Readme

Kartengenerierung
Ich habe mich darauf entschieden, einen Random Kartengenerierung zu implementieren. Am Anfang werden die Grassfeldern generiert und danach Wasser und Bergfeldern. Ich habe auch eine Validierungsklasse implementiert, die die Business rules prüft. Wenn etwas davon nicht erfolgreich geprüft werden kann, wird die Map nochmals generiert, usw bis diese korrekt ist.
Fur das Validieren der Inseln der Karte habe ich das Floodfill Algorithmus implementiert. Die Idee wie diese implementiert werden muss habe ich von https://www.geeksforgeeks.org/flood-fill-algorithm-implement-fill-paint/ genommen.

Wegfindung
Fur die Wegfindung habe ich Dijkstra Algorithmus verwendet, die ich basierend auf die Quelle https://www.baeldung.com/java-dijkstra geschrieben habe.
Ich habe eine Klasse TargetSelector implementiert die die nächste Target selektiert basierend auf die aktuelle GameStateData. Erstens werden die beiden Hälfte festgestellt.
Wenn keine von den aktueller Position Nachbarn verfügbar sind (schon betreten) dann werden die noch nicht betretene Felder die auf den aktuellen “interessante” Kartenhälfte der Reihe nach genommen.
Wenn die Treasure noch nicht aufgenommen wurde, sucht der KI nach sie auf seine Kartenhälfte. Wenn diese schon aufgenommen wurde, sucht der KI nach dem gegnerische Burg auf den anderen Kartenhälfte. Die Target wird darauf basierend festgestellt.
Die PathCalculator Klasse bekommt diese neu festgestellte Target von dem MovementController und berechnet die kürzeste Weg von der aktuellen Position bis zu diesem Target. Danach wird die MovesList davon berechnet und zurück zu den MovementController zugeschickt.

Netzwerk
Die NetworkConverter Klasse konvertiert die Messages von dem Server zu Client Typen und die Messages von den Client zu entsprechende Server Messages.
Die Network Klasse kommuniziert direkt mit dem Server. Die beide Klassen kommunizieren miteinander mithilfe von einer NetworkController, die zum Beispiel ein Client Typ nimmt, dies in einem Server type umwandelt und zum Server schickt.

MVC
Das Pattern habe ich implementiert indem ich eine große Controller GameStateController erstellt habe, die das Zusammenspiel aller anderen Controllers koordiniert. Ich habe mehrere Controllers und deren entsprechender Models wie zum Beispiel die GameStateData Model die fur die GameStateController gedacht ist. Ich habe ein View, die CLI view, die mit dem Model GameStateData kommuniziert indem diese darauf benachrichtigt wird immer wenn eine Änderung an der Karte vorkommt und diese in die Console darstellt.