
8
9 elem.addEventListener("input", rangeValue);
```

## LIITE 8: CREATETHIRDDIVS.JS

```

1 leftUpperDiv.style.display = "grid";
2 leftUpperDiv.style.gridTemplateColumns = "1fr 1fr";
3 leftUpperDiv.style.gridTemplateRows = "1fr 1fr";
4
5 for (let i = 0; i < 4; i++) {
6   let leftUpperDivs = document.createElement("div");
7   leftUpperDivs.id = "leftUpperDivs" + i;
8   leftUpperDivs.className = "leftUpperDivs";
9   leftUpperDiv.appendChild(leftUpperDivs);
10 }
11
12 let leftCenterInnerDiv = document.createElement("div");
13 leftCenterInnerDiv.style.display = "grid";
14 leftCenterInnerDiv.style.gridTemplateColumns = "1fr 1fr 1fr 1fr 1fr 1fr";
15 leftCenterInnerDiv.style.gridTemplateRows = "1fr 1fr 1fr 1fr 1fr 1fr";
16 leftCenterInnerDiv.id = "leftCenterInnerDiv";
17 leftCenterInnerDiv.className = "leftCenterInnerDiv";
18 leftCenterInnerDiv.style.height = "100%";
19 leftCenterInnerDiv.style.width = (6/7)*leftCenterDiv1.offsetHeight + "px";
20 leftCenterInnerDiv.style.margin = "auto";
21 leftCenterDiv1.appendChild(leftCenterInnerDiv);
22
23 for (let i = 0; i < 42; i++) {
24   let leftCenterDivs = document.createElement("div");
25   leftCenterDivs.id = "leftCenterDivs" + i;
26   leftCenterDivs.setAttribute("index", i);
27   leftCenterDivs.className = "leftCenterDivs";
28   leftCenterInnerDiv.appendChild(leftCenterDivs);
29   let leftCenterInnerDivs = document.createElement("div");
30   leftCenterInnerDivs.id = "leftCenterInnerDivs" + i;
31   leftCenterInnerDivs.className = "leftCenterInnerDivs";
32   leftCenterDivs.appendChild(leftCenterInnerDivs);
33 }
34
35 leftLowerDiv.style.display = "grid";
36 leftLowerDiv.style.gridTemplateColumns = "1fr 1fr";
37
38 for (let i = 0; i < 2; i++) {
39   let leftLowerDivs = document.createElement("div");
40   leftLowerDivs.id = "leftLowerDivs" + i;
41   leftLowerDivs.className = "leftLowerDivs";
42   leftLowerDiv.appendChild(leftLowerDivs);
43 }
44
45 rightCenterDiv.style.display = "grid";
46 rightCenterDiv.style.gridTemplateRows = "1fr 1fr 1fr";
47
48 for (let i = 0; i < 3; i++) {
49   let rightCenterDivs = document.createElement("div");
50   rightCenterDivs.id = "rightCenterDivs" + i;
51   rightCenterDivs.className = "rightCenterDivs";
52   rightCenterDiv.appendChild(rightCenterDivs);
53 }
54
55 rightLowerDiv.style.display = "grid";
56 rightLowerDiv.style.gridTemplateColumns = "1fr 1fr 1fr 1fr";
57
58 for (let i = 0; i < 4; i++) {
59   let rightLowerDivs = document.createElement("div");
60   rightLowerDivs.id = "rightLowerDivs" + i;
61   rightLowerDivs.className = "rightLowerDivs";
62   rightLowerDiv.appendChild(rightLowerDivs);
63 }

```

## LIITE 9: CREATEFOURTHDIVS.JS

