

Introduksjon

I denne oppgaven skal vi ta utgangspunkt i animasjonene vi lagde i oppgaven [Partikkel-animasjon](#). Dersom du ikke har gjort denne oppgaven anbefaler vi deg om å gå tilbake å gjøre den før du fortsetter her.

Oppgaven her går ut på å legge til gravitasjon på `Partikkel`-objektet. Du vil få presentert hva du skal gjøre, så du må selv finne ut hvordan man skriver kode for å få det til. Til slutt kommer det til å se noe slik ut:



Steg 1: Få partikkelet til å falle

Det første vi skal gjøre er å få partikkelet til å falle, for å få det til å skje må du programmere følgende:

- ☐ Legg til to attributter i `particle` som skal holde styr på `gravitasjonen` og `hastigheten` den faller med. `Gravitasjons`-attributtet bør være et lite tall, som for eksempel `0.01`. `Hastigheten` bør være `0` fra starten av.
- ☐ Sett et passende startsted for `particle`, helst høyt oppe

For hver gang `draw()` kjører, skal følgende skje:

- ☐ `gravitasjons`-attributtet skal endres med `hastighets`-attributtet
- ☐ `particle` sin `y`-posisjon blir endret med `hastighets`-attributtet

Hint

Faller partiklet nå? Nå som partikkelet faller ser du kanskje også at det faller igjennom skjermen, derfor må vi legge til en sjekk om gjør at den stopper nederst.

Steg 2: Stopp nederst på skjermen

For at partikkelet vårt skal stoppe nederst på skjermen trenger vi enkelt og greit en sjekk hver gang vi kjører `draw()`:

- ☐ Lag en sjekk som sjekker om `particle` sin `y`-verdi er større enn høyden på `canvas`-objektet. OBS! Du vil sikkert merke at den "rister" helt nederst, derfor må du også ta hensyn til størrelsen på partikkel-objektet når du sjekker hvor langt ned den er.

Hint

- ☐ I sjekken må du stoppe partikkelet ved å sette `y`-verdien. Kan du tenke deg til hva `y`-verdien bør være?

Nå har du et fallende objekt!

Utfordringer

- ☐ Når partikkelet treffer kanten, snu gravitasjonen sånn at partikkelet går oppover
- ☐ Klarer du å få partikkelet til å gå til venster eller høyre også?
- ☐ "Steng" alle kantene i `canvas` ved hjelp av flere `if`-setninger

Steg 3: Få partikkelet til å fly!

Nå som vi har fått gravitasjon på partikkelet vårt kan vi legge til styring på partikkelet ved hjelp av knapper. Vi skal nå programmere slik at når du trykker `pil opp` så gir vi partikkelet negativ akselerasjon og når vi slipper så blir den påvirket av gravitasjonen. Får å gjøre dette bruker vi noe som heter `onkeyup`, `onkeydown` og `keyCode`.

For å kunne utføre en operasjon når vi trykker på en knapp må vi ha denne koden:

```
window.onkeydown = function(inputKey) {  
  var key = inputKey.keyCode ? inputKey.keyCode : inputKey.which;  
  
  if (key == //knapp-koden) {  
    // Kode som skal utføres  
  }  
}
```

- ☐ Bruk `keyCode.info` til å finne ut hvilke `knapp-kode` knappen du vil bruke har.

Forklaring

- ☐ `window.onkeydown` er en funksjon som sjekker om en knapp trykkes ned, dersom den gjør det, så kjøres funksjonen med `knapp-kode` `inputKey`.
- ☐ `var key = inputKey.keyCode ? inputKey.keyCode : inputKey.which;` er en enklere måte å skrive dette på:

```
if(inputKey.keyCode) {  
  var key = inputKey.keyCode;  
} else {  
  var key = inputKey.which;  
}
```

Grunnen til at du må bruke `keyCode` og `which` er fordi ikke alle nettlesere støtter `keyCode`, men bruker `which` istedet.

- ☐ I `if (key == //knapp-kode)` må du skrive kode for at gravitasjonen skal være negativ, prøv med forskjellige tall.

- ☐ For at du skal få gravitasjonen tilbake når du slipper knappen må du lage en funksjon for `window.onkeyup`.
- ☐ Les mer om `window.onkeyup` osv her: [w3school](https://www.w3schools.com/jsref/prop_window_onkeyup.asp)

Dersom du står bomfast: [Forlag til kode](#)

Lisens: CC BY-SA 4.0