



Die Hausaufgaben müssen von jedem Studierenden einzeln bearbeitet und abgegeben werden. Für die Hausaufgabe sind die aktuellen Informationen vom Blog zu berücksichtigen (<http://seblog.cs.uni-kassel.de/category/current-term/pmws1516/>).

Alle Abgaben müssen über unser Gitlab unter <https://docker.cs.uni-kassel.de> erfolgen. Abgaben per Mail werden nicht mehr akzeptiert.

Aufgabe 1 - Game Logic (50P)

1.1 - Statemachine

Implementiert in der Klasse *GameLogicController* eine Statemachine, welche für die verschiedenen Zustände des Spiels verwendet wird. Implementiert die Methode *endState()*, welche die verschiedenen States im Risk Objekt setzt. Die vorgeschlagenen States findet ihr im Modell. Bei folgenden Übergängen muss zusätzliche Funktionalität ausgeführt werden:

- Gamestate PLACEUNITS zu Gamestate ATTACK: Es kann nur gewechselt werden, wenn alle unplatzierten Einheiten gesetzt wurden.
- Gamestate ATTACK zu Gamestate MOVEUNITS: Keine Bedingungen.
- Gamestate MOVEUNITS zu PLACEUNITS: Es findet ein Wechsel des aktuellen Spielers statt. Der Spieler, welcher nun an der Reihe ist, erhält Verstärkungstruppen.

1.2 - Clicks

Implementiert in der Klasse *GameLogicController* die Methode *handleClickOnCountry*, welche abhängig von dem aktuellen Zustand der Statemachine, also dem Gamestate im Risk Objekt, das Spielgeschehen steuert. Folgende Funktionalität soll gegeben sein:

- Gamestate PLACEUNITS: Beim Klicken soll das ausgewählte Land um eine Einheit aus dem Vorrat des Spielers verstärkt werden.
- Gamestate ATTACK: Beim Klick auf ein Land des aktuellen Spielers wird das Land als angreifendes Land ausgewählt. Beim Klick auf ein Land des Gegners wird die *attack* Methode mit den ausgewählten Ländern gerufen.
- Gamestate MOVEUNITS: Beim Klicken mit der linken Maustaste wird das Herkunftsland ausgewählt, beim Klicken mit der rechten Maustaste das Zielland. Es wird immer eine Einheit bewegt. Für das Erkennen der Maustaste muss das *MouseEvent* übergeben werden, dazu muss die Methode um einen entsprechenden Parameter erweitert werden.

Aufgabe 2 - GUI Tests (50P)

Verwendet TestFX zum Testen eurer GUI.

2.1 - Init Game Test

Erstellt eine Klasse *GameInitializationTest.java*.

Dieser Test soll im MainApplication Screen (Auswahl der Spielerfarben und -namen) einen dritten Spieler hinzufügen (per Klick auf dem entsprechenden Button) und Namen setzen ("Älbert", "Christoph" und "Lennert"). Anschließend wird der Start Button gedrückt und im folgenden Screen im Info Bereich die Player Label daraufhin untersuchen, ob sie die gesetzten Namen enthalten.

2.2 - Update Units Attack Test

Erstellt eine Klasse *UpdateUnitLabelTest.java*.

Dieser Test soll ein Spiel starten, alle Verstärkungseinheiten des Spielers in ein Land platzieren, testen, ob das Unit Label entsprechend aktualisiert wurde, den Gamestate per Klick auf den endState Button verlassen, ein angrenzendes Land angreifen (analog zu Aufgabe 1 Aktionen ausführen) und anschließend die Änderungen an den Unit Labels testen. Um reproduzierbare Ergebnisse zu erhalten muss die Verteilung der Länder und das Ergebnis des Angriffs deterministisch implementiert werden. Hierzu muss analog zu den Tests der *attack* Methode auch das Verteilen der Länder mit einem gesetzten Seed erfolgen.