CodingSpace IAD Interaction Design
1. Semester, 2013
David Wyssen

CodingSpace
Dokumentation

CodingSpace
Dokumentation

Einführung

Für den Einstieg in das Programmieren verwenden wir das Programm Processing. Die Sprache die Processing benützt basiert auf Java.

Diese Dokumentation gilt als Repetition und Zusammenfassung der in den 2 Wochen gelerntem Stoff im Kurs «CodingSpace Basic». Es soll kleines Nachschlagewerk sein, in dem ich in den kommenden Monaten bei Bedarf einige Funktionen nachschauen kann. Anschließend werden einige Programmbespiele abgebildet, mit inhaltlicher Erklärung zum Aufbau der Programme. Anhand von diesen Bespielen gehe ich auf einige Begriffe und deren Funktion ein. Die Programmbeispiele fassen verschiedene Lerntage zusammen.

Um den Ablauf eines Programms zu begreifen half mir oft ein vereinfachtes Flussdiagramm, welches ich zu einigen Bespielen gezeichnet habe.

Den Source-Code der verschiedenen Übungen kann man auf meiner Blogseite downloaden.

Teils der Erklärung sind auch als Kommentare direkt im jeweiligen Code gespeichert und sind nach dem Download im Code sichtbar.

Editorial

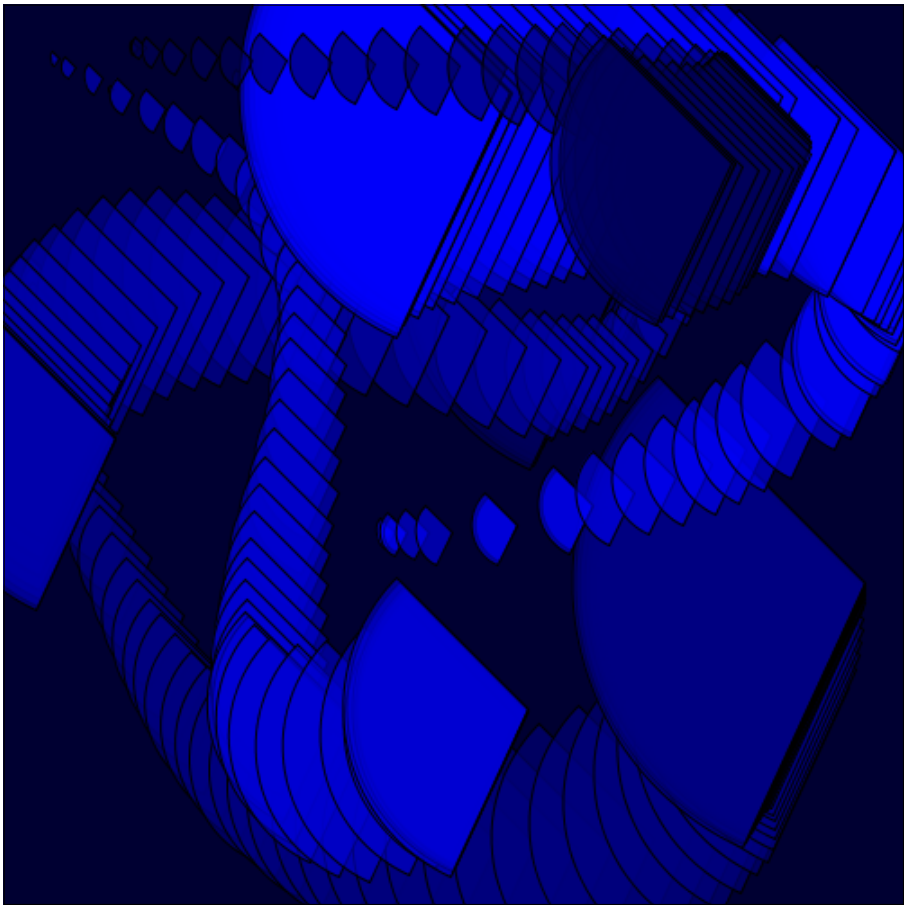
Zeichenprogramm
Code 1

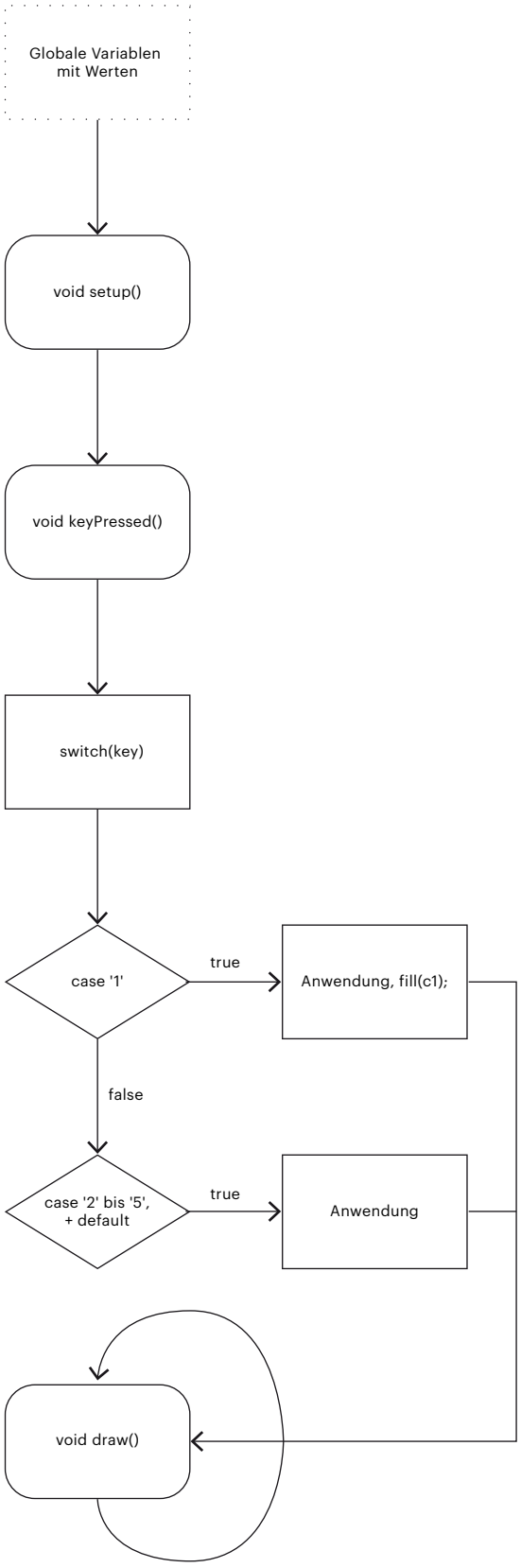
Aufgabe / Ziel

Mein Ziel war es, dass bei Aktivierung der linken Maustaste auf einer vordefinierter Bühne ich einen grafischen Fächer zeichnen kann. Mit den Tasten 1 bis 5 habe ich eine Auswahl von verschiedenen Farben, drücke ich die rechte Maustaste lösche ich vereinzelte gemalte Flächen. Das Löschwerkzeug ist eine Ellipse mit einer vordefinierten Grösse. Die Grösse der gezeichneten Fächer ist abhängig wie lange ich die linke Maustaste drücke. Je länger die Taste gedrückt wird desto grösser ist die gezeichnete Fläche. Damit wird eine Tiefenwirkung erzeugt. Jeder neugezeichnete Fächer beginnt mit der gleichen Grösse.

Referenz

Zeichenprogramm_Code1.pde





`color c1 = color(0, 0, 50, 150);` Farbdefinition der einzelnen Füllfarben. Die Farben sind als Globale Variable definiert und werden in rgba angegeben. Beispiel `(0,0,50,150)` ist ein dunkles Blau mit einem Trasparenzwert von 150.

`int x=0;` Globale Variable mit geradem Wert.

Im Event `void setup()` (Programmstart) wird die Grösse und Hintergrundfarbe der ausgegeben Bühne definiert. In diesem Fall ist die Grösse `size` 500x500 Pixel. Wird einmalig aufgerufen beim Start des Programms.

In dem Ereigniss `void keyPressed()` werden die Bedingungen als `cases` definiert. Falls der case das Zeichen «1» gedrückt wird, dann nehme die Farbe `(C1);`

In `void draw()` wird nun der Fächer `arc` gezeichnet. `void draw()` ist in einer Endlosschlaufe und wird 60/s aufgerufen.

Fallls die Maus gedrückt wird: `if (mousePressed) {x=x+5;`
Wird der Wert der globalen Variable neu definiert, wird der Wert x mit 5 addiert

Falls: `if (mouseButton == LEFT)`
Wenn die linke Maustaste gedrückt ist dann zeichne einen Fächer.
Die Zeichnungs-Position ist mit der Funktion `mouseX` und `mouseY` gespeichert. Heisst die Fläche wird an die Stelle des Cursor gezeichnet.
Breite und Höhe des Fächer ist mit der Variable `x` definiert. Die `ellipse` (falls die rechte Maustaste gedrückt ist) hat eine Grösse von 100 auf 100 Pixel sowie keinen Rahmen `noStroke();` und die gleiche Füllfarbe wie der Hintergrund.

In diesem Fall sieht man eine Bedingung `(mousePressed)` welche Bedingungen `(mouseButton == LEFT)` beinhaltet.

Anhand diesem Beispiel sieht man auch das die Parameter einer Form (Farbe, Dicke, etc.) vor der Eingabe der Form definiert werden.

Wald
Code 2

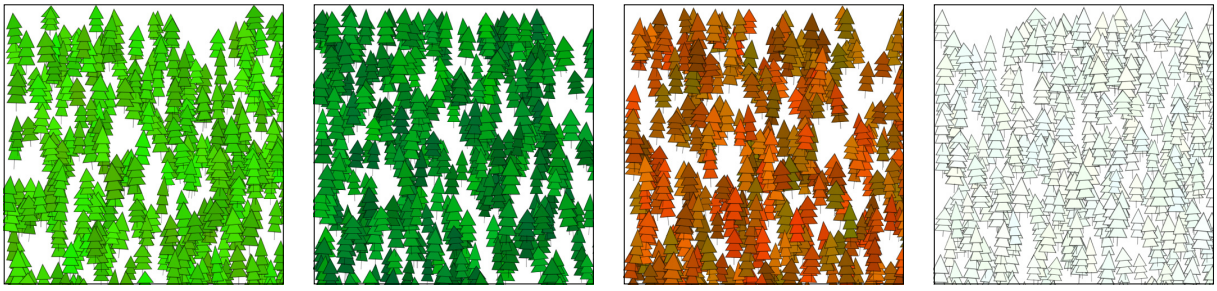
Aufgabe / Ziel

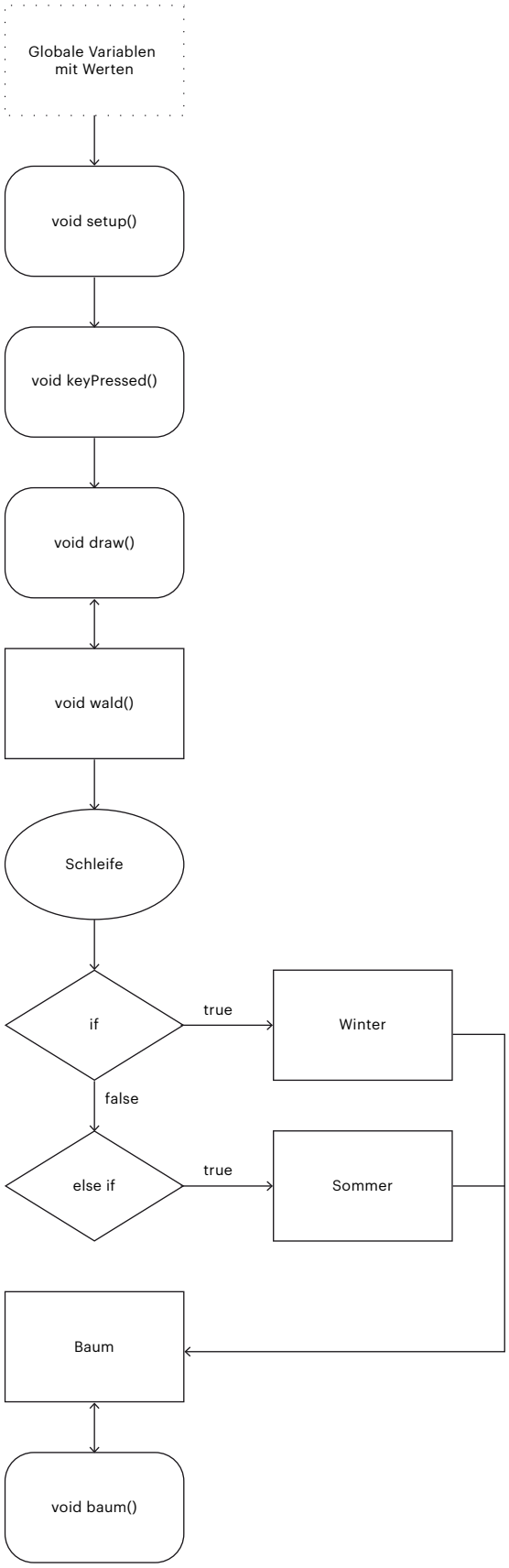
Die Aufgabe war ein Programm zu erstellen welches beim Starten einen Wald auf die vordefinierte Bühne zeichnet.

Bei meiner Lösung zeichnet das Programm zu Beginn anhand von einer Schleife die Anzahl der grafisch reduzierten Bäume für den Wald. Per Aufruf verschiedener Tasten kann ich meinen Wald den vier Jahreszeiten anpassen. Die Taste «f» zeichnet eine Wald im Frühling, «s» einen Sommerwald, mit der Taste «h» den im Herbst und «w» steht für eine Winteransicht. Die Farben der einzelnen Bäume, die Pflanzkoordinaten, deren Grösse und in welche Richtung sie stehen, entstehen per Zufall innerhalb vorgebenen Werte.

Referenz

Wald_Code2.pde





Im `void setup` ist neu die Funktion `noLoop();` deren Aktion ist, das Programm die ganze Sequenz nur einmalig abzuspielen.

Im Event `void draw()` wird bei `wald();` ein neuer Datenstamm eingelesen welcher einige Zeilen weiter unten ausgeschrieben wird.

Eine weitere Funktion kommt im Event `void keyPressed()` vor. In der Funktion `redraw();` wird dem Programm hingewissen, die Sequenz bei `void draw()` wieder weiterzulesen.

In `void wald()` sind folgende Aktivitäten, Farbdefinition der Bäume, Auswahl der Jahreszeit, und das Versetzen der Pflanz-Koordinaten. Im Datentyp `color` sind die Farben jeweils in den gewünschten Farbkanälen mit einer Zufallfunktion definiert. Der Zufall wird mit der Aktion `random` definiert, z.B. `color c1 = color(random (240, 254), 255, random (240, 254));` wird in den Kanälen Rot und Blau die Farbwahl per Zufall vom Wert 240 – 254 gewählt.

Die neuen Pflanz-Koordinaten werden innerhalb von `pushMatrix();` und `popMatrix();` gezeichnet. Per Zufall wird die Position `translate(random(0, width), random(-10, height));` die Richtung `rotate(random(-0.1, 0.1));` und die Grösse `scale(random(1, 1.65), random(1, 1.65));` in der die Bäume wachsen sollen. Danach wird der Datenstamm `baum();` gelesen.

In diesem neuen Datenstamm `baum();` wird nun der einmalige Baum gezeichnet. Bis jetzt wurde definiert wie gross und dicht der Wald sein soll, das aussehen der Bäume und wie ihr verhalten sein soll. Nun muss die Form des Baums gezeichnet werden. In `void baum()` wird der Baum, die Tanne in unserem Beispiel, anhand von übereinanderliegenden Dreiecken `triangle` gezeichnet. Der Stamm ist eine Linie `line`.

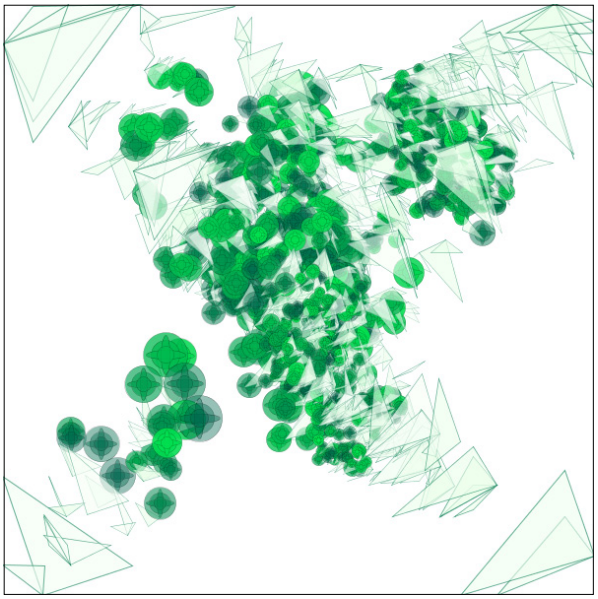
Wurzel
Code 3

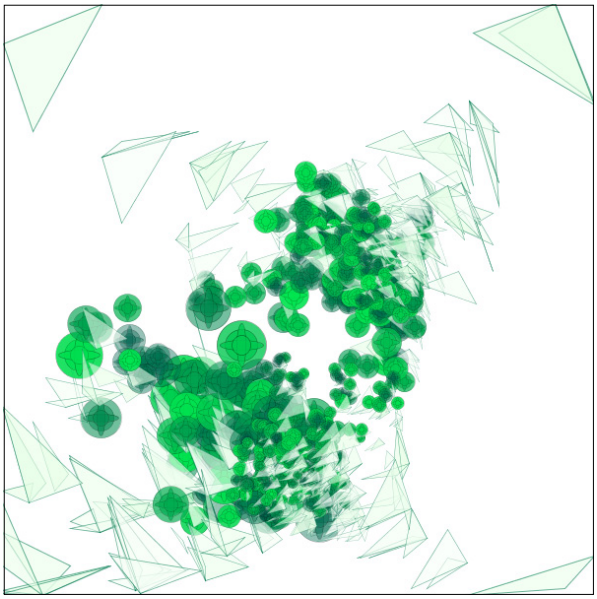
Aufgabe / Ziel

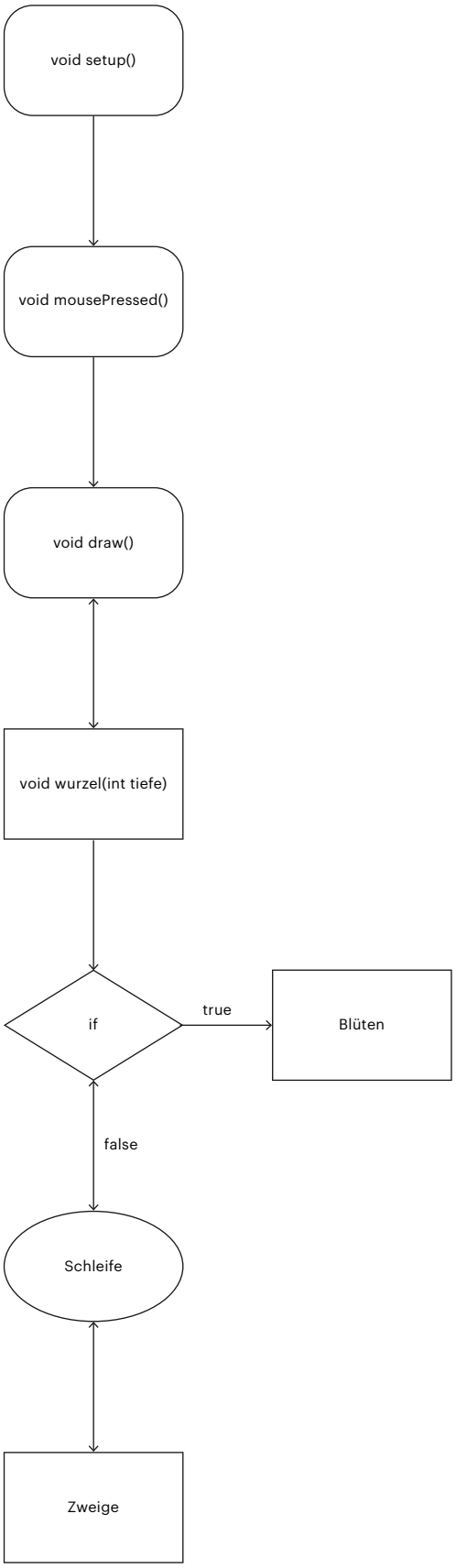
Anhand von Rekursive Funktionen/Fraktalen zeichne ich einen Baum mit verschieden langen Ästen und daran wachsenden Blüten. Bei meinem Beispiel werden von allen Ecken der Bühne aus abstrakte Bäume gerendert. Bäume deren Ästen aus der Form eine Dreiecks kommen und Blüten deren Grundform aus einer Ellipse hergeleitet sind. Per Mausklick kann ich jederzeit ein neues Zufallbild rendern. Anhand von Schleifen, verknüpften Datenstämmen und Zufallszahlen werden die Bäume gezeichnet.

