

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. `ifguebung107.pdf` für Übung Nr.1 Gruppe 7. Für die erste Übung sind die Gruppennummern noch unbekannt, daher bitte 00 verwenden. Die Gruppen stehen dann auf dem ersten rückgegebenen Zettel. 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)

Die folgenden Aufgaben erfordern es, jeweils einige Zeilen in Processing zu schreiben. Fügt diese Zeilen bitte vollständig in das pdf ein, damit wir das Ergebnis nachprüfen können.

Aufgabe 1.1 Uhren malen

Malt eine Uhr, die ähnlich wie in Abbildung 1 aussieht. Die exakten Farbe und Maße müßt ihr nicht nachbilden, aber das Aussehen sollte ähnlich sein. Die Uhrzeiger sollen einmal auf 3 Uhr stehen (wie im Beispielbild) und einmal auf 5:30Uhr.

Aufgabe 1.2 Wo ist die Maus?

Gebt forlaufenden die aktuelle Mausposition im Meldungsfenster (Konsole) aus. Die Ausgabe soll ungefähr folgende Form haben: Maus Position: X-Wert ist 23, Y-wert ist 42

Aufgabe 1.3 In der Referenz

Sucht in der Processing Online-Referenz (<http://processing.org/reference/>) den Befehl `ellipseMode()` heraus. Schreibt kurz in eigenen Worten auf, welche Werte an die Funktion übergeben werden können und wie sich sich auf das Malen von Ellipsen auswirken.

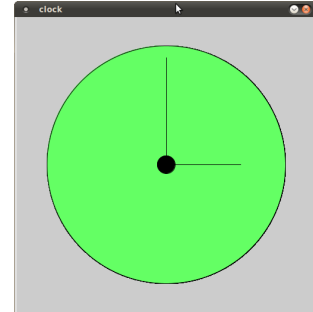


Abbildung 1: Beispiel für eine Uhr.

Aufgabe 1.4 Schnelle Kreise

Erstellt einen Processing-Sketch, der einen Kreis in die Mitte des Bildes malt. Der Kreis soll größer sein, je schneller ihr die Maus in X- oder Y-Richtung bewegt. Wenn man die Maus nicht bewegt soll nur ein Punkt zu sehen sein. Die Geschwindigkeit erhaltet ihr, indem ihr die aktuelle (`mouseX,mouseY`) und die vorherige (`pmouseX,pmouseY`) Mausposition betrachtet. Tipp: Die Funktion `abs(eineZahl)`, liefert den Betrag einer Zahl (z.B.: `abs(7)=7`, `abs(-7)=7`).

Aufgabe 2 Zoogs Cousin(-e)? (8 Punkte + 2 Kreativpunkte)

Denkt Euch eine Figur, ähnlich wie Zoog aus und programmiert ein Processing Sketch in dem die Figur auf den Mauscursor reagiert. Geht dabei schrittweise vor:

- Programmiert zuerst ein statisches Skript das die Figur zeichnet
- Macht das Skript dann technisch dynamisch indem Ihr `setup` und `draw` verwendet
- Integriert dann Interaktion mit der Maus, indem Ihr Parameter der Malfunktionen von der Mausposition abhängig macht

Entwickelt schrittweise, aber gebt nur das Endergebnis ab. Die Figur sollte von der Komplexität ungefähr Zoog entsprechen. Die 8 Punkte gibt es für die Realisierung als Programm, die fehlerfrei, verständlich und der Aufgabe angemessen sein soll. Die 2 Kreativpunkte vergeben wir abhängig davon wie pfiffig die Figur gestaltet ist, wobei pfiffig nicht kompliziert heißt.

Aufgabe 3 Ein Plan für Pacman (5 Punkte)

Pacman (Abb. 2) ist ein beliebtes Computerspiel aus der Anfangszeit des Computers (<http://www.google.com/pacman/>) das dieses Jahr seinen 30ten Geburtstag feiert. Die Spielfigur Pac-Man muss Punkte in einem Labyrinth fressen, während sie von Gespenstern verfolgt wird. Frisst man eine "Kraftpille", kann man für eine gewisse Zeit umgekehrt selbst die (nun blau eingefärbten) Gespenster verfolgen [Wikipedia].

Stellt Euch vor, Ihr wolltet dieses Spiel nachprogrammieren und zerlegt das Projekt in einzelne Entwicklungsschritte. Startet mit einer Zerlegung in 5-8 große Schritte und verfeinert diese zu je 3-5 Unterschritten. Versucht, aus dem Bauch heraus zu schätzen, wie lange die Entwicklung wohl dauern würde. Nehmt dazu an, Ihr hättet alle notwendigen technischen Fähigkeiten und einen Erfahrungsschatz, wie nach Absolvierung der Vorlesung.

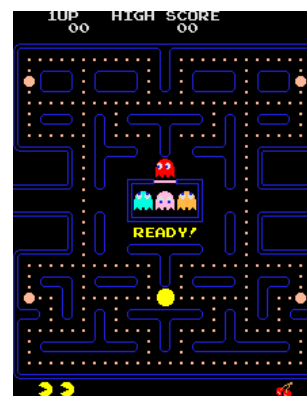


Abbildung 2: Screenshot aus dem Spiel Pacman (Quelle:Wikipedia).

Aufgabe 4 Bonusschnipsel (2 Bonuspunkte)

In der Bonus-Aufgabe sollt ihr einen Processing-Sketch mit Interaktion erstellen. Dargestellt ist eine Ziege, die an einen Pfahl gebunden ist und sich demnach immer nur in festem Abstand zu diesem Pfahl bewegen kann. Diese besondere Ziege, die ihr dort erstellen sollt hat Angst vor dem Mauszeiger und möchte immer möglichst weit von ihm weg sein.

Es reicht, die Ziege und den Kreis, in dem sie sich bewegen darf schematisch darzustellen (Abb. 3). Überlegt euch zuerst, wie ihr die Position beschreiben würdet, auf der sich die Ziege relativ zum Mauszeiger befindet. Die Abbildung sollte dabei hilfreich sein. Ihr werdet dabei den Abstand eines zwischen zwei Punkten berechnen müssen. Das geht wie schon in der Schule gelernt mittels Pythagoras. Wurzeln ziehen könnt ihr dafür mit der Funktion `sqrt(eineZahl)` und das Quadrat nehmen ($eineZahl^2$) mit `sq(eineZahl)`.

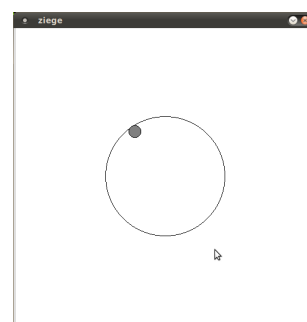


Abbildung 3: Beispiel für den Ziegen-Sketch.