

Contents

Space Journey - Dokumentation	1
Projektbeschreibung	1
2. Einleitung	1
2.1 Voraussetzungen	1
2.2 Projekt herunterladen	2
3. Kinect Azure	2
3.1 Installation	2
4. Touch Designer	2
4.1 Technische Voraussetzungen	2
4.1 Projektkonfiguration	2
4.2 Body Tracking	2
5. Blender	4
5.1 Blender Projektkonfiguration	4
5.2 Blender-OSC Connection	5
5.3 Erläuterung der aktuellen OSC Adressen	5
5.4 Weitere Blender Objekte über OSC steuern	6
6. Ableton Live Setup	6
6.1 TouchDesigner-Ableton Connection	6
6.2 Anleitung	6
6.3 Erweitern der Ableton Live Steuerung	8
6.4 Sounddesign	9
7. Probleme	11

Space Journey - Dokumentation

Creative Technologies - Fachhochschule Kiel

Sommersemester 2023

Dozenten: Prof. Dr. Steffen Prochnow, Prof. Dr. Robert Manzke

Von: David Credo, Lasse Knott, Tara Obersteller

Projektbeschreibung

Space Journey ist eine interaktive 3D-Installation, die im Rahmen von Creative Technologies an der Fachhochschule Kiel erstellt wurde. Der Nutzer kann mit seinem Körper eine Kamera durch das Weltall steuern und wird dabei dynamisch von Musik unterstützt.

Das Projekt ist eine Kombination aus Touch Designer, Blender und Ableton Live verbunden mit OSC. In Touch Designer werden Body-Tracking Daten der Kinect Azure gesammelt und an Blender bzw. Ableton Live versendet. Diese reagieren dann mit z.B. einer Änderung der Kameraperspektive oder eines Sound-Filter Cutoffs.

2. Einleitung

Nachfolgend wird die Konfiguration der einzelnen Programme, die für das Starten des Projekts benötigt werden, beschrieben. Die Programme können auf einem Rechner oder auf mehreren Rechnern ausgeführt werden. Wenn ein einzelner Rechner verwendet wird, muss dieser eine entsprechend leistungsstarke Hardware besitzen. In jedem Fall muss mindestens ein Windows Rechner verwendet werden, da die Kinect Azure nur unter Windows funktioniert. Ableton Live und Blender können auch auf MacOS ausgeführt werden. (Blender sollte nach Möglichkeit auf einem Windows Rechner ausgeführt werden, da die Performance dort besser ist).

2.1 Voraussetzungen

Um das Projekt starten zu können werden folgende Programme benötigt:

- Touch Designer
- Ableton Live
- Blender
- Kienct / Kinect Azure

Die genaue Installation und detaillierte technische Anforderungen folgen im jeweiligen Abschnitt.

2.2 Projekt herunterladen

Das gesamte Projekt ist auf dem GitLab der FH-Kiel bereitgestellt. Zum herunterladen, nutze folgenden Befehl in deiner Git-Bash:

```
git clone https://gitlab.iue.fh-kiel.de/mediaip-works/2023ss-space-journey
```

Alternative: Lade die Dateien [hier](#) herunter.

3. Kinect Azure

3.1 Installation

1. Kinect mit Stromkabel verbinden
2. Kinect mit USB 3.0 Port am Computer verbinden
3. Warten bis Treiber automatisch installiert wurden
4. Die Status-LED auf der Rückseite sollte konstant leuchten

Bei Problemen siehe [Kinect Azure Dokumentation](#)

4. Touch Designer

4.1 Technische Voraussetzungen

- Nicht-Kommerzielle Version von [Touch Designer](#) Build 2022.32660 (Auflösung von 1280 x 1280px)
- SSD empfohlen
- Leistungsstarke Grafikkarte (Schwache Modelle beeinflussen Body Tracking und Latenz)

Windows

- Windows 8.1 / Windows 10 / Windows 11
- Video RAM 4GB
- GeForce 1080 bzw. AMD HD 6000 oder besser

MacOS

- macOS 10.14 oder neuer
- Mac / MacBook 2015 oder neuer

4.1 Projektkonfiguration

1. Wenn das Projekt erfolgreich von Git-Lab geklont bzw. heruntergeladen wurde, öffne in den Projektdateien die Datei **“TD_Space_Journey.toe”**.
2. Wähle im **“Kinect Azure” Operator** (Links) unter “Sensor” die Kinect Azure aus.
3. Jetzt sollte im selben Operator das Kamerabild angezeigt werden. Außerdem sollten im “TrackingInfo” Operator Body-Tracking Daten angezeigt werden, solange jemand vor der Kamera steht.

Bei Fehlern den “Active” Schalter ausschalten und wieder einschalten für einen Reset

4.2 Body Tracking

In diesem Projekt wurden Bodytracking Daten von der Linken und Rechten Hand (XYZ) sowie die Entfernung Z von der Wirbelsäule verwendet.

