Pflichtenheft

ElferRaus

P23

08.05.2015

# Management- und Dokumentationsattribute

|  |  |
| --- | --- |
| **Dokumentationsattribute** | |
| **Autor** | Adrian Schmidt, Lara Sievers, Fabian Schneider |
| **Eindeutige Teamnummer** | P23 |
| **Quelle** |  |
| **Version** | 1.0 |
| **Bearbeitungsstatus** | Final |

# Visionen und Ziele

/PV10/ Digitalisierung des Spiels als taktisches Kartenspiel

/PZ10/ Der Gegner soll eine K.I. sein.

/PZ20/ Das Spiel soll in der Kommandozeile laufen.

# Rahmenbedingungen

/PR10/ Das Spiel soll auf einem Computer unabhängig vom Betriebssystem laufen.

/PR20/ Die Entwicklungsumgebung soll Eclipse sein.

/PR30/ Als Programmiersprache soll Java verwendet werden.

/PR40/ Das digitale Kartenspiel soll von Laien bedienbar sein.

/PR50/ Das Spiel soll für 1-2 Runden geeignet sein.

/PR60/ Das Programm soll als jar-Datei gepackt und ausführbar sein.

/PR70/ Das Spiel soll für Leute der Altersgruppe von 8 – 60 geeignet sein.

/PR80/ Höfliche Anredeform.

/PR90/ In dem Spiel dürfen keine gewalttätigen, rassistischen oder pornografischen Inhalte dargestellt werden.

# Kontext und Überblick

/PK10/ Das Programm soll von allen zugänglich und benutzbar sein.

/PK20/ Zum Spielen wird eine herkömmliche Computermaus oder ein vergleichbares Zeigegerät und eine Tastatur oder ein vergeichbares Eingabegerät benötigt.

/PK30/ Das Spiel soll auf einem Computer unabhängig vom Betriebssystem laufen.

/PK40/ Das Spiel wird in einer IDE (Eclipse) entwickelt.

# Funktionale Anforderungen

/PF10/ Muss eine K.I. enthalten.

/PF20/ Kann einstellbare Schwierigkeitsgrade haben (Leicht, Schwer).

/PF30/ Kann einen Multiplayermodus haben für insgesamt maximal 4 Spieler.

/PF40/ Kann mehrere K.I.-Gegner gleichzeitig haben (maximal 3).

/PF50/ Muss mit der Maus und Tastatur bedienbar sein.

/PF60/ Muss den Gewinner ausgeben.

/PF70/ Muss die Karten des aktiven Spielers (ausgenommen K.I.) anzeigen.

/PF80/ Muss die Karten der K.I. immer verbergen.

/PF90/ Muss die Anzahl der Karten auf der Hand darstellen.

/PF100/ Muss die Karten auf dem Spielfeld darstellen.

/PF110/ Soll den Stapel mit der Anzahl der verbleibenden Karten darstellen.

/PF120/ Muss zwischen vier Farben unterscheiden können (Rot, Orange, Grün, Blau).

/PF130/ Muss pro Farbe je eine Karte der Zahlen 1 bis 20 beinhalten.

/PF140/ Muss zwischen den Spielphasen unterscheiden, sodass der Gegner nicht in den Zug des aktiven Spielers eingreifen kann.

/PF150/ Das Spiel kann ein Hilfedokument zur Verfügung stellen.

/PF160/ Kann eine GUI haben.

# Qualitätsanforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Systemqualität** | **Sehr gut** | **Gut** | **Normal** | **Nicht relevant** |
| **Funktionalität** |  | X |  |  |
| **Zuverlässigkeit** |  | X |  |  |
| **Benutzbarkeit** |  | X |  |  |
| **Effizienz** |  |  | X |  |
| **Wartbarkeit** |  |  | X |  |
| **Portabilität** |  |  |  | X |

Tabelle 1: Qualitätsanforderungen

# Abnahmekriterien

Das Programm startet und erstellt ein Set aus 20 Karten pro Farbe, bei vier Farben.

Die Karten werden gemischt.

Jeder Spieler erhält 11 Karten.

Eine Karte kann nur auf einen Stapel der gleichen Farbe gelegt werden.

Karten gleicher Farbe können nur zu Karten auf einen Stapel gelegt werden, wo vorher eine Karte um 1 kleiner der zu legenden Karte lag, bei Zahlen größer 11.

Karten gleicher Farbe können nur zu Karten auf einen Stapel gelegt werden, wo vorher eine Karte um 1 größer der zu legenden Karte lag, bei Zahlen kleiner 11.

Bei gelegten finalen Karten (1 oder 20) können keine weiteren Karten gelegt werden.

Sobald eine 11 vom Stapel gezogen wird muss diese gelegt werden.

Sofern beim Ziehen der jeweils ersten Karte pro Durchgang vom Stapel keine 11 gezogen wird, müssen zwei Karten nachgezogen werden.

Karten mit einer 11 die beim Nachziehen (zwei Karten ziehen) gezogen werden, können erst in der nächsten Runde gelegt werden.

Der Wert der Karte im System stimmt mit dem angezeigten Wert überein.

Die Karten des Gegners sind nicht einsehbar.

# Glossar

Eclipse ↗IDE auf der Basis von Java

IDE Integrated Development Environment = Entwicklungsumgebung zum Programmieren von Software

GUI Graphical User Interface (Grafische Benutzeroberfläche)

# Literatur

## Hinweis zu dieser Vorlage

Die Vorlage für dieses Pflichtenheft wurde Balzert (2009), S. 492 ff. entnommen.

Literaturliste

Balzert, Helmut (2009). Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. 3. Auflage. Heidelberg: Spektrum, Seite 492 ff.