



Frantic Felix

Introduksjon:

I dette prosjektet skal du lage et komplett spill. Det er et plattformspill slemminger og samler nøkler for å slippe ut av hulen. Når han kommer

I del 1 av dette prosjektet kommer du til å lære hvordan vi får Felix til skal du designe egne brett og nivåer. Til slutt skal du sette brettene så

Del 1: Lag byggeklossene

Plattformspill, som for eksempel **Manic Miner** og **Mario Bros**, handler om å hoppe inn i ting. Noen ting, som vegger, stopper deg. Andre ting, som gjenstander, kan du samle på. Andre ting er bakgrunner og påvirker ikke spillet i det hele tatt.

Det betyr at det å vite når du har kommet bort i noe er viktig. Scratch har to blokker: **berører sprite**, **berører farge?**-kloss, **farge berører?**-kloss. Du kan vite mer enn at to figurer har kommet borti hverandre; du trenger å vite om de har kommet borti noe til venstre, kan du ikke gå lengre til venstre. Men du kan fortsatt gå til gulv å stå på.

Hvis du berører en slemming, kan dette skade deg; men hvis du berører en gjenstand, kan du samle på den. Av de innebygde Scratchklossene kan si noe om **retningen av berøring** berøringssensorer.

Steg 1: Berøringssensor

Vi skal ha fire figurer som følger etter Frantic Felix. **Hver figur oppfører seg annerledes hvis det oppstår en berøring.** Hovedfiguren Felix bruker disse variablene: **retningen av berøring**. Hver berøringssensor har en farget linje og bruker **farge berører?**-kloss.

grønn for hinder. Vi bruker **rød** for berøringssensorene.

✓ Sjekkliste

- ☐ **Åpne et nytt prosjekt.** Legg til **frantic-felix/testlevel** bakgrunn.
- ☐ **Endre navn på figur Sprite1 til Felix.** Kontroller at drakten blir riktig.
- ☐ Lag fire nye figurer fra bildene (be om hjelp hvis du ikke har disse bildene): **frantic-felix/left**, og **frantic-felix/right**. Disse figurene skal bli berøringssensorer **venstre**, og **høyre**. Kontroller at draktene deres ikke kan flippe.
- ☐ Lag fire variabler (for hver av figurene): **blokkert topp**, **blokkert bunn**, **blokkert venstre**, og **blokkert høyre**.
- ☐ Lag dette skriptet for hver av sensorene:



- ☐ Endre variablene for hver av sensorene. Den nederste sensoren **bunn** hvis den berører grønn eller svart.

Tips: Det er enklere å velge fargene i én figur, for så å kopiere skriptet som blir satt. Da slipper du å finne de riktige fargene fire ganger. Inntil

musepekeren evig.



Test prosjektet

Trykk på det grønne flagget.

Du bør se at Felix følger etter musepekeren, omringet av et rødt rektangel. Når Felix følger med på variablene, ser du at de endrer seg når du drar Felix rundt. I øyeblikket beveger Felix seg gjennom plattformene og grønne hindre.

Steg 2: Bevegelse, gulv og



Sjekkliste



Neste steg er å få Felix til å gå. Vi bruker **venstre** og **høre piltast**. Hvis Felix står på et gulv, faller han ned (vi fikser hopping etterpå). Vi vil at Felix skal gå til venstre og flytter seg litt. Men vi vil ikke at han skal flytte seg



