Danskernes besiddelse af elektronik

Af Tobias Grünbaum Hoffmann, 2.i
H.C. Ørsted Gymnasiet Lyngby
20. maj 2022 - 31. maj 2022

Studieretning:
Kom/it og programmering

Vejledere:
Morten Gundel
Anders Juul Refslund Petersen

Indledning

Os mennesker er interessant designet. Vi har en unik logisk venstre hjernehalvdel som gør det mulig for os at være detaljeorienteret og at forstå og beregne komplekse matematiske udfordringer[1]. Alligevel så vil visuelt design, der kæler for øjet, trænge langt bedre igennem hos mennesket. Vi bliver simpelthen bare bedre tiltrukket af noget der er visuelt appellerende og vi optager også den information og det budskab der gives, meget bedre. Under coronakrisen var “knæk-kurven-grafen” et klart eksempel på hvordan to kurver og en streg kunne give enormt meget information om hvad grundlaget var bag nedlukningen af verden[2]. En Infografik er et kraftigt værktøj som skal bruges med respekt, men samtidig bruges når det er muligt.

[1] (Biggart, 2020)
[2] (Christensen, 2020)

Nøglebegreber til objektorienteret programmering

Objektorienteret programmering åbner en masse nye muligheder for at effektivisere og overskueliggøre et program. I objektorienteret programmering gør man brug af klasser, objekter, metoder, konstruktører, parametre, returtyper, datatyper, nedarvning, superklasser og underklasser. Klassen er en slags kategori hvori opskrifter til objekter ligger i. En klasse indeholder også data for objekterne. For at give et eksempel så kunne en klasse være en person. De forskellige personer ville så være objekter der hver indeholder data så som alder, højde, vægt, navn, hudfarve, hårfarve osv. Objekterne har det til fælles at de indeholder de samme data blot med forskellige værdier.
Man kan bruge metoder til at ændre objekters data. Dette kunne for eksempel være, ændreVægt() hvor at metoder ændre allerede eksisterende data eller beregnBMI() hvor på at metoden bruger eksisterende data til at beregne ny data. Man kan også lave nye klasser der nedarver kode fra anden klasse. Det vil altså sige at man genbruger kode og udbygger klassen ved enten at lave nye metoder eller skrive over nogle gamle fra superklassen.
Når man stiller en klasse op, så starter man men at give klassen et navn. Dette kan være hvilket som helst navn. Derefter laver man de variabler der skal bruges i klassen. Variablerne kan også bruges uden for klassen hvis de er bundet op på et objekt, ved at skrive *"objektnavn.variablenavn"*. Til sidst kan man skrive de metoder man gerne vil have i klassen.

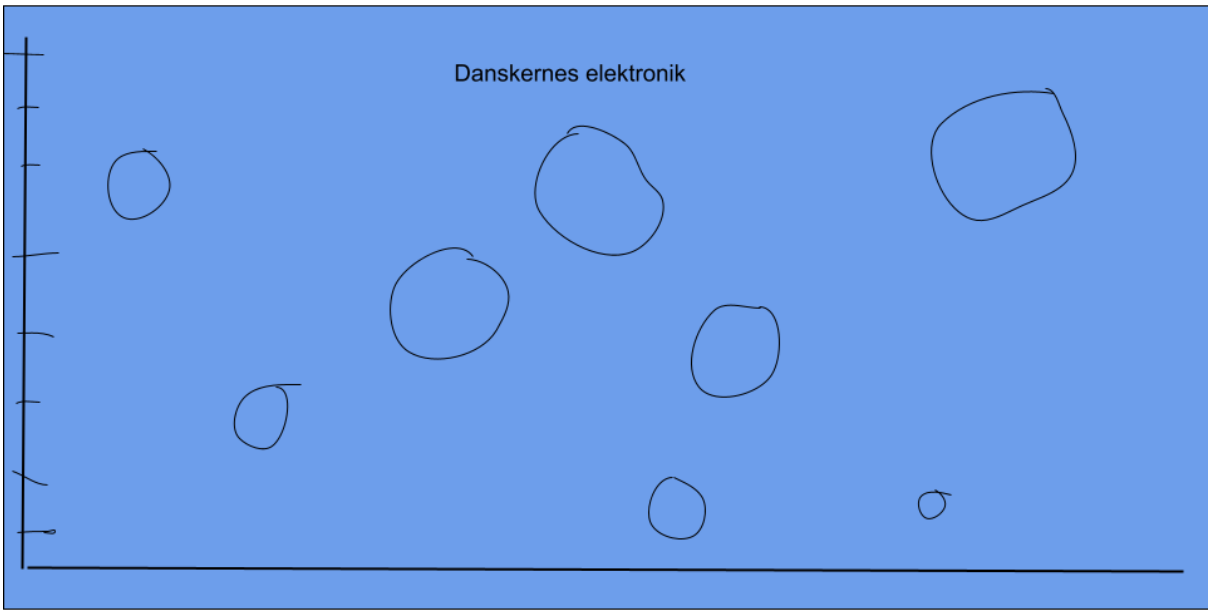
Hvilke kommunikationsmæssige overvejelser skal man skal gøre sig når man designer en infografik?

