

△ PXT: Stein, saks, papir

↓ LAST NED PDF

Introduksjon

Her skal du lære å programmere micro:biten slik at du kan spille stein, saks, papir med den eller mot den.

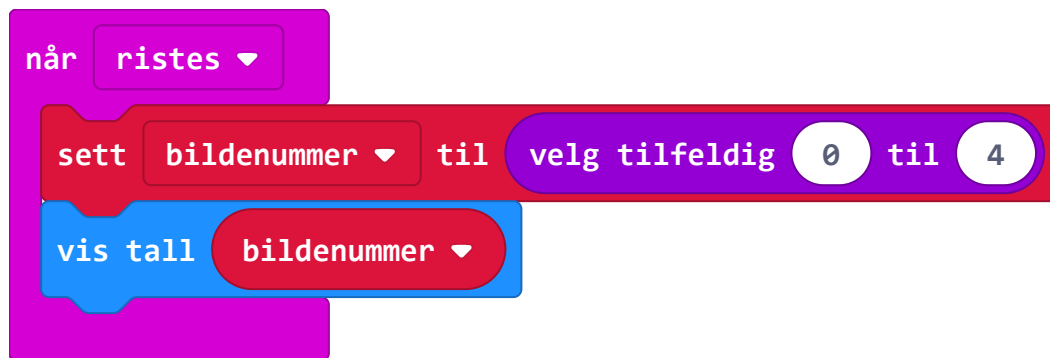
Steg 1: Velge et tilfeldig tall

I første del av oppgaven skal vi få micro:biten til å velge et tilfeldig tall når den ristes. For at du senere skal kunne bruke dette tallet, må tallet lagres i en variabel.



Sjekkliste

- ☐ Start et nytt PXT-prosjekt, for eksempel ved å gå til makecode.microbit.org.
- ☐ Micro:biten skal velge et tilfeldig tall når du rister på den. Til dette skal du bruke `når ristes-klossen` som finnes i kategorien `Inndata`.
- ☐ Deretter trenger du en variabel som skal lagre det tilfeldige tallet i. Klikk på `Variabler-kategorien` og deretter på knappen `Lag en variabel`. Den nye variabelen kan hete hva som helst, men her velger vi å kalle den `bildenummer` og klikk `OK`. Vi velger å kalle den `bildenummer`, fordi vi senere vil at tallet som velges skal representere et bilde: enten stein, saks eller papir. Du vil se at det dukker opp en kloss som heter `bildenummer` i `Variabler-kategorien`.
- ☐ Nå vil vi at den nye variabelen vi opprettet skal få en tilfeldig verdi. Plasser `sett variabel til 0-klossen` fra `Variabler-kategorien` inne i `når ristes-klossen` vi fant tidligere. Trykk på den lille pilen bak `variabel` og endre til den nye variabelen du lagde - `bildenummer`.
- ☐ For å få micro:biten til å velge et tilfeldig tall hver gang vi spiller, kan vi bruke klossen `velg tilfeldig 0 og 4` fra `Matematikk-kategorien`. Koble denne til `sett bildenummer til 0-klossen` i stedet for tallet `0`.
- ☐ For å vise hvilket tall som ble valgt kan du sette sammen klossen `vis tall 0` fra `Basis-kategorien`, og erstatte `0` med den nye variabelen `bildenummer`. Denne settes sammen med de andre klossene slik at programmet ditt ser slik ut:



Test prosjektet

Det er to forskjellige måter du kan teste micro:bit-programmer på:

- ☐ Til venstre på skjermen er det et bilde av en micro:bit. Dette er en simulator som kan kjøre programmet ditt uten at du trenger å laste det og overføre det til din micro:bit:

Siden din kode skal reagere når du rister på micro:biten kan du simulere dette ved å klikke på den hvite prikken til venstre for teksten `SHAKE` på micro:bit-simulatoren. Det tilfeldige tallet som ble valgt skal vises på skjermen til micro:bit-simulatoren. Prøv flere ganger og se at tallet forandrer seg.

- ☐ Enda morsommere er det å teste programmet på micro:biten din! Koble micro:biten din til datamaskinen med en USB-kabel. Gi prosjektet ditt et navn, for eksempel `stein-saks-papir` i feltet til høyre for `Last ned`-knappen nede i venstre hjørne av skjermen. Trykk på `Last ned`-knappen for å laste ned programmet.

Det lastes nå ned en fil som heter `stein-saks-papir.hex` til datamaskinen din. Samtidig dukker det opp et vindu som sier at du må flytte denne filen til MICROBIT-disken. Dersom du trenger hjelp til dette så spør en av veilederne.

