PROGRAMMING BASICS

Processing Dokumentation von Simon Gwinner

//Kurs BDE-VIAD-V-1040 Programming Basics
bei Max Rheiner vom 17.09.2013 bis 27.09.2013

EINLEITUNG
GRUNDÜBUNG

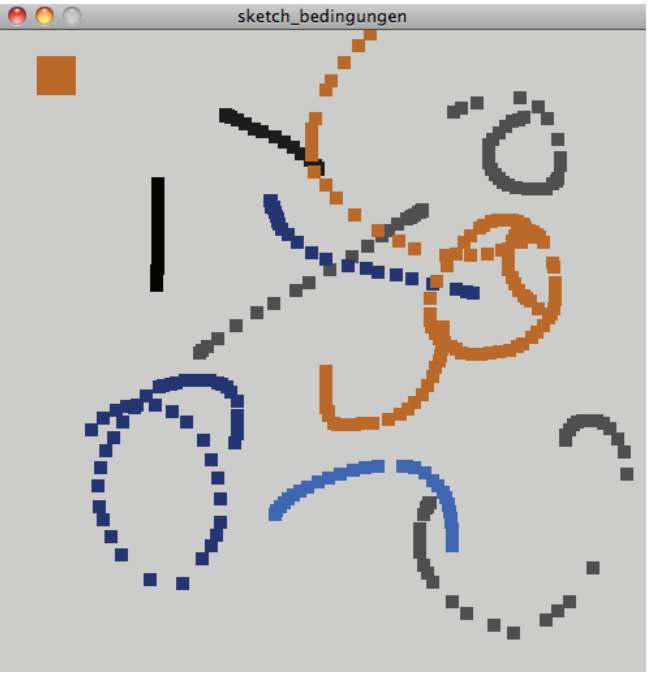
Die Grundübung hat mir gezeigt wie die Grundbefehle in Processing funktionieren. Mit size(); background(); bestimmt man die Grösse und Hintergrundfarbe des Fensters, zeichnen kann man mit Linien (line();), Kreisen (ellipse();), Rechtecken (rect();) oder mehrere, einzelne Punkte, welche zusammen eine Form bilden (vertex();). All diese Objekte werden mit Punkten bestimmt, als Beispiel rect(x-Achse, y-Achse, width, height);, die x und y Achse bestimmen die Position des Objekts und die width und height Angabe die Grösse des Objekts. Mit fill(); und stroke(); kann man dem Objekt eine Farbe zuordnen. Dafür nutzt man vier Werte, fill(r,g,b,t);, die ersten drei Werte sind aus dem RGB Farbschema und bestimmen die Farbe und das t die Transparenz dieser Farbe. Bei der Kontur funktioniert das genau gleich, um jedoch die Konturdicke zu verändern, benutzt man strokeWeight();, mit der gewünschten Pixeldicke der Kontur.

Allgemein Gilt, alles was das Programm braucht um zu starten, kommt in den void setup(); Bereich, alles was während dem Prozess gezeichnet werden soll in den void draw(); Bereich. Andere Ereignisse sind z.B. Tastatureingabe oder Maseingaben. Als Beispiel musste ich hier ein Programm generieren, bei welchem ein gezeichneter Kreis dem Mauszeiger folgt.

Ein ziemich wichtiger Bestandteil beim Programmieren sind Funktionen. Diese vereinfachen die darstellung und funktionieren wie Macroblöcke und können auch mehrmals aufgerufen werden.

BEDINGUNGEN ZEICHENPROGRAMM

Das Programmieren auch viel mit Mathematik zu tun hat, hab ich spätestens bei den Bedingungen gemerkt. Abfragen bei welchen man mit if(); und else(); oder bei mehreren Fällen mit switch(); und case x: bestimmen kann, ob ein gewisser Wert, gleich, ungleich, grösser/kleiner als, ... ist. So kann man als Beispiel bestimmen ob oder wann eine gewisse Aktion ausgelöst oder eben nicht ausgelöst werden soll.

