**Projektisuunnitelma**

****

DOKUMENTIN VERSIOHISTORIA

|  |  |  |  |
| --- | --- | --- | --- |
| VERSIONUMERO | PÄIVÄMÄÄRÄ | MUUTOSPERUSTE | TEKIJÄ / HYVÄKSYJÄ |
| 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

Sisällysluettelo 2

1 DOKUMENTIN TARKOITUS 3

2 PROJEKTIN SISÄLTÖ 3

2.1 Tausta ja lähtökohdat 3

2.2 Tehtävä ja tavoitteet 4

2.3 Buttons moduli (buttons.h ja buttons.cpp tiedostot) 5

2.4 Display moduli (display.h ja display.cpp tiedostot) 6

2.5 Leds (leds.h ja leds.cpp tiedostot) 8

2.6 Gamelogic (spedenSpelit.ino tiedosto) 8

2.7 Projektiryhmän määrittelemät ylimääräiset tavoitteet 9

2.8 Kilpailullinen osio (ehdolle) 10

3 TOTEUTUSSUUNNITELMA 12

3.1 Projektin vaiheistus ja aikataulu 12

3.2 Projektin toimitukset 12

Lähteet 13

# DOKUMENTIN TARKOITUS

OAMK:n tietotekniikan tutkinto-ohjelmassa on tietotekniikan sovellusprojekti -niminen opintojakso, joka toteutetaan ensimmäisenä lukuvuotena syksyllä toisessa periodissa. Sovellusprojekti toteutetaan neljän opiskelijan ryhmissä ja tässä dokumentissa määritellään kaikkien ryhmien projektiaihe sekä projektin tavoitteet ja toteutuksen vaiheet.

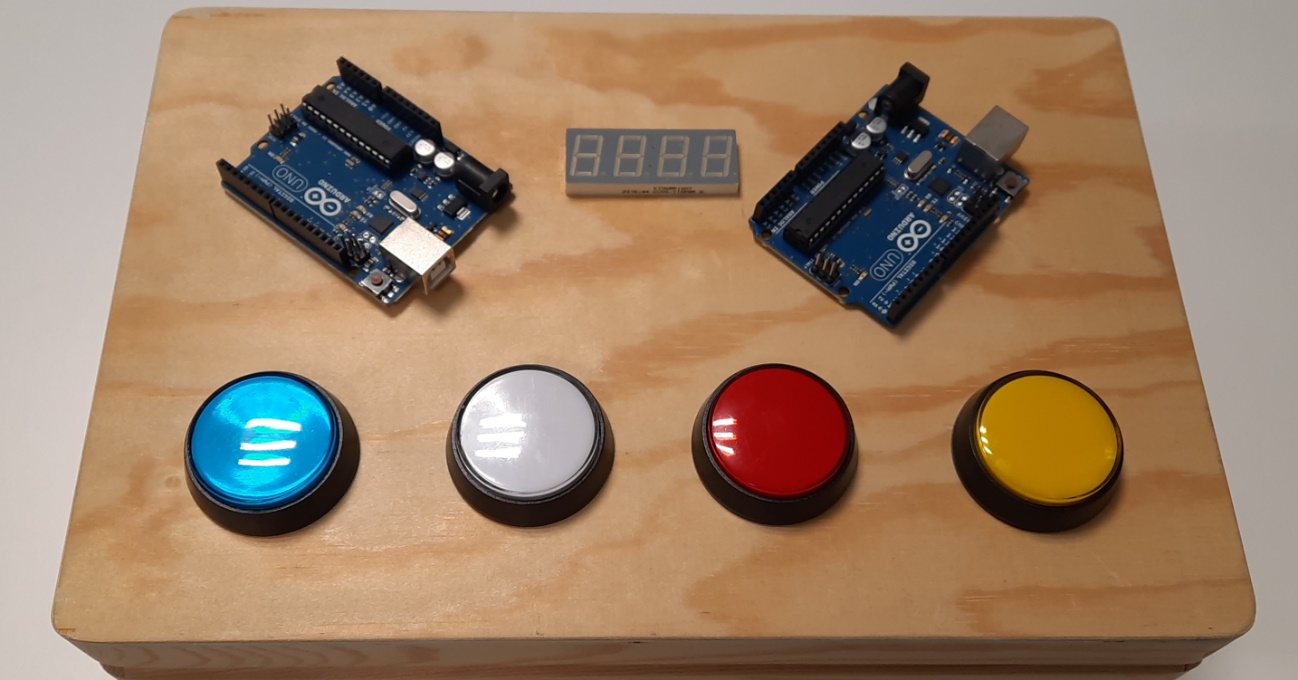
Jos jokin opiskelijaryhmä haluaa kuitenkin määritellä projektiaiheensa itse, joutuu tuo projektiryhmä itse tekemään tätä dokumenttia vastaavan määrittelyn omasta projektiaiheestaan. Oman projektin tulee olla vaativampi kuin tässä dokumentissa määritelty aihe.

