Contents

1 Sekventiel/procedural -programmering

Denne opgave handler om sekventiel og procedural programmering. Sekventiel programmering, betyder at instruktionernes rækkefølge ikke er ligegyldig. Procedural programmering betyder at vi kan opdele vores program i forskellige procedurer eller funktioner og bare kalde proceduren når vi har brug for den. Det ville være smart med en procedure som kan beregne en vares pris med moms.

Det vigtige i denne opgave er, at du skal lære at bruge dokumentationen til processing/Java.

Herudover vil du blive introduceret til rutediagrammer. Et værktøj som skal hjælpe dig med at strukturere dine programmer.

Der er altså hele tre ting som kræver din opmærksomhed.

1.1 Opgave 1

1.1.1

Du skal lave en kopi af min tegnig: opg1-hoejhat.pdf. Det primære fokus i denne opgave er, at bruge dokumentationen i processing. Der findes i processing en lang række forud definerede procedurer/funktioner. Vi kan se at det er en funktion fordi er altid er () bagefter. For eksempel size(); Size er en funktion som kræver to parameter, brede og højde. size(400,600); Men hvordan finder man ud af hvor mange parameter og hvilke man kan bruge til en funktion? Det gør man ved at kigge i dokumentationen til processing. Du finder alle funktioner i processings reference guide: processing.org/reference.

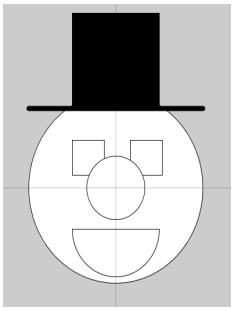

Figure 1: Højhat

Du kan bruge disse otte instruktioner for at kunne lave tegningen.

- size();
- line();

- strokeWeight();
- rect();
- square();
- circle();
- arc();
- fill();

Brug Processings dokumentation: processing.org/reference, for at finde ud af hvilke parameter de forskellige funktioner skal have.

- canvas (vindue som processing åbner) kan have størrelsen $400, 600$
- strokeWeight() er tykkelsen på stregen.
- fill() udfylder figuren med en RGB-farve.

Husk at koordinaterne $0, 0$ er øverste venstrehjørne (normalt vil det være nederste venstre) og er efter princippet: "hen ad vejen, ned til stegen". X,Y.

Brug rutediagrammet i figur 2 for at skrive koden:

1.1.2

Hvad sker der, hvis du bytter om på rækkefølgen? Altså hvis du starter med øjne,næse og mund og så tegner ansigtet.

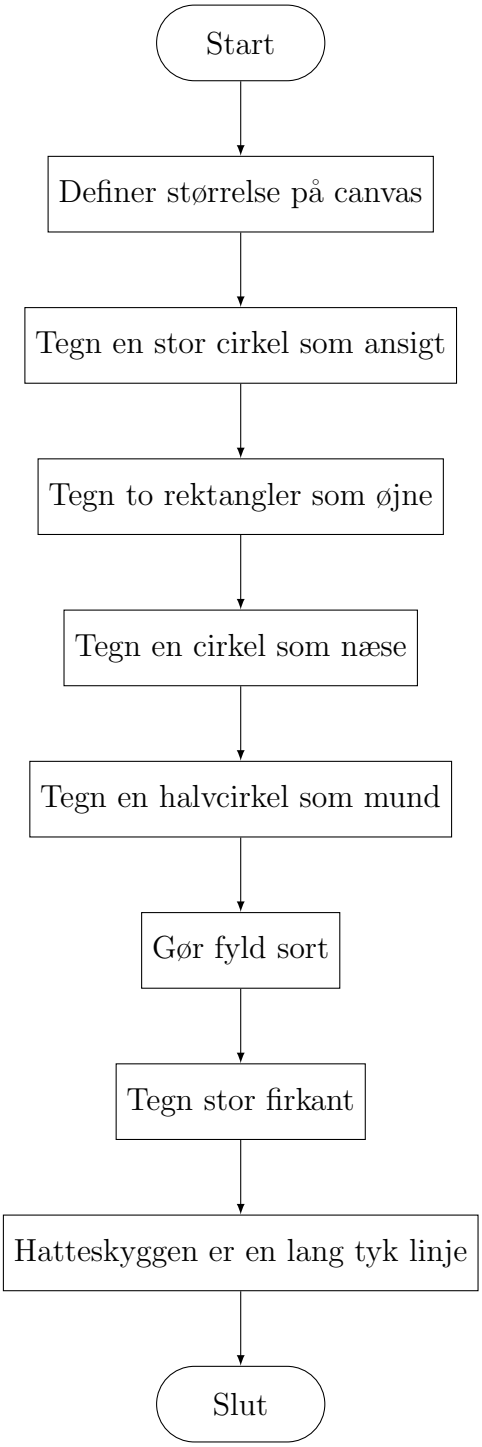

Figure 2: Rutediagram

1.2 Opgave 2

Du har nu en fornemmelse af hvad sekventiel programmering er. Men en af styrkerne i Java er, at det understøtter produceral programmering. Hver procedure programmeres sekventielt. I Proccesing skal der altid være to procedurer. Setup og draw. Setup bruger vi til f.eks. at sætte størrelse på canvas eller andre initieringer. Draw er hoveddelen. Proceduren (vi kalder det også en funktion) looper 60 gange i sekundet. Vi kan selv definere alle de procedurer som vi har brug for.

Tilføj nu følgende linjer til dit program og flyt alt dit kode ned i proceduren/funktionen hoved.

```
void setup() {
                    size(480, 120);
}
void draw() {
                    head();
}
void head(){
                    // Din kode skal stå her! I mellem de to tuborg paranteser.
}
```

1.3 Opgave 3

Del dit program yderligere op. Således at én funktion tegner hatten. En anden tegner øjne, en tegner munden og den sidste tegner ansigtet. Kald funktionerne for: void hat(){} ,void eyes(){} ,void mouth(){}, void face(){}, husk at fjerne funktionen head(), når du er færdig.

1.4 Opgave 4

Gør nu dit canvas så stort at der er plads til to hoveder ved siden af hinanden. Tilføj to parametere til dine funktioner, en float x og en float y koordinat udfra hvilke du kan tegne alle dine elementer af ansigtet. Ret dine linjer til så de tager udgangspunk i dine x og y koordinater.

2 Primitiv Animation

Det vigtige i denne opgave er, at du skal lære at bruge dokumentationen til processing/Java. Herudover skal du bruge rutediagram til at strukturere din kode. Tænk på animation som stop motion teknik.

Vi skal bruge om nogle nye instruktioner

- frameRate()
- frameCount()
- noLoop()
- rotate()
- translate()
- popMatrix()
- pullMatrix()

Brug Processings dokumentation: processing.org/reference, for at finde ud af hvad de forskellige funktioner gør og hvilke parameter de forskellige funktioner skal have.

I sidste kapitel introducerede jeg proceduralprogrammering. Husk at processing altid skal have to funktioner: void setup() void draw() Herefter kan man selv definere sine egne funktioner.

2.1 Opgave

Lav et program, hvor et hjul med minimum tre eger ruller over skærmen. Start med at lave et rute diagram.

2.2 Opgave

Lav et program som animerer en mand som kan gå. Start med at lave et rute diagram.

Du kan finde inspiration i mine to eksempler.