Når man producerer en infografik, skal man formidle noget data sådan at det er mere overskueligt og et budskab kan komme frem til modtageren. Rå data vil for de fleste være meget uhåndgribeligt og det er der hvor at infografikker er gode. Infografikker er ofte visuelt appellerende lige som et maleri. Men i modsætning til malerier så udløser infografikker dopamin hos modtageren når man skal på jagt efter information i stedet blot at stirre på et maleri[3]. En infografik er en slags opgave for modtageren hjerne og det kan den godt lide. Det er tilfredsstillende for hjernen når den gennemskuer et budskab på en infografik. Dette betyder altså at det er meget vigtigt at afsenderen sikrer at modtageren vil gennemskue budskabet da der ellers ikke vil være nogen dopamin belønning og infografikken vil fremstå som dårlig og svær at forstå.
For at kunne sikre netop det, skal en infografik have en klar og tydelig overskrift der gør modtageren klar på hvad de skal oplyses om[4]. Uden en klar overskrift vil modtageren blive forvirret og det er ikke sikkert at modtager vil få den ønsket information.

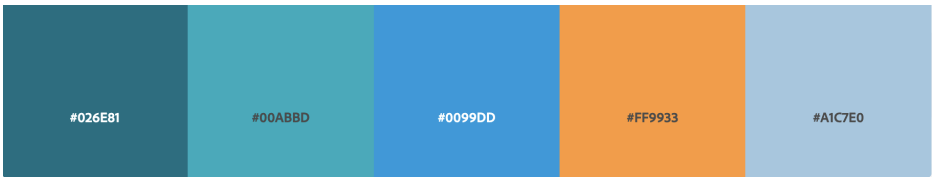
[3] (Emoticon, ll. 9-18)
[4] (Emoticon, ll. 27-38)

Hvordan har jeg designet min infografik?

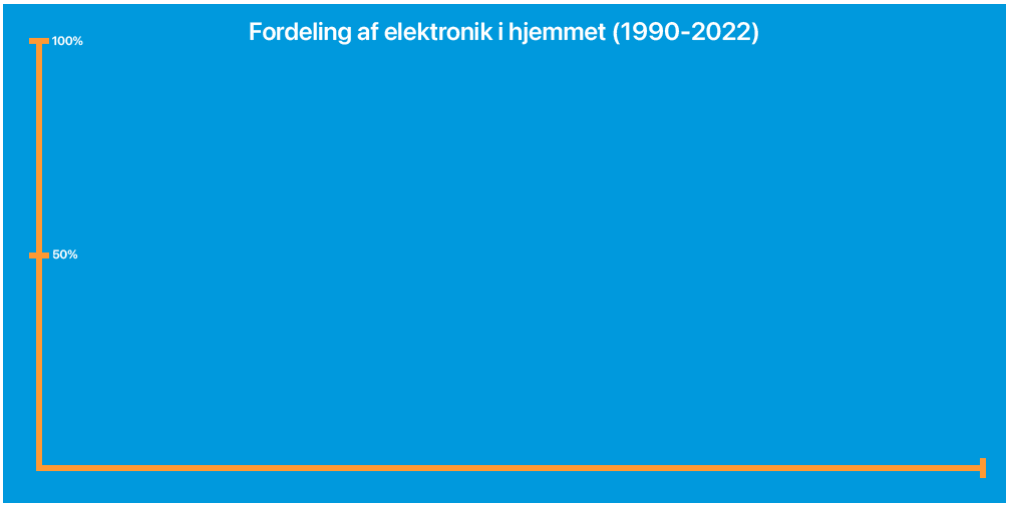
Jeg startede med at lave en skitse i Google Drawings.