# PROJEKTIN SISÄLTÖ

## Tausta ja lähtökohdat

Tietotekniikan sovellusprojektin laajuus on 12 opintopistettä ja kurssi pitää sisällään 6 opintopisteen laajuisen projektiosuuden, 3 opintopisteen laajuisen mikrokontrollerin rakenne- ja toimintaperiaate osuuden sekä 3 opintopisteen laajuisen matematiikkaa projektin tarpeisiin sisältävän osuuden. Tässä dokumentissa määritellään vain 6 opintopisteen projektiosuuden tehtävät ja tavoitteet.

Kaikille ryhmille yhteisenä projektiaiheena toteutetaan kuvan 1 mukainen laite. Kuvan laatikon sisällä on Arduino-mikrokontrolleri, jolle tehdään projektiryhmän toimesta ohjelmallisesti painonappien lukua, led-valojen vilkuttelua ja tulosten näyttämistä 7-segmenttinäytöillä.

****

Kuva 1. Speden Spelit toteutettuna Arduino-mikrokontrollerilla.

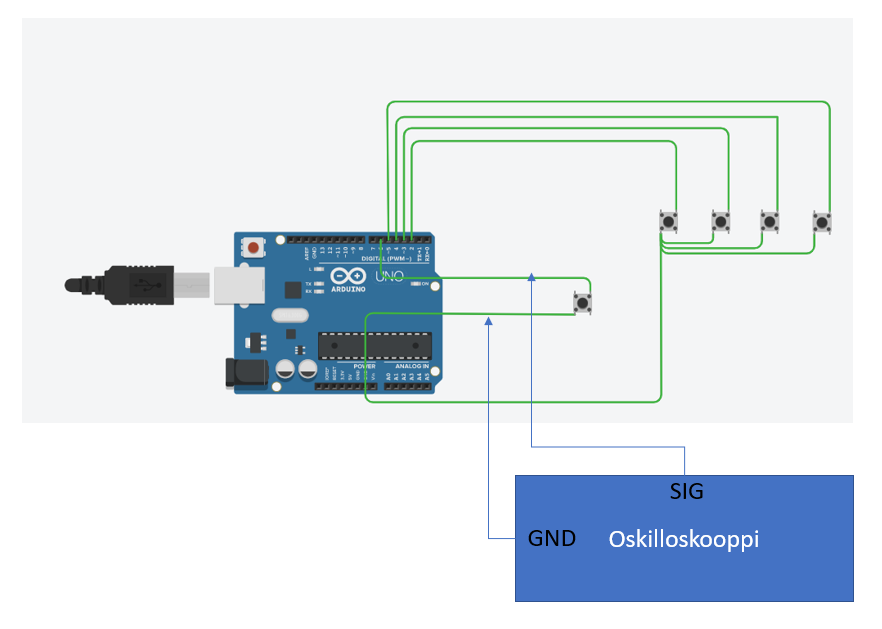
Speden Spelit -projektiaihe valittiin yhteiseksi projektiaiheeksi, koska siinä voidaan hyödyntää edellisessä periodissa digitaalitekniikassa ja ohjelmoinnin perusteissa opittuja asioita (7-segmenttinäyttö, datalehden lukutaito, aliohjelmien tekeminen ja testaaminen). Lisäksi toisen periodin alussa mikrokontrollerit-kurssilla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

## Tehtävä ja tavoitteet

Projektin tehtävät jaetaan pienempiin osakokonaisuuksiin, jotka toteutetaan ja testataan omina osinaan ja lopulta osista integroidaan toimiva peli. Pelin osat (modulit) ovat buttons, leds, display ja gamelogic. Seuraavissa kappaleissa 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.

