**Game**

*Das Game verwaltet die Eingaben der Player und gibt diese an das aktuelle Subgame weiter.*

*Es kann von jeder Stelle des Spiels benutzt werden, d.h. es ist nicht an die Existenz eines GameObjects gebunden.*

* besitzen alle eine einzige Klasseninstanz (Singleton).
* besteht aus verschiedenen Subgames.
* besitzt 1-4 Player.
* besitzt Erfolgs- und Misserfolgsmeldungen an die Spieler.
* kann gestartet werden.
* kann pausiert werden.
* kann nach dem Pausieren fortgesetzt werden.
* kann einem Spieler einen Punkt zuschreiben.
* kann einem Spieler einen Punkt abziehen.
* Kann den Gewinner ermitteln.

**Subgame**

*Ein Subgame ist ein Spielmodus, bzw. eine Phase des Spiels. In Unity bildet jedes Subgame eine eigene Scene.*

* besitzen alle eine gemeinsame Zahl, die beschreibt auf wie viele ReactionTasks bis zum Fortschreiten zum nächsten Subgame korrekt reagiert werden muss.
* besitzt einen Namen.
* besitzt eine knappe Aufgabenbeschreibung.
* besitzt eine Zahl, die beschreibt auf wie viele ReactionTasks bisher korrekt reagiert wurde.
* kann seine Scene laden.
* kann das aktuelle ReactionTask ersetzen.
* kann prüfen, ob die eingetretene Reaktion korrekt ist.
* kann prüfen, ob es abgeschlossen (beendet) ist.

**TerminableTaskSubgame : Subgame**

*Ein TerminableTaskSubgame ist ein Subgame, das aus zeitlich begrenzten Abschnitten (ReactionTasks) besteht. Während der gesamten Anzeigezeit behält die kritische Eigenschaft des ReactionTasks seinen Zustand bei.*

* besitzt ein aktuelles TerminableReactionTask.
* kann prüfen, ob das aktuelle TerminableReactionTask abgelaufen ist.

**IndefiniteTaskSubgame : Subgame**

*Ein IndefiniteTaskSubgame ist ein Subgame, das aus zeitlich unbegrenzten ReactionTasks besteht. Seine ReactionTasks bleiben solange angezeigt, bis ein Spieler reagiert. Während der Anzeigezeit des ReactionTasks verändert sich dessen kritische Eigenschaft ein- oder mehrmalig.*

* besitzt ein aktuelles IndefiniteReactionTask.

**ReactionTask**

*Ein ReactionTask ist ein einzelner Abschnitt eines Subgames. Während seiner Anzeige können die Spieler auf Selbiges reagieren. Je nachdem, ob es die gesuchte Eigenschaft im Moment der Spielerreaktion besitzt oder nicht, werden dem reagierenden Spieler Punkte hinzugefügt oder abgezogen.*

* besitzt die gesuchte Eigenschaft oder besitzt sie nicht (ist im geforderten Zustand).

**TerminableReactionTask : ReactionTask**

*Ein BoundedReactionTask ist ein zeitlich begrenzter Abschnitt eines Subgames. Während seiner gesamten Anzeigezeit behält die kritische Eigenschaft ihren Zustand bei.*

* besitzen alle eine gemeinsame Anzeigedauer bis zu ihrem Ablauf.
* besitzt eine bisherige Anzeigedauer.
* Kann prüfen, ob es selbst abgelaufen ist.

**IndefiniteReactionTask : ReactionTask**

*Ein UnboundedReactionTask ist ein zeitlich unbegrenzter Abschnitt eines Subgames. Er bleibt solange angezeigt, bis ein Spieler reagiert. Während seiner Anzeigezeit verändert sich der Zustand der kritischen Eigenschaft ein- oder mehrmalig.*

**Player**

*Ein Player ist das Modell eines teilnehmenden Spielers. Dieser reagiert über sein Steuerungselement auf die einzelnen ReactionTasks.*

* besitzt eine aktuelle Punktezahl.
* besitzt einen Button.
* besitzt ein Anzeigefeld für Meldungen von *Game*.
* kann auf eine ReactionTask reagieren.

**zu XML-Dateien:**

Eine Entität in einer XML-Datei enthält immer 2 Einträge.

Dies kann eine Kombination zweier Kindelemente sein, die zusammen auftretend eine Reaktion korrekt machen. Eine andere Art ist eine Kombination eines Kindelementes und eines einfachen Wahrheitswertes, der angibt, ob eine Reaktion auf das Element korrekt ist.