```

1  //The Divs for upper left colour changers
2
3  const colorDivColors = [ "#fd012a", "#f00fe", "#012fff", "#ffff" ];
4  const colorDivText = [ "Oikea", "Molemmat", "Vasen", "Ei valittu" ];
5  for (let i = 0; i < 4; i++) {
6    let colordiv = document.createElement("div");
7    colordiv.id = "colorDivs" + i;
8    colordiv.className = "colorDivs";
9    colordiv.style.backgroundColor = colorDivColors[i];
10   colordiv.innerHTML = colorDivText[i];
11   colordiv.style.height = "80%";
12   colordiv.style.width = "80%";
13   document.getElementById("leftUpperDivs" + i).appendChild(colordiv);
14 }
15
16 //Slider to change block size
17
18 let sliderContainer = document.createElement("div");
19 sliderContainer.id = "sliderContainer";
20 sliderContainer.className = "sliderContainer";
21 sliderContainer.style.height = "100%";
22 sliderContainer.style.width = "100%";
23 sliderContainer.style.margin = "auto";
24 document.getElementById("leftCenterDiv2").appendChild(sliderContainer);
25 let slider = document.createElement("input");
26 slider.id = "slider";
27 slider.className = "slider";
28 slider.type = "range";
29 slider.min = "0";
30 slider.max = "100";
31 slider.value = "50";
32 slider.step = "25";
33 sliderContainer.appendChild(slider);
34 sliderValue();
35 slider.oninput = function() {
36   sliderValue();
37 }
38
39
40 // Left bottom buttons to choose all or empty blocks
41
42 const bottomDivColors = [ "red", "#ffff" ];
43 const bottomDivText = [ "Valitse Kaikki", "Tyhjennä Valinnat" ];
44 for (let i = 0; i < 2; i++) {
45   let bottomDiv = document.createElement("div");
46   bottomDiv.id = "bottomDivs" + i;
47   bottomDiv.className = "colorDivs";
48   bottomDiv.style.backgroundColor = bottomDivColors[i];
49   bottomDiv.innerHTML = bottomDivText[i];
50   bottomDiv.style.height = "80%";
51   bottomDiv.style.width = "80%";
52   document.getElementById("leftLowerDivs" + i).appendChild(bottomDiv);
53 }
54
55 //Dropdown menu for presets
56
57 let dropDownContainer = document.createElement("div");
58 dropDownContainer.id = "dropDownContainer";
59 dropDownContainer.className = "dropDownContainer";
60 let dropDown = document.createElement("select");
61 dropDown.setAttribute("selectedIndex", -1);
62 dropDown.setAttribute("required", "");
63 dropDown.setAttribute("onclick", "DropDownSelect(dropDown.options.selectedIndex)");
64 dropDown.id = "dropDown";
65 dropDown.className = "dropDown";
66 dropDownContainer.appendChild(dropDown);
67 document.getElementById("rightUpperDiv").appendChild(dropDownContainer);
68

```