## 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.



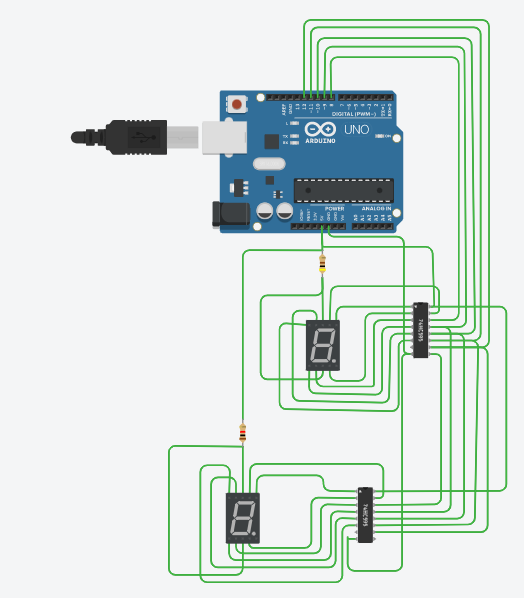
Kuva 2. Kytkimien liittäminen Arduinoon.

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.

## 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.

## 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.

## Gamelogic (spedenSpelit.ino tiedosto)

Pelilogiikka toteutetaan spedenSpelit.ino tiedostoon, missä on ***void setup(void)*** ja ***void loop(void)*** funktiot ja näiden lisäksi tiedostoon toteutetaan keskeytyspalvelija ***ISR(PCINT2\_vect)*** 5 napin painallusten käsittelemiseksi, ***void initializeTimer(void)*** Arduino Timer1:n alustamiseksi, ***ISR(TIMER1\_COMPA\_vect)*** timerin keskeytysten käsittelemiseksi, ***void startTheGame(void), void stopTheGame(void)*** pelin käynnistämiseen ja ***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 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 virheelinen, 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 10% kymmenen oikean painalluksen jälkeen.

## Projektiryhmän määrittelemät ylimääräiset tavoitteet

## Kilpailullinen osio

Edellä esitellyn perustoteutuksen lisäksi jokainen ryhmä toteuttaa oman erityisen pelilogiikan muovaamalla edellä kehitettyä perusratkaisua vapaasti päättämällään tavalla siten, että ryhmä pärjäisi mahdollisimman hyvin tässä kappaleessa kuvatussa ”loppukilpailussa” parhaiten.

Kilpailullisen osion idea on se, että mitä paremmin ryhmä pystyy itse rakentamaan ja hallitsemaan oman laitteensa pelilogiikan ja toisaalta mitä vaikeampi kyseinen muovattu pelilogiikka on sitä ennestään tuntemattomalle (eli toisille ryhmille), sitä paremmin ryhmä pärjää.

Kehitettyään oman pelilogiikkansa ryhmän tulee siten itse harjoitella suorittamaan juuri oma pelinsä mahdollisimman hyvin. Samalla luonnollisesti omaa loogisten toteutusten analysointi- ja havainnointikykyä kannattaa kehittää ja harjoitella.

Mitä tämä tarkoittaa käytännössä? Kullakin ryhmällä on aluksi kurssilla määritellyn tehtävän mukaisesti käytössään alkuperäinen ja samanlainen pelitoteutus ”Speden Spelit”. Näin ollen, ottaa ryhmä eteensä itse tekemänsä pelitoteutuksen tai jonkin toisen ryhmän tekemän toteutuksen, menestyminen pelissä on luonnollisesti keskimäärin samanlainen.