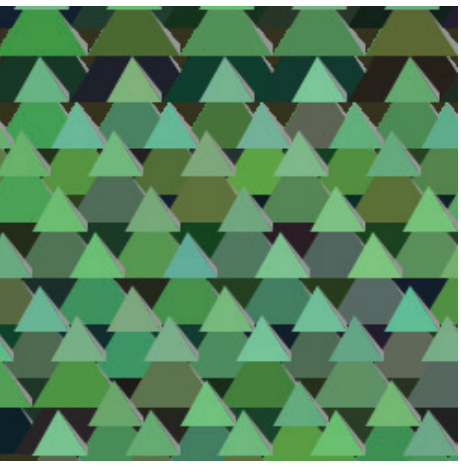


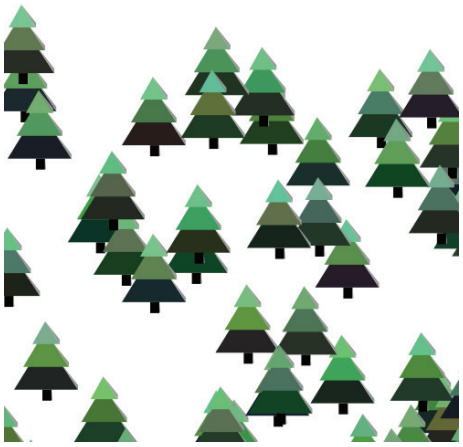
Screenshot Zeichenprogramm

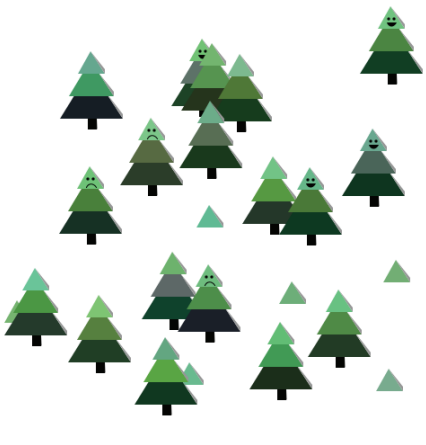
Auch hier konnte ich mit Hilfe einer Aufgabe, bei welcher wir ein Zeichenprogramm programmieren sollten, indem mit fünf verschiedenen Tastenkombinationen die Farbe geändert werden kann und mit der linken Maustaste gezeichnet und mit der rechten Maustaste radiert wird, diese Bedingungen schnell nachvollziehen und ausprobieren. Jedoch hatte ich immernoch ziemliche Probleme beim Verstehen und Aufbau des Codes.

SCHLEIFEN
WALD

Bei einer nächsten Übung haben wir erfahren, dass es nicht notwendig ist, wenn man z.B. mehrere Punkte nebeneinander will, diese draw Funktion jedes mal auszuführen, resp. im Code zu kopieren, sondern mit Hilfe einer Schleife automatisch mehrere mal auszuführen und zu zeichen.







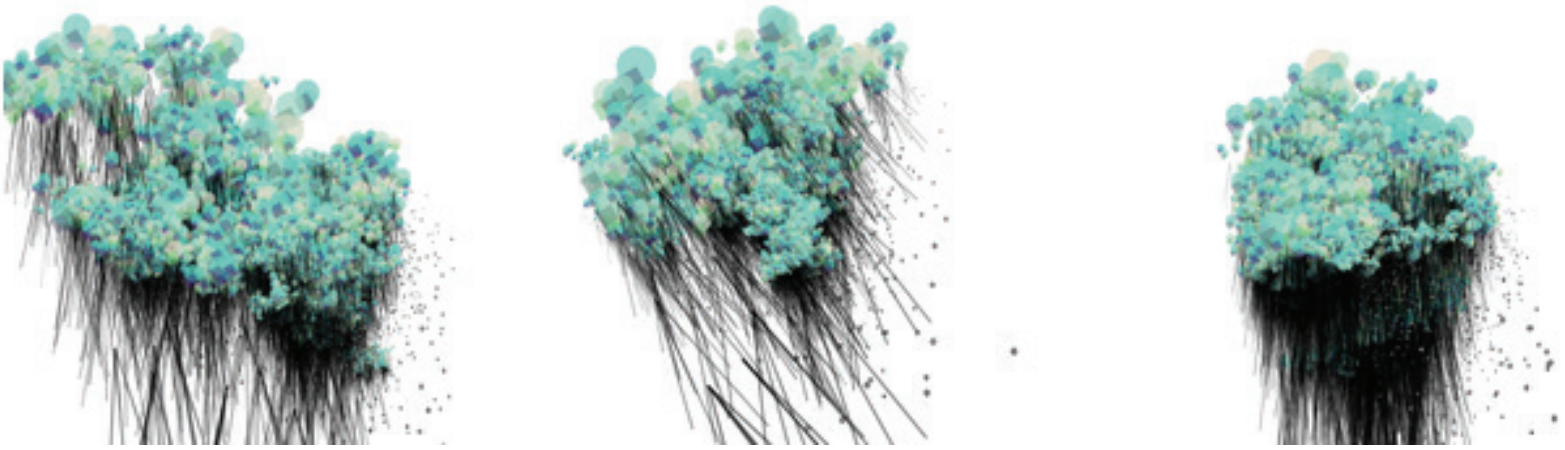
Screenshot der drei Programme und dem Prozess, welcher entstanden ist