```

69 // Number elements
70
71
72 let headerTexts = [ "Kesto", "Aikaväli", "Felin kesto osumina"];
73
74 for (let i = 0; i < 3; i++) {
75   let headerDiv = document.createElement("div");
76   headerDiv.id = "HeaderDivs" + i;
77   headerDiv.className = "HeaderDivs";
78   headerDiv.innerHTML = headerTexts[i];
79   document.getElementById("rightCenterDivs" + i).appendChild(headerDiv);
80   let numberDiv = document.createElement("div");
81   numberDiv.id = "NumberDivs" + i;
82   numberDiv.className = "NumberDivs";
83   numberDiv.innerHTML =
84     `<div class="container"><span class="next" onclick="nextNum(nums[' + i + '], indexes[' + i + '], ' + i + ')"></span><span class="prev"
85     onclick="prevNum(nums[' + i + '], indexes[' + i + '], ' + i + ')"></span><div id="box" + i + "*"></div></div>`;
86   document.getElementById("rightCenterDivs" + i).appendChild(numberDiv);
87
88 // Bottom right buttons
89
90 let buttonTexts = [ "Laser <br> On/Off", "Tallenna <br> Nimellä", "Tallenna", "Poista" ];
91
92 for (let i = 0; i < 4; i++) {
93   let rightButton = document.createElement("div");
94   rightButton.id = "rightButton" + i;
95   rightButton.className = "rightButton";
96   rightButton.innerHTML = buttonTexts[i];
97   document.getElementById("rightLowerDivs" + i).appendChild(rightButton);
98   rightButton.style.height = "100px";
99   rightButton.style.width = rightButton.offsetHeight + "px";
100
101 }
102
103 function sliderValue() {
104   let list = document.getElementsByClassName("leftCenterInnerDivs");
105   for (let i = 0; i < list.length; i++) {
106     list[i].style.height = 20 + 0.6*slider.value + "%";
107     list[i].style.width = 20 + 0.6*slider.value + "%";
108   }
109 }
110
111 let rightButtons = document.getElementsByClassName("rightButton");
112
113 let savePresetDiv = document.createElement("div");
114 savePresetDiv.id = "savePresetDiv";
115 savePresetDiv.className = "savePresetDiv";
116 savePresetDiv.innerHTML =
117   `<h1>Tallenna asetukset nimellä</h1> +
118   <input type="text" placeholder="Nimi:" id="savePresetName"></input> +
119   <div class="FormButton" id="savePresetSubmitButton">Tallenna</div> +
120   <div class="FormButton" id="cancelButton1">Peruuta</div>`;
121 document.getElementById("mainDiv").appendChild(savePresetDiv);
122
123 let deletePresetDiv = document.createElement("div");
124 deletePresetDiv.id = "deletePresetDiv";
125 deletePresetDiv.className = "deletePresetDiv";
126 deletePresetDiv.innerHTML =
127   `<h1></h1> +
128   <div class="FormButton" id="deletePresetSubmitButton">Kyllä</div> +
129   <div class="FormButton" id="cancelButton2">Peruuta</div>`;
130 document.getElementById("mainDiv").appendChild(deletePresetDiv);
131
132 function CreatePresetList(data) {
133   let placeholder = document.createElement("option");
134   placeholder.innerHTML = "-- Valitse asetukset -- ";
135   placeholder.value = "placeholder";
136   placeholder.selected = true;
137   placeholder.disabled = true;
138   dropdown.appendChild(placeholder);
139   for (let i = 1; i < data.presets.length; i++) {
140     let option = document.createElement("option");
141     option.value = data.presets[i].presetname;
142     option.text = data.presets[i].presetname;
143     option.setAttribute("index", i);
144     dropdown.appendChild(option);
145   }
146 }
147 //dropdown.options.selectedIndex
148 function DropdownSelect(index) {
149   dropdown.setAttribute('selectedIndex', index);
150   PlaceDataToDom(index);
151 }

```

## LIITE 10: NUMBERINPUT.JS

```
1  var indexes = [2, 2, 2];
2
3  nums = [CreateSpans(0, 9, 1, 1), CreateSpans(1, 9, 1, 1), CreateSpans(2, 400, 0, 10)];
4
5  for (let i = 0; i < 3; i++) {
6    let spans = document.getElementById("box" + i).getElementsByTagName('span');
7    document.getElementById("box" + i).setAttribute("value", spans[indexes[i]].innerHTML);
8  }
9
10 function CreateSpans(elementindex, amount, start, step) {
11   var numbers = document.getElementById("box" + elementindex);
12   for (let i = 0; i < amount; i++) {
13     var span = document.createElement("span");
14     span.textContent = i*step + start;
15     numbers.appendChild(span);
16     if(i == indexes[elementindex]) { span.style.display = "initial"; }
17     else { span.style.display = "none"; }
18   }
19   var num = numbers.getElementsByTagName('span');
20   return num;
21 }
22
23
24 function prevNum(num, index, ind) {
25   num[index].style.display = 'none';
26   indexes[ind] = (index - 1 + num.length) % num.length;
27   num[indexes[ind]].style.display = 'Initial';
28   document.getElementById("box" + ind).setAttribute("value", num[indexes[ind]].innerHTML);
29 }
30
31 function nextNum(num, index, ind) {
32   num[index].style.display = 'none';
33   indexes[ind] = (index + 1) % num.length;
34   num[indexes[ind]].style.display = 'Initial';
35   document.getElementById("box" + ind).setAttribute("value", num[indexes[ind]].innerHTML);
36 }
```

## LIITE 11: BUTTONSLOGIK.JS

