Beschreibung

In diesem Projekt nehmen Sie die Rolle einer imaginären Firma Quic(k)lean ein, welche Roboterstaubsauger produziert. Ihr Ziel ist es einen Staubsauger zu programmieren, welcher den Staub im Raum möglichst effizient saugen kann. Der Staubsauger weiss

quic(k)lean

SEARCHING FOR CLEANLINESS

anfangs nicht, wo der Dreck liegt. Sie wissen aber, wie dreckig der Raum ist. Es befindet sich immer drei Dreck im Raum. Der Dreck wird jedes mal zufällig generiert. Zur Notation verwenden wir ein Koordinatiensystem mit x und y. $x =$ Spalte (von links nach rechts): $0, 1, 2, 3$; $y =$ Reihe (von unten nach oben): $0, 1, 2, 3$; In diesem Projekt beginnen Sie bereits mit einer implementierten Suche in der Codevorlage. Diese Suche ist jedoch nicht optimal.

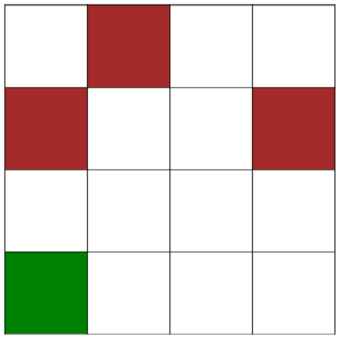

Abbildung 1: Screenshot von WebTigerJython. Ein Raum mit Dreck, der gesaugt werden sollte. Was wäre hier wohl der schnellste Weg?

Regeln

Ziel: Das Zimmer soll so effizient wie möglich gesaugt werden. Der Raum ist dann sauber, wenn Sie drei Dreck gesaugt haben.

Züge: Ihr Roboterstaubsauger kann sich horizontal, vertikal und diagonal jeweils ein Feld bewegen.

Farbencodes

Grün Die Station des Staubsaugers, die Suche beginnt hier. Sie liegt immer bei $(0, 0)$

Braun Dreck, welcher vom Staubsauger gesaugt werden sollte.

Schwarz (Siehe optionale Aufgaben) Möbel, welche im Raum stehen und den Staubsauger behindern können.

Weiss Freier Boden im Raum.

Aufgabe

Hier werden zusätzliche Aufgaben beschrieben, welche spezifisch für Ihr Projekt sind. Die Aufgaben sind in zwei Gruppen unterzeilt: Pflichtaufgaben und optionale Aufgaben. Die Pflichtaufgaben sollen am Ende des Projektes beantwortet sein. Die optionalen Aufgaben dienen dazu, sich weiter im Thema zu vertiefen. Diese Aufgaben sind Vorschläge. Sie dürfen auch eigene Fragestellungen vertiefen. Deklarieren Sie Ihre Fragestellungen klar im Projektbericht. Für alle Projekt gilt natürlich, dass Sie zuerst die allgemeine Aufgabenstellung und den Code (falls vorhanden) verstehen sollten, bevor Sie die untenstehen Aufgaben bearbeiten.

Pflichtaufgaben

- In der Vorlage ist bereits die Tiefensuche Implementiert. Was fällt auf?
- Versuchen Sie für diese Problem als erstes Die Breitensuche zu implementieren. Was für Probleme ergeben sich? Ist die Lösung optimal?
- Reflektieren Sie, welche Regeln bezüglich Pruning für Ihr spezifisches Problem sinnvoll sind und welche nicht. Was für eine Auswirkung auf mögliche Lösungen haben die verschiedenen Pruningstrategien?
- Schreiben Sie ein Programm (oder verbessern Sie das existierende), welches jeweils eine optimale Lösung für Ihr Problem findet. Um dies Umzusetzen, überlegen Sie sich, was man an der Breitensuche verbessern müsste, damit Sie funktionieren würde.

Optionale Aufgaben

- Wie ändert sich das Problem, wenn Sie in der Konfiguration Möbel einschalten?
- Können Sie diese Problem mit einer Heuristik lösen? Wenn ja, welche? Wenn nicht, warum nicht?
- (Sehr schwierig) Recherchieren Sie den Fachbegriff *Landmark Heuristik Search*. Überlegen Sie, wie Sie diese Technik in diesem Problem anwenden könnten.