

DD1311 Programmeringsteknik med PBL

Föreläsning 18

Nyheter

- Dagens grupptimme flyttad - assen meddelar ny tid och plats
- Bokningslistor finns på kursens webbsida (nya tider läggs upp varje vecka)
- Fyll i kursenkäten på webben!

Dagens föreläsning

- P-redovisning
- Komplettering
- Vad händer om man inte hinner klart?
- Föreläsningsskurer
- Kursenkät
- Livewires/Pygame

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 måste granska ett program!

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)

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?

- Sista redovisningstillfället är i början av maj (se schemat)
- Efter det anordnas uppsamlingstillfällen, se <http://www.csc.kth.se/utbildning/kth/kurser/DD1310/uppsamling/>
- Men det är bra att bli klar i tid - då har du möjlighet att *plussa* för högre betyg (fram t o m **30 januari 2012**)

Fortsättningskurser

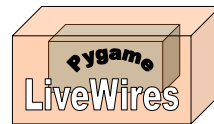
- Nästa kurs från oss är DD1214/1240 Numeriska metoder i höst
- Valfria datakurser att fortsätta med är t ex
 - DD1320 Tillämpad datalogi
 - DD2310 Javaprogrammering för Pythonprogrammerare
 - DD2385 Programutvecklingsteknik
 - DD2334 Databasteknik
- Lista över alla kurser finns här: <http://www.kth.se/csc/utbildning/kurser>
- Välkomna!

Kursenkät

- Kursenkäten finns på kursens webbsida, under rubriken "Kursanalys"
- Berätta hur vi kan göra kursen bättre till nästa år!

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
```

uppdateringshastighet

ärver attribut och metoder

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 grafikfönstret

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(
    images = explosion_files,
    x = games.screen.width/2,
    y = games.screen.height/2,
    n_repeats = 0,
    repeat_interval = 5)
```

bildlistan

position

antal filmvisningar

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!