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.

Vorbereitung

Für die Bearbeitung der Hausaufgabe 5 benötigen Sie:

- Die Implementierung des Klassendiagramms von Risk. Entweder benutzen Sie ihre Lösung aus der vorigen Hausaufgabe oder laden eine fertige Version aus unserem Blog und importieren diese als Projekt in Eclipse.

Aufgabe 1 - Storyboards (100P)

Erstellen Sie SDMLib Storyboards zu den unten beschriebenen Szenarien. Passen Sie die Methoden
`Risk::checkEnd()` und
`Player::receiveReinforcements()` sowie
`Risk::attack(Country attacker, Country defender)` soweit an, bis alle Storyboard JUnit Tests erfolgreich durchlaufen. Die Endsituation und dementsprechend die Implementierung der zwei Methoden müssen den offiziellen PMWS1516-Risk Regeln genügen (zu finden im Blog und im Übungsvideo).

Randbedingungen:

- Ein Risk mit dem Objektnamen game
- Zwei Player:
  Ein Player mit dem Objektnamen alice, name = ”Alice” und color = ”aliceblue”
  Ein Player mit dem Objektnamen bob, name = ”Bob” und color = ”brown”
- Alle weiteren Objekte dürfen beliebige Objektnamen haben.
- In den Storyboards müssen jeweils sämtliche sichtbaren Teile des Spiels modelliert werden (alle Länder, deren Namen im Bild sichtbar sind, alle Einheiten in den Ländern).

Über Git gebt ihr euer Projekt, welches die drei SDMLib Storyboards (in den Klassen
`de.uniks.pmws1516.test.StoryboardCheckEnd.java`,
`de.uniks.pmws1516.test.StoryboardReinforce.java` und
`de.uniks.pmws1516.test.StoryboardAttack.java`) sowie die korrekte und allgemein gültige Implementierung der drei obigen Methoden enthält, ab.

- *Alice hat die Welt befreit*

Abbildung 1: Alice hat alle Länder besetzt

Alice ist noch am Zug und hat soeben das letzte Land von Bob erobert. Aufruf von `Risk::checkEnd()` auf dem Risk-Objekt risk.
Die Gewinner-Kante `winner` muss auf Alice gesetzt sein.
Bob darf kein `wonGame` kennen.

- *Alice neue Einheiten*

Abbildung 2: Alice erhält Verstärkung

Alice Zug beginnt und Sie erhält Verstärkungstruppen. Ein Aufruf der Methode `Player::receiveReinforcements()` auf alice soll dazu führen, dass diesem Objekt die korrekte Anzahl neuer Units hinzugefügt wird. Die Units sind noch in keinem Land platziert. Die Anzahl der erhaltenen Einheiten ermittelt sich durch die Anzahl der besetzten Länder geteilt durch sieben, abgerundet. Ist diese Zahl kleiner drei, erhält der Spieler drei Einheiten. Hinzu kommen Einheiten für besetzte Kontinente, d.h. hat ein Spieler alle Länder eines Kontinents besetzt, bekommt er entsprechend mehr Einheiten. Die Werte sind der Karte zu entnehmen, im Kasten unten links. Für den Kontinent Arlas gibt es beispielsweise drei zusätzliche Einheiten.

- *Bob greift Alice an*

Abbildung 3: Bob greift Alice von Arlas Barrens aus in Hagros an

Bob ist am Zug und hat bereits Verstärkungstruppen erhalten und platziert. Er greift Alice in Hagros an, mit seinen Einheiten in Arlas Barrens. Dafür wird die Methode `Risk::attack(Country attacker, Country defender)` mit den entsprechenden Objekten gerufen. Für den Unit Test muss ein reproduzierbares Ergebnis erzielt werden, während die Methode an sich allerdings zufällige Ergebnisse produzieren soll. Dafür muss die zufällige Zahlengenerierung entsprechend initialisiert werden, sodass bei jedem Durchlauf die selben Ergebnisse entstehen.