Referenz

Wald_Code3.pde







In `void draw()` werden die Datenstämme `wurzel()` in der Anzahl verschieden geladen. Die Anzahl wird in der Klammer angegeben. Über `rotate(PI/2);` wird der Baum gedreht.

In `void wurzel(int tiefe)`, ein Event mit Rückgabewert, wird zuerst über eine Schleife abgefragt, ob schon genügend Zweige gezeichnet sind. Danach werden die Blüten gerendert. Die Zweige werden in der x-, und y-Achse per Zufall zwischen fix gesetzen Werten gezeichnet. Der Event `void wurzel()` hat eine Abbruchbedingung damit keine Endlosschleife geschieht.

Basketball
Code 4

Aufgabe / Ziel

In dieser Übung konnte ich anhand Klassen einige verschiedene Bälle auf eine Bühne laden. Die Bälle werden als SVG-Grafik eingelesen. Mit der Maus definiert man per klick ein Ziel wohin die Bälle unabhängig voneinander hinfliegen. Die Geschwindigkeit definiert durch den Abstand des Mausklick zu den Bällen, je grösser der Abstand umso schneller fliegen sie. Auch prallen sie von dem Bühnenrand zurück wenn sie diesen touchieren. Ist die Taste «g» gedrückt springt der grüne Ball an die aktuelle Mausposition, und wenn die Taste «w» gedrückt wird springt der weisse Ball in 20 Pixel Schritten um 45 Grad. Im Hintergrund habe ich noch ein Bild im Format .jpg geladen.

Tab Bouncingball ist die Hauptklasse, darin wird das ganze Verhalten des Balles definiert. Die Klassen Ballextend und Ballextendfort beziehen ihr Verhalten von der Hauptklasse. In den Klassen kann man aber auf das Verhalten vereinzelt einfluss nehmen, bzw. verändern.

