|  |
| --- |
| Unreal Cup  **STUDIENARBEIT**    im Studiengang Informationstechnik  an der DHBW Ravensburg  Campus Friedrichshafen  von  Michael Kekeisen,  Michael Möbius,  Daniel Rapp,  Maximilian Schmitz  Abgabedatum  Bearbeitungszeitraum xx.xx.2014 – yy.yy.2015  Matrikelnummer Matrikelnummer Michael K,  2459628,  Matrikelnummer Daniel,  9322432      Betreuer Prof. Erwin Fahr |

|  |
| --- |
|  |

# 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](#_Toc407443526)

[Kurzfassung II](#_Toc407443527)

[Abstract III](#_Toc407443528)

[Inhalt IV](#_Toc407443529)

[1 Einleitung 1](#_Toc407443530)

[2 Aufgabenstellung 2](#_Toc407443531)

[3 Stand der Technik 3](#_Toc407443532)

[3.1 Einleitung 3](#_Toc407443533)

[3.2 Unreal Engine 4 3](#_Toc407443534)

[3.3 LUA Script 3](#_Toc407443535)

[3.4 Robo Cup 3](#_Toc407443536)

[3.5 Künstliche Intelligenz 3](#_Toc407443537)

[3.6 Zusammenfassung 3](#_Toc407443538)

[4 Planung / Struktur 4](#_Toc407443539)

[4.1 Projektmanagement 4](#_Toc407443540)

[4.2 Analyse der Aufgabenstellung 4](#_Toc407443541)

[4.2.1 Muss-Kriterien 4](#_Toc407443542)

[4.2.2 Soll-Kriterien 5](#_Toc407443543)

[4.2.3 Kann-Kriterien 5](#_Toc407443544)

[4.2.4 Analyse der Kriterien 5](#_Toc407443545)

[4.3 Architektur 5](#_Toc407443546)

[5 Umsetzung 7](#_Toc407443547)

[5.1 Generell 7](#_Toc407443548)

[5.2 Spielfeld 7](#_Toc407443549)

[5.3 Funktionen über LUA Script 7](#_Toc407443550)

[5.4 Interne Umsetzung 7](#_Toc407443551)

[6 Ausblick 8](#_Toc407443552)

[7 Fazit 9](#_Toc407443553)

[Literaturverzeichnis I](#_Toc407443554)

[Abbildungsverzeichnis II](#_Toc407443555)

[Abkürzungsverzeichnis III](#_Toc407443556)

[Glossar IV](#_Toc407443557)

# Einleitung

Stichworte:

* Spielend Lernen
  + Competitive! (deutsches Wort nicht eingefallen…don’t judge me)
* Technische Herausforderung Logik
  + Echtzeit?
* Robocup 🡪Alt, was neues her?
  + Höhenachse hat Robocup 2d nicht…gibt aber auch 3d…-.-
* Unreal Engine 🡪 Stand der Zeit
  + Aktuelle Techniken…!

# Aufgabenstellung

Das Ziel dieser Arbeit ist es, eine grafische Fußballsimulation zu erstellen, in welcher der Spieler eigene Logiken für sein Fußballteam hinterlegt und somit deren Verhalten beeinflusst. Dabei besteht ein Team aus jeweils 10 Feldspielern und einen Torwart. Während eines Fußballspiels werden die gängigen Regeln beachtet und bei einer Missachtung je nach Stärke des Vergehens geahndet. Innerhalb der Simulation werden die Bewegungen des Balls physikalisch korrekt simuliert. Die Spieler besitzen als Limitierung ein Ausdauersystem, welches verhindern soll, dass ein Spieler dauerhaft sprintet oder den Ball in kurzer Zeit mit maximaler Kraft kickt.

Zur Umsetzung wird die „Unreal Engine 4“ der Firma „Epic Games“ genutzt. Dies ist eine Spiele-Engine, welche bereits viele Werkzeuge für eine schnelle Entwicklung bereitstellt. Mehr dazu finden Sie im Kapitel „3.2 Unreal Engine 4“. Diese Engine wird um eine Schnittstelle zur einer Skriptsprache (Siehe „3.3 LUA Script“) erweitert, mit deren Hilfe einzelne Skripte an die Spieler angehängt werden können und somit deren Verhalten definieren werden kann.

