Mikkel Svagård

A* Algorithm

I followed the pseudocode from the attachment, and produced the following agenda loop.

My function is able to run all the three algorithms, A*, BFS, and Dijkstra's, depending on the setup.

The algorithms run in the two classes Node and A_star, while the rest of the functions are helping functions such as representing the GUI and getting the input from the files.

I've noticed that since I sort my open-list (agenda here) a different place than many others the results from the BFS search differs in some of the cases.

```
def agenda_loop(self): #The A_star algorithm
  closed = []
  agenda = [self.start] #Open
  while(agenda): #Agenda Loop
    #The different sort methods, depending on algorithm chosen
    if(self.alg=="BFS"):
      pass #dont sort
    elif(self.alg=="Djikstra"):
      agenda.sort(key=lambda n: n.path_cost)
    else:
      agenda.sort(key=lambda n: n.path_clear)

    node = agenda.pop(0) #X <- pop(OPEN)
    self.table[node.pos()[0]][node.pos()[1]]="P" #updates the table, visitied current node
    closed.append(node) #push(X,closed)

    if node.goal: #if X is a solution retun (X,Succeed)
      self.update_path() #fills the path[] with it parents back to start
      print("Shortest Path Found")
      return True, self.table,self.path

    self.generate_all_children(node) #Succ <- generate_all_successors(X) + push(S,kids(X))
    for child in node.children:
      #Did not need "If node S* has previosly beed created, and if
      #state(S*)==state(S) then S <-S*
      if child not in agenda and child not in closed:
        self.attach_and_eval(child,node) #attach-and-eval(S,X)
        agenda.append(child) #insert(S,OPEN)
      elif node.path_cost + child.cost < child.path_cost:
        self.attach_and_eval(child,node)
        if child in closed:
          self.propagate_path_improvements(child)
  print("No Path Found") #if it breaks without retring, Loop Failed
  return False,self.table,self.path
```

Mikkel Svagård

My program doesn't take in any files, but rather reads them from the folders. Due to this it wont run if the the boards aren't structured in the directory in the following way: boards/boards/"filename".txt.

To run a file outside this structure described above, simply edit the "name" to the file path and set shortcut to false.

```
def main():
  #make sure to have a the following folders:
  #outputs/BFS , outputs/A_star, outputs/Djikstra
  names = files()
  name = names[1][3]
  shortcut = True
  #for specific file: name = "bords/boards/1-1.txt" while setting shortcut to False

  algorithm = "A_star" #"A_star", "BFS", "Djikstra"
  t,explored_t,path_found,path = execute_board(name,alg=algorithm,shortcut=shortcut)
  create_gui(algorithm,t,explored_t,path,name,save=True,show=True)

  #To execute all files, show disabled for these
  #execute_all_files(names,"A_star")
  #execute_all_files(names,"Djikstra")
  #execute_all_files(names,"BFS")
```

The execute_all_files execute all files in the boards/boards/ folder with the set algorithm, and saves them to a predefined folder. Hence the folder outputs/A_star/, "outputs/Djikstra" and "outputs/BFS" must exist. See the folder structure of my zipped file for clarification.