Die Aufgabe bestand darin einen Wald zu generieren. Ich schrieb bei dieser Aufgabe drei Programme, als jeweilige Weiterführung des vorherigen Programmes. Das erste Programm zeichnete automatisch einen dichten Wald, in welchem ich den Abstand zum anderen Baum und zur nächsten Linie vorbestimmt habe. Das zweite Programm zeichnet einen Wald sobald man mit der Maus im Fenster klickt. Die Bäume werden nicht wie beim ersten Programm an einer bestimmten Position gesetzt, sonder Random irgendwo im Feld platziert. Das dritte Programm basiert auf dem Mausklick des zweiten Programmes, jedoch setzt man je einen Baum bei jedem Klick auf die Fläche wo geklickt wird. Auch kann man bei diesem Programm aus drei verschiedenen Bäumen (Neutral, mit Smile, oder traurigem Gesicht) auswählen und kann auch nur einen Busch zeichnen. Die Farben der Bäume werden nach einem Random-Farb-Bereich, welcher vorher eingeschränkt wurde generiert.

Spannend wäre es jetzt noch mit einer Funktion die Bäume zu fällen, jedoch war ich mit dieser Idee zu diesem Zeitpunkt noch überfordert.

REKURSIVE FUNKTIONEN
BAUM

Rekursive Funktionen sind Funktionen die sich selber wieder ausführen. Dies kann z.B. zum Zeichnen von Fraktalen verwendet werden. In einer Übung um diese Funktion zu vertiefen habe ich mich vom ursprünglich vorgegebenen Baum entfernt, indem ich die Äste oder die Wurzel vom Boden gelöst habe und mit der Position der MouseX und MouseY Achse verbunden habe. Die Striche (vorher Äste) folgen der Mausposition mit einer translate-Vorgabe.



Screenshot von drei Ergebnissen des Programmes, welche für mich Wolken oder Unterwassertieren ähneln

