

Programmering og Problemløsning

Datalogisk Institut, Københavns Universitet

Arbejdsseddel 1 - gruppeopgave

Jon Sparring

6. september - 11. september.
Afleveringsfrist: lørdag d. 11. september kl. 22:00.

Velkommen til kurset “Programmering og Problemløsning”

Kurset består af forelæsninger og øvelser. Forelæsningerne er både video-, on-campus og streamede forelæsninger og gives af forelæserne Jon Sparring (kursuskoordinator), Ken Friis Larsen og Martin Elsmann. De vil omhandle de væsentligste teoretiske elementer i programmering for begyndere. Til øvelserne er alle fordelt på øvelseshold og fokuserer på de praktiske elementer i programmering for begyndere. Til hvert øvelseshold er der en instruktør, og øvelsesholdene vil være on-campus. Kurset ligger i skemagrupper A (tirsdag formiddage og torsdag).

Dette er kursets første *arbejdsseddel*. Vi har 16 undervisningsuger, og de fleste arbejdsedler vil svare en undervisningsuge. Arbejdsedlerne vil beskrive pensum, som gennemgås til forelæsningerne, øvelsesopgaver, som der vil blive arbejdet med under øvelserne, og afleveringsopgaver, som der også bliver tid til at kigge på til øvelserne. Pensum angives under “Forelæsnings- og læseplan” på Absalon og evt. tilhørende materiale finder under Absalonpunktet “Noter, links, software m.m.”. Ca. halvdelen af afleveringsopgaverne bliver individuelle opgaver, alle andre opgaver løses i grupper.

Bemærk at langt de fleste opgaver på disse arbejdsedler vil være på engelsk, da vi er igang med at oversætte dem til Fsharp bogen. Denne omgivende tekst har vi dog valgt at beholde på dansk.

Emnerne for denne arbejdseddel er:

- Programmere et spil i Scratch,
- skrive en rapport i LaTeX vha. en teksteditor og kommandolinjen,
- aflevere en opgave via Absalon.

Opgaverne er opdelt i øve- og afleveringsopgaver. I denne periode skal I arbejde i grupper med jeres afleveringsopgaver. Regler for gruppe- og individuelle afleveringsopgaver er beskrevet i ”Noter, links, software m.m.” → ”Generel information om opgaver”.

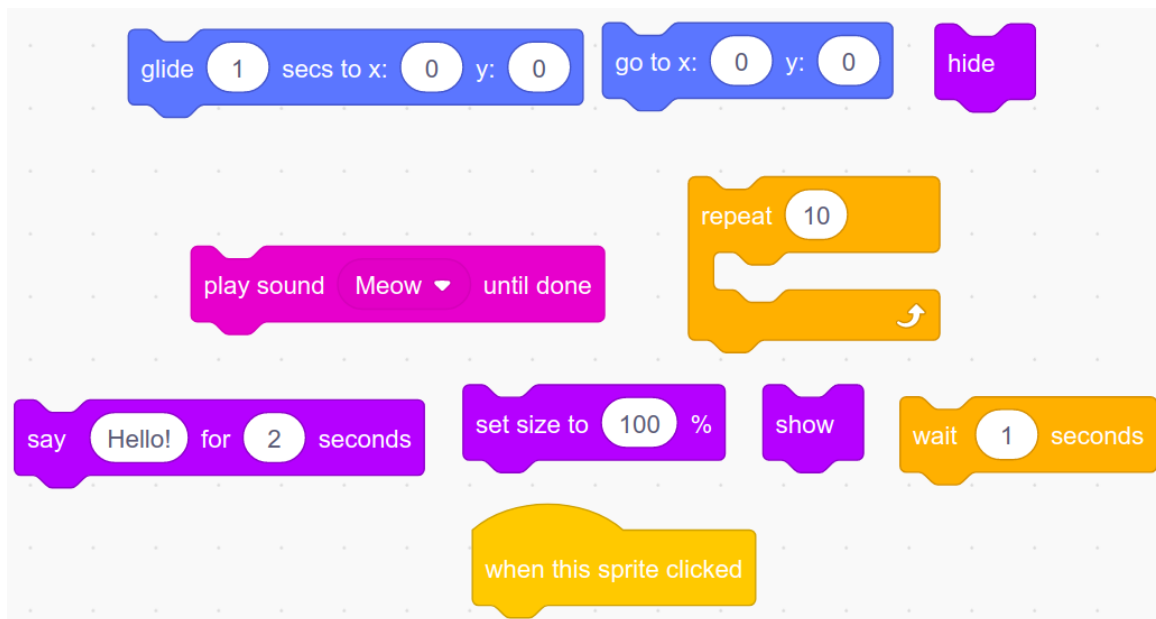
Øveopgaver (in English)

