OHJELMAN PROTOTYYPIN

VAATIMUSMÄÄRITTELY

Taitaja9 sovellus

 KEHITYSTYÖ

Kirjasto: C:\Users\matsk\OneDrive - Raision seudun koulutuskuntayhtymä

Tiedosto: Taitaja9-kilpailu R.H, A.M, D.T, J.O.docx

Versio: 0.1

Tila: luonnos [x]

ehdotus []

hyväksytty []

Muutoshistoria:

Versio Pvm. Tekijä Huomautus

0.1 20.8.2024 Rasmus

JAKELU: Projektin johtoryhmä

Jussi Kuosa

Opiskelija nimi: Rasmus, Arthur, Dan, Joona

**LYHENTEET**

MTBF (Mean Time Between Failures) on keskimääräinen aika laitteen vikaantumiseen sen edellisestä alkuperäiseen kuntoon saattamisesta (korjauksesta).

PC Personal Computer

SW Software

**1. YLEISTÄ**

Tässä dokumentissa on kuvattu vaatimukset taitaja9-kilpailuun.

Tässä vaatimusmäärittelyssä otetaan kantaa myös taitaja9-kilpailun kehitysvaiheisiin ja vaatimuksiin.

Projektin tavoitteena on kehittää taitaja9-kilpailua.

[määrittele tähän, miten ohjelma toimii pääpiirteittäin. Määrittele tähän kenelle ohjelma tehdään. Määrittele tähän mitä ohjelmalla voidaan tehdä ja mitä se sisältää pääpiirteittäin].

Katso projektin speksit dokumentista viivojen kohtaan tarkennuksia.

Yleistä kappaleeseen tulee korkeintaan yksi A4 tekstiä (Hyvin lyhyesti kuvausta spekseistä, kirjoita tarkempi kuvaus Johdanto-kappaleeseen)

- **Projektihenkilöstö**



Dan

Joona

Rasmus

Arthur

**1. JOHDANTO / TOTEUTUSSUUNNITELMA**

* vaatimusmäärittelyssä otetaan ensimmäisen kerran kantaa tuotteelle asetettaviin teknisiin vaatimuksiin
* vaatimusmäärittelyssä asiat esitetään asiakkaan kannalta nähtynä
* vaatimusmäärittelyssä asiat esitetään tavalla, joka on kenenkä tahansa ymmärrettävissä (formaalien kielten ja kuvaustapojen käyttöä harkittava tarkoin)
* Toteutussuunnitelmassa viitataan liitteessä olevaan projektiaikatauluun

**2. TOIMINNALLISET VAATIMUKSET** (tiimissä toteutetut toiminnot)

* toimintokuvaus selkokielellä (sanalliset kuvaukset toiminnoille)
* Jokainen Taitaja9-projektiin osallistuva kuvaa oman projektiosuutensa sanallisen toimintakuvauksen tähän kappaleeseen (tarkennettu kuvaus ja vuokaavio tulee kohtaan "yksityiskohtainen suunnittelu")

**3. ARKITEHTUURI (Use Case)**

* Määrittele tähän millaisia Java Paketteja, kirjastoja ja luokkia ohjelma sisältää. Millaisia käyttäjiä ohjelmistolla on. Tässä voidaan esitellä käyttötapauskaavio ja luokkakaaviot.
* Use Case kaavio
* Jokainen Taitaja9-projektiin osallistuva lisää oman projektiosuutensa lisätyn toiminnon Use Case- kaavioonja lisää päivitetyn kuvan tähän kappaleeseen
* Jos lisätty toiminto on laajennus jo olemassa olevaan toimintoon niin se kuvataan "exclude" nuolella ja jos se on lisätty ominaisuus kuten tietorakenne niin se kuvataan "include" nuolella olemassa olevaan toimintoon (toiminnot kuvataan ellipsillä).



