

# ● Lærarrettleiing - Labyrint

*Skrevet av: Stein Olav Romslo*

*Kurs: Scratch*

*Tema: Blokkbasert, Spill*

*Fag: Programmering*

*Klassetrinn: 1.-4. klasse, 5.-7. klasse, 8.-10. klasse*

## Om oppgåva

I denne oppgåva skal elevane lage eit enkelt spel der ein kontrollerer ein liten utforskar som leitar etter skatten gøymt inne i ein labyrint.



## ✓ Oppgåva passar til:

**Fag:** Programmering.

**Anbefalte trinn:** 3.-10. trinn.

**Tema:** Løkker, testar, variablar, tilfeldighet.

**Tidsbruk:** Dobbeltime eller meir.

## Kompetansemål

- ☐ **Programmering, 10. trinn:** bruke grunnleggjande prinsipp i programmering, slik som løkker, testar, variablar, funksjonar og enkel brukarinteraksjon

## Forslag til læringsmål

- ☐ Elevane kan forklare korleis løkker, testar og variablar fungerer, og kvifor det er verdifullt å bruke desse i denne oppgåva.
- ☐ Eleven kan forklare korleis brukaren kan styre figuren med piltastane, og korleis figuren interagerer med labyrinten.
- ☐ Eleven kan forklare korleis tilfeldigheitsgeneratoren fungerer.

## Forslag til vurderingskriterium

- ☐ Eleven syner middels måloppnåing ved å fullføre oppgåva slik det er beskrive.
- ☐ Eleven syner høg måloppnåing ved å leggje til fleire moment, og utvikle ein meir utfordrande labyrint.
- ☐ Dette er ei oppgåve der elevane fint kan prøve kvarandre sine labyrintar og vurdere kvarandre.

## Føresetnader og utstyr

- ☐ **Føresetnader:** Ingen, fin introduksjon til Scratch.
- ☐ **Utstyr:** Datamaskiner med Scratch installert. Eventuelt kan elevane bruke Scratch i nettlesaren viss dei har ein brukar (eller registrerer seg) på [scratch.mit.edu/](https://scratch.mit.edu/) (<https://scratch.mit.edu/>). Elevane kan gjerne jobbe to og to saman.

## Framgangsmåte

Her finn du tips, erfaringar og utfordringar til dei ulike stega i oppgåva. Klikk her for å sjå oppgåveteksten. ([../labyrint/labyrint\\_nn.html](#))

### Steg 3: Utforskaren kan ikkje gå gjennom veggen

- ☐ Eit vanleg problem her er at **utforskaren kan gå rett gjennom veggen**. Typisk vil det vere fordi ein ikkje har rett farge i XXX-klossen. Eventuelt at ein har brukt fleire fargar på veggane i labyrinten. Det er viktig at alle veggane er teikna med same farge.

Ein kan òg oppleve at utforskaren går rett gjennom veggen viss hastigheit er for høg. Det er fordi utforskaren *hoppar* hastigheit steg kvar gong ein trykkar ein piltast, og då kan den hoppe over ein vegg.


- ☐ Eit anna problem er at **utforskaren hoppar gjennom vegg når den snur**. Alle figurane har eit definert senterpunkt som dei roterer rundt (sjå steg 2 i prosjektet Soloball ([../soloball/soloball\\_nn.html](#)) for eit godt døme på korleis dette virkar). Viss dette senterpunktet ikkje er midt på *Utforskar*-figuren vil det sjå ut som den hoppar rundt når den snur. For å setje senterpunktet riktig kan de trykkje på Drakter-fana og så på . Korset viser kor senterpunktet er, og det kan bli dratt slik at det er midt på figuren.

- ☐ Det kan skje at **det er vanskeleg å bevege seg i labyrinten**. Viss gangane er for smale eller veggane for skrå blir det vanskeleg for utforskaren og froskekongen å bevege seg.

# Steg 5: Froskekongen voktar i gangane

- ☐ Eit vanleg problem her kan vere at **utforskaren eller froskekongen set seg fast i veggen**. I denne oppgåva har me prøvd å halde koden så enkel som mogleg. Spesielt er koden som passar på at figurane ikkje går gjennom veggane litt *for enkel*. Nokre enkle tips for å motverke problemet er presentert i tipsboksen nedst i steg 5 i oppgåva.

## Stopp alle

Me brukar klossen  for å stoppe alle skripta i programmet når froskekongen tek utforskaren. Det stoppar alle skripta som starta då me trykka på det grønne flagget, men det hindrar ikkje nye skript å starte. Difor kan framleis utforskaren bli flytta rundt med piltastane etterpå.

Sidan oppgåva er eit introduksjonsprosjekt er det ikkje gjort noko med dette. For elevar som har lyst å prøve seg på ei løysing kan du foreslå det følgjande:

- ☐ Lag ein `game_over` -variabel som er sett til `false` eller `0` når spelet pågår, som blir sjekka kvar gong spelaren trykkjer ein piltast, før utforskaren flyttar seg. Set variabelen til `true` eller `1` når spelet er over.
- ☐ Ei meir vanleg (og betre) løysing er å bruke ei `for alltid`-løkke med `viss ... er trykt?` -klossar. Desse blir stoppa av `stopp alle`-klossen.

## Variasjonar

- ☐ Dette er eit introduksjonsprosjekt, så elevane blir ført ganske detaljert gjennom korleis spelet skal programmerast. Det er framleis rom for ein del kreativitet. Elevane kan gjerne bli oppfordra til å
  - ☐ **velje sine egne figurar**. Dei kan fritt velje figurane som blir brukt for `Utforskar`, `Skatt` og `Froskekonge` utan at det har nokon effekt på programmeringa.

- ☐ **teikne sin heilt egne labyrinth.** I oppgåva finn de eit dåme (eller to om ein ser på teikninga fyrst i oppgåva) på ein labyrinth, men elevane kan gjerne teikne ein annan. Pass på at du ber elevane tenke på at det skal vere enkelt for utforskaren og froskekongen å bevege seg rundt, så labyrinthen bør ha rette veggar og breie nok gonger.
- ☐ **eksperimentere med hastigheit.** I steg 1 i oppgåva blir det vist korleis ein kan endre kor raskt ein figur flyttar seg ved å bruke ein hastigheit -variabel. La elevane eksperimentere med denne for utforskaren og froskekongen, og spør dei korleis det forandrar vanskegraden i spelet.
- ☐ Viss elevane allereie er komfortable med Scratch kan du nytte anledninga til å prate om korleis teikneverktøyet i Scratch fungerer og gi dei nokre tips til korleis dei kan bruke det effektivt.
- ☐ For dei meir avanserte elevane kan du vise fram kode som gjer ein betre sjekk av kollisjon med vegg.

## Eksterne ressursar

- ☐ Førebels ingen eksterne ressursar...