Processing.org workshops
Workshop 4

Open Space Aarhus

Bryggervej 30, 8240 Århus N

8. december 2011

Dagens program

- Introduktion
- Resume af sidste gang
- Kode
  - Klasser
- Grafik
  - Billeder
  - Fonte (Tekst)
  - Snevejr
  - Pixel-pushing - måske
- *Afslutning*

OPEN SPACE AARHUS

Introduktion

Slides og processing filer

`http://poodle/processing`

Slides kan sikkert bruges til at kigge i eller kopiere fra.

OPEN SPACE AARHUS

OpenProcessing.org

Hvad har de flittige lavet

- `http://www.openprocessing.org/classrooms/?classroomID=1075`

OPEN SPACE AARHUS

Resume af sidste gang

- Kode
  - Arrays:
    - `float[] boldX = new float[10];`
    - `boldX[5] = 200;`
    - `ellipse(boldX[5], boldY[5], 5, 5);`
  - Løkker:
    - `while (betingelse) { ... }`
    - `for (start; betingelse; opdatering;) { ... }`
- Grafik
  - Mange bolde
  - Fyrværkeri
- Spørgsmål?

OPEN SPACE AARHUS

Det er jo jul!

- Et animeret julekort
- Snevejr for at fortsætte partikel temaet
- Brug klasser til at strukturere koden

OPEN SPACE AARHUS

Snefnug

```
float posX = 200;
float posY = 200;
float velX = 0;
float velY = 1;
```

```
posX += velX;
posY += velY;

if (posY > height) {
  posY = 0;
}
```

OPEN SPACE AARHUS

Funktioner - fra sidste gang

```
returtype navn(parametre) {
 // implementation
}
```

Eksempler

```
void draw() {
}

void drawTarget(float x, float y) {
  fill(255, 0, 0);
  ellipse(x, y, 40, 40);
  fill(255, 255, 255);
  ellipse(x, y, 30, 30);
  fill(255, 0, 0);
  ellipse(x, y, 10, 10);
}

float vinkelTilMus(float x, float y) {
  return atan2(mouseY - y, mouseX - x);
}
```

Snefnug med funktioner

```
void update() {
  posX += velX;
  posY += velY;

  if (posY > height) {
    posY = 0;
  }
}

void paint() {
  ellipse(posX, posY, 10, 10);
}
```

```
void draw() {
  background(50, 100, 200);
  update();
  paint();
}
```

OPEN SPACE AARHUS

Klasser

- Brugerdefinerede datatyper
- Samler variable og de funktioner der benytter variablene i en logisk enhed
- Genbrugelig
- Masser af nyttige klasser på nettet
- Man behøver ikke forstå implementationen for at bruge dem!
- Abstraktion
- Objektorienteret programmering

OPEN SPACE AARHUS

Klasser og Objekter

Klasser

Abstrakt beskrivelse so alle objekter af typen overholder Definerer hvilke variable og funktioner der er tilgængelige.
En brugerdefineret datatype.

Objekter

Faktiske instanser af klassen. De "virkelige"objekter som man arbejder med.
En variabel af typen.

OPEN SPACE AARHUS

Din første klasse

```
class Fnug {
  float posX = 200;
  float posY = 200;
  float velX = 0;
  float velY = 1;

  void update() {
    posX += velX;
    posY += velY;
    if (posY > height) {
      posY = 0;
    }
  }

  void paint() {
    ellipse(posX, posY, 10, 10);
  }
}
```

```
// variabel af typen Fnug
// initialiseret til et nyt Fnug
Fnug fnug = new Fnug();

void draw() {
  background(50, 100, 200);
  fnug.update();
  fnug.paint();
}
```

OPEN SPACE AARHUS

Tabs

- Brug tabs til at organisere koden
- God ide: En tab per klasse
- Giv tab'en samme navn som klassen

OPEN SPACE AARHUS

Arrays - fra sidste gang

Et array er en opslagstabel. Man kan lave et array med et fast antal pladser. Derefter kan man skrive og læse værdier på de enkelte pladser i arrayet.

Syntax

- erklæring: type[] navn;
- initialisering: float[] boldX = new float[10];
- tildeling: boldX[5] = 100;
- læsning: ellipse(boldX[5], boldY[5], 30, 30);

Bemærk at det første element i et array har indeks 0. Det vil fx sige at et array med 10 elementer indekseres med tallene 0-9.

OPEN SPACE AARHUS

Arrays af objekter

- Erklæring: `Fnug[] snevejr;`
- Initialisering: `Fnug[] snevejr = new Fnug[10];`
- Tildeling: `snevejr[5] = new Fnug();` VIGTIGT!!!
- Metodekald: `snevejr[5].update();`
- Læsning: `Fnug fnug = snevejr[5];`
- Metodekald: `fnug.update();`

OPEN SPACE AARHUS

Skrifttyper

- Klassen PFont holder en skrifttype
- Brug Tools/Create Font
- Filen ender i en data folder
- Variabel der kan holde en font
  - `PFont font;`
- Indlæs font
  - `font = loadFont("Verdana-Bold-48.vlw");`
- Vælg font
  - `textFont(font, 100);`
- Tegn tekst
  - `text("God Jul", 100, 100);`

OPEN SPACE AARHUS

Billeder - Indlæs og tegn

- Klassen PImage repræsenterer et billede
- Variabel der kan holde et billede
  - `PImage img;`
- Indlæs billede
  - `img = loadImage("osaa.png");`
- Tegn billede
  - `image(img, x, y, width, height);`

OPEN SPACE AARHUS

Pixels - teori

- Direkte adgang til pixel data
- Een kontinuerlig blok i hukommelsen
- Skal selv udregne indeks fra x,y koordinater

OPEN SPACE AARHUS

Pixels - læs og skriv

- readPixels()
- pixels[x + y * width] = color(255);
- color c = pixels[x + y * width];
- updatePixels()

OPEN SPACE AARHUS

Snevejr

- Brug Fnug klassen til at simulere svævende fnug
- Brug `pixels` arrayet til at sætte fastlåste fnug
- Test for kollision med pixels
- Udfordring: Undgå tårne og pyramider

OPEN SPACE AARHUS

Libraries / Biblioteker

- Eksempler på indbyggede
  - PFont - skrifttyper
  - PImage - indlæs bitmap-baserede billeder
  - PVector - en matematisk vektor
  - OpenGL - 3D grafik og hardware acceleration
  - Minim - Lyd input og output
- Eksempler på udvidelser
  - controlP5 - GUI

OPEN SPACE AARHUS

Tak for denne gang

- Hvad syntes *du* om workshoppen?
- Hvad så nu?

Klasseværelset

`www.openprocessing.org/classrooms/?classroomID=1075`

OPEN SPACE AARHUS

Resume

- Variable
- Primitive datatyper
- Operatorer
- Forgreninger
- Løkker
- Funktioner
- Arrays
- Klasser

- Koordinatsystemet
- Farver
- Tegnefunktioner
- Input fra tastatur og mus
- Simpel partikel simulering
- Billeder
- Pixeldata

OPEN SPACE AARHUS

Hvad så nu

- På egen hånd
- Kontaktsport!
- Kom og spørg
- Næste år
  - Tema-kode-aftener

OPEN SPACE AARHUS