```

1  const colorDivsList = document.getElementsByClassName("colorDivs");
2  const colors = [ "#fd012a", "#00fe", "#012ff", "#ffff" ];
3  const blockDivList = document.getElementsByClassName("leftCenterDivs");
4  const blockList = document.getElementsByClassName("leftCenterInnerDivs");
5  const rightLowerButtons = document.getElementsByClassName("rightButton");
6  rightLowerButtons[0].setAttribute("on_off", false)
7  ToggleLaser();
8
9  colorDivsList[0].onclick = function() {ChangeColor(0);};
10 colorDivsList[1].onclick = function() {ChangeColor(1);};
11 colorDivsList[2].onclick = function() {ChangeColor(2);};
12 colorDivsList[3].onclick = function() {ChangeColor(3);};
13 colorDivsList[4].onclick = function() {ChooseAll();};
14 colorDivsList[5].onclick = function() {EmptyAll();};
15
16 for (let i = 0; i < 42; i++) {
17   blockDivList[i].onclick = function() {ChooseBlock(i);};
18 }
19
20 rightLowerButtons[0].onclick = function() {ToggleLaser();};
21 rightLowerButtons[1].onclick = function() {ToggleSavePresetByNameBox();};
22 if(dropDown.value == "") { console.log("true"); }
23 rightLowerButtons[2].onclick = function() {
24   if(document.getElementById("dropDown").getAttribute("selectedIndex") > 0) {
25     createSaveObject(dropDown.value, true);
26   }
27 };
28 rightLowerButtons[3].onclick = function() {ToggleDeletePresetBox();};
29
30 document.getElementById("savePresetSubmitButton").onclick = function() {SaveWithName();};
31 document.getElementById("CancelButton1").onclick = function() {CancelSaving();};
32 document.getElementById("deletePresetSubmitButton").onclick = function() {DeletePreset();};
33 document.getElementById("CancelButton2").onclick = function() {CancelSaving();};
34
35 function ChangeColor(col) {
36   document.getElementById("leftUpperDiv").setAttribute("active_color", colors[col]);
37   for (let i = 0; i < 4; i++) {
38     if(i == col) {
39       colorDivsList[i].style.opacity = "1";
40     } else {
41       colorDivsList[i].style.opacity = "0.5";
42     }
43   }
44 }
45
46 function ChooseAll() {
47   for (let i = 0; i < 42; i++) {
48     blockList[i].setAttribute("active_color", document.getElementById("leftUpperDiv").getAttribute("active_color"));
49     blockList[i].style.backgroundColor = document.getElementById("leftUpperDiv").getAttribute("active_color");
50   }
51 }
52
53 function EmptyAll () {
54   for (let i = 0; i < 42; i++) {
55     blockList[i].setAttribute("active_color", colors[3]);
56     blockList[i].style.backgroundColor = colors[3];
57   }
58 }
59
60 function ChooseBlock (block) {
61   blockList[block].setAttribute("active_color", document.getElementById("leftUpperDiv").getAttribute("active_color"));
62   blockList[block].style.backgroundColor = document.getElementById("leftUpperDiv").getAttribute("active_color");
63 }
64
65 function ToggleLaser () {
66   if(rightLowerButtons[0].getAttribute("on_off") == "true") {
67     rightLowerButtons[0].setAttribute("on_off", false);
68     rightLowerButtons[0].style.backgroundColor = "red";
69     rightLowerButtons[0].innerHTML = "Laser OFF";
70   } else {
71     rightLowerButtons[0].setAttribute("on_off", true);
72     rightLowerButtons[0].style.backgroundColor = "green";
73     rightLowerButtons[0].innerHTML = "Laser ON";
74   }
75 }
76
77 function ToggleSavePresetByNameBox () {
78   document.getElementById("savePresetDiv").style.display = "flex";
79 }
80
81 function ToggleDeletePresetBox () {
82   document.getElementById("deletePresetDiv").style.display = "flex";
83   document.getElementById("deletePresetDiv").getElementsByTagName("h1")[0].innerHTML =
84   "Haluatko varmasti poistaa asetuksen: " +
85   document.getElementById("dropDown").value + "?";
86 }
87
88 function SaveWithName() {
89   let existingPresets = document.getElementById("dropDown").getElementsByTagName('option');
90   console.log(existingPresets[0].value);
91   for (let i = 0; i < existingPresets.length; i++) {
92     if(existingPresets[i].value == document.getElementById("savePresetName").value) {
93       alert("Tällä nimellä on jo olemassa asetukset!");
94       return;
95     }
96   }
97   if(document.getElementById("savePresetName").value == "") {
98     alert("Uudelle asetukselle on annettava nimi!");
99   } else {
100    createSaveObject(document.getElementById("savePresetName").value, false);
101  }
102 }
103
104 function CancelSaving() {
105   document.getElementById("savePresetDiv").style.display = "none";
106   document.getElementById("deletePresetDiv").style.display = "none";
107 }

```

## LIITE 12: CREATESAVEOBJECT.JS

