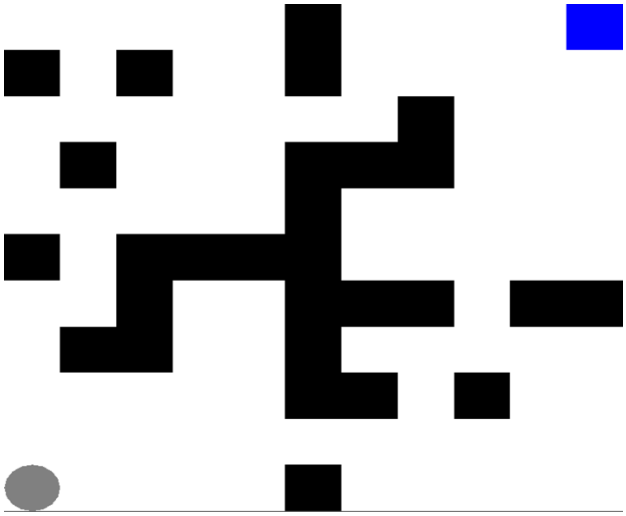
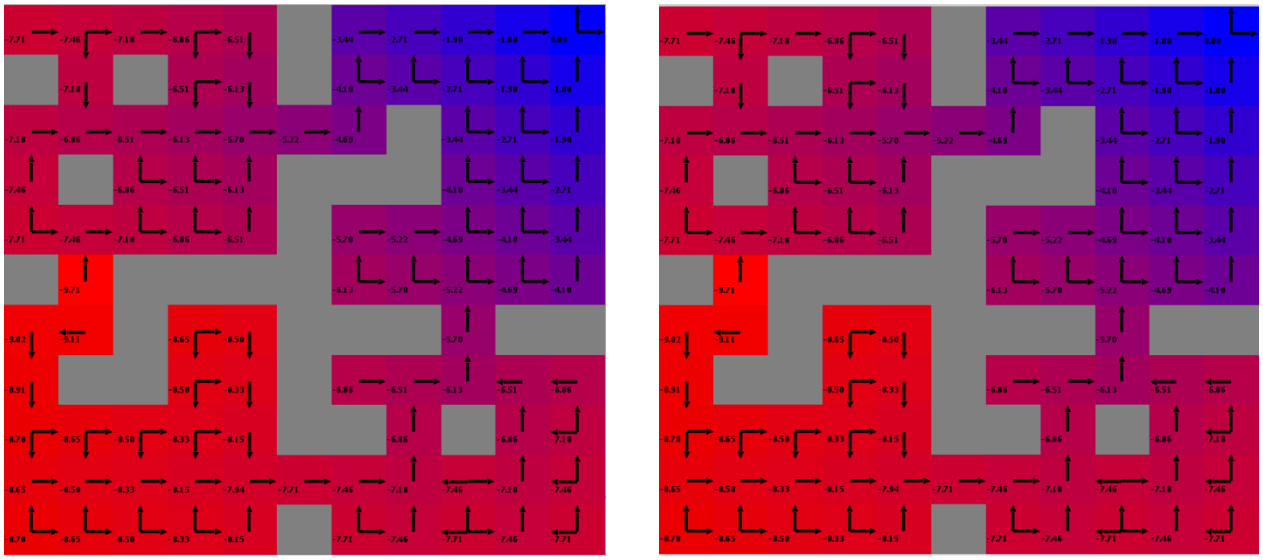
Assignment 4 Report

MDP Description:
Both MDPs are simply gridworlds, where a single agent attempts to find the most optimum policy in traversing through the world. The difference is one has a much larger number of grids, meaning more states to traverse through. This problem can be interesting as a part of training a reinforce learning based agent for videogame AIs. If we can discretize the map of a game we can do reinforcement learning for adaptive enemy or friendly AIs to create a sort of reaction to player actions. Movement would be the first step to this.

Small Gridworld:

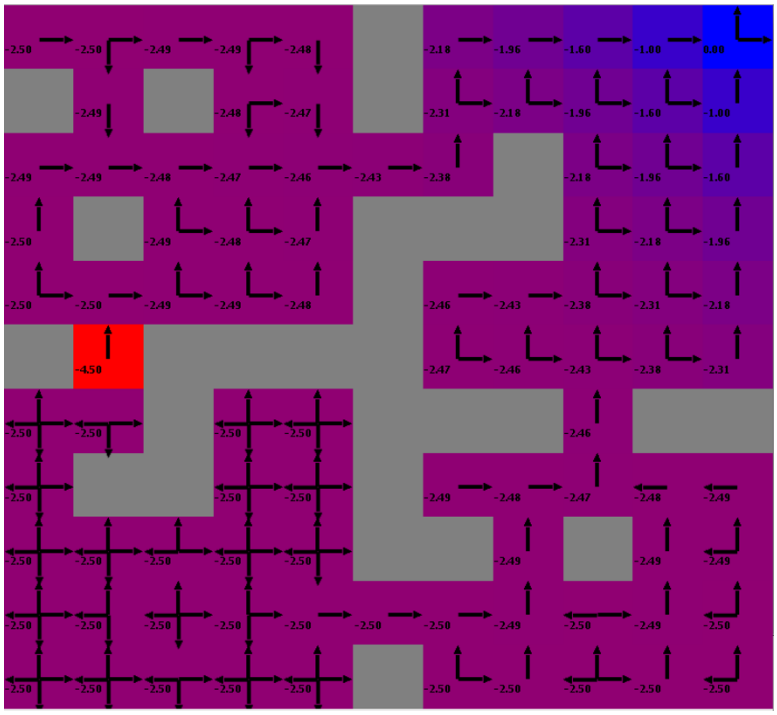


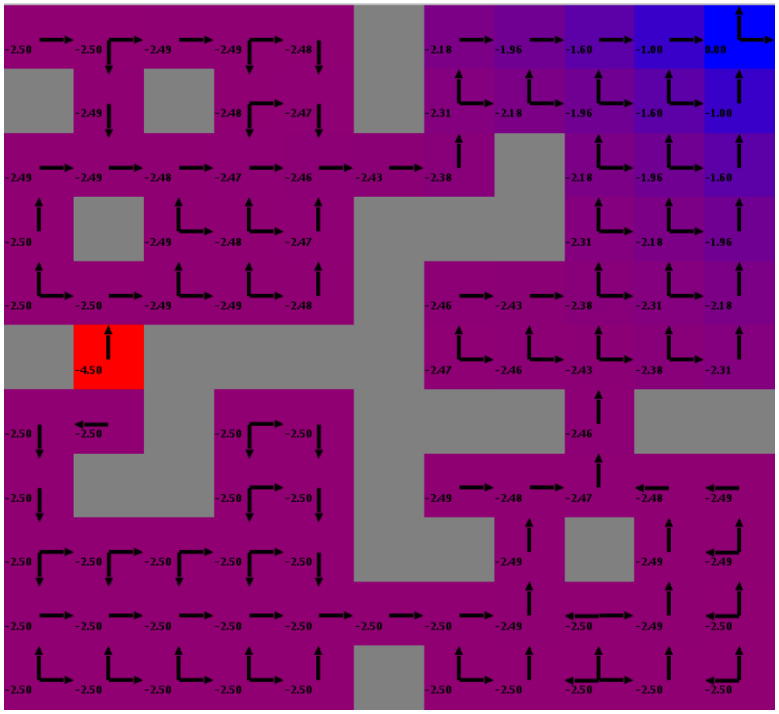
Here is the map of how the gridworld looks. Black tiles are walls, and the blue tile is the terminal state. The rewards given to each tile is -1, and 0 for going out of the terminal state, also the entrance to the left room is given a -3 reward.



On the left is the resulting policy from value iteration and on the right is from policy iteration. It can be seen that there is no difference in either any of the states values or the policies to take. This means that this is already the optimum policy and the algorithms just terminated themselves as no more improvements can be made. Iteration wise, policy iteration did a lot more iterations with 87 iterations, while value iteration did 23 iterations. This is due to the fact that policy iteration did multiple policy improvement each of which runs many iterations

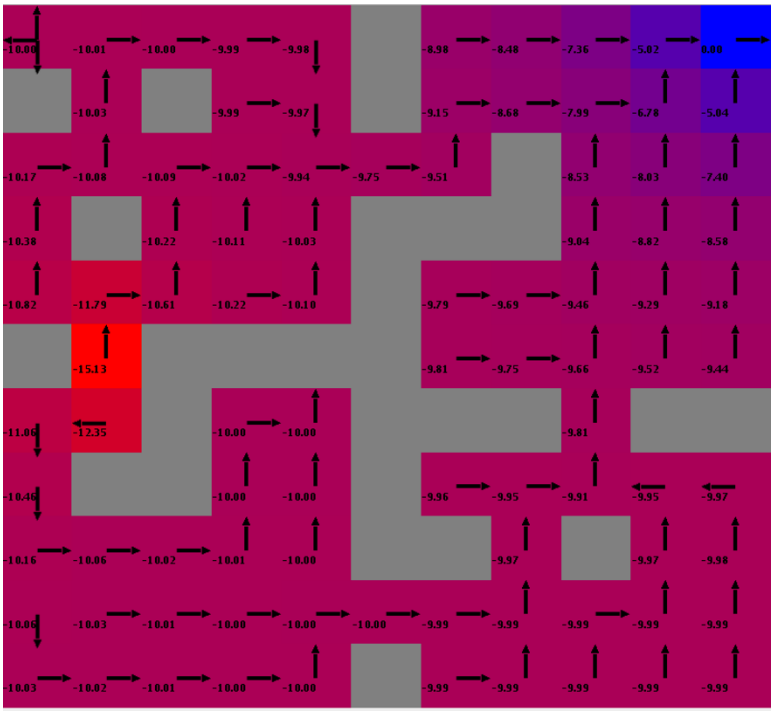
of policy iteration. Also note that this used a large gamma value, which is 0.9, from my experiments, as I lowered the gamma more and more the policy gets less and less optimal. However with the same gamma values, policy iteration has a more optimum policy compared to value iteration.

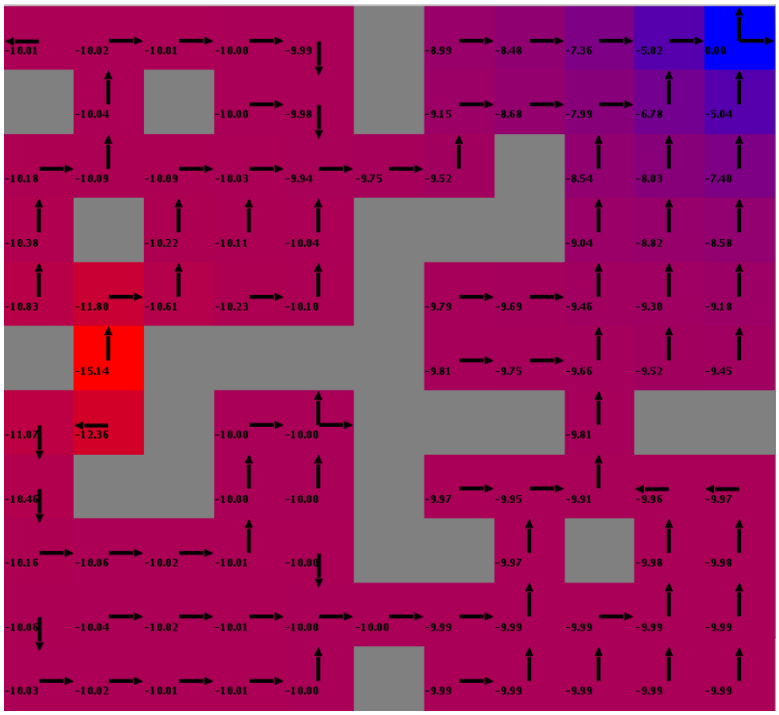




As can be seen, the value iteration results are a lot more confused when it is farther from the reward state. This is due to the reward state being too far and the possible 0 reward being too negligible. Thus near the initial point, the states all seem equally as bad. However due to the policy updating nature of policy iteration, we can still see a clearer policy. Policy iterations still need a larger number of iterations, however for both methods, the amount of iterations needed are far reduced compared to when the gamma is high. This is because the value change is much lower due to the small gamma number and small reward numbers, thus causing it to converge to a value faster. Which can be solved by altering the difference in rewards in the reward function to give a larger living cost (currently living cost is 1, can increase to a larger value). This would result in even larger iteration numbers for large gamma values as the values keep on propagating, resulting in it converging slower.

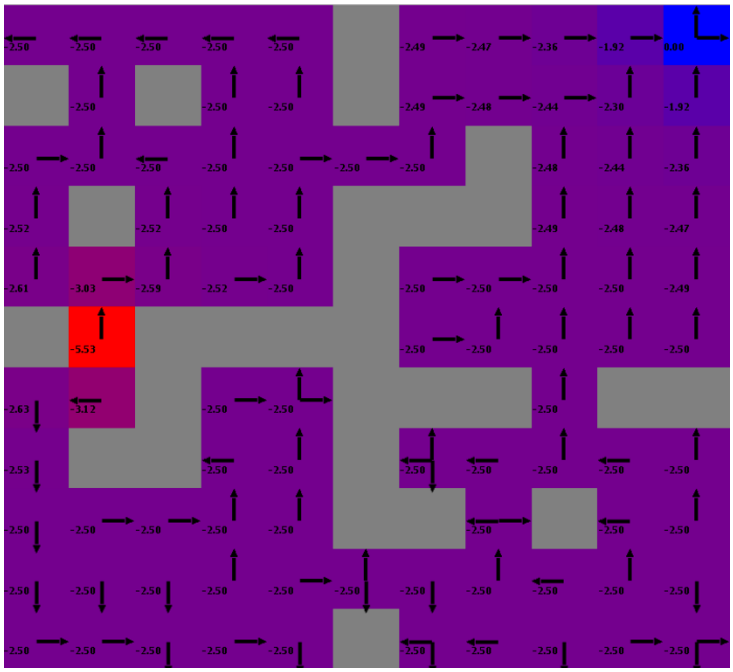
The pictures below denote a situation where I modified the transition function to not be as certain. The tests above were done when the transition function has a 1.0 probability of moving between tiles. Now when the transition probability is lowered to about 0.3 we can see a difference in some of the policies

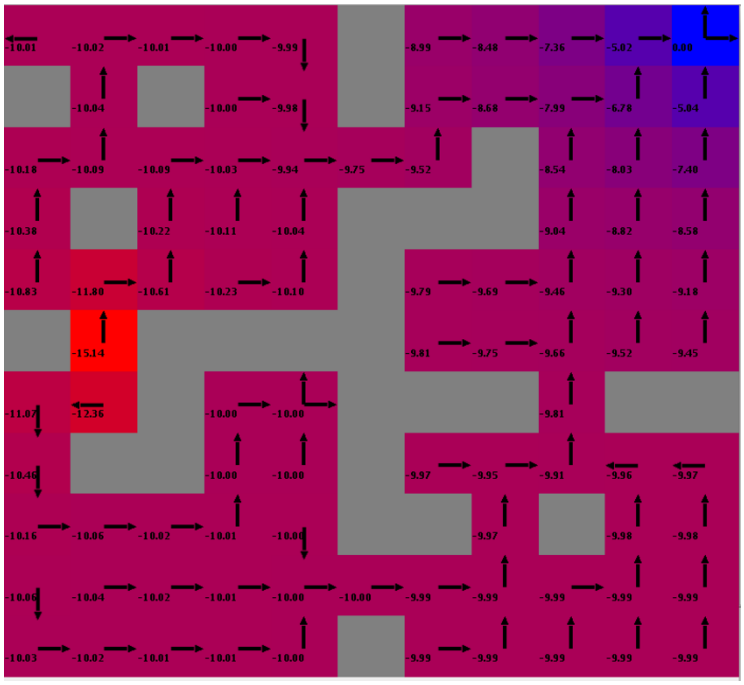




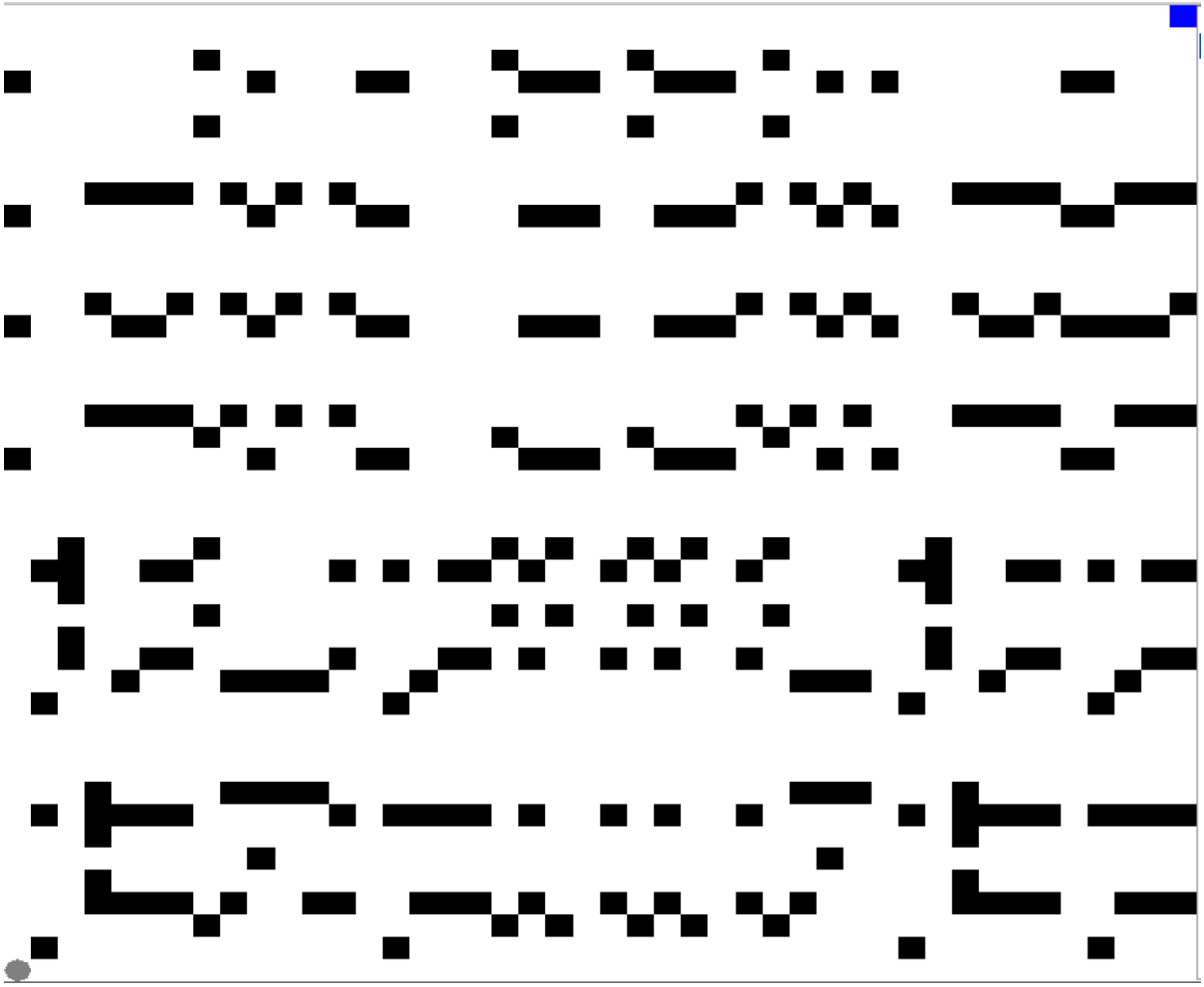
Now here we can still see the policies being fairly similar, but noticeably in states farther from the goal state, the policies are making the agent not go towards the closest exit of the room. It makes the agent move away from the room, this is likely to be caused by the larger probability of the movement failing and the agent ending up in a position where it wasn't going to. However as the terminal state moves closer, the agent simply attempts to reach that terminal state as fast as possible due to the living cost, and is not that affected as the chance of actually ending up where it wants to be in (0.3) is larger than ending up in one of the other states (0.233).

When the discount is adjusted to a smaller number, a similar thing happens with the previous experiment, where the agent will not be able to see how good the terminal state is that is too far away. The behaviors of PI and VI are the same.



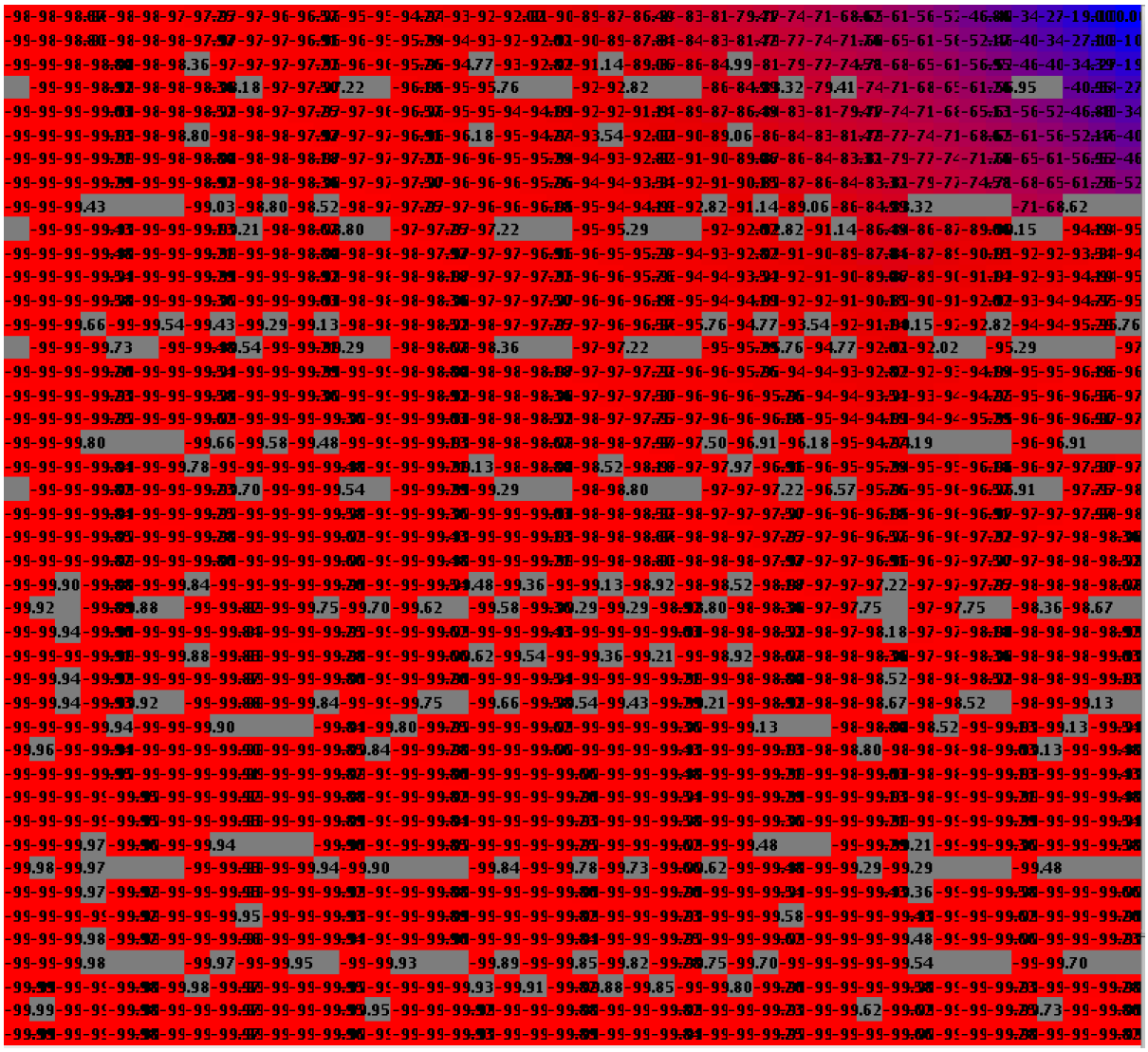


Large Gridworld

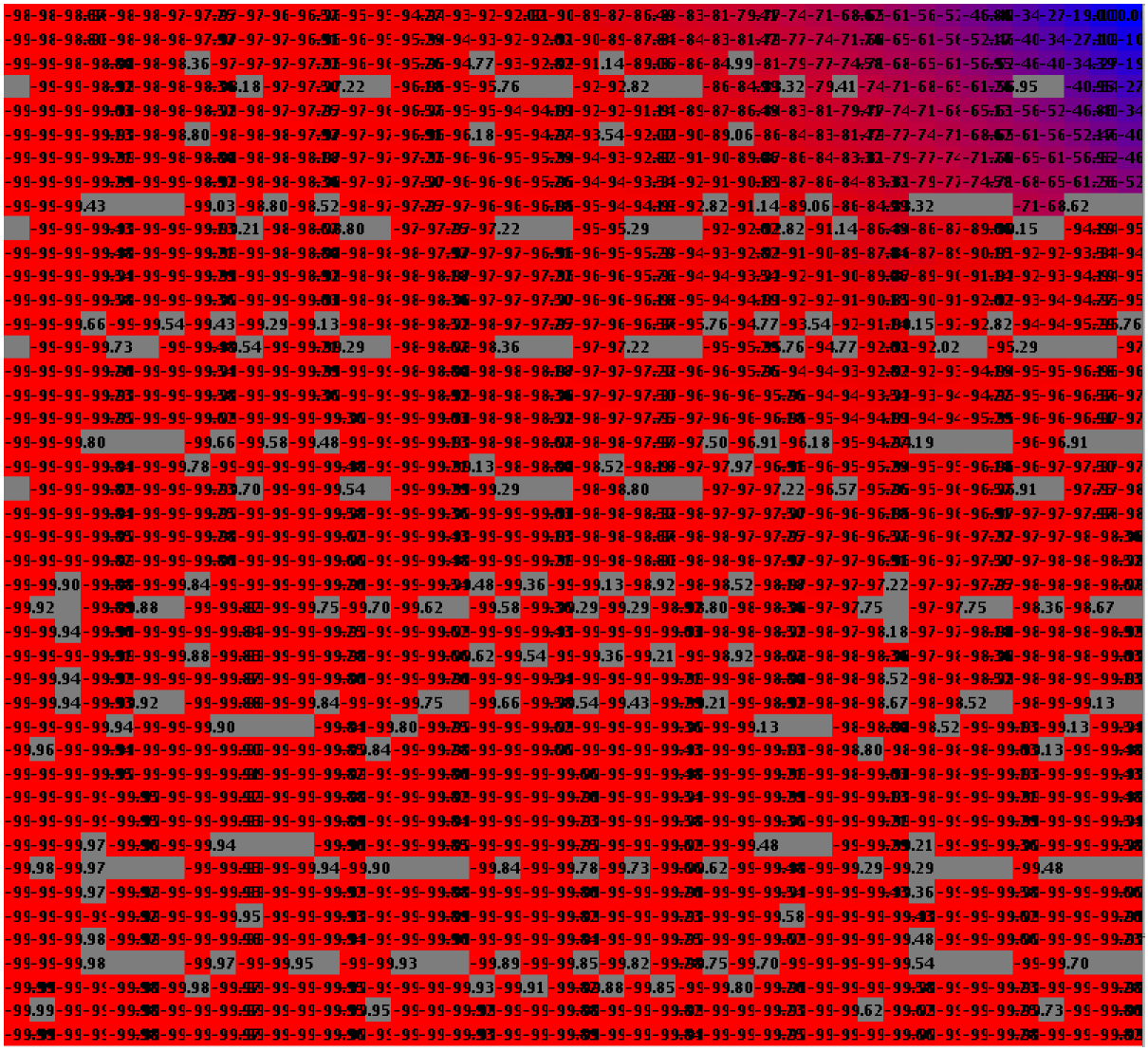


In comparison this is the large gridworld's map, where there are a larger number of grids ant thus the agent has to go around further and more states for the algorithms to keep track of. Note that this gridworld is actual 16 times larger than the small gridworld.

First things first is that there is a considerable amount of time increase for both policy and value iteration, this took around 10 times longer than the small gridworld. Also another thing is that the reward had to be readjusted to a larger value so that the algorithm can actually converge. With a smaller reward value, most of the states have a really really miniscule difference between states and as such the agent just sees every single state as equally as bad, and this is a convergent policy as the values and policies have already converged to the resulting final values.

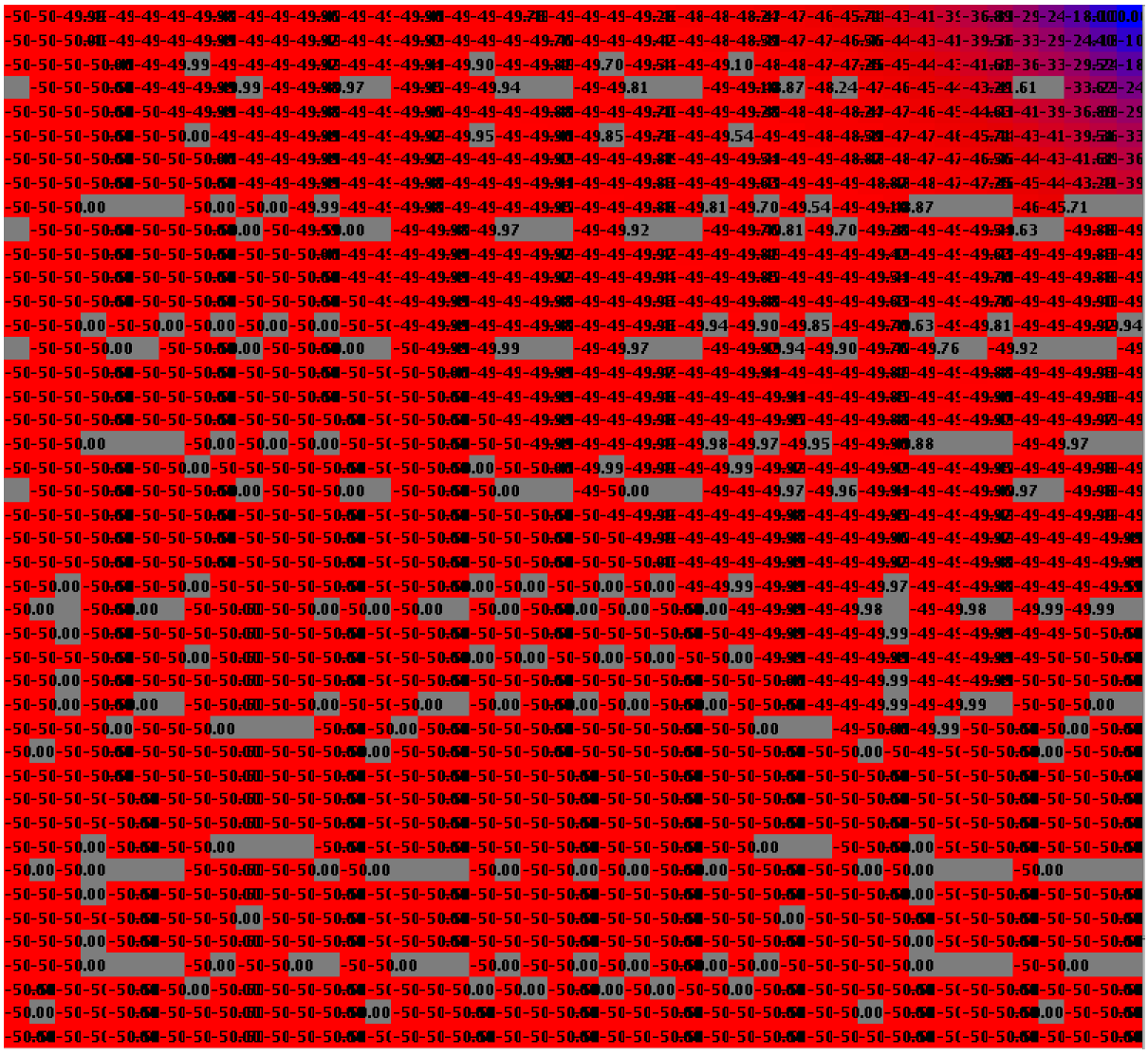


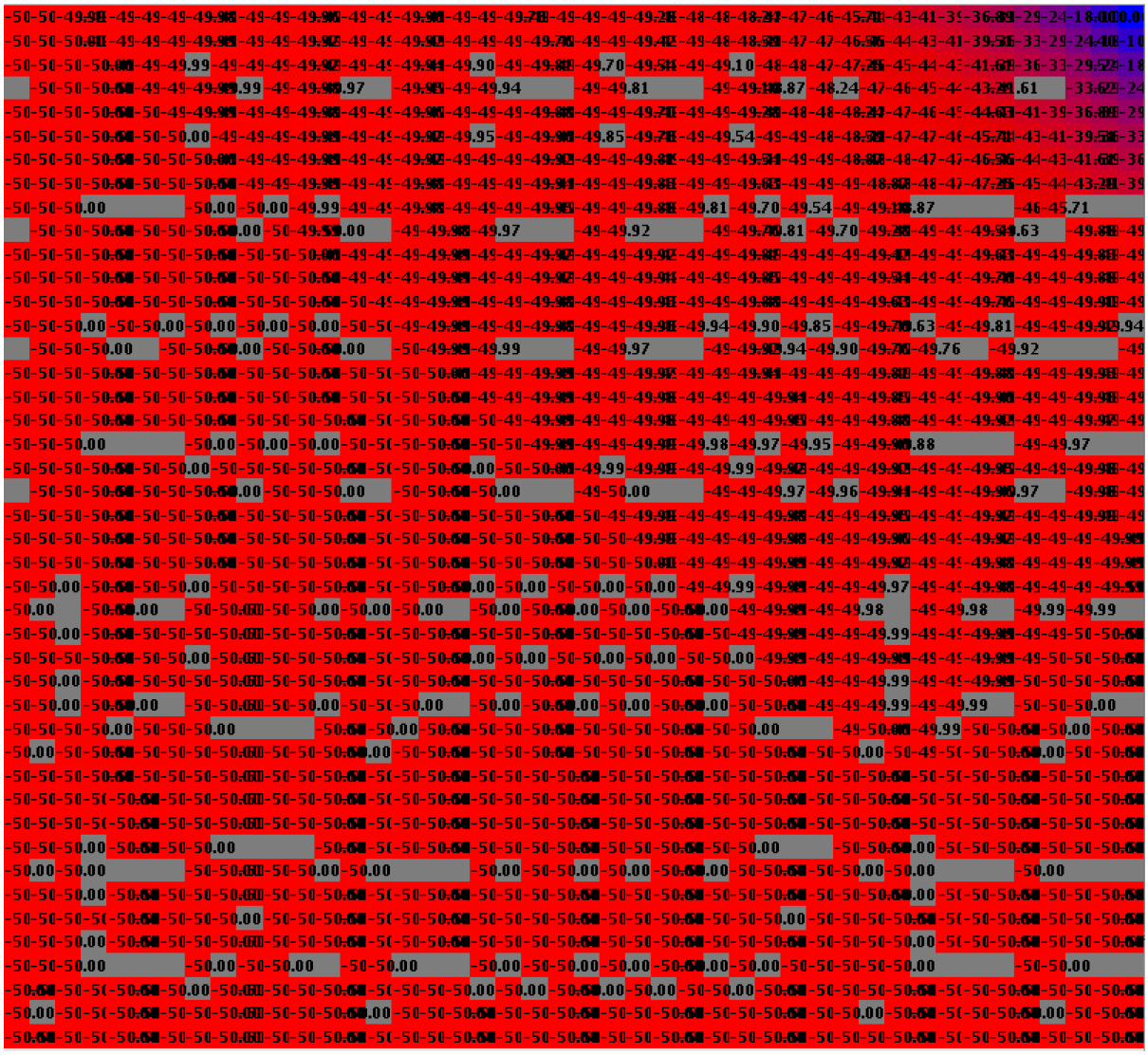
Here there is a considerably much larger amount of states such that I am not able to fully show the policies for every single state but on the above is the results of the value iteration and the values denoted by it, here the agent will simply attempt to do a beeline straight from its' starting position to the terminal state as there are no major obstacles along the way. Which is expected given the reward function which is just a living cost, some small rewards scattered along the grid (which is just a smaller living cost), and the 0 terminal state.



And again for policy iteration, there is not much difference as the resulting policies are basically the same. Except that in policy iteration, the policy does not take the straight middle road as in as close as a straight line to the target as possible. This is probably due to the policy converging faster than the values, such that it is taking a less optimal policy compared to the perfectly optimized values in value iteration. However iteration wise, PI still takes more iterations compared to VI for the same reason as is stated in the small gridworld.

On the previous page, the first one is the result of VI with a significantly lower chance of ending in the target location, and the second one is PI (a modified transition function). As can be seen from the color in the graph, there is a considerable amount of red, and as such this is reflected in the resulting policies of both algorithms. Both simply attempt to move towards whatever direction unless they are really close to the terminal state, as it just sees any direction as equally as bad due to the really low chance of getting anywhere it is trying to go. This is the really extreme case of only 30% chance of getting to the target location, however even with a 80 or 90% chance, there is a lot more confusion regarding to the policy, where states really far away from the final location will have more wandering around in the policy. This might be fixable by increasing the living cost again, further magnifying the difference between the states.





And lastly, the above are the respective graphs for VI and PI when the discount factor is lowered (from 0.9 to a smaller number), notably even with a small decrease, from 0.9 to 0.8, the reward amounts scale down really quickly, resulting in reaching convergence faster due to the really small difference. Thus again causing the issue of each state being equally as bad as its surroundings except when close to the terminal state.

Conclusion:

So a larger state space causes problems in the sense that the reward function needs to be varied as the difference between states will get smaller and smaller as the number of states increase. This can result in less than optimal results due to the implementation of the training algorithms. This can usually be solved by increasing the living cost (or reward of the terminal state depending on implementation, this gridworld is implemented such that when the agent enters the terminal state, it always gets 0 reward, thus to increase the reward of entering the terminal state, we must lower the reward of every other state). The discount factor also matters in the case of how good the policies are, a too low discount will cause the agent to focus too much on local rewards, and a larger discount will help it focus on a more global reward system.