Deliverables

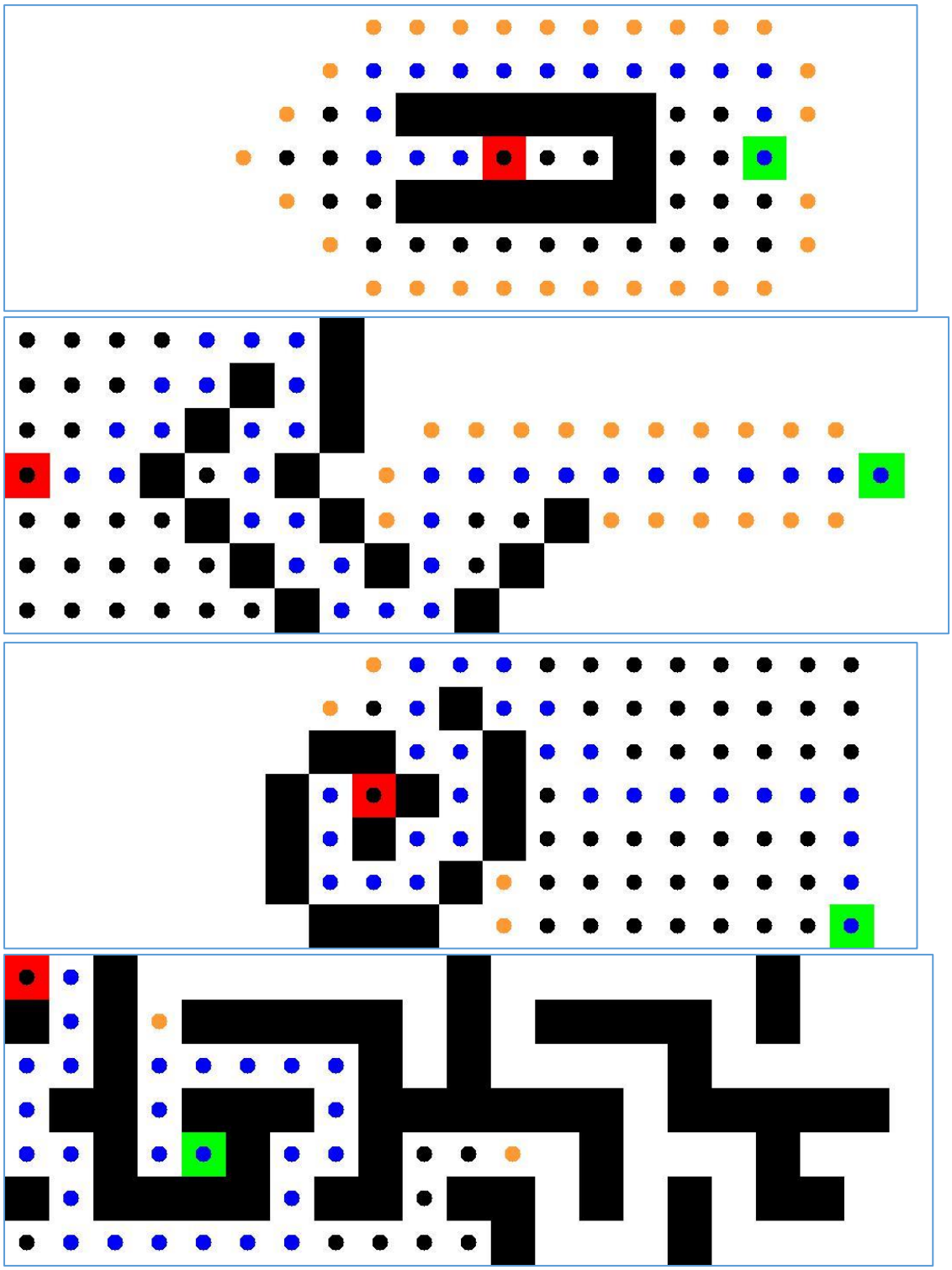
Paths from deliverable 1, 2 and 3. For full output look in the "outputs/A_star/" folder of the zip file.

For the boards with walls the three algorithms are almost identical, but sometimes chooses to differ in the open fields.

In the weighted boards on the other hand, Djikstra and A* always finds the similar routes. BFS only cares about the shortest path, hence takes shortcuts through costly fields and won't find the most optimal paths.

