PROJEKTISUUNNITELMA

DOKUMENTIN VERSIOHISTORIA

VERSIONUMERO	PÄIVÄMÄÄRÄ	MUUTOSPERUSTE	TEKIJÄ / HYVÄKSYJÄ
1.10	16.10.2025	Loppukilpailun periaate päivitetty. Tarkennettu projektin tavoitteita ja reunaehtoja, kuten tekoälyn käyttöä.	Olli Himanka
1.02	11.10.2024	Poistettu matematiikan osuuden kuvaus ja lyhennetty pelilogiikka osuuden kuvausta	Kari Jyrkkä
1.01	24.4.2024	Ledit moduuliin lisätty vaatimuksia	Kari Jyrkkä
1.00	19.10.2023	Katselmoitu ohjaavien opettajien toimesta.	Kari Jyrkkä
0.06	13.10.2023	Mannen kilpailuehdotus päivitetty Mannen ehdotuksen mukaiseksi, lähteet lisätty.	Kari Jyrkkä
0.05	10.10.2023	Kilpailullinen osio viimeistelty	Manne Hannula

0.04	23.8.2023	Mannen ehdottamat korjaukset versioon 0.03 tehty.	Kari Jyrkkä
0.03	20.5.2023	Tekstin hiontaa ym., kilpailullinen osio ehdolle kohtaan 2.6.	Manne Hannula
0.02	15.5.2023	Muutoksia vaatimuksiin	Kari Jyrkkä
0.01	9.5.2023	Dokumentin pohja valmis	Kari Jyrkkä

SISÄLLYSLUETTELO

1 TÄMÄN DOKUMENTIN TARKOITUS

OAMK:n tietotekniikan tutkinto-ohjelmassa on tietotekniikan sovellusprojekti -niminen opintojakso, jolla opiskelijat pääsevät käytännössä soveltamaan oppimaansa osaamista ohjelmoinnin, elektroniikan ja digitaalitekniikan parissa.

Tietotekniikan sovellusprojektin laajuus on 9 opintopistettä ja kurssi pitää sisällään 6 opintopisteen laajuisen projektiosuuden ja 3 opintopisteen laajuisen mikrokontrollerin rakenne- ja toimintaperiaate osuuden. Tässä dokumentissa määritellään 6 opintopisteen projektiosuuden tehtävät ja tavoitteet.

Tässä dokumentissa kuvataan Speden Spelit – aiheisen projektin sisällön, joka on oletusarvoisesti kaikkien ryhmien aihe. Jos jokin opiskelijaryhmä kuitenkin haluaa kuitenkin määritellä projektiaiheensa itse, joutuu tuo projektiryhmä itse tekemään tätä dokumenttia vastaavan määrittelyn omasta projektiaiheestaan. Oman projektin tulee olla teknisesti vaativampi kuin tässä dokumentissa määritelty aihe.

Ryhmän on joka tapauksessa päivitettävä tähän dokumenttiin ainakin kappaleet:

- 2.9 Projektiryhmän määrittelemät ylimääräiset tavoitteet
- 3.1 Projektin vaiheistus ja aikataulu

2 PROJEKTIN SISÄLTÖ

Projektissa suunnitellaan elektroninen nopeuspeli. Sovellusprojekti toteutetaan maksimissaan neljän opiskelijan ryhmissä. Jokaiselle ryhmän jäsenelle pitää olla vähintään yksi projektin osa-alue, josta jäsen on päävastuussa.