Steg 2: Vise tallet som ble tilfeldig valgt

Micro:biten skal vise om det er stein, saks eller papir som ble valgt.

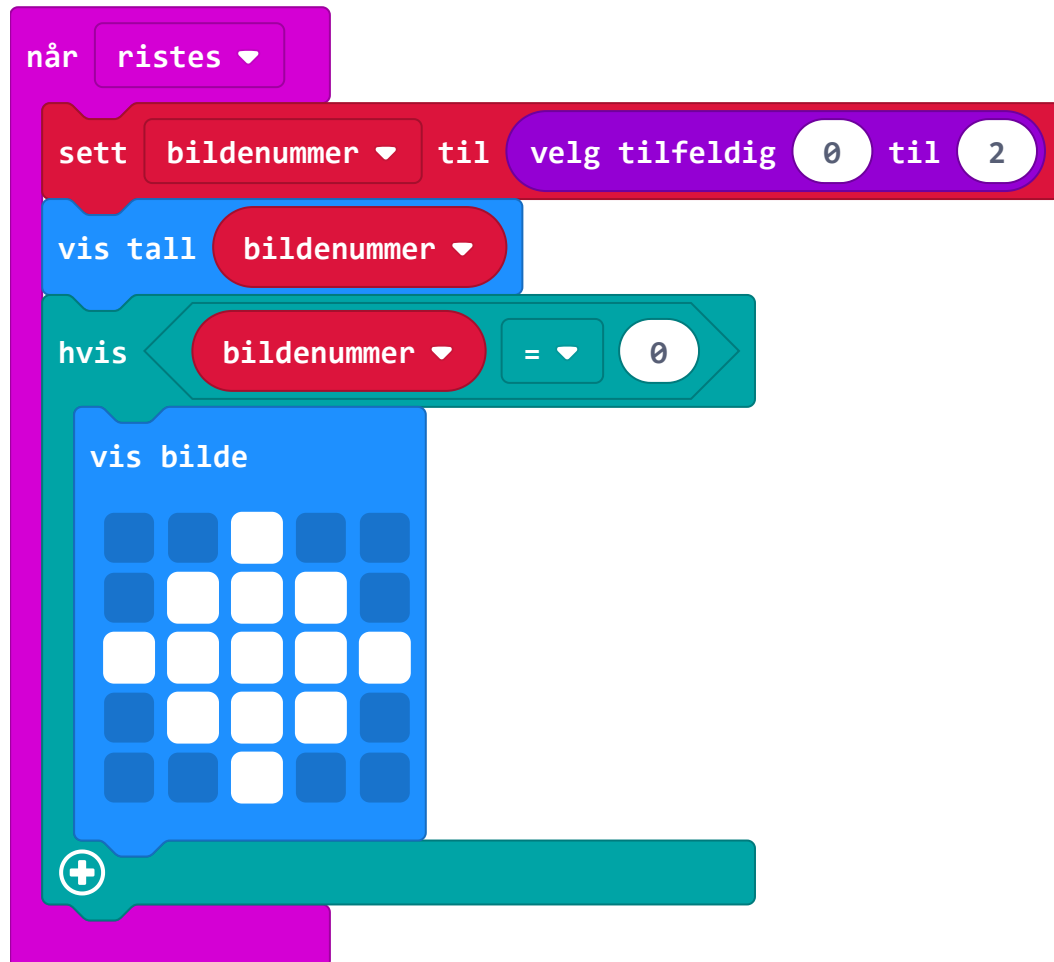


Sjekkliste

- ☐ Du har bare tre mulige trekk å velge mellom: stein, saks eller papir. Siden klossen som velger et tilfeldig tall starter å telle på null, må vi endre den slik at den enten velger 0, 1 eller 2. Klossen som velger tilfeldig tall, må derfor velge et tall mellom 0 og 2.
- ☐ Variabelen `bildenummer` skal nå inneholde en av verdiene 0, 1 eller 2. La 0 være stein, 1 være saks og 2 være papir.
- ☐ Vi vil nå vise forskjellige bilder på skjermen, avhengig av hvilket tall som ble valgt. Vi trenger derfor en `hvis-sann`-kloss fra kategorien `Logikk`. Vi plasserer denne nederst i blokken fra Steg 1.
- ☐ Den nye `hvis-sann`-klossen skal brukes til å sammenligne det tilfeldige tallet med 0, 1 og 2 for å kunne vise riktig bilde på skjermen. Vi vil derfor bytte ut `sann` med en ny kloss. Dette er `0 = 0-`

klossen fra Logikk-kategorien.

