

## Beschreibung

In diesem Projekt nehmen Sie die Rolle der imaginären Firma Food Sprint ein, welche Essen liefert. Ihr Ziel ist es für Ihren Fahrer möglichst effiziente Routen zu planen, da-



mit dieser das Essen liefern kann, während es noch warm ist. Der Einfachheit halber verwenden wir nicht eine Stadtkarte, sondern ein simples Koordinatensystem um zu navigieren. Zur Notation verwenden wir  $x$  und  $y$ .

$x$  = Spalte (von links nach rechts): 0, 1, 2, 3, 4;  $y$  = Reihe (von unten nach oben): 0, 1, 2, 3, 4; Sie wissen bereits im Voraus, wohin die Lieferungen gehen müssen (dies wird jeweils zufällig generiert). Und zwar haben Sie zu jeder Lieferung die genauen Koordinaten  $(x, y)$ . Sie müssen immer genau drei Lieferungen ausliefern. In diesem

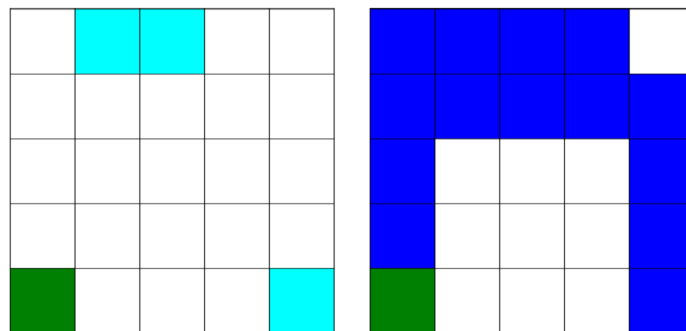


Abbildung 1: Screenshot von WebTigerJython. Eine mögliche Route, um das Essen zu liefern. War das wohl der schnellste Weg?

Projekt beginnen Sie bereits mit einer implementierten Suche in der Codevorlage. Diese Suche ist jedoch nicht optimal.

## Regeln

**Ziel:** Das Essen soll so effizient wie möglich ausgeliefert werden. Das Ziel ist erreicht, wenn Sie mit der Suche alle Kunden gefunden haben und das Essen geliefert ist.

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

## Farbencodes

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

**Cyan** Ort der Bestellung, hier können Sie das Essen abliefern.

**Weiss** Eine Strasse in der Stadt.

## Aufgabe

Hier werden zusätzliche Aufgaben beschrieben, welche spezifisch für Ihr Projekt sind. Die Aufgaben sind in zwei Gruppen unterteilt: 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 eine Heuristik, welche die Koordinaten der Lieferorte verwendet.

### **Optionale Aufgaben**

- Was passiert, wenn Sie die Konfiguration ändern, also beispielsweise die Breite und die Höhe verdoppeln und statt 3, 5 Lieferungen durchführen müssen? Können Sie sich eine Heuristik, welche auch bei sehr grossen Problemen zu einer effizienten Suche führt?
- (Sehr schwierig) Recherchieren Sie den Fachbegriff *Landmark Heuristik Search*. Überlegen Sie, wie Sie diese Technik in diesem Problem anwenden könnten.