

# Hausarbeit

Generatives Gestalten

Felix Bukatz 550105

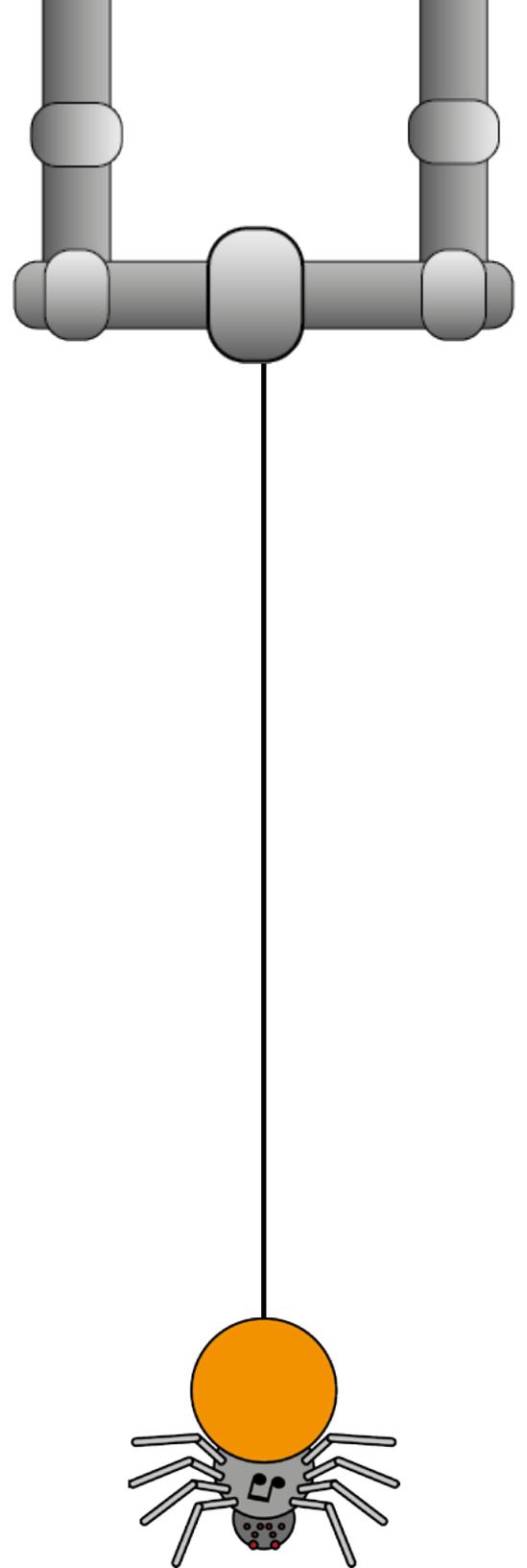
„DropWebs“

## Einführung

Für die Hausarbeit in dem Fach „Generatives Gestalten“ galt es ein Tool mit Processing zu entwickeln, Ich habe mich für das Erstellen eines Sound- und Beatgenerators entschieden.

Ziel des Tools ist es auf spaßige, spielerische Art der Interaktion zusammenhängende Töne, kleine Melodien und Rhythmen zu erzeugen. Im Vordergrund steht hier die Interaktion mit der Physik und das Herumexperimentieren mit den verschiedenen Elementen um sich ohne Musikverständnis auf kreative Art mit Musik auseinander setzen kann. Weiterhin bietet das Tool die Möglichkeit Gehör und Rhythmusgefühl zu trainieren.

Im Verlaufe der Entwicklung haben sich erstaunliche Möglichkeiten dieses Tools gezeigt. So haben sich die Anwender längere Zeit darin „verloren“ oder tatsächlich musikalisches Talent entdeckt.

