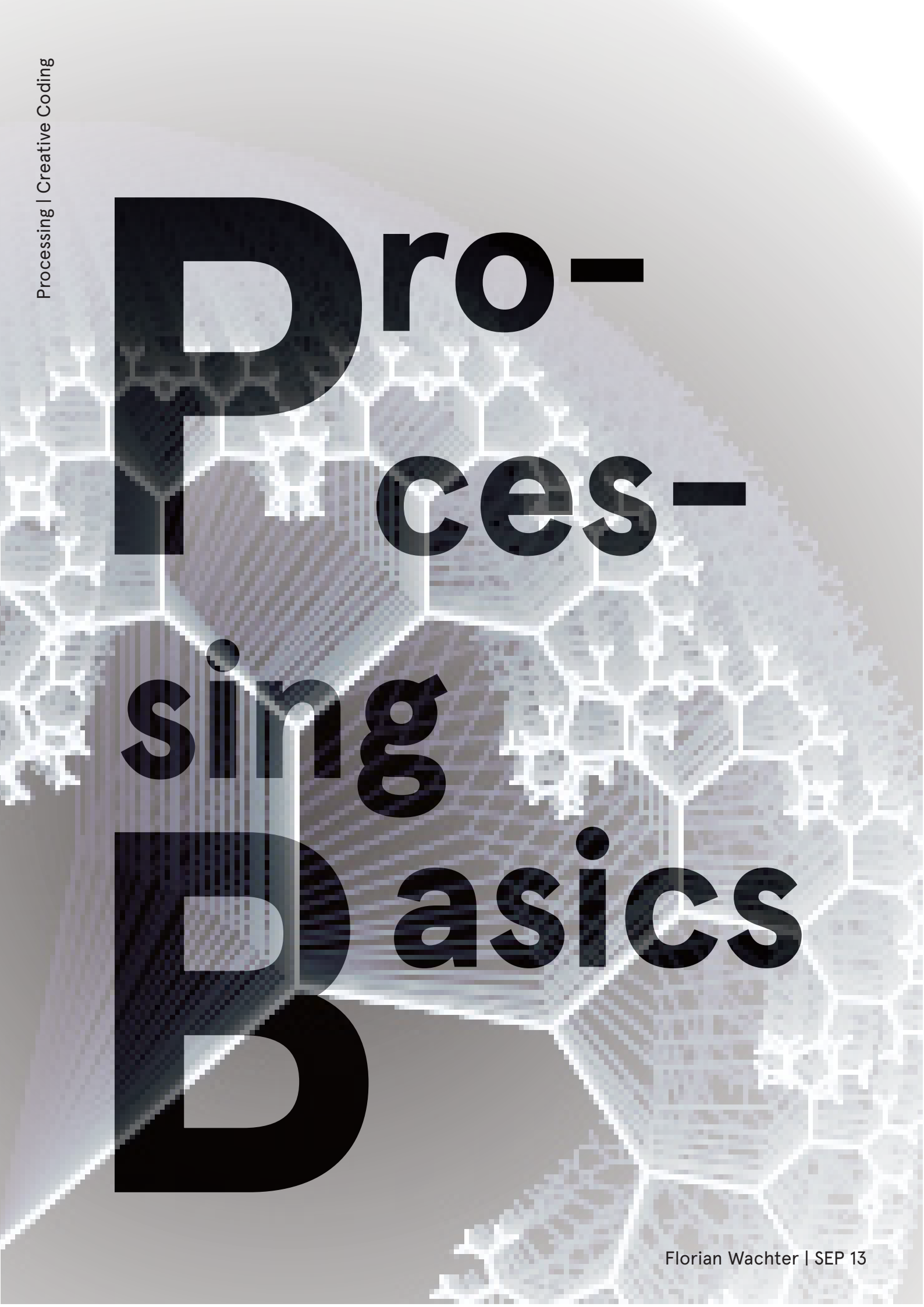
Processing | Creative Coding

# Pro-
# ces-
# sing
# Basics

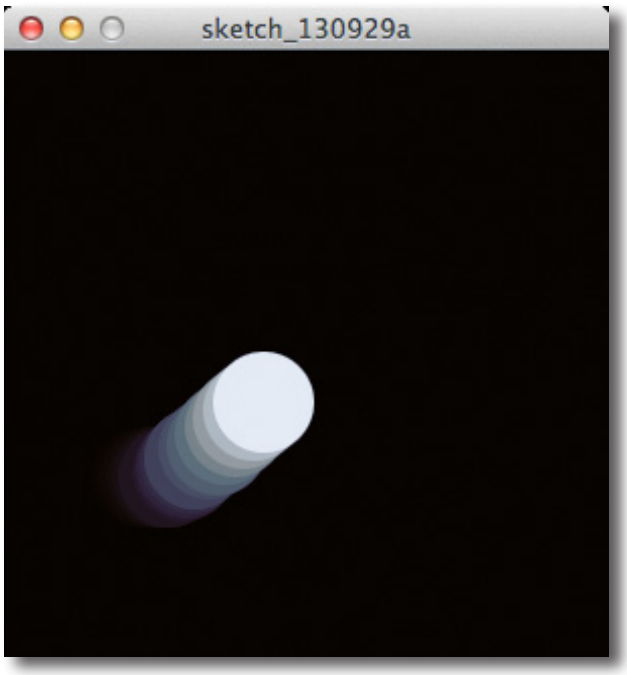


Florian Wachter | SEP 13

Inhalt:

Der Titel des Moduls lautete „Programming Basics". Es lässt erahnen um was es hier ging. Es ging um Processing eine auf die Einsatzbereiche Grafik, Simulation und Animation spezialisierte, objektorientierte, stark typisierte Programmiersprache mit zugehöriger integrierter Entwicklerumgebung.
Innerhalb zwei Wochen haben wir diverse Funktionen der Programmiersprache behandelt und einige Aufgaben gelöst, die uns das Programmieren näher bringen sollten.