1. Wenn die Kinect erfolgreich verbunden wurde, sollte im “TrackingInfo” Operator Daten angezeigt werden. Es gibt eine Liste von 100 Körperteilen deren Position getracked wird.

```

*          p1/wrist_l:tx    p1/hand_r:ty    p1/knee_r:tz
frame      p1/wrist_l:ty    p1/hand_r:tz    p1/ankle_r:tx
timestamp   p1/wrist_l:tz    p1/handtip_r:tx  p1/ankle_r:ty
p1/id       p1/hand_l:tx    p1/handtip_r:ty  p1/ankle_r:tz
p1/pelvis:tx  p1/hand_l:ty    p1/handtip_r:tz  p1/foot_r:tx
p1/pelvis:ty  p1/hand_l:tz    p1/thumb_r:tx    p1/foot_r:ty
p1/pelvis:tz  p1/handtip_l:tx  p1/thumb_r:ty    p1/foot_r:tz
p1/spine_navel:tx  p1/handtip_l:ty    p1/thumb_r:tz  p1/head:tx
p1/spine_navel:ty  p1/handtip_l:tz    p1/hip_l:tx    p1/head:ty
p1/spine_navel:tz  p1/thumb_l:tx    p1/hip_l:ty    p1/head:tz
p1/spine_chest:tx  p1/thumb_l:ty    p1/hip_l:tz    p1/nose:tx
p1/spine_chest:ty  p1/thumb_l:tz    p1/knee_l:tx    p1/nose:ty
p1/spine_chest:tz  p1/clavicle_r:tx  p1/knee_l:ty    p1/nose:tz
p1/neck:tx        p1/clavicle_r:ty  p1/knee_l:tz    p1/eye_l:tx
p1/neck:ty        p1/clavicle_r:tz  p1/ankle_l:tx    p1/eye_l:ty
p1/neck:tz        p1/shoulder_r:tx  p1/ankle_l:ty    p1/eye_l:tz
p1/clavicle_l:tx  p1/shoulder_r:ty  p1/ankle_l:tz    p1/ear_l:tx
p1/clavicle_l:ty  p1/shoulder_r:tz  p1/foot_l:tx    p1/ear_l:ty
p1/clavicle_l:tz  p1/elbow_r:tx    p1/foot_l:ty    p1/ear_l:tz
p1/shoulder_l:tx  p1/elbow_r:ty    p1/foot_l:tz    p1/eye_r:tx
p1/shoulder_l:ty  p1/elbow_r:tz    p1/hip_r:tx    p1/eye_r:ty
p1/shoulder_l:tz  p1/wrist_r:tx    p1/hip_r:ty    p1/eye_r:tz
p1/elbow_l:tx     p1/wrist_r:ty    p1/hip_r:tz    p1/ear_r:tx
p1/elbow_l:ty     p1/wrist_r:tz    p1/knee_r:tx    p1/ear_r:ty
p1/elbow_l:tz     p1/hand_r:tx    p1/knee_r:ty    p1/ear_r:tz

```

- Mit einem **Select-Operator** können beliebige Body Tracking Informationen aus der Liste ausgewählt werden z.B. "hand_r:tx" für die X-Koordinaten der rechten Hand.



Die Namen wurde im Select-Operator von "hand_r:tx" auf "Rechts_x" umbenannt.

- Per Drag-and-Drop kann jetzt eine Referenz zu einem der ausgewählten Werte erstellt werden. Diese Referenz kann für z.B. einen Ableton-Operator verwendet werden. Siehe dafür [Punkt 6.3](#)

