**Beschreibung UML-Diagramm**

Aufbauend auf dem Sequenzdiagramm, ist im Folgenden das UML-Diagramm zu dem Unity-Projekt unserer Gruppe. Hierbei lassen wir von der Darstellung der Komponenten der Unity-Engine ab, da es keine Eigenleistung der Gruppe ist. Trotzdem ist die folgende Darstellung des UML-Diagramms von diesen Komponenten beeinflusst, worauf unter Anderem in dieser Beschreibung eingegangen wird.

Die Entwicklung in Unity ist Objektorientiert aufgebaut. Die Klasse „Object“ fungiert als Superklasse für alles was im Spiel dargestellt wird. Unter ihr in der Hierarchie stehen die Klassen „GameObject“ und „Component“, denen alle benutzererstellten Klassen untergeordnet werden. Um dies darzustellen wurden Operatorblöcke benutzt. Diese dienen nur der Übersichtlichkeit und haben keine weitere Funktion. Für eine bessere Übersichtlichkeit wurden zusätzlich vor dem Bezeichner der Klassen jeweils ein Zeichen mit einem Buchstaben und einer Farbe platziert.

Objekte die von der Klasse GameObject erben werden in der Szene der Engine dargestellt. Diese Szene ist ein Objekt, die alles beinhaltet, was innerhalb des Spieles sichtbar für den Nutzer ist. Variablen, Animationen, Eigenschaften o.Ä. dieser Objekte werden im Inspektor der Engine als sogenannte „Components“ erfasst. Diese „Components“ können zu Objekten hinzugefügt werden, um das Verhalten oder die Physik des Objektes zu verändern.

Zusätzlich zu den von Unity vorgegeben „Components“ (von „Components“ erbende Klassen), können eigene „Components“ in Form von „Scipts“ erstellt werden. Diese erben alle von „MonoBehaviour“ welche wiederum von „Behaviour“ erbt. Sie können also wie andere „Components“ einem „GameObject“ zugewiesen werden, und bieten so die Möglichkeit, Verhalten von z.B. „Player1“ genauer und „freier“ zu beschreiben.

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