Mikkel Svagård

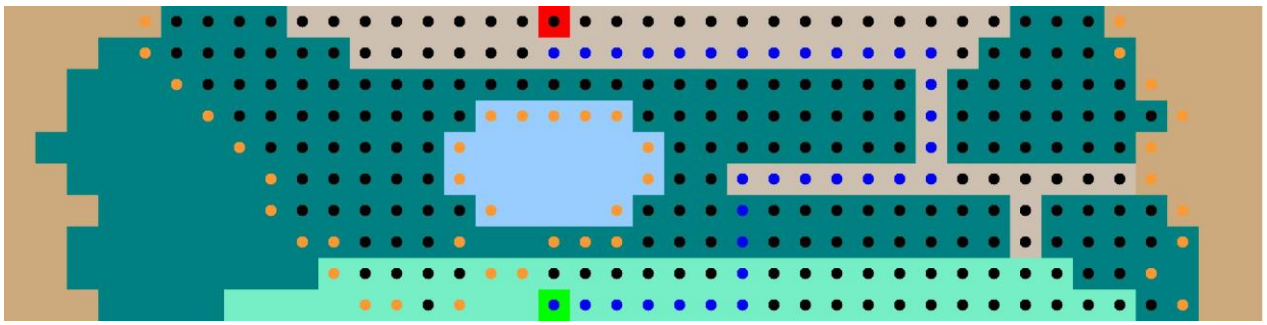
A* paths

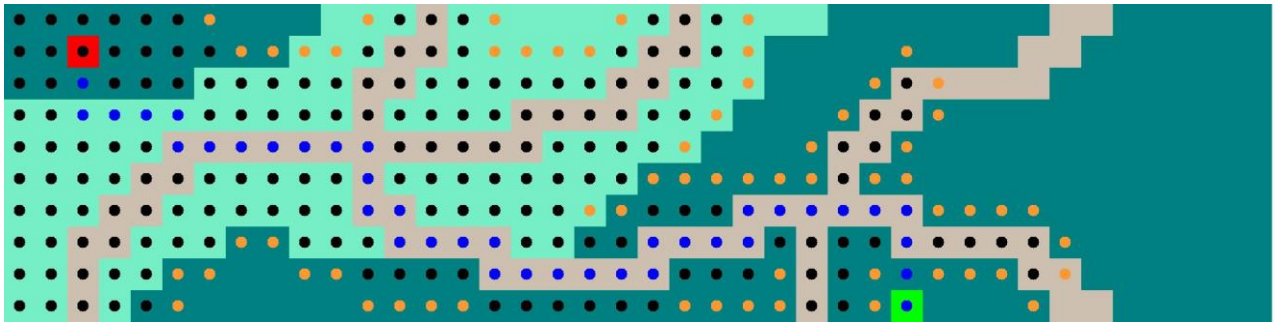


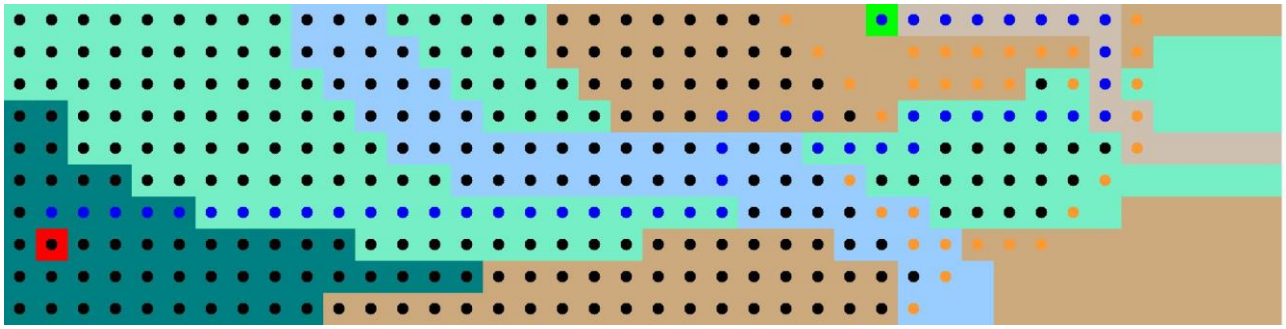
Oving 3 – A*
AI Intro
TDT4136

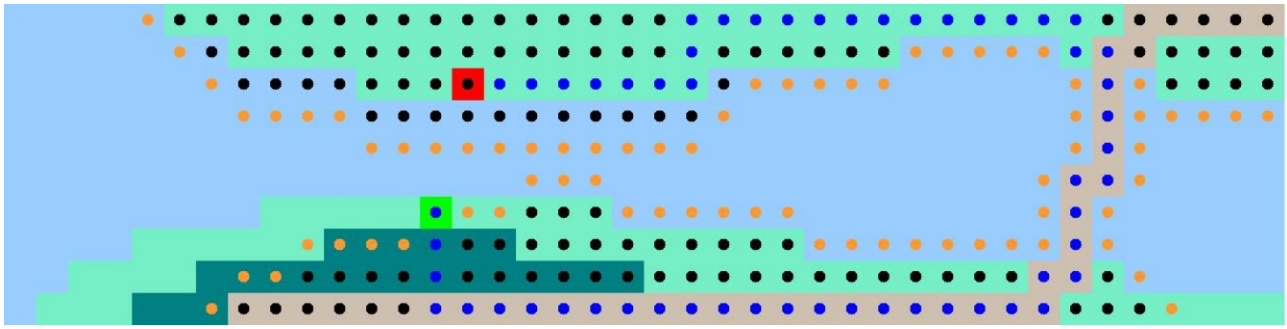
Mikkel Svagård

A* Paths



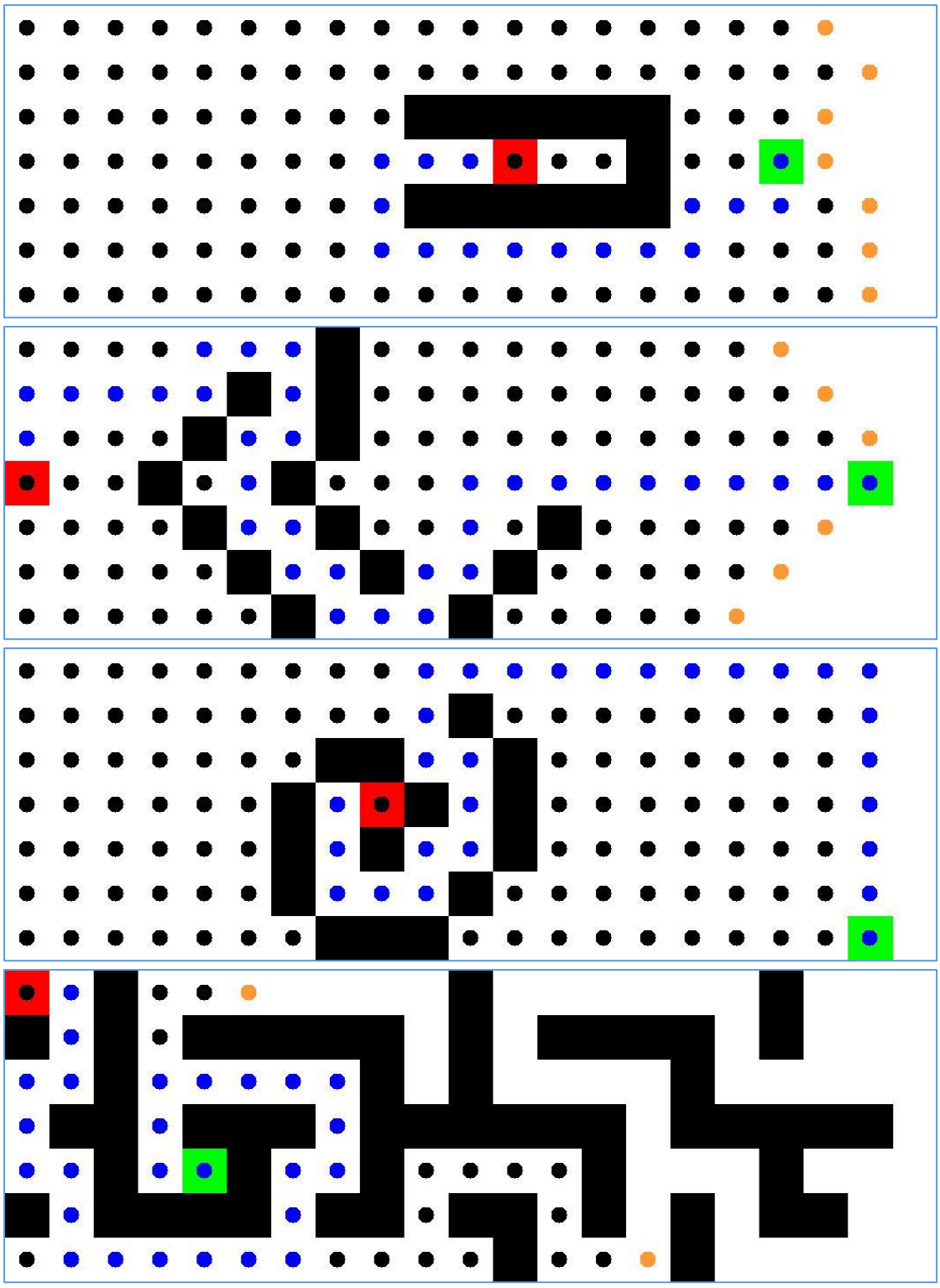






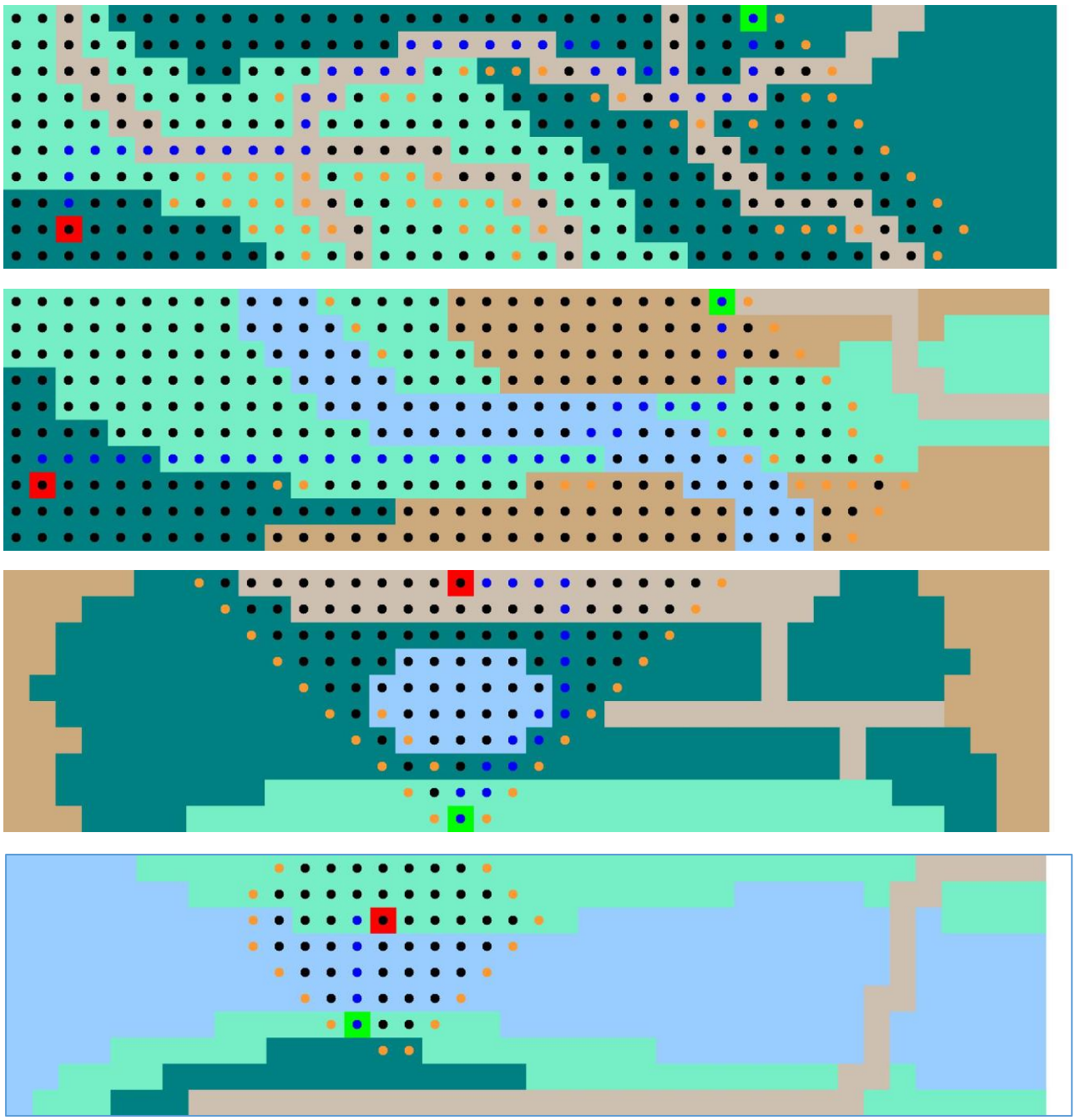
Mikkel Svagård

BFS's Paths



