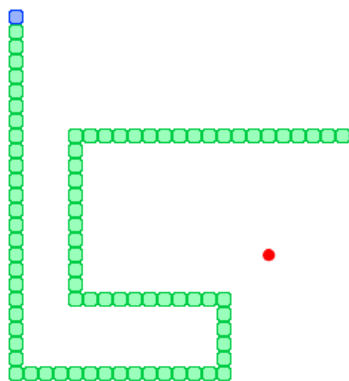


Introduksjon

En eller annen variant av Snake har eksistert på nesten alle personlige datamaskiner helt siden slutten av 1970-tallet. Ekstra populært ble spillet da det dukket opp i Nokias mobiltelefoner i 1997, og de seneste årene har spillet til og med blitt innlemmet i New Yorks Museum of Modern Arts samling.

Selve spillet går enkelt og greit ut på å styre en slange rundt på skjermen, mens slangen må unngå å krasje i kanten av skjermen eller seg selv. Slangen vokser ved å spise epler som dukker opp tilfeldige steder på skjermen. Snake kan videreutvikles på mange måter, enten ved å lage ekstra hindringer på skjermen, ved forskjellige typer bonusepler, eller for eksempel ved at to slanger konkurrerer om å spise eplene og om å stenge hverandre inne.

82



Oversikt over prosjektet

Mesteparten av kodingen av Snake skal du gjøre selv. I Snake bruker vi kloner på en litt spesiell og ganske smart måte. Vi vil derfor fokusere på kloning i begynnelsen av denne leksjonen.

✓ Plan

- ☐ Slangen flytter på seg ... eller?
- ☐ Styr slangen til den krasjer!
- ☐ Epler og annet snadder.
- ☐ Vegger, bonusepler, flere slanger og andre utfordringer.

Steg 1: Slangen flytter på seg ... eller?

Snake er i prinsippet et enkelt spill å lage. Men en utfordring er hvordan selve slangen skal flyttes rundt. Først virker det kanskje som om man trenger en eller annen liste som husker hvor hver del av slangen er slik at man kan flytte den.

I stedet for å bruke lister skal vi bruke kloning på en litt spesiell måte. Husk at når vi kloner kopierer vi både utseendet og oppførselen til en figur. Vi vil starte med en enkel boks som vil være en del av kroppen til slangen. Denne boksen vil vi flytte, kloner, flytte, kloner og så videre. Trikket for at det skal se ut som om slangen flytter på seg er at de gamle klonene sletter seg selv etter litt tid.

I figuren er den blå boksen hodet til slangen, de grønne boksene er kroppen til slangen, mens de hvite boksene indikerer hvor slangen har vært (men er egentlig slettede klonede bokser).

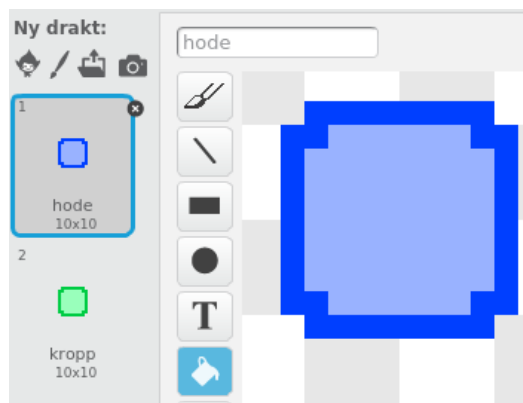


For å vite når vi skal slette kloner bruker vi tre variabler: **lengde** er lengden på slangen, **teller** er en enkel teller som passer på hvor mange steg slangen har gått siden begynnelsen av spillet. Til slutt vil **min id** være et tall som forteller hvilket nummer i rekken en gitt klon er. Over er **min id** skrevet i hver boks, **teller** er 16 siden slangen har gått 16 steg, og **lengde** er 6.

Trikset er nå ganske enkelt. Hver klon sletter seg selv hvis **min id** er mindre enn **teller** - **lengde**. La oss prøve dette ut i praksis.