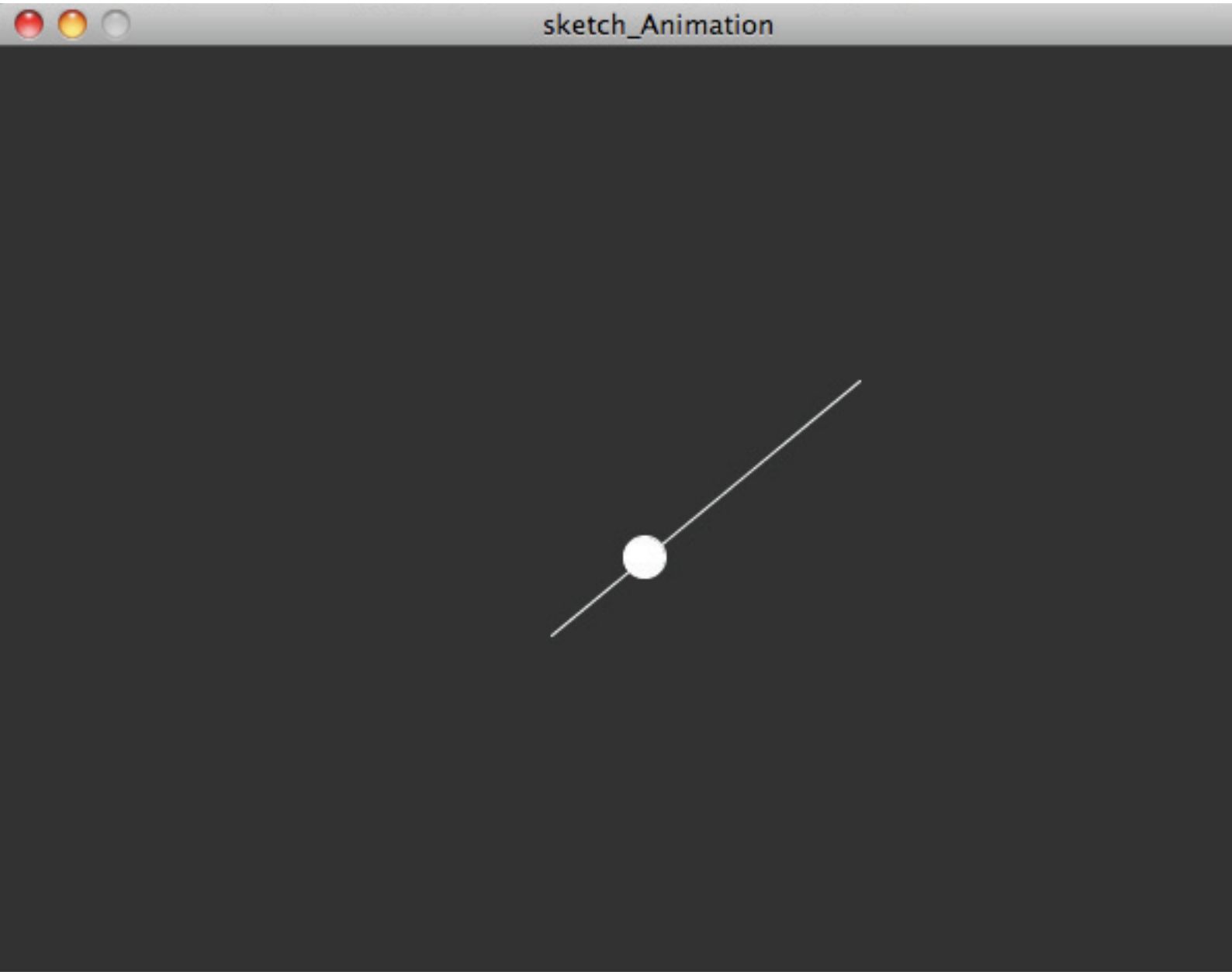
Es wirkt im Moment jedoch noch alles ein bisschen Random und auch kann ich (noch) nicht alle Änderungen nachfolgen, wie diese überhaupt entstehen.

Die Bilder oder der Output sind aber komplexe, schöne Gebilde von Abstrakten Wolken mit Regen oder ein Art Unterwassertieren ähnlich der Qualen.

VECTOR & ANIMATION
BALL AUF LINIE

Mit Hilfe einer int Funktion, welche die Mausklicks zählt ist es möglich bei einem Klick den Startpunkt und bei einem weiteren, zweiten Klick den Endpunkt zu bestimmen. Das indem eine if-Abfrage nachschauen kann, wie gross der value-wert, resp. wie viel mal der Mausklick betätigt wurde.

