



Straffespark

Introduksjon


Vi skal lage et enkelt fotballspill, hvor du skal prøve å score på så mar



Steg 1: Katten og fotballban

Vi begynner med å lage en katt som kan bevege seg på en fotballban

Sjekkliste

- ☐ Start et nytt prosjekt, for eksempel ved å klikke **Programmering** du begynne på et nytt ved å velge **Fil** og deretter **Ny**.
- ☐ Klikk på **i** i hjørnet av kattefiguren, og gi katten navnet **Leo**.
- ☐ Klikk på  nederst til venstre på skjermen for å hente inn en ny **Utendørs/goal2**.
- ☐ Vi skal nå skrive et lite program som flytter katten **Leo** når vi kl skjermen, og klikk **Skript**-fanen slik at du ser de forskjellige ko følgende klossene i skriptvinduet til høyre:



Test prosjektet

Klikk på det grønne flagget.

- ☐ Flytter **Leo** på seg når du klikker på ham?
- ☐ Hva skjer om du klikker mange ganger på **Leo** slik at han når b

Sjekkliste

- ☐ Du kan flytte **Leo** tilbake på skjermen ved å trykke på ham og c
- ☐ Vi vil likevel programmere en enklere måte å få **Leo** tilbake på allerede har laget, som ser slik ut:



- ☐ Nå vil **Leo** komme tilbake på banen hver gang du klikker det gr

Lagre prosjektet

Du har nå skrevet et lite program! Scratch lagrer alt du gjør med jevn også innimellom.

- ☐ Over scenen er det et tekstfelt hvor du kan gi et navn til spillet
- ☐ I menyen **Fil** kan du velge **Lagre nå** for å lagre prosjektet.

Steg 2: Vi sparker ballen

Det neste vi trenger i spillet vårt er en ball!

Sjekkliste


- ☐ Vi skal nå legge til en ball-figur i spillet vårt. Klikk på  og velg

- ☐ Gi fotballen navnet **Ball**.
- ☐ Lag ballen mindre ved først å trykke på  øverst mot midten a' ganger.
- ☐ Vi skal nå skrive et litt lengre program. I dette programmet legg ballen berører **Leo** (det vil si **Leo** sparker ballen) begynner ball programmet de forskjellige tingene skjer:



Test prosjektet

Klikk på det grønne flagget.

- ☐ Sparker **Leo** ballen når du klikker på ham?
- ☐ Kan **Leo** sparke ballen en gang til dersom du trykker det grønne
- ☐ Hvis **Leo** sparker ballen før du klikker på ham (med en gang du tallene i  -klossen slik at ballen ligger i ro foran **L**

Steg 3: Vi trenger en keeper

Vi skal nå gjøre spillet litt vanskeligere ved å programmere en keeper.

✓ Sjekkliste

- ☐ Legg til en figur til som kan være keeper. Vi har brukt **Dyr/Octo**
- ☐ Flytt figuren inn i målet. Om det ser ut som figuren står med ryggen på **Drakter**-fanen, og deretter på snu-knappen som er merket i



- ☐ Gi figuren navnet **Keeper**.

- ☐ Lag det følgende programmet for keeperen:



Det er et par nye klosser i dette skriptet. Les programmet nøye.

Test prosjektet

Klikk på det grønne flagget.

- ☐ Starter keeperen i målet?
- ☐ Beveger **Keeper** seg opp og ned på skjermen?

Sjekkliste

- ☐ En ting som ser litt rart ut er at **Keeper** ikke ser på ballen. Vi fikk -kloss helt i begynnelsen av skriptet vårt.
- ☐ Du synes kanskje at keeperen er litt stor? Vi har tidligere sett hv
annen måte å endre størrelsen på er ved å bruke klosser fra **Ut**

Legg klossen **sett størrelse til 100 %** inn i **for alltid**-løkka. Du l
tall til du finner en passende størrelse på keeperen. Om du har l
bra. Prøv deg frem!



Test prosjektet

Klikk på det grønne flagget.

- ☐ Ser keeperen i retning av **Leo** og ballen?
- ☐ Hva skjer etter at katten skyter mot mål? Kan keeperen redde b

Steg 4: Keeperen redder!

Nå er det på tide at keeperen redder ballen!

Vi skal nå lage noen tester som sier i fra når keeperen redder ballen, €



Sjekkliste

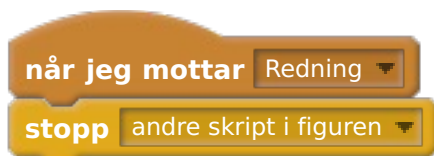
- ☐ Vi begynner med å finne ut når keeperen redder ballen. Klikk på



I klossen **send melding** `message1` må du velge **Ny melding** og de

Meldinger er beskjeder som figurene i spillet kan sende seg i mellom i mellom figurene å reagere på ting som skjer.

- ☐ Vi vil nå at både ballen og keeperen skal slutte å bevege seg ved **Ball** og **Keeper**:



Et triks er å først skrive denne koden for **Ball**. Deretter kan du figuren i figurvinduet nederst på skjermen.

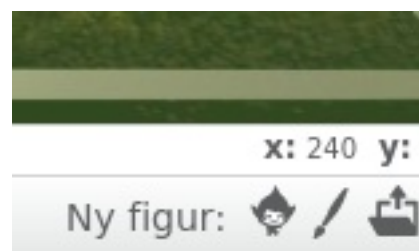
Test prosjektet

Klikk på det grønne flagget.

