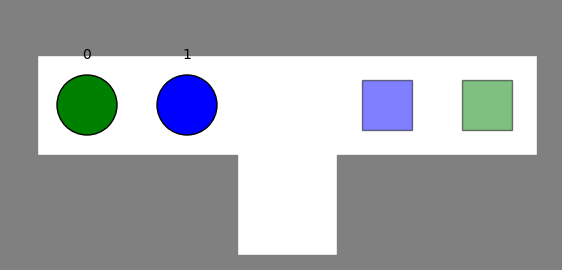
CBS PROJECT

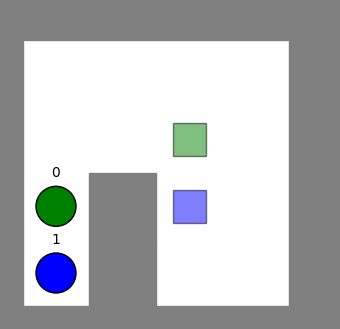
* Design a MAPF instance for which prioritized planning does not find an (optimal or suboptimal) collision-free solution for a given ordering of the agents.  
    
  we can use the classic from the handout. Prioritizing 1 first will imply no solutions to this problem that are collision-free.



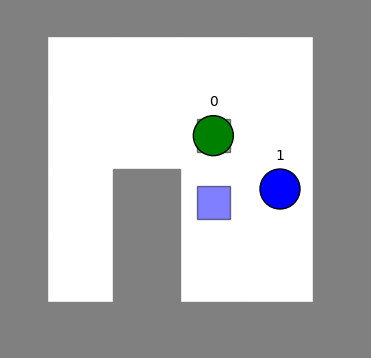
• Design a MAPF instance for which prioritized planning does not find an (optimal or suboptimal) collision-free solution, no matter which ordering of the agents it uses.  
  
we can use this example of some corridor. There will be no solution that is collision free no matter the ordering.

* Design a MAPF instance for which prioritized planning does not find an (optimal or suboptimal) collision-free solution for a given ordering of the agents even if an ordering of the agents exists for which prioritized planning finds an optimal collision-free solution.

The example of the first design question fits here as well. Prioritizing 1 as the top player will imply no solutions as we stated above , but prioritizing agent 0 instead will achieve our desired optimal solution where agent 1 will proceed to the “pit” to let agent 0 pass it as it comes forward to it’s goal.

* Design a MAPF instance for which prioritized planning finds a suboptimal (but not optimal) collision-free solution for a given ordering of the agents even if an ordering of the agents exists for which prioritized planning finds an optimal collision-free solution  
    
  

The above example is a good example for MAPF finding suboptimal solution for one ordering and optimal for other ordering. Prioritizing agent 0 will lead agent 1 to surround agent 0 from it’s right which lead to much more expensive than prioritizing agent 1 which leads to agent 0 moving up to making the way clear for agent 0 towards it’s goal.



Vs

A white paper with blue green and grey circles and a grey background

Description automatically generated

* Design a MAPF instance for which prioritized planning does not find an optimal collision-free solution, no matter which ordering of the agents it uses, even if a collision-free solution exists  
  In the figure below we can see that no matter who we will prioritize eventually both agent will proceed to the middle of the corridor . then the less prioritized when will come back on his path, making a way for the favourite one , and only after it reaches the desired location , the second will come for his. Priortized planning will be not optimal for both optins since the cost of path will be greater than just waiting for the favourite one to pass and only then begin the journey

A white rectangular object with blue green and white squares and circles

Description automatically generated

A white rectangular object with blue green and purple circles

Description automatically generated