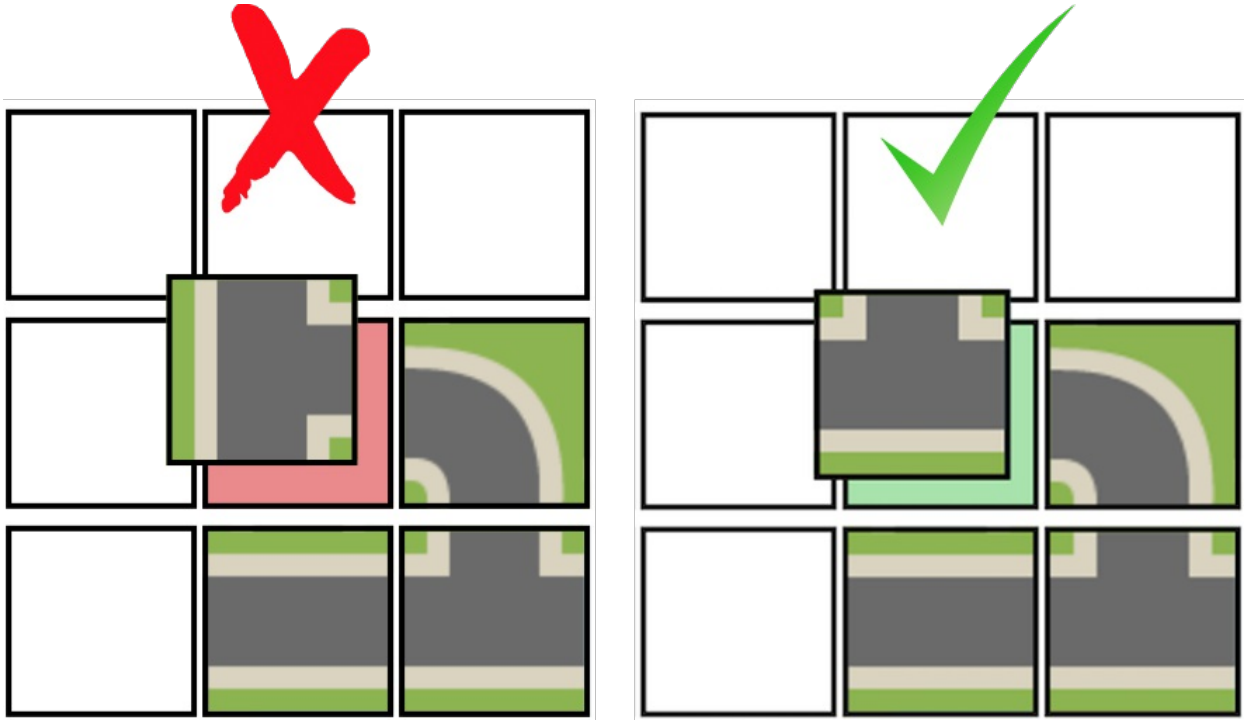
Using Dijkstra's Algorithm for Score Calculation in a Grid-based Game

Recently, I've been building a multiplayer tile-laying game inspired by games like Carcassonne and Settlers of Catan. I'll soon have some other blog posts about the game and some of the technologies and architectures I'm building it with, but today I'm going to walkthrough and illustrate how I used Dijkstra's Algorithm to implement the scoring component of the game.

The Game — Rules and Objectives

To begin, I'm going to explain the game, so as to motivate why I even needed a pathfinding algorithm in the first place. If all you want to see is my implementation, feel free to skip to the next section).

Like Carcassonne, the game begins with a single tile on the board. When it is each players' turn, they draw a random tile and must place it somewhere on the board. In order to place the tile, all of its edges must match with those surrounding it. Streets have to match up to streets, and grass has to match up with grass.



The Code — Implementation in Javascript

```
calcCompletedPathScore: function(path) {
```

```
    // calculates the value of a completed path
    // for the various players
    if (this.debug) console.log("Calculating Path Score...", path);
    ...
```

Inspiration and Resources:

- https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
- https://hackernoon.com/how-to-implement-dijkstras-algorithm-in-javascript-abdfd1702d04
- https://en.wikipedia.org/wiki/Maze_solving_algorithm