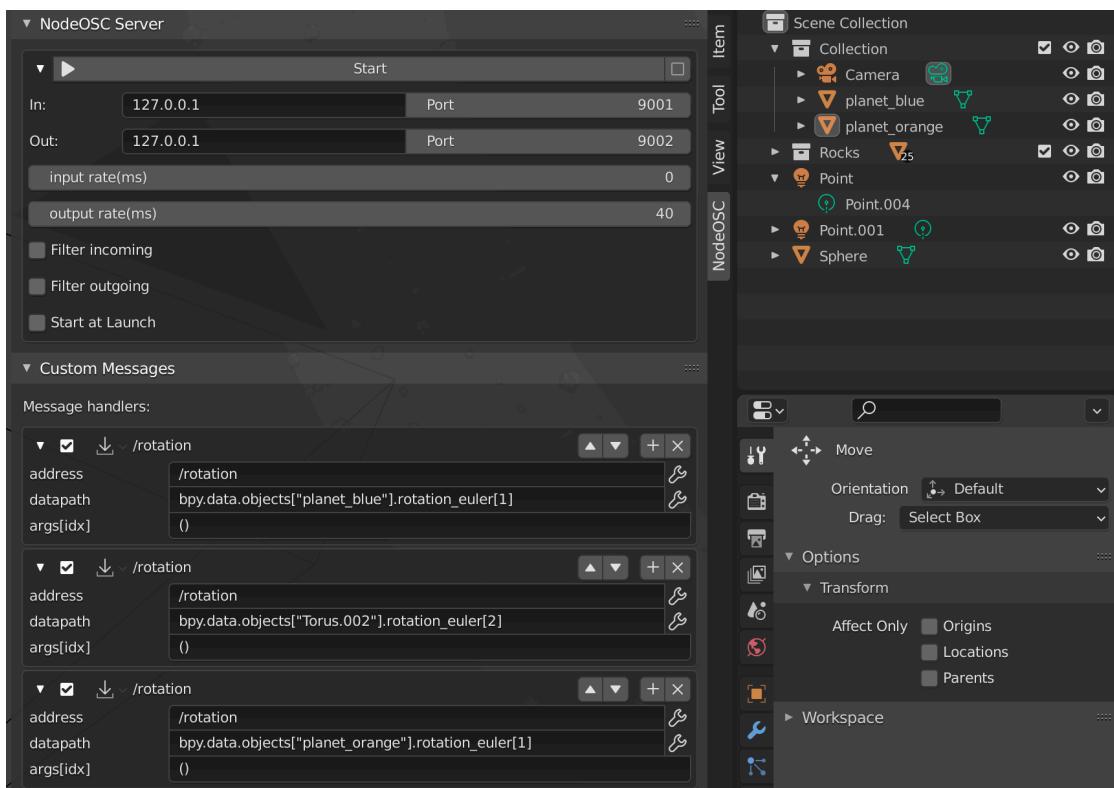
5. Blender

Technische Voraussetzungen

- Mindestens Blender 2.93
- NodeOSC Blender Plugin [Repository des Plugins](#)

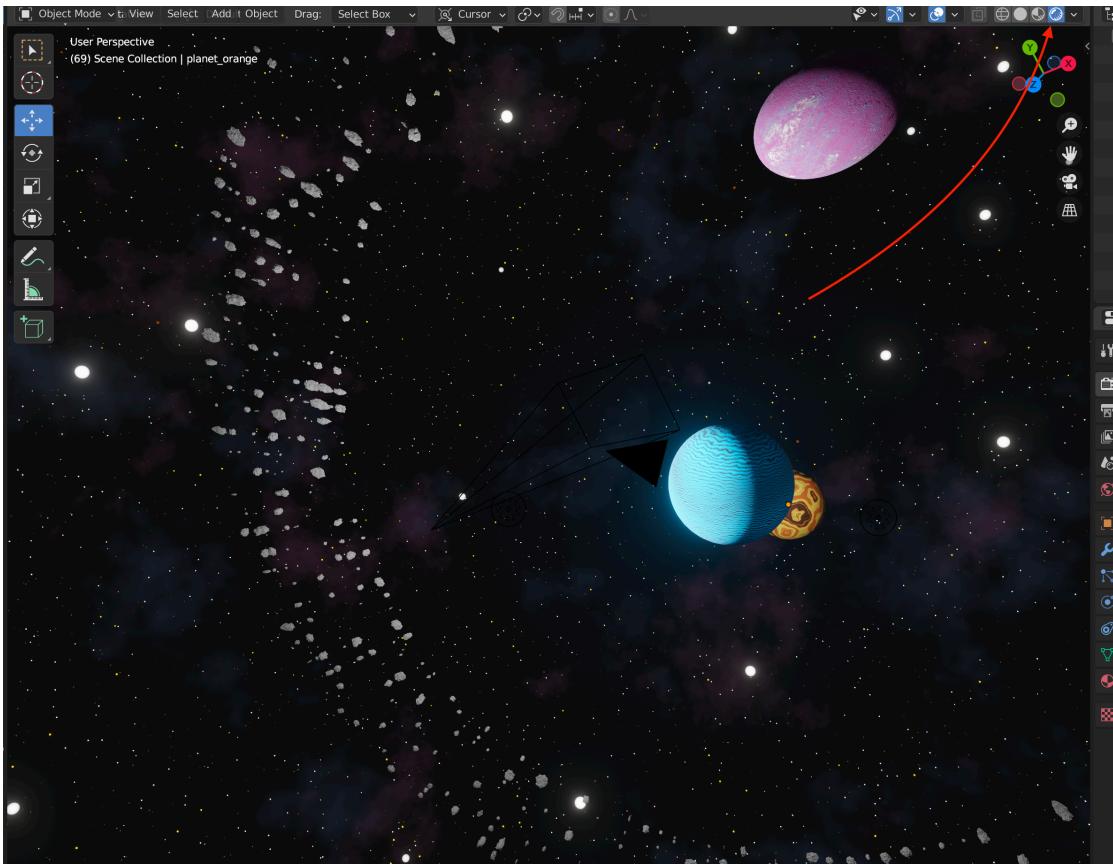
5.1 Blender Projektkonfiguration

1. Installiere das NodeOSC Blender Plugin
2. Öffne die Datei `Space_Journey.blend` im Ordner `models`
 - Lade das Plugin aus dem oben verlinkten Repository herunter
 - Speichere die ZIP-Datei in einem Ordner deiner Wahl
3. Öffne das Blender Projekt "Space_Journey.blend"
4. Navigiere zu `Edit > Preferences > Add-ons`
5. Klicke auf `Install...` und wähle die ZIP-Datei aus
6. Aktiviere das Plugin mit einem Klick auf das Kästchen links neben dem Plugin-Namen
7. Auf der rechten Seite sollte nun ein neuer Reiter mit dem Namen "NodeOSC" erscheinen



