Benutzerhandbuch UnrealCup

Inhalt

[Inhalt 1](#_Toc424140396)

[Erster Start 2](#_Toc424140397)

[Linux 2](#_Toc424140398)

[Windows 2](#_Toc424140399)

[Aufbau der Teams 2](#_Toc424140400)

[Konfiguration 2](#_Toc424140401)

[Die Teams-XML-Datei 2](#_Toc424140402)

[Die Team-XML-Dateien 3](#_Toc424140403)

[Erstellung von KI-Skripten 3](#_Toc424140404)

[Grundlagen 3](#_Toc424140405)

[Spielablauf 3](#_Toc424140406)

[Anstoß 4](#_Toc424140407)

[Körperliche Eigenschaften der Spieler 4](#_Toc424140408)

[Funktionen der Spieler 4](#_Toc424140409)

[Aufbau der Skripte 6](#_Toc424140410)

# Erster Start

## Linux

Unter Linux kann die Simulation nach dem Entpacken des Archivs ohne weitere Vorbereitungsschritte mit der mitgelieferten, grundlegenden KI gestartet werden.

Bei der Ausführung im Linux-Labor muss allerdings darauf geachtet werden, dass die korrekten, proprietären Grafiktreiber verwendet werden, da es sonst zu Performanceproblemen kommen kann.

Zum Starten der Simulation muss lediglich die Datei *run.sh* im Wurzelverzeichnis ausgeführt werden.   
Optional kann das Programm auch mit der ausführbaren Datei *UnrealCup\Binaries\Linux\UnrealCup-Linux-Shipping* gestartet werden, in diesem Fall muss diese jedoch zunächst ausführbar gemacht werden.

Die Simulation kann zu jedem beliebigen Zeitpunkt mit der Tastenkombination Alt + F4 beendet werden.

## Windows

Unter Windows ist es möglich, dass vor der Ausführung zuerst die UnrealEngine-Daten installiert werden müssen.  
Diese werden unter Verwendung der Datei *Engine\Extras\Redist\en-us\UE4PrereqSetup\_x64.exe* installiert.

Gestartet wird die Simulation durch Ausführung von *UnrealCup.exe* im Wurzelverzeichnis   
Beendet wird die Simulation, wie auch bei der Linux-Version, mit der Tastenkombination Alt + F4.

# Aufbau der Teams

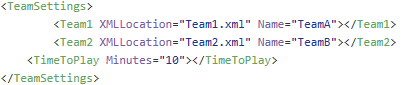
## Konfiguration

Die Teams, die in der Simulation gegeneinander antreten, können mithilfe von XML-Dateien konfiguriert werden. Die Dateien sind unter *\UnrealCup* zu finden, die vorhandenen Dateien und deren jeweiliger Verwendungszweck wird in der folgenden Tabelle erläutert.  
Die Dateien müssen zwingend unter dem genannten Pfad zu finden sein.

|  |  |
| --- | --- |
| Datei | Verwendungszeck |
| Teams.xml | Verknüpfung der Teamnamen mit den jeweiligen Konfigurationsdateien,  Einstellung der Spieldauer |
| TeamX.xml | Konfiguration der einzelnen Spieler eines Teams |
| Skripte | Beinhalten die KI der einzelnen Spieler |

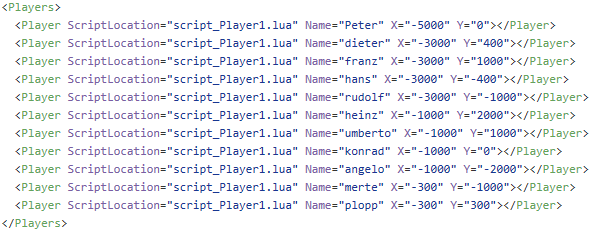
## Die Teams-XML-Datei

In der Teams XML-Datei werden die beiden Teams genannt, mit ihren Konfigurationsdateien verknüpft und die Spieldauer gesetzt. Die Datei ist wie folgt aufgebaut:



## Die Team-XML-Dateien

Die Team-XML-Dateien existieren für jedes Team und konfigurieren die einzelnen Spieler des Teams im Detail. Die Datei ist wie folgt aufgebaut:



X und Y stellen dabei die Startposition des Spielers dar.  
Die Positionen werden als kartesische Koordinaten Abhängig zum Ursprung (Anstoßpunkt) angegeben.

