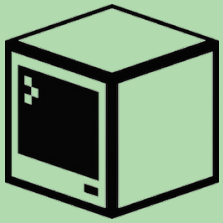


Hendelser

Introduksjon

I denne leksjonen skal vi se nærmere på hvordan datamaskinene bruk
tastetrykk, museklikk og så videre. Etterhvert vil vi bli enda bedre kje
og programmer rundt omkring på en datamaskin eller mellom forskje

Steg 1: Skattejakt

Datamaskiner bruker noe som kalles hendelser for å registrere tastetr
vi kan lage et enkelt spill hvor vi styrer en figur med piltastene.

Sjekkliste

- [ ] Start et nytt program ved å skrive `edit skattejakt`, skriv inn d

```lua
local x = 20
local y = 10

term.clear()
term.setCursorPos(x, y)
print('O')
```

Lagre og kjør programmet. Skjønner du hva det gjør? Prøv å fora

Så langt skriver programmet bare ut en `O` på en gitt posisjon. N
figuren rundt på skjermen.

For å vente på hendelser bruker vi `os.pullEvent()`. Vi vil spesi
`key` i ComputerCraft. Forandre programmet ditt som under:

```lua
local x = 20
local y = 10

term.clear()
term.setCursorPos(x, y)
print('O')

local hendelse, tast = os.pullEvent('key')     -- ny linj
print(hendelse)                                -- ny linj
print(tast)                                    -- ny linj
```

Når du kjører programmet ditt nå blir programmet stående og v
key samt et tall til skjermen. Teksten **key** betyr bare at det var
interessant for oss nå, men kan være nyttig i andre programme

Tallet vi fikk er derimot veldig viktig. Hver tast på tastaturet har
ganger, ser du at om du trykker samme tast får du samme tall t
alltid er 200.

Vi trenger heldigvis ikke huske disse kodene. Biblioteket `keys` k
`keys.q` for å representere `Q`-tasten. Endre programmet ditt igj

```lua
local x = 20
```

```
local y = 10

term.clear()
term.setCursorPos(x, y)
print('O')

local hendelse, tast = os.pullEvent('key')
if tast == keys.q then                          -- ny linj
    print('Du trykket Q')                       -- ny linj
else                                            -- ny linj
    print('Du trykket ikke Q')                  -- ny linj
end                                             -- ny linj
```

Vi kan nå lage en løkke hvor vi alltid sjekker hvilken tast som er
med `break`.

```
local x = 20
local y = 10

while true do                                   -- ny linj
    term.clear()
    term.setCursorPos(x, y)
    print('O')

    local hendelse, tast = os.pullEvent('key')
    if tast == keys.q then
        break                                   -- endret
    end
end                                             -- ny linj
```

Når du kjører dette programmet vil det tilsynelatende ikke skje
hendelsen vi har kode som reagerer på.

Vi er nå klare til å sjekke om piltastene trykkes, og flytte figuren
endre verdiene av `x` og `y` avhengig av hvilken piltast som tryk

```
local x = 20
```

```
local y = 10

while true do
    term.clear()
    term.setCursorPos(x, y)
    print('O')

    local hendelse, tast = os.pullEvent('key')
    if tast == keys.q then
        break
    end

    if tast == keys.right then                    -- ny linj
        x = x + 1                                 -- ny linj
    end                                           -- ny linj
end
```

Når du kjører dette programmet vil du se at du kan bruke `pil h`

Prøv selv

De andre piltastene kan du programmere selv på samme måte. Du `keys.down` og `keys.up`. Hvordan må du endre verdiene av `x` og `y`

Sjekkliste

- [ ] Til slutt vil vi legge til en skatt som figuren vår skal lete etter. Vi ligge. Endre begynnelsen av programmet ditt som følger:

```
local x = 20
local y = 10
local skattX = math.random(1, 50)             -- ny linj
local skattY = math.random(1, 18)             -- ny linj
```

```
while true do
    term.clear()
    term.setCursorPos(skattX, skattY)          -- ny linj
    print('X')                                 -- ny linj
    term.setCursorPos(x, y)
    print('O')

    if x == skattX and y == skattY then        -- ny linj
        term.setCursorPos(1, 1)                -- ny linj
        print('Du fant skatten!')              -- ny linj
        break                                  -- ny linj
    end                                        -- ny linj

    local hendelse, tast = os.pullEvent('key')
    -- resten av programmet er som tidligere
```

Prøv spillet! Fungerer det som du hadde trodd? Klarer du å kans

Prøv selv

Ved hjelp av `local maxX, maxY = term.getSize()` kan du finne stø
begrense figuren din slik at den ikke kan gå av skjermen?