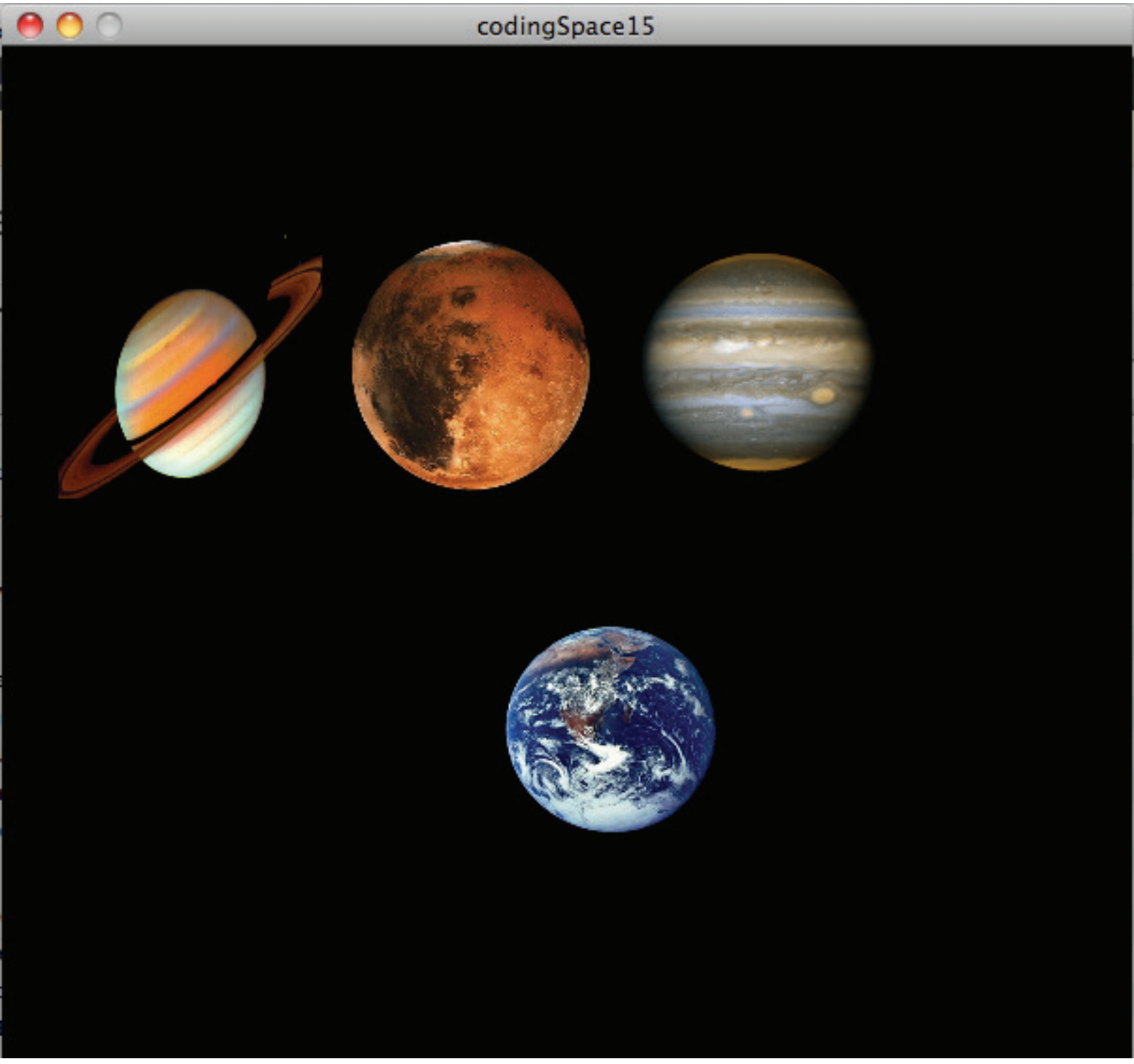
Um den Ball wieder zurück zu schicken zum Anfangspunkt nimmt man auch eine if, else Abfrage, welche Anstatt den Ball am Schluss wieder an den Anfangspunkt befördert, den Ball einfach rückwärts zum Startpunkt laufen lässt.



Screenshot von der mit zwei Mausklicks generierten Linie und dem Punkt, welcher von A nach B und wieder zurück läuft

SVG + BILDER
PLANETEN

Bei dieser Aufgabe habe ich den Code von Max so geändert, dass die Erde als .png ohne Hintergrund geladen wird und wie in der Aufgabe gefordert der Mausposition folgt. Für die popMatrix werden nur die ersten 3 Planeten(jpg) der imageList geladen und ausserhalb dieser Matrix noch die Erde (png) geladen, welche mit den Positionen mouseX und mouseY immer dem Mauszeiger folgen. Indem ich von der Funktion imageMode(CENTER) gebrauch mache, ist der Mittelpunkt der Erde auch immer schön dort wo der Mauspfeil hinzeigt.

