Technische Dokumentation Super Star Battles

Autoren: - Kevin Künzi   
- Casimir Platzer  
- Michael Schmid

Inhalt

[Einleitung 3](#_Toc470013637)

[Klassendiagramm 4](#_Toc470013638)

[Main Package 5](#_Toc470013639)

[GameMain 5](#_Toc470013640)

[Game Panel 5](#_Toc470013641)

[Config 5](#_Toc470013642)

[ArgumentChecker 5](#_Toc470013643)

[Package gameState 6](#_Toc470013644)

[GameStateManager 6](#_Toc470013645)

[GameState 6](#_Toc470013646)

[MenuState 6](#_Toc470013647)

[OnePlayerVsBot 6](#_Toc470013648)

[Package tileMap 7](#_Toc470013649)

[PackageRenderer 8](#_Toc470013650)

# Einleitung

Super Star Battles ist ein in Java geschriebenes Action Spiel. Graphisch setzt es auf Swing und TileMaps. Genauere Infos zum Inhalt des Spiels befinden sich in der Spielbeschreibung.

Diese Dokumentation soll eine rasche Übersicht über die verwendeten Klassen und den Gedanken dahinter geben.

# Klassendiagramm

# Main Package

Im Main Package befinden sich 4 Klassen. Diese müssen für Anpassungen/Erweiterungen des GamePlays normalerweise nicht angepasst werden,

## GameMain

Beinhaltet die main Methode. Die Klasse startet nichts weiter als ein Jframe und setzt ein GamePanel als ContentPane

## Game Panel

Das GamePanel beinhaltet das Swing Image sowie den GameStateManager. Im Game Panel findet der Game Loop statt. Der Loop sorgt dafür, dass das Spiel in einem konstanten Intervall von 60 FPS (= 16 millisekunden) immer die Update() und Draw() Methode des jeweiligen GameStates ausführt .

Mehr Infos zu dem Game Loop Pattern befinden sich auf der sehr empfehlenswerten Seite: <http://gameprogrammingpatterns.com/game-loop.html>

## Config

In dieser Datei werden einige Balance Technische Variablen gesetzt, wie zum Beispiel der Startgesundheit und Bewegungsgeschwindigkeit der Spieler oder der Schaden der Standartrakete. Diese Zentralisierung der Variablen hat zum Ziel, das Spiel auf einfache Weise zu balancen.

## ArgumentChecker

Casimir

# Package gameState

Ein GameState ist ein „Level“ des Spiels. Jeder Level und Modus des Spiels hat eine eigene gameState Klasse. In der Spielversion 1.00 sind, dass die zwei gameStates „OnePlayerVsBotState“ und „TwoPlayerLevel1State“. Der gameState beinhaltet unterschiedliche Models ( Klassen des Package „models“ wie Players, Raketen, Powerups) beeinhalten. Der Vorteil dieser Klassen liegen in der einfachen Erweiterbarkeit des Spiels. Für ein neues Level müssen nur ein neuer GameState sowie ein neues TileMap erstellt werden.

## GameStateManager

Der Gamestate Manager beinhaltet alle GameStates und verwaltet diese. Die Klasse reicht den GameLoop vom GamePanel an den CurrentState ( == momentan aktiver GameState weiter).

## GameState

SuperKlasse aller gameStates,

## MenuState

Dieser State stellt das Menü dar. Er beeinhaltet eine Liste mit allen verfügbaren gameStates. Nach dem auswählen (Pfeiltasten) und bestätigen (EnterTaste) eines gameStates wird dieser als currentState markiert und somit gestartet. Per Escape Taste wird in jedem gameState wieder der menuState als currentState gesetzt.

## OnePlayerVsBot

# Package modell

Im Modell Package sind alle „SpaceObjects“ vorhanden. Als SpaceObject verstehen wir alle darstellbaren Entitys des eigentliche Spiels wie die Player (Schiffe), Projektile oder Pickups.

## Superklasse SpaceObject

# Package tileMap

Kevin

# PackageRenderer