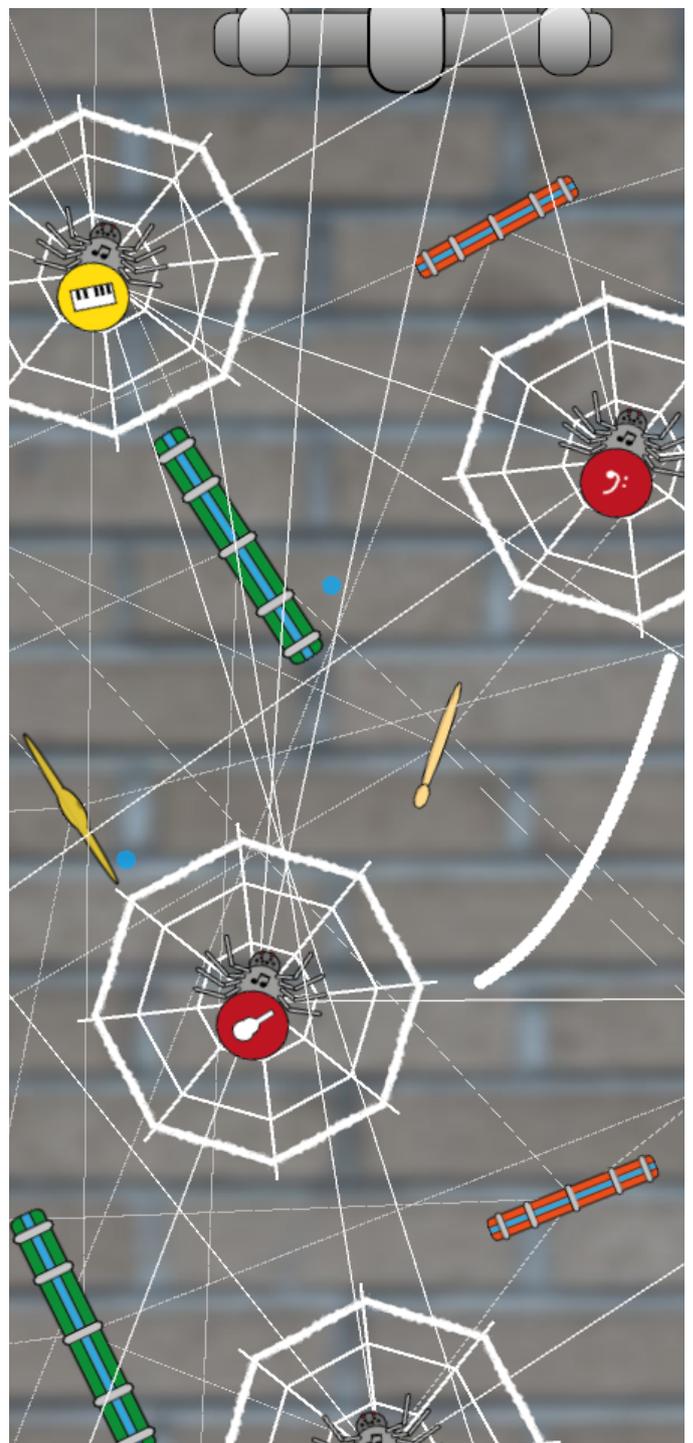


## Inhalt und Funktion

Die Szenerie in DropWebs stellt eine Kellerwand dar, an deren Decke tropfende Rohre sitzen. An dieser Wand leben Spinnen, die sich zu ihrer regelmäßigen Probe dort versammeln. Mit Hilfe einiger Tools ist der Anwender in der Lage sie zu platzieren und mit weiteren Instrumenten auszustatten. Einmal platziert spielen die Spinnen auf ihren Netzen und Rhythmusinstrumenten Musik.

Ist man mit seiner Kreation zufrieden, kann man das Ergebnis im Livemodus überprüfen. In diesem Fall gehen die Spinnen Live on Stage und spielen Ihre Musik zu einem Backingtrack. Nun stellt sich heraus wie sauber gearbeitet wurde.

Zur genaueren Anwendung mehr im Abschnitt „Tutorial“.