# 1 Framework-Events und Funktionen

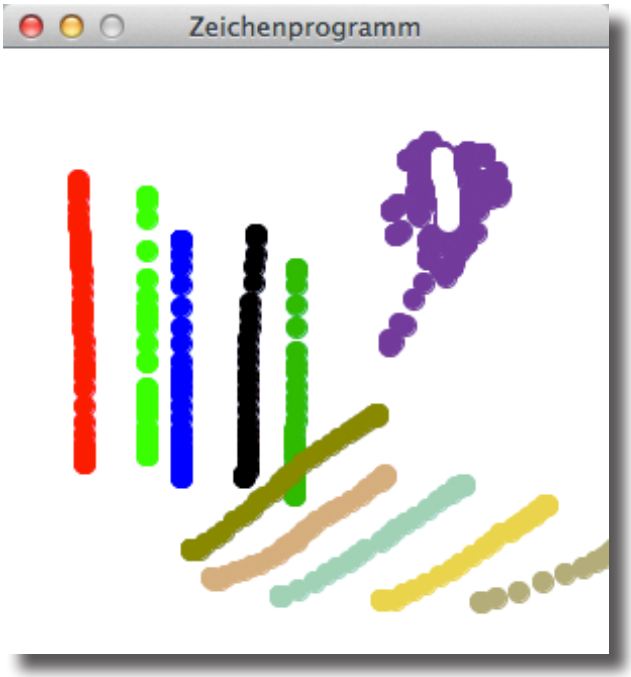



1-1 Erstelle ein Programm, in welchem ein Kreis dem Mauszeiger folgt.

Schwierigkeitsstufe: 1

Diese Aufgabe brachte einem die meist unverzichtbaren Funktionen setup() und draw() näher. Dabei lernte man auch die speziellen Eigenheiten dieser beiden Funktionen kennen. Die Funktion setup() hat die spezielle Eigenschaft, dass sie die erste Funktion ist, die bei einem Programmstart ausgeführt wird. Die Funktion draw() hingegen ist eine Funktion, die während einem Programmstart sich selbst 60 mal pro Sekunde ausführt, insofern keine andere Funktion wie frameRate() einfluss auf sie nimmt.