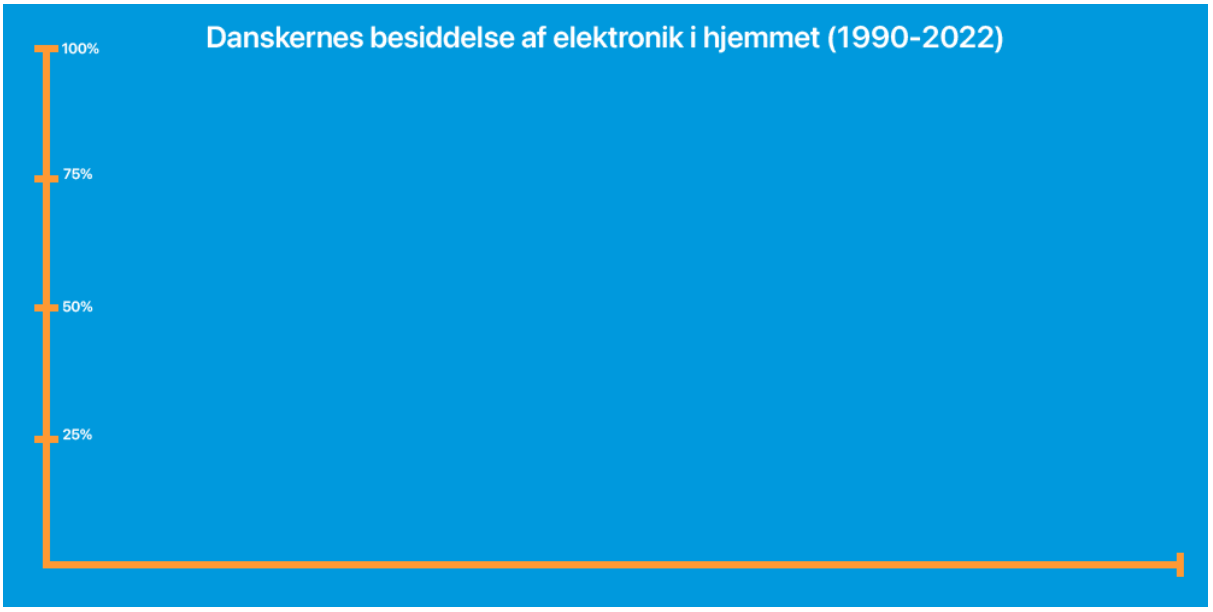
Derefter brugte jeg Adobe Color til at finde en farvepalette. Da jeg kiggede efter min farvepalette, havde jeg i bagtanke at det skulle være nogle farver der stak lidt ud for at fange øjet men samtidig ikke skære for meget ud sådan at det er ubehageligt for øjet at kigge på.