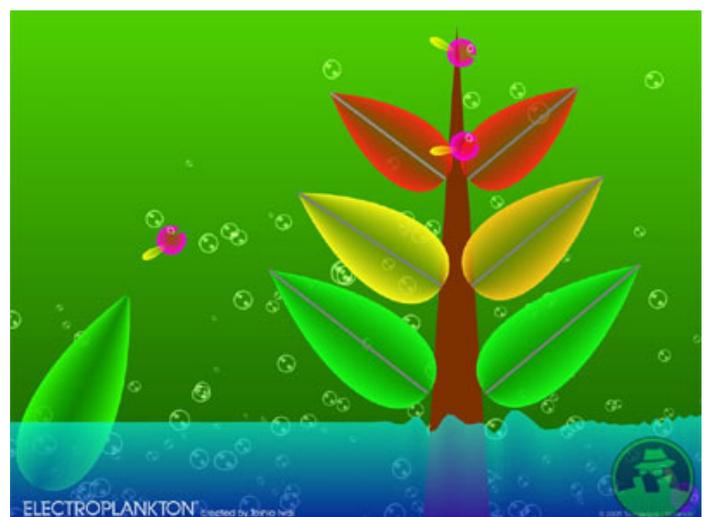
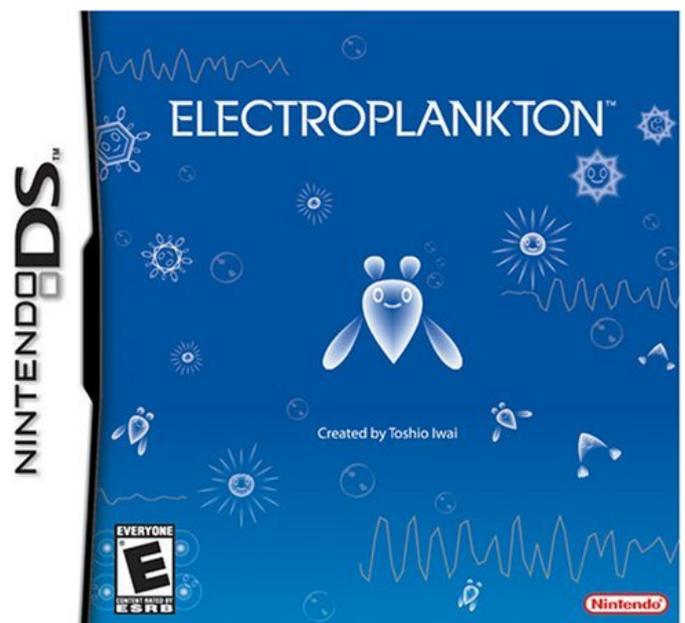
## Inspiration

Die Ursprüngliche Idee stammt aus dem Spiel „Elektroplankton“ für den 2004 erschienenen Nintendo DS.

In dieser Anwendung gab es keinerlei Spielziel, außer mit der Umgebung und den Akteuren angenehme Klänge zu erzeugen. Inspiriert hat mich der Modus „Hänenbow“ indem kleine Zooplankton von einem Blatt auf andere Blätter springen. Bei einem Aufprall springen sie weiter und erzeugen einen Klang, abhängig vom Winkel des Blattes mit dem sie kollidieren. Am ende Ihres Weges fallen sie mit einem leisen „plopp“ ins Wasser zurück.

Ich wollte gerne eine Form von Tool erstellen mit dessen Hilfe man kreativ mit Tönen experimentieren kann - jedoch so simpel, dass auch Nichtmusiker Spaß mit dieser Anwendung haben können.

Als Gitarrist sind mir dann sofort Spinnen eingefallen, die ihr Fäden spannen um mit abprallenden Gegenständen Musik zu erzeugen.



