**Projektarbeit – Teil 2**

DstGrd Vorname Name: Oberleutnant Martin Tennert

DstGrd Vorname Name: Oberfähnrich Dennis Freitag

|  |
| --- |
| **Bearbeitungshinweise:**  Sie haben für die Bearbeitung des Projekts Zeit vom 16.08.2017 bis 25.08.2017 – 07:00 Uhr  Die Präsentation erfolgt am 25.08.2017 um 07:30 Uhr.  Die Gesamtleistung ist in allgemeine Punkte und Aufgabenpakete aufgeteilt. Ein Aufgabenpaket soll von **genau** einem Teammitglied bearbeitet werden. Diese Zuteilung **muss** dokumentiert sein.  Es sind maximal 120 Punkte erreichbar – zum Bestehen sind min. 50 % der Punkte erforderlich.  Die Nutzung eines GUI-Builders (oder ähnlicher Hilfsmittel) ist in diesem Abschnitt ausdrücklich untersagt!  Erzeugen Sie zur Abgabe eine ZIP-Datei, die folgende Ordnerstruktur enthält:   * bin: Ordner mit der ausführbaren JAR-Datei * src: Ordner mit allen notwendigen Quelldateien aus dem Eclipse-Projekt. Innerhalb dieses Ordners können Sie selbstverständlich Unterpakete anlegen. * res: Ordner für alle extern verwendeten Ressourcen (z.B. Logdatei, Bilder) * doc: JavaDoc im HTML-Format * dia: Alle erstellten Diagramme im PDF-Format, sowie **die eine** Enterprise-Architect-Datei * man: Das Handbuch der Applikation im PDF- und Word-Format |

|  |  |  |
| --- | --- | --- |
| **Gesamtpunktzahl** | **120** | **Korrektur** |
| **Erreichte Punkte** |  |  |
| **Note** |  |

Das Kartenspiel: „6 Nimmt“

## Aufgabenbeschreibung

Das Ziel bei dieser Aufgabe ist es, das bereits auf Konsolenebene implementierte Spiel „6 nimmt!“ mit einer grafischen Oberfläche (GUI) zu versehen. Die Forderungen aus der ersten Projektphase bleiben also bestehen. Mögliche Änderungen des Kunden und absolut notwendige Anpassungen können natürlich vorgenommen werden. Zum Erstellen der GUI sollen Elemente aus AWT und Swing / JFC genutzt werden.

## Zur Umsetzung

1. Erstellen sie ein fehlerfrei übersetzbares Java-Programm unter Nutzung des MVC-Paradigma. Nutzen Sie soweit irgendwie möglich das bestehende Model / den Controller aus dem ersten Projektabschnitt und binden Sie die neue View über bekannte Design Patterns an. Nutzen Sie passende Komponenten aus Swing und ermöglichen Sie dem Benutzer die größtmögliche Flexibilität bei der Bedienung des Programms. **(10 Pkt)**
2. **Aufgabenpakete:**

GUI

* 1. Spielfeld
  2. Optionen
  3. MenuBar
  4. ScoreBoard
  5. Protokoll
  6. StatusBar

