

Beschreibung UML-Diagramm

Im Folgenden UML-Diagramm sind Elemente der Unity-Engine nicht dargestellt, welche keine Eigenleistung der Gruppe ist. Trotzdem ist die folgende Darstellung des UML-Diagramms von der Unity-Engine beeinflusst, worauf unter Anderem in dieser Beschreibung eingegangen wird.

Die Entwicklung in Unity ist objektorientiert aufgebaut. Es erben alle Klassen von der Klasse „Object“. Grundsätzlich erben zwei Klassen – „GameObject“ und „Component“ von „Object“ direkt. Alle weiteren vom User erstellten Klassen erben von diesen beiden Klassen oder von Ihren Subklassen. Um dies darzustellen wurden Operatorblöcke benutzt. Diese dienen der Übersichtlichkeit und haben keine weitere Funktion. Zusätzlich wurden vor den Bezeichner der Klassen jeweils ein Zeichen mit einem Buchstaben und einer Farbe platziert. Analog zu den Operatorblöcken dienen diese Zeichen der Übersichtlichkeit.

Klassen bzw. Objekte, die von „GameObject“ erben, werden in einer sogenannten Szene dargestellt. Eine Szene (gehört zur UnityEngine) stellt den vom Nutzer der Software sichtbaren Bereich dar. Wie im UML-Diagramm zu sehen, haben Klassen, die von „GameObject“ erben, fast ausschließlich „Components“ als Attribute. Diese beschreiben die Verhaltenseigenschaften der Instanz des jeweiligen „GameObjects“. „Components“ enthalten meist grundlegende Datentypen, die als Parameter für die Unity Engine dienen, um z.B. das Verhalten der Physik mit dem Objekt zu verändern.

Zusätzlich zu den von Unity vorgegebenen „Components“ (Klassen die von „Component“ erben und ein orangefarbenes Zeichen haben), können eigene „Components“ in Form von „Scripts“ erstellt werden (erben von Behaviour <- MonoBehaviour). Sie können also wie andere „Components“ einem „GameObject“ zugewiesen werden, und bieten so die Möglichkeit, Verhaltensweisen von z.B. „Player1“ im Detail zu definieren.

In der im UML-Diagramm als Klasse dargestellten Legende sind die Bezeichner für die Eigenschaften public und private von Klassenattributen- und -methoden zu finden.