- 1ø0 Install Scratch on your machine.
 - 1ø1 Make your own “hello world” Scratch-program. The program must make the default sprite say “Hello World” when you press the green flag.
 - 1ø2 Make a Scratch program with a sprite of your own choosing, which moves on the screen using the ‘glide’-block and the ‘forever’ loop-block.
 - 1ø3 Make a Scratch-program, which counts down from 10 to 1. The countdown should start/begin when you click the mouse.
 - 1ø4 Take one or more screenshots of one of your Scratch-programs while it runs.
-

- 1ø5 Start the command line (or terminal on MacOS). Use the `cd`-command to navigate to a suitable directory for your work. (e.g. the Documents folder). Use the ‘`mkdir NAME`’ command to create a new directory from the command line. Replace NAME with the name of your new directory.
- 1ø6 Write a short report in \LaTeX with a text editor and translate the `tex`-file to a `pdf`-file using the command line. The report should as minimum contain:
 - A title produced using `\maketitle`,
 - A section with a section title using `\section`,
 - One or more figures of screenshots from your program, using the `figure`-environment. All figures must include a caption text using `\caption`.
 - A reference to the figure using the `\label`–`\ref` pair.
 - The Danish letters ‘æ’, ‘ø’, and ‘å’.

Afleveringsopgaver (in English)

- 1g0 What can you make with 10 different Scratch-blocks?
In Figur 1 is shown 10 Scratch-blocks. Your task is to make a fun program only by using these blocks. Each block may be used 0 or more times. Try first to connect the blocks on paper, write down what you think the program will do and then create the program in Scratch and run it. Describe to what extent the program did as expected.
- 1g1 Design a game in Scratch:
You are to design a game of your own choosing. The game must
 - include 2–5 sprites
 - have a typical gameplay of about 1 minute
 - must include at least 1 variable



Figur 1: 10 Scratch-blocks

You may use any existing block in Scratch, and the game may be similar to an existing game. The graphical appeal and the sound aspects of the games are of little importance.

A good approach is to:

- Start by brainstorming about a game, you would like to make and what the game mechanics should be.
- Sketch a design on paper about the gameplay.
- Implement your design as a sketch of a Scratch program, still on paper.
- Enter your prototype into Scratch and test it.
- Return to the top and update your game until you are satisfied with the result.

Krav til afleveringen

Afleveringen skal bestå af

- upload af de udviklede programmer i øvelsesholdets studio på Scratch' hjemmeside,
- en zip-fil og
- en rapport i pdf-format.

Rapporten skal dokumentere jeres proces og beskrive jeres programmer. Den skal skrives i \LaTeX , indeholde et passende antal figurer og henviser til jeres programmer (i øvelsesholdets studio på Scratch' hjemmeside). \LaTeX koden (.tex) inklusiv billeder skal organiseres i en mappe kaldet tex, således at pdf-filen kan laves ved kørsel af pdf_latex kommandoen i mappen. Mappen skal zippes (.zip) og afleveres sammen med den oversatte pdf fil (.pdf) som 2 filer i Absalon.

God fornøjelse.