Die Simulation ist komplett dreidimensional und bietet animierte Fußballspieler, wobei der Aktionsumfang der Spieler zuerst auf ein Basisrepertoire beschränkt wird. Dazu gehört die Aktionen „Laufen“, „Schießen“ und „Ball annehmen“. Die Ausarbeitung der grafischen Komponente wird in der Priorität hinter den funktionalen Teil der Arbeit gestellt.

# Stand der Technik

## Einleitung

## Unreal Engine 4

## LUA Script

## Robo Cup

## Künstliche Intelligenz

## Zusammenfassung

# Planung / Struktur

## Projektmanagement

Die Projektentwicklung ist in zwei Phasen unterteilt. In der ersten Phase, die während des fünften Semesters stattfindet, wird zunächst eine Planung sowie die passende Architektur und ein Prototype entwickelt. In der zweiten Phase, die während des sechsten Semesters stattfindet, wird eine Implementierung nach der entwickelten Architektur erstellt und getestet.

Das Projekt wird mit einem vier Mann staken Team umgesetzt, aus diesem Grund wird ein guter und regelmäßiger Austausch gepflegt. Hierzu findet wöchentlich ein Meeting des gesamten Teams statt um den aktuellen Vorschritt zu besprechen und neue Aufgaben zu bestimmen. Zur Sicherstellung, dass in die richtige Richtung gearbeitet wird findet monatlich ein Treffen mit dem Auftraggeber statt. Hierbei wird über den aktuellen Stand berichtet und weitere Ziele diskutiert. (Siehe Anhang XX Meetings-Protokolle)

Aufgrund dieser agilen Softwareentwicklung entstehen nur grobe Zeitpläne und Meilensteine.

## Analyse der Aufgabenstellung

Aus der Vorgegebenen Aufgabenstellung werden zunächst die elementaren und zusätzlichen Eigenschaften herausgearbeitet und in Muss-, Kann- und Sollkriterien unterteilt. Die erstellen Kriterien werden nach ihrer Machbarkeit analysiert und eine geeignete Umsetzung ausgewählt.

### Muss-Kriterien

* Simulation eines Fußballspiels
  + Physikalische Korrektheit
  + Einhaltung der Fußballregeln
  + Kompatibel mit Windows und Linux
* Erstellen einer KI für die Fußballspieler
* Nachträgliches erstellen neuer KIs

### Soll-Kriterien

* Bestrafung der Spieler bei Nichteinhaltung der Regeln
* 3D-Darstellung

### Kann-Kriterien

* Grafischer Editor zur Erstellung von einer KI

### Analyse der Kriterien

Die Anforderung einer physikalisch korrekten 3D Simulation wird über eine Game-Engine realisiert, da diese eine eingebaute Physik-Engine besitzen. In diesem Projekt wird die Unreal Engine von „Epic Games“ verwendet. Die Unreal Engine erfüllt die Bedingung der Plattformunabhängigkeit, es wird Windows und Linux unterstützt.

Um die KIs der Spieler nachträglich ändern zu können kann eine Skriptsprache verwendet werden, in diesem Projekt wird auf die freie Skriptsprache LUA gesetzt. Um die Skripte den Speilern zuzuordnen oder andere Einstellungen des Teams vorzunehmen wird eine XML Schnittstelle implementiert.

## Architektur

Die Architektur baut auf der Unreal Engine auf und wird aus diesem Grund vollständig in C++ entwickelt. Das Projekt greift auf existierende Technologien wie XML und LUA zurück:



# Umsetzung

## Generell

## Spielfeld

## Funktionen über LUA Script

## Interne Umsetzung

# Ausblick

# Fazit

# Literaturverzeichnis

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

# Abbildungsverzeichnis

# Abkürzungsverzeichnis

# Glossar