```

1  let blocks = "";
2  let size;
3  let duration;
4  let frequency;
5  let hits;
6  let laser;
7
8  function createSaveObject(presetname, save) {
9
10     for (let i = 0; i < 42; i++) {
11         if(document.getElementById("leftCenterInnerDivs" + i).getAttribute("active_color") == colors[0]) {
12             blocks += "r";
13         } else if(document.getElementById("leftCenterInnerDivs" + i).getAttribute("active_color") == colors[1]) {
14             blocks += "b";
15         } else if(document.getElementById("leftCenterInnerDivs" + i).getAttribute("active_color") == colors[2]) {
16             blocks += "l";
17         } else {
18             blocks += "n";
19         }
20     }
21     size = document.getElementById("slider").value;
22     duration = document.getElementById("box0").getAttribute("value");
23     frequency = document.getElementById("box1").getAttribute("value");
24     hits = document.getElementById("box2").getAttribute("value");
25     laser = rightLowerButtons[0].getAttribute("on_off");
26
27     //let objectArray = [presetname, blocks, size, duration, frequency, hits, laser];
28
29     let json = JSON.stringify({ presetname: presetname, blocks: blocks, size: size, duration: duration, frequency: frequency, hits: hits, laser: laser });
30     console.log(json);
31     let index = parseInt(document.getElementById("dropDown").getAttribute("selectedIndex"));
32     console.log(presetname);
33     if(save == true) {
34         window.open('./php/replacejson.php?json=' + json + '&index=' + index, "_self");
35     } else {
36         window.open('./php/savejson.php?json=' + json, "_self");
37     }
38 }
39
40 function DeletePreset() {
41     console.log("toimiva funktio");
42     let index = parseInt(document.getElementById("dropDown").getAttribute("selectedIndex"));
43     window.open('./php/deletepreset.php?index=' + index, "_self");
44 }

```



## LIITE 13: FETCHFROMJSON.JS