Ansicht des NodeOSC Reiters

8. Beim Öffnen des Projekts ist das **View-Port Shading** auf **Solid** eingestellt. Um die Materialien zu sehen, muss das Shading auf **Rendered** gestellt werden.



Hinweis auf das View-Port Shading

5.2 Blender-OSC Connection

1. Wenn das Projekt auf dem selben Rechner wie TouchDesigner läuft, muss die IP-Adresse nicht verändert werden
 - Wenn das Projekt auf einem anderen Rechner läuft, muss die IP-Adresse des Rechners, auf dem TouchDesigner läuft, eingetragen werden
2. Die Ports müssen nicht verändert werden, solange du diese nicht bereits für ein anderes System/Projekt verwendetest.
3. Klicke auf **Start OSC Server**
4. Wenn die Verbindung erfolgreich war und die TouchDesigner Session mit der Kinect läuft, sollte die Kamera sich nun bewegen, wenn du dich vor der Kinect bewegst.

5.3 Erläuterung der aktuellen OSC Adressen

In dem Bild unter **Punkt 5.1** sind die aktuellen OSC Adressen zu sehen. Die Adressen deuten auf die gleichnamigen TouchDesigner Channels hin. Die Channels werden von den OSC Out CHOPs in TouchDesigner gesendet.

1. **/rotation** - Rotation der Planeten
 - Die Rotation der Planeten werden von einem LFO CHOP in TouchDesigner gesteuert. So rotieren die Planeten in einer konstanten Geschwindigkeit.
2. **/body_x** - X-Koordinate der Szenenkamera
 - Die X-Koordinate der Szenenkamera wird von der X-Koordinate des Oberkörpers gesteuert. So bewegt sich die Kamera erwartungsgemäß mit dem Oberkörper.
3. **/body_z** - Z-Koordinate der Szenenkamera
 - Die Z-Koordinate der Szenenkamera wird von der Z-Koordinate des Oberkörpers gesteuert. So kann man sich von den Objekten entfernen und näher an sie heran bewegen.

4. `/hand_lz` - Z-Komponente der Quaternionrotation der Szenenkamera
 - Stellt einen Teil der Kamerasteuerung dar, die das Umsehen in der Szene ermöglicht.
5. `/hand_rx` - X-Komponente der Quaternionrotation der Szenenkamera
 - Siehe 4.
6. `/hand_ry` - Y-Komponente der Quaternionrotation der Szenenkamera
 - Siehe 4. (Wichtig: Lässt man Teile der Quaternionrotation weg, kann es zu unerwarteten Ergebnissen kommen. Es ist mit der aktuellen Konfiguration sinnvoll, alle drei Komponenten gleichzeitig zu steuern.)

5.4 Weitere Blender Objekte über OSC steuern

- Um weitere Objekte in Blender über OSC steuern zu können, müssen diese zunächst mit dem NodeOSC Plugin verbunden werden.
1. Wähle das Objekt in der Hierarchie aus, das du steuern möchtest
 2. Klick auf `Create New Message Handler` im NodeOSC Reiter
 3. Das Eingabefeld `address` erwartet eine OSC-Adresse
 - Die Adresse muss mit einem / beginnen
 - Der Name der Adresse ist abhängig von dem **Channelnamen** des Operators in TouchDesigner. (Der Operator, der vor dem OSC Out CHOP steht)
 - Beispiel: `/position`, wenn der Channelname `position` ist
 - Du brauchst die Position des Werts nicht angeben, da diese von NodeOSC automatisch erkannt wird
 - Tipp: Verschicke je OSC Out CHOP nur einen Wert, so ist die Zuordnung einfacher
 4. Im `Data Path` Feld des NodeOSC Reiters kannst du einen beliebigen Pfad eines Wertes in Blender einfügen.
 - Beispiel: `bpy.data.objects["Cube"].location.x` = X-Position des Objekts `Cube`
 - Tipp: Wenn du den DataPath nicht kennst, kannst du ihn über Rechtsklick auf das Feld und `Copy Data Path` kopieren
 - Dieser Wert wird dann vom OSC Signal überschrieben
 5. Vergiss nicht den `Start OSC Server` Button zu drücken, um die Verbindung zu starten
 6. Rinse and repeat für alle weiteren Objekte, die du steuern möchtest!