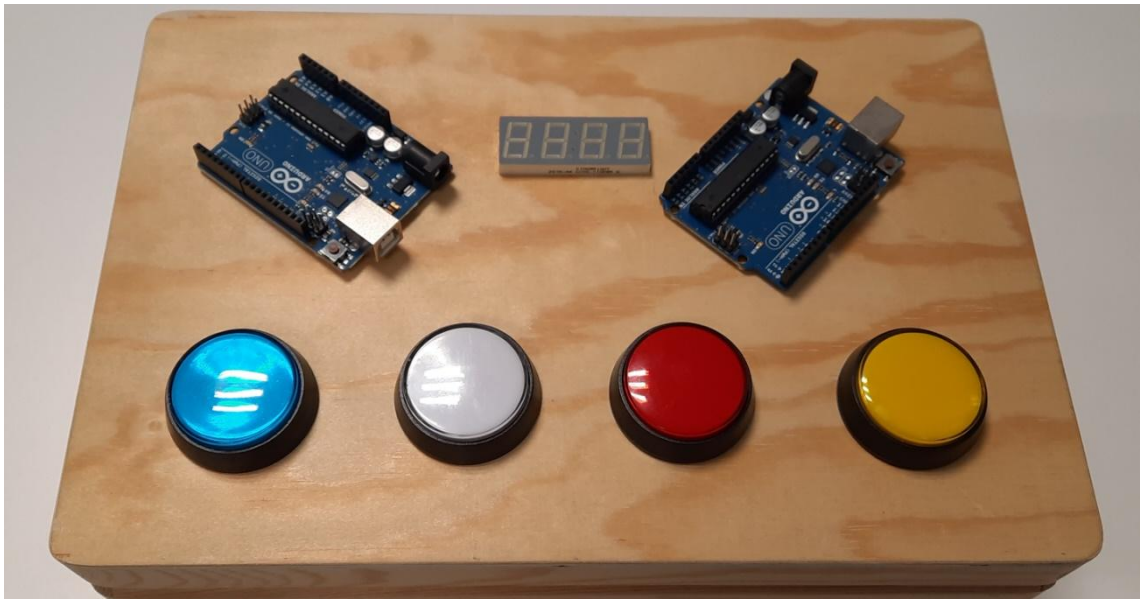
2.1 Tekoälyn käyttö

Tässä projektissa ei tehtäviä saa teetättää tekoälyllä. Tekoälyn käyttäminen oppimisen tukena on mahdollista, mutta jokaisen tulee ymmärtää, ja osata selittää

tekemänsä koodi ja ratkaisut. Ohjaava opettaja tulee katselmoinnissa pyytämään jokaista selittämään oman vastuualueensa koodin toiminta. Mikäli käy ilmi, että oman koodin toimintaa ei ymmärretä, eli sen on tehnyt joko tekoäly tai joku muu henkilö, tulkitaan tämä vilpiksi. Seuraus vilpistä on hylätty arvosana koko opintojaksolta.

2.2 Projektin tavoite

Tavoitteena projektissa on toteuttaa nopeuspeli - ”Speden Spelit” tyyliin. Pelin tarkoituksena on painaa pelin nappuloita samassa järjestyksessä, jossa niitä vastaavat valot syttyvät. Pelaaja voi jäädä hetkellisesti myös jälkeen, joten pelin on muistettava missä järjestyksessä nappeja pitää painaa. Peli laskee pisteitä onnistuneista painalluksista, ja peli päättyy, kun pelaaja painaa väärää nappia. Esimerkkitoteutus on esitetty Kuva 1. Kuvan laatikon sisällä on Arduino-mikrokontrolleri, jolla pelin logiikka pyörii, ja joka huolehtii painonappien lukemisesta, led-valojen ohjaamisesta ja tulosten näyttämistä 7-segmenttinäytöillä.



Kuva 1: Speden Spelit toteutettuna Arduino-mikrokontrollerilla.

Videoesimerkki yhdestä esimerkkitoteutuksesta, jossa on myös esimerkki vaihtoehtoisesta lisäpelimuodosta:

https://www.youtube.com/watch?v=YbE_IUfS-qo

Speden Spelit -projektiaihe valittiin yhteiseksi projektiaiheeksi, koska siinä voidaan hyödyntää edellisessä periodissa digitaalitekniikassa ja ohjelmoinnin perusteissa opittuja asioita (7-segmenttinäyttö, datalehden lukutaito, aliohjelmien tekeminen ja testaaminen). Lisäksi toisen periodin alussa mikrokontrollerit-kurssilla käydään läpi keskeytysten toiminta, Arduinon pinnien toiminta ja mikrokontrollerin sisäisiä laitteita, kuten timerien käyttö. Näitä opittuja tietoja ja taitoja hyödynnetään tässä projektissa.

