

# ◆ Snøballkrig

*Skrevet av: Geir Arne Hjellev*

*Kurs: Scratch*

*Tema: Blokkbasert, Spill, Animasjon*

*Fag: Matematikk, Programmering, Kunst og håndverk*

*Klassetrinn: 5.-7. klasse, 8.-10. klasse*

## Introduksjon

I denne oppgaven vil vi lage et spill hvor det er om å gjøre å jage bort de slemme guttene ved å kaste snøball på dem. Underveis vil vi lære hvordan vi kan gjøre et spill mer utfordrende etterhvert ved at figurene vi spiller mot blir raskere og vanskeligere å jage bort.



## Steg 1: En snøballkaster

Vi begynner med å lage oss en passende bakgrunn, samt snøballkasteren som vi skal styre.

## Sjekkliste

- ☐ Start et nytt prosjekt. Slett kattefiguren.
- ☐ La oss lage en bakgrunn. Til dette spillet trenger vi en ganske enkel bakgrunn med litt vinterfølelse. Den kan vi tegne selv:

Velg  for å tegne en ny bakgrunn. Pass på at du bruker vektorgrafikk, og tegn en stor firkant som dekker hele bakgrunnen. Velg så en mørkeblå forgrunnsfarge og en lysere bakgrunnsfarge. Klikk på malingsspannet for å fylle firkanten med farge, og velg en overgang nederst til venstre. Klikk på bakgrunnen for å fylle den med farge.

Velg nå hvit som forgrunnsfarge, og lag en litt mindre boks som dekker nedre del av bakgrunnen. Fyll også denne med passende farger.



Kall bakgrunnen for `Spill`.

- ☐ Legg følgende lille skript på scenen:



Du har kanskje sett dette trikset tidligere. Dette gjør det enklere for oss å senere komme tilbake og legge på en startmeny og lignende.

- ☐ Neste steg er å finne en passende figur som vi kan styre rundt og kaste snøballer med. Vi har brukt snømannen, Fantasi/Snowman , men du kan bruke en figur du liker. Gi figuren navnet He1ten .
- ☐ Lag en ny variabel, hastighet , som gjelder kun for denne figuren, og lag dette skriptet:



På det grønne flagget legger vi innstillinger som skal gjelde for He1ten gjennom hele spillet. Endre gjerne på størrelsen og hastigheten etterhvert som du tester spillet, slik at du finner innstillinger du liker.

- ☐ Nå kan vi lage selve hovedløkken til He1ten . Lag først en løkke som starter på meldingen start :



- ☐ Inne i løkken trenger vi to `hvis`-klosser som flytter `Helten` hastighet steg mot høyre eller venstre når piltastene høyre og venstre er trykket. Lag disse selv.
- ☐ Til slutt lager du en `hvis`-kloss hvor du sender ut meldingen `kast` når mellomrom-tasten trykkes.

---

## Test prosjektet

Klikk på det grønne flagget.

- ☐ Beveger figuren din seg rundt på scenen når du trykker høyre og venstre piltast? Det skal foreløpig ikke skje noe når du trykker mellomrom, siden ingen figurer svarer på `kast`-meldingen.
- ☐ Du må sannsynligvis endre litt på `y`-posisjonen figuren din starter på, slik at det ser naturlig ut i forhold til bakgrunnen.
- ☐ Hvis du har en figur med flere drakter kan du også bruke `neste drakt`-klosser for å animere figuren din slik at den beveger seg mer naturlig.

## Steg 2: Mange snøballer

*Det blir ingen snøballkrig uten snøballer. La oss lage kjempemange av dem! Med kram snø!*

### Sjekkliste

- ☐ Tegn en ny figur. Bruk sirkelverktøyet til å tegne en liten hvit sirkel, fargelegg den deretter slik at den blir helt hvit. Trykk på  og pass på at senterpunktet er satt midt på snøballen. Kall figuren `Snøball`.

- ☐ På samme måte som for `Helten` setter vi først opp noen standardinnstillinger for snøballen. Lag en variabel `hastighet` som kun gjelder for denne figuren:



- ☐ Vi bruker `skjul` fordi vi vil lage kloner (kopier) av denne snøballen som vi kaster av gårde. Dette er et veldig nyttig triks i Scratch. Selve koden som reagerer på `kast`-meldingen er veldig enkel.