✓ Sjekkliste

- ☐ Start et nytt prosjekt. Slett kattefiguren.
- ☐ Lag en boksfigur. Tegn denne selv. Du bør lage den ganske liten slik at du får plass til en lang slange på skjermen. Pass også på at boksen blir like bred som høy. Et sted mellom **10 x 10** og **20 x 20** er en fin størrelse.



- ☐ Lag så de tre variablene: **lengde** og **teller** skal gjelde for alle figurer, mens **min id** må gjelde kun for denne figuren siden den skal være forskjellig for hver klon.
- ☐ Vi skal nå lage hovedløkken i spillet. Først setter vi de nødvendige variablene, deretter bruker vi en løkke til å lage stadig nye slangebokser.



gå til x: 10 * tilfeldig tall fra -23 til 23 y: 10 * tilfeldig tall fra -16 til 16

Husk at skjermen har koordinater fra -240 til 240 i x-retning, og -180 til 180 i y-retning. Pass på at eplene dine lander godt innenfor skjermen slik at slangen kan spise dem.

- ☐ Nå trenger vi et skript som sender ut slike `Lag eple`-meldinger. Lag et skript som starter når det mottar `Nytt spill`. Dette skriptet skal `skjul` eplet og deretter sende en `Lag eple`-melding.
- ☐ Til slutt lager vi oppførselen for et slikt kloneeple. Lag et nytt skript som starter med `når jeg starter som klon`. Dette skriptet må `vise` eplet, `vente til` det `berører slangen`, `sende` en `Lag eple`-melding og til slutt `slette denne klonen`.
- ☐ Legg på noen enkle lydeffekter! For eksempel passer lyden `chomp` ganske bra når et eple blir spist. Hvilken lyd passer når slangen krasjer?

Steg 4: Videreutvikling av spillet

Du står helt fritt i hvordan du vil jobbe videre med spillet ditt, men her er noen ideer som kan gjøre spillet enda morsommere å spille:

✓ Ideer til videreutvikling

- ☐ Legg til en poeng-teller. Det enkleste er bare å bruke `lengde` som poeng. Vis denne variabelen på skjermen. Høyreklikk på den og velg `stor`.
- ☐ La hastigheten øke etterhvert i spillet. Vanligvis gjør vi dette ved å forandre hvor mange steg en figur går. Det kan vi ikke gjøre her siden hver boks i slangekroppen må henge sammen. I stedet kan du forandre på hvor lenge det ventes mellom hver klon som lages.
- ☐ Kanskje du kan videreutvikle hele konseptet, slik at det er mulig å plukke opp forskjellige bonusepler underveis. For eksempel kan du ha epler som øker lengden på slangen med mer enn 1, epler som lager flere epler, ekstra store epler, eller noe helt annet.
- ☐ Det trenger jo ikke bare være ett eple om gangen. Om du for eksempel lager tre epler i starten av spillet vil det være litt mindre leting etter eplene og spillet kan være litt morsommere. Du kan gjøre dette ved hjelp av kloning, bare pass på at ikke klonene lager nye kloner igjen!
- ☐ La eplene flytte seg om det går en viss tid uten at de blir spist. For å holde styr på tiden kan du bruke `tid`-klossen i `Sansning`-kategorien.
- ☐ I stedet for at slangen bare kan krasje i seg selv eller i kanten, kan du også lage hindringer på selve brettet. Disse kan du for eksempel tegne på bakgrunnen i en spesiell farge og deretter undersøke om slangen `berører fargen`. Du kan til og med ha flere brett med dører mellom.
- ☐ Hva med å lage en to-spiller versjon? Spillerene styrer hver sin slange, og samtidig som de konkurrerer om å spise eplene prøver de å sperre hverandre inne.
- ☐ Spillet ditt fortjener også en forside og en meny som kan starte spillet. Her kan du også la spillerene velge vanskelighetsgrad ved å endre på ting som lengde, hastighet, hinder i banen og så videre.