2.3 Projektissa tarvittavat osat

Projektin tärkein osa on Arduino UNO R3 – mikro-ohjainalusta. Alusta ja ohjelmointiin tarvittava tietokone, koekytkentäalusta, sekä USB-kaapeli tulee löytyä jokaiselta ryhmän jäseneltä. Tällöin jokainen ryhmän jäsen voi itsenäisesti kehittää oman osa-alueensa toteutusta. Myös muun Atmega-pohjaisen kehitysalustan käyttö on mahdollista (Esimerkiksi Arduino Mega), mutta ei suositeltavaa aloittelijalle, koska tällöin piirin ominaisuudet ja rekisterikartta poikkeavat osin opetetusta, ja tämä vaatii itsenäisempää otetta koko ryhmältä.

Jokaiselle ryhmälle jaetaan OAMK:in kustantamana projektissa minimissään lisäksi tarvittavat komponentit:

- 74HC595 – serial to parallel -muunnin
- 7- segmenttinäytöt
- Ledit
- Painonapit
- Juotettava kytkentälevy

Elektroniikan laboratorion komponenttitornien komponentteja voi lisäksi käyttää projektissa kohtuullisissa määrin. Myös elektroniikan laboratorion 3D-tulostimet ovat käytettävissä.

Mikäli ryhmä haluaa käyttää projektissa komponentteja, joita laboratoriosta ei löydy, voi komponentteja hankkia myös omakustanteisesti.

2.4 Projektin osatoteutukset

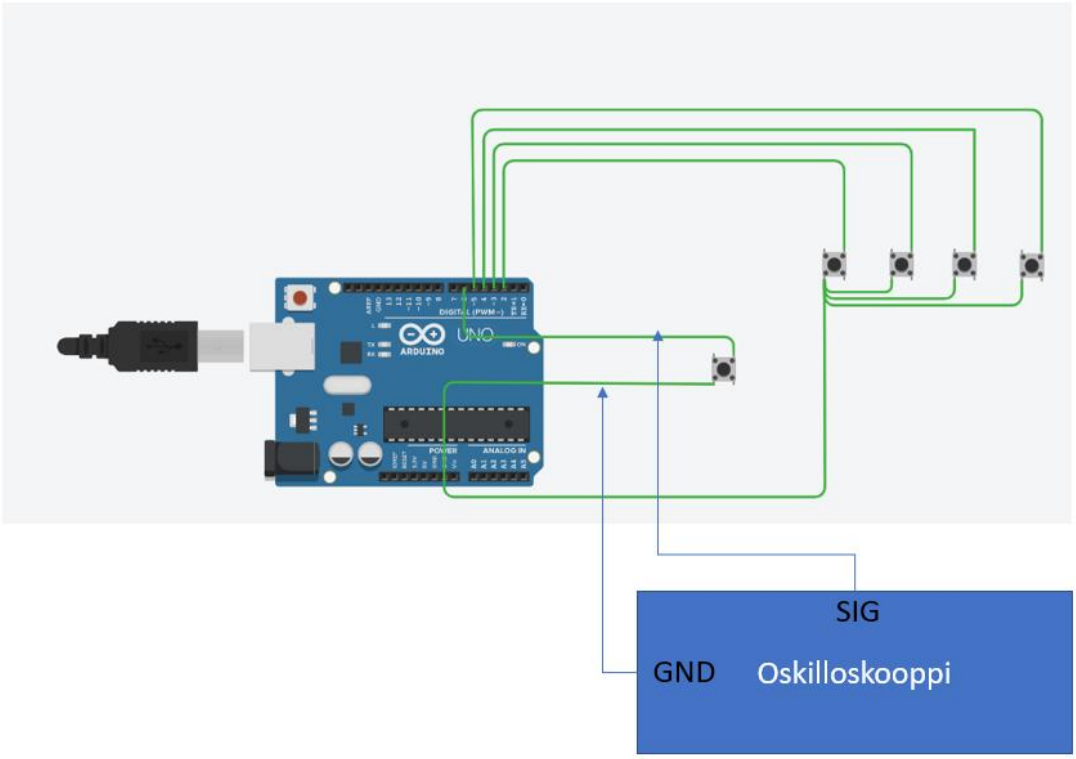
Projektin tehtävät jaetaan pienempiin osakokonaisuuksiin, jotka toteutetaan ja testataan rinnakkain itsenäisinä osinaan ja lopulta osista integroidaan yhteen toimivaksi peliksi. Pelin osakokonaisuudet (moduulit) ovat seuraavat:

- Buttons – rekisteröi painonappien painallukset
- Leds – Ohjaa painonappeja vastaavia valoja
- Display – Näyttää pisteet 7-segmenttinäytöllä
- Gamelogic – Ohjaa pelin kulkua

Kappaleissa 2.5 - 2.8 kerrotaan lyhyesti, minkälaista toiminnallisuutta eri ohjelma moduuleilta odotetaan ja annetaan vinkkejä, kuinka moduulin toteutuksessa kannattaa edetä. Lisäksi ryhmälle annetaan lähtökohdaksi ohjelmakoodipohjat, joihin toteutettavien aliohjelmien prototyypit on jo valmiiksi kirjoitettu. Opiskelijoiden tehtäväksi jää toteutuksen tekeminen ja testaaminen.

2.5 Buttons moduuli (buttons.h ja buttons.cpp tiedostot)

Kytketään Arduinoon 4 kpl kytkimiä kuvan 2 mukaisesti. Kytkintä painettaessa kytkin ”maadoittaa” eli kytkee Arduinon pinnin arvoon 0V. Kuvan 2 mittauskytkennän mukaisesti mitataan kytkimen mahdollinen kytkinvärähtely oskilloskoopin avulla. Mittauksen tarkoituksena on varmistaa, ettei kytkinvärähtely aiheuta useita keskeytyksiä kytkintä painettaessa.



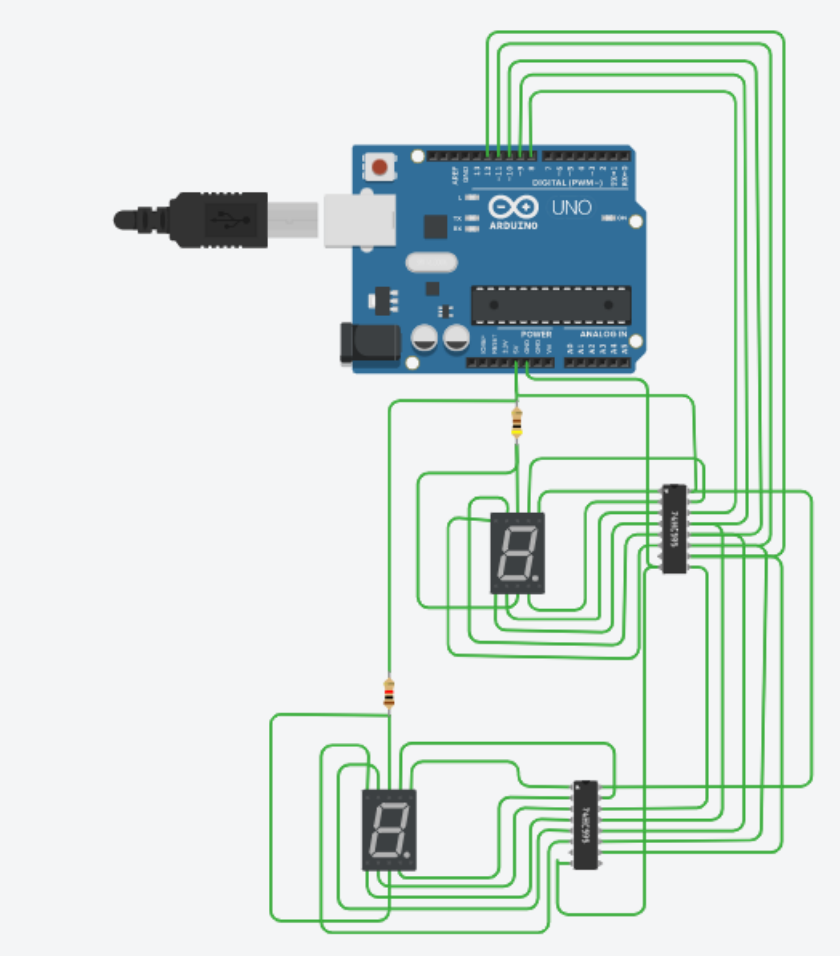
Kuva 2. Kytkimien liittäminen Arduinoon.