- ☐ Selve oppførselen til hver enkelt snøball programmerer vi så på en `når jeg starter som klon`.



## Test prosjektet

Klikk på det grønne flagget.

- ☐ Hva skjer når du trykker mellomrom-tasten? Kastes det snøballer i riktig retning?
- ☐ Det ser kanskje ikke ut som om `He1ten` kaster snøballene med hendene? Du kan legge på noen `Bevegelse` -klosser rett før du `vis` er snøballen for at det skal se bedre ut.
- ☐ Som en liten utfordring kan du også prøve å legge på litt effekt av tyngdekraften på snøballen ved å endre litt på `y` mens snøballen flyr.
- ☐ Et lite problem er at `He1ten` vår er *for* flink til å kaste snøball! Hver gang vi trykker mellomrom blir det kastet mange snøballer. Dette kan vi løse ganske enkelt ved å vente til mellomrom-tasten er sluppet før vi kaster snøballen. Legg til



på `He1ten` -figuren før `kast` -meldingen sendes.

## Steg 3: Slemme gutter

*Nå skal vi se hvordan vi kan lage slemme gutter som prøver å ta oss. La oss jage dem bort med snøballer.*

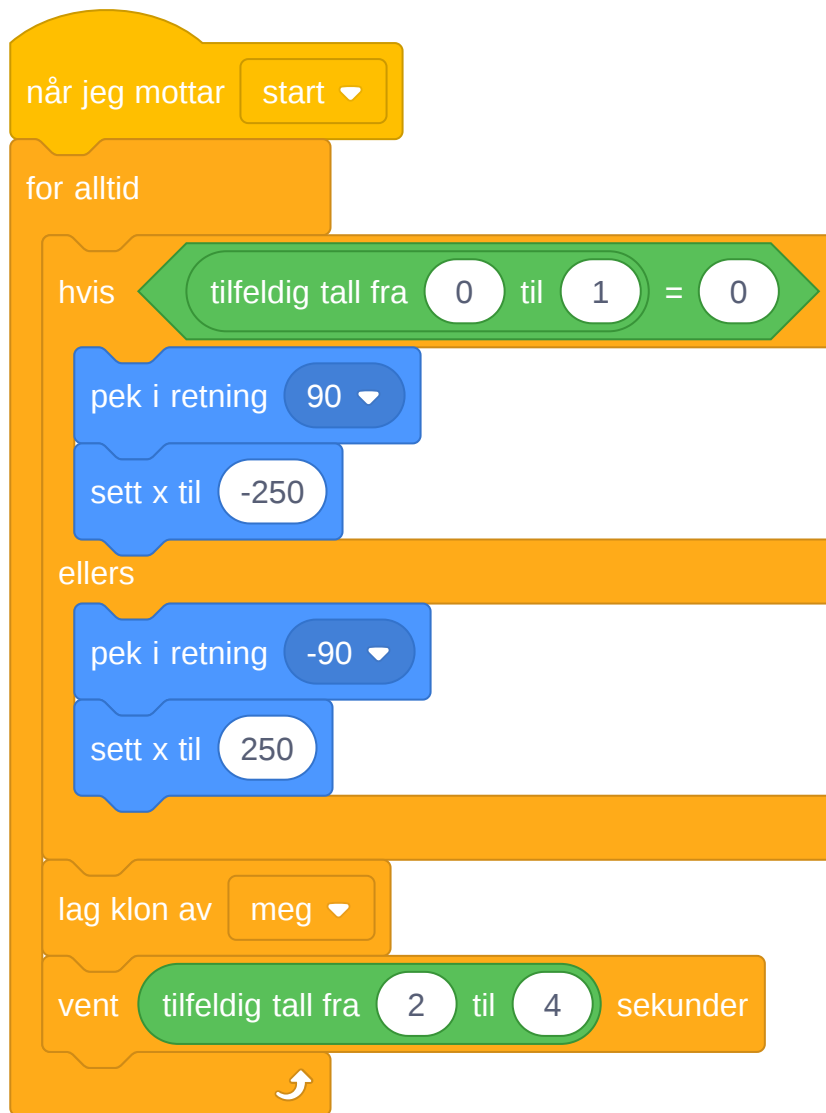
### Sjekkliste

- ☐ Lag en ny figur. Vi har brukt `Mennesker/Boy3 walking`, men du kan bruke den figuren du liker. Kanskje det er morsommere å ha snøballkrig mot et monster? Kall figuren `Skumling`.