# 2 Bedingungen




2-1 Programmiere ein Zeichenprogramm. Mit den Tasten 1-5 kann die Farbe verändert werden.Mit der linken-Maustaste wird gezeichnet und mit der rechten-Maustaste wird radiert.

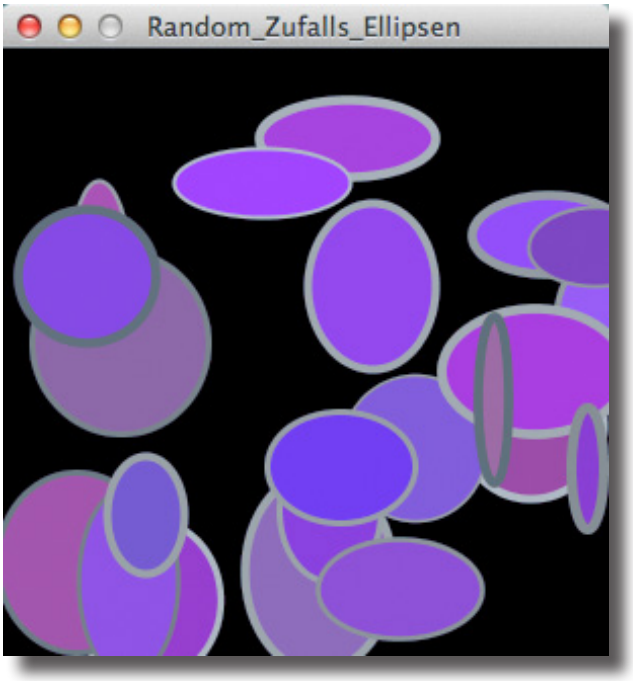
Schwierigkeitsstufe: 1

In dieser Aufgabe wurden einem Bedingungen in der Programmierung näher gebracht. Umgesetzt wurde dies mit If(...){} else{}. Die Schwierigkeit lag darin, zu erkennen, welche Bedingungen in einem Programmablauf abgefragt werden müssen, um zum richtigen Ergebnis zu gelangen.

# 3 Zufallszahlen

3-1 Erweitere das Beispiel so, dass die Füllfarbe der Ellipse abhängig von der Grösse der Ellipse ist(Durchmesser). Je grösser desto rötlicher soll die Ellipse sein.

