RHEINISCHE FACHHOCHSCHULE KÖLN

University of Applied Sciences

Studiengang: Informatik (Bachelor of Science)



**Transfer-Projekt**

Pathfinding Algorithmen Visualisierung in JavaScript

Vorgelegt von: Jurek Ohrndorf

651212013

Dozent: Prof. Dr. Friedel Mager

Wintersemester 2021/22

# Themenstellung

## Allgemeine Vorstellung und Eingrenzung der Thematik

In diesem Projekt möchte ich verschiedene Pathfinding Algorithmen mit HTM, CSS und JavaScript visualisieren. Es soll ein Start- und Endpunkt in einem Raster gewählt werden können. Außerdem sollen Flächen im Raster blockiert werden können und in die Wegberechnung mit einfließen.

## Konkrete Aufgabenstellung (oder Problemstellung)

Die Auseinandersetzung mit Pathfinding Algorithmen und deren Einbindung in einen von dem Nutzer manipulierbaren Raster.

## Vorgehensweise und Methodik

Zuerst werde ich mich mit der Theorie hinter den Pathfinding Algorithmen auseinandersetzten, dann wähle ich geeignete Algorithmen aus und werde diese in JavaScript schreiben. Danach werde ich die Seite in HTML und CSS designen und die Eingaben des Rasters in JavaScript definieren. Als letztes werden die Algorithmen im Raster eingebunden und die Visualisierung erstellt.

## Zielsetzung

Eine Seite auf der der Benutzer sich die Funktionalität von Pathfinding Algorithmen genauer veranschaulichen kann.

# Umsetzung

Schon bei meinem alten Arbeitgeber Check24 konnte ich einige Erfahrung mit JavaScript sammeln. Außerdem bekam ich durch das Modul Algorithmen und Datenstrukturen von der HHU, dass ich abgeschlossen habe, einen Einblick in Graphen und Algorithmen, die für sie geschrieben wurden. Also wollte ich eine Pathfinding Visualisierung in JavaScript umsetzen.

## Erster Versuch

In einem ersten Versuch setzte ich eine HTML Seite auf, designte diese mithilfe von CSS und fügte mit JavaScript die Möglichkeit ein einen Start- und Endpunkt und Wände in das Raster einzuzeichnen. Danach erstelle ich eine Klasse um die Datenstruktur um das Raster zu bauen. Diese wollte ich eigentlich in die Haupt JavaScript Datei importieren, jedoch stellte sich dies als nicht so einfach heraus. Da ich an dieser Stelle nicht weiterkam fragte ich meine Mitarbeiter bei meinem jetzigen Arbeitgeber Antwerpes. Diese konnten mir bei meinem ursprünglichen Projekt, selbst nach längeren probieren, nicht weiterhelfen. Einer meiner Mitarbeiter gab mir die Idee das Ganze doch mit der JavaScript library von Facebook umzusetzen, da ich diese ohnehin lernen müsste.

## Zweiter Versuch mit React

So war ich bei meinem zweiten Versuch angekommen und musste die Seite neu in React Syntax umsetzten. In React wird das HTML in jsx geschrieben und es gibt die Möglichkeit Komponenten zu benutzten. Diese lassen sich einfach wiederverwenden und es werden erstens viele Zeilen Code gespart und zweitens lassen sich die Komponenten gut in verschiedene Dateien aufspalten, um das Projekt übersichtlicher zu gestalten. Doch nahm dies viel Zeit in Anspruch, da ich nun erstmal lernen musste wie die unterschiedlichen Komponenten untereinander mithilfe von hooks und props kommunizieren. So lässt sich ein hook benutzen, falls man etwas in einer Variable speichern möchte, dass nicht bei jedem rerendering der Komponente auf seinen ursprünglichen Wert zurückgesetzt werden soll. Jeder hook beinhaltet auch eine set Funktion, die zur Veränderung der Variable genutzt wird. Wenn diese Funktion ausgeführt wird rerendern die Komponenten die die Variable des hooks benutzen. Als ich die Komponenten und die Funktionalitäten wieder auf dem Stand des ersten Projektes hatte habe ich eine Datei erstellt, um die Datenstruktur für den Graphen zu initialisieren. Die Datenstruktur bestand aus einem zweidimensionalen Array das Knoten Objekte enthielt. In den Knoten Objekt wurden verschiedene Zustände und Zeiger, für den Aufbau des Rasters und die Durchführung der Algorithmen, gespeichert. Als nächstes erstellte ich die Algorithmen, die die Knoten in einer Kopie der Datenstruktur mit Informationen befüllen und am Ende eine geordnetes Array mit den besuchten Knoten zurückgeben. Auch erstelle ich eine Funktion die ein geordnetes Array mit dem berechneten Weg vom Start- zum Endpunkt, mithilfe der veränderten Kopie der Datenstruktur, zurückgibt. Erst wird jeder Knoten, der in dem Array mit den besuchten Knoten ist, in der Datenstruktur, die in einem hook gespeichert wurde, in einem Intervall verändert. Danach wird nach gleichem Vorgehen bei dem Weg vom Start- zum Endpunkt die Datenstruktur verändert. Am Ende fiel mir noch ein schwerwiegendes Problem auf, dass beim Vergrößern des Rasters aufkommt. Denn wie vorher erwähnt wird eine Komponente jedes Mal neu gerendert, wenn sie eine hook Variable beinhaltet, die sich verändert. Der hook der den Graphen enthält wird nun in kürzester Zeit sehr häufig verändert und dadurch gibt React einen Fehler aus. Bei einer Größe des Rasters wie ich sie gewählt habe führt dies noch nicht zu Problemen, aber umso größer das Raster ist umso mehr steigt das Risiko für den Fehler. Diese Problem konnte ich leider selbst mit der Hilfe meiner Mitarbeiter nicht mehr lösen.

## Fazit

Eine Umsetzung mit der library React macht einiges einfacher und übersichtlicher, doch ist die Performance mit meiner Lösung noch nicht optimal lösbar. Die Zielsetzung ist mir jedoch trotzdem gelungen, da es in meiner gewählten Rastergröße keine Probleme gibt.