Es ist außerdem möglich, mehreren oder gar allen Spielern eines Teams dieselbe KI zu übergeben.

# Erstellung von KI-Skripten

## Grundlagen

### Spielablauf

Nach dem Starten des Spiels werden die Spieler auf die Startpositionen, die in den XML-Dateien definiert sind, gesetzt. Daraufhin wird der erste Anstoß ausgeführt.

Nach jedem gefallenen Tor wird der Punktestand des jeweiligen Teams angepasst, und die Spieler des gegnerischen Teams führen einen Anstoß aus.

Weitere Fußballregeln wie Aus, Abseits oder ähnliches sind nicht implementiert.

Nach dem Ablauf der eingestellten Zeit wird das Spiel eingefroren und der endgültige Punktestand ist sichtbar, das Spiel muss nun manuell beendet werden.

### Anstoß

Die Spieler werden mit dem Befehl „IsKickoff“ über die verschiedenen Anstoßzustände informiert:

|  |  |
| --- | --- |
| Zustand | Erklärung |
| Kein Anstoß | Derzeit muss kein Anstoß ausgeführt werden. Das Spiel wird normal ausgeführt, die Spieler können sich frei bewegen. Fällt ein Tor, wird der Zustand auf „Anstoß wird vorbereitet“ gesetzt. Wert: -1 |
| Anstoß wird vorbereitet | Der Anstoß wird vorbereitet. Die Spieler dürfen den Ball nicht berühren und müssen sich in ihre eigene Spielfeldhälfte zurückziehen. Nach TODO Sekunden wird der Ball am Anstoßpunkt platziert und der Anstoß wird mit dem Zustand „Anstoß kann durchgeführt werden“ freigegeben. Wert: 0 |
| Anstoß kann durchgeführt werden | Das Team, das den Anstoß ausführen darf, darf den Ball nun berühren. Sobald der Ball berührt wurde, wird der Zustand wieder auf „Kein Anstoß“ gesetzt und alle Spieler dürfen sich wieder frei bewegen. Wert: TeamID des Teams, das den Anstoß ausführen darf. |

Zwar werden die Spieler über den Zustand des Anstoßes informiert, die Ausführung des korrekten Verhaltens für jede Situation muss jedoch in der KI selbst gestaltet werden.

### Körperliche Eigenschaften der Spieler

Die Spieler sollen in ihren Eigenschaften annähernd menschlichen Fußballspielern gleichen. Das bedeutet, sie besitzen ein eingeschränktes Sichtfeld und keine unerschöpfliche Ausdauer.

Die Spieler können alles in einem Field of View (Sichtfeld) von 120 Grad vor sich sehen, die Entfernung spielt dabei keine Rolle. Außerdem ist es den Spieler möglich alles in einem Radius von 5 Metern wahrzunehmen.   
Die Position der beiden Tore ist den Spielern zu jeder Zeit bekannt, unabhängig davon, ob das Tor sich derzeit im Sichtfeld befindet.

Neben der Fähigkeit, Dinge zu sehen ist es den Spielern auch möglich, Gesprochenes von anderen Spielern zu hören. Auch hierbei spielt die Entfernung keine Rolle, jegliche Gespräche können auf dem gesamten Spielfeld vernommen werden.

Neben diesen Eigenschaften besitzen die Spieler auch eine Ausdauer. Diese Ausdauer wird für Aktionen wie das Kicken oder die Bewegung an eine bestimmte Position verbraucht. Dabei spielt es eine Rolle, wie stark bzw. schnell diese Aktion ausgeführt wird:  
Ein Spieler kann problemlos auf dem Spielfeld auf und ab gehen, beim rennen dieser Strecke wird ihm hingegen die Ausdauer ausgehen.

