Assignment 1

Anders Bjorklund Jensen, Zohaib Butt

September 22, 2018
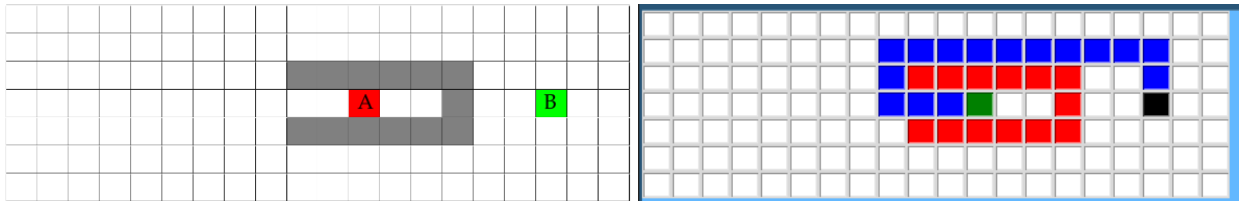
Part 1: Grids with Obstacles

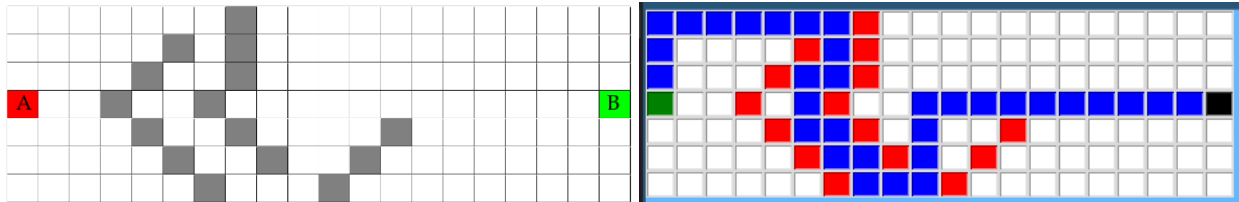We are using a slightly different color scheme than than the example figures, so just to clarify:

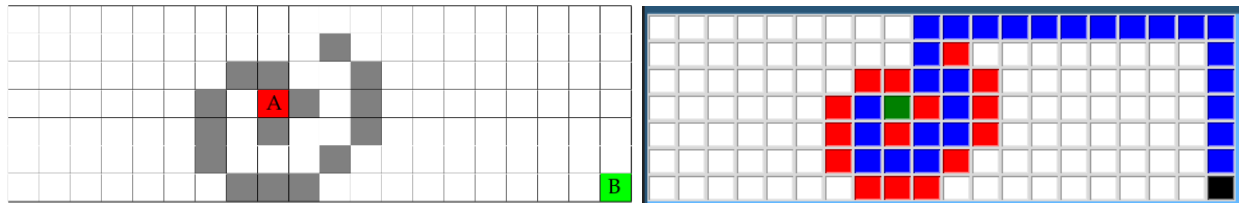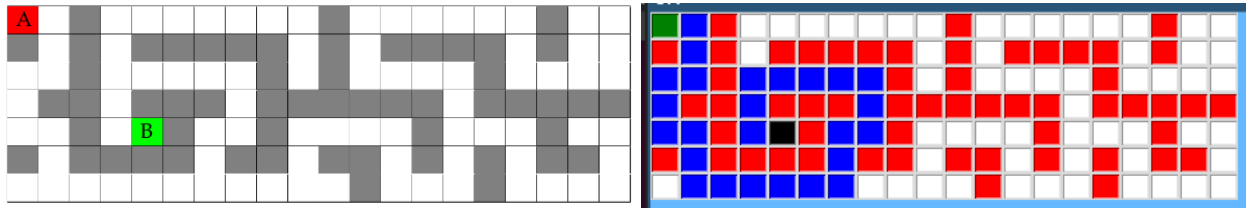| Example color	Our color	Description
Red	Green	Start
Green	Black	Goal
Gray	Red	Walls/Obstacle
	Blue	Path

Side by side comparisons
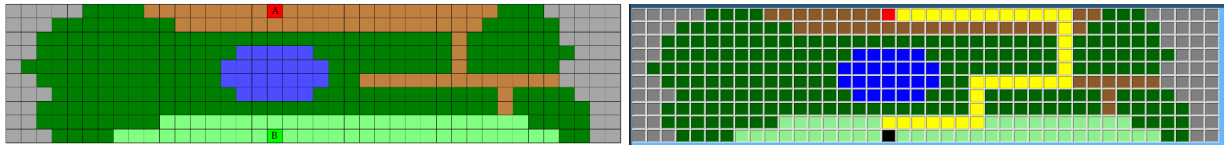
Board 1:

Board 2:

Board 3:

Board 4:

Part 2: Grids with Different Cell Costs

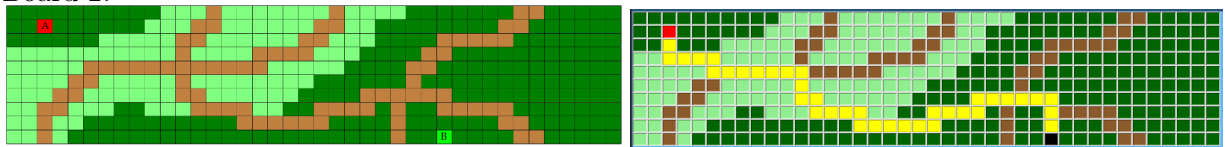In this part there is also some slight difference, so again just to clarify:

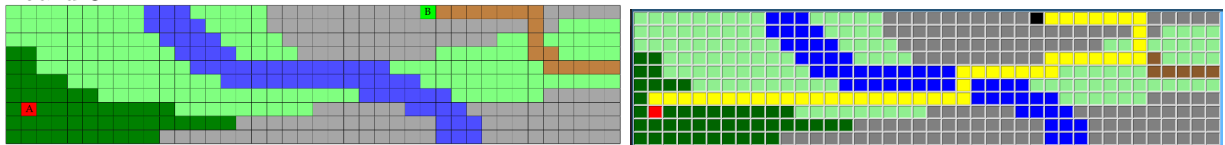| Example color	Our color	Description	Weight
Blue	Blue	Water	100
Gray	Gray	Mountain	50
Dark green	Dark green	Forests	10
Light green	Light green	Grasslands	5
Brown	Brown	Roads	1
Red	Red	Start	
Green	Black	Goal	
	Yellow	Path	

Side by side comparisons

Board 1:
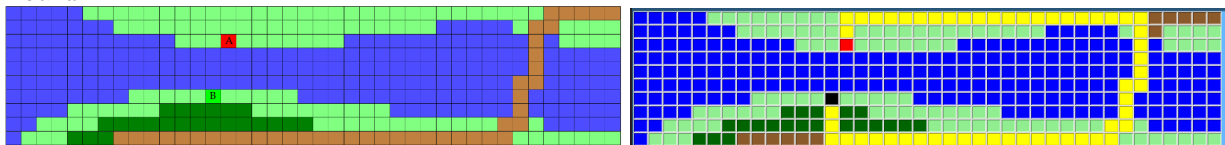

Board 2:

Board 3:

Board 4:

Code explanation

def gridParser(self, file)

This function parses the board files and creates a data structure which stores the environment and weights of the tiles, if any.

def printGrid(self)

This function prints the data structure that holds the environment data into the console.

def makeWindow(self, master)

This function prints a visual representation of the board's environment and the path choosen by the A* algorithm.

def makeShortestPath(self, cameFrom, current)

This function retraces the path and puts the total path selected into a data structure.

def heuristicValue(self, start, goal)

This function calculates the heuristic value using the Manhattan distance.

def aStar(self)

This function executes the A* algorithm to find the least weighted path.

- Selects best neighbour using manhattan distance as heuristic value

- Appends the current location to the closed set and removes it from the open set

- Goes to the best neighbour

- Repeats this process until goal is found

def main()

In this function, the visual window is configured and made.