Praktikum 5: Design

# Systemoperation: breakShip

## Beschreibung

Die Systemoperation breakShip prüft mittels der LoadRating-Klasse, ob die Ladung des Ship-Objekts ausgeglichen beladen ist. Ist die Ladung an einem Ort schlecht verteilt (d.h. der breakValue ist höher als 1), soll das Schiff brechen (eine Animation wird abgespielt).

## Diagramm

# 

# Systemoperation: deployContainer

## Beschreibung

Diese Systemoperation beschreibt den Vorgang, wie ein Container mittels des Krans vom Zug auf das Schiff transportiert wird. Ausgelöst wird der Vorgang von den Touch-Eingaben des Benutzers.

### Hinweise

Der Kran soll, wenn die Methode deployContainer(Container container) aufgerufen wurde und er so das Containerobjekt erhalten hat, sich selbstständig zu den Zielkoordinaten bewegen und anschliessend das Containerobjekt dem Schiff übergeben. Realisiert wird das mit Actions, einem Konzept des libGDX-Frameworks. Eine Action beschreibt eine Aktion, die über eine gewisse Zeitdauer automatisch ausgeführt wird. In diesem Fall wird eine Aktionssequenz zusammengestellt, bestehend aus einer MoveToAction (bewege dich zu Punkt (x,y)), einer von uns definierten Action, welche die addContainer(Container container)-Methode auf dem Schiff aufruft, und anschliessend eine weitere MoveToAction (bewege dich zurück zum Ausgangspunkt). Diese Aktionssequenz wird dann von libGDX eigenständig abgearbeitet. Deshalb ist der letzte Aufruf asynchron.

## Sequenzdiagramm



# Systemoperation: startCareerGame

## Beschreibung

StartCareerGame beschreibt das Initialisieren eines CareerGames. Dabei wird, nachdem die technischen Objekte wie die Stage erstellt wurden, anhand einer id ein Level aus der Persistenzschicht geladen. Aus dem neue erstellten Levelobjekt werden dann die nötigen Objekte und Daten geholt.

## Sequenzdiagramm



# Systemoperation: saveGame

## Beschreibung

Die Operation saveGame wird vom CareerGame nach erfolgreicher Beendigung eines Levels ausgeführt. Sie setzt die neuen Daten im Statistics-Objekt (vermutlich ein Singleton, da es sehr ähnliche Aufgaben wie eine Log-Klasse hat) und ruft anschliessend die Methode save() auf, welche das Statistics-Objekt dazu veranlasst, alle Daten über die Persistenzschicht zu speichern.

### Hinweis

Die Statistik- und Persistenzfunktionen sind noch nicht vollständig modelliert und dementsprechend auch nicht im Klassendiagramm vorhanden. Ausserdem werden die Spielstandsdaten vermutlich noch von den Statistikdaten entkoppelt. Anstatt :Statistics wird wohl in Bezug auf den Spielstand in Zukunft von :GameState oder ähnlichem die Rede sein.

## Kommunikationsdiagramm



# Anhang: Klassendiagramm