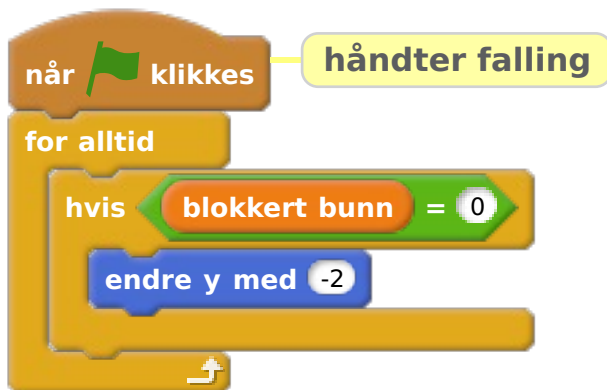
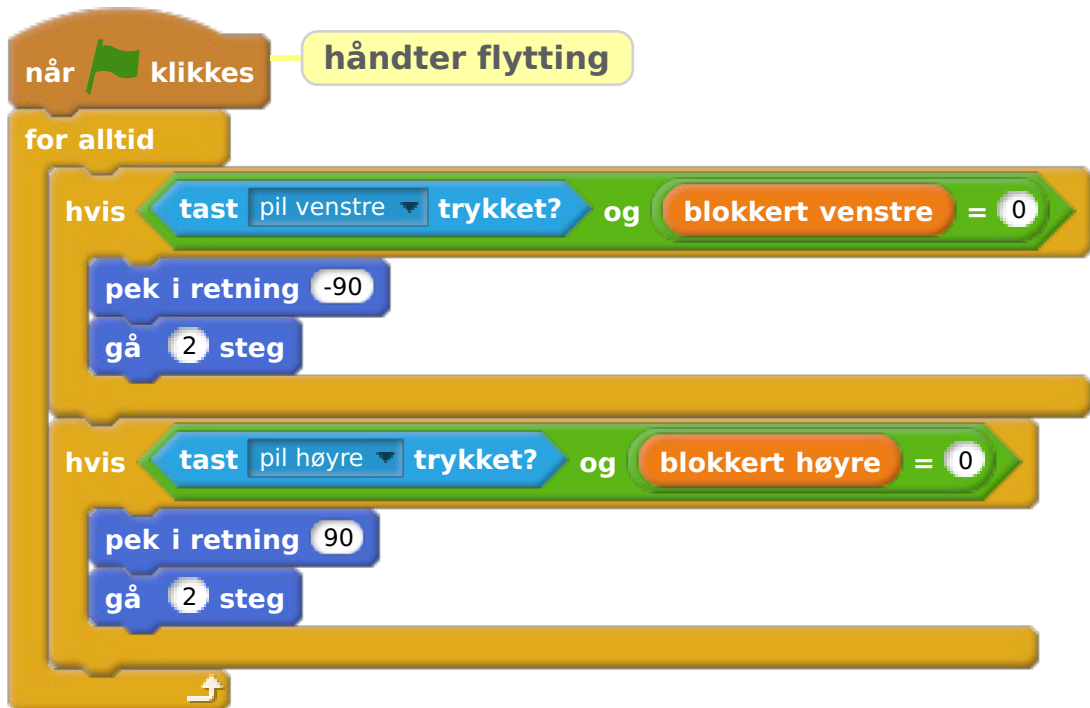
Vi kunne brukt **når tast trykkes** for å flytte Felix, men det gir problemer hvis du bruker **hvis tast pil venstre trykkes?**-klosser i en **for**-løkke. Vi kan teste **blokkert venstre** variabelen i **hvis**, og bruke en **og kloss** for å sikre at når du trykker på en tast og **blokkert venstre** er 0. Vi må gjøre tilsvarende



Vi kunne animere Felix sine bein i samme kloss, men da beveger Felix seg **alltid**-løkke under en annen **når grønt flagg klikkes**-høyt.



Det siste vi må gjøre er å ordne **falling**. Hvis Felix ikke står på et gulv, faller han ned. Vi kan gjøre dette med en **for alltid**-løkke under en **når grønt flagg klikkes**-høyt.



Test prosjektet ditt

Bruk musen for å dra Felix et sted på scenen og trykk så på den. Så må du fjerne det skriptet! Du bør kunne bruke venstre og høyre knapp

står på gulvet, bør han falle sakte nedover.

- ☐ Vi vil også at berøringssensorene ikke skal vises. Det er ikke mulig å bli oppdaget. Istedet, legg til en **sett gjennomsliktig effekt til kollisjonssensor**. Dette gjør figuren gjennomsliktig, uten å skjule



Test prosjektet ditt

Nå du klikker på det grønne flagget, skal kollisjonssensorene forsvinne fra stoppskiltet.

Steg 3: Hopping

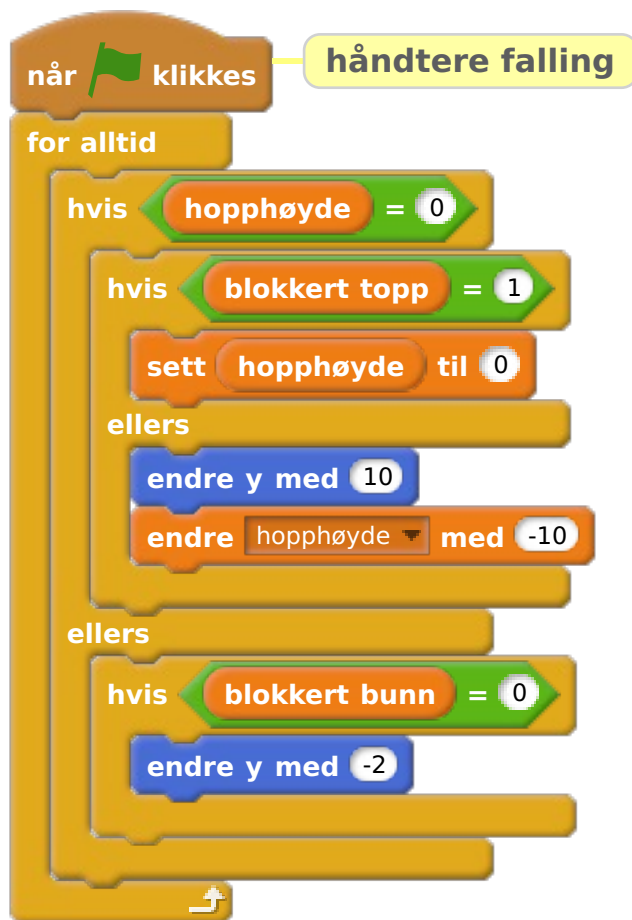
Det er et par utfordringer med hopping.

- ☐ Vi vil ikke at Felix **faller** hvis han er **på vei opp**.
- ☐ Vi vil ikke at Felix skal hindres av gulv når han er på vei opp, men **er på vei ned**.
- ☐ Vi vil ikke at Felix skal *_hoppe opp* i undersiden av et **grønt hind**.
- ☐ Vi vil at hoppet skal være en myk bevegelse, så vi vil ikke at Felix skal **slås** mot gulvet.
- ☐ Felix skal bare kunne hoppe hvis han **står på gulvet**. Spillet blir



Sjekkliste

- ☐ Det som gjenstår nå, er å få Felix til å hoppe. La oss bruke **mell**
- ☐ For å kontrollere hoppet, bruker vi en ny variabel, **hopp høyde**. Hvis den er 0, faller han (eller har falt), som beskrevet over.
- ☐ Vi vil at Felix skal hoppe opp til **100 pixler**. Legg til en ny **hvis** tastetrykk. Hvis vi trykker **mellomromstasten** og Felix står på, setter vi **hopp høyde** til 100.
- ☐ Vi må endre falle-skriptet. I **for alltid**, trenger vi en **hvis-ellers** klossen. Kravet for denne **hvis-ellers**-klossen er at hvis **hopp høyde** klossen går inn i ellers-delen av den nye **hvis-ellers**-klossen.
- ☐ Når vi vet at Felix hopper opp, må vi sjekke om hodet hans treffer (Dette gjør at Felix ikke kan hoppe inn i hindringer). **Ellers, flyt**
- ☐ Du ender opp med noe tilsvarende dette:



Test prosjektet ditt

Trykk på det grønne flagget. Kan Felix hoppe? Hopper han fra en kant? Hva om han hopper oppover over kanten av en plattform? Hvor høyre? Hva skjer når du trykker på mellomromstasten mens Felix faller?

Steg 4: Nøkler og mål

Vi har klart å få Felix til å bevege seg rundt i verdenen. Nå må vi plassere tre nøkler rundt i hulen. Felix kan plukke dem opp ved å gå til en redningskapsel og redde seg selv ut av hulen.

Vi plasserer tre nøkler rundt i hulen. Felix kan plukke dem opp ved å gå til en redningskapsel og redde seg selv ut av hulen.

- ☐ Vi lager en ny variabel, **nøkler igjen**, som holder orden på hv **scenen** skal sette denne til **3** når det **grønne flagget** klikkes.
- ☐ Både nøklene og redningskapselen må være figurer. (Bruk **frant pod** for kapselen.)
- ☐ Hver nøkkel trenger **to skript**: det første plasserer nøklen på rille for å endre fargen (som gjør det enklere å se den på skjerm) venter til Felix kommer bort til nøkkelen. Når dette skjer, skjuler
- ☐ Redningskapselen er et litt vanskeligere skript. Den bruker en **f** **0**. Med en gang dette skjer, begynner kapselen å blinke (for å v en annen **hvis** for å oppdage når Felix berører den blinkende r kapselen en **seier**-melding og sier "Du vant!". Felix svarer på r



Test prosjektet ditt

Trykk på det grønne flagget.

Steg 5: Slemminger og dø

Nå er det på tide med slemminger!

Det skal være to typer farlige ting. En type vil være slemminger som (Den andre typen er farlige ting i bakgrunnen.

La oss først lage en slemming. Den vil bare bevege seg langs

Sjekkliste

- ☐ **Lag en ny figur**, bruk hvilken som helst drakt. Den bør være så **vase** drakt, og gjorde den fire steg mindre). Slemmingen trenger om den berører Felix.
- ☐ Lag tre **hvis**-klosser inni en **for alltid**-løkke. Den første sjekl sender den en **tap**-melding. De to andre **hvis**-klossene sjekke har det, snur slemmingen. Tilslutt, tar slemmingen to steg. (Ved kontrollere hvor raskt slemmingen går.)

Vi trenger ikke å bruke kollisjonssensor her, da vi ikke br slemmingen.



- ☐ Legg skript til både Felix og redningskapselen slik at den svarer Kapselen skal si "Du tapte!".



Test prosjektet ditt

Trykk på det grønne flagget. Beveger vassen seg? Stopper den og skjer hvis Felix hopper på den, fra oversiden eller undersiden? Forsvin forstatt vinne spillet?

- ☐ **Neste, farlige ting!** La oss si at alt som er lyseblå er dødlig for har en blå rose på det øverste nivået. Legg til enda et skript på



Test prosjektet ditt

Trykk på det grønne flagget. Dør Felix hvis han kommer borti den

Oppsummering

Det du har laget nå er et veldig enkelt plattformspill. Akkurat ikke poenget. Dette spillet er en verktøykasse som du kan bruke til å lage egne brett.

Del 2: Brettdesign

Forrige gang bygde du alle delene et plattformspill består av. brett..

Kort oppsummert gjorde du:

- Felix kan gå til venstre og høyre og han kan hoppe.
- Felix faller ned hvis han ikke står på gulvet.
- Felix kan ikke gå gjennom grønne hindere.
- Blå ting i bakgrunnen og slemminger tar livet av Felix.
- Slemminger beveger seg i faste ruter.

- Slemminger kan ta livet av Felix hvis han kommer borti dem.
- Felix kan samle nøkler ved å gå bort til dem.
- Når Felix har samlet alle nøklene, kommer en redningskapsel til (på et vis nivå).

Dette kan være byggeklossene dine. **Bruk dem til å lage egne brett.**

Du kan lage en hel rekke med brett som Felix må klare. Neste gang skal du lage brett.

Brett kan ha store eller små plattformer, mange eller få plattformer. De kan ha mange hindringer eller dødelige ting i bakgrunnen. Prøv å lage flere brett. Kanskje er enklere enn et annet. Tenk over hvor vanskelig eller enkelt det er.

Du kan endre de spesielle fargene (svart, grønn og blå), men da må du endre skriptene. Det må være samme farge i alle brett. (Du kunne hatt forskjellige farger, men måtte legge inn mange **eller**-klosser rundt fargeberøringsblokkene.)

Test brettene dine. Hvis du har tid, opprett brett i Scratch og spill dem. **enkle.** Hvis du programmerer brettene, må du **lagre bakgrunnen** din for å teste eventuelle slemminger. I tillegg må du huske retningen slemmingene går.

Hvis du har laget et par brett og lagt dem inn i Scratch, prøv disse eksemplene.

Utvidelse: Tramping

Hva om slemmingene døde hvis de ble trampet på? Kanskje kan du gjøre noe hvis slemmingen berører den nederste kollisjonssensoren.

Utvidelse: Kraftpiller

Du kan lage en “kraftpille” som gjør at Felix kan ødelegge slemmingene. Felix slemmingene han berører. Effekten går over etterhvert.

Det er opp til deg hvordan du får dette til å fungere. Kanskje vil du gjøre det på en annen måte.



Test prosjektet ditt

Trykk på det grønne flagget.

Del 3: Sett sammen spillet

Nå har du noen brett. Du har nok verktøy til å få dem til å fun sammen til et ferdig spill! Hvis du klarer å sette sammen brettene begynner å spille spillet ditt.

Steg 1: Vise et nytt brett

Når Felix klarer ett brett, må vi flytte til neste. Det betyr at vi må sende en melding (istedet for `seier`-meldingen) når Felix har funnet alle nøkler, og en melding for å klargjøre neste brett. Vi trenger også en variabel, `nåværende`, som sender `start brett`-melding.

Alt må stort sett svare på `start brett`-meldinger for å klargjøre neste brett.

Scenen må vise riktig bakgrunn. Hver av slemmingene, alle nøkler, og **posisjon**. Slemmingenes rute trenger **oppdatering**. Felix må flyttes til start.

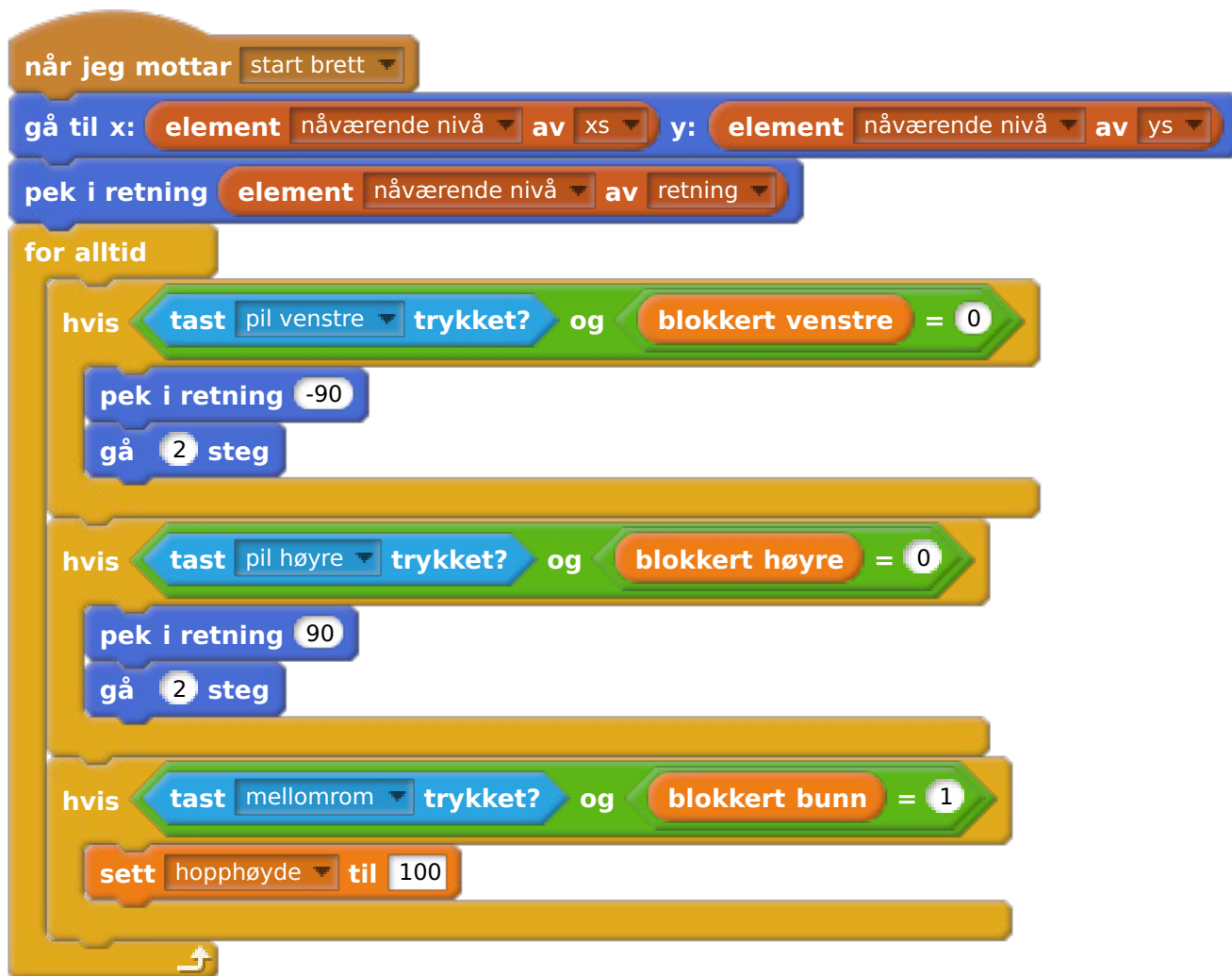
De fleste figurene må svare på `start brett`-meldingen istedet for de **klossene** i de fleste skript.



Test prosjektet ditt

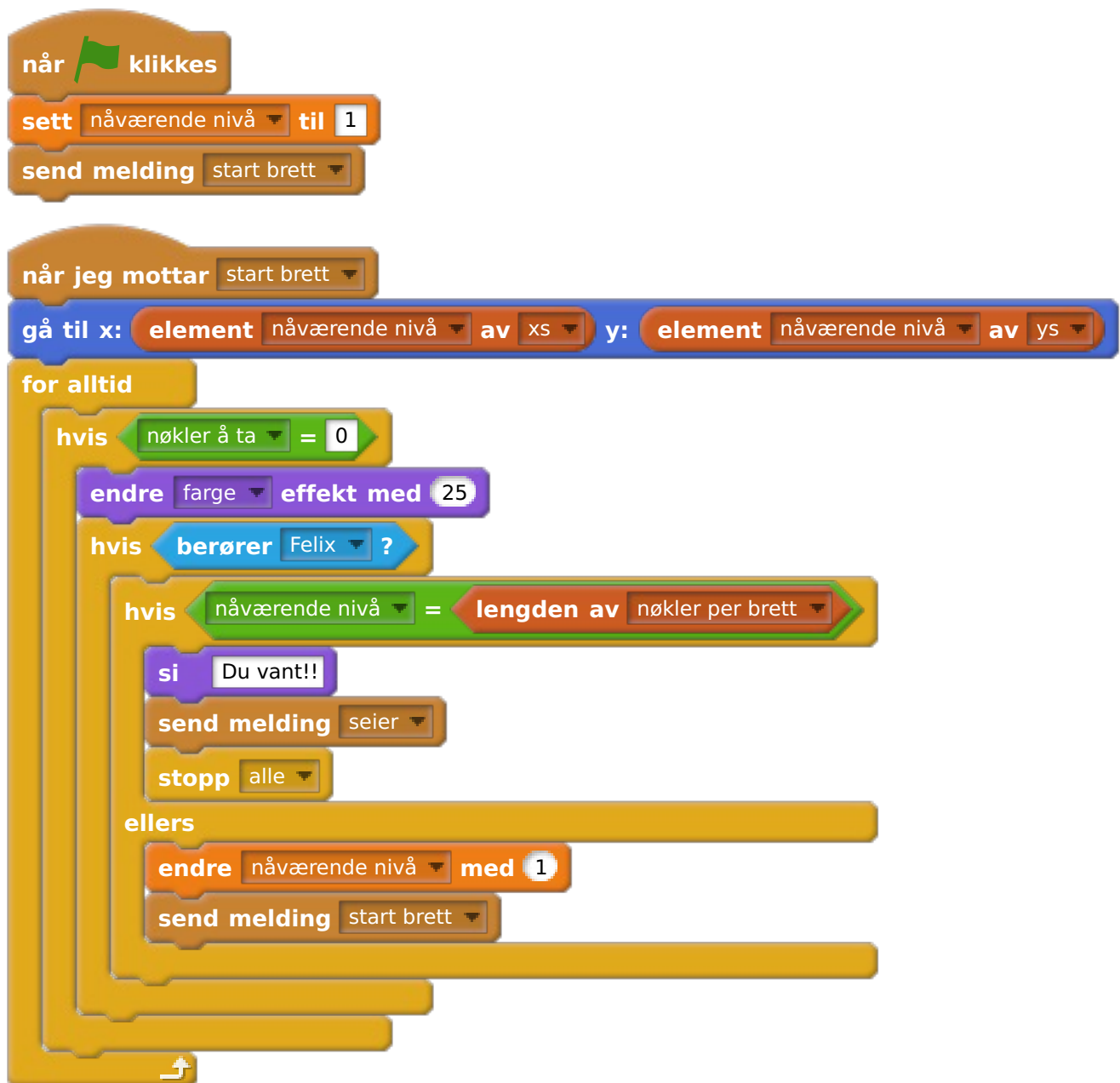
Fortsett å teste spillet etter hver endring. Husk å teste de delene som fortsatt virker som planlagt.

Vi kommer ikke beskrive alle endringene, men vi skal vise hvordan Felix



Du legger kanskje merke til at start verdiene **x**, **y**, og **retning** for Felix (hver liste privat til den figuren) for å lagre verdiene vi trenger for der trenger ikke bruke lister, du kan bruke **hvis**-klosser som sjekker hvilke

Og her er redningskapselen, som håndterer all nivå-endringen:



Steg 2: Spill!

Du har laget spill. De andre i kodeklubben har laget spill. Spill de for de andre runde spillet ditt?

Utvidelse 1: Flere liv

Felix trenger kanskje et par sjanser, eller et par liv for å klare seg i

drakter, som viser **ett**, **to**, og **tre hjerter**.

Bruk frantic-felix/1-heart, frantic-felix/2-heart, and frantic-felix/3-heart. Når det **grønne flagget** blir trykket, skal det vises tre hjerter. Hvis hjerte mindre. Isteden for å vise ingen hjerter, skal figuren skjules.

spill slutt - og **seier** -meldinger bør håndteres av en **ny figur**, viser riktig plakat når **spill slutt** - og **seier** -meldinger sendes. I viser seg selv.

Du må endre hvordan redningskapselen håndterer vinning og tap, Kanskje er slemmingen litt for kjappe på foten, de tar Felix før brett når han kommer borti en slemming, **skjul** slemmingen så snart du (inkludert Felix) tid å nullstille før slemmingene oppdager en ny ko

Utvidelse 2: Tidsbegrensninger

Ta tiden på Felix! Bruk et **Timer Scratchkort** for å legge til en **spill over** -melding. Husk å **nullstille timeren** når et nytt brett :

Lisens: Code Club World Limited Terms of Service **Forfatter:** O
Översetter: Lars-Erik Wollan