Besitzt der Spieler keine Ausdauer oder zu wenig Ausdauer, um eine gewisse Aktion auszuführen, so wird diese Aktion mit verringerter, aber maximal möglicher Stärke bzw. Geschwindigkeit ausgeführt.

Die Ausdauer der Spieler wird kontinuierlich regeneriert, nach einer kurzen Pause besitzt der Spieler also wieder seine volle Ausdauer.

## Funktionen der Spieler

Die Skripte der Spieler werden in der Skriptsprache LUA erstellt. Neben allen standardmäßigen LUA-Funktionen werden die folgenden weiteren Funktionen angeboten:



|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Input | Output | Beschreibung |
| MoveForward | Double: Way | - | Vorwärtsbewegen des Spielers |
| MoveTo | Double: PosX, PosY, Speed | - | Spieler an eine bestimmte Position bewegen |
| Rotate | Double: Angle | - | Spieler um seine Achse drehen |
| Kick | Double: DirX, DirY, DirZ, Force | - | Den Ball in eine Richtung kicken |
| StopBall | - | - | Den Ball stoppen |
| Speak | String: Message | - | Mit anderen Spielern kommunizieren |
| Listen | - | String: Message | Hören, ob ein anderer Spieler kommuniziert |
| AllowedToRun | - | Boolean: Active | Gibt zurück, ob die künstliche Intelligenz ausgeführt werden darf |
| TimePlayed | - | Double: TimePlayed | Gibt die aktuelle Spielzeit in Prozent zurück |
| IsKickoff | - | Boolean: Kickoff | Gibt zurück, ob ein Anstoß stattfindet |
| GetTeamId | - | Int: Team-Id | Gibt die Team-Id des Spielers zurück |
| HasBall | - | Boolean: HasBall | Gibt zurück, ob der Spieler am Ball ist |
| GetStamina | - | Double: Stamina | Gibt die aktuelle Ausdauer des Spielers zurück |
| GetOwnLocation | - | Double: PosX, PosY, PosZ | Gibt die aktuelle Position des Spielers zurück |
| GetBallLocation | - | Double: PosX, PosY, PosZ | Gibt die aktuelle Ballposition zurück, wenn sichtbar |
| GetVisiblePlayers | - | Array[Int Team-Id; Int Player-Id; Double PosX; Double PosY] | Gibt Positionen der sichtbaren Spieler zurück |
| GetGoal1Position | - | Double: PosX, PosY, PosZ | Gibt die Position des 1. Tors zurück |
| GetGoal2Position | - | Double: PosX, PosY, PosZ | Gibt die Position des 2. Tors zurück |

Außerdem stehen Ihnen, wie bereits kurz erwähnt, sämtliche Funktionen der LUA Bibliothek zur Verfügung, dazu gehören beispielsweise die grundlegenden Mathematikbefehle. Weitere Informationen finden sie unter: http://www.lua.org/manual/5.3/

So können die Folgenden Mathematikbefehle sehr hilfreich sein:

* Math.abs -> Absolutwert der Zahl
* Math.pow -> Potenz einer Zahl
* Math.random -> Zufallszahl

## Aufbau der Skripte

Um einen problemlosen Ablauf zu gewährleisten, sollte der Script zwei grundlegende Voraussetzungen erfüllen:

* Der Script muss von einer while-Schleife umgeben sein, die den Status von AllowedToRun() abfragt. Mittels dem Rückgabewert von AllowedToRun() muss überprüft werden ob das Skript noch ausgeführt werden darf.
* Es sollte in jedem Schleifendurchlauf überprüft werden, ob ein Anstoß ansteht.

Exemplarische Skripte sind im Archiv enthalten, es bietet sich die Möglichkeit, diese grundlegenden KIs als Grundlage für eigene Entwicklungen zu verwenden.