

ESA 2 – KENTUCKY DERBY (IDEE: KLAUS SCHOLL)

1. Aufgabe

Früher wurde auf vielen Rummelplätzen "Kentucky Derby" gespielt, ein "Pferderennen", bei dem bis zu 12 Spieler gegeneinander antreten. Jeder Spieler treibt sein Pferd in seinem eigenen Tempo, indem er eine Kugel immer wieder über ein Spielbrett in Löcher rollen lässt. Die Löcher sind 1-3 Punkte wert, das eigene Pferd bewegt sich entsprechend 1-3 Züge. Wird kein Loch getroffen, bleibt das Pferd stehen. Es gibt nur eine Kugel pro Spieler, die immer wieder zurückrollt. Die Länge der Rennbahn beträgt 27 Züge.

Schreibt eine Java- Application, die dieses Spiel automatisiert spielt. Wie ihr die einzelnen Aspekte des Spiels umsetzt, steht euch frei - solange alle Akzeptanzkriterien erfüllt sind.



2. Akzeptanzkriterien

- Verwendet bei der Umsetzung ein beliebiges **Framework** (UI oder Grafik) für die Oberfläche.
- Aller Spieler sollen gleichzeitig (!) spielen (**Threading**).
- Setzt mindestens eines der gelernten **Design Pattern** an.
- Achtet auf **sauberen Code** (Formatierung, Bezeichnungen, Kommentare).
- Achtet bestmöglich auf die drei K-s (**Kapselung, lose Kopplung und hohe Kohäsion**)

Hinweis:

Ihr könnt das Spiel weitestgehend ohne Threading entwickeln und die Parallelität nachrüsten.

3. Bewertung

- **Framework (3):** Es wurde ein Framework sinnvoll angewendet.
- **Multithreading (5):** Threading wurde umgesetzt. Alle Spiele finden "gleichzeitig" statt.
- **Design Pattern (5):** Es wurde ein Design Pattern an einer sinnvollen Stelle eingesetzt.
- **Clean Code (3):** Der Quelltext ist sauber, lesbar und verständlich.
- **3 K-s (4):** Es wurde gegen Schnittstellen implementiert. Jede Schnittstelle und Klasse hat eine wohldefinierte Aufgabe und alle Attribute sind privat.

In den Bereichen Clean Code und 3 K-s kann es bei mangelhaftem Quelltext auch zu Punktabzügen kommen!

5. Abgabe

Die Abgabe ihres Projekts erfolgt via Moodle. Am besten ist ein Dokument mit dem **Link auf euer Git-Repository** und evtl. einigen erklärenden Zeilen, insbesondere zu evtl. dazuzuladenden Bibliotheken. (Dann müssen natürlich auch alle Bildressourcen in Git vorhanden sein.)

Alternativ könnt ihr eine **Jar-Datei eures Projektes** (eclipse, Netbeans...) hochladen (Quelltexte, Ressourcen und Libraries).

Für Anmerkungen fügt bitte ggf. eine Word-, Text- oder PDF-Datei Readme hinzu.

Die Bearbeitungszeit beträgt 3 Wochen nach Ausgabe. Die erste Woche danach gilt als "Puffer", d.h. es gehen nur 2 Punkte für die Verspätung verloren. Danach gilt die 50%-Regel.

Zusätzlich erfolgt eine **Abnahme im Chat** (Terminabsprachen), bei der jedes Gruppenmitglied Fragen zum Code beantworten muss. Primär dient dies der Diskussion und Beratung vor der Projektaufgabe. Es kann dabei aber für den Einzelnen zu Extrapunkten oder Punktabzügen kommen. Denn **es wird erwartet, dass jeder alle Aspekte des Codes gut kennt.**