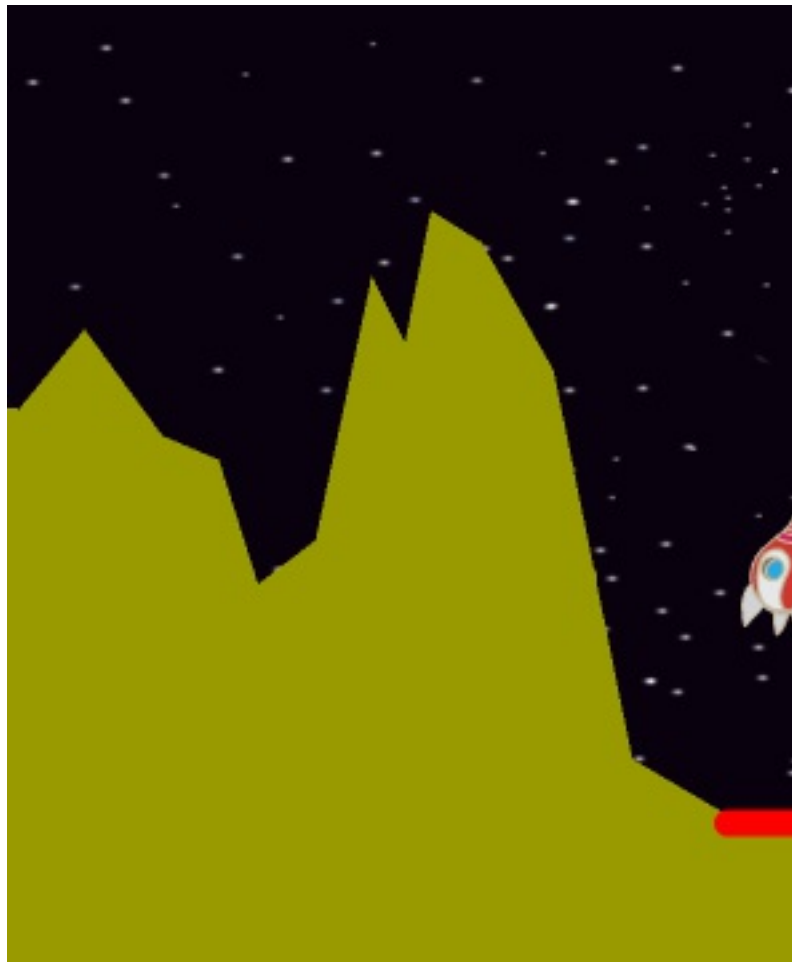




Lunar Lander

Introduksjon

Lunar Lander ble opprinnelig utviklet på slutten av 1970-tallet. Målet i kontrollen av romskipet ble videreført i spillet Asteroids som ble utgitt



Oversikt over prosjektet

Mesteparten av kodingen av Lunar Lander skal du gjøre selv. I Lunar L

modelleringen av romskipet ganske realistisk.

Plan

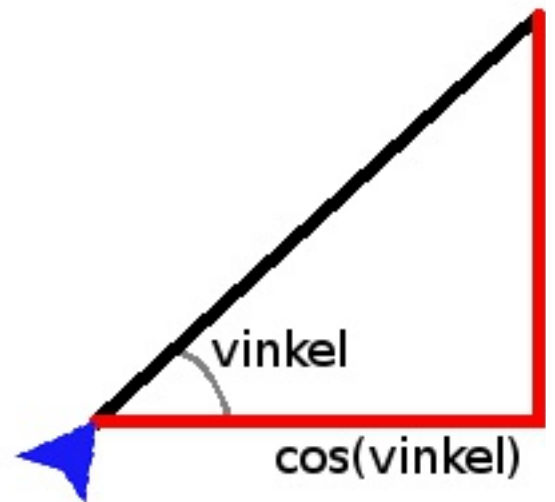
- ☐ Et flyvende romskip
- ☐ Vi ankommer månen
- ☐ Klar for landing
- ☐ Begrenset drivstoff, flere landingsplasser og andre utfordringer

Steg 1: Et flyvende romskip

Vi skal nå programmere et romskip som flyr rundt på skjermen. For å `fartX` og `fartY`, som beskriver hvor raskt romskipet beveger seg her og der. Disse samtidig vil romskipet kunne bevege seg i hvilken som helst retning.

Utfordringen i programmeringen av spillet er at romskipet kan rotere. Når spilleren trykker pil-opp for å bruke motoren vil vi gi romskipet ekstra fart i den retningen. Hvor mye sidelengs fart det får, og i hvilken retning, vil vi se på senere.

Dette virker umiddelbart kanskje litt vanskelig, men det er akkurat det vi skal gjøre. I funksjonen nedenfor forteller for eksempel `cosinus(vinkel)` om hvor mye av farten som går i den retningen.



✓ Sjekkliste

- ☐ Start et nytt prosjekt. Slett kattefiguren.
- ☐ Legg til en romskip-figur. Du kan gjerne bruke `Romfart/Spacesh`
- ☐ Lag romskipet ganske lite. Du kan for eksempel bruke `sett størrelse til 20 %` rett etter at det grønne flagget er klikket.
- ☐ Lag to nye variable, `fartX` og `fartY` som gjelder kun for denne
- ☐ Vi lager først hovedløkken som lar romskipet falle, kun påvirket



Om du prøver spillet ditt nå skal romskipet falle stadig raskere r

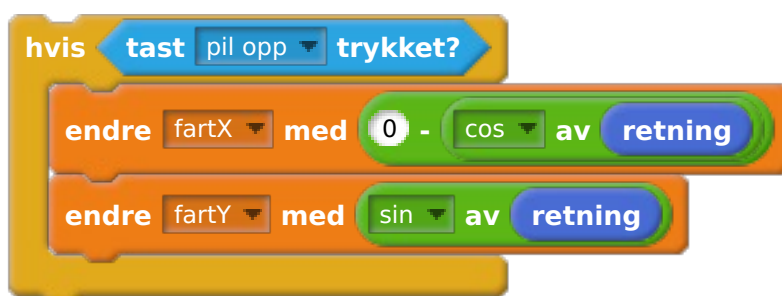


Det er denne som modellerer tyngdekraften. Om du forandrer v eller det kan til og med falle oppover om du lar tyngdekraften v

- ☐ Vi vil nå programmere kontrollen av romskipet. Først og fremst piltastene mot høyre og venstre.

Legg til to **hvis**-blokker inne i **for alltid**-løkken hvor du **venstre** avhengig av hvilken piltast som trykkes.

- ☐ Når pil opp-tasten trykkes vil vi at romskipet skal få litt ekstra fart tidligere kan vi bruke de matematiske funksjonene sinus og cos i **for alltid**-løkken din.



Du finner både sinus og cosinus-funksjonene som valg på klosse

kvadratroten av 9

Vi må bruke $0 - \cos \text{ av retning}$ i stedet for $\cos \text{ av retning}$ hvordan man måler vinkler i matematikk.

- ☐ Prøv spillet ditt. Kan du styre romskipet rundt på skjermen? Der du justere **endre fart**-klossene, for eksempel slik,



Pass på at du justerer begge klossene med det samme tallet.

Steg 2: Vi ankommer månen

Nå skal vi legge til et månelandskap hvor spilleren skal prøve å lande.

✓ Sjekkliste

- ☐ Lag en ny bakgrunn, hvor du tegner et passende månelandskap også inn en eller flere landingsplasser hvor romskipet skal lande landingen senere, så det enkleste er å bruke en farge for landskapet
- ☐ For at romskipet skal slutte å fly når det treffer bakken kan du bruke **berører fargen** du har brukt på bakgrunnen
- ☐ Legg også til en **send melding** **Sjekk landing** rett etter **gjenta til**
- ☐ Prøv spillet ditt igjen. Du skal nå kunne fly rundt helt til romskipet

Steg 3: Klar for landing

Vi vil til slutt sjekke hvor og hvordan romskipet landet.

✓ Sjekkliste

- ☐ Når vi sjekker landingen til romskipet vil vi skille mellom tre for

1: Romskipet landet utenfor landingsplassen (kræsjet).

2: Romskipet landet på landingsplassen, men landet enten på s

3: Romskipet landet perfekt på landingsplassen.

Lag et nytt skript på romskipet som starter når det mottar meld
bruke **hvis**- og **hvis ellers**-klosser som tester for de tre tilfel
dersom romskipet lander på landingsplassen, **fartY** er større e

- ☐ Finn en måte å fortelle spilleren hvordan romskipet landet. Det
kan også bruke lydeffekter, forskjellige drakter eller kanskje en

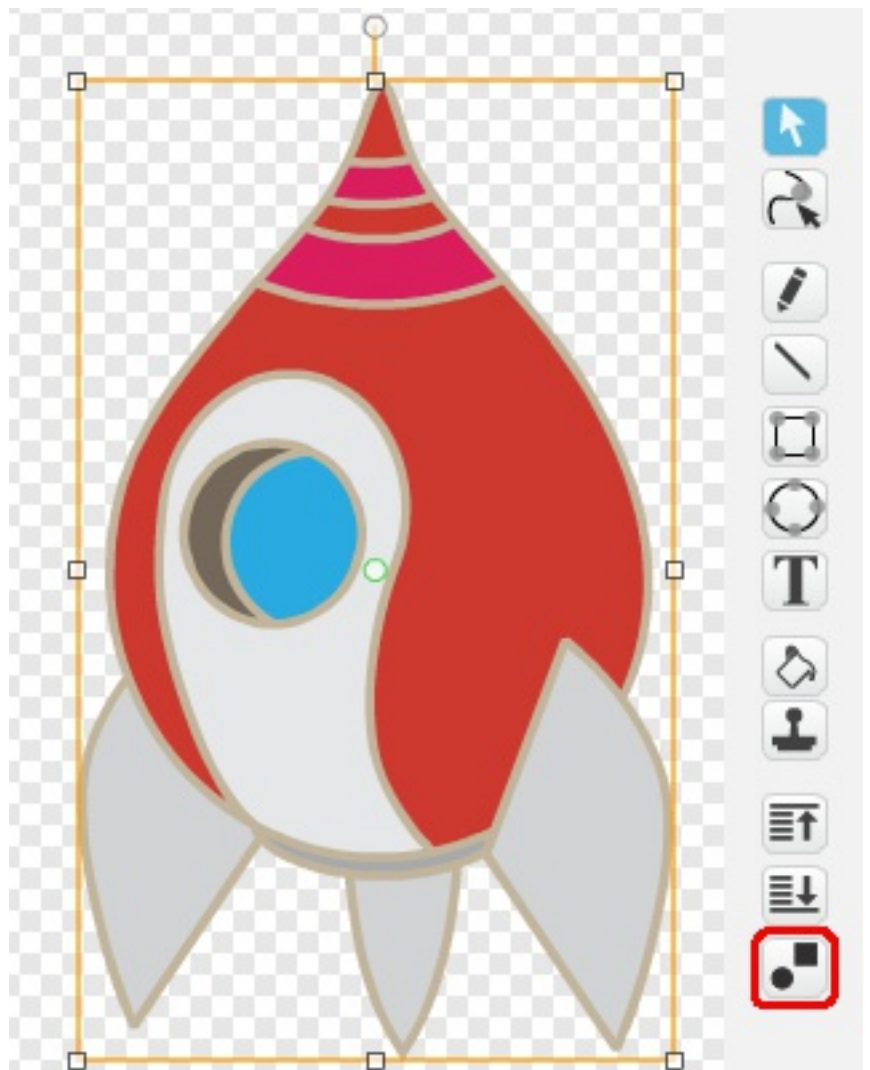
Steg 4: Videreutvikling av

*Du har nå laget en enkel variant av Lunar Lander. Men prøv å gjøre sp
bestemmer selv hvordan du vil jobbe videre, men nedenfor er noen ic*

✓ Ideer til videreutvikling

- ☐ Lag flere landingsplasser. Kanskje noen kan være vanskeligere å
eller fordi selve landingsplassen er mindre).

- ☐ Lag flere bakgrunner eller brett. Du kan enten la spilleren velge eller gå videre fra brett til brett etterhvert som spilleren klarer å
- ☐ Du kan også bruke flere bakgrunner som et brett. Dersom roms dukke opp igjen på venstre side av en annen bakgrunn, og mot
- ☐ Prøv å animere bruken av motoren. For eksempel, kan du lage e som følger. Klikk først på romskipet, og deretter på den nye kna med rødt i figuren til venstre). Denne vil dele opp romskip-figuren bruk malingsspannet til å fargelegge de tre finnene på romskipe motoren.



- ☐ En av utfordringene i det originale Lunar Lander-spillet var at sp

kan legge til dette i ditt spill ved å lage en **Drivstoff**-variabel som trykker på piltastene. Dersom romskipet går tomt for drivstoff kan romskipet sveve til det til slutt kræsjer.

- ☐ Du kan lage forskjellige bonusing som spilleren kan plukke opp på måneoverflaten slik at spilleren må manøvrere forsiktig for å ekstra drivstoff, bonuspoeng, større landingsplass og så videre.
- ☐ En større endring i spillet kan være at landingsplassen er skjult grave denne frem ved å slippe bomber fra romskipet. Du kan innklossene under **Penn**-kategorien.

Lisens: [CC BY-SA 4.0](#) **Forfatter:** Geir Arne Hjelle