- ☐ På samme måte som for snøballene vil vi lage kloner av Skumling . Men først bestemmer vi hvordan han skal se ut. Lag igjen en variabel hastighet som gjelder kun for denne figuren. Deretter,



- ☐ Vi vil la klonene dukke opp med litt tilfeldig mellomrom, og fra begge sider av skjermen.



- ☐ På samme måte som for snøballene må vi også bestemme oppførselen til hver enkelt Skumling .

Lag først en når jeg starter som klon -kloss hvor du først viser figuren, og deretter lager en for alltid -løkke hvor figuren beveger seg hastighet steg og deretter venter et lite øyeblikk, for eksempel 0.1 sekunder.

- ☐ Deretter lager vi en ny når jeg starter som klon -kloss hvor vi undersøker om vi treffer enten en **snøball** eller Helten .





Grunnen til at disse må ligge i et eget skript er at det første skriptet venter litt mellom hver gang figuren tar et steg. Hadde vi lagt disse **hvis**-klossene på det skriptet ville vi bare sjekket om **Skumlingen** var truffet av en snøball innimellom ventingen. Ved å lage et eget skript sjekker vi dette hele tiden.

## Test prosjektet

Klikk på det grønne flagget.

- ☐ Dukker det opp skumlinger? Kommer de fra begge sider? Går de mot midten?
- ☐ Hva skjer når en `Skumling` blir truffet av en **Snøball**? Du kan kanskje legge på en passende lydeffekt og animasjon før klonen slettes? Vi ser også at snøballen flyr videre etter at den har truffet en `Skumling`. Prøv å endre i skriptet for **Snøball** slik at snøballen også blir borte.
- ☐ Hva skjer når en `Skumling` tar `Helten`?

# Steg 4: Telle poeng og avslutte spillet

*Nå som vi nesten har et ferdig spill, vil vi også ha poeng!*

## Sjekkliste

- ☐ Å telle poeng er ganske lett. Lag først en variabel som heter `Poeng` og la den gjelde for alle figurer. La variabelen også vises på scenen, slik at vi ser hvor mange poeng vi har fått.
- ☐ Pass på at `Poeng` blir satt til 0 når meldingen `start` sendes, for eksempel med et skript på scenen.
- ☐ Endre `Poeng` med 1 når en `Skumling` blir truffet av en snøball.

Vi har tidligere laget meldingen `slutt` som sendes ut når `Helten` blir tatt av en `Skumling`. Nå vil vi bruke denne til å avslutte spillet. Men først, la oss lage en meny og en bakgrunn som forteller oss at vi tapte.

- ☐ Klikk på `Scene` til venstre for `Figurer`, og velg `Bakgrunner` -fanen. Lag to kopier av bakgrunnen din og kall dem henholdsvis `Meny` og `Slutt`.

På `Meny` -bakgrunnen kan du lage en fin tittel. Skriv også Trykk 'S' for å starte.

På `Slutt` -bakgrunnen kan du skrive en passende melding for når spillet er slutt. Skriv også Trykk 'S' for å spille igjen.

- ☐ Endre nå litt på skriptene på scenen. Først vil vi bare vise menyen når det grønne flagget klikkes:



Deretter vil vi starte spillet når `s` trykkes:



Pass også på at du bytter til bakgrunnen `spill` når meldingen `start` mottas, og til bakgrunnen `slutt` når meldingen `slutt` mottas.

- ☐ Til slutt må vi passe på at spillet faktisk avsluttes etter at `slutt` -meldingen er sendt. Legg på skript for å slette denne klonen på **Snøball** og **Skumling**, og skript for å skjule **Helten** når `slutt` mottas.

