

Kartengenerierung:

1. Ich habe eine Karte erstellt (HalfMapGenerator.java), die vollständig mit Gras bedeckt ist.
2. Ich habe die Karte mit der erforderlichen Menge an Wasser und Bergfeldern gefüllt.
 - a. Für die Wasserfelder habe ich zuerst eines in jede Ecke gestellt. Der Rest wird zufällig platziert.
 - b. Die Bergfelder sind völlig zufällig angeordnet.
3. Der HalfMapValidator wird rekursiv aufgerufen, bis eine gültige Karte erzeugt wird.

Wegfindung-Algorithmus:

1. Für die Wegfindung (PathFinder.java) habe ich den Dijkstra-Algorithmus anhand einer Quelle implementiert, die ich im Internet gefunden habe:
<https://www.baeldung.com/java-dijkstra>
2. Um das richtige Target (TargetFinder.java) zu wählen, habe ich mehrere Methoden ausprobiert, aber am Ende habe ich mich entschieden, den nächstgelegenen Grasknoten zu suchen, der die meisten Grasnachbarn hat. Falls der aktuelle Knoten nur einen Grasnachbarn hat, bewegt sich der Spieler direkt zu diesem (ich habe festgestellt, dass der Knoten sonst übersprungen wird).

Anmerkungen: Der Wegfindungsalgorithmus war für mich der schwierigste Teil des Projekts, und er ist nicht so effizient geworden, wie ich es mir gewünscht hätte. Leider hatte ich nicht genug Zeit, um ihn richtig zu optimieren.

Netzwerk:

1. Mein Netzwerk ist in 2 Klassen aufgeteilt – ClientNetwork.java und NetworkConverter.java, die miteinander kommunizieren.
2. Das Netzwerk basiert auf der auf Moodle bereitgestellten Vorlage.
3. Ich habe den NetworkConverter in enger Übereinstimmung mit der Dokumentation implementiert. Dies schien mir die logische Option zu sein.

MVC:

1. Jede Komponente hat ein eigenes Paket: execute(Controller.java), ui(CLI.java), game(Model.java und GameData.java)
2. Der Controller ist für die Ausführung des Spiels verantwortlich.
3. Das Model und die CLI kommunizieren über PropertyChangeListener.