En litt utfordrende oppgave: Prøv å skriv et tilsvarende program til
en figur på skjermen skal piltastene flytte roboten!

Steg 2: Hvordan bevege se filsystemet

Datamaskiner organiserer informasjon i filer, og disse filene legges i e
vanlig datamaskin i programmene Windows Utforsker eller Finder på M
se på filene. Vi skal her se på noen enkle kommandoer for å kopiere o

Sjekkliste

- [ ] Lag en ny **Computer**, sett den ut og start den ved å høyre-klikk

- [ ] Kommandoen `dir` brukes for å se på innholdet i en katalog (dir
  Prøv den nå! Skriv `dir` og trykk enter.

  Datamaskinen svarer **rom** og **skattejakt**. Den første er en kata
  på datamaskinen (rom er en forkortelse for *Read Only Memory* s
  Vi skal se mer på denne katalogen senere.

- [ ] Når du bruker `dir` er det vanskelig å se forskjell på filer og kata
  å skriv `type skattejakt`. Datamaskinen vil da fortelle deg at **sk**
  deg at **rom** er en katalog.

- [ ] Du kan lage egne kataloger om du vil, for å organisere filene dir
  `mkdir` lager nye kataloger (mkdir er en forkortelse for *make dir*
  katalogen **mine_programmer** ble laget ved å skrive `dir` og `t`

- [ ] Kommandoen `move` flytter filer. Skriv `move skattejakt mine_pr`
  katalogen **mine_programmer**. Hvis du nå skriver `dir` vil du se
  flyttet riktig kan du skrive `dir mine_programmer`. Dette viser all
  mine_programmer.

- [ ] Vi kan også flytte oss rundt i filsystemet. Dette vil si at vi endre
  eksempel når vi skriver `dir`). Til dette bruker vi `cd` (cd er en fo
  katalog). Skriv `cd mine_programmer`. Du vil se at det som står fø
  i. Prøv også å skriv `dir` for å bekrefte at du er i samme katalog

  For å gå tilbake en katalog bruker du det spesielle navnet **..**. Skr

Prøv selv

Du har nå sett ganske mange kommandoer: `dir`, `edit`, `type`, `mk`
kan brukes til å slette filer og kataloger, og `copy` som brukes på sa
stedet for å flytte dem.

Prøv å bruke disse kommandoene til å flytte deg litt rundt i filsyste
videre inntil du er ganske komfortabel med hvordan filsystemet fur

Steg 3: Et bedre passord-p

Vi har tidligere laget et passord-program. Dette passord-programmet

- [ ] Programmet må startes manuelt ved å skrive `passord` etter at
- [ ] I stedet for å skrive passordet kan man bare trykke `Ctrl-T` for
finne det hemmelige passordet.

Vi skal her se på et par triks for å gjøre passord-programmet litt trygg

Sjekkliste

- [ ] Om du ikke allerede har gjort det: Sett opp en datamaskin ved s
datamaskinen, det vil si skriv `edit passord` og skriv inn følgen

```
local passord = 'kodeklubben'

while true do
    term.clear()
    term.setCursorPos(1, 1)
    print('Hva er passordet?')
    svar = read('*')

    if svar == passord then
        redstone.setOutput('left', true)
        sleep(5)
        redstone.setOutput('left', false)
    end
end
```

Kjør programmet, og sjekk at det virker som det skal.

- [ ] Når en datamaskin starter sjekker den først om det finnes et pr
Om den finner dette programmet kjøres dette før noe annet skj
for **startup** så vil det kjøre automatisk.

Skriv `move passord startup`. Dette endre navnet på passord-p
ved å skrive `reboot`. Datamaskinen vil nå direkte spørre deg or

- [ ] Trykk `Ctrl-T` for å stanse passord-programmet. At vi kan bruke
`os.pullEvent` gjør for oss automatisk uten at vi trenger å gjøre
bytte ut `os.pullEvent` med noe som heter `os.pullEventRaw`. D
bryr seg ikke om `Ctrl-T`.

Skriv `edit startup` og legg til en linje øverst i koden din:

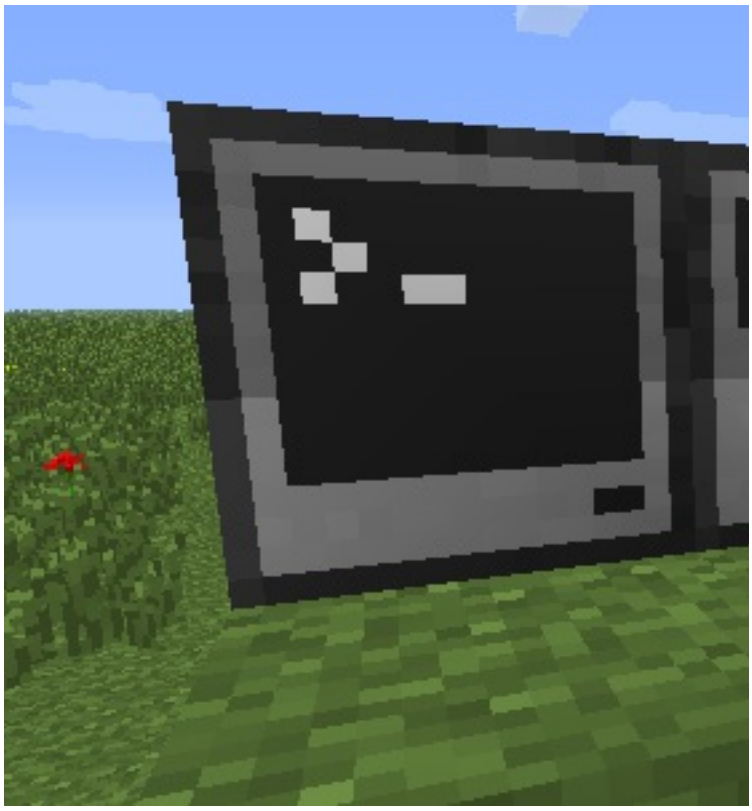
```
os.pullEvent = os.pullEventRaw                    -- ny linj
local passord = 'kodeklubben'

while true do
    term.clear()
    term.setCursorPos(1, 1)
    print('Hva er passordet?')
    svar = read('*')

    if svar == passord then
        redstone.setOutput('left', true)
        sleep(5)
        redstone.setOutput('left', false)
    end
end
```

- [ ] Du kan nå starte datamaskinen på nytt igjen med `reboot`. Nå h
tiden!

  Vær litt forsiktig med dette siden det ikke er noen enkel måte å
passord! Men det du lærer i de to neste stegene kan være nytti

Steg 4: Bruk av diskettstas

Vi skal nå se hvordan vi kan bruke disketter og diskettstasjoner til å fl



Sjekkliste

- [ ] Lag en **Disk Drive** (diskettstasjon) og plasser den inntil datama
inventory'et ditt (du kan velge hvilken farge som helst).

- [ ] Åpne diskettstasjonen ved å høyre-klikke på den. Sett inn diskett

- [ ] Start datamaskinen din ved å høyre-klikke på den. Skriv `dir`. D
Dette er disketten vi nettopp satte inn.

- [ ] La oss lage et enkelt program. Skriv `edit navn` og skriv inn følg

```
print('Hva heter du?')
navn = read()
print('Hei, ' .. navn)
```

Test at programmet virker ved å skrive `navn`.

- [ ] Vi kan nå kopiere dette programmet over til disketten ved å skri

- [ ] Vi kan nå ta med oss dette programmet til en annen datamaskir

  1: Steng datamaskinen.

  2: Åpne diskettstasjonen, og flytt disketten til inventory'et ditt.

  3: Lag en ny datamaskin, også denne med en diskettstasjon inn

  4: Sett disketten inn i den nye diskettstasjonen.

  5: Åpne den nye datamaskinen. Skriv `dir` og `dir disk` slik at
  datamaskinen ved hjelp av disketten.

- [ ] Vi kan nå kopiere programmet fra disketten til denne nye datam
  likevel bruke programmet. For å kopiere filen kan du skrive `copy`
  på slutten. Dette er et spesielt katalognavn som alltid betyr *der*
  at programmet har blitt kopiert.

Hva er en diskett?

Disketter var en vanlig måte å lagre programmer og filer på fra de
2000-tall. Disketter var også den vanligste måten å overføre filer m
internett, samt USB minnepenner og eksterne harddisker overtatt

Morsomt nok, lever likevel diskettene videre som det mest vanlige

Steg 5: Skrive kode utenfo

Vi kan også se på og endre programmene våre utenfor Minecraft og C

1: Om vi ved et uhell ødelegger en datamaskin kan vi hente tilbake pr
en annen datamaskin.

2: Vi kan raskere kopiere filer mellom datamaskiner enn om vi bruker

3: Programmet `edit` som vi bruker til å skrive programmer er ikke så Notepad eller andre tekstprogrammer vi har installert.

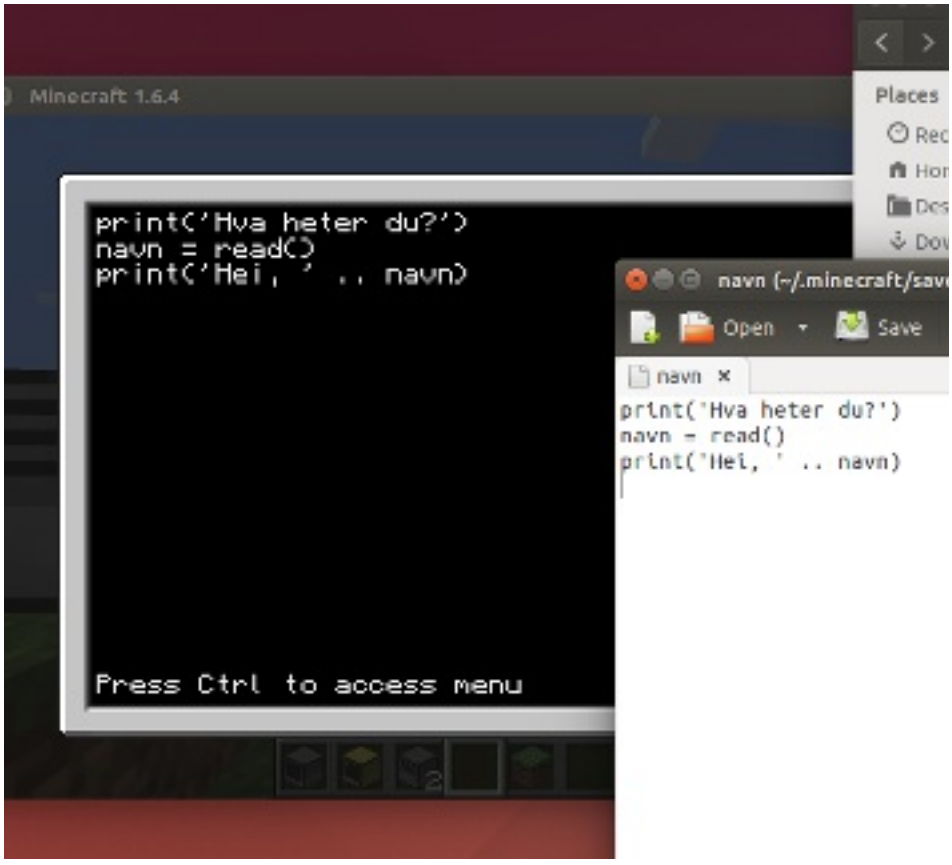
Som du kanskje vet blir omtrent alle data i Minecraft lagret i en *Minec* kan du gjøre følgende (utenfor Minecraft):

Windows: Under Windows finner du *Minecraft*-katalogen under `%appd` eller i kjør-feltet etter å ha klikket start-knappen.

Mac OS X: Under Mac ligger *Minecraft*-katalogen i `Library/Applicat` På norsk heter `Library` `Bibliotek`.

Linux: På Linux finner du *Minecraft*-katalogen som en skjult katalog `.`

Finn *Minecraft*-katalogen din i en filutforsker. Gå videre til katalogen `s` spiller nå, og til slutt katalogen `computer`. Denne katalogen inneholde katalogene representerer de forskjellige datamaskinene i spillet ditt. D nummererte kataloger inne i seg. Disse representerer diskettene i spil



Sjekkliste

- [ ] Gå tilbake til Minecraft-spillet ditt. Åpne en datamaskin hvor du vil fortelle deg hvilket nummer denne datamaskinen er.
- [ ] I filutforskeren kan du nå finne katalogen som representerer der i et tekstprogram som for eksempel Notepad. Gjør en liten endr
- [ ] Gå tilbake til Minecraft igjen. Åpne det samme programmet mec gjorde?
- [ ] I filutforskeren kan du også kopiere filer mellom forskjellige data endret til en annen datamaskin. Finner du igjen dette programm

Flytt en datamaskin

Om du oppdager at du må flytte en datamaskin må du være litt for og setter ut en ny er alle programmene borte. Du kan da bruke me programmene tilbake, men det finnes en bedre måte.

Med kommandoen `label` kan vi gi en datamaskin navn. Prøv for e denne datamaskinen navnet **snakker**, du kan gi maskinen akkurat vil du se at du kan plukke den opp igjen, og at den da legger seg i setter ut maskinen igjen vil du se at alle programmene du har skre

Steg 6: De innebygde prog

Vi skal nå kikke raskt på katalogen **rom**. Dette er som nevnt kataloge datamaskinen. Ved hjelp av kommandoene vi har lært kan vi nå se hv med lage våre egne versjoner av dem.

Sjekkliste

- [ ] Bruk `cd` kommandoen til å gå først til **rom**, deretter **programs** underveis for å se på hvilke andre filer og kataloger som finnes.

- [ ] Denne **fun**-katalogen inneholder flere spill og programmer. La o er en variant av **heisann** som vi skrev tidligere.

  Prøv først å kjøre programmet ved å skrive `hello`. Teksten **Hell**

- [ ] La oss se på koden til **hello**. Skriv `edit hello`. Du vil se det føl

  ```
  if term.isColour() then
      term.setTextColour( 2^math.random(0,15) )
  end
  textutils.slowPrint( "Hello World!" )
  term.setTextColour( colours.white )
  ```

  Ser du hvilken kodelinje det er som har ansvaret for å skrive tek

- [ ] Nå vil vi endre teksten **Hello World!** til noe annet. Men om du ikke går an. Videre, om du trykker `Ctrl` vil du se at valget `Save` **rom**, det skrivebeskyttede minnet.

- [ ] Hvis vi vil lage vår egen versjon av **hello** må vi først kopiere file mystiske rekken av `..` og `/` betyr at vi kopierer filen tre nivåer

- [ ] Nå vil vi flytte oss tilbake til utgangspunktet eller roten av filsys skal du se filen **hello** i tillegg til **rom**.

- [ ] Nå kan du skrive `edit hello` og endre teksten **Hello World!** ti ditt? Lagre og lukk filen, og skriv deretter `hello` for å se om du

Resten av programmet

Vi har så langt bare brydd oss om linje 4 i **hello**-programmet. Skjø

Ut fra kommandoene `term.isColour()` og `term.setTextColour()`
tekstfargen å gjøre? Faktisk sier de første tre linjene at dersom pro
skal tekstfargen settes til en tilfeldig farge. Den siste linjen setter t

Om du vil se hvordan dette virker kan du prøve å lage en **Advance**

Prøv selv

Prøv å se på noen av de andre programmene du kjenner til, som fo
lete litt i katalogstrukturen for å finne dem. Disse programmene er
Prøv likevel å se om du skjønner hva deler av koden gjør. Finner du
eller linjen som rapporterer fuelnivået i `refuel`?

Steg 7: Andre typer datam

Vi har så langt stort sett bare brukt vanlige datamaskiner, **Computer**
flere andre typer datamaskiner, inkludert **Advanced Computer** og **P**
vanlig datamaskin, men har noen ekstra muligheter.

Sjekkliste

- [ ] Lag og start en **Advanced Computer**. De viktigste ekstra muli
farger og du kan bruke musen.

- [ ] La oss lage et enkelt tegneprogram. Skriv `edit tegne`, og skriv

```
term.clear()

while true do
    local hendelse, knapp, x, y = os.pullEvent('mouse_cl
    print('Du klikket ' .. knapp)
    print('Posisjon: x = ' .. x .. ', y = ' .. y)
end
```

Kjør programmet og prøv å klikk litt rundt omkring på skjermen.
hvordan museklikk-hendelser fungerer? Bruk `Ctrl-T` for å avslu

- [ ] La oss legge til litt kode som tegner på skjermen når du venstre
det som er tegnet. Endre koden til

```
term.clear()

while true do
    local hendelse, knapp, x, y = os.pullEvent('mouse_cl
    term.setCursorPos(x, y)                          -- ny linj

    if knapp == 1 then                               -- ny linj
        print('#')                                   -- ny linj
    end                                              -- ny linj

    if knapp == 2 then                               -- ny linj
        print(' ')                                   -- ny linj
    end                                              -- ny linj
end
```

Prøv selv

Kan du legge til farger i tegneprogrammet? Se tilbake på `hello`-pr
`term.setTextColour()` til å endre farge. Kanskje du kan bruke tallt

Når du vil sjekke forskjellige typer hendelser, for eksempel både m

`os.pullEvent()` uten navnet på en hendelse i parantes. Deretter k
hendelse som faktisk skjedde.

Etterhvert som du tegner merker du at du må klikke hver gang du v
kunne bare klikket en gang, og deretter dra musen rundt. Se på he
programmet ditt.