# Steuerung:

Tool auswählen: Klick auf Button in der Toolbar

Tool rotieren: Mausrad/Scroll

Tool platzieren: Klick bei ausgewähltem Tool in die Szene

Tool entfernen: Rechtsklick auf platziertes Tool

ScreenShot: Enter

Live-Modus: Alt

Bsp. Rhythmus

## Tutorial

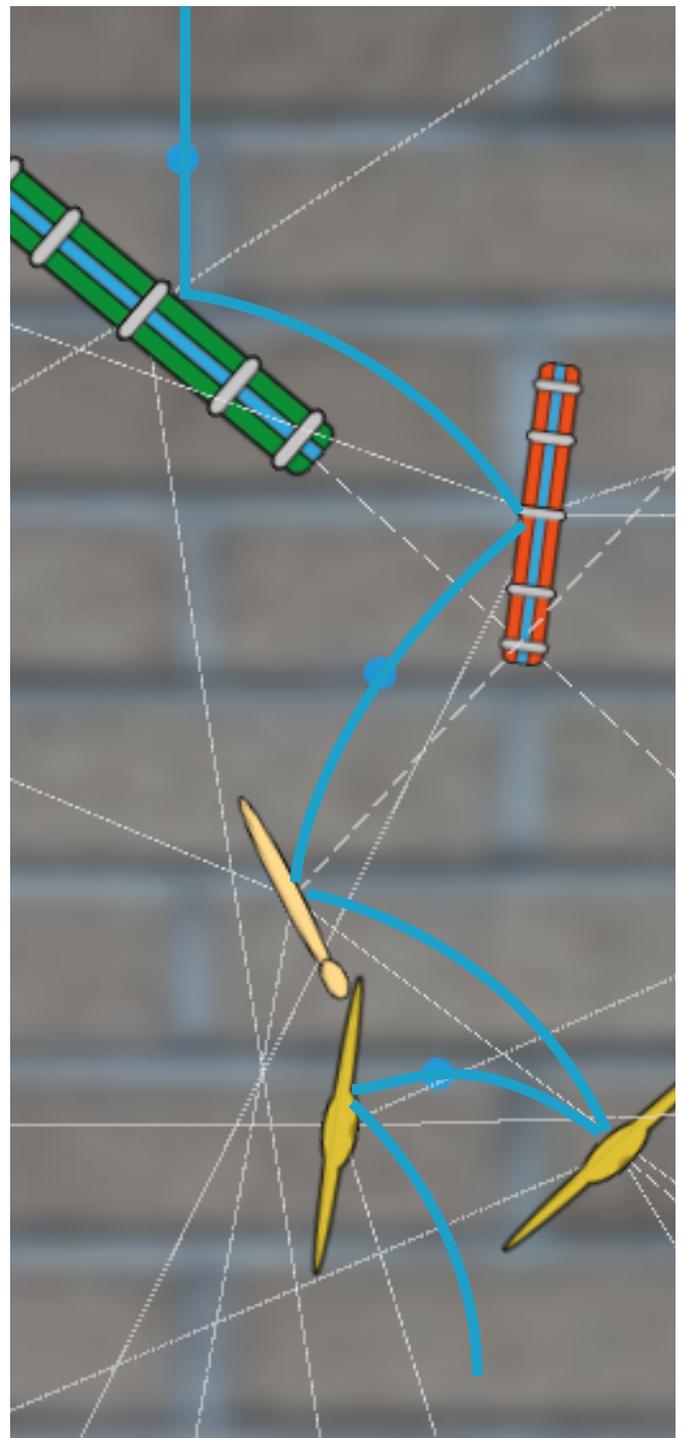
Bevor die technischen Hintergründe erläutert werden, zunächst eine Einweisung in die Bedienung.

Das Programm startet mit einem internen Tutorial, welches jedoch durch drücken der Leertaste deaktiviert werden kann (Um Verwirrung zu vermeiden Pausiert die Anwendung bei einer aktiven Meldung - einige Nutzer haben dies als störend empfunden).

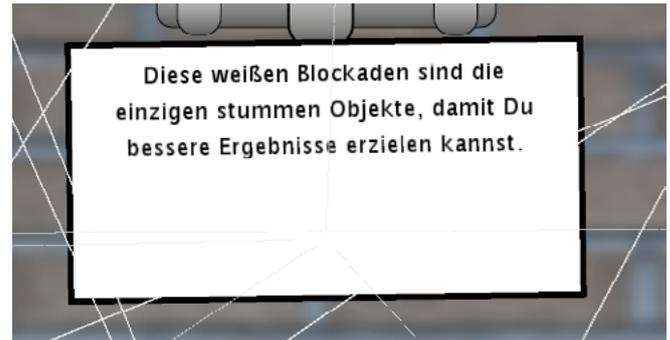
Zu Beginn ist es als Anfänger ratsam, zuerst alle Elemente zu entfernen, die für das Tutorial erzeugt wurden. Playtests haben gezeigt, dass es vielen Anwendern leichter fällt zuerst einen passenden Beat zu erzeugen und dann mit Hilfe der Spinnen Töne passend erklingen zu lassen.