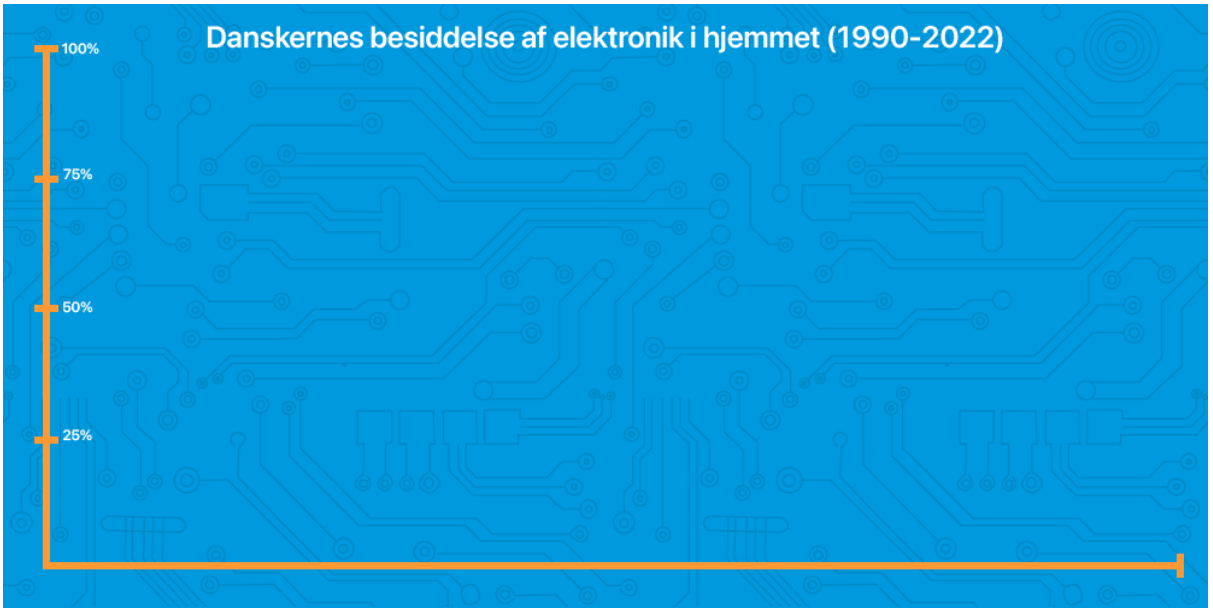
Jeg har brugt Adobe Illustrator til at designe min infografik. Jeg vidste at jeg skulle bruge en del plads til ikonerne og har derfor trukket teksten og akserne helt ud til kanterne.



Derefter fandt jeg ret hurtigt ud af at min overskrift var uklar i det at jeg viste den til mine venner og de næsten alle spurgte mig hvad den viste. Derudover kunne jeg også se at det var nødvendigt at have flere afmærkninger på infografikken så den blev lettere at aflæse.



Til sidst følte jeg at jeg manglede noget der tiltrak øjet. Derfor tilføjede jeg en baggrund der skulle forestille noget kredsløbs agtigt. Da jeg tilføjede baggrunden, var jeg meget opmærksom på at den ikke måtte være dominerende og jeg benyttede gestaltloven om figur og baggrund til at sikre mig at modtageren ikke er i tvivl om hvad der er forgrund og hvad der er baggrund.



Da jeg skulle vælge mine ikoner til min infografik, gik jeg meget op i at de skulle ligne hinanden i designet. De skulle være sort/hvid, todimensionelle, simple og nemme at forstå. Dette var vigtigt da en hel masse farve og forskellige vinkler gør det forvirrende for modtageren.

Hvordan har jeg programmeret min infografik?

I min kode har jeg valgt at visualisere hvor stor en andel af danskere der besidder nogle bestemte elektronik genstande over 32 år (fra 1990 til 2022). Jeg har brugt to klasser i min kode. En klasse der hedder Elektronik som indeholder to metoder og en anden klasse der hedder DataCircle som indeholder en metode.

Den første metode hedder loadData() og den indlæser den data som jeg har hentet fra Danmarks Statistik[5] fra et komma separeret csv fil med information omkring genstanden, årstallet og procentdelen af danskere der ejede netop denne genstand i det pågældende år. Derefter kommer den anden klasse der hedder DataCircle ind i billedet, da jeg nu for hver række i datasættet, ved hjælp af et for-loop, laver et nyt objekt i DataCircle klassen med det data der står på det pågældende år (Se bilag 1).

```
void loadData() {
  Table table = loadTable("elektronik.csv", "header");

  for (TableRow r : table.rows()) {
    String item            = r.getString("ITEM");
    int year               = r.getInt("TIME");
    float  percentage      = r.getFloat("PERCENTAGE");
    String listeKey = item + year;

    DataCircle d           = dataList.get(listeKey);
    if (d == null) {
      dataList.put(listeKey, new DataCircle(item, year, percentage));
    } else {
      return;
    }
  }
}
```

