|  |
| --- |
| Unreal Cup  **STUDIENARBEIT**    im Studiengang Informationstechnik  an der DHBW Ravensburg  Campus Friedrichshafen  von  Michael Keckeisen,  Michael Möbius,  Daniel Rapp,  Maximilian Schmitz  Abgabedatum  Bearbeitungszeitraum xx.xx.2014 – yy.yy.2015  Matrikelnummer Matrikelnummer Michael K,  Matrikelnummer Michael M,  Matrikelnummer Daniel,  9322432      Betreuer Titel Vorname Nachname |

|  |
| --- |
|  |

# Eidesstattliche Erklärung

gemäß § 5 (3) der „Studien- und Prüfungsordnung DHBW Technik“ vom   
22. September 2011.

Hiermit erkläre ich, dass ich die vorliegende Arbeit mit dem Titel

**Unreal Cup**

selbständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt, keine anderen als die angegebenen Hilfsmittel benutzt und wörtliche sowie sinngemäße Zitate als solche gekennzeichnet habe.

xx, den xx.xx.2015

Unterschrift

# Kurzfassung

# Abstract

# Inhalt

[Eidesstattliche Erklärung I](#_Toc406781336)

[Kurzfassung II](#_Toc406781337)

[Abstract III](#_Toc406781338)

[Inhalt IV](#_Toc406781339)

[1 Einleitung 1](#_Toc406781340)

[2 Aufgabenstellung 2](#_Toc406781341)

[3 Stand der Technik 3](#_Toc406781342)

[3.1 Einleitung 3](#_Toc406781343)

[3.2 Unreal Engine 4 3](#_Toc406781344)

[3.3 LUA Script 3](#_Toc406781345)

[3.4 Robo Cup 3](#_Toc406781346)

[3.5 Künstliche Intelligenz 3](#_Toc406781347)

[3.6 Zusammenfassung 3](#_Toc406781348)

[4 Planung / Struktur 4](#_Toc406781349)

[4.1 Projektmanagement 4](#_Toc406781350)

[4.2 Analyse der Aufgabenstellung 4](#_Toc406781351)

[4.3 Architektur 4](#_Toc406781352)

[5 Umsetzung 5](#_Toc406781353)

[5.1 Generell 5](#_Toc406781354)

[5.2 Spielfeld 5](#_Toc406781355)

[5.3 Funktionen über LUA Script 5](#_Toc406781356)

[5.4 Interne Umsetzung 5](#_Toc406781357)

[6 Ausblick 6](#_Toc406781358)

[7 Fazit 7](#_Toc406781359)

[Literaturverzeichnis I](#_Toc406781360)

[Abbildungsverzeichnis II](#_Toc406781361)

[Abkürzungsverzeichnis III](#_Toc406781362)

[Glossar IV](#_Toc406781363)

# Einleitung

# Aufgabenstellung

# Stand der Technik

## Einleitung

Die Arbeit baut auf unterschiedlichen Technologien auf, deren Verständnis für die Projektbearbeitung unumgänglich ist. Da die Simulation, wie sie in Kapitel 2 beschrieben wurde, nicht im gegebenen Zeitrahmen von Grund auf Programmiert werden kann, muss eine Game-Engine gewählt werden, die Grundlagen wie die 3D-Animation oder die Physik-Berechnungen bereitstellt.

Des Weiteren muss eine Technologie gewählt werden, die die Steuerung der Charaktere über im Voraus erstellte Abläufe erlaubt.

In diesem Kapitel sollen die Technologien, die für die genannten Zwecke in Betracht gezogen wurden, kurz dargestellt werden und die Technologieentscheidung begründet werden

## Game Engine

http://de.wikipedia.org/wiki/Liste\_von\_Spiel-Engines

### Grundsätzliches

Wie in Kapitel 3.1 angesprochen ist eine Game-Engine notwendig, die die Grundfunktionalitäten einer 3D-Simulation bereitstellt. Es existieren heutzutage mehrere solcher Frameworks, auf die im Folgenden kurz eingegangen werden soll.

### Source Engine

Source Engine wird von Valve entwickelt und ist durch die weit verbreiteten Spiele, die mit ihr Entwickelt wurden, bekannt. Dazu gehört neben Half Life 2 oder Counter Strike: Source auch Portal.

Die Source Engine kann neben den Betriebssystemen Windows, Linux und MAC auch auf der Playstation 3 oder der Xbox360 verwendet werden.

Neben der Grafik-Engine beinhaltet dieses Framework auch eine Physik-Engine.

Ein Problem mit der Source Engine ist, dass sie nicht frei verwendet werden kann sondern kostenpflichtig lizensiert werden muss.

### Unity

<http://de.wikipedia.org/wiki/Unity_%28Spiel-Engine%29>

http://unity3d.com/unity/licenses

Die Engine Unity wird von Unity Technologies entwickelt und unterstützt, wie die Source Engine, sowohl Grafik als auch Physikberechnungen. Ein großer Vorteil von Unity ist die große Anzahl der unterstützten Plattformen. Neben den Betriebssystemen Windows, Mac und Linux und den bekanntesten Konsolen (Playstation 3, Xbox360, Nintendo Wii) unterstützt Unity auch Smartphones (iOS, Android, Windows Phone) und weitere Plattformen.

