

Programmeringsteknik

Föreläsning 18

Dagens föreläsning

- P-redovisning
- Komplettering
- Vad händer om man inte hinner klart?
- Plussa
- Föreläsningsskärmar
- (bonusmaterial: livewires för spelprogrammering)

Tidsgränser

Sista dag för redovisning av P-uppgiften

- CL 25 april
- Datorlingvister 25 april
- S 10 maj

Granskning

- Innan du redovisar ditt program ska det granskas.
- Den som granskar ska fylla i granskningsprotokollet* och avgöra om programmet är användarvänligt, flexibelt, mm
- Granskaren ska vara med vid redovisningen.
- Alla ska granska *ett* program!
- * Se KTH Social, länk under "P-uppgiften"

Boka tid

- Du måste boka en redovisningstid, länk finns på KTH Social (under "P-uppgiften").
- Det finns plats för två namn - ditt och din granskare.
- Boka varsin redovisningstid!

P-redovisning

- Assen kollar leg
- Granskaren får berätta
- Assen provkör
- Assen tittar på koden
- Assen frågar
- Assen fyller i protokollet
- Assen kollar att programmet inte är kopierat
- Du får betyg (E,D,C,B,A eller komplettering)

Bonus

- Du har samlat ihop bonuspoäng från
 - Labbarna 1-5 (max 20 bp)
 - Provet (max 4 bp)
 - Specen (max 8 bp)
 - Prototypen (max 8 bp)
- Om du har 25 bp eller mer höjs ditt betyg ett steg, från E->D osv
- Men om du får F höjs inte betyget till E

Komplettering

- Om ditt program inte uppfyller kraven kan du få komplettera:
 - Antingen senare samma dag (om assen har tid)
 - Annars: boka ny tid
- Granskaren behöver inte vara med vid kompletteringen (om inte granskaren också ska komplettera sin granskning).

Inte klar i tid?

- Efter kursens slut anordnas uppsamlingstillfällen (som omtentor)
- Info kommer på KTH Social
- Om du inte blir godkänd på P-uppgiften i tid (med minst E) missar du chansen att plussa.

Plussa

- Om du blir godkänd i tid med minst E ...
- ... så har du möjlighet att *plussa* för högre betyg på uppsamlingstillfällen fram t o m **30 januari 2013**
- Du behöver inte göra ny spec/prototyp/granskning när du plussar.

Mer programmering?

DN1214, 1240, 1241 Numeriska metoder
Metoder för att få en approximativ lösning till problem som inte går att lösa exakt.

Viktiga förkunskaper:

Matte + programmering (ni lär er Matlab)

CSAMH & MaEnM: åk 2

MaFy, MaKT, MaKe : åk 3

Mer programmering?

DD1325 Tillämpad datalogi och etik
Algoritmer, datastrukturer och abstraktion

MaKT: åk 2

Valfria fortsättningskurser

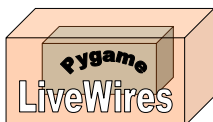
- Valfria datakurser att fortsätta med är t ex
 - DD1320 Tillämpad datalogi
 - DD2310 Javaprogrammering för Pythonprogrammerare
 - DD2385 Programutvecklingsteknik
 - DD2334 Databasteknik
- Lista över alla kurser finns här:
<http://www.kth.se/csc/utbildning/kurser>
- Välkomna!

OBS!

- CL och Datorlingvistik har ett moment kvar efter P-uppgiften

Spelprogrammering

- Pygame är en samling moduler för spelprogrammering .
- Se exempel i Dawsons bok: `moving_pan` (kap 11), `astrocrash` (kap 12), och på webben:
<http://www.pygame.org/>



- LiveWires är ett extralager som gör det enklare att använda Pygame.

Vad finns i LiveWires?

- Modulerna `games` och `color` (färgkonstanter)
- `games` innehåller klasserna
 - `Screen` (grafikfönstret)
 - `Sprite` (figurer som kan flytta sig i grafikfönstret)
 - `Text` (text i grafikfönstret)
 - `Message` (text som bara syns en stund)
 - `Animation` (en samling bilder som bildar en film)
 - `Mouse` (tar emot inmatning från musen)
 - `Keyboard` (tar emot tangenttryckningar)
 - `Music` (laddar och spelar musikfiler)

moving_pan.py

Ett exempel från Dawsons kap 11, där

1. Grafikfönstret öppnas
2. En bakgrundsbild laddas in
3. Figuren "Pan" (en Sprite) skapas
4. Inmatning från musen tas emot för att styra Pan

```
# Moving Pan
# Demonstrates mouse input

from livewires import games

games.init(screen_width=640,
           screen_height=480,
           fps=50)

class Pan(games.Sprite):
    """ A pan. Controlled by the mouse. """

    def update(self):
        """ Move to mouse coordinates. """
        self.x = games.mouse.x
        self.y = games.mouse.y
```

Annotations in the image:

- A line points from `screen_height=480` to the text "uppdateringshastighet".
- A line points from `games.Sprite` to the text "ärver attribut och metoder".
- A line points from `games.mouse.x` and `games.mouse.y` to the text "ger muspekarens koordinater".

```
def main():
    wall_image=games.load_image("wall.jpg",
                                transparent = False)
    games.screen.background = wall_image

    pan_image = games.load_image("pan.bmp")

    the_pan = Pan(image = pan_image,
                  x = games.mouse.x,
                  y = games.mouse.y)

    games.screen.add(the_pan)
    games.mouse.is_visible = False
    games.screen.event_grab = True
    games.screen.mainloop()

main()
```

*muspekaren håller sig
innanför grafikefönstret*

Animationer

- En Animation är en serie bilder som kan spelas upp som en film.
- Explosionen i explosion.py i Dawson kap 12 är ett exempel.
- Först skapar man en lista med bildfiler.
- Sen ett Animation-objekt, med bildlistan, position och visningsparametrar.

Bildlistan

```
explosion_files = ["explosion1.bmp",
                  "explosion2.bmp",
                  "explosion3.bmp",
                  "explosion4.bmp",
                  "explosion5.bmp",
                  "explosion6.bmp",
                  "explosion7.bmp",
                  "explosion8.bmp",
                  "explosion9.bmp"]
```

Animation-objekt

```
explosion = games.Animation(
    bildlistan = images = explosion_files,
    position = (x = games.screen.width/2,
               y = games.screen.height/2),
    antal = n_repeats = 0,
    filmvisningar = repeat_interval = 5)

tid mellan successiva bilder
```

Ljud

- Ljudfiler måste vara på formatet *.wav
- Ljudet laddas med metoden `load_sound`
- och styrs med metoderna `play` och `stop`
- Exempel:


```
missile_sound =
    games.load_sound("missile.wav")
    missile_sound.play(3)
```
- Man kan ha upp till åtta ljud igång samtidigt!

Musik

- Musik fungerar ungefär som ljud, men
- Det finns bara en musikkanal (motsvaras av objektet `games.music`)
- Man kan ladda musikfiler på flera olika format, t ex MP3 och MIDI
- Exempel:


```
games.music.load("theme.midi")
games.music.play(-1) #-1=evigt
```
- `sound_and_music.py` demonstrerar!