Bilag 1

I Elektronik Klassen er der endnu en metode, display(). Denne metode bliver hentet ind når koden starter i setup() og tres gange i sekundet i draw(). Denne metode starter med at lave baggrunden til min infografik som jeg har lavet i Adobe Illustrator og derefter skriver den årstallet samt benytter et for-loop til at køre en anden display metode i klassen DataCircle for hvert objekt. Denne anden display metode kontrollere hvilken genstand der er og derfor hvilket ikon den skal vise samt hvor på skærmen (se bilag 2).

```
void display(int selectedYear) {
  //Baggrundsbilledet bliver tegnet
  background(photo);

  //Årstallet udskrives
  fill(255);
  textSize(25);
  text(selectedYear, 850, 35);
  int x =50;
  translate(10, 10);
  for (String k : dataList.keySet()) {
    DataCircle d = dataList.get(k);
    if (d.year == selectedYear) {
      d.display(x);
      x +=50;
    }
  }
}
```

Bilag 2

[5] (*Elektronik I Hjemmet*)

I setup() starter jeg med at hente alle billederne ind i programmets hukommelse og tildele dem variabler. Derefter kører jeg loadData() for at hente data ind fra data filen og skabe objekterne. Så sikrer jeg at der ikke er noget på skærmen med clear() og derefter køres display() som starter på år 1990. Til sidst i setup kører jeg funktionen noStroke() som sikrer at der ikke er nogen outline på figurer (se bilag 3).

```
void setup() {
  loadBilleder();
  size(1000, 500);
  elektronik.loadData();
  clear();
  elektronik.display(selectedYear);
  noStroke();
}
```

Bilag 3

I display() rydder jeg skærmen og tegner display() med det valgte år, tres gange i sekunden for at sikre at når man trykker enten frem eller tilbage i tiden så sker det også på skærmen med det samme (se bilag 4).

```
void draw() {
  clear();
  elektronik.display(selectedYear);
}
```

Bilag 4

I keyPressed() ændre jeg på variablen selectedYear med et if-statement sådan at hvis man enten trykker “1” eller “2”. “1” går tilbage og “2” går frem i tiden. Jeg sikrer også at man ikke kan gå længere tilbage eller frem end der er data på (se bilag 5).

```
void keyPressed() {
  if (key == '1' && selectedYear > 1990) {
    selectedYear = selectedYear - 1;
  } else if (key == '2' && selectedYear < 2022) {
    selectedYear = selectedYear + 1;
  }
}
```

Bilag 5

Hvorfor har jeg valgt mit datasæt og hvad er målgruppe til min infografik?

Mit datasæt bygger på data fra Danmarks Statestik fra 1990 til 2022. Dataen er delt op i 16 forskellige elektroniske apperater og viser procentdelen af danskere som besidder det. Jeg synes at det er interessant at kigge på fordi lige netop den tidsperiode fra 1990 til 2022 er der skete helt utrolig meget udvikling af teknologier og software. Samtidig er elektronik blevet mindre og mere tilgængeligt. En iPhone er omtrent 120 millioner gange hurtigere end computeren der sendte Apollo 13 til månen[6]. Jeg forestille mig at min interaktive infografik kan bruges som et supplement til en digital artikel i et videnskabs blad, for eksempel illustreret videnskab eller lignende. Derfor er mine målgruppe altså læsere af videnskabs blade.

Hvad taler for og imod at brug objektorienteret programmering til at lave infografikker?

Objektorienteret programmering er generelt en stor fordel når man skal programmere noget som en infografik. Da data til en infografik typisk kommer i form af et stort sorteret datasæt, vil man forholdsvis nemt kunne indlæse, opdele og tildele data til variabler og objekter. Fejlfinding er også noget der er lettere i objektorienteret programmering da man typisk har mindre kode og koden er mere overskuelig. For at tage et eksempel i mit eget program så hvis det er en fejl i den måde objekterne er lavet på, er der kun én fejl og ikke seksten fejl i de seksten objekter. Alt efter hvor god man er til at give sine klasser, objekter og metoder gode navne bliver koden også mere logisk hvis man skal kigge på det igen efter lang tid. Hvis man er god til objektorienteret programmering, vil det også være en programmeringsmetode der er langt hurtigere end bare at “hardcode” det ind, da man kan få programmet til at gøre flere hundred ting på blot et par linjer. I min kode læser jeg blandt andet over 300 linjers data ind på blot otte linjers kode. Min mening er altså at det helt klart er en fordel at bruge objektorienteret programmering til at lave en infografik.

