

Assignment 2

Learning Python

29.11.2013



Universitetet I Agder
Is - 206 Åpen Kildekode Programvare
Av Martine Sivertsen Nilsen

Denne teksten vil være kortere enn «Assignment 1». Det var ikke krav om å skrive en tekst til «Assignment 2» eller «Assignment 3», men jeg har valgt å gjøre det, da jeg mener det er ting jeg ikke kan uttrykke gjennom kode, men som kan uttrykkes gjennom tekst. Hensikten er å vise at jeg har forstått oppgaven. Strukturen i denne teksten vil ikke følge «Bachelor Template».

Hva går Assignment 2 ut på?

“The aim of this assignment is simple, to learn to program in Python programming language. To approach this aim, you are expected to do exercises in <http://learnpythonthehardway.org/book/>. All of your code and written material related to the learning of Python have to be committed in GitHub. The only thing to deliver in Class Fronter will be a public URL for your GitHub repository”(Janis Gailis, 2013).

Hvordan?

Tutorialen vi har fulgt(Learn python the hard way) skal munne ut i et spill. Jeg har valgt å lage et “vanlig” spill med 5 rom/filer. Spillet er svært enkelt. Spillet forteller en historie, enden på historien er basert på brukerens input. Det er også implementert et minispill i spillet - Stein Saks Papir, ved hjelp av random funksjon.

Walkthrough av spillet

spilleren starter i hovedrommet, en historie blir fortalt, brukeren blir anbefalt å gå vest, her er det en slosskamp. spilleren vinner spillet ved å taste “punsh” det blir så senere anbefalt å gå nord, eller spille et spill(stein saks papir). Går spilleren nord, åpnes et nytt rom. Spilleren kan så gå sør. Spillet er ikke ferdig utviklet, det er derfor ingen ordentlig slutt på spillet. Foreløpig kan spilleren dø, på forskjellige måter avhengig av hvilket rom spilleren er i. Hensikten med utviklingen av spillet er å få en basis forståelse av Python. Noe jeg mener jeg har, da jeg kan skrive classer, løkker og bruke funksjoner fra python biblioteket. Jeg påstår ikke at jeg er en python ekspert, men at jeg har et godt utgangspunkt for videre bruk av python.

Illustrasjon, hovedrommet til spillet

```
from sys import exit # from sys import exit. Exit is a fuction that exits the program when called
# Import my other files, which will be used as rooms for this game
import east
import north
import west
import south

def dead(why): #A function to exit the program, if the player dies

    print "You walk around until you die, goodbye sir!"
    exit(0)

def start():#My main function, automatically started when the file "hovedrom" is called.
    print """
    Welcome to our humble town Rekssand.
    Many years ago Oggie Boogie casted darkness ouver our city.
    Within the darkness lies an evil power that makes the citizens insane.
    Help us get back our sunlight to rescue Rekssand!
    Press CTRL-C to cancel
    or ENTER if you wish to start your journey. """

    continue = raw_input("> ") #Raw input from user, used to determin the next direction in the game

    print """
    You are standing in our main street, to your west you see man
    running, you can also go east, north or south, which direction
    do you take?"""

    next = raw_input("> ")

    if next == "east": # if input from user is east, go to file east, rund function get_winner
        east.get_winner()
    elif next == "south":
        south.south()
    elif next == "north":
        north.north()
    elif next == "west":
        west.west()
    elif next == "play":
        east.game()
    else:
        dead("You walk around until you die, goodbye sir!")

start()
```

Tanker og refleksjoner rundt Assignment 2:

Vi brukte “Learning Python the hard way” for å tilegne oss kunnskap om Python. Denne tutorialen inneholder 52 oppgaver av varierende størrelse, å er ment for læring over en lengre periode, eksempel 3 - 6 måneder(Learning Python the hard way, 2013). Vi har gått igjennom denne tutorialen på 3 uker(Fronter, IS-206- Initial plan - 2013.pdf). Pga. den korte tidsrammen som ble gitt finnes lite rom for refleksjon over det man har lært. Dette medfører at jeg raskere glemmer det jeg har lært i tutorialen. Jeg mener at dette er noe av grunnen til at jeg opplevde tutorialen som svært vanskelig mot slutten. Samtidig er det selvforklarende at en tutorial som skal utføres på 3 - 6 måneder har en større arbeidsmengde enn tidsrammen vår innfridde. Samtidig vil jeg trekke fram at jeg mener tutorialen var godt strukturert. Det finnes også gode forum som er knytta opp mot tutorialen. På disse forumene har man mulighet til å stille spørsmål eller lignende hvis man opplever problemer med tutorialen.

EX26 Fix someone elses code

Koden som skulle fikses var relativt simpel. Det var hovedsaklig lagt inn skrivefeil som skulle rettes opp i. Disse var enkle å oppdage, å jeg mener at en hver person som kan lese, har mulighet til å oppdage samt rette opp i disse feilene. Det var også lagt inn enkelte “feil” i koden som krevde at leseren hadde kunnskap til python, men heller ikke disse feilene var vanskelig å verken finne eller rette opp i.

EX45 Make your own intelligent agent

- tok inspirasjon fra tidligere ex
- Skrev litt kode av gangen, testet denne koden og expanderte videre ut fra det.

EX46 - EX52

I ex46 skal man opprette et prosjekt skeleton. Det vil si et rammeverk man kan bruke når man senere oppretter prosjekter. For å gjøre dette må man installere pip, distribute og nosetests.

Disse 3 programmene installeres fra sourcecode, alle 3 kom med setup.py. Jeg trodde dette skulle bli enkelt, men når jeg skulle installere første program fikk jeg

problemer med rettigheter på min egen bruker på min maskin. Jeg løste dette ved å bruke kommandoen “SUDO” for å få tilgang. Jeg søkte også rundt på forum med fokus på installasjon av disse programmene i OSX. Det fantes gode veiledninger og derfra var det ingen problem å installere.

Fra ex46 til ex52 har det vært mye å gjøre. Det virket faktisk uoverkommelig ved første øyekast. Når jeg fikk satt i gang med disse gikk det overraskende greit sammenlignet med hva jeg først antok. Jeg var underveis helt avhengig av muligheten til å google feilmeldinger etc. Jeg klarte ingen av deloppgavene ved første forsøk, jeg måtte hele tiden endre små ting for å få det til å fungere. Dette var demotiverende, men mestringsfølelsen jeg fikk når jeg var ferdig var veldig motiverende.

Litt om Kodestruktur

Jeg har valgt å kommentere koden der jeg har behov for det, brukes et «Statement» eller «Funksjon» flere ganger, er det som regel kun forklart første gang det ble brukt. Jeg ønsker ikke å ha 40-50 ekstra linjer med kommentarer som forklarer eks «print». Før jeg begynte å kommentere koden, har jeg prøvd å lese meg opp på python filosofi. Jeg prøvde å følge anbefalt kodestruktur, kommentar filosofi og generelle normer for utvikling av python. Jeg mener det er viktig å tilegne seg gode programmeringsrutiner, selv om jeg ikke alltid er like flink til dette.

Det finnes flere gode kilder til informasjon og tutorials om python i dag, et enkelt google søk vil derfor forklare metoder, begreper og lignende i koden. Dette er også en sikkerhetslinje for meg selv, som hjelper meg til å forstå kode.

Som verktøy for å skrive python har jeg brukt Textvangler. Textvangler har forhåndsinnstillinger som hjelper en med å strukturere koden. Det er derfor mye automatikk i denne prosessen.