Schwierigkeitsstufe: 1

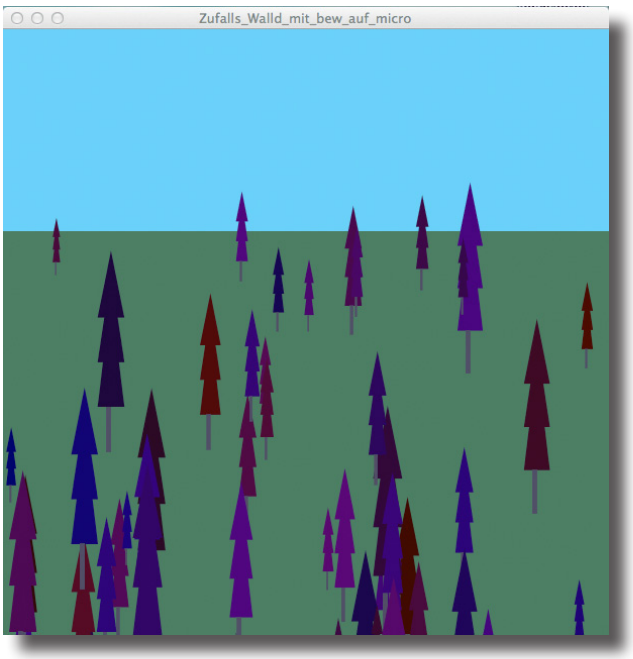



Frage war es zunächst wie man eine Zufallszahl generieren kann. Dabei kann die von Processing mitgelieferte Funktion random() helfen. Sie kann mit Hilfe zweier Zahlenwerten eine Gleitkommazahl generieren und gibt diese dann wieder zurück. Nun wurden Kenntnisse aus der zuvor erledigten Aufgabe erforderlich. Um auf den Durchmesser der Ellipsen zu schließen, musste man diese zunächst mit einer Bedingung prüfen. Anschließend konnte man, dann mit einem Randomwert der addiert wurde, die Ellipse einfärben.

# 4 Schleifen

4-1 Erstelle ein Programm welches einen Wald auf den Bildschirm zeichnet

Schwierigkeitsstufe: 2




Diese Aufgabe beschäftigte sich mit einigen Fragen:
Wie lasse ich den Baum zeichnen?
Wie bekomme ich den Baum an verschiedenen Positionen gezeichnet?
Wie kann ich Einfluss auf die Perspektive nehmen?
Wie vervielfältige ich den Baum?
Gelöst wurden diese Fragen alle mit Funktionen aus den zuvor gelösten Aufgaben. Hierbei wurden geometrische Formen, Bedingungen, Zufallszahlen und Schleifen verwendet. Schleifen werden für gewöhnlich mit for(){},while{},do{}while gebildet.

# 5 Koordinatensystem

5-1 Schreibe ein Programm, welches einen Smiley am Mauszeiger folgen lässt. Dazu soll ein kleineres Smiley um den grossen Smiley in einer Umlaufbahn kreisen.

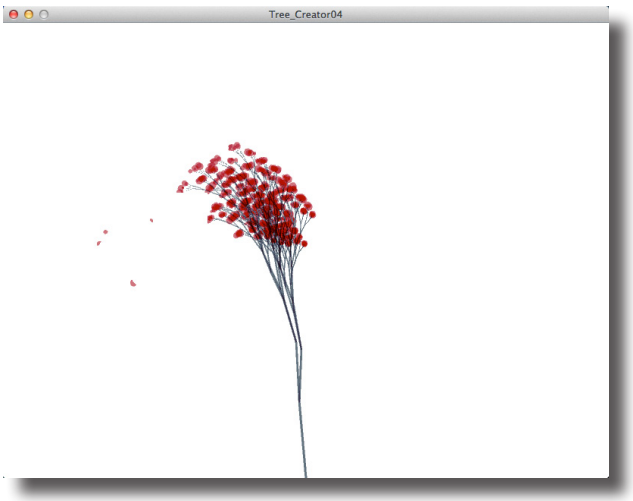
Schwierigkeitsstufe: 2


Diese Aufgabe brachte einem die Möglichkeiten näher, Grafiken im zweidimensionalen Raum zu bewegen und zu steuern. Dabei wurden Funktionen wie pushMatrix(), popMatrix(), translate(), scale() verwendet. Sie ermöglichen es, das Koordinatensystem zu fixieren, zu verschieben und zu skalieren ohne die Proportionalität der Objekte darin zu verändern. Dies ermöglicht einen kürzeren Quellcode und weniger Programmieraufwand.

# 6 Rekursive Funktionen / Fraktale

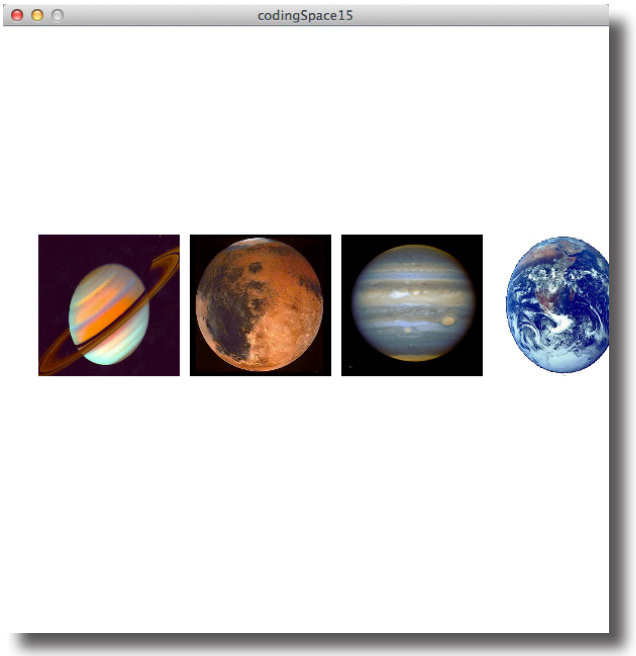
6-1 Verändere das Beispiel, mach andere Äste, Blüten, etc.

Schwierigkeitsstufe: 2




Die Aufgabe beinhaltete eine Funktion, die sich selber aufrufen kann. Das ermöglicht durch eine geschickte Programmierweise, dass Zeichnen von organischen Formen. In dieser Aufgabe wurde ein busch-baumähnliches Gebilde gezeichnet.
Die Herausforderung der Aufgabe lag darin, zu erkennen wie sich der Ablauf einer Funktion verändert, wenn sie sich bei ihrem Aufruf selbst wieder aufruft und was das Resultat davon sein kann (Endlosschleifen).
Meine Veränderung des Beispiels enthielt eine veränderte Aststruktur, die Fähigkeit sich im Wind zu bewegen und dabei nach und nach seine Blätter zu verlieren.

## 7 SVG + Bilder




7-1 Zeichne einen 4.Planeten (4.jpg) und lass diesen Planeten der Maus folgen. Verändere das Programm und die Daten in der Weise, dass der 4.Planet und nur der 4.Planet dargestellt wird (kein eckiges Bild, sondern nur der Planet).

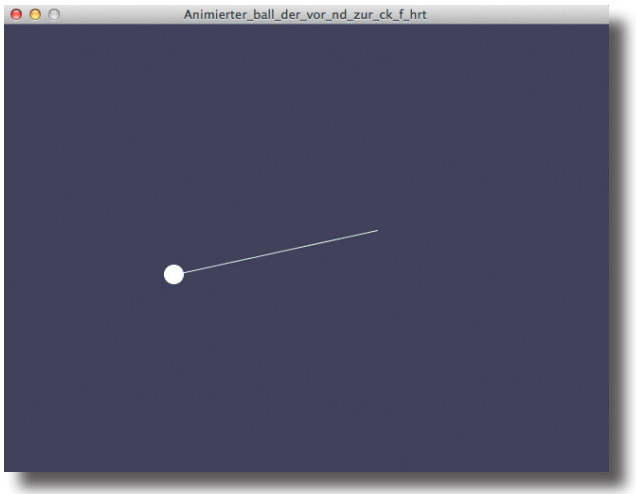
Schwierigkeitsstufe: 2

In dieser Aufgabe kamen auch noch externe Programme zum Einsatz. Hierbei musste man ein Bild in Photoshop bearbeiten (Ecken transparent machen) und einfügen. Interresant war hier die einfache Handhabung wie man ein Bild einfügt und der Umgang vergleichbar ist mit einer gezeichneten geometrischen Form.

## 8 Animation

8-1 Verändere das Beispiel nach folgenden Kriterien:
Der Pfad der Animation soll per Mausklick definiert werden.
Der Ball soll sich vorwärts und rückwarts bewegen

Schwierigkeitsstufe: 3

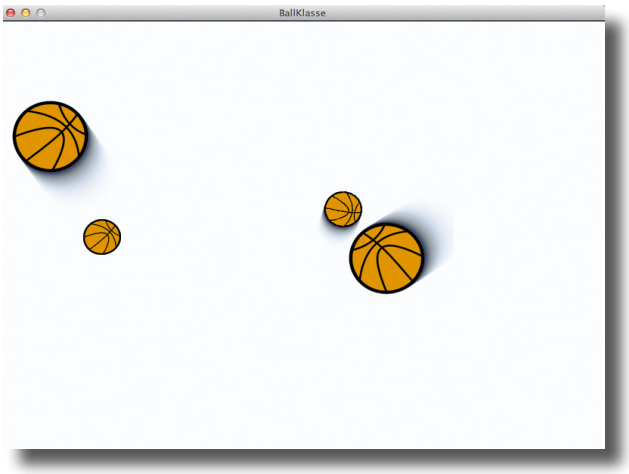



Die Aufgabe konfrontiert einen mit dem Begriff Vektoren. Was Anlass dazu gab, sich mit den Eigenschaften eines Vektors zu beschäftigen. Ein Vektor ist eine Instanz, die Größe und Richtung hat. Den Vektor gibt es sowohl im zweidimensionalen, als auch im dreidimensionalen Raum. Dabei entscheiden sich die Angaben der jeweiligen benötigten XYZ-Achsen.
Die Schwierigkeit dieser Aufgabe lag darin, die Stellen zu erkennen, an die man Bedingungen richten musste und wie man mit den Werten der Vektoren richtig umgeht. Interresant war dabei auch die Lösung des richtungswechselnden Balles. Man konnte dies mit einem einfachen Vorzeichenwechsel erreichen wie auch mit einer simplen Sinus-Funktion.

# 9 Klassen

9-1 Erweitere das Programm, dass mehrere Bälle unabhängig voneinander platziert werden können. Erweitere die Klasse um eine neue Art von Bällen.

Schwierigkeitsstufe: 3

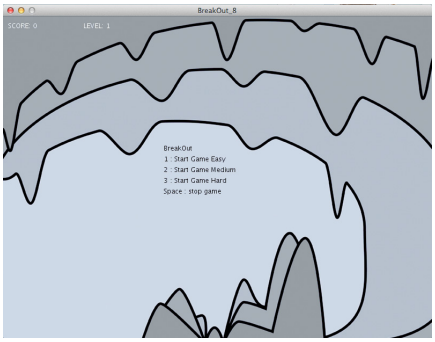



In dieser Aufgabe kamen Klassen zum Einsatz. Dabei handelt es sich um kleine Zusatzfunktionen die extern abgespeichert, im Hauptcode aufgerufen und verarbeitet werden. Mit Klassen kommt man an eine komplexere Programmierweise heran. Die Schwierigkeit in dieser Aufgabe war es, Klassen und Arrays richtig zu kombinieren um an das gewünschte Ziel zu kommen. Man spricht dann auch von Objekten, die in ihrer Funktion gleich sind aber unterschiedlich gesteuert werden.

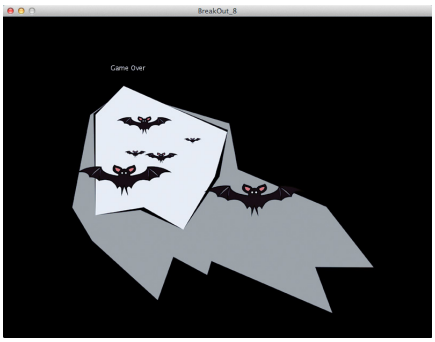
# 10 Endaufgabe: Breakout

10-1 Verändere das Game Breakout.

Schwierigkeitsstufe: 3








Diese Aufgabe stellte eine gewisse Herausforderung dar, da sie alles bisher Gelernte und noch mehr verband. Man konnte das Spiel anderst interpretieren und ihm so eine neue Spielbedeutung geben. Es war möglich und auch gefordert, Funktionen neu zu gestalten und die Grafik und Animationen umzubauen.
In meinem Breakout Game, habe ich die spiel Funktion so ausgebaut, dass es unendlich viele Levels gibt. Weiterhin habe ich das Game so interpretiert, dass ich daraus ein Vampirjäger Game gestaltete. In dem man Vampirfledermäuse jagen und abschießen musste.