6. Ableton Live Setup

Technische Voraussetzungen

- Ableton Live 10+ (Intro, Standard, Suite)

6.1 TouchDesigner-Ableton Connection

Wenn du nur am Sounddesign interessiert bist, siehe Punkt 6.4 Um die Ableton Session über OSC Signale von TouchDesigner steuern zu können, müssen beide Programme dafür konfiguriert werden.

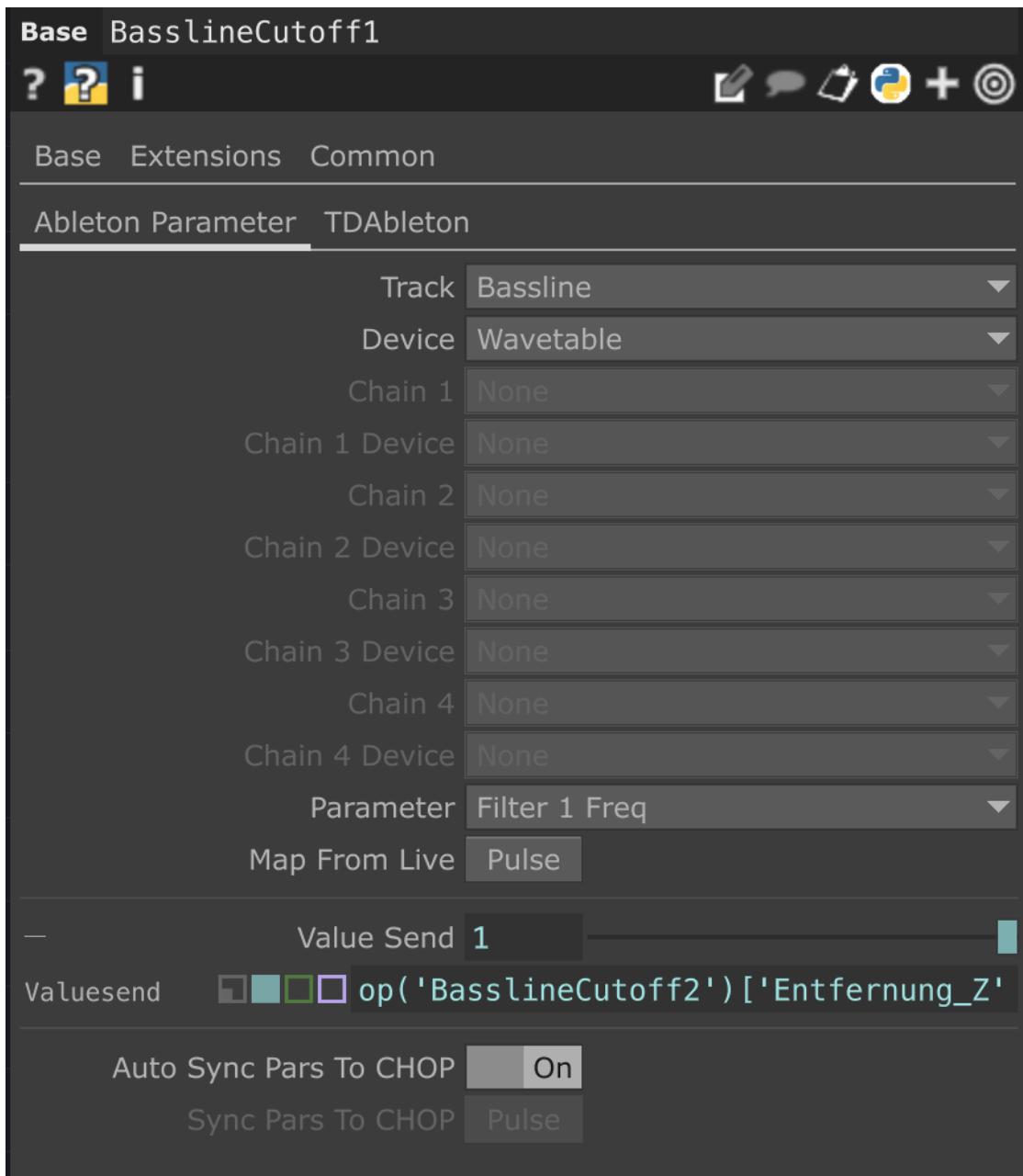
Annahmen für die folgende Anleitung

1. Die Ableton Session befindet sich nicht auf dem gleichen Rechner, wie die TouchDesigner Session.
2. Die MIDI-Remote Skripte für TouchDesigner sind installiert. (Hierfür gibt es eine [Anleitung von Derivative](#) unter dem Abschnitt **Install the latest TD Ableton System**)

6.2 Anleitung

1. Stelle sicher, dass beide Rechner im selben Netzwerk sind.
 - **Wichtig:** Im eduroam Netz ist das Setup nicht möglich, da die Kommunikation zwischen den beiden Rechnern unterbunden wird.
 - Erstelle im Zweifel ein lokales Netz (Hotspot) und verbinde beide Rechner damit.

