

Bearbeitungsbeginn: 01.09.2014

Vorgelegt am: TBA

# Thesis

zur Erlangung des Grades

**Master of Science**

im Studiengang Medieninformatik

an der Fakultät Digitale Medien

***Dominik Steffen***

***Matrikelnummer: 245857***

**TODO: Code Content Verschränkung im Bezug auf 3D Game Engines.**

*Erstbetreuer:* Prof. Christoph Müller

*Zweitbetreuer:* Prof. Dr. Wolfgang Taube



# Abstract

---

Arbeitsprozesse in heutigen Game Engines verlangen von Entwicklern meist das Erlernen neuer Tools und das während eines meist zeitlich sehr beschränkten Projektes. Es wäre für Entwickler einfacher sich mit den bekannten Tools zu beschäftigen und mit diesen großartige Ergebnisse zu erreichen. Designer müssen sich oft in unbekannte Editoren und SDKs einarbeiten während Entwickler sich in Grafische Editoren einarbeiten müssen um ihren Code an der richtigen Stelle des Projekts zu platzieren. Diese Arbeit baut eine Brücke zwischen beiden Welten. Durch die Konzeption und Umsetzung eines Software Tools wird eine Trennung der Abhängigkeiten in einem Projekt erreicht. Mit Hilfe eines Plugins ist es möglich, dass Designer oder Entwickler jederzeit in die Entwicklung eines Projektes einsteigen. Es wird ermöglicht mit Cinema 4D und einer IDE wie VS2013 an einem Projekt mit der FUSEE Engine zu arbeiten ohne die bereits bekannte Welt zu verlassen. Ein Projekt managed sich durch die Nutzung des entstandenen Plugins selbst. Das zuerst Konzeptionell visualisierte Tool wurde während dieser Arbeit umgesetzt und bietet ausreichende Basisfunktionalität um ein Projekt zu erstellen. Hierzu wurden verschiedene Konzepte betrachtet und andere Game Engines auf Workflow und Anwendbarkeit untersucht. Es wurden einige Kernkonzepte erkannt und für eine Implementierung in FUSEE analysiert und weiter entwickelt.



# Eidesstattliche Erklärung

---

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Masterthesis selbstständig und ohne unzulässige fremde Hilfe angefertigt habe. Alle verwendeten Quellen und Hilfsmittel die sowohl zum schreiben dieser Arbeit als auch zum Entwickeln des dazugehörigen Sourcecodes benutzt wurden, habe ich angegeben.

---

Dominik Steffen, Küssaberg den 10. November 2014



"Hier steht ein wichtiges Zitat zur Entstehung dieser Arbeit."

- TBD.

Kontakt:

Dominik Steffen (Matr.-Nr.: 245857)

Hochschule Furtwangen

E-Mail:

dominik.steffen@hs-furtwangen.de, dominik.steffen@gmail.com





# Inhaltsverzeichnis

---

<b>1</b>	<b>Anforderungen, Ziele und eine Fragestellung</b>	<b>1</b>
1.1	Entwicklungsprozesse in Games . . . . .	1
1.1.1	Mitglieder eines Entwicklerteams . . . . .	1
1.2	Ein Arbeitsprozess wird entwickelt . . . . .	1
1.2.1	Game Authoring . . . . .	1
1.2.2	Tool Development . . . . .	1
1.2.3	Warum eine Trennung von Code und Content? . . . . .	1
<b>2</b>	<b>Entwicklung eines Konzeptes</b>	<b>2</b>
2.1	Use Cases der verschiedenen Entwickler . . . . .	3
2.1.1	Was möchten Designer? . . . . .	3
2.1.2	Was möchten Entwickler? . . . . .	3
2.2	Aktuelle Engines und deren Arbeitsprozesse . . . . .	3
2.2.1	Prozesse in Game Engines und einem Framework . . . . .	3
2.2.2	Unreal Engine 4 . . . . .	3
2.2.3	Unity 3D . . . . .	3
2.2.4	Android SDK . . . . .	3
2.3	Konzeptentwurf . . . . .	3
2.3.1	Entfernen von Abhängigkeiten . . . . .	3
2.3.2	Zeitersparnis durch bekannte Tools . . . . .	3
2.3.3	Warum Fusee und Cinema 4D? . . . . .	3
2.4	Die Implementierung . . . . .	3
2.4.1	Cinema 4D Plugin API und SDK . . . . .	3
2.4.2	Fusee . . . . .	3
2.5	Das eigentliche Plugin . . . . .	3
2.5.1	Generieren eines Fusee Projektes . . . . .	3
2.5.2	Code Generation und die Vermeidung von Roundtrips (nicht so ganz roundtrips, generierung um generierung etc.) . . . . .	3

2.5.3	XPresso Schaltungen - Programmieren ohne Programmieren . . . . .	3
<b>3</b>	<b>Ergebnisse und Erkenntnisse</b>	<b>4</b>
3.1	Game Authoring Entwicklungsprozesse jetzt und in Zukunft . .	4
3.2	Wie weit ist die Implementierung fortgeschritten? . . . . .	4
3.3	Welcher Mehrwert wurde erreicht? . . . . .	4
3.4	Integration des Systems in den weiteren Projektverlauf von FU-SEE . . . . .	4
	<b>Verzeichnis der Sourcecode Beispiele</b>	<b>6</b>
	<b>Tabellenverzeichnis</b>	<b>7</b>
	<b>Abbildungsverzeichnis</b>	<b>8</b>
	<b>Literaturverzeichnis</b>	<b>9</b>
	<b>UML Diagramme</b>	<b>10</b>



# 1 Anforderungen, Ziele und eine Fragestellung

---

## 1.1 Entwicklungsprozesse in Games

### 1.1.1 Mitglieder eines Entwicklerteams

Designer

Entwickler

## 1.2 Ein Arbeitsprozess wird entwickelt

### 1.2.1 Game Authoring

### 1.2.2 Tool Development

### 1.2.3 Warum eine Trennung von Code und Content?



## 2 Entwicklung eines Konzeptes

---

### 2.1 Use Cases der verschiedenen Entwickler

#### 2.1.1 Was möchten Designer?

#### 2.1.2 Was möchten Entwickler?

### 2.2 Aktuelle Engines und deren Arbeitsprozesse

#### 2.2.1 Prozesse in Game Engines und einem Framework

#### 2.2.2 Unreal Engine 4

#### 2.2.3 Unity 3D

#### 2.2.4 Android SDK

### 2.3 Konzeptentwurf

#### 2.3.1 Entfernen von Abhängigkeiten

#### 2.3.2 Zeitersparnis durch bekannte Tools

#### 2.3.3 Warum Fusee und Cinema 4D?

### 2.4 Die Implementierung

#### 2.4.1 Cinema 4D Plugin API und SDK

#### Uniplug

#### 2.4.2 Fusee

#### Der Fusee Szenengraph

### 2.5 Das eigentliche Plugin

#### 2.5.1 Generieren eines Fusee Projektes

#### 2.5.2 Code Generation und die Vermeidung von Roundtrips (nicht so ganz roundtrips, generierung um generierung etc.)

#### 2.5.3 XPresso Schaltungen - Programmieren ohne Programmieren<sub>3</sub>

## 3 Ergebnisse und Erkenntnisse

---

3.1 Game Authoring Entwicklungsprozesse jetzt und in Zukunft

3.2 Wie weit ist die Implementierung fortgeschritten?

3.3 Welcher Mehrwert wurde erreicht?

3.4 Integration des Systems in den weiteren Projektverlauf von FUSEE

# Anhang



## Verzeichnis der Sourcecode Beispiele

---

## Tabellenverzeichnis

---

## Abbildungsverzeichnis

---

## Literaturverzeichnis

---

- [1] Michael Abrash. *Graphics Programming Black Book*. TBD, 1997.
- [2] E Gamma, R Johnson, R Helm, and J Vlissides. *Design Patterns: Elements of REusable Object-Oriented Software*. Pearson Deutschland, 2014.
- [3] Jason Gergory. *Game Engine Architecture*. Taylor & Francis Ltd., 2009.
- [4] Eric Lengyel. *Mathematics for 3D Game Programming and Computer Graphics*. Course Technology, Cengage Learning, 2012.
- [5] Graham Wihlidal. *Game Engine Toolset Development*. Thomson Course Technologies, 2006.

## UML Diagramme

---