



Lærerveiledning - Straffespark

Informasjon til veiledere

Straffespark er et enkelt fotballspill, hvor man skal prøve å score på s. passer bra som en introduksjon til Scratch.



Forberedelser



Antatt tidbruk: 1.5 - 2 timer for hele prosjektet.



Nødvendige forkunnskaper: Ingen.

Dersom dette prosjektet brukes som en introduksjon til Scratch anbefaler vi [Scratch](#).

Om elevene allerede er komfortable med Scratch kan du benytte anledningen til [ulik oppførsel på forskjellige figurer](#).

Typiske utfordringer

Nedenfor er en liste over utfordringer vi har opplevd at noen elever kan ha.



I Straffespark må man holde styr på tre figurer i tillegg til scenebakgrunnen. Det er viktig å være nøye med at skriptene ligger på riktig figur som beskrevet i oppgaven.



Katten skyter ballen før man klikker på den, eller den må gå flere ganger før den flytter på hvor **Leo** og **Ball** plasseres ved å endre på **gå til x** og **gå til y**. Hvor mange steg man kan gå flere steg kan man også endre på hvor langt **Leo** **går** når han skyter ballen. I koordinatsystemet er det en fin øvelse å tenke på hvilke koordinater man kan flytte på figurene ved å klikke og dra, og deretter se på resultatet.



I steg 4 jobber vi videre med skriptet som ble skrevet på **Ball** i steg 3. Om de lager to forskjellige skript vil effekten stort sett være den samme. Hvis begge skriptene flytter ballen.




I steg 4 og 5 lages det mange små skript som starter på meldingen "Hvis du klikker på" og ender på "si noe". Det vil si at **Ball** og **Keeper** har skript med **si noe** mens Scenen har skript som teller **Mål** og **Redninger**.

Dersom elevene vil at også **Ball** eller **Keeper** skal si noe kan en mulig løsning er som følger:



Det er her viktig å *ikke* bruke  siden den kl
bevege seg før etter 2 sekunder. For at snakkeboblen skal bli be

Denne kan legges øverst i  - eller 

- ☐ For enkelhets skyld settes aldri retningen på **Ball** i dette prosje
beveger seg alltid horisontalt fra venstre mot høyre - er dette sj
retning på **Ball** slik at den spretter på skrå over skjermen må r
klikke på klossen  (eller ved å legge denne klos

Variasjoner

Dette er et introduksjonsprosjekt, og elevene ledes derfor ganske det
er likevel rom for en del kreativitet. Elevene kan gjerne oppfordres til

- ☐ velge sine egne figurer og bakgrunner. **Leo** trenger absolutt ikk
med noe annet enn en fotball.
- ☐ eksperimentere med hastigheten til **Ball** og **Keeper**. Ved å end
seg saktere eller raskere. Det er nyttig læring å teste effekten a
vanskelighetsgraden i spillet forandrer seg (se også boksen **Enc**
- ☐ forandre på tekstene i snakkeboblene til **Leo**, eller tekstene sor
- ☐ legge på passende lydeffekter. Dette nevnes i oppgaven mot sl
Scratch fra tidligere kan de gjerne gjøre dette underveis i progr

Tema: Skript tilhører figure

Dersom elevene allerede er komfortable med Scratch er dette prosjekt forskjellige figurer en unik oppførsel ved å legge ulike skript på dem.

Et viktig konsept i Scratch er at man koder ved å beskrive egenskaper (skript) til figurer. På fagspråket kalles dette **objektorientert programmering**, men forskjellen er ikke relevant her). Dette virker så samtidig skaper det sjelden problemer.

Presentasjon

- ☐ Start et nytt Scratchprosjekt ved å klikke **Programmering** fra hovedmenyen.
- ☐ Legg til en ekstra figur - for eksempel **Bat1** - slik at det er to figurer i hvert sitt hjørne.
- ☐ Spør elevene hvordan de vil kode at katten beveger seg mot flua. Spør elevene om de er bevisst hvilken figur som må programmeres (*Katten*). Spør elevene om de vil skrive et skript på den andre figuren (*Nei, siden Katten beveger seg er det Katten som må programmeres*).



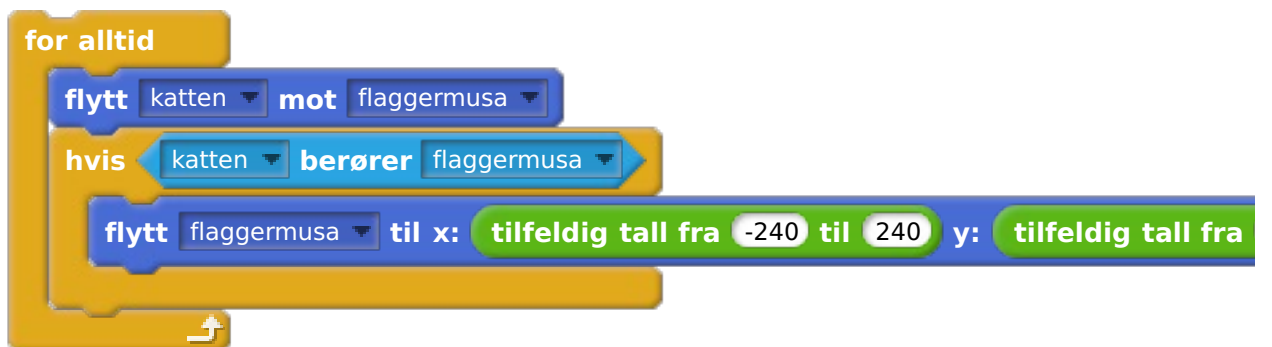
- Hvordan kan vi programmere at flaggermusa rømmer fra katter programmeres? Kanskje begge? *Vi må programmere flaggermu-*
noe nytt program for katten så lenge den ikke reagerer på at de

Det er mange måter å skrive kode for at flaggermusa rømmer. I flaggermusa):



- ☐ Spør elevene om de kan tenke seg noen annen måte (enn objektskriptene) til figurene?

Et eksempel på en annen type programmering er **imperativ** programmering, der kommandoer uten at det skilles mellom hvilken figur som komr skrives som *ett* skript omtrent som dette (ikke alle disse klosser)



Vis gjerne denne koden til elevene. I tillegg til at det bare er ett skript, *forskjellen er at man alltid må fortelle hvilken figur som skal utføre*

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