2. Notiere dir die IP-Adresse des Rechners, auf dem das Ableton Projekt laufen soll.
3. **Wähle die Komponente “Ableton OSC” in dem TouchDesigner Projekt aus**
 - In dem TouchDesigner Projekt befindet sich auf der rechten Seite ein Node mit dem Name “Ableton OSC”. Wähle diesen aus und navigiere zu dem Reiter “TD Ableton”
 - Trage die vorher notierte IP-Adresse in das Eingabefeld “Ableton IP Adress” ein
 - Die Ports müssen nicht verändert werden, solange du diese nicht bereits für ein anderes System/Projekt verwendest.
4. **Starte das Ableton Live Projekt**
 - Das Projekt befindet sich im Ordner **sounddesign**
 - Stelle sicher, dass die MIDI Remote Skripte installiert sind.
 - Sollten die TD Ableton Devices nicht gefunden werden, kannst du den Pfad manuell hinzufügen. Anleitung von Ableton: [How to find and replace missing media files – Ableton](#)
5. **Wähle den Master Track des Ableton Projekts aus**
 - Nun siehst du ein **TDAMaster Device**.
 - Du solltest auf dem TDAMaster Device nun “connected” oder “disconnected” sehen können.
 - Wenn keine Verbindung hergestellt werden kann, siehe Troubleshooting von Derivative [TD-Ableton - Troubleshooting](#)
6. **Öffne wieder die TouchDesigner Session**
 - Wenn alles geklappt hat, zeigt die Ableton OSC Node nun **1 connected** an.
 - Um zu überprüfen, ob die bereits gemappten Ableton Parameter richtig erkannt werden, wähle einen der Parameter aus.



Exemplarisch wird hier die Filterfrequenz eines Wavetable Synthesizers gesteuert.

- Hier ist ein funktionierendes Beispiel zu sehen. Über das Dropdown Menü “Parameter” können sämtliche Parameter des Wavetable Devices ausgewählt werden.
- Wenn du den Regler neben “Value Send” bewegst, sollte sich der ausgewählte Parameter in Ableton Live verändern.

6.3 Erweitern der Ableton Live Steuerung

Wenn du weitere Parameter des Ableton Live Projekts durch einen TouchDesigner Operator steuern möchtest, lies die folgende Kurzanleitung.

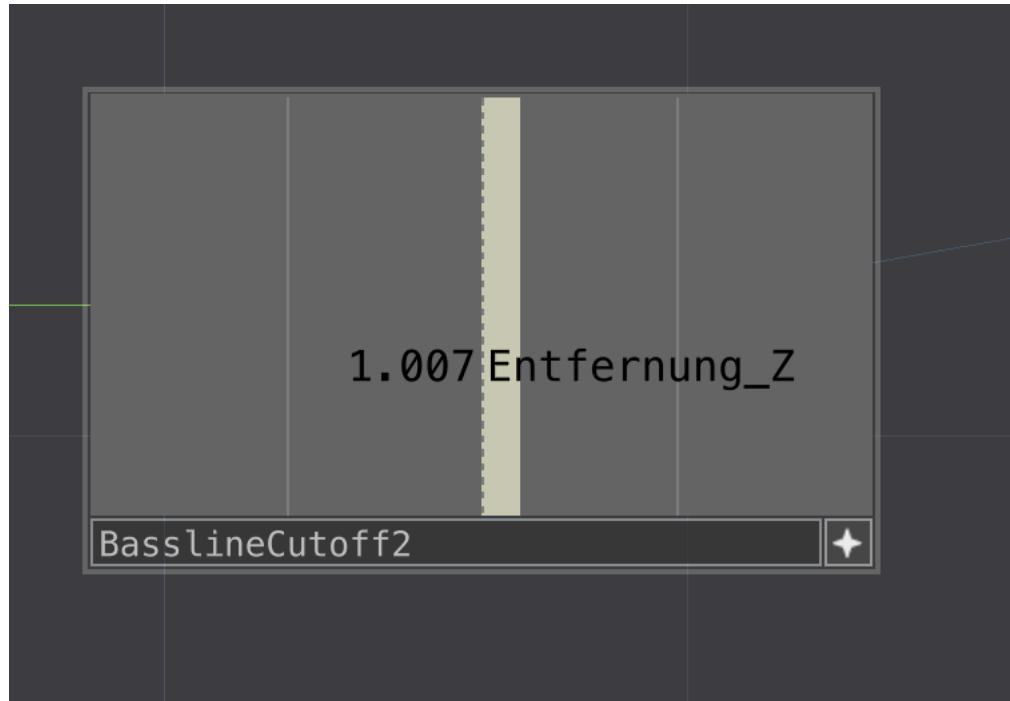
TouchDesigner -> Ableton Workflow

1. Wähle einen TouchDesigner Operator aus
 - CHOPs eignen sich hierfür sehr gut **CHOP - Derivative**
2. Füge einen **abletonParameter** im Projekt hinzu

- Um z.B. einen Parameter eines Synthesizers steuern zu können, benötigt man die TouchDesigner Komponente **abletonParameter**. Diese Komponente findet man unter **Palette > TD Ableton > Live 9 & 10 > abletonParameter**
- Wenn die Palette nicht angezeigt wird, wähle in der Menüleiste **Dialogs > Palette Browser** aus.
- Wähle im Menü des Ableton Parameters (siehe voriges Beispiel) den Track des Synthesizers, das zu steuernde Device, und den Parameter (z.B. Filter 1 Freq) aus.