Buttons moduuli sisältää ainakin aliohjelman ***void initButtonsAndButtonInterrrupts(void)***, joka alustaa Arduinon digitaalipinnit 2,3,4,5,6 kytkinkäyttöön ja alustaa pin change keskeytykset PCICR (Pin Change Interrupt Control Register) ja PCMSK2 (Pin Change MaSK) rekistereiden avulla [2].

Toteutetaan keskeytyspalvelija mikrokontrollerit kurssilla esitetyllä tavalla 5 kytkimen painallusten tunnistamiseksi ja toteutetaan testiohjelma, jolla todistetaan, että kaikki 5 kytkintä aiheuttavat keskeytyksen ja jokaisen kytkimen painalluksen jälkeen testiohjelma tulostaa Arduinon sarjamonitorille tekstin ”Nappia X painettu”, missä X on Arduinon pinnin numero, johon kytkin on kytketty. Jos kohdassa 1 havaittiin, että kytkinvärähtely on mahdollista, toteutetaan myös kytkinvärähtelyn eliminointi ohjelmallisesti ja varmistetaan testein, että yksi kytkimen painallus aiheuttaa vain yhden tulostuksen sarjamonitorille.

2.6 Display moduli (display.h ja display.cpp tiedostot)

Liitetään Arduinoon kuvan 3 mukaisesti kaksi kappaletta 74HC595 serial-to-parallel muuntimia. Tutustutaan 74HC595 piirin datalehteen [1], jotta osataan asettaa tietyt pinnit vakioarvoihinsa ja osataan siirtää Arduinolta bitti bitiltä 2 kappaletta 8-bittisiä lukuja ensimmäisen serial-to-parallel piirin lähtöön. Tässä ei saa käyttää Arduinon valmiita kirjastokomponentteja, vaan opiskelijoiden on tehtävä oma aliohjelma, jonka prototyyppi on muotoa ***void writeByte(uint8_t number, bool last)***, missä number on 8-bittinen 7-segmentille kirjoitettava luku ja last parametri määrä kirjoitetaanko luku outputtiin vai ei. Tämän aliohjelman toiminta todistetaan kirjoittamalla aliohjelman avulla luvut 0x55 ja 0xaa vuorotellen serial-to-parallel muuntimille, joiden outputteihin on kuhunkin kytketty ledi. Tämä kannattaa testata TinkerCad simulaattorissa, sillä kytkennän tekeminen on siellä helpompi.



Kuva 3. Serial-to-parallel muuntimien liittäminen Arduinoon.

Kun 8-bittinen luku kyetään kirjoittamaan oikein kahdelle sarjaan kytketylle serial-to-parallel muuntimelle ja niihin kytketyille 7-segmenttinäytöille, niin seuraavaksi tehdään aliohjelma, jonka prototyyppi on muotoa ***void writeHighAndLowNumber(uint8_t tens,uint8_t ones)***. Aliohjelma ottaa siis ensimmäisenä parametrinaan kympit ja toisena parametrinaan ykköset, jotka pitää kirjoittaa edellistä write8bits-aliohjelmaa hyödyntäen 7-segmenttinäytölle. Aliohjelma testataan siten, että testiohjelma kirjoittaa writeHighAndLowNumber funktiota käyttäen luvut 0-99 näytölle.

Edellä esitettyjen aliohjelmien lisäksi display moduliin toteutetaan ***void initializeDisplay(void)*** ja ***void showResults(byte result)*** aliohjelmat. InitializeDisplay alustaa näytön tarvitsemat pinnit ja showResults aliohjelmalla voidaan näytölle kirjoittaa pelin tulos 0,1,..,99. Eli, tämä aliohjelma osaa irrottaa kympit ja ykköset ja käyttää sen jälkeen edellä esiteltyä aliohjelmaa kymppien ja ykkösten kirjoittamiseen näytölle.