```

1  let jsonData = [];
2
3  fetch('../php/save.json?upd=' + Math.random())
4    .then((response) => response.json())
5    .then((data) => SaveData(data));
6
7  function SaveData(data) {
8    console.log(data.presets[0]);
9    for (let i = 0; i < data.presets.length; i++) {
10     console.log(data.presets[i].blocks[0]);
11     jsonData.push(data.presets[i]);
12     console.log(jsonData[i].blocks);
13     if(i == 0) {
14       PlaceDataToDom(i);
15     }
16   }
17   CreatePresetList(data);
18 }
19
20 function PlaceDataToDom(i) {
21   dropdown.value = jsonData[i].presetname;
22   for (let j = 0; j < 42; j++) {
23     if(jsonData[i].blocks[j] == "r") {
24       blockList[j].setAttribute("active_color", colors[0]);
25       blockList[j].style.backgroundColor = colors[0];
26     } else if(jsonData[i].blocks[j] == "b") {
27       blockList[j].setAttribute("active_color", colors[1]);
28       blockList[j].style.backgroundColor = colors[1];
29     } else if(jsonData[i].blocks[j] == "l") {
30       blockList[j].setAttribute("active_color", colors[2]);
31       blockList[j].style.backgroundColor = colors[2];
32     } else {
33       blockList[j].setAttribute("active_color", colors[3]);
34       blockList[j].style.backgroundColor = colors[3];
35     }
36   }
37   document.getElementById("slider").value = jsonData[i].size;
38   sliderValue();
39   document.getElementById("box0").setAttribute("value", jsonData[i].duration);
40   indexes[0] = jsonData[i].duration - 1;
41   SetValueNumberElement(indexes[0], 0)
42   document.getElementById("box1").setAttribute("value", jsonData[i].frequency);
43   indexes[1] = jsonData[i].frequency - 1;
44   SetValueNumberElement(indexes[1], 1)
45   document.getElementById("box2").setAttribute("value", jsonData[i].hits);
46   indexes[2] = jsonData[i].hits/10;
47   SetValueNumberElement(indexes[2], 2)
48   if(jsonData[i].laser == "true") {
49     rightLowerButtons[0].setAttribute("on_off", "true");
50     rightLowerButtons[0].style.backgroundColor = "green";
51   } else {
52     rightLowerButtons[0].setAttribute("on_off", "false");
53     rightLowerButtons[0].style.backgroundColor = "red";
54   }
55 }
56
57 function SetValueNumberElement(forWhatIndex, boxIndex) {
58   let spans = document.getElementById("box" + boxIndex).getElementsByTagName('span');
59   for (let i = 0; i < spans.length; i++) {
60     if (i == forWhatIndex) {
61       spans[i].style.display = "initial";
62     } else {
63       spans[i].style.display = "none";
64     }
65   }
66 }

```

## LIITE 14: WINDOWRESIZE.JS

```
1 windowResize();
2 function windowResize() {
3     var height = window.innerHeight;
4     var width = window.innerWidth;
5     var aspectRatio = height/width;
6
7     if (aspectRatio < 1.5) {
8         mainDiv.style.gridTemplateColumns = "4fr 5fr";
9         mainDiv.style.gridTemplateRows = "1fr";
10    } else {
11        mainDiv.style.gridTemplateColumns = "1fr";
12        mainDiv.style.gridTemplateRows = "1fr 1fr";
13    }
14
15    mainDiv.style.height = (window.innerHeight - 2*mainDivMargin) + "px";
16    leftDiv.style.height = mainDiv.style.height + "px";
17    rightDiv.style.height = mainDiv.style.height + "px";
18
19    if(leftCenterDiv1.offsetHeight/leftCenterDiv1.offsetWidth < 1.16 ) {
20        leftCenterInnerDiv.style.height = "100%";
21        leftCenterInnerDiv.style.width = (6/7)*leftCenterDiv1.offsetHeight + "px";}
22    else {
23        leftCenterInnerDiv.style.width = "100%";
24        leftCenterInnerDiv.style.height = (7/6)*leftCenterDiv1.offsetWidth + "px";
25    }
26
27    let rightbuttons = document.getElementsByClassName("rightButton");
28    if(rightLowerDiv1.offsetHeight < rightLowerDiv1.offsetWidth) {
29        for (let i = 0; i < rightbuttons.length; i++) {
30            rightbuttons[i].style.height = "100%";
31            rightbuttons[i].style.width = rightbuttons[i].offsetHeight + "px";
32        }
33    } else {
34        for (let i = 0; i < rightbuttons.length; i++) {
35            rightbuttons[i].style.width = "100%";
36            rightbuttons[i].style.height = rightbuttons[i].offsetWidth + "px";
37        }
38    }
39 }
```

## LIITE 15: DELETEPRESET.PHP