3. TouchDesigner Operator verbinden mit Parameter

- Um einen CHOP mit dem Parameter verbinden zu können, klicke das sternförmige Icon unten rechts an



Sternförmiges Icon unten rechts sichtbar

- Wenn du nun über den CHOP hoverst, sollte der Cursor sich zu einem “Dach” ändern.
- Klicke auf den CHOP, wenn das Dach erscheint und halte die Taste gedrückt.
- Ziehe die Maus auf den **Value Send** Reiter des Ableton Parameters, und lasse die Taste los.
- Wähle in dem erscheinenden Menü **CHOP Reference**
- Fertig!** Der Ableton Parameter wird nun von dem CHOP gesteuert.

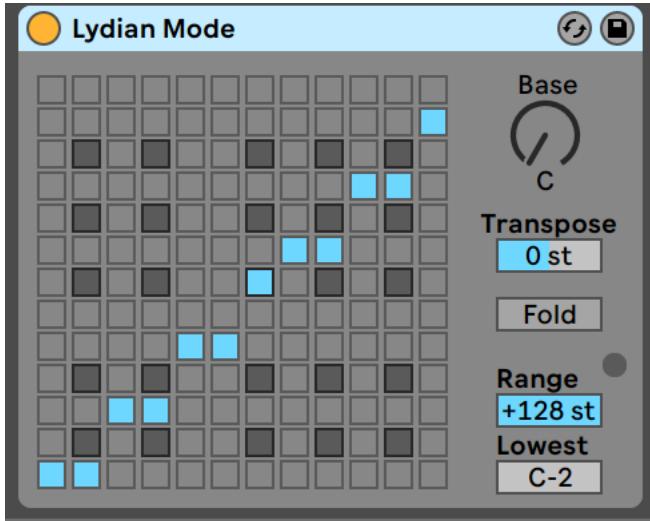
4. Bemerkungen

- Um brauchbare Ergebnisse zu erzielen, müssen die Werte in TouchDesigner meistens erst normalisiert und auf einen passenden Wertebereich angepasst werden. Beispielsweise die Positionsdaten, die die Kinect liefert. Im Projekt sind einige solcher Umformungen von uns vorgenommen worden, lass dich hier gern inspirieren!
- Wenn ein neuer Track in Ableton hinzugefügt wird, stelle sicher, dass du diesen umbenennst. Ableton fügt beim Erstellen eines Tracks eine Raute vor dem Namen ein. Dies kann zu Problemen mit TouchDesigner führen!

6.4 Sounddesign

- Sämtliche Sounds wurden mit Ableton Instrumenten und Effekten erzeugt. Samples wurden nicht verwendet. Du kannst also einfach loslegen und mit dem Projekt experimentieren, solange du Ableton Live 10 aufwärts installiert hast.

- Wenn du nicht an der TouchDesigner Steuerung interessiert bist, lösche einfach die TD Ableton Instrumente auf den einzelnen Spuren.
- Damit man sich auf das Sounddesign konzentrieren kann, liegt jeweils ein **Scale MIDI-Effekt** auf den MIDI Spuren. So braucht man sich keine Gedanken um Harmonien machen. - Wenn du dies beibehalten möchtest, kopiere folgenden Effekt auf deine neue Synthesizer Spur



Ansicht des MIDI Scale Effekts. Zu finden auf allen Synthesizer Spuren

- Die einzelnen Sounddesign Patches werden nachfolgend kurz präsentiert, jedoch nicht tiefer erläutert. Es empfiehlt sich, die Ableton Instrumente zu öffnen und die Effekte zu deaktivieren, um zu sehen, wie die Sounds erzeugt wurden. Wenn du tiefer in die Materie einsteigen möchtest, starte mit dem Bassline Patch. Dieser besteht aus einem Wavetable Synthesizer und ein paar Effekten. Ein guter Einstiegspunkt ist die Filtersektion. Hier wird ein Filter mit zwei LFOs moduliert. Verändere die LFOs und höre dir an, wie sich der Sound verändert. Viel Spaß beim Experimentieren!