- ☐ Slutter både ballen og keeperen å bevege seg om keeperen red

✓ Sjekkliste

Nå skal vi også sjekke om ballen har gått i mål. Vi gjør dette ved å undersøke om ballen beveget seg. Posisjonen til en figur i Scratch er beskrevet ved hjelp av koordinater x og y. x er sidelengs på skjermen, mens **y-posisjon** sier hvor langt opp eller ned på skjermen. Disse viser koordinatene til musepekeren.



- ☐ Vi sier at det har blitt mål om ballen går langt nok til høyre uten å treffe målstangen med musepekeren ser du at den står omtrent ved x: 160. Dette seende slik ut:



- ☐ På samme måte som for redning kan vi avslutte bevegelsen til figurene:



- ☐ Vi kan også la **Leo** juble litt når han scorer mål. Klikk på kattefig



- ☐ Lag et tilsvarende skript der **Leo** sier noe om at han er lei seg c



Test prosjektet

Klikk på det grønne flagget.

- ☐ Klarer du å score mål?
- ☐ Klarer keeperen å redde noen skudd?

Endre farten

Du kan gjøre spillet enklere eller vanskeligere ved å endre farten på disse er bestemt av tallet som står i **gå** **steg** -klossene til henho

Prøv forskjellige tall for både fotballen og keeperen til du finner det vanskelig, men ikke umulig.

Steg 5: Førstemann til 10!

Vi skal tilslutt se på hvordan vi kan telle hvor mange mål som scores.

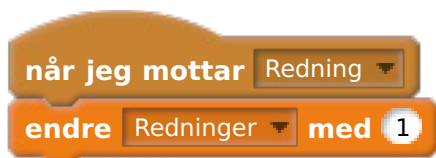
✓ Sjekkliste

For å telle hvor mange mål du har scoret, og hvor mange redninger **K**

- ☐ Klikk på scenen til venstre for figurlisten.
- ☐ Klikk på **Data**-kategorien og lag en ny variabel. Gi den nye variabelen en ny boks på scenen som er merket **Mål**, og som viser tallet **0**.
- ☐ Vi vil nå telle målene. Lag et nytt skript som endrer **Mål** hver gang på Scenen:



- ☐ Vi kan gjøre tilsvarende for å telle antall redninger. Lag en ny variabel.
- ☐ Lag deretter et nytt skript for å telle redningene:





Test prosjektet

Klikk på det grønne flagget.

- ☐ Teller spillet hver gang du scorer mål?
- ☐ Telles også hvor mange redninger keeperen gjør?



Sjekkliste

Avslutningsvis vil vi legge inn en test på om **Leo** klarer å score 10 mål omfattende.

- ☐ Først vil vi lage to nye bakgrunner, en vi kan bruke om spilleren spilleren taper (keeperen redder 10 ganger). Klikk på **Bakgrunn** bakgrunnen din midt på skjermen og velg **Lag en kopi**.
- ☐ Velg en passende farge, og klikk deretter på tekstverktøyet. Vel tekst som ligner på **Gratulerer, du vant!** litt høyt på kopien a
- ☐ Lag deretter en ny kopi av den originale bakgrunnen. Kall denne tekst.
- ☐ Lag et skript på bakgrunnen som setter variabelene dine til 0 v



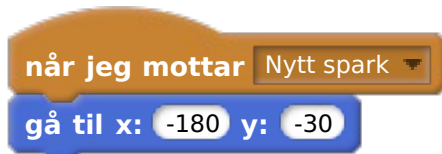
- ☐ Vi vil nå spille uten at vi må klikke på det grønne flagget for hver meldingen **Nytt spark**. Vi må nå bytte ut



med



på både **Leo**, **Ball** og **Keeper**. For eksempel, på **Leo** blir skriptet



- ☐ Til slutt legger vi på testen om vi har scoret 10 mål, eller om keeperen har scoret 10 mål. Vi vil seende slik ut:



- ☐ På samme måte endrer du **Redning**-skriptet på scenen:



Test prosjektet

Klikk på det grønne flagget.

- ☐ Får du skyte flere straffespark uten å måtte trykke på det grønne
- ☐ Bytter spillet til riktig bakgrunn om du scorer 10 mål?

- ☐ Bytter spillet til riktig bakgrunn om keeperen redder 10 ganger?
- ☐ Nullstilles målene og redningene når du klikker det grønne flagget

Lydeffekter

Du kan legge lydeffekter på spillet ditt ved å bruke klossene under. Hver kloss gir en lyd som spilles når ballen sparkes, når keeperen redder eller når det blir mål.

For å finne flere lyder å bruke i spillet ditt kan du velge **Lyder**-fanen i biblioteket eller til og med ta opp egne lyder. Eksperimenter og prøv!

Lagre spillet

Da er vi ferdige med Straffespark! Om du har noen ideer til hvordan du kan gjøre spillet ditt enda bedre, kan du gjøre det selv!

Om du klikker **Legg ut** vil prosjektet ditt bli lagt ut på Scratch-hjemmesiden.

Lisens: CC BY-SA 4.0 **Forfatter:** Erik Kalstad og Geir Arne Hjelle