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 - Info Screen

In dieser Aufgabe ist die Verknüpfung der Benutzeroberflächen mit dem Datenmodell gefordert. Fügen Sie die bereitgestellten Klassen und Ressourcen ihrem Projekt hinzu (siehe Übung).

Erweitern Sie für diese Aufgabe die Klasse `GameInfoController` (Methode `createBindings()`).

1.1 End State Button

Fügen Sie dem End State Button einen Handler hinzu, der beim Drücken des Buttons die Methode `endState()` des GameLogicController-Objekts aufruft.

1.2 Player List

Fügen Sie für jeden Spieler ein farbig markiertes Label hinter dem Players Label hinzu. Diese Aufgabe wurde bereits in der letzten Hausaufgabe gestellt und ist ggf. nur zu übertragen.

1.3 Current Player (Property Change Listener)

Erstellen Sie einen PropertyChangeListener der bei Änderungen am Feld `currentPlayer` im Risk-Objekt im entsprechenden Label den Namen und die Farbe des Spielers setzt.

1.4 Current Phase (Property Change Listener)

Erstellen Sie einen PropertyChangeListener der bei Änderungen am Feld `gamestate` im Risk-Objekt im entsprechenden Label den Namen der Spielphase setzt.

Aufgabe 2 - Game Screen

In dieser Aufgabe muss die Verknüpfung der einzelnen Elemente der Game Screen Oberfläche mit dem Datenmodell über Controller nach dem Model-View-Controller Pattern hergestellt werden.

2.1 CountryController

Implementieren Sie an den markierten Stellen die Methoden der Klasse `CountryController` um folgende Funktionalität.

- Bei Mausklick auf das Land soll die Methode `handleClickOnCountry(Country country)` im GameLogicController-Objekt aufgerufen werden.
- Änderungen an den Units sollen visualisiert werden, indem das unitLabel-Objekt die Anzahl der Units und die Farbe des Spielers, dem die Einheiten gehören, anzeigt.