



ulm university universität  
**uulm**

**Fakultät für  
Ingenieurwissenschaften,  
Informatik und  
Psychologie**

Institut für Software-  
technik und Program-  
miersprachen

# Softwaregrundprojekt Meilenstein 6

Softwaregrundprojekt an der Universität Ulm

**Vorgelegt von:**

Gruppe 10

**Dozent:**

Florian Ege

**Betreuer:**

Stefanos Mytilineos

2019

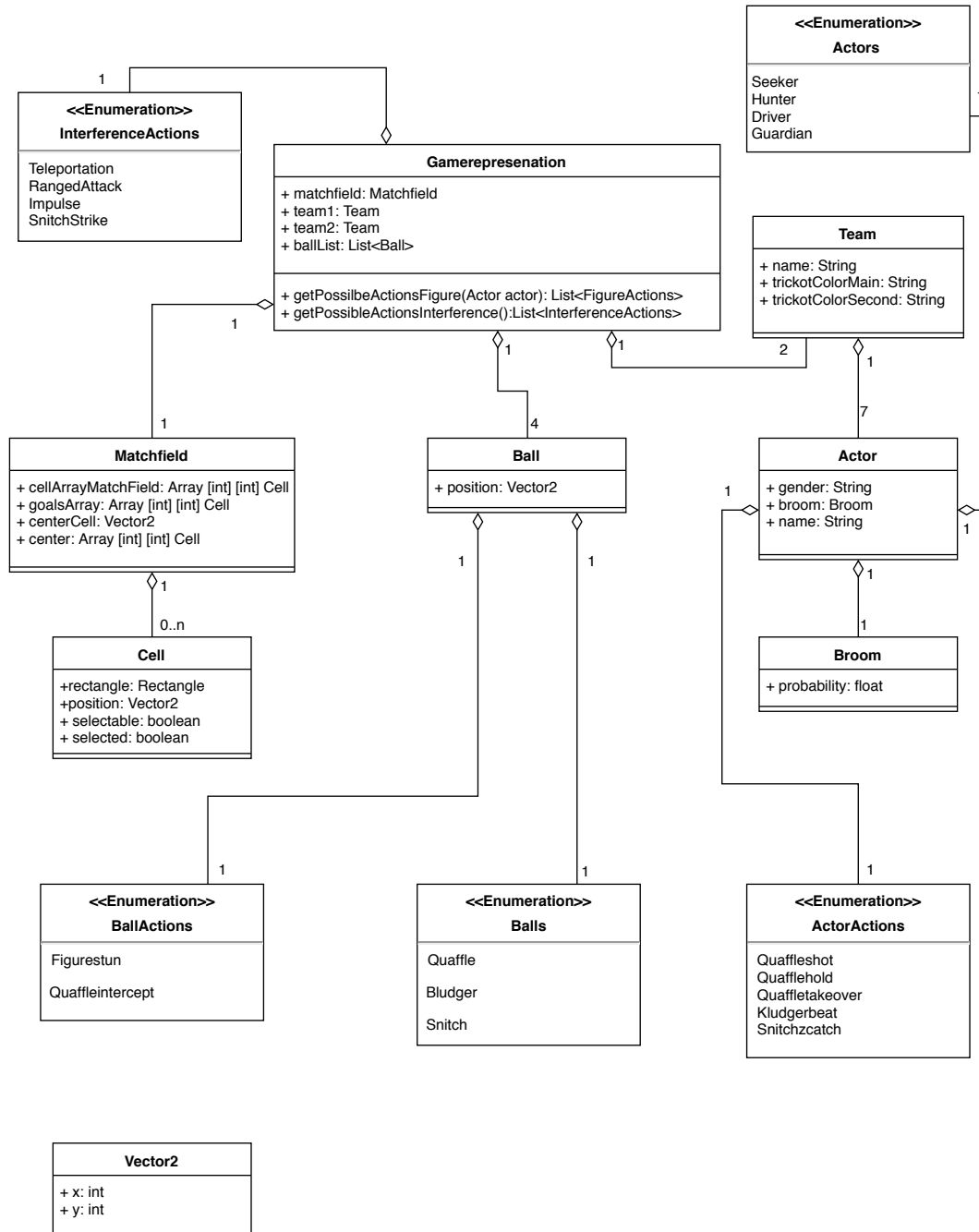
---

# Inhaltsverzeichnis

<b>1</b>	<b>Kommunikator</b>	<b>3</b>
1.1	Klassen-Diagramm . . . . .	3
1.2	Beschreibung . . . . .	4
1.2.1	Connector (Klasse) . . . . .	4
1.2.2	Connection (Klasse) . . . . .	4
1.2.3	LWS (Externe Bibliothek) . . . . .	4
1.3	Zuordnung der Funktionalen Anforderungen . . . . .	5

# 1 Model

## 1.1 Klassen-Diagramm



---

## 1.2 Beschreibung

Die Model ist für die Repräsentation für die Daten verantwortlich. Hier sind alle Daten hinterlegt, die für das Spiel relevant sind und angezeigt werden in der Benutzeroberfläche. Die Daten wertet der Controller aus und gibt diese dann der View weiter, welche diese dann auf der Benutzeroberfläche an.

### 1.2.1 Gamerepresentation (Klasse)

**Konstruktor** Der Konstruktor erstellt eine Instance des Objektes Gamerepresentation. In diesem Objekt sind alle Objekte die für das Spiel notwendig sind und auf dem Spielfeld agieren enthalten und jeweils eine Instanz erstellt.

**getPossibleActionsFigure (Methode)** Diese public Methode hat den Zweck eine Liste zu erstellen, in der alle möglichen Aktionen aufgelistet werden, die eine übergebene Spielfigur machen kann. Dazu überprüft die Methode welchen Typ Figur die übergeben Spielfigur hat, und welche AKtionen sie dadurch durchführen darf und gibt diese als Liste wieder.

**getPossibleActionsInterference (Methode)** Diese public Methode gibt die in der Enumeration definierten Einmischungen als Liste zurück. Dazu liest sie die Enumeration ein und speichert diese in einer Liste, welche dann zurückgegeben wird.

### 1.2.2 Matchfield (Klasse)

**Konstruktor (Methode)** In dem Konstruktor für das Matchfield, wird das eigentliche Spielfeld erstellt. Dazu erstellt diese Klasse ein Array von Zellen, welche das Spielfeld repräsentieren. Außerdem gibt es noch Bereiche, die besonders sind, aber auch Zellen auf dem Spielfeld sind. So sind die Torringe ebenfalls in einem Array hinterlegt, das Zentrum ist als Vektor hinterlegt und das Mittelfeld ist ebenfalls als Array hinterlegt.

### 1.2.3 Team (Klasse)

**Konstruktor (Methode)** Das Team ist ebenfalls Bestandteil der Klasse Gamerepresentation. In dem Team werden die 7 Spieler instantiiert, so wie der Name, die Trikotfarbe und die Ersatzfarbe für die Trikots.

---

#### 1.2.4 Ball (Klasse)

**Konstruktor (Methode)** Durch die Klasse Ball wird ebenfalls in der Gamerepresentation ein Bestandteil des Spieles instantiiert. Die Bälle bestehen aus einer Enumeration, welche Bälle es gibt, so wie aus einer Position, die als Vektor repräsentiert wird. Außerdem besitzt die Klasse Ball noch eine Enumeration, welche Aktionen ein Ball machen kann.

### 1.3 Zuordnung der Funktionalen Anforderungen

Die Funktionalen Anforderungen werden den Methoden folgendermaßen zugeteilt:

Methode	Funktionelle Anforderungen
getPossibleActionsFigure	FA21, FA24, FA26, FA28, FA30
getPossibleActionsInterference	FA32 - FA35