Für Rhythmen eignet sich das linke Rohr besonders gut, da es 1 mal pro Sekunde tropft (60 BPM) und wiederkehrende Beats ermöglicht. Ein Beispiel ist ein Simple-8-Beat.

Es sind natürlich viele andere Kombinationen möglich!



## Bsp. Tutorial



## Bsp. Spinnen

Ist einmal ein Beat erzeugt worden, können die Spinnen nun platziert werden. Besonders zu beachten sind hier die Gewünschten Abprallwinkel der Netze.

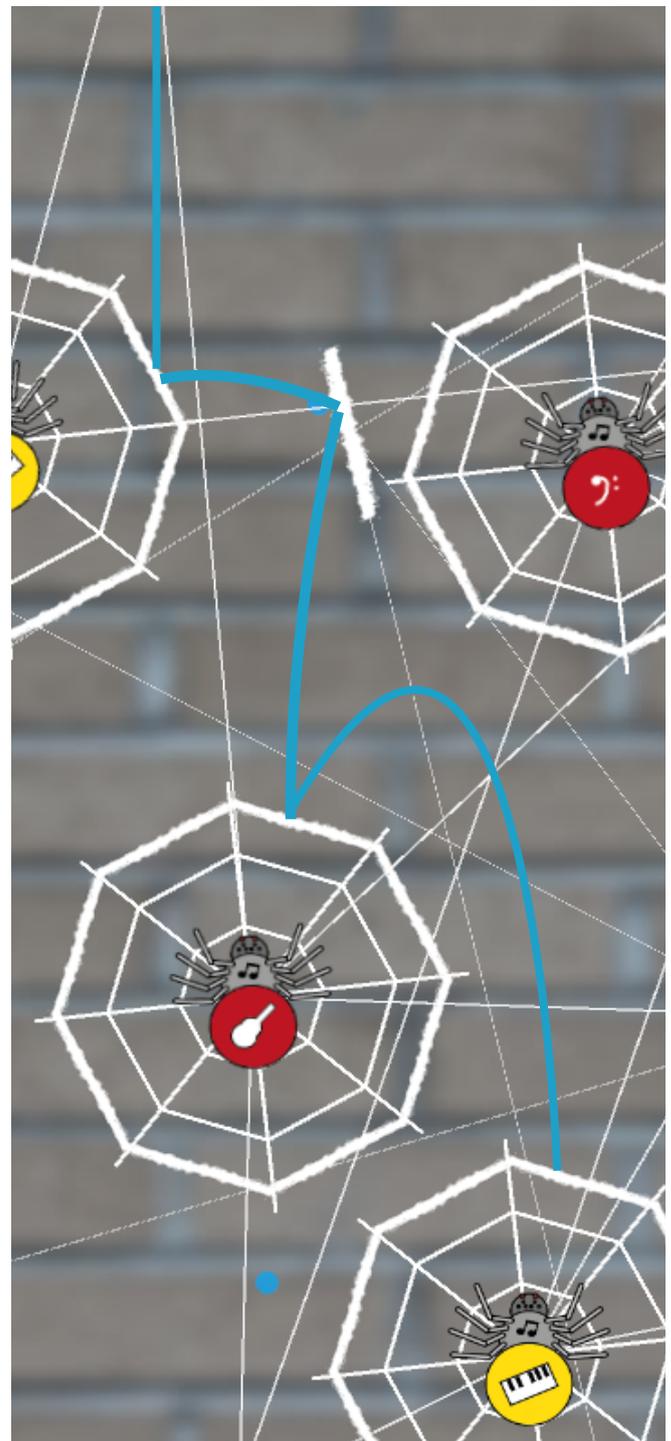
Um bestimmte bereiche des Netzes erklingen zu lassen, sind manchmal Tricks notwendig. Daher wurde ein Tool implementiert, welches Stumm ist um die Soundkulisse nicht zu überladen wenn die Tropfen an ein bestimmtes Ziel geleitet werden sollen.

BeatTools und Spinnen können natürlich nach Belieben kombiniert werden.

(Blaue Linien nur zur Illustration des Sprungverhaltens der Tropfen)

Das beste Anwendungserlebnis habe ich auf 1920\*1080 Pixeln in Playtests herausgefunden.