2.7 Leds (leds.h ja leds.cpp tiedostot)

Toteutetaan 4 ledin sytyttely ja sammuttelu aliohjelmat ***void setLed(byte)***, ***void clearAllLeds(void)*** ja ***void setAllLeds(void)***. SetLed aliohjelman byte parametri kertoo Arduino pinnin numeron, johon kytketty ledi joko sytytetään tai sammutetaan. Speden Speleissä vain yksi ledi palaa kerrallaan, joten setLed aliohjelma tulee toteuttaa siten, että se sammuttaa edellisillä setLed aliohjelman kutsukerroilla sytytetyt ledit ja sytyttää vain halutun ledin. setAllLeds aliohjelmaa voidaan pelissä hyödyntää pelin alkutilan ja lopputilan esittämiseen. Aliohjelmat tulee toteuttaa siten, että pelilogiikassa ei tarvitse huomioida missä Arduinon pinneissä ledit kulloinkin sattuvat olemaan. Eli pelilogiikka voi olettaa, että setLed(0) sytytää pelin ledin 0 ja setLed(3) sytyttää ledin 3 vaikka ledit olisikin kytkettynä Arduinon analogiapinneihin A2,A3,A4,A5, mihin ledit pitääkin tässä pelissä kytkeä, jotta digitaalipinnejä jää näppäinten ja näytön käyttöön.

Toteutetaan myös testiohjelma, jolla edellä kuvattujen 3 aliohjelman toiminta tulee todistettua. Koska Leds moduli on melko yksinkertainen, niin tämän modulin tekijä tekee lisäksi ylimääräiset **void show1()** ja **void show2()** funktiot, joilla saadaan ennen peliä ja pelin jälkeen esittää näyttäviä valojuttuja laitteen neljällä ledillä. Show1 funktio kirjoittaa numerot 0,1,,,15 binääriesityksenä ledeillä. Jokaisen numeron jälkeen funktio tekee pienen viiveen, jotta binärilukuesitys on helposti tunnistettavissa. Show2 funktio kirjoittaa ledejä 0:sta lediin 3 siten, että ensin palaa ledi0, pienen viiveen jälkeen ledi0 ja ledi1, pienen viiveen jälkeen ledit0,1,2 ja lopuksi kaikki ledit palavat. Ja tämä ”valoshow” toistuu kiihtyvällä tahdilla.

2.8 Gamelogic (spedenSpelit.ino tiedosto)

Pelilogiikka toteutetaan spedenSpelit.ino tiedostoon, missä on ***void setup(void)*** ja ***void loop(void)*** funktiot ja näiden lisäksi ainakin ***void startTheGame(void), void stopTheGame(void)*** pelin käynnistämiseen ja sammuttamiseen sekä ***void checkGame(void)*** käyttäjän inputin tarkistamiseen.

Pelilogiikka toteutetaan siten, että peli arpoo numeroita 0,1,2,3 satunnaisesti. Arpominen tapahtuu Timer1:n keskeytyspalvelijassa. Arvotut numerot talletetaan 100 alkion mittaiseen taulukkoon (randomNumbers). Kun käyttäjä painaa kytkimiä 0,1,2,3 nämä näppäinten painallukset talletetaan 100 alkion mittaiseen taulukkoon (userNumbers) näppäinten keskeytyspalvelijassa. Tuossa samaisessa keskeytyspalvelijassa asetetaan bool muuttuja timeToCheckGameStatus true arvoon ja tuota bool muuttujaa tarkistetaan loop funktiossa. Ja aina, kun tuo muuttuja on arvossa true kutsutaan checkGame aliohjelmaa. CheckGame aliohjelma vertaa randomNumbers ja userNumbers taulukoita aina siihen taulukon indeksiin asti, kuinka monta kertaa käyttäjä on näppäimiä pelin alusta asti painanut. Jos taulukoiden alkiot ovat samat, peli jatkuu ja pelin tilanne näytetään showResults aliohjelmalla. Jos taas viimeinen näppäinpainallus oli virheellinen, niin peli lopetetaan stopTheGame funktiolla. StopTheGame funktio disabloi

Timer1 keskeytykset ja startTheGame enabloi Timer1 keskeytykset. Peli siis joko alkaa tai keskeytyy, kun Timer1 alkaa tai lopettaa antamasta keskeytyksiä, joiden tahdissa satunnaisia lukuja arvotaan ja ledejä sytytetään.