Unity bietet eine Gratisversion und eine Pro-Version an, die für 1500$ erworben werden kann.

### Unreal Engine 4

https://www.unrealengine.com/what-is-unreal-engine-4

Unreal Engine wird von Epic Games entwickelt und ist auf allen verbreiteten Computer-Betriebssystemen lauffähig. Neben diesen unterstützt es allerdings auch die neue Konsolengeneration mit Playstation 4 und Xbox One, sowie mobile Anwendungen mit iOS oder Android.

Seit der erstmaligen Verwendung der Unreal Engine im Computerspiel „Unreal“ im Jahr 1998 konnten viele, teilweise sehr bekannte Spiele erfolgreich auf der Engine aufbauen. Die Unreal Engine 4, das aktuellste Framework der Reihe, wurde in Spielen wie Tekken 7 oder Kingdom Hearts 3 eingesetzt.

Lizensiert wird die Engine normalerweise für 19€ pro Monat, ein großer Vorteil der Engine ist allerdings die Tatsache, dass sie Studenten und Hochschulen kostenlos zur Verfügung gestellt wird.

### Verwendung im Projekt

Da die erwähnten Frameworks sich im Funktionsumfang kaum bis überhaupt nicht unterscheiden war die Auswahl der geeigneten Engine nicht ohne weiteres möglich.

Eines der Hauptmerkmale, die für die Entscheidung relevant waren, war der Preis und die Lizensierung des Produkts.  
Durch diese Betrachtung wurde schnell klar, dass die Source Engine für dieses Projekt nicht in Frage kommt, da hier keine Möglichkeit zur freien Verwendung existiert.  
  
Ein weiterer wichtiger Punkt war die Erweiterbarkeit des Codes zum Einbinden der externen KI. Da ein solcher Ansatz natürlich nicht den Anforderungen an normale Spiele entspricht, bietet keine der von uns in Betracht gezogenen Frameworks eine vorgefertigte Lösung für dieses Problem.  
<TODO>

Aus den oben genannten Gründen fiel die Wahl auf die Unreal Engine 4.

<TODO>

## Scripting-Ansätze

### Grundsätzliches

Da die Steuerung der Spieler, wie in der Aufgabenstellung erwähnt, von außerhalb der Simulation stattfindet, muss ein geeigneter Ansatz gewählt werden, über den eine programmierte KI in die Simulation eingespeist werden kann.

Hierbei fiel die Wahl auf eine Scriptsprache, da diese typischerweise leichtgewichtig und einfach zu erlernen sind.

### Lua

https://en.wikipedia.org/wiki/Lua\_%28programming\_language%29

Lua ist eine leichtgewichtige Scriptsprache und kann auf verschiedenen Plattformen eingesetzt werden. Dabei bietet Lua eine Menge von Basisfunktionen die zur Behandlung eines Problems erweitert werden können. Da Lua leicht einzubinden und zu erlernen ist wird die Sprache bereits in vielen Programmen, unter anderem in Videospielen wie Garrys Mod, eingesetzt.

Die Einbindung von Lua in den Code der Simulation <TODO>

## Robo Cup

## Künstliche Intelligenz

## Zusammenfassung

# Planung / Struktur

## Projektmanagement

## Analyse der Aufgabenstellung

## Architektur

# Umsetzung

## Prototypenentwicklung

### Begründung

Aufgrund der geringen Erfahrung des Teams zum Thema Spieleentwicklung war es wichtig, dass eine Einarbeitungsphase in die Technologien und die Materie stattfinden konnte.  
Aus diesem Grund wurde die Entscheidung getroffen, Teile des fünften Semesters zur Prototypenentwicklung aufzuwenden.

Zu dieser Entwicklung wurde eine grobe, grundlegende Architektur angelegt, die im Laufe des Semesters mit der gesammelten Erfahrung überarbeitet und verbessert werden sollte. Da diese Architektur eher temporärer, veränderlicher Natur war soll sie hier nicht dargestellt werden.

Der Prototyp dient neben der Bereitstellung einer Lernphase auch zur Implementierung der Grundfunktionen, die später mit geringerem Aufwand in die endgültige Version übernommen werden können.

Mit der gesammelten Erfahrung aus dem Prototypen und der so entwickelten Architektur, die in Kapitel <TODO> erläutert wird, sollte nun im sechsten Semester eine endgültige Softwareversion entwickelt werden.

### Umfang

Der Prototyp beinhaltet, wie bereits erwähnt, die Grundfunktionalitäten auf denen die spätere Simulation aufbaut. Dazu gehört neben dem Aufbau des Spielfeldes und dem Einbinden der Lua-Schnittstelle auch die Entwicklung der Basisfunktionen der Spieler, wie zum Beispiel die Implementierung eines Sichtfelds oder das Bewegen zu einer bestimmtem Position auf dem Spielfeld.

## Generell

## Spielfeld

## Funktionen über LUA Script

## Interne Umsetzung

# Ausblick

# Fazit

# Literaturverzeichnis

**Im aktuellen Dokument sind keine Quellen vorhanden.**

# Abbildungsverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

# Abkürzungsverzeichnis

# Glossar