Die die Verwendung von Physiksimulation und zweidimensionaler Optik ermöglicht eine berechnbare Interaktion durch Beobachten der Tropfen und Einschätzung der Entfernungen. Die Interaktion lädt ein zu „was wäre wenn ich...“.



## Entwicklung

Für die Entwicklung wurden Minim (Soundbibliothek) und Box2D (Physik) als externe Bibliotheken verwendet.

### Struktur

DropWebs wurde aufgrund der Toolimplementierung sehr modular aufgebaut. Die eingerückten Klassen erben von der vorherigen oder werden von dieser implementiert. Die Struktur gliedert sich in:

- Hauptklasse (setup, draw etc)
  - DropManager
    - Drop
  - ToolBar
    - Toolbutton
    - Tool
      - Bridge
        - Particle
      - Platform
        - Snare
        - [...]
        - Base
      - Web
        - Spider
  - Message

## Hauptklasse

### **void setup()**

Läd alle notwendigen Ressourcen. Dies sind vor allem Sounds und Grafiken. Diese werden in öffentliche Variablen der Hauptklasse gespeichert um sie für andere Klassen einfach verfügbar zu machen.

Weiterhin erstellt sie die Basisszenarie und gibt eine erste Tutorial Nachricht aus.

Die DropManager und ToolBar werden erzeugt, welche wiederum ihre implementierten Objekte erzeugen. Diese entnehmen Ihre Ressourcen den zuvor geladenen Variablen.

### **void draw()**

Haupt Zeichnen/Update Methode.

Je nach Modus (Live/Training/Nachricht sichtbar) werden hier alle akteure geupdated und gezeichnet. Es werden die **void display()** und **void update()** methoden der jeweiligen Objekte aufgerufen. Diese handeln dann für sich selbst, bzw zeichnen sich selbst. So ist ein weiteres Implementieren zukünftiger Tools, Objekten, Sounds, Grafiken , etc. stark vereinfacht.

Die Methode teilt sich in **void drawAlways()** und **void drawNoMessage()** um die Darstellung von Objekten bei sichtbaren Nachrichten zu trennen, bzw die Update-Funktionen außer Kraft zu setzen um eine Interaktion zu verhindern wenn eine Nachricht aktiv ist.

Ein Tutorial Counter zählt hoch um die TutorialEvents zu ermöglichen und schickt bei entsprechender Zeit ein neues Message() Objekt heraus, wenn eine Mitteilung gemacht werden muss. Die Mitteilungsklasse verfügt über einen Konstruktor, welcher einen String erwartet und Zeichnet diese Meldung an den angegebenen Koordinaten mit einer leichten zufälligen Drehung bis ein Klick getätigt wird.

Alle Methoden für die Interaktion (MouseEvents etc) werden abgefangen und bei entsprechender Notwendigkeit die Werte an alle Objekte weitergegeben.

Mittels ALT kann ein boolean getoggled werden, der den LiveModus mit Backingtrack aktiviert und deaktiviert.

## DropManager

Die DropManager Klasse nimmt eine Position und ein Tempo entgegen. Sie verwaltet die von sich produzierten Tropfen, updated und zeichnet diese und sich selbst.

Ursprünglich war keine Physik implementierung geplant, daher ist ein festlegen auf ein Spawntempo noch so detailliert möglich aber nicht notwendig.

## Drop

Die Dropklasse basiert hauptsächlich auf eine beispiel implementation eines simplen Box2D akteurs. Tritt eine Collision auf, so werden die Teilnehmer in der Hauptklasse via **void beginContact()** ermittelt. Handelt es sich bei einem Teilnehmer um einen ClassMember von Drop, wird dieser etwas beschleunigt um die Szenerie lebendig zu halten. Handelt es sich um einen ClassMember von Tool wird (nach der Prüfung innerhalb der Klasse, ob der sound != null ist) der entsprechende Sound abgespielt.



## ToolBar

Die Toolbar Klasse verwaltet alle platzierten Tools und ruft deren **void display()** und **void update()** Methoden auf. Sie überprüft die Mausposition und zeichnet den Cursor, bzw macht das platzieren auch nur dann möglich, wenn die Maus sich in der Szene befindet und nicht über der Toolbar.