- ☐ Du skal sammenligne og se om variabelen `bildenummer` er lik 0. For å få til dette henter vi ut en ny kloss med `bildenummer` fra kategorien Variabler, slik at vi nå har en blokk som sier `hvis-bildenummer = 0`.
- ☐ Dersom variabelen `bildenummer` inneholder verdien 0 ønsker vi å vise et bilde av en stein, som bestemt over. Velg klossen `vis ikon` fra Basis-kategorien og velg et bilde som skal representere en stein. Eventuelt kan du lage bildet selv med `show leds`-klossen. Plasser klossen inne i `hvis-bildenummer = 0`-blokken



Test prosjektet

Nå skal du teste programmet ditt ved å trykke på den hvite `SHAKE`-knappen.

- ☐ Hvis tallet 0 velges, vises 0 en kort stund før det erstattes av bildet tegnet av en stein.
- ☐ Hvis tallet 1 eller 2 velges, vises kun det valgte tallet. Resten av programmet skal du lage nå.

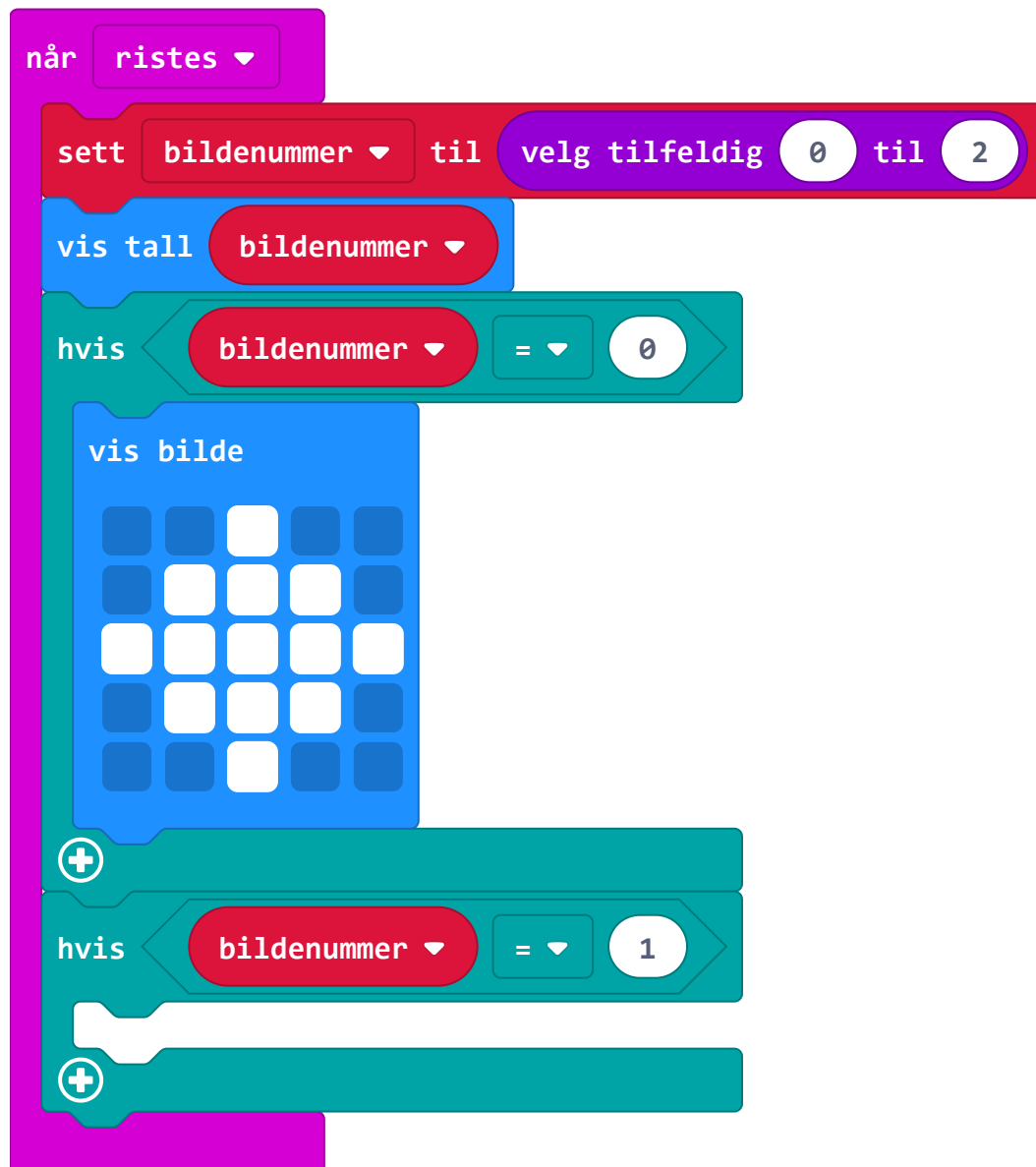
Steg 3: Vise saks

Tegne saks når tallet 1 blir valgt.



