Micro:bit

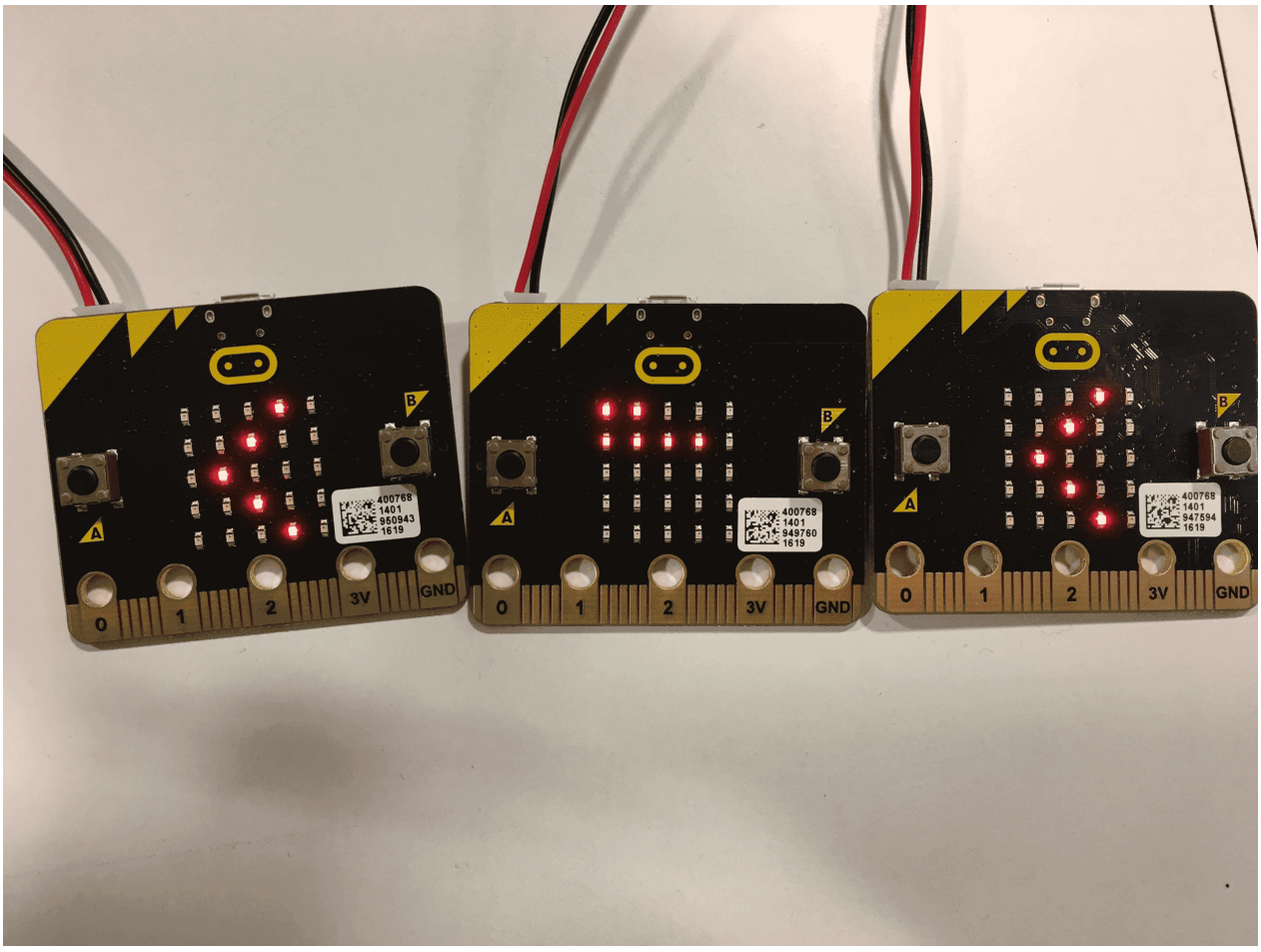
PXT: Hermegåsa

Skrevet av: Felix Bjerke og Tjerand Silde

Kurs: Microbit
Tema: Elektronikk, Blokkbasert, Spill
Fag: Programmering, Teknologi
Klassetrinn: 5.-7. klasse, 8.-10. klasse, Videregående skole

Introduksjon

Hermegåsa er et spill der en person er *spilleder*, og går ut på at han utfører instruksjoner på micro:biten sin som de andre spillerene skal gjenta, altså herme etter, på sin micro:bit. Dette er et multiplayer-spill, og det er om å gjøre å være den raskeste spilleren til å herme etter spillederen. Spillederen har oversikt på skjermen sin over poengene til de ulike spillerene, og spillet avsluttes når en spiller har vunnet 5 ganger.