Sie positioniert die Buttons und weist ihnen die Grafiken zu, zeichnet diese und sich selbst.

Innerhalb der Klasse ist ein Switch für die Auswahl des zu platzierenden Tools welcher auf einem Enum basiert. Jeder Button erwartet in seinem Konstruktor eine Konstante dieser Art, damit er das entsprechende Tool als Rückgabe liefern kann. Auch der Cursor wird aufgrund des gewählten Tools gezeichnet.

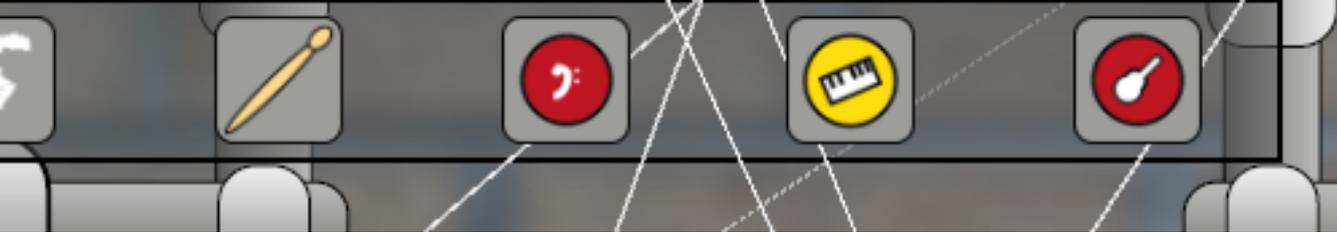
So werden einfach neue Tools an Ort und Stelle erzeugt und der Toolliste hinzugefügt. In dieser und anderen Klassen wurden Methoden wie **boolean isMouseOver()** implementiert um sie programmlogisch zu verwenden.

Jedes Objekt in der Box2D Welt kann nicht einfach aus den ArrayLists() entfernt werden, da sonst ihre Bodies zurückbleiben und weiter interaktiv in der Welt bestehen. Das Ergebnis sind dann unsichtbare Wände, die die Anwendungserfahrung natürlich zunichte machen. Daher wurden an einigen Stellen **void checkForDelete()** und **void clearForRemove()** Methoden implementiert, die für die Löschung dieser Bodies sorgen. Erstere nimmt die Mausposition und prüft auf notwendige Löschung und zweitens räumt die Bodies auf.

## Tool

Um die Erzeugung von Werkzeugen zu vereinfachen wurde die Tool Klasse erzeugt. Sie liefert alle Basisfunktionen und die Möglichkeit alle Tools in ArrayLists() zu verwalten.

Vor allem erzeugt die Klasse die notwendigen Physik-Bodies und setzt Positionen. Weiterhin liefert diese Klasse eine **void drawWeb()** Methode, die das Zeichnen der feinen Spinnenweben, die die Objekte halten übernimmt.



Es gibt drei Arten von Tools:

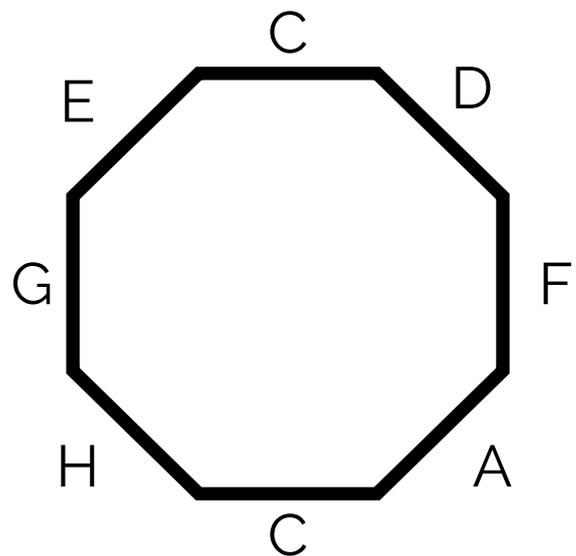
- Plattformen
- Spinnenfäden
- Spinnennetze

Alles Rhythmus relevanten Objekte erben ihre Eigenschaften von Plattform. Spinnenfäden sind von einem Beispiel der Box2D Engine abgeleiteten verknüpfte Partikel, die zu einem elastischen Band kombiniert werden. Spinnennetze sind eine Komposition aus Plattform Objekten (die Ihren Ton bekommen und entsprechend angeordnet werden. Für genauere Informationen die entsprechenden Klassen betrachten), gezeichneten Linien und einem Objekt vom Typ Spider, welcher sich als Agent nur etwas im Takt bewegt.

