Processing.org workshops
Workshop 3

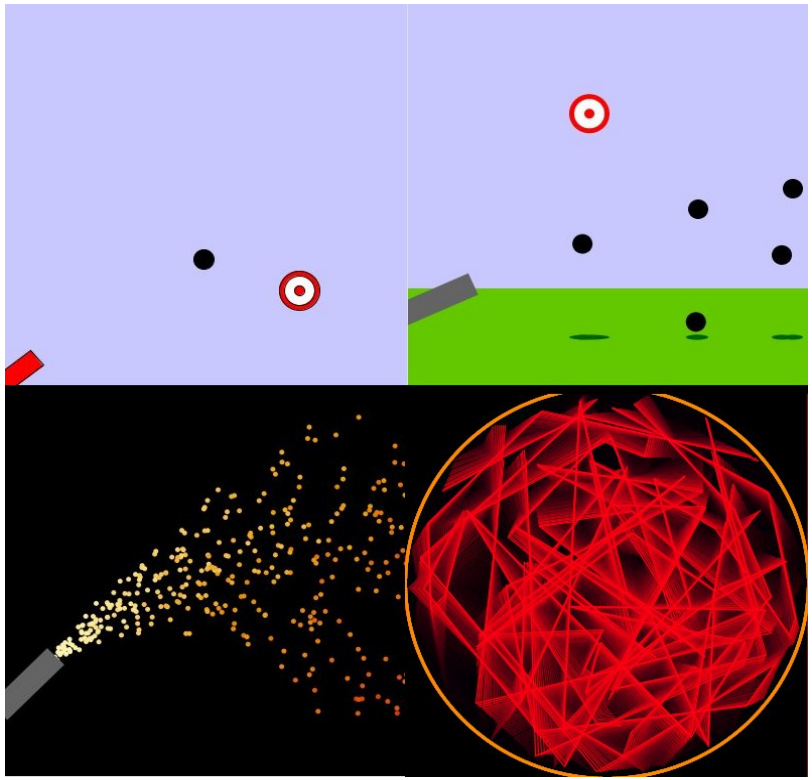
Open Space Aarhus

Bryggervej 30, 8240 Århus N

1. december 2011

Dagens program

- Introduktion
- Resume af sidste gang
- Kode
  - Arrays og løkker
  - Funktioner
  - Klasser
- Grafik
  - Mange bolde på en gang!
  - Fyrværkeri (fontæne)
  - Partikelsjov
- *Afslutning*



Figur: Workshop 3

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

- Variable: `float boldX = 200;`
- Operationer: `boldX = boldX + deltaX;`
- Løkker:
  - `while (betingelse) { ... }`
  - `for (start; betingelse; opdatering;) { ... }`
- Forgreninger:
  - `if (betingelse) { ... }`
  - `if (betingelse) { ... } else { ... }`
  - `if (`x `>` $width$`) { dx = -dx; }`
- Simpel fysik.
  - acceleration = summen af kræfter delt med partiklens masse
  - ny hastighed = gammel hastighed + acceleration
  - ny positioin = gammel position + hastighed
- Spørgsmål?

OPEN SPACE AARHUS

More, More, MOOOOOOOOOREEE!!!

Vi vil have flere bolde på skærmen på samme tid!

Erklæring

```
float bold1X;
float bold1Y;
float delta1X;
float delta1Y;

float bold2X;
float bold2Y;
float delta2X;
float delta2Y;
```

Opdatering

```
bold1X += delta1X;
bold1Y += delta1Y;

bold2X += delta2X;
bold2Y += delta2Y;

if (bold1X > width) ...

if (bold2X > width) ...
```

Hvad så med 100 bolde? Der må være en nemmere måde...

SPACE AARHUS

Arrays

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 til mange bolde

Hvis vi laver variablene for bolden om til arrays, kan vi holde styr på tilstanden af mange bolde.

Boldenes tilstandsvariable som arrays

```
float[] boldX = new float[10];
float[] boldY = new float[10];

float[] deltaX = new float[10];
float[] deltaY = new float[10];
```

Nu kan vi bruge løkker til at opdatere og tegne boldene.

OPEN SPACE AARHUS

Sidespring: Demonstration af Løkker

Lav en løkke som tegner 10 bolde ved siden af hinanden. Brug x som tæller

med while-løkken

```
int x = 0;
while (x < 10) {
  ellipse(20 + x * 40, 200, 30, 30);
  x += 1;
}
```

med for-løkken

```
for (int x = 0; x < 10; x += 1) {
  ellipse(20 + x * 40, 200, 30, 30);
}
```

Opgave: Et gitter af figurer

Lav to løkker indeni hinanden, så rækken bliver gentaget under hinanden. Brug x og y som tæller.

Løkker i løkker

```
for (int y = 20; y < height; y += 40) {
  for (int x = 20; x < width; x += 40) {
    ellipse(x, y, 30, 30);
  }
}
```

Prøv at bruge x og y eller random() til at styre farve eller størrelse. Brug evt forgreninger til at tegne forskellige figurer.

OPEN SPACE AARHUS

Arrays og Løkker

Arrays og løkker er som skabt til hinanden. Vi kan bruge en for-løkke til at løbe igennem alle kuglerne og opdatere deres positioner.

```
int ANTAL = 10;

float[] kugleX = new float[ANTAL];
float[] kugleY = new float[ANTAL];
float[] deltaX = new float[ANTAL];
float[] deltaY = new float[ANTAL];

for (int i = 0; i < ANTAL; i++) {
  kugleX[i] += deltaX[i];
  kugleY[i] += deltaY[i];
}
```

OPEN SPACE AARHUS

Så skal der kodes!

Kanoneksemplet udvidet til at skyde med flere kanonkugler samtidigt.
http://www.openprocessing.org/visuals/?visualID=47076

OPEN SPACE AARHUS

Fra kanon til fontæne

Slagplan

- skyd automatisk
- mindre kugler
- flere kugler
- hold styr på kuglernes alder
- ændr kuglernes farve baseret på deres alder

http://www.openprocessing.org/visuals/?visualID=47191

OPEN SPACE AARHUS

Funktioner

Små genbrugelige stumper kode. Også nyttig til at gøre koden mere overskuelig.

Du har allerede brugt en masse funktioner fra processing. Du har også skrevet dine egne fx. setup() og draw(). Nu vil vi lave vore helt egne.

En funktion:

- kan tage input via parametre
- kan returnere et output
- kan have sideeffekter (f.eks. tegne på skærmen)

OPEN SPACE AARHUS

Funktioner

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

Eksempler

```
float doubleUp(float value) {
  return value * 2;
}

void draw() {
}

float sumOf(float a, float b) {
  return a + b;
}
```

Klasser

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

OPEN SPACE AARHUS

Tak for i dag

- Hvad syntes *du* om i dag?
- Næste gang: ?
- T^3 på tirsdag. I må meget gerne hjælpe med at rydde lokalet.

Klasseværelset

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

OPEN SPACE AARHUS