---

## Test prosjektet

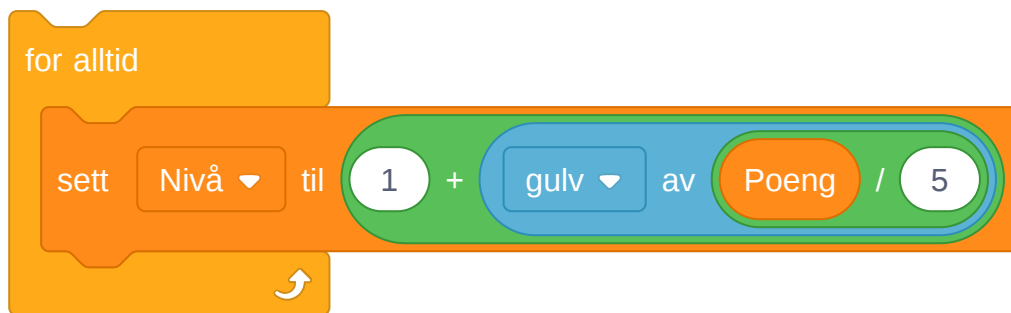
Klikk på det grønne flagget.

- ☐ Fungerer programflyten? Starter spillet når du trykker `s`? Avsluttes spillet når **Helten** blir fanget? Viser de riktige bakgrunnene når de skal?
- ☐ Legg gjerne på litt lydeffekter og enkle animasjoner der det passer.

## Steg 5: Store, slemme gutter

*Til slutt skal vi gjøre spillet vanskeligere jo lengre vi spiller*

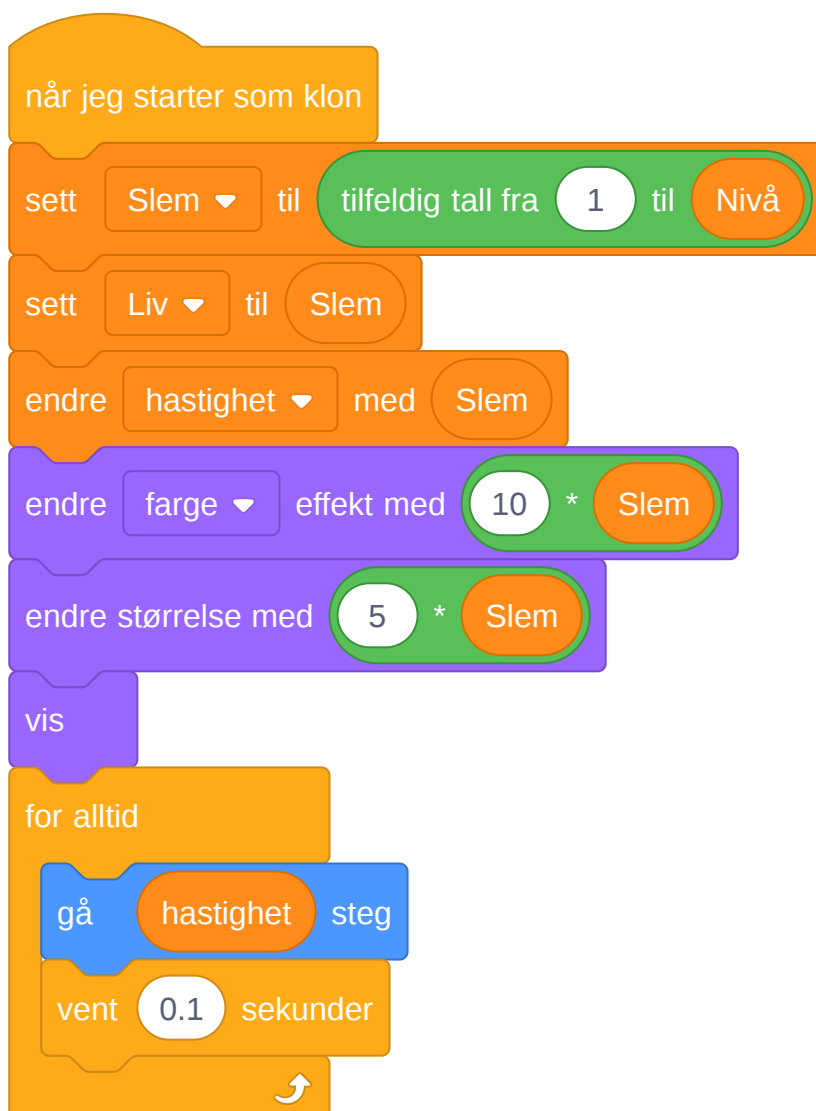
- ☐ Vi innfører først en ny variabel `Nivå` som skal gjelde for alle figurer. Denne styrer hvor vanskelig spillet skal være. La den gjerne være synlig.
- ☐ Vi regner ut `nivå` basert på `Poeng`. Utvid skriptet ditt som mottar `start` -meldingen på scenen med den følgende løkken:



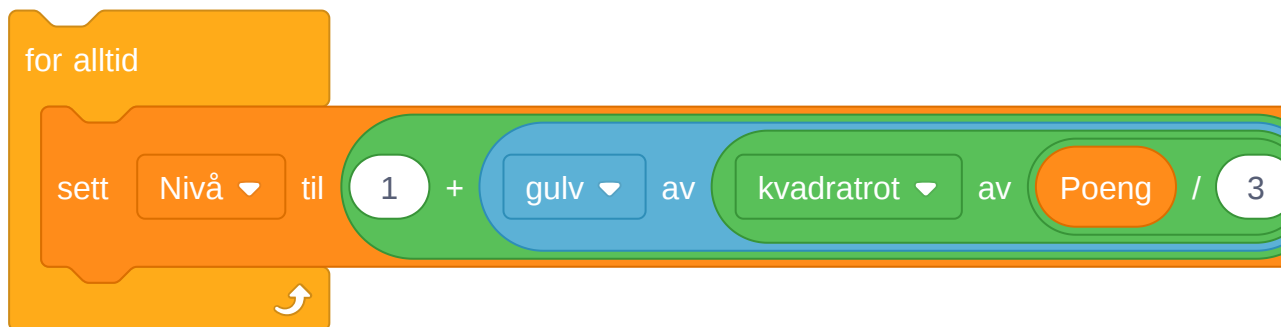
Funksjonen `gulv` runder nedover. Dermed sier vi at `Nivå` skal øke for hver femte `Skumling` vi jager vekk.

- ☐ Nå kan vi bruke `Nivå` til å lage **Skumlingene** slemmere. For eksempel kan de bli større, gå raskere og kanskje trenge flere snøballer før de blir borte.

Lag et par nye variabler som gjelder for denne figuren: `Slem` og `Liv`. `Slem` vil si noe om hvor stor og slem den enkelte **Skumlingen** er. Ved å la dette være et tall mellom 1 og `Nivå` blir spillet vanskeligere etterhvert som `Nivå` øker. For eksempel kan du legge til disse klossene på `Skumling` sitt hovedskript:



- ☐ For at en `Skumling` skal tåle flere snøballer må vi også endre litt på hva som skjer når han blir truffet. I stedet for å bare slette klonen vil vi endre `Liv` med `-1`, og deretter gi poeng og slette klonen bare om `Liv` er 0.
- ☐ Vi bør kanskje gi flere poeng for å jage bort de slemme **Skumlingene**? Dette gjør du ved å endre `Poeng` med `Slem` når en `Skumling` jages bort.
- ☐ Om vi får mer enn ett poeng for hver `Skumling` bør vi også endre på hvordan vi regner ut `Nivå`. Hvis ikke vil spillet bli veldig fort vanskeligere etterhvert. Du kan for eksempel bruke denne utregningen:



Da vil vi fortsatt øke nivået for omtrent hver femte `Skumling` som jages bort.

---

## Test prosjektet

Klikk på det grønne flagget.

- ☐ Blir spillet vanskeligere etterhvert? Prøv å endre på de forskjellige variablene og innstillingene vi har laget slik at spillet blir passe vanskelig.
- ☐ Har du flere ideer til hvordan spillet kan gjøres enda morsommere? Prøv dem ut!