Sjekkliste

- ☐ Du trenger en ny `hvis`-kloss for å tegne en saks når tallet 1 er valgt. Denne plasseres rett under den forrige `hvis`-klossen.
- ☐ På samme måte som du gjorde med `stein`, skal du bruke en `0 = 0`-kloss, men erstatte det venstre `0`-tallet med variabelen `bildenummer`.
- ☐ Du skal vise en saks når variabelen `bildenummer` inneholder tallet 1, så nå må det høyre `0`-tallet erstattes med et `1`-tall. Programmet ser nå slik ut:



- ☐ Det finnes et ferdig bilde av en saks som du kan bruke. Gå til kategorien `Basis` for å finne `vis ikon`-klossen. Ved å trykke på den lille pila, kan du endre hvilket ikon som skal vises.

ved start

vis ikon



Test prosjektet

Før du går videre er det på tide å teste programmet igjen. Det skal vise bildet av stein hvis 0 blir valgt og saks hvis tallet er 1. Når tallet 2 blir valgt vil det ikke vises noe bilde ennå.

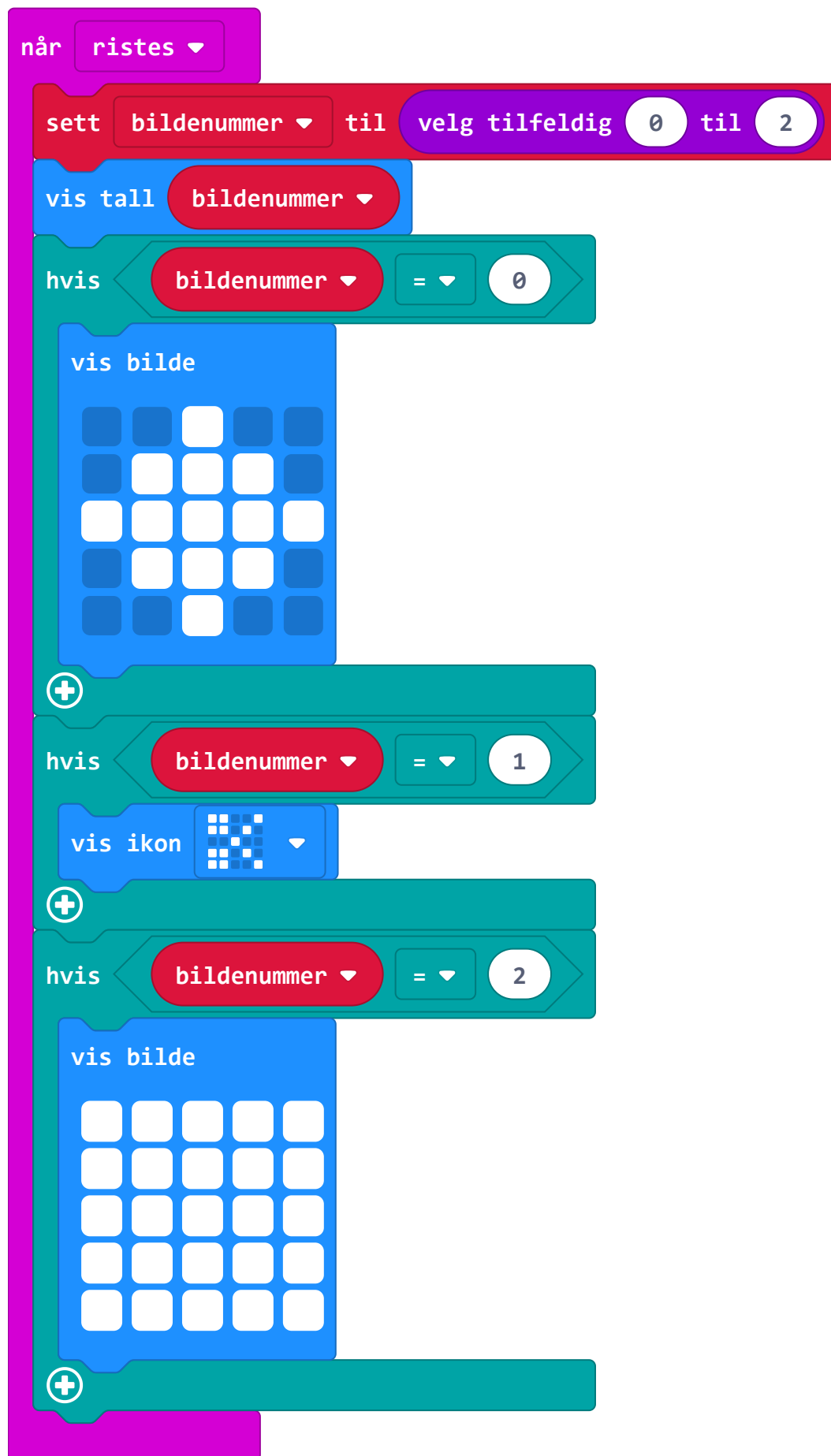
Steg 4: Vise papir

Å Tegne papir når variabelen `bildenummer` har verdien 2 blir veldig likt steg 1 du gjorde tidligere.



Sjekkliste

- ☐ Fremgangsmåten for å lage en `hvis`-blokk for når tallet 2 velges, er akkurat den samme som får både stein og saks. Prøv å lage denne selv. *Husk: Du må bytte ut begge 0-ene i denne blokken også.*
- ☐ Det ferdige programmet skal nå se slik ut:



Test prosjektet

Nå kan du teste programmet ditt. Det skal nå vise enten stein, saks eller papir avhengig av hvilket tall som blir valgt når du trykker på `SHAKE`-knappen.

- ☐ Dersom du laster ned programmet på micro:biten din, bør du koble til en strømkilde (spør veilederne) slik at du ikke er uheldig og river ned noe mens du rister på den.

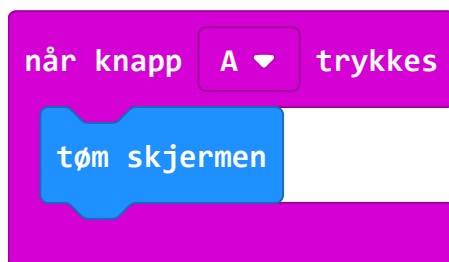
Steg 5: Tømme skjermen mellom hvert spill

