HOCHSCHULE FURTWANGEN UNIVERSITY | HFU

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 Toolsets und das während eines meist zeitlich sehr beschränkten Projektes. Es wäre für Entwickler einfacher sich mit den bereits 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 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 mit ihren eigenen Tools 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 Engine Projektstruktur managed sich durch die Nutzung des entstandenen Plugins selbst. Das zuerst Konzeptionell entworfene Tool wurde während dieser Arbeit umgesetzt und bietet ausreichende Basisfunktionalität um ein Projekt zu erstellen. Hierzu wurden verschiedene Konzepte betrachetet 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 selbständig 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 8. Dezember 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

1 Anforderungen, Ziele und eine Fragestellung

1.1 Entwicklungsprozesse in Games

1.1.1 Projektmanagement Modelle

1.1.2 Tool Developing anstelle von kaufen

1.1.3 Mitglieder eines Entwicklerteams

Artists

Designer

Entwickler

1.1.4 Weitere für diese Arbeit nicht relevante

1.2 Stakeholderanalyse intern

Warum wieso, wie führt uns das an das Thema heran

1.3 Ein Arbeitsprozess wird entwickelt

1.3.1 Game Authoring / Game Development

Was ist das, was beschreibt es, wieso ist es hier relevant?

1.3.2 Tool Development

Nach Wihlidal ... Warum wurde oben geklärt, hier den Prozess erläutern. Möglichkeiten vorstellen. Beispiele geben.

1.3.3 Planung

- Eine Software soll es ermöglichen, dass Artists, Designer und Developer an ein und dem selben Projekt arbeiten können ohne die gewohnte Ar-

beitsumgebung (3D-Modellierungssoftware, IDE) zu verlassen und etwas komplett neues (Level-Editor) zu erlernen.

- Die komplette Entwicklung und der Entwurf bedienen sich der FUSEE Engine.
- Ziel ist es in Cinema 4D ein FUSEE Projekt anzulegen, zu speichern und es zu öffnen (In Form einer Szene, Level, Welt.). Weiterhin müssen Ässetsïns Spiel integriert werden können die von Artists, Designern und Entwicklern bearbeitet werden können.
- Das Projekt sollte aus C4D heraus gebaut werden können.

1.3.4 Analyse

- Eine Stakeholderanalyse schafft klarheit, welche Parteien mit dem zu erstellenden Tool arbeiten müssen.
- Die Game Engines sollten Kategorisiert werden. Hierzu ist ein "Gitternetzäufzustellen.
- Verschiedene Game Engines müssen darauf untersucht werden, wie die Arbeitsschritte in diesen zu erledigen sind. Jede Untersuchung sollte dokumentiert werden.
- Es ist zuerst zu analysieren, welche Schritte für welche Art der Arbeit notwendig sind. Hierzu werden Usecases der verschiedenen Rollen und Aufgaben erstellt.

1.3.5 Design

- Aufgrund der erstellten Use Cases wird ein System Design erstellt.
- Das Systemdesing sollte Roundtrips vermeiden
- Hierzu können Partial Classes untersucht werden.
- Es soll möglichst wenig "geparsedöder "konvertiert"werden
- Dateien sollen Version Control kompatibel bleiben (wenig bis keine Binarys)
- Das Projekt muss Strukturiert sein
- Eine Fusee Solution soll gehandelt werden können

1.3.6 Implementierung

- Die Software wird auf basis der FUSEE Engine und Cinema4D R16 erstellt.

Asset Pipeline und Feedback

SEHR KRITISCHE MARKE! Das Asset aus dem Editor in die Engine bekommen, die Darstellung etc kontrollieren. Zeitkritisch beim export etc. Carter Seite 6 und folgende.

Die Struktur von Game Assets

Assets und das aufbrechen in Bestandteile eines Projektes. Level etc. bestehen aus Assets und mehr.

1.3.7 Warum eine Trennung von Code und Content?

2 Entwicklung eines Konzeptes

2.1 Use Cases der verschiedenen Entwickler

2.1.1 Was möchten Artists?

2.1.2 Was möchten Designer?

2.1.3 Was möchten Entwickler?

2.1.4 Projekt bezogen

Projekt anlegen

Projekt öffnen

Projekt speichern

Projekt bauen

In Projekt einsteigen

Projekt clonen etc.

2.1.5 Prozess bezogen

Gleichzeitig an Projekt arbeiten

Model Datei importieren

Gleichzeitig an einem Objekt arbeiten

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 idTech X

2.2.5 Weitere

2.3 Konzeptentwurf

2.3.1 Systemdesign für ein Plugin

2.3.2 Systemdesign für einen Project-Handler

Irgend etwas dass im Zentrum steht. Alles andere muss auf Level etc aufgeteilt werden und bis zum einzelnen Asset heruntergebrochen werden.

2.5 Das eigentliche Plugin

2.5.1 Visualisierung der Systemarchitektur

Welche Programme und Systeme sind beteiligt?

2.5.2 Generieren eines Fusee Projektes

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

2.5.4 XPresso Schaltungen - Programmieren ohne Programmieren

2.5.5 Partial Classes in .NET

3 Ergebnisse und Erkentnisse

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] Ben Carter. *The Game Asset Pipeline*. Delmar Cengage Learning, 2004.

[3] Heather Chandler. *The Game Production Handbook*. Thomson Delmar Learning, 2006.

[4] E Gamma, R Johnson, R Helm, and J Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Deutschland, 2014.

[5] Jason Gergory. *Game Engine Architecture*. Taylor & Francis Ltd., 2009.

[6] Eric Lengyel. *Mathematics for 3D Game Programming and Computer Graphics*. Course Technology, Cengage Learning, 2012.

[7] Graham Wihlidal. *Game Engine Toolset Development*. Thomson Course Technologies, 2006.

UML Diagramme