```
1 |<?php
2
3 $jsonString = file_get_contents('save.json');
4 $indextodelete = $_GET['index'];
5 $data = json_decode($jsonString);
6
7 unset($data->presets[$indextodelete]);
8 $data->presets = array_values($data->presets);
9
10 $newJsonString = json_encode($data, JSON_PRETTY_PRINT);
11 file_put_contents('save.json', $newJsonString);
12 header("Location: ../index.html");
13 ?>
```

## LIITE 16: SAVEJSON.PHP

```
1 <?php
2
3 $jsonString = file_get_contents('save.json');
4 $newPresetJsonString = $_GET['json'];
5
6 $data = json_decode($jsonString);
7 $dataToSave = json_decode($newPresetJsonString);
8 $data->presets[0] = $dataToSave;
9 array_push($data->presets , $dataToSave);
10 $newJsonString = json_encode($data, JSON_PRETTY_PRINT);
11 file_put_contents('save.json', $newJsonString);
12 header("Location: ../index.html");
13 ?>
```

## LIITE 17: REPLACEJSON.PHP

```
1  <?php
2
3  $jsonString = file_get_contents('save.json');
4  $newPresetJsonString = $_GET['json'];
5  $indextoreplace = $_GET['index'];
6  $data = json_decode($jsonString);
7  $dataToSave = json_decode($newPresetJsonString);
8  $data->presets[0] = $dataToSave;
9  $data->presets[$indextoreplace] = $dataToSave;
10 $newJsonString = json_encode($data, JSON_PRETTY_PRINT);
11 file_put_contents('save.json', $newJsonString);
12 header("Location: ../index.html");
13 ?>
```

## LIITE 18: NEXTINDEX.CS

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using TMPro;
5 using System;
6
7 # Unity Script | references
8 public class NextIndex : MonoBehaviour
9 {
10     private GameObject serverloader;
11     public TextMeshPro serverText;
12     private GameObject cubeMatrix;
13     public GameObject sizeCube;
14     private float[] gameMatrix;
15     public int i;
16     public float size;
17     public float size_1;
18     public float size_2;
19     public float size_4;
20     public float size_5;
21     public TextMeshPro duration;
22     public TextMeshPro frequency;
23     public TextMeshPro hits;
24     //private bool firstPressGone = true;
25     // Start is called before the first frame update
26     # Unity Message | references
27     void Start()
28     {
29         serverloader = GameObject.Find("ServerLoader");
30         cubeMatrix = GameObject.Find("Toggle Ruudukko");
31         size = sizeCube.GetComponent<Transform>().localScale.x;
32         size_1 = size * 0.5f;
33         size_2 = size * 0.75f;
34         size_4 = size * 1.25f;
35         size_5 = size * 1.5f;
36     }
37     # Unity Message | references
38     private void OnTriggerEnter(Collider other)
39     {
40         if (other.tag == "sininenpuikko" || other.tag == "punainenpuikko")
41         {
42             if (this.name == "next")
43             {
44                 if (serverloader.GetComponent<ServerLoader>().index == serverloader.GetComponent<ServerLoader>().list.presets.Length - 1)
45                 {
46                     serverloader.GetComponent<ServerLoader>().index = 1;
47                 }
48                 else
49                 {
50                     serverloader.GetComponent<ServerLoader>().index++;
51                 }
52             }
53             else if (this.name == "previous")
54             {
55                 if (serverloader.GetComponent<ServerLoader>().index < 2)
56                 {
57                     serverloader.GetComponent<ServerLoader>().index = serverloader.GetComponent<ServerLoader>().list.presets.Length - 1;
58                 }
59                 else
60                 {
61                     serverloader.GetComponent<ServerLoader>().index--;
62                 }
63             }
64             serverText.text = serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].presetname;
65             int x = 0;
66             foreach (Transform child in cubeMatrix.transform)
67             {
68                 if (serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].blocks[x].Equals('b'))
69                 {
70                     child.GetComponent<togglecube>().PunainenAktiivinen = false;
71                     child.GetComponent<togglecube>().SininenAktiivinen = false;