**4. YKSITYISKOHTAINEN MÄÄRITTELY** (toiminnalliset kuvaukset)

Määrittele tähän millaisia toimintoja ohjelma sisältää.

Esimerkiksi Taitaja9– ohjelma sisältää käyttöliittymän josta käyttäjä voi syöttää rekisteröintitietoa osallistuviin joukkueisiin liittyen (kuvattu esim. vuokaaviolla).

* Yksityiskohtainen suunnittelu (Vuokaaviot Use Casen toiminnoille)
* Jokainen Taitaja9-projektiin osallistuva lisää oman projektiosuutensa lisätyn toiminnon yksitiskohtaisen kuvauksen vuokaaviona tähän kappaleeseen

Huomioitavat asiat kuvauksessa:

* Tietorakenteet (kuvaukset tauluista ja kyselyistä sekä csv- ja txt-tiedostot, ja raportit)

Muut määriteltävät asiat:

* ohjelmointiympäristön määritely
* parametrit, tietoliikenne, protokollat
* liityntöjen määritykset

Määrittele tähän kaikki ohjelman liitynnät esim. Jos ohjelmassa on tietokanta- liityntä tms. (käyttöliittymät on kuvattu myöhemmin omassa kappaleessaan)

* uudelleenkäytettävyys suunnittelussa
* toimintaan liittyvät standardit

**5. KÄYTETTÄVYYSVAATIMUKSET**

* käyttöliittymän vaatimukset

esim. Jos ohjelmassa on web- liityntä ja Java UI-liityntämoduli

* Käyttöliittymän määritys
* Asennusvaatimukset

**6. TESTAUSRAPORTTI** (testauskoodit)

* Taitaja9-projektissa testausvastuussa oleva lisää seuraavat asiat tähän kappaleeseen. Jos testausvastaavaa ei ole tiimissä niin kukin täydentää tätä kappaletta omalta osaltaan.
* Testaussuunnitelma ja vuokaavio
* Junit Test Case kuvakaappaukset liitteeksi
* Yksikkötestit (vuokaaviot)
* Aja koodi ja laita ajotuloksista kuvakaappaukset tähän kappaleeseen

**7. LIITTEET**

PROJEKTIAIKATAULU

Gant- kaavio tai muu vastaava (Määrittele tähän Excelillä tekemäsi projektin aikataulu)

**TOTEUTUSSUUNNITELMA TUOTANTOVERSIOLLE**

* Tavoitteet projektille
* Työnjako tiimin sisällä, kuka vastaa mistäkin toiminnosta
* Aikataulun päivitys (gant-kaavio)
* Toiminnallisten vaatimusten päivitys
* Arkkitehtuurin päivitys
* Yksityiskohtaisen määrittelyn päivitys
* Käytettävyys / käyttöliittymän päivitys
* Testaussuunnitelman päivitys

Koodit:

Lisää liitteeseen Notepad++ ohjelman avulla koodiin rivinumerot seuraavasti.

a) asenna Notepad++, asenna AcrobatReader

b) kopioi koodi Notepad++ ohjelmaan

c) valitse koodi ctrl-a ja poista rivitoiminnolla tyhjät rivit

d) printtaa koodi pdf-tiedostoon

e) valitse koodi pdf-tiedostosta ctrl-a, jolloin saat rivinumerot mukaan koodiin

f) liitä koodi, jossa on rivinumerot liitteeseen Loppuraporttiin

def on\_received\_number(receivedNumber):

global outdoorTemp, time

outdoorTemp = receivedNumber

time = receivedNumber

radio.on\_received\_number(on\_received\_number)

def on\_button\_pressed\_a():

basic.show\_number(input.temperature())

input.on\_button\_pressed(Button.A, on\_button\_pressed\_a)

def on\_button\_pressed\_b():

basic.show\_number(outdoorTemp)

input.on\_button\_pressed(Button.B, on\_button\_pressed\_b)

def on\_logo\_touched():

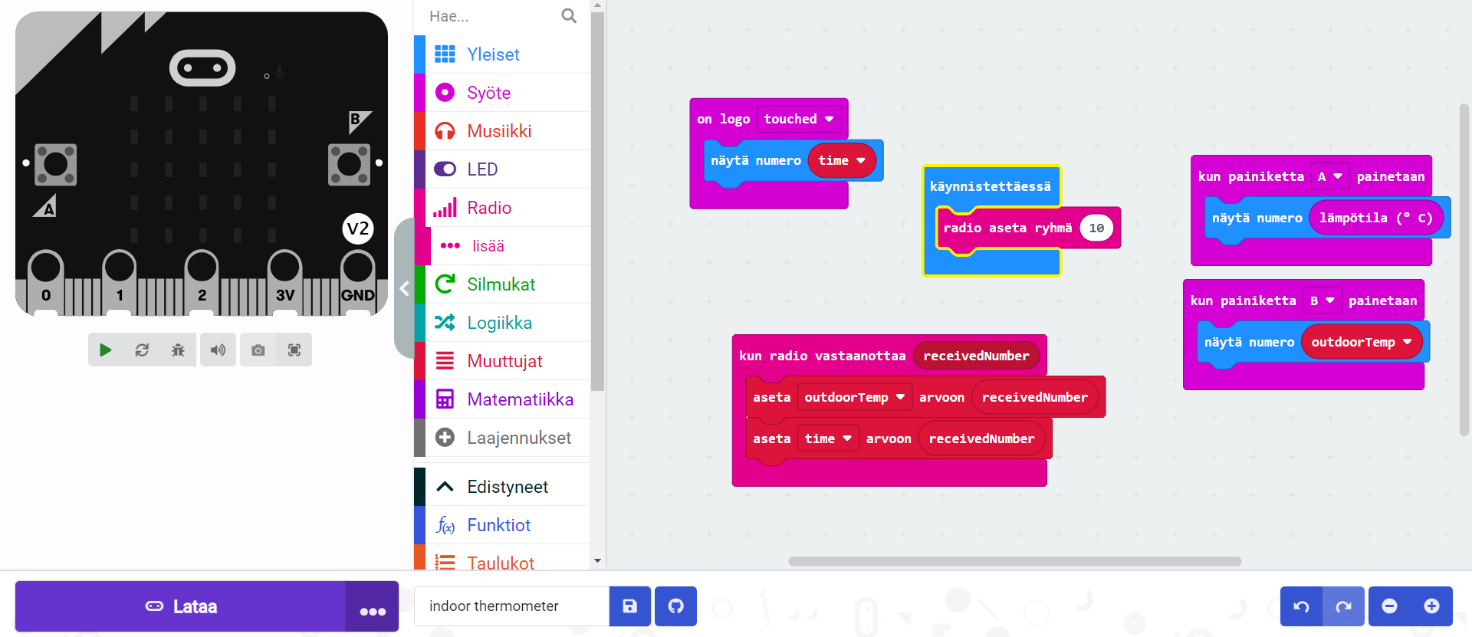
basic.show\_number(time)

input.on\_logo\_event(TouchButtonEvent.TOUCHED, on\_logo\_touched)

time = 0

outdoorTemp = 0

radio.set\_group(10)



def on\_logo\_touched():

    global start

    start = input.running\_time()

    basic.show\_icon(IconNames.HEART)

input.on\_logo\_event(TouchButtonEvent.TOUCHED, on\_logo\_touched)

def on\_logo\_released():

    global time

    time = input.running\_time() - start

    radio.send\_number(Math.idiv(time, 1000))

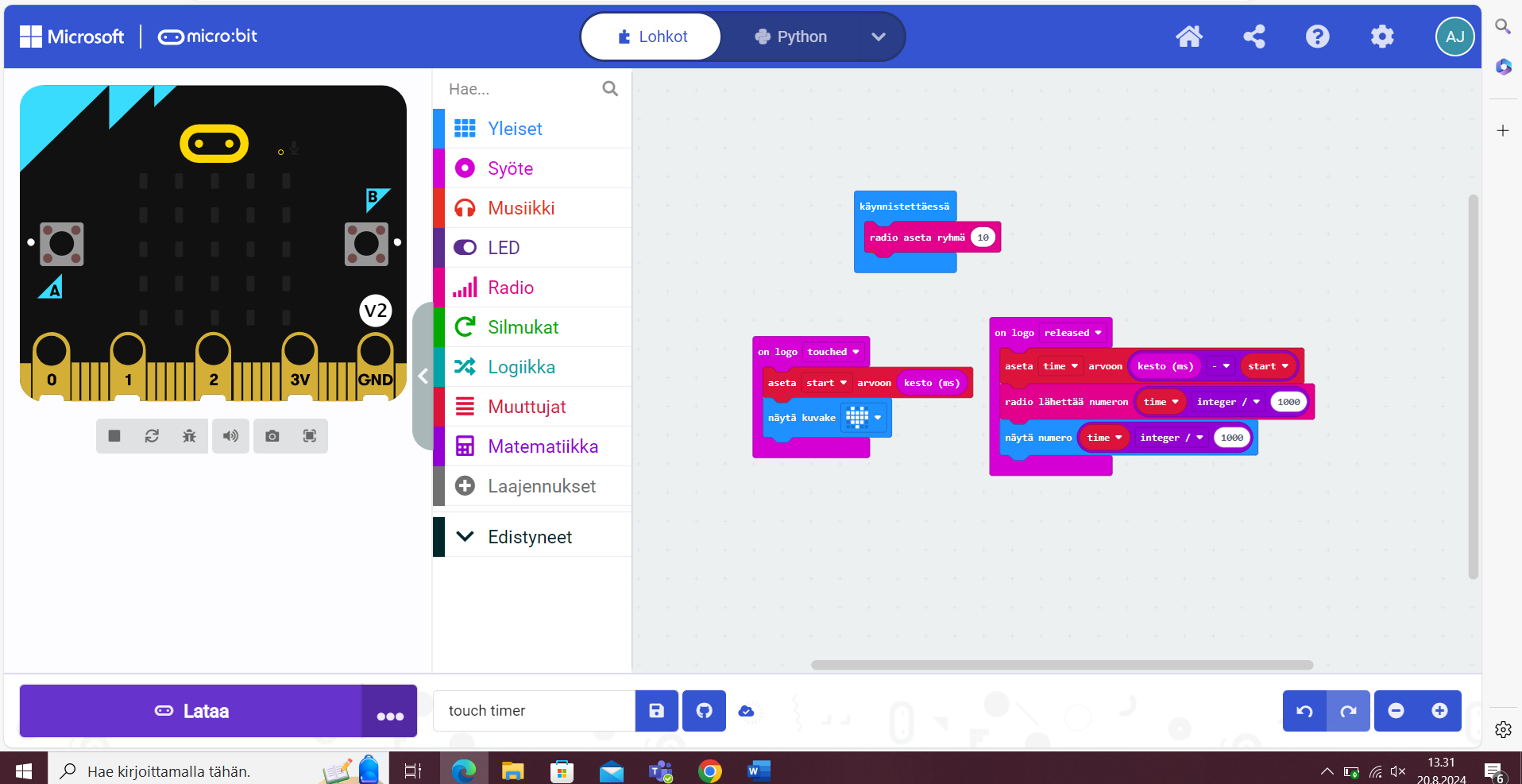
    basic.show\_number(Math.idiv(time, 1000))

input.on\_logo\_event(TouchButtonEvent.RELEASED, on\_logo\_released)

time = 0

start = 0

radio.set\_group(10)



radio.set\_group(10)

def on\_forever():

    radio.send\_number(input.temperature())

    basic.pause(5000)

basic.forever(on\_forever)

