Projektdokumentation

Idee:

Die Projektidee war zunächst ein Spiel zu programmieren, wobei ich auf der Suche nach geeigneten Brettspielen bei mir zu Hause auf WildLife kam.

Das Spiel ist auf der Basis der Library Slick2d programmiert. Die Idee dazu kam, da ich Slick bereits kannte.

Quellen:

Die Bilder zu den Spielelementen, sowie die grundsätzliche Spielidee kommen aus dem Brettspiel WildLife von Clementoni (Spieleautor: Wolfgang Kramer). Ich habe das Spiel allerdings vom Inhalt her leicht verändert und ein paar Elemente gestrichen.

Meine ursprünglichen Kenntnisse zu Slick2d stammen aus der Projektarbeit: „Eine Einführung in die 2D Spieleprogrammierung mit Java und Slick“ von Tobias Fritz.

Das Schreiben und Auslesen von Textdateien (welches ich zum Speichern benötige) habe ich auf folgenden Websites gelernt:  
https://www.tutorials.de/threads/einlesen-und-auslesen-einer-textdatei-in-java.339827/  
http://board.gulli.com/thread/610553-java-textdatei-schreiben/

Spielaufbau:

Klassen:

WildLife

Vererbt StateBasedGame  
-> Hauptklasse für Slick2d; Initialisierung der States

States (BasicGameStates) (verschiedene Modi des Spiels):

MainGame

* Eigentliches Spiel nach Auswahl der Spielerzahl
* Alle Spielaktionen laufen über diesen State

SpielerMenü

* Auswahl der Spielerzahl

MainMenu

* Menü (wird beim Start zuerst geöffnet)

Übergabespeicher:

Instanzen

* Getter und Setter zum vereinfachten Austausch zwischen den anderen Klassen
* Außerdem **SpielLaden**, was zum laden eines Speicherstandes dient. Allerdings ist diese Funktion noch nicht verfügbar.
* All diese Klassen sind im Package game.life

Andere Packages:

*Game.objects:*

Beinhaltet die Klassen **Button**; **Label**; **Text** , diese dienen zur Erstellung von Objekten ihres Namens (Button, etc.)

*Game.fields:*

Alle Klassen, deren Aufgabe die Funktionen des Spielplans sind.

Hierbei repräsentiert **Spielplan** den letztendlichen Plan, der Zusammengesetzt aus **Felder** ist. In Felder werden die einzelnen **Spielfeld**er zusammengesetzt und es wird die in **Feldtyp** definierte Feldart zugewiesen.

*Game.gameobjects:*

Beinhaltet Spielobjekte bzw. Bestandteile mit Ausnahme des Spielplans.

**Charakter:** dient zur Erstellung der Charaktertafeln (in den Ecken).  
**Charakterkarten**: hier wird jedem Spieler ein Charakter zugeteilt -> Aktionen zu *Charakter* laufen hierrüber.  
**Handkarte**: dient zur Erstellung und Ausgabe der einzelnen Karten.  
**Handkartenmenü**: Erstellung des Handkartenmenüs (Aufruf mit i); Zufällige Verteilung der Karten und Austausch der Karten; Interaktion mit dem Handkartenmenü  
**Spielregel**: dient zur Erstellung des Spielregelobjekts (Aufruf mit a)  
**Tierplättchen**: die Spielplättchen der Spieler werden hierrüber gezeichnet und erstellt.

*Game.gameactions:*

Komplexe Abläufe und Interaktionen werden aus *MainGame* outgesourct / exportiert -> Diese Aktionen laufen hauptsächlich über ihre Klasse und nicht über *MainGame.*

**Kartenaktion**: dient zur Handhabung der Aktionen die mit dem Auswählen und Benutzen von Karten zu tun haben.

* Je nach Karte wird dann:
  + - **Joker**: Aktion für die Jokerkarte; Jokermenü Erstellung und Interaktion
    - **PlättchenSpielen**: Aktion für die Landschaftskarten; Prüfen von Daten in Bezug auf das Spielen eines Tierplättchens
    - **UpgradeSkill**: Aktion für die Upgrade Karte; Erstellung des Upgrademenüs und Verknüpfung zu den Charakterfähigkeiten

ausgeführt.

**SpielerAmZug**: hierrüber wird geregelt, wer am Zug ist. Somit auch die Ausgabe in rot und das Handkartenmenü wird angepasst.

**Wertung**: das Prüfen, ob eine Wertung notwendig ist und die Abhandlung von Wertungen wird hierrüber geregelt.