Når `micro:biten` skal velge et tilfeldig tall, kan det samme tallet blir valgt to ganger på rad. Det kan derfor være vanskelig å vite om det samme tallet ble valgt igjen, eller om du ikke ristet hardt nok. For å gjøre det enkelt å se, vil vi tømme skjermen mellom hvert spill.



Sjekkliste

- ☐ Gå til kategorien `Inndata` og velg klossen `når knapp A trykkes`, og plasser den for seg selv. Den skal ikke henge sammen med den andre kodeblokken.
- ☐ Inne i `når knapp A trykkes`-klossen legger du klossen `tøm skjerm`. Når `A`-knappen trykkes skal `micro:biten` slå av alle lysene slik at den er klar til nytt spill.
- ☐ Den nye programblokken ser da slik ut:



Test prosjektet

Nå kan du teste programmet ditt. Det skal nå vise enten stein, saks eller papir avhengig av hvilket tall som blir valgt når du trykker på `SHAKE`-knappen.

- ☐ Dersom du laster ned programmet på `micro:biten` din, bør du koble til en strømkilde (spør veilederne) slik at du ikke er uheldig og river ned noe mens du rister på den.

Utfordring

- ☐ Dersom du ikke ønsker å se tallet på skjermen før bildet tegnes, kan du fjerne `vis tall` `'bilenummer'`-blokken vi la inn i steg 1.
- ☐ Greier du å slå sammen de tre `hvis`-klossene til bare én?
- ☐ I denne oppgaven har vi brukt at et trykk på knapp `A` nullstiller skjermen til et nytt spill. Greier du å legge all koden inn i samme blokk ved å bruke for eksempel en `pause`-kloss og en `tøm skjerm`-kloss? Her er det ingen fasit, det er bare å prøve seg fram.

- ☐ Hva med å la micro:biten vise de forskjellige bildene av stein, saks og papir før den stopper på den det ble? *Tips: Bruk klossen* `gjenta for bildenummer 0 til 2` *til å først vise alle én gang. Bruk den samme blokken igjen for å vise bildenummer fra 0 og opp til bildenummeret som ble valgt (vi lagret det i variabelen `bildenummer`). I tillegg trenger du en kloss som viser bildene på skjermen som du gjorde i steg 2, 3 og 4!*

Lisens: CC BY-SA 4.0