Mutta, muovatun pelilogiikan kehittämisen myötä esimerkiksi ryhmä ”Etanat” luo pelilogiikasta oman version ”Etanan Spelit”. Ja, toisaalta ryhmä ”Hyttyset” luo pelilogiikasta vastaavasti version ”Hyttysen Spelit”. Molemmissa logiikoissa on sama taustaidea kun ”Speden Speleissä” – eli pelaajan tulee muistaa tai päätellä, mikä onkaan seuraavaksi painettava näppäin. Mutta – esimerkiksi ”Etanat” voivat luoda omaan peliinsä erityisen ovelan logiikan, jonka myötä aina vaikkapa joka viides painallus on aina sininen näppäin. Ja, tämän avulla ”Etanat” ajattelevat, että he tämän salaisuuden tietämisen myötä pärjäävät muille ryhmille! Ja – vastaavasti ”Hyttyset” muovaavat pelilogiikkaansa tahollaan ehkäpä siten, että aina valkoisen ja punaisen jälkeen seuraa keltainen. Ja, samalla tavalla ”Hyttyset” hykertelevät että tällä idealla he kyllä peittoavat helposti kaikki muut kilpailijat!

Eli – muovatussa toteutuksessa on jotakin erityistä, jonka myötä itse peli ”äkkiseltään” näyttää olevan ihan kuin ”Spede Spelit”, mutta siinä onkin vaivihkaisia muutoksia edellä kuvatun tapaan. On huomattava, että modifikaatiot voivat olla edellä kuvattuja monimutkaisempia. Esimerkiksi voi olla, että mikäli näppäimiä ei paina tietyllä rytmillä, pelilogiikka on täsmälleen sama kuin ”Speden Speleissä”, mutta sen vauhti kiihtyy ennalta-arvaamattomiin nopeuksiin, jne. Ja, tämän tietävät osaavat tietysti ”väärää rytmiä” varoa, ja niin edelleen. Vain mielikuvitus on rajana pelilogiikan muovaamisessa, kunhan vain ”Speden Spelien” perusidea säilyy taustalla!

**Loppukilpailu:**

Varsinainen kilpailu tapahtuu ”ensimmäisessä karsinnassa” siten että opettajien arpomat ryhmät kilpailevat keskenään. Voittaneet ryhmät pelaavat tämän jälkeen toisiaan vastaan, ja niin edelleen. Tätä jatketaan niin monta kertaa, että voittaja on selvinnyt.

Kuva, joka sisältää kohteen luonnos, piirros, Piirrokset, kuvitus

Kuvaus luotu automaattisesti

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

Kohtia 1) ja 2) toistetaan tarvittaessa (pelilogiikoiden yleinen vaikeustaso) opettajien täsmällisemmin määrittämä toistokerta. Tärkeää on se, että A voi havainnoida B:n suoritusta ja toisinpäin. Molemmilla ryhmillä on sopivasti aikaa keskustella keskenään siitä, ”mikähän tuon vastustajaryhmän pelilogiikka oikein on?” Tästä tarkemmin ohjeet itse kilpailun aikana opettajavetoisesti.

Kun vaihe 1) ja 2) ovat valmiit, laitetaan ylös ryhmän tulos (jos useita toistokertoja, tuloksista paras).

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

Kohdista 3) ja 4) laitetaan ylös ryhmien tulokset (jos useita toistokertoja, tuloksista paras). Huomattavaa on se, että mikäli A pystyi havainnoimaan B:n toimintalogiikan, tästä on merkittävä hyöty pelissä. Samoin toisinpäin, mikäli B pystyi havainnoimaan A:n toimintalogiikan.

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.

Koko kilpailun kaikkien karsintakierrosten myötä voittanut ryhmä palkitaan.

Kilpailullisen osion tarkoituksena on perehdyttää tekijöitä digitaalilogiikan ja loogisen päättelykyvyn luovaan soveltamiseen sekä myös motivoida ryhmiä ratkaisun toteutuksessa ja sen taitavassa käyttämisessä.

# TOTEUTUSSUUNNITELMA

## 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.

*TAULUKKO 1. Projektin vaiheet ja aikataulu*

|  |  |  |
| --- | --- | --- |
| MÄÄRÄAIKA | TEHTÄVÄN KUVAUS | VASTUUHENKILÖ |
| Viikko 46 | Tehtävä1…  Tehtävä2…  Tehtävä3…  Tehtävä4… | Jäsen NN1  Jäsen NN2  Jäsen NN3  Jäsen NN4 |
| Viikko 47 |  |  |
| Viikko 48 |  |  |
| Viikko 49 |  |  |
| Viikko 50/51 | Projekti valmis, dokumentaatio valmis |  |

## 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. Vuokaavio Arduino ohjelman toiminnasta

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>