72                     child.GetComponent<togglecube>().aktiivinen = true;
73                     child.GetComponent<Renderer>().material.color = Color.magenta;
74                 }
75                 else if (serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].blocks[x].Equals('r'))
76                 {
77                     child.GetComponent<togglecube>().PunainenAktiivinen = true;
78                     child.GetComponent<togglecube>().SininenAktiivinen = false;
79                     child.GetComponent<togglecube>().aktiivinen = false;
80                     child.GetComponent<Renderer>().material.color = Color.red;
81                 }
82                 else if (serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].blocks[x].Equals('l'))
83                 {
84                     child.GetComponent<togglecube>().PunainenAktiivinen = false;
85                     child.GetComponent<togglecube>().SininenAktiivinen = true;
86                     child.GetComponent<togglecube>().aktiivinen = false;
87                     child.GetComponent<Renderer>().material.color = Color.blue;
88                 }
89                 else
90                 {
91                     child.GetComponent<togglecube>().PunainenAktiivinen = false;
92                     child.GetComponent<togglecube>().SininenAktiivinen = false;
93                     child.GetComponent<togglecube>().aktiivinen = false;
94                     child.GetComponent<Renderer>().material.color = Color.white;
95                 }
96                 x++;
97             }
98             if (serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].size == 0)
99             {
100                 i = 1;
101             }
102             else if (serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].size == 25)
103             {
104                 i = 2;
105             }
106             else if (serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].size == 50)
107             {
108                 i = 3;
109             }
110             else if (serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].size == 75)
111             {
112                 i = 4;
113             }
114             else { i = 5; }
115             Change_size();
116             duration.text = serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].duration.ToString();
117             GameObject.Find("nextopaneli").GetComponent<Uzeropaneli>().i = serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].duration;
118             frequency.text = serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].frequency.ToString();
119             GameObject.Find("alioopaneli").GetComponent<Uzeropaneli>().i = serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].frequency;
120             hits.text = serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].hits.ToString();
121             Debug.Log(hits.text);
122             GameObject.Find("pelinkestopaneli").GetComponent<Qalinkestopaneli>().i = serverloader.GetComponent<ServerLoader>().list.presets[serverloader.GetComponent<ServerLoader>().index].hits;
123         }
124     }

```

```
123 public void Change_size()
124 {
125     switch (i)
126     {
127         case 1:
128             sizeCube.GetComponent<Transform>().localScale = new Vector3(size_1, size_1, size_1);
129             break;
130         case 2:
131             sizeCube.GetComponent<Transform>().localScale = new Vector3(size_2, size_2, size_2);
132             break;
133         case 3:
134             sizeCube.GetComponent<Transform>().localScale = new Vector3(size, size, size);
135             break;
136         case 4:
137             sizeCube.GetComponent<Transform>().localScale = new Vector3(size_4, size_4, size_4);
138             break;
139         case 5:
140             sizeCube.GetComponent<Transform>().localScale = new Vector3(size_5, size_5, size_5);
141             break;
142     }
143     foreach (Transform child in gameMatrix.transform)
144     {
145         child.GetComponent<cube>().size = i;
146         child.GetComponent<cube>().Change_size();
147     }
148 }
149 }
150 }
151 }
152 }
```