Bassline

- Der Bassline Sound ist ein Wavetable Synthesizer, der mit einem Filter und zwei LFOs moduliert wird.
- Die Position der Waveform wird ebenfalls von einem LFO moduliert.
- Das Cutoff des Filters wird von der Tiefenposition des Oberkörpers, die von der Kinect erfasst wird, gesteuert. (Näher an der Kinect = höherer Cutoff)
- Es wird ein FM Effekt verwendet, der von der Y-Position der linken Hand gesteuert wird. (Höher = stärkere FM-Modulation)

Arpeggio

- Der Arpeggio Sound wird von einem Operator erzeugt.
- Eine Sinuswelle wird von einer Sägezahnwelle und einer weiteren Sinuswelle in unterschiedlichen Oktaven moduliert.
- Das Cuttoff des Filters wird von der Tiefenposition des Oberkörpers, die von der Kinect erfasst wird, gesteuert. (Näher an der Kinect = höherer Cutoff)
- Eine Hüllkurve mit schnellem Attack und Decay wird auf das Filter gelegt, um den Sound perkussiv zu gestalten.

Pad

- Der Pad Sound besteht aus mehreren Layern.
- Zwei Wavetable Synthesizer und ein Physical Modelling Synthesizer (Collision) wurden verwendet.

- Die Filter der Synthesizer wird ebenfalls von der Tiefenposition des Oberkörpers, die von der Kinect erfasst wird, gesteuert. (Näher an der Kinect = höherer Cutoff)
- Ein Wavetable Synthesizer erzeugt warme, weiche Flächen durch eine Sägezahnwelle und eine Whitenoise Textur. Die Sägezahnwelle wirkt durch Detuning breiter und interessanter. Als subtilen Effekt wird ein Corpus Effekt verwendet, der interessante Resonanzen erzeugt. Die Waveform wird von einem LFO mit sehr niedriger Frequenz moduliert.
- Der zweite Wavetable Synthesizer erzeugt einen helleren Sound, realisiert durch eine Wellenform mit mehr Obertönen. Der Effekt wird vom Spieler durch die Tiefenposition des Oberkörpers gesteuert.
- Der Collision Synthesizer besitzt Einstellungsmöglichkeiten für interessante Noise Texturen und Resonanzen. Der Noise Amount wird ebenfalls von der Tiefenposition des Oberkörpers gesteuert. So entsteht ein sehr dichtes Klangbild, das sich mit der Position des "Spielers" verändert.
- Gemeinsam erzeugen die Synthesizer einen sehr dichten, warmen Sound, der als klangliches Fundament für die anderen Sounds dient und mit der Position des "Spielers" interagiert.

SFX

- Der perkussive SFX Sound wird von einem Operator erzeugt.
- Eine Sinuswelle wird von einer weiteren Sinuswelle sowie einer White Noise Textur, jeweils in unterschiedlichen Oktaven, moduliert.
- Ein Tiefpassfilter wird von einem Sample&Hold LFO im 1/4 Takt moduliert. So entsteht ein rhythmischer Effekt. Der Sample&Hold LFO steuert ebenfalls die Oszillatoren, um ständig neue Klänge zu erzeugen.
- Mit Annäherung an die Kinect wird die Lautstärke des SFX erhöht. (Näher an der Kinect = lauter)

7. Probleme

Im Laufe des Projekts sind wir auf einige Probleme gestoßen, die wir hier kurz auflisten möchten.

1. **OSC Plugins - Fluch und Segen** Die Plugins, die wir für die Kommunikation zwischen Ableton und TouchDesigner verwendet haben, sind sehr mächtig. Jedoch erzeugen Sie eine Abstraktionsschicht, die es manchmal schwierig macht, Fehler zu finden. Wir haben uns bei Verbindungsproblemen oft gefragt, ob ein Fehler in Ableton/, TouchDesigner oder dem Plugin liegt. Das trifft auch auf das NodeOSC Plugin zu, das wir für die Kommunikation zwischen TouchDesigner und Blender verwendet haben. Das Format der Pfade, sowie die Position der Variablen war anfangs nicht klar.
2. **Quaternions** Die Rotation der Szenenkamera in Blender wird realisiert durch Quaternions. Da die Kinect keine Rotationsdaten einzelner Gelenke liefert, haben wir die Positionsdaten der rechten Hand verwendet. Diese wurden dann in den Wertebereich von -1 bis 1 normalisiert und an Blender gesendet. Dies hat zu Problemen geführt, da das Verhalten der Quaternions für uns nicht immer vorhersehbar war. Die Ergebnisse mit Euler Winkeln waren jedoch gravierend schlechter, also behielten wir die Quaternion-Lösung bei. Unter dem Strich haben wir viel Zeit damit verbracht, die Kamera zu debuggen, sind jedoch mit dem Ergebnis zufrieden.
3. **Projection Mapping** In einem sehr frühen Stadium unseres Projektes haben wir in Erwägung gezogen ein Projection Mapping zu realisieren. Wir mussten leider feststellen, dass die hierfür benötigten Tools finanziell nicht tragbar oder nicht ausgereift genug waren. Wir haben uns dann entschieden, die Projektion auf eine Leinwand zu beschränken. So konnten wir uns auf die Interaktion und das Sounddesign konzentrieren.