

Wegfindung: für diesen Teil habe ich Tiefensuche implementiert. Dafür habe ich den Pseudocode von Wikipedia übernommen, mit einigen Anpassungen. So wurden z.B. als Nachbarn eines Feldes nur Terrains GRASS und MOUNTAIN gezählt, um das Betreten von Wasserfeldern zu vermeiden. Außerdem habe ich so eine Art Backtracking hinzugefügt, die es ermöglicht, den Weg zurück zu nehmen.

Kartengenerierung: Die Karte wird zufällig generiert. Die Mindestanzahl von jeweiligen Terrains wird berücksichtigt. Um aber meinen dummen Wegfindung-Algorithmus zu kompensieren, habe ich die Mindestanzahl von Grasfeldern von 25 auf 33 erhöht. So gewinnt meine KI viel mehr, als davor. Die Validierung erfolgt innerhalb von Kartengenerierungsmethode, es werden also sicher mehrere Karten erstellt, und nur die richtige wird dann weiterverwendet. Für den Validator habe ich FloodFill implementiert. Diesen Algorithmus habe ich auch übernommen, die entsprechende Quelle befindet sich im Code. Die Ränder werden auch validiert.

Netzwerk: Converter ist ein Feld im Netzwerkklass und wird da intern verwendet und ist eigentlich für die anderen Klassen "unsichtbar". Das ist der einzige Bereich, wo ich auch die Exceptions werfe.

MVC: Ich habe den PropertyChangedListener implementiert. View-Klasse hat bei mir output-Methoden, die dann für das jeweilige Property ausgeführt werden. Controller-Methoden und die Verwendung von while- und ifs stellen meine Spiellogik dar. Controller registriert den User, prüft regelmäßig, wer ist dran und anhand von Model-Daten wird entschieden, was jetzt zu machen ist.