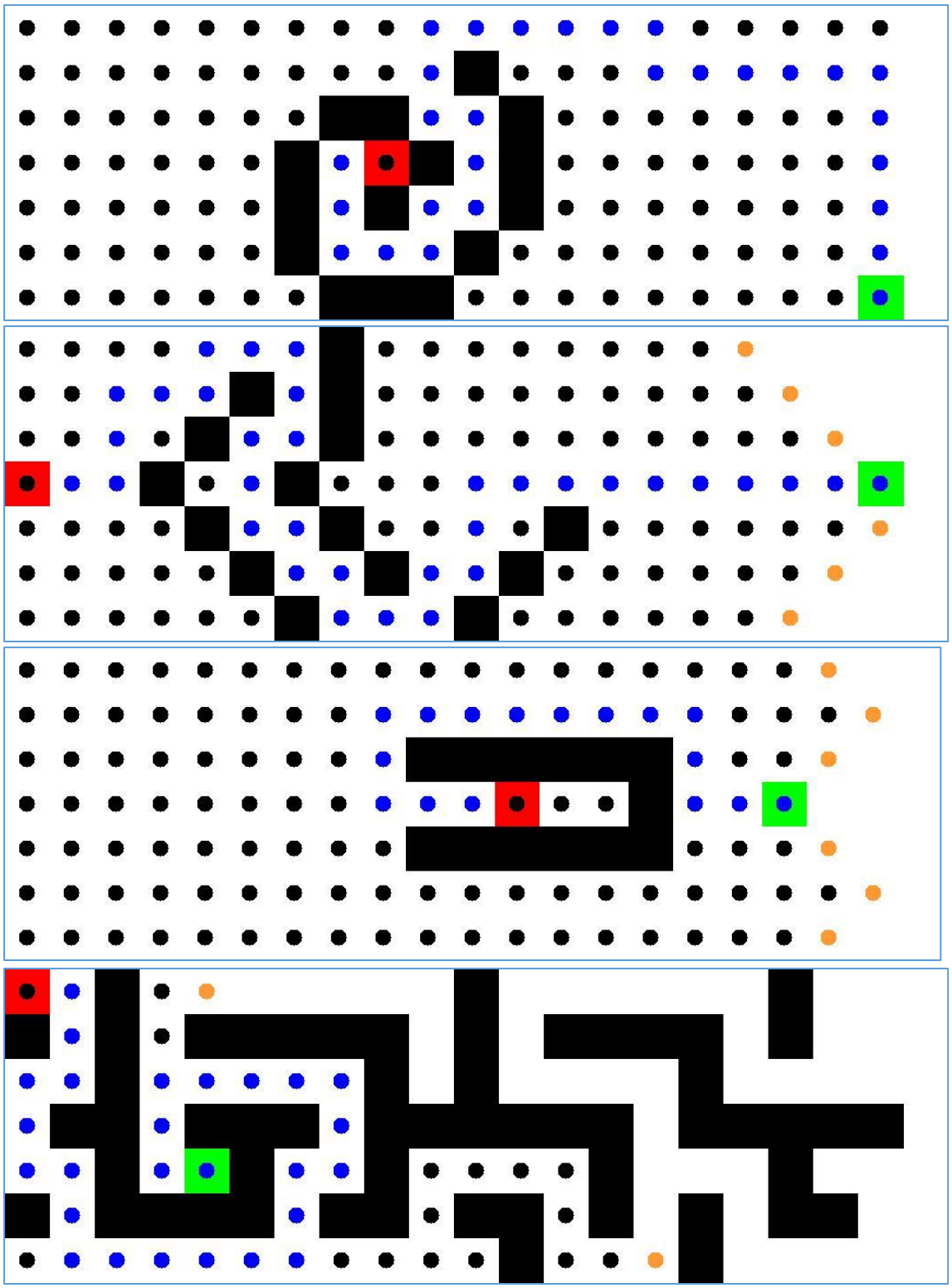
Mikkel Svagård

BFS's Paths



Mikkel Svagård

Djikstra's Paths:



Mikkel Svagård

Djiksta's Paths