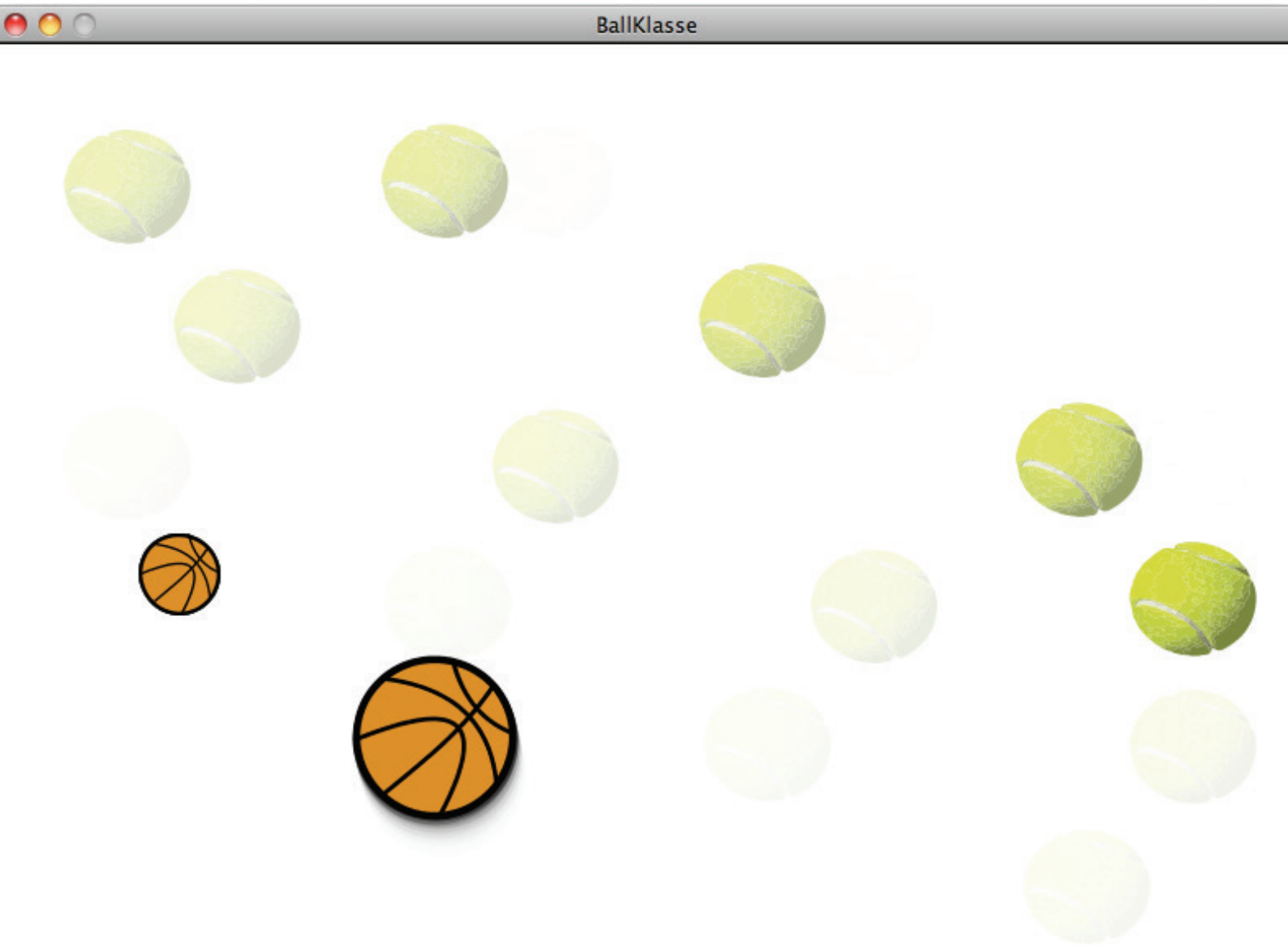


Screenshot, drei Planeten und die Erde welche dem Mauszeiger folgt

KLASSEN
BÄLLE

Bei der letzten Aufgabe vor der Endaufgabe, lernten wir eine nicht ganz so einfach nachzuvollziehende aber wichtige Programmierkonstruktion in Java. Klassen kann man als Model von Objekten benutzen, ein Art als Bauplan für ein Objekt. Das Heisst man definiert wie das Objekt aussehen soll, gibt diesem Objekt einen Namen und kann dieses Objekt dann später im Code nur mit Hilfe des Namens zeichnen.

Wir mussten ein Programm erweitern, damit mehrere Bälle unabhängig voneinander platziert werden und eine neue Art von Bällen ins Programm implementieren.



Screenshot des Klassenprogrammes, mit einem neuen Ball (Tennisball) und den beiden unterschiedlich grossen Basketbällen

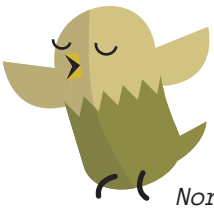
ENDAUFGABE
BIG FAT WHALE

Entwickelt aus dem Gameklassiker Break Out ein neues Spiel oder eine abgeänderte Spielvariante.

