Here we assume that the first 90 games that the agent plays are very random and after that it uses the reinforcement learning to make a proper decision.

Uses an NXN grid with

1. Agent
2. Wumpus
3. 3 pits
4. Treasure/Gold
5. Agent has 3 bullets initially.

Input contains Number of games and Number of iterations:

