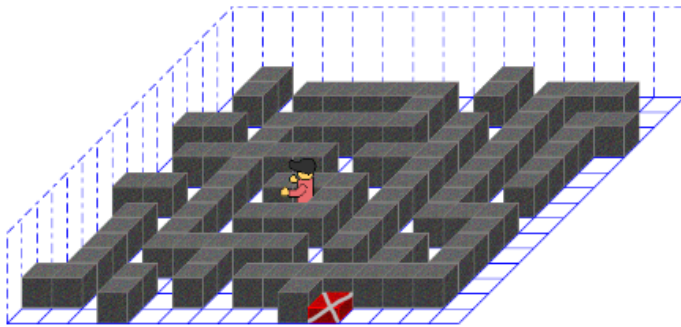
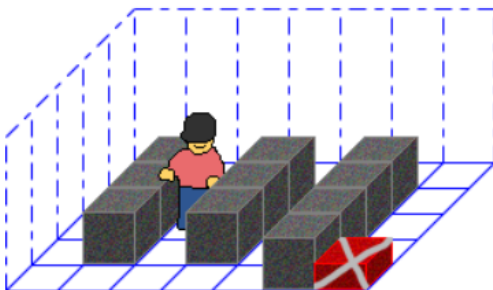


# Anleitung Labyrinth

Heyo, in dieser Anleitung findest du ein Programm, mit dem Karol [einfache und komplexe Labyrinthe lösen kann](#). Um das mit den begrenzten Mitteln zu lösen, wird an ein paar Stellen tief in die Trickkiste der Informatik gegriffen :)



Die meisten Labyrinthe lassen sich mit der Rechten-Hand-Regel lösen. Dieser Algorithmus kommt hier nicht zum Einsatz, sondern eine allgemeinere Version. Diese funktioniert auch dann, wenn das Ziel nicht mit einer Wand direkt verbunden ist.



Das Problem beginnt hier: Egal ob man die Linke-Hand- oder Rechte-Hand-Regel anwendet, in beiden Fällen wird Karol im Kreis um die Dreier-Reihe herumlaufen. Das Ziel ist nicht über eine Wand verbunden.

Um dieses Problem zu lösen, muss Karol zusätzlich die Felder markieren, die sie schon besucht hat. Damit kann sie erkennen, wann sie im Kreis läuft und entsprechend abbrechen.

Erstelle eine eigene Anweisung [Suche](#) und [SchrittZurück](#). Letztere Methode spart uns ein wenig Tipparbeit. Das Hauptprogramm besteht aus einem Aufruf der Methode [Suche](#).



Die Hauptarbeit findet in **Suche** statt. Hier ist der Code dazu.

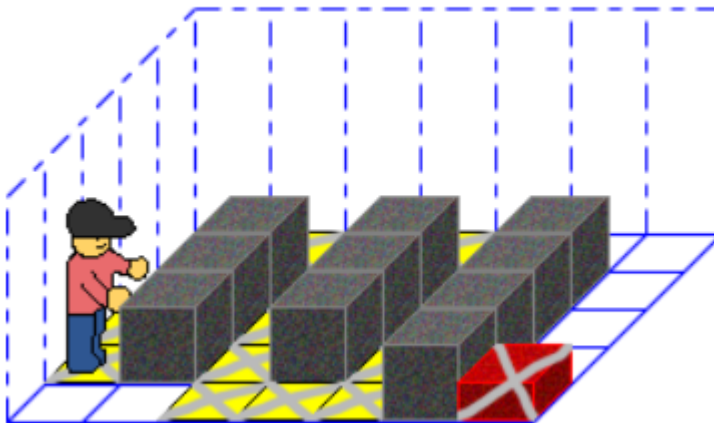


Die erste Bedingung prüft, ob das Ende erreicht ist oder nicht. Später wird darauf noch genauer eingegangen.

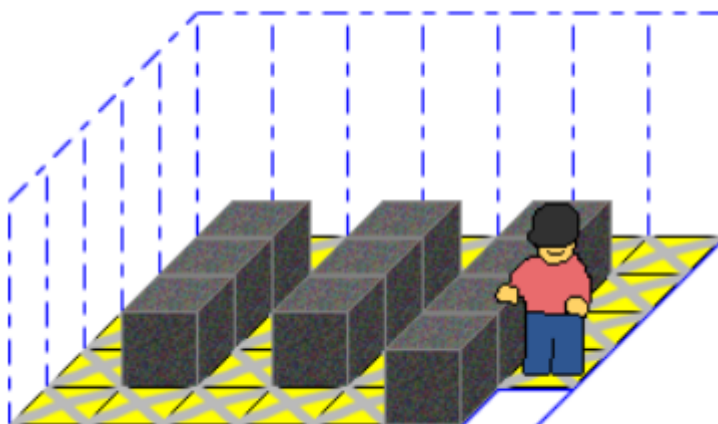
Bei der Suche versucht Karol, einen Schritt nach vorne zu gehen. Ist eine Wand oder das Feld schon markiert, dann gehe ich wieder zurück. Es wirkt, als ob nichts passiert wäre.

Wenn es aber ein neues Feld ist, wird dieses Feld markiert. Karol dreht sich dann nach links, vorne und rechts und ruft für jede Richtung erneut Suche auf. Damit merkt sich Karol, dass sie hier alle drei Richtungen durchsuchen muss. Wenn die Suche nicht erfolgreich war, dann geht Karol wieder den Schritt zurück.

Nun kann Karol erfolgreich aus Endlos-Schleifen ausbrechen und das Ziel finden. Karol markiert dabei besuchte Felder und wiederholt die Suche nicht mehrmals.



Jetzt gibt es nur ein Problem: Der Ziegel ist aufgehoben, aber es liegen noch überall Marken verstreut. Diese müssen aufgeräumt werden.



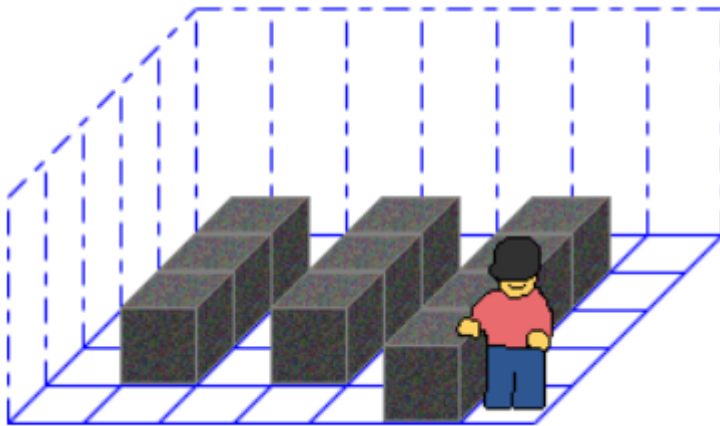
Die Lösung? Wir laufen nochmal durch das Labyrinth und heben die Marken wieder alle. Vervollständige dazu den Code für das Ende und lege eine neue Methode **Aufräumen** an.



Das Aufräumen funktioniert genauso wie die Suche über eine Rekursion.



Damit kommt Karol wieder am Ziel an und hat alle Marken gelöscht.



Zum Schluss nochmal der gesamte Code im Überblick.