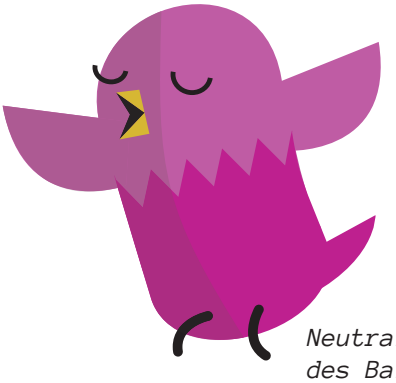
Für die Endaufgabe entwickelte ich eine andere Variante des klassischen Break Out Spieles.
In Big Fat Whale will ein verspielter Wal mit Hilfe seiner Blasdüse mit Möven spielen, diese jedoch werden durch die Wassermenge die der Wal hochschiesst getroffen und versinken im Meer.

Das Spiel besteht aus drei verschieden farbigen Möven (Blöcke), einem Wal (Schläger) und einem Wassertropfen (Ball). Zwei der drei Möven lösen eine einfache Funktion aus wenn man sie vom Himmel schiesst.

Beschleunigt den Ball um das Doppelte



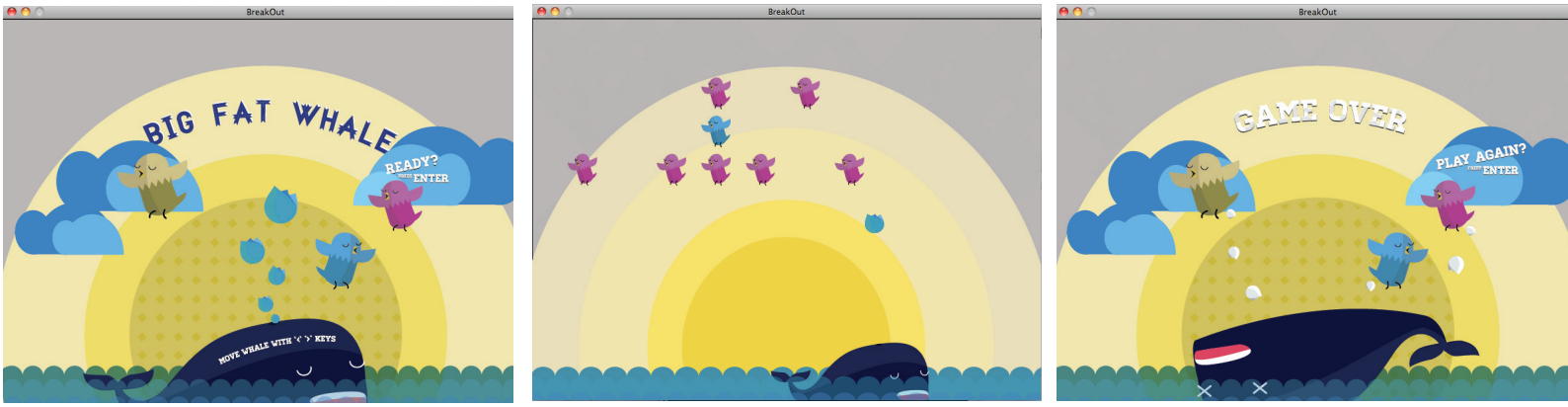
Normale Möve (Block)



Neutralisiert die Geschwindigkeit des Balles

Da Programmieren bis jetzt noch nicht meine grösste Stärke ist, beschloss ich mich vermehrt auf das Design und die Zusammengehörigkeit zu konzentrieren. Inspiriert von diversen, geometrisch, aufgebauten (Comic)Welten entstand diese kleine, geometrische Tier/Wasserwelt.

Mir war wichtig, dass alles möglich einfach aussieht, aus normalen, geometrischen Formen besteht und gut zusammenpasst. Gleichzeitig sollte aber auch eine verspielte Welt entstehen, welche den Spieler ansprechen soll (und vielleicht auch ein bisschen von der einfachen Gamekomplexität ablenken).



Screenshots des Games mit Startbildschirm, während dem Spiel und Game Over-Bildschirm

Als nächstes möchte ich dem Spiel noch ein bisschen mehr komplexität geben. Als Beispiel, dass die Möven auf den Wal zurück ~~scheissen~~ schiesen können und sich so der Spielmodus ein bisschen erschwärt. Auch mehr verschieden grosse Möven mit speziellen Funktionen wie z.B. mehr Robustheit oder solche die den Wal schrumpfen lassen für eine gewisse Zeit möchte ich ins Spiel einfügen. Und schön wäre auch wenn der Wal jedes mal wenn er einen Vogel trifft vor Freude einen kleinen Sprung macht. Dafür würde ich ein GIF verwenden, welches den Wal springen lässt oder wenn möglich nur mit Code animieren.