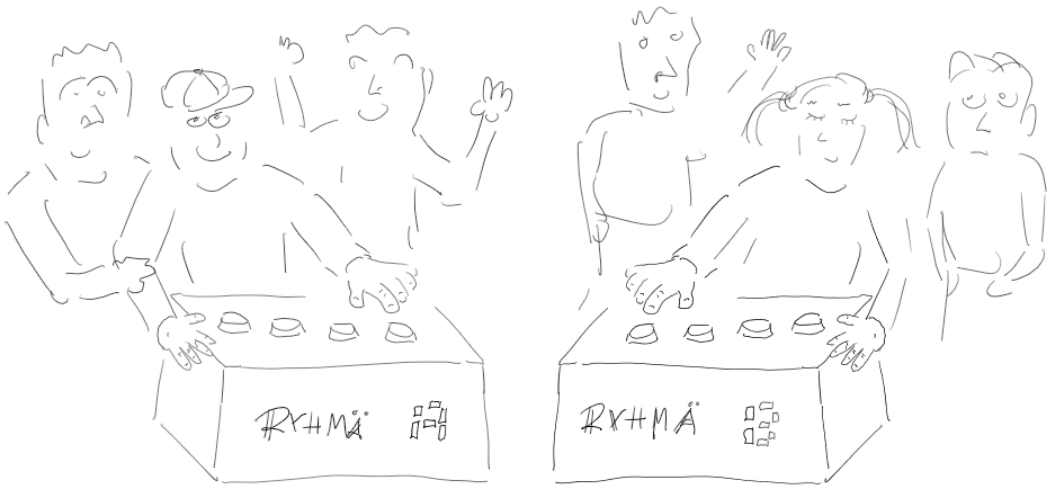
Jotta kaikkien ryhmien Speden Spelit toimisivat samalla vaikeustasolla, täytyy Timer1:n alkukeskeytystahdista ja sen nopeutumisesta sopia. Alustavasti määritellään, että keskeytystahti on alussa 1 Hz ja tahtia nopeutetaan 20% kymmenen oikean painalluksen jälkeen. Lisäksi kannattaa huomata, että ohjelmaa ei saa eikä kannata toteuttaa siten, että kaikki toiminta tehdään keskeytyspalvelijassa. Esimerkiksi timer1 keskeytyspalvelijan tulee vain arpoa ledi, joka pitää seuraavaksi sytyttää ja nappikeskeytyspalvelija tunnistaa vain mitä nappia on painettu, mutta kaikki muu suoritetaan pääohjelmassa (eli loop() funktiosta käsin).

2.9 Projektiryhmän määrittelemät ylimääräiset tavoitteet

Ylimääräinen tavoite on musiikki, johon teimme peli-, häviö- ja voittomusiikit.

2.10 Loppukilpailu

Projekti huipentuu ennen joululomalle jääntiä tapahtuvaan loppukilpailuun. Kilpailu toteutetaan pudotuspeliperiaatteella, eli kaksi ryhmää kilpailee aina keskenään ottelussa, ja voittanut jatkaa eteenpäin seuraavaan otteluun. Kilpailussa jaetaan palkintoja, mutta sillä ei ole vaikutusta opintojakson arviointiin.



Täsmällisemmin kuvattuna kilpailu siten haastaa toteutuksen tehneiden digitaalilogiikan osaamista, loogista päättelykykyä, oppimiskykyä ja sorminäppäryyttä seuraavasti:

1) ryhmä A tekee kilpailusuorituksen omalla laitteellaan A ryhmän B läsnä ollessa ja tarkkaillessa vierestä suoritusta. Ryhmä B laittaa ylös ryhmän A tuloksen laitteella A

2) vastaavasti ryhmä B tekee kilpailusuorituksen omalla laitteellaan B ryhmän A tarkkaillessa. Ja, ryhmä A laittaa ylös ryhmän B tuloksen laitteella B

Seuraavaksi siirrytään tekemään suoritukset samoin kuin kohdissa 1) ja 2) mutta siten, että laitteet vaihdetaan keskenään, eli:

3) ryhmä A tekee kilpailusuorituksen laitteella B

4) vastaavasti ryhmä B tekee kilpailusuorituksen laitteella A

Kilpailukierroksen jälkeen tehdään vertailu seuraavasti:

ryhmän A tulos = vaiheen 1 tulos + vaiheen 3 tulos

ryhmän B tulos = vaiheen 2 tulos + vaiheen 4 tulos

Se, kummalla ryhmällä on vertailussa parempi tulos, pääsee jatkoon, jossa se kohtaa uuden ryhmän.

Mikäli ryhmien määrä ei mene tasan, voidaan otella jokin vaihe pudotuspelistä myös kolmen ryhmän kesken, siten että kaikki pelaavat kolmella eri laitteella.

Loppukilpailun finaalissa ottelevat vastakkain SPL- ja SPO-ryhmien voittajat.

Kilpailun aikana suoritetaan myös äänestykset SPL- ja SPO ryhmissä, joissa äänestetään kunkin ryhmän toteutuksista yleisön suosikki, eli se ryhmä, joka on toteuttanut kaikkein hienoimman ja teknisesti kiinnostavimman nopeuspelin.

2.10.1 Jaettavat palkinnot:

Loppukilpailussa jaetaan palkinnot seuraavasti:

- Loppukilpailun voittaja
- Loppukilpailun 2. sija
- Yleisön suosikki SPL-ryhmästä
- Yleisön suosikki SPO-ryhmästä

3 TOTEUTUSSUUNNITELMA

3.1 Projektin vaiheistus ja aikataulu

Projektia aloitetaan toteuttamaan periodin kolmannen viikon aikana, koska periodin alussa opetetaan intensiivisenä toteutuksena mikrokontrollerin toimintaperiaate ja rakenne -kurssi. Tähän kappaleeseen projektiryhmä suunnittelee, miten aikoo projektin kunkin neljän moduulin osalta toteuttaa ja kuinka nuo neljä moduulia integroidaan yhteen toimivaksi peliksi. Projektiryhmän tulee tehdä taulukon 1 suunnitelma viikon 46 aikana ja se tarkistetaan ohjaavan opettajan toimesta viikon 47 viikkopalaverissa.

Ryhmän kiinteät virstanpylväät ovat seuraavat:

- Viikko 47 viikkopalaveri: Projektisuunnitelman katselmointi
- Viikko 50 viikkopalaveri: Koodien katselmointi
- Viikko 51 loppupelit: Toteutus ja dokumentaatio valmis

Muilta osin ryhmä määrittelee itse aikataulunsa.

TAULUKKO 1. Projektin vaiheet ja aikataulu

MÄÄRÄAIKA	TEHTÄVÄN KUVAUS	VASTUUHENKILÖ
Viikko 46	Tehtävä1: Ledit ja pelimusiikki Tehtävä2: Napit Tehtävä3: Näytöt Tehtävä4: Pelilogiikka	Tuomo Salo Kasperi Salmela Topi Kuha Robert Pudas
Viikko 47	Projektisuunnitelman katselmointi	-//-
Viikko 48	Projektia ja koodeja eteenpäin	-//-
Viikko 49	Kytkennän juottaminen ja koodit valmiiksi	-//-
Viikko 50	Koodien katselmointi	-//-
Viikko 51	Projekti valmis, dokumentaatio valmis	-//-

3.2 Projektin toimitukset

Projektiryhmän tulee toimittaa alla olevan listan dokumentit ryhmän Teams-alustalle:

1. Projektipalaverien asialistat ja muistiot
2. Ryhmän katselmoitu ja hyväksytty projektisuunnitelma
3. Kommentoidut Arduino-koodit zip-tiedostona
4. Vuokaaviot Arduino ohjelman toiminnasta
5. Lyhyt video (esim. linkki Youtubeen), jossa näkyy toteutetun laitteen toiminta, kun sitä pelataan. Videosta tulee käydä ilmi myös mahdollisten lisäominaisuuksien toiminta. Videolla tulee käydä ilmi mikä ryhmä sen on toteuttanut.

LÄHTEET

[1] DIODES, 2018, 74HC595 datalehti. Hakupäivä 13.10.1023. https://www.diodes.com/assets/Datasheets/74HC595.pdf

[2] ATMEL, ATMega328P datasheet, s.56. Hakupäivä 13.10.2023. https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf