Gebt bitte die Bearbeitung als .pdf oder .doc Datei an Tobias Hammer (`hammer@informatik.uni-bremen.de`) ab. Bitte stellt die Abgaben zu allen Aufgaben in einer Datei zusammen, die wir drucken und korrigieren können. Der Dateiname soll die Nummer des Übungszettels und der Gruppe beinhalten, z.B. `ifguebung307.pdf` für Übung Nr.3 Gruppe 7. Gebt Namen und Email aller Gruppenmitglieder an. Gebt ausserdem die Zeit (Maximum aller Gruppenmitglieder) an, die Ihr für den Zettel benötigt habt. Diese Angabe ist nur als Rückmeldung für uns und geht nicht in die Bewertung ein.

Aufgabe 1 Schnipsel (5 Punkte)

Dieses mal geht es anders rum: Nicht ihr schreibt das Programm und wir müssen die Fehler finden sondern wir geben euch ein Programm und ihr findet (und korrigiert) die Fehler.

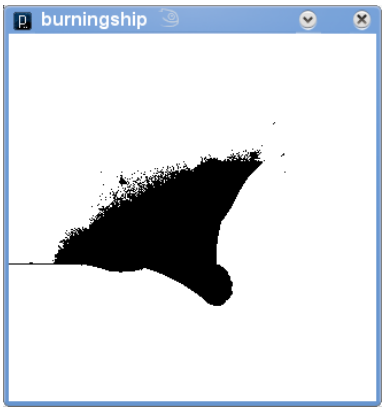
Das folgende kleine Programm besteht aus drei Bildschirmen. Einem Start- und Endebildschirm und einer kleinen Animation dazwischen. Der Start- und Endebildschirm soll nur einen kurzen Text in der Bildmitte anzeigen und auf Druck auf die Enter-Taste auf den nächsten Bildschirm weiterschalten. Die Animation soll 10 Balken anzeigen. Diese sind gleichmäßig in der horizontalen verteilt und bewegen sich von oben nach unten. Wenn sie die Unterkante des Fensters berühren, soll automatisch zum Endebildschirm gewechselt werden.

Nur leider war, als das Programm erstellt wurde der Tag lang, die Nacht kurz und die Konzentration schon lange weg. Deshalb haben sich 20 Fehler eingeschlichen. Findet sowohl die logischen Fehler (Programm verhält sich nicht wie beschrieben) wie auch die syntaktischen Fehler (das Programm läuft garnicht) und schreibt sie Stichpunktartig alle auf. Gebt zusätzlich eine korrekt funktionierende Version des Programms ab.

```
int state 0;
float blockHeight = 100;
float yBlockPosition = blockHeight/2,
floot blockSpeed = 50;

void Setup()
{
  size(400,400);
}

void draw()
{
  background(128,128,128);

  if state == 0
  {
    background(255);
    fill(255,0,0);
    text("Enter zum Starten", width/2, height/2);
  }
  els if ( state = 1 )
```

```
  {
    background(0);
    rectmode(center);
    stroke(255);
    fill(255,0,0);

    for ( int i = 0; i < 5 i = i-1 )
      rect(width/10*i + width/20, yBlockPosition, 10, blockHeight);

    yBlockPosition = yBlockPosition;

    if ( yBlockPosition - blockHeight/2 >= height )
      state == 3;
  }
  else if ( state == 2 )
    background(255);
    fill(255);
    textAlign(CENTER);
    text("Enter zum beenden", width, height);



void keyPressed()
{
  if ( state == 0 && key == ENTER )
    state = 1
  if ( state == 2 )
    exit();
}
```

Aufgabe 2 Das brennende Schiff (8 Punkte + 2 Kreativpunkte)



Interessante visuelle Gebilde, sogenannte Fraktale, lassen sich erzeugen, in man eine Rechenvorschrift wiederholt anwendet und beobachtet, ob das Ergebnis immer größer wird oder beschränkt bleibt. Die folgenden Gleichungen definieren so eine Vorschrift für das sogenannte Brennende-Schiff-Fraktal.

$$z'_x = z_x^2 + z_y^2 + c_x \quad (1)$$
$$z'_y = 2|z_x z_y| + c_y \quad (2)$$

Für jeden Wert von c_x und c_y startet man mit $z_x = z_y = 0$ und wiederholt (1) und (2), wobei man immer das Ergebnis z'_x, z'_y im nächsten Schritt als Ausgangswert z_x, z_y einsetzt. Strebt dabei $z_x^2 + z_y^2$ gegen unendlich, wird der Punkt c_x, c_y z.B. schwarz gefärbt, bleibt $z_x^2 + z_y^2$ beschränkt, z.B. weiß. Interessante Werte für c_x, c_y liegen im Bereich ± 2.

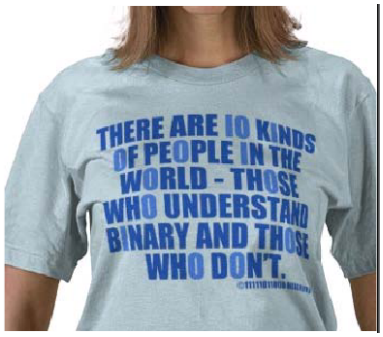
Entwickelt ein Programm in Processing, das das Brennende-Schiff-Fraktal zeichnet. Dazu müsst Ihr die abstrakte mathematische Formulierung “Strebt dabei $z_x^2 + z_y^2$ gegen unendlich” durch

einen konkreten Test ersetzen, der oberhalb eines gewissen Wertes für $z_x^2 + z_y^2$ annimmt, es strebt gegen unendlich und nach einer einer gewissen Anzahl Wiederholungen annimmt, es strebt nicht gegen unendlich.

Modifiziert nun die Darstellung, so dass sie animiert ist und in ihrer visuellen Gestaltung ansprechend. Führt dazu einen Parameter a ein, der mit der Zeit wächste und integriert ihn in den Formeln (1) und (2) irgendwie, so dass sich das dargestellte Bild interessant verändert. Überlegt Euch eine ansprechende Farbgestaltung. Von was könnte die Farbe kontinuierlich abhängen, wenn man nicht nur schwarz und weiß zeigen möchte?

Aufgabe 3 10 Kinds of People (5 Punkte)

Vergleicht in einem kurzen Aufsatz (ca. 1 Seite) die Praktikabilität vom Zehnersystem (0, 1, ..., 9, 10, 11, ..., 19, 20, 21, ..., 29, ..., 99, 100, 101, ..) mit dem Dualsystem (0, 1, 10, 11, 100, 101, 110, 111, ...) für den Bau von Rechenmaschinen. Anregungen: Welche Vorzüge hat das Dualsystem? Welche das Zehnersystem? Wie sieht das "kleine $1 + 1$", das "kleine $1 \cdot 1$" bei beiden aus? Warum verwenden wohl historische Rechenmaschinen meist das Dezimalsystem, moderne (ab ca. 1940) das Dualsystem?



Eine Quelle ist `http://de.wikipedia.org/wiki/Dualsystem`.

Aufgabe 4 Schleifen, die im Kreis zählen (2 Bonuspunkte)

Analysiert das untenstehende Programm! Kommt es irgendwann einmal zu einem Ende? Warum oder warum nicht? Wenn nicht, was müsste man ändern, damit es abbricht?

```
int counter1 = 0;
int counter2 = 1;
while (counter1+counter2>0) {
   counter1++
   if (counter1==1040) counter1=0;
   counter2++;
   if (counter2==65162) counter2=0;
}
```