## Web

Die Web Klasse erzeugt um das Netz herum eine besondere Art von Plattformen. Sie erhalten die Töne der C-Dur Tonleiter. Diese werden jedoch nicht auf- oder absteigend um das Netz verteilt, sondern auf den Seiten jeweils abwechselnd. So kann man die

Netze interessanter zueinander kombinieren.



Dies sollte die Wahrscheinlichkeiten von Dissonanzen durch nah bei einanderliegenden Tönen eliminieren (Gut platziert, kann man zwei Töne gleichzeitig mit einem Tropfen zu einem Akkord spielen).

## Assets

Die Grafiken sind bis auf zwei CC Stock Fotos (Hintergründe) selbst erstellt worden. Auch die Sounddateien sind, inklusive dem Backingtrack, selbst erzeugt worden. Hierfür wurde GarageBand und ein Instrumenten Konvertierer genutzt.

## Live-Modus

Der Live-Modus dient mehreren Zwecken: Durch den Backingtrack fügt sich die Soundkulisse besser zusammen und man kann sich inspirieren lassen. Andersherum kann man jedoch auch zuerst ohne weitere Einflüsse verschiedene Anordnungen testen und sich dann von dem Ergebnis überraschen lassen. Durch hin- und herschalten und das geringfügige abändern von Positionen und Rotationen geben einem die Möglichkeit selbst herauszufinden wie empfindlich ein musikalisches konstrukt sein kann. Er bietet einem Anwender ohne Töne verwenden zu müssen die Möglichkeit seinen Beat zur Musik spielen zu lassen. Der Backingtrack

wurde in der Tonart C-Dur geschrieben. Da alle Töne des Netzes auch in der C-Dur Tonleiter zu finden sind, schließen sich Dissonanzen so aus und der Generativen Soundgestaltung steht somit nichts im Wege bis auf die selbst erzeugten Dissonanzen.

## Umsetzungs Ziel

Um die Wirkung und Anwendbarkeit dieses Tools zu testen habe ich viele Playtests durchgeführt. Die zielführende selbst gestellte Aufgabe zu diesem Projekt war das spielerische und experimentelle Interagieren mit den gestellten Werkzeugen und Musik auf eine Simple und unterhaltsame Weise. Die Wichtigsten Beobachtungen während der Playtest waren unter anderem:

- motiviertes Kombinieren
  - verrückte Ansätze
  - testen der Extreme
  - Anfragen für TabletApp
  - Spaß und Auseinandersetzung mit Rhythmen und Tönen gleichermaßen vorhanden
- (Warum klingt das jetzt nicht so gut wie das was ich vorher hatte, etc)

Mit diesen Ergebnissen ist die Anwendung meinen Ansprüchen ihrer Funktion gegenüber gerecht geworden - bietet aber auch noch viel Luft nach oben.

Besonders gefallen hat der Live-Modus und die Spinnenfäden (Tool) welche für interessante ergebnisse sorgen. Sehr beliebt war auch das austesten der Grenzen des Machbaren (Tropfen einfangen, katapultieren, sammeln).

## Weiterführende Arbeit

Trotz der schier unendlichen Kombinationsmöglichkeiten kann die Anwendung natürlich noch um vieles erweitert werden. Durch den Aufbau des Programmes können in Zukunft ohne großen Aufwand viele Erweiterungen vorgenommen werden, da der Grundstein für eine sehr umfangreiche Version der Anwendung nun gesetzt ist. Da die Erzeugung der Assets jedoch viel Zeit in Anspruch nimmt, wurde der Schwerpunkt auf die Funktionalität gelegt. Zukünftige Erweiterungen können sein:

- Mehrere Tonarten
- Mehr Instrumente
- Das nutzen von Soundsets (Rock, Techno, etc)
- Lichteffekte
- Eine Recordfunktion für den Soundexport
- Abspeichern der aktuellen Szene