Referenz

Basketball_Code4.pde

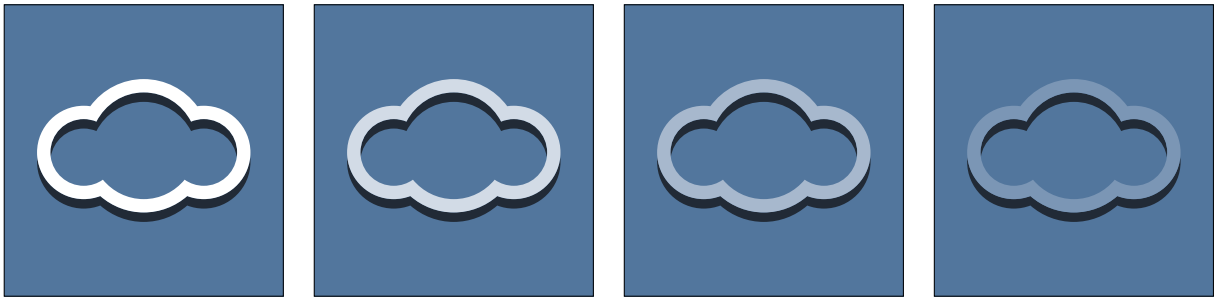
Wetter Game
Code 5

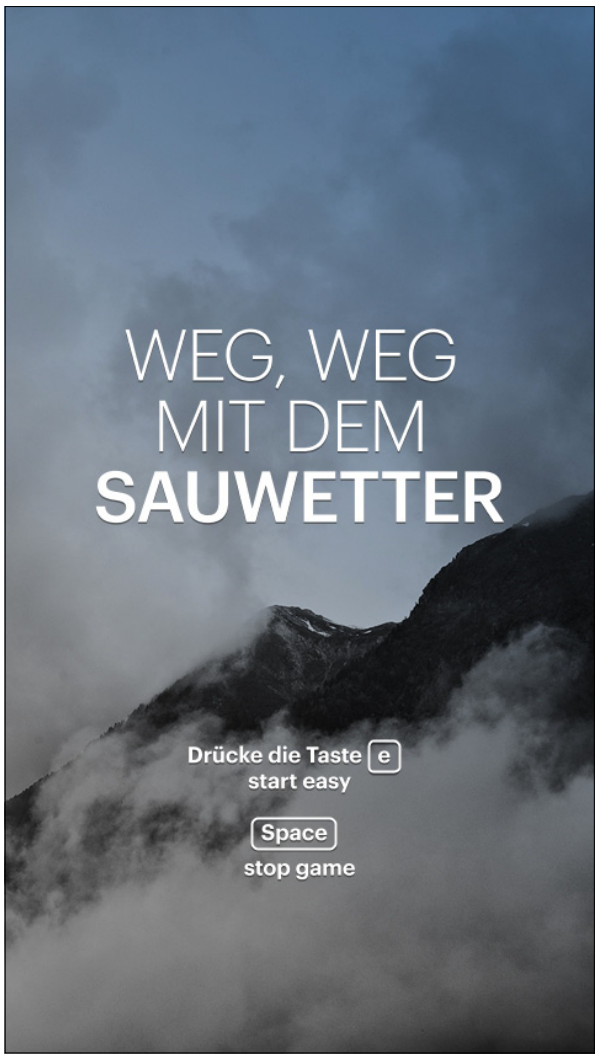
Aufgabe / Ziel

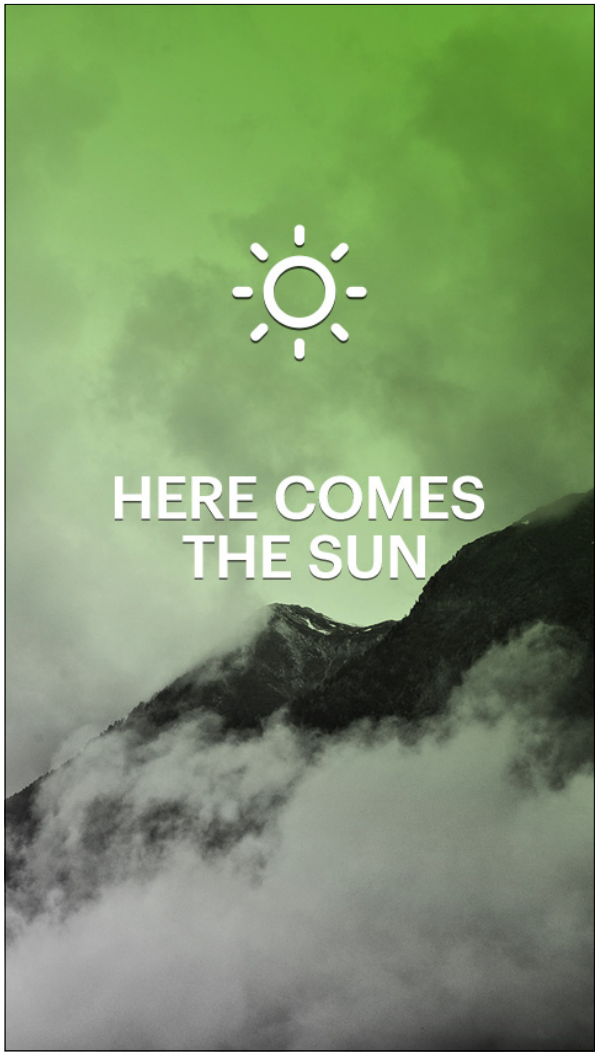
Auf Basis von dem Spiel BreakOut entwickelte ich ein neues Spiel. Es handelt sich dabei um ein Wetter-Spiel. Sauwetter regiert die Welt. Ziel ist es, wieder freie Sicht mit Sonnenschein zu erspielen. Dies geschieht in dem man diverse Wolken vom Himmel schiesst. Die verschiedenen Wolken-Arten habe unterschiedliche Lebenslängen, welche durch abgestufte Transparenz sichtbar werden. Einige muss man öfters treffen damit sie entgültig vom Bildschirm verschwinden. Das Spielformat hat die Proportionen eines Iphone5. Denn die Weiterentwicklung der Idee wäre, dass unabhängig vom Standort wo man es spielt, im Hintergrund ein Bild aus der Umgebung eingeblendet wird. Spiel ich also auf dem Säntis erscheint im Hintergrund der Säntis. Oder das Hintergrundbild wird per Kamera des Smartphone's live hineingeladen, eine Art Augmented reality Game.

Referenz

WetterGame_Code5.pde




WEG, WEG
MIT DEM
SAUWETTER
Drücke die Taste e
start easy
Space
stop game


HERE COMES
THE SUN


GAMEOVER
LUSCHE