Til gengæld kan objektorienteret programmering også være en stor ulempe for nogen da det godt kan være meget uoverskueligt at lære da det er lidt abstrakt. Hvis man er ny til at kode skal man lige knække koden først og i mange tilfælde skal man ikke skrive så advanceret en kode at det kan give mening at bruge objektorienteret programmering.

[6] (Puiu, 2021)

Konklusion

Infografikker er en rigtig god måde at formidle data og budskaber til mennekser da hjernen godt kan lide at finde ud af hvad infografikken går ud på og hvad budskabet er. Dette er fordi at når hjernen regner ud hvad budskabet er, udløser det dopamin hvilket føles tilfredsstillende for modtageren. Jeg har brugt min viden om infografikker og objektorienteret programmering til at designe en interaktiv infografik der viser hvor mange danskere der besidder forskellige elektroniske apparater gennem tiden for at vise udvikling af elektronik og ændringen af danskerens besiddelse af elektronik. Jeg har taget forbehold for gestalt lovene og min viden for hvad der fanger øjet til at udvikle min interaktive infografik. For at gøre den interaktiv og indlæse data fra et datasæt har jeg brugt objekorienteret programmeret hvilket helt klart er en fordel da man kan gøre en kode langt mere overskuelig og mindre.

Link til Github kode:
https://github.com/TobiasHoffmann52/ElektronikBesiddelse.git

Link til video af infografik:
https://drive.google.com/file/d/1kiSQRsuXXTAS0_ma9GGRZH0WNhYIAMqj/view?usp=sharing

Kilder

Biggart, A. (2020, July 1). *Din hjerne. Bruger du højre eller venstre hjernehalvdel?* Alun.dk. Retrieved May 30, 2022, from https://www.alun.dk/livsstil/din-hjerne-hoejre-venstre.html

Christensen, M. S. (2020, May 14). Brug visualiseringer til de komplicerede budskaber. *Mandag Morgen*. https://www.mm.dk/artikel/visualiseringer-viser-vej-gennem-kriserne

Elektronik i hjemmet. (n.d.). Danmarks Statistik. Retrieved May 31, 2022, from https://www.dst.dk/da/Statistik/emner/oekonomi/forbrug/elektronik-i-hjemmet

Emoticon. (n.d.). *Infografik er de forklarende tegninger, som din hjerne bare elsker*. Emotion Infografik. Retrieved May 20, 2022, from https://www.emotion.dk/infografik-er-de-forklarende-tegninger-som-din-hjerne-bare-elsker/

Nordfalk, J. (n.d.). *5 Nedarvning*. Javabog. Retrieved May 20, 2022, from http://javabog.dk/OOP/kapitel5.jsp

Nordfalk, J. (n.d.). *4 Definition af klasser*. Javabog. Retrieved May 20, 2022, from http://javabog.dk/OOP/kapitel4.jsp

Nordfalk, J. (n.d.). *3 Objekter*. Javabog. Retrieved May 20, 2022, from http://javabog.dk/OOP/kapitel3.jsp

Puiu, T. (2021, May 13). *Smartphone is millions of times faster than NASA's 1960s computers*. ZME Science. Retrieved May 31, 2022, from https://www.zmescience.com/science/news-science/smartphone-power-compared-to-apollo-432/

Fuld kode:

ElektronikBesiddelse | Billeder | DataCircle | Elektronik

```
Elektronik elektronik = new Elektronik();
int selectedYear = 1990;
int stepYear = 0;
int m = 0;

void setup() {
  loadBilleder();
  size(1000, 500);
  elektronik.loadData();
  clear();
  elektronik.display(selectedYear);
  noStroke();
}

void draw() {
  clear();
  elektronik.display(selectedYear);
}

void keyPressed() {
  if (key == '1' && selectedYear > 1990) {
    selectedYear = selectedYear - 1;
  } else if (key == '2' && selectedYear < 2022) {
    selectedYear = selectedYear + 1;
  }
}
```

ElektronikBesiddelse | Billeder | DataCircle | Elektronik

```
PImage photo;
PImage camera;
PImage cdPlayer;
PImage computer;
PImage dishwasher;
PImage dryer;
PImage gpsNavigator;
PImage landline;
PImage microwave;
PImage mobilePhone;
PImage radio;
PImage smartphone;
PImage videoConsole;
PImage videoTape;
PImage videoCamera;
PImage washer;
PImage watch;

void loadBilleder() {
  photo        = loadImage("infografik.png");
  camera       = loadImage("camera.png");
  cdPlayer     = loadImage("cd-player.png");
  computer     = loadImage("computer.png");
  dishwasher   = loadImage("dishwasher.png");
  dryer        = loadImage("dryer.png");
  gpsNavigator = loadImage("gps-navigator.png");
  landline     = loadImage("landline.png");
  microwave    = loadImage("microwave.png");
  mobilePhone  = loadImage("mobile-phone.png");
  radio        = loadImage("radio.png");
  smartphone   = loadImage("smartphone.png");
  videoConsole = loadImage("video-console.png");
  videoTape    = loadImage("video-tape.png");
  videoCamera  = loadImage("video-camera.png");
  washer       = loadImage("washer.png");
  watch        = loadImage("watch.png");
}
```

ElektronikBesiddelse | Billeder | DataCircle | Elektronik

```
class DataCircle {

  int year;
  String item;
  float percentage;


  DataCircle(String c, int y, float p) {
    item  = c;
    year     = y;
    percentage = p;
  }

  void display( int x) {
    float scale = 1;
    fill(255);
    textSize(14);
    text(item + " " + percentage + "%", (x*scale)+(percentage*0.6), (420-(420*percentage/100)+20)+20);

    fill(2, 110, 129);
    if (item.equals("Toerretumbler")) {
      image(dryer, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("Vaskemaskine")) {
      image(washer, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("Opvaskemaskine")) {
      image(dishwasher, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("Mikroboelgeovn")) {
      image(microwave, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("Digitalt videokamera (-2021)")) {
      image(videoCamera, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("Digitalkamera (-2021)")) {
      image(camera, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("CD-afspiller ( -2017)")) {
      image(cdPlayer, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
```

```
    if (item.equals("Tv-harddiskoptager (-2021)")) {
      image(videoTape, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("PC")) {
      image(computer, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("Mobiltelefon")) {
      image(mobilePhone, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("Smartphone")) {
      image(smartphone, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("Fastnettelefon")) {
      image(landline, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("DAB radio")) {
      image(radio, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("GPS navigation")) {
      image(gpsNavigator, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("GPS-ur")) {
      image(watch, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
    if (item.equals("Spillekonsol")) {
      image(videoConsole, x*scale, (420-(420*percentage/100)+20), 0.6 * percentage, 0.6 * percentage);
    }
  }
}
```

ElektronikBesiddelse | Billeder | DataCircle | Elektronik

```
class Elektronik {

  HashMap<String, DataCircle> dataList  = new HashMap<String, DataCircle>();

  void loadData() {
    Table table = loadTable("elektronik.csv", "header");

    for (TableRow r : table.rows()) {
      String item            = r.getString("ITEM");
      int year               = r.getInt("TIME");
      float  percentage      = r.getFloat("PERCENTAGE");
      String listeKey = item + year;

      DataCircle d           = dataList.get(listeKey);
      if (d == null) {
        dataList.put(listeKey, new DataCircle(item, year, percentage));
      } else {
        return;
      }
    }
  }


  void display(int selectedYear) {
    //Baggrundsbilledet bliver tegnet
    background(photo);

    //Årstallet udskrives
    fill(255);
    textSize(25);
    text(selectedYear, 850, 35);
    int x =50;
    translate(10, 10);
    for (String k : dataList.keySet()) {
      DataCircle d = dataList.get(k);
      if (d.year == selectedYear) {
        d.display(x);
        x +=50;
      }
    }
  }
}
```