Backbone

* 1. Protokoll speichern / laden
  2. Grübelzeit
  3. 1

1. Die GUI des Spiels soll in jedem Fall (mindestens) zweigeteilt sein:
   1. Der erste Teil soll das Spielfeld (die ausgelegten Reihen und gerade gespielten Karten) optisch ansprechend darstellen (einzelne Karten mit entsprechenden Grafiken). Außerdem soll eine kleine Anzeige existieren, in der sich erkennen lässt welcher Spieler gerade an der Reihe ist und wie viel Prozent der Hornochsen-Punkte bis zum Verlieren des Spiels der Spieler bereits erreicht hat (mit einem Fortschrittsbalken). **(20 Pkt)**
   2. Der zweite Teil soll für die Steuerung, Beeinflussung und Protokollierung des Spielablaufs zuständig sein und folgende Unterelemente enthalten:
      * Eine Ansicht der Handkarten des menschlichen Spielers, der gerade an der Reihe ist. Hier sollen – je nach eingestellter Option – auch die Karten der anderen Spieler (mindestens der Computerspieler) angezeigt werden.  
        **(10 Pkt)**
      * Eine Punktetabelle, in der die Ergebnisse jeder einzelnen Spielrunde (= dem Verteilen und Ausspielen von 10 Karten) übersichtlich dargestellt werden und aus den Einzelergebnissen jeweils ein gesamter Punktestand berechnet wird. Diese Tabelle soll grafisch sortierbar sein (z.B. nach Punkteanzahl, Namen, etc). **(10 Pkt)**
      * Eine Einstellungsseite für die Spieloptionen (Anzahl der Spieler, Anzahl menschlicher Spieler, Spielerstärken, Protokollierung, Anzeige der Handkarten der anderen Spieler, Geschwindigkeit des Spielablaufs bei Computerspiel). Diese Optionen sollen nur dann deaktiviert sein, wenn eine Nutzung absolut keinen Sinn macht (z.B. bei Veränderung der Spieleranzahl im laufenden Spiel). **(5 Pkt)**
      * Eine Seite, auf der das Protokoll des Spiels in Textform geführt wird. Dieses Protokoll soll während des Spiels kontinuierlich gefüllt werden (und für den Nutzer auch kontinuierlich betrachtbar sein). Der Nutzer soll aber auch die Möglichkeit haben, das angezeigte Protokoll zu löschen **(5 Pkt)**

Die Größe der beiden Teile soll sinnvoll vorgegeben werden, durch den Benutzer aber grafisch auch geändert werden können.

1. In der Oberfläche soll eine Menüleiste existieren. In der Menüleiste soll es (in passenden Menüs) folgende Einträge geben: **(10 Pkt)**

* Laden und Speichern der Protokolldatei
  + - Neustart des Spiels
    - Beenden des Programms
    - Anzeige eines kleinen Informationsfensters (mit Versionsnummer, Namen der Entwickler, etc.)
    - Dialog zur Auswahl einer möglichst hippen Hintergrundfarbe für das Spielfeld geben (es soll schon Anwendungsprogrammierer gegeben haben, denen das besonders wichtig war!).

1. Es soll eine Statusbar geben, in der wichtige direkte Informationen aus unterschiedlichen Bereichen angezeigt werden, wenn diese mit der Maus berührt werden: **(5 Pkt)**

* Ausgespielten Karte: Wer hat sie gelegt?
* Spieler: Wieviele Punkte hat er gerade?
* Menüeinträge / Knöpfe: Was wird hier ausgelöst?

1. Jede Aktion, die durch den Benutzer ausgelöst werden kann, soll im Handbuch beschrieben sein und durch passende Tooltips auch im Programm angezeigt werden können. Außerdem sollen bei allen Komponenten, die dazu in der Lage sind, Tastaturkürzel und Tastatur-Shortcuts verwendet werden. **(5 Pkt)**
2. Führen Sie eine zusätzliche Option ein, mit der die maximale „Grübel-Zeit“ für menschliche Spieler festgelegt werden kann. Sobald ein menschlicher Spieler an der Reihe ist, läuft diese Zeit rückwärts (sichtbar in der Statusleiste). Ist der Timer abgelaufen, so wählt das Spiel für den menschlichen Gegner einfach eine zufällige Karte zum Spielen aus (bzw. eine zufällige Reihe im Fall „Niedrigste Karte“). **(10 Pkt)**
3. Erstellen / erweitern Sie das Klassendiagramm und das Handbuch. Dokumentieren Sie Klassen und (nicht-triviale) Methoden mit JavaDoc. **(10 Pkt)**
4. Stellen Sie Ihr Programm und die erstellte Dokumentation (vor allem das Klassendiagramm) in einer kurzen Präsentation vor.

## Optionale Erweiterungen

Das Spiel soll zu einem beliebigen Punkt unterbrochen und wieder fortgesetzt werden können – sprich: Beim Schließen des Programms soll ggf. der aktuelle Spielzustand gespeichert werden und beim nächsten Start an genau dieser Stelle wieder fortgesetzt werden können. Dies soll nicht nur beim Schließen geschehen, sondern auch explizit über einen Menüeintrag durch den Nutzer initiiert werden können. **(+20 Pkt)**