På bildet ser du at spillederen er i midten og har poengsummen til spiller 1 og spiller 2 på skjermen sin. På sidene har spillerne fått beskjed om hva de skal gjøre, i form av en pil på skjermen. Førstemann som vipper micro:biten sin sammen vei som pilen vinner.

Steg 1: Sjekk at du har riktig utstyr

Det er viktig at du har alt utstyr og tilbehør for å kunne gjøre denne oppgaven.

Sjekkliste

- [ ] 1 micro:bit med en micro-usb-kabel som spilleder.
- [ ] 2-5 micro:biter med strømforsyning (micro-usb-kabler eller batterier).
- [ ] En datamaskin med Internett.

Spillregler

Her er en oversikt over reglene i spillet:

Skjerm viser	Spiller utfører
A	Trykk på A
B	Trykk på B
C	Trykk på A + B
<	Vipp til venstre
>	Vipp til høyre

Steg 2: Programmere spillederen

Når spillet skal starte så må vi sørge for at alle micro:bitene kan kommunisere med hverandre, og da må vi bestemme et gruppe-nummer som alle deltagere bruker når man sender informasjon over Bluetooth. I tillegg må vi lagre at vi har sendt ut en melding til deltagere, slik at vi ikke sender ut flere meldinger før vi har fått svar på den forrige.

Sjekkliste

- [ ] Start et nytt PXT-prosjekt, for eksempel ved å gå til makecode.microbit.org (https://makecode.microbit.org/?lang=no) .

- [ ] Finn blokken `ved start` under *Basis* og dra den inn i kodefeltet.

- [ ] Gå til *Radio* og finn blokken `radio sett gruppe` . I utgangspunktet står det tallet `1` i denne blokken, men du bestemmer selv hvilket tall du vil bruke.

- [ ] Til slutt går vi til *Variabler* og velger blokken `sett variabel til` . Denne kan du bytte navn på ved å klikke på *variabel* og så på *Rename variable*. Den bør hete *MessageSent*.

- [ ] Til *MessageSent* kobler vi blokken `usann` , som du finner under *Logikk*.

- [ ] Når spillet starter skal alle spillerne ha `0` poeng. Lag en variabel for hver spiller med navn `Player1` , `Player2` osv. som du setter til `0` poeng.

- [ ] Dersom du har gjort alt rett så vil koden din se slik ut:

OBS!

Husk at gruppenummeret du har valgt må brukes alle steder i spillet hvor du skal bestemme gruppenummer. I denne oppgaven har vi valgt tallet `37` .

I oppgaven antar vi at vi har fem spillere. Dersom dere er færre spillere enn det så trenger du ikke legge til så mange spillere.

Steg 2: Vise poeng på skjermen

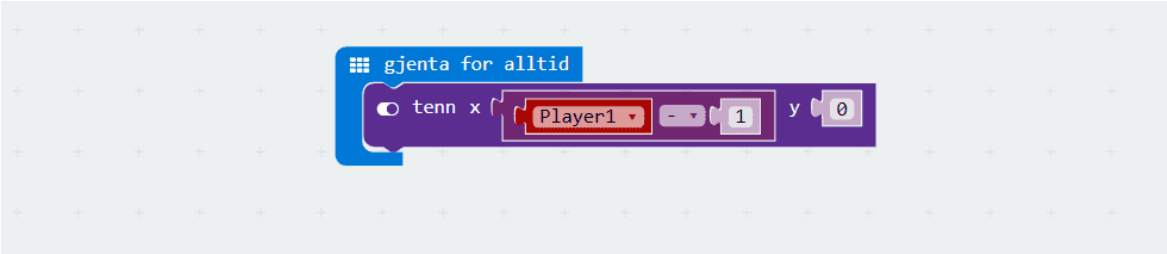
For å holde oversikt over hvor mange poeng de forskjellige spillerne har så viser vi dette på skjermen til spillederen. Dersom noen av de får 5 poeng så gir vi beskjed om at den spilleren vant.

Sjekkliste

- [ ] Først legger vi til `gjenta for alltid`-blokken fra *Basis*.

- [ ] Vi vil vise poengsummen til hver spiller i radene på skjermen til spillederen. `Player1` vises i rad `1`, `Player2` vises i rad `2` osv. Legg til blokken `tenn x y` fra *Skjerm*. La `y` være lik `0`.

- [ ] Vi må sette `x` til å være lik poengsummmen til spillerne. Fordi koordinatene til skjermen starter på `(0,0)` og *ikke* `(1,1)` så må vi trekke fra ett poeng hele tiden, slik at lysene sier hvor mange poeng spilleren har. Det kan gjøres slik:

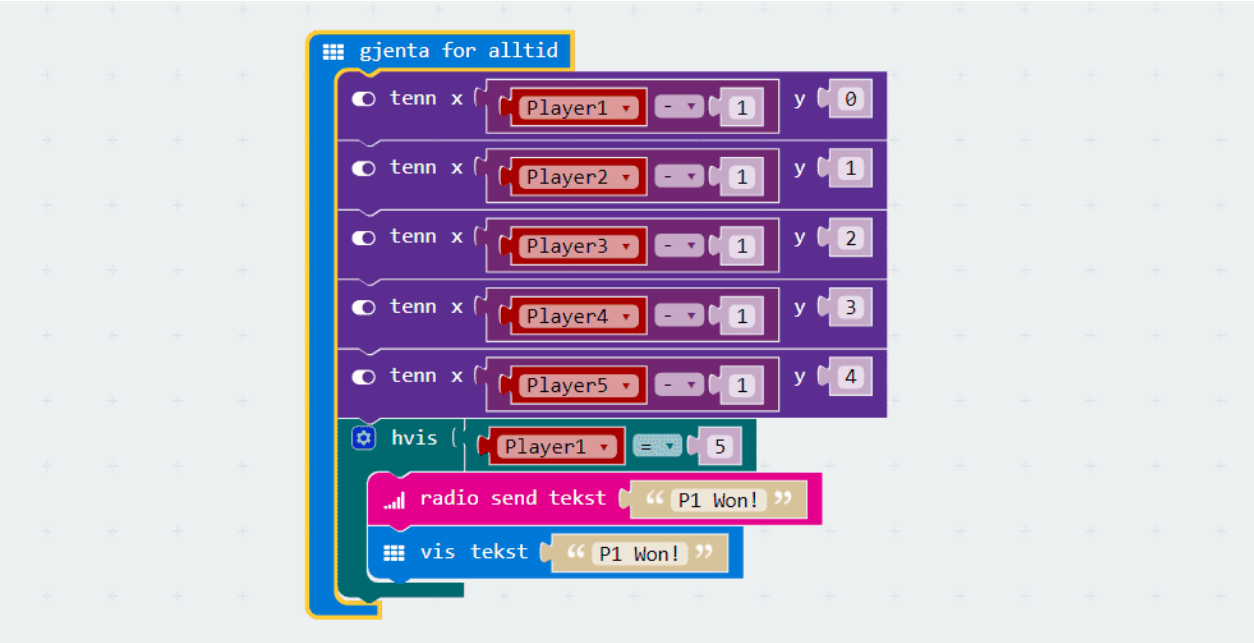


- [ ] Gjør det samme for `Player2`, `Player3`, `Player4` og `Player5`.

- [ ] Legg til en `hvis-blokken` og sjekk om `Player1` er lik `5` poeng. Da har spilleren vunnet.

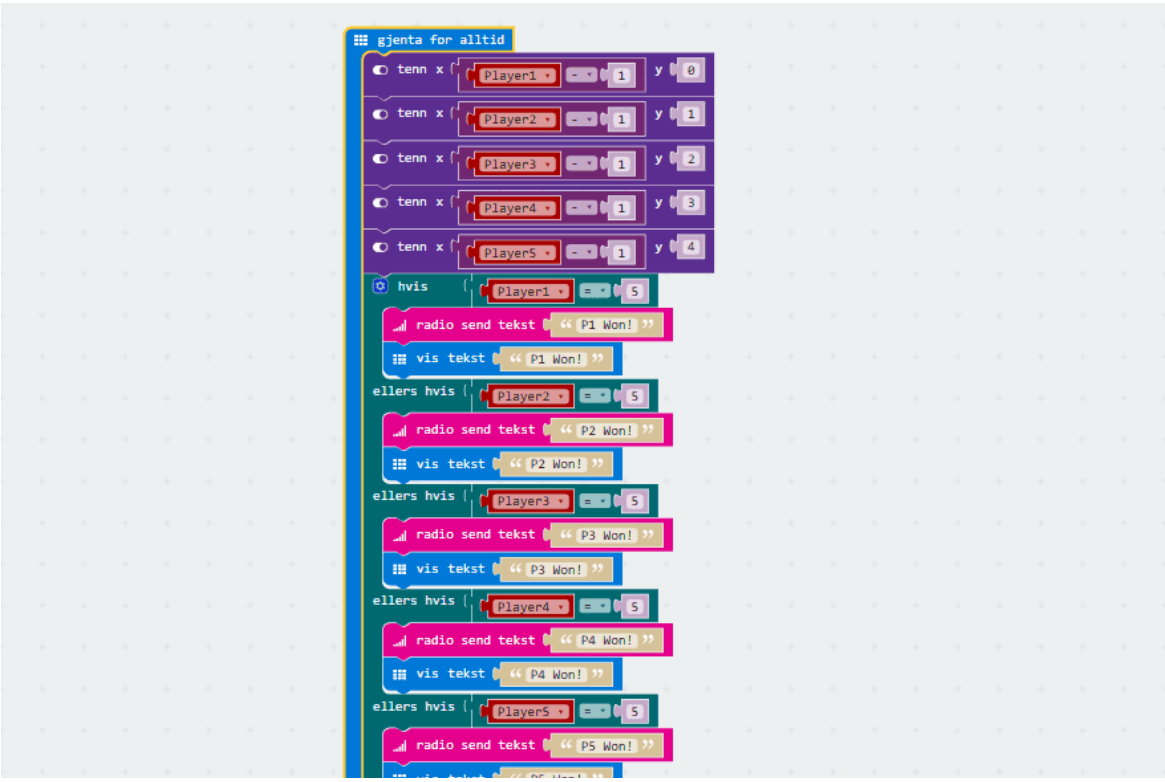
- [ ] Dersom `Player1` har `5` poeng så skal vi sende ut beskjed til alle om dette. Først legger vi til `radio send tekst`, og så legger vi til `vis tekst`. Sett begge

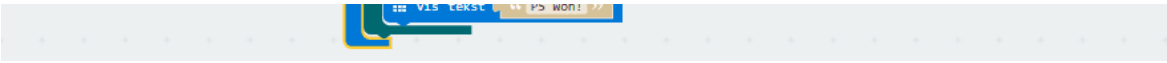
tekstene til *P1 Won!*. Da ser koden slik ut:



- [ ] Utvid `hvis-blokken` til `hvis - ellers hvis - ellers hvis - ellers hvis - ellers hvis`, slik at du kan sjekke poengene til alle spillere, og gjør det samme for `Player2`, `Player3`, `Player4` og `Player5` dersom de har vunnet istedenfor.

- [ ] Dersom du har gjort alt rett så vil koden din se slik ut:



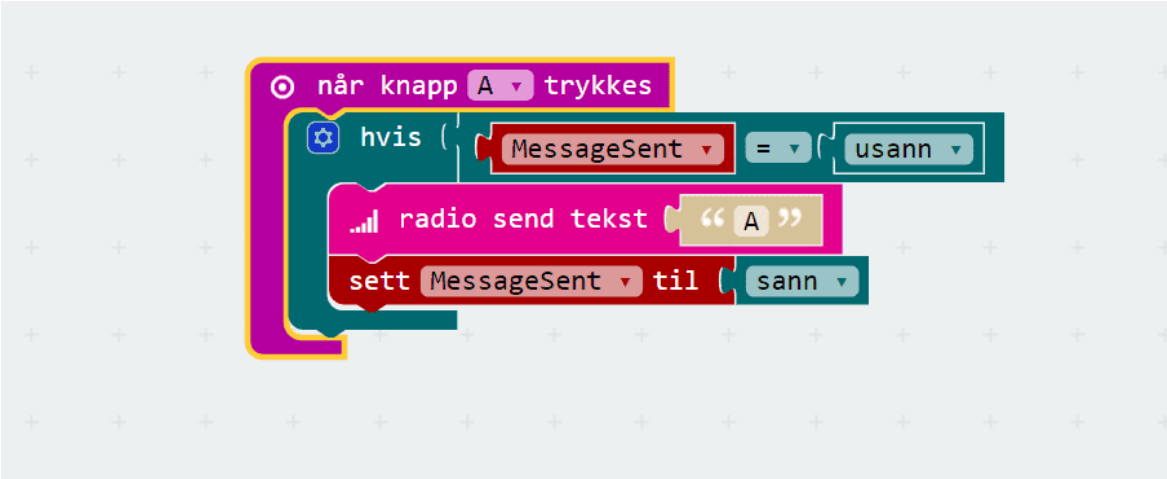


Steg 3: Sende ut meldinger til deltagerne

Så må vi sende ut meldinger til deltagere om hva de skal herme etter. Vi lager en blokk for hver av de forskjellige meldingene, ut ifra reglene beskrevet over.

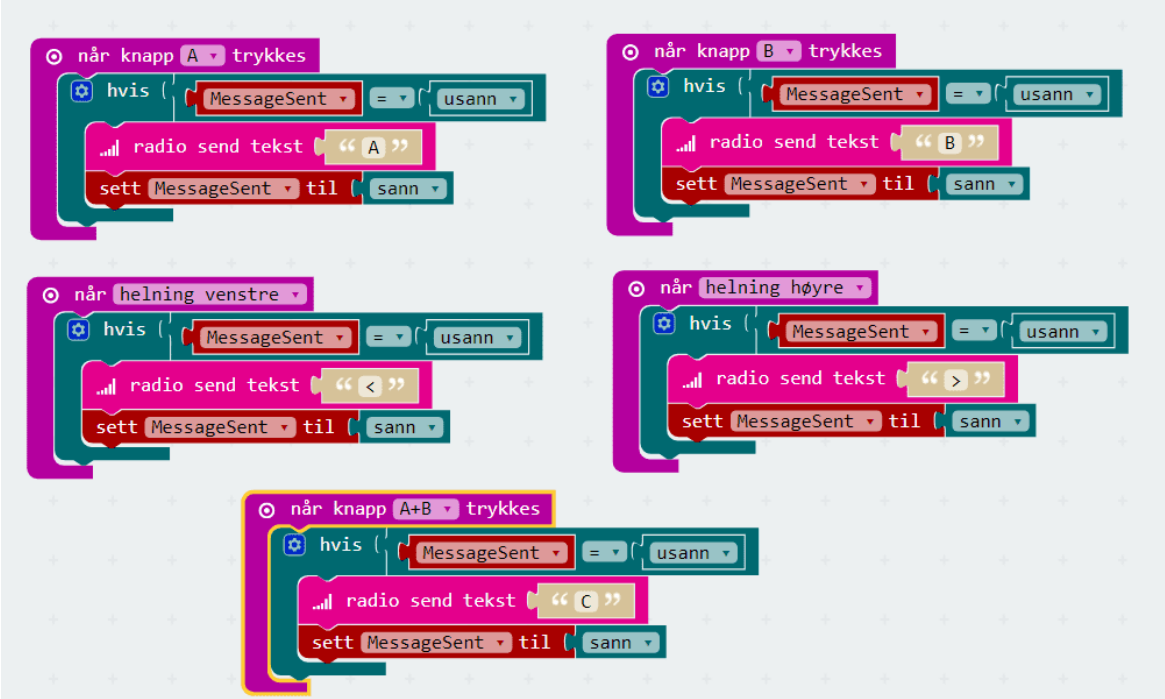
Sjekkliste

- [ ] Først må vi finne blokken som heter `når knapp A trykkes`. Den ligger under *Inndata*.
- [ ] Så henter vi `hvis-blokken` fra *Logikk* og setter den sammen med `erlik-blokken (=)` fra samme kategori. Her skal du sjekke om `MessageSent` er lik `usann`. Klarer du å sette dette sammen?
- [ ] Inni `hvis-blokken` setter vi `radio send tekst` med teksten `A`.
- [ ] Til slutt legger vi også til blokken `sett MessageSent til` fra *Variabler* og setter verdien til `sann`.
- [ ] Dersom du har gjort alt rett så vil koden din se slik ut:



- [ ] Gjør det samme for `B`, `C`, `<` og `>`. Sjekk tabellen ovenfor for å se hva som skal gjøres.

- [ ] Dersom du har gjort alt rett så vil koden din se slik ut:



Steg 4: Sjekke svaret til deltagerne

Etter hver runde må spillederen sjekke hvem av deltagerne som var raskest til å herme.

Sjekkliste

- [ ] Finn `når radio mottar receivedNumber` fra *Radio*.

- [ ] Sjekk om `MessageSent` er lik `sann`.

- [ ] Dersom `MessageSent` er lik `sann` så skal du sjekke om `receivedNumber` er lik `1`, for da har `Player1` vunnet denne runden. Vi får altså en `hvis-blokk` inni en annen `hvis-blokk`.

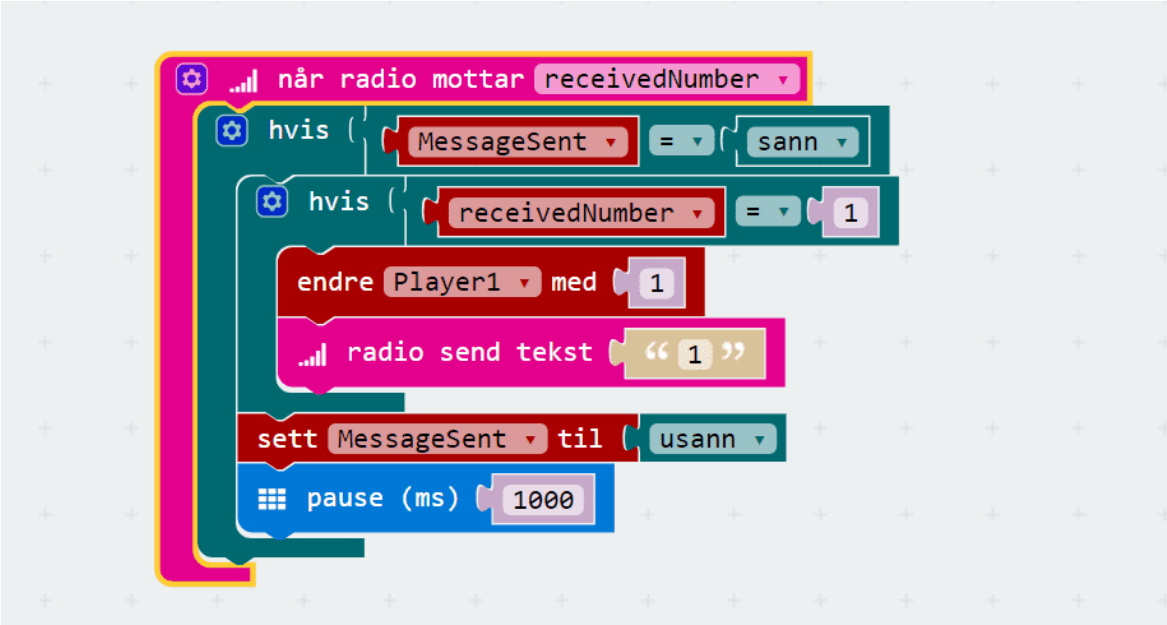
- [ ] Dersom `receivedNumber` er lik `1` så tar vi blokken `endre Player1 med 1` fra *Variabler* inni den innerste `hvis-blokken`.

- [ ] Under `endre Player1 med 1` legger du til `radio send tekst` med tallet `1`. Da sender vi beskjed til alle om hvem som vant. Legg også til `vis tall` med tallet `1`. Da viser vi det også på spillederen sin micro:bit.

- [ ] Så legger vi til en pause på `1` sekund mellom de to `hvis-blokkene` før vi går videre til neste runde. Dette gjør du ved å legge til blokken `pause (ms)` og setter den til `1000`. Tusen millisekund er det samme som ett sekund.

- [ ] Til slutt endrer vi `MessageSent` til `usann`.

- [ ] Dersom du har gjort alt rett så vil koden din se slik ut:



- [ ] Utvid `hvis-blokken` til `hvis - ellers hvis - ellers hvis - ellers hvis - ellers hvis`, slik at du kan sjekke om det var noen av de andre spillerne som vant denne runden, og gjør det samme for `Player2`, `Player3`, `Player4` og `Player5` dersom de har vunnet istedenfor.

- [ ] Dersom du har gjort alt rett så vil koden din se slik ut:

Steg 5: Overføre programmet til spillederen

- [ ] Gi navnet `spilleder` til programmet ditt. Dette gjør du ved å endre teksten helt nede på midten på skjermen din.

- [ ] Koble den enen micro:biten din til datamaskinen med en USB-kabel. Denne micro:biten blir spillederen i spillet.

- [ ] Klikk deretter på knappen [illegible] på skjermen. Det lastes på

- [ ] Klikk deretter på knappen `Last ned` nede til venstre på skjermen. Det lastes da ned en fil som heter `spilleder.hex` til datamaskinen din.

- [ ] Flytte denne filen over til MICROBIT-disken på maskinen din.

Steg 6: Programmere spillerne

Først må vi sørge for at spillerene kobler seg til spillederen og gir de et spillernummer, slik at vi kan sjekke hvem som vinner.

Sjekkliste

- [ ] Start et nytt PXT-prosjekt.

- [ ] Finn blokken `ved start` under *Basis* og dra den inn i kodefeltet.

- [ ] Gå til *Radio* og finn blokken `radio sett gruppe` . Husk å fylle inn det samme tallet her som du gjorde på `spilleder` .

- [ ] Gå så til *Variabler* og velger blokken `sett variabel til` . Denne kan du bytte navn på ved å klikke på *variabel* og så på *Rename variable*. Den bør hete *Player*. Sett verdien til `1` .

- [ ] Til slutt viser vi tallet vårt til skjerm, ved å hente blokken `vis tall` og koble den til variabelen `Player` .

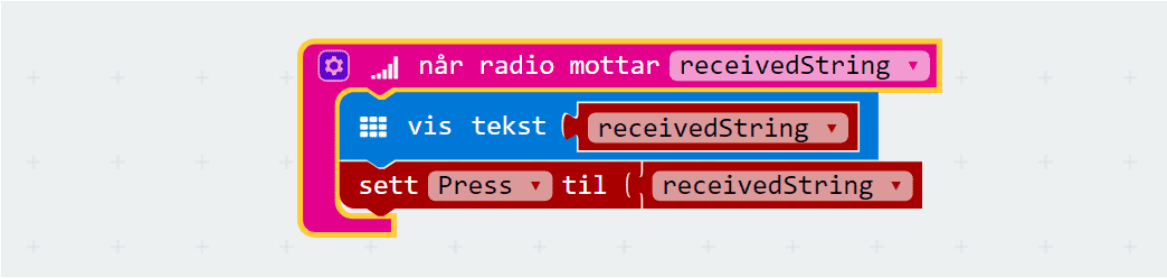
- [ ] Dersom du har gjort alt rett så vil koden din se slik ut:

Steg 7: Når spiller mottar melding

Neste steg er å motta meldingen om hva vi skal herme etter, og lagre den. Da kan vi senere sjekke om vi har gjort rett.

Sjekkliste

- [ ] Finn `når radio mottar receivedString` fra *Radio*.
- [ ] Vis teksten `receivedString` på skjermen.
- [ ] Opprett en ny variabel med navn `Press` og sett denne til `receivedString`. Da har vi lagret dette til å sjekke om vi har klart å herme etter spillederen.
- [ ] Dersom du har gjort alt rett så vil koden din se slik ut:



Steg 8: Når spiller mottar melding

Siste steg er å herme etter spilleder og sende beskjed i retur.

Sjekkliste

- [ ] Vi starter med å herme etter spilleder dersom meldingen var `A`. Først må vi finne blokken som heter `når knapp A trykkes`. Den ligger under *Inndata*.
- [ ] Så henter vi `hvis-blokken` fra *Logikk* og setter den sammen med `erlik-`

`blokken (=)` fra samme kategori. Her skal du sjekke om `Press` er lik `A`.

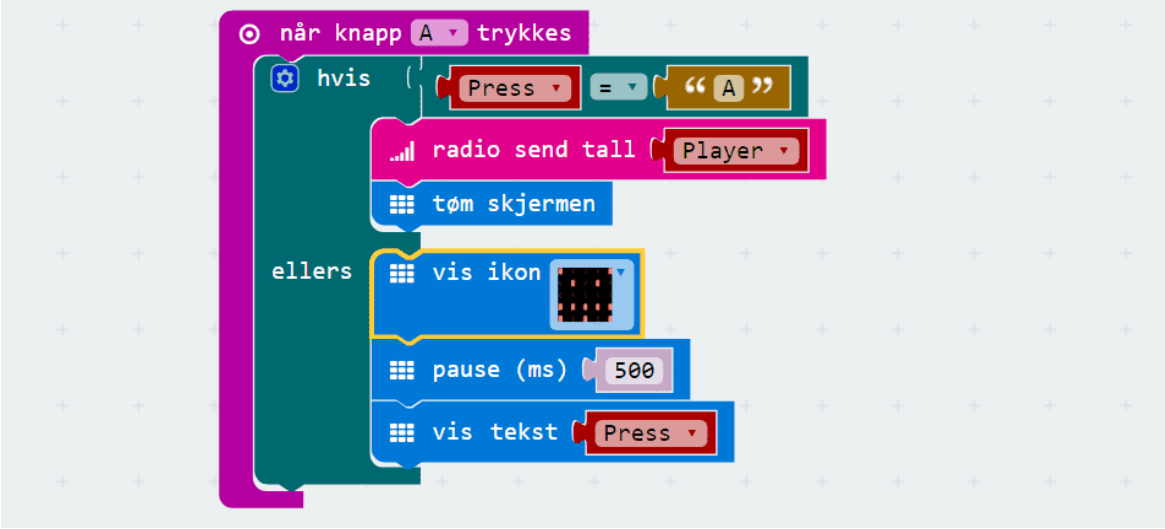
- [ ] Dersom `Press` er lik `A` så skal vi sende spillernummeret vårt tilbake til spilleder. Legg til blokken `radio send tall` inni `hvis-blokken` og koble den til `Player`. Legg også til blokken `tøm skjermen` fra *Basis*.

- [ ] Utvid `hvis-blokken` til `hvis - ellers`.

- [ ] `Press` ikke er lik `A` så skal man vise et surt fjes. Du finner blokken `vis ikon` under *Basis*. Velg et passende ikon.

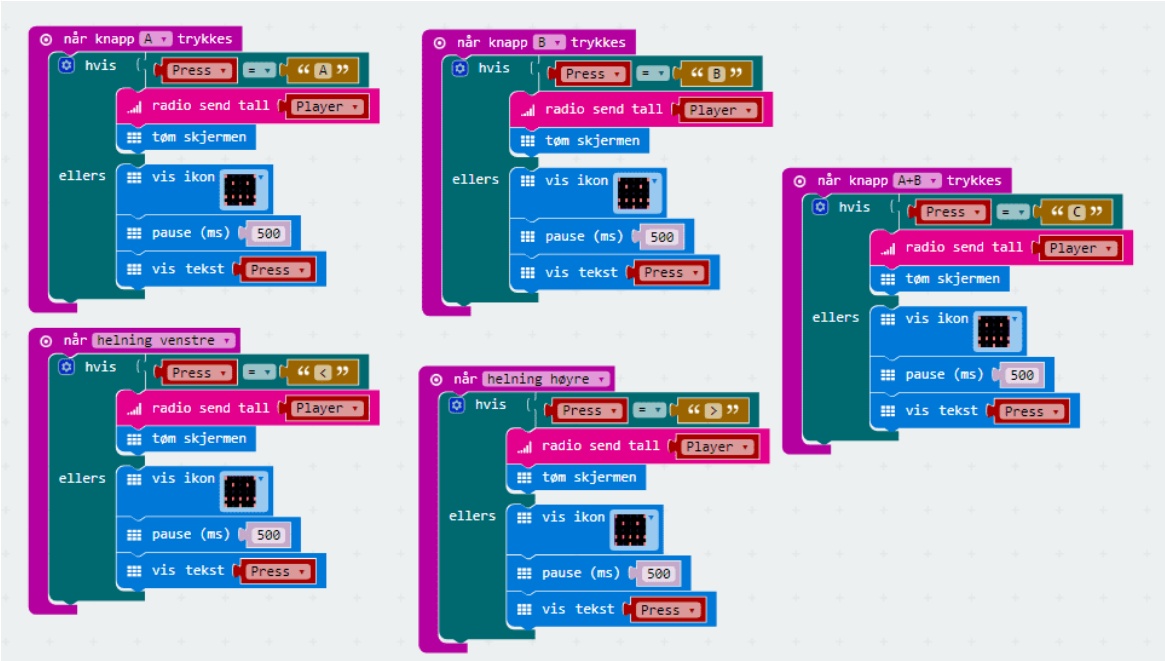
- [ ] Legg også til `pause` i `500` ms. Til slutt kan du finne blokken `vis tekst` og legge variabelen `Press` inni den, for å vise hva du egentlig burde gjort.

- [ ] Dersom du har gjort alt rett så vil koden din se slik ut:



- [ ] Gjør tilsvarende for `B`, `C`, `<` og `>`.

- [ ] Dersom du har gjort alt rett så vil koden din se slik ut:



Steg 9: Overføre programmet til spillerne

Overfør programmet til hver av spillerne sine micro:biter. Husk å endre programmet med spillernummer mellom hver gang og last ned på nytt.

Sjekkliste

- [ ] Gi navnet `spiller` til programmet ditt. Dette gjør du ved å endre teksten helt nede på midten på skjermen din.
- [ ] Koble den enen micro:biten din til datamaskinen med en USB-kabel. Denne micro:bit'en blir spillederen i spillet.
- [ ] Klikk deretter på knappen `Last ned` nede til venstre på skjermen. Det lastes nå ned en fil som heter `spiller.hex` til datamaskinen din.
- [ ] Flytte denne filen over til MICROBIT-disken på maskinen din.

- [ ] Endre spillernummer i programmet og gjør samme prosess for hver av spillerne som skal være med.

OBS!

Husk å endre hvilket nummer hver spiller er mellom hver nye microbit du kobler til og laster ned programmet til. Du må laste ned programmet på nytt for hver av spillerne med de endringene du har gjort.

Utfordringer

- [ ] Legge til flere elementer som spillerne skal reagere på. For eksempel dersom micro:biten ristes.
- [ ] Endre koden til spilleder slik at micro:biten automatisk velger helt tilfeldig hva spillerne skal herme etter.
- [ ] Endre koden slik at det er mulig å jukse i spillet for spillerne. Kan du gjøre slik at de alltid svarer rett? Kan du gjøre slik at en spiller alltid svarer raskest?
- [ ] Endre koden til spillederen slik at han sjekker at spillerne faktisk har svart rett, for å passe på at de ikke har jukset. Vær obs på at du da også må endre på spillerne sin kode.

Sikkerhet med server og klient

Vi har nå laget en applikasjon med server (spilllederen) og klienter (spillerene). Spillerne kommuniserer gjennom hver sin klient, men koordinerer (teller hvem som vinner) på en server.

Sikkerhet i spillet vårt

Hver klient kan i prinsippet bestemme hvilken kode som skal kjøre på sin egen enhet. Dette gjør det mulig å jukse! For å unngå at det er mulig å jukse må vi tenke oss om for hvordan vi koder serveren vår og hvordan vi koder klienten vår.

Å la klienten sende tilbake "jeg hadde rett" er ikke sikkert. Da kan jeg lage min egen klient som alltid gir meg alle poengene, uansett om jeg har svart rett eller galt.

Å la klienten sende tilbake "jeg svarte B" er bedre. Da må jeg faktisk vite svaret for å kunne få poeng. *Hva tror du skjer dersom du gjetter alle svarene hele tiden? Klarer du tenke på en måte å unngå at dette er mulig?*

Sikkerhet på Facebook

Facebook er en annen applikasjon som er delt inn i klient og server.

Klienten viser bildene dine og vennene dine. Du ser klienten når du går til `facebook.com` i en nettleser. Når du bruker Facebook-appen på en mobiltelefon, bruker du en annen klient.

Serveren holder styr på all informasjonen. Serveren kan si om du prøver å logge inn med rett brukernavn og passord. Serveren vet hvem som er vennene dine, og den vet hvem som kan administrere hvilke sider og grupper.

Hva hadde skjedd dersom gruppeadministrasjon ble håndtert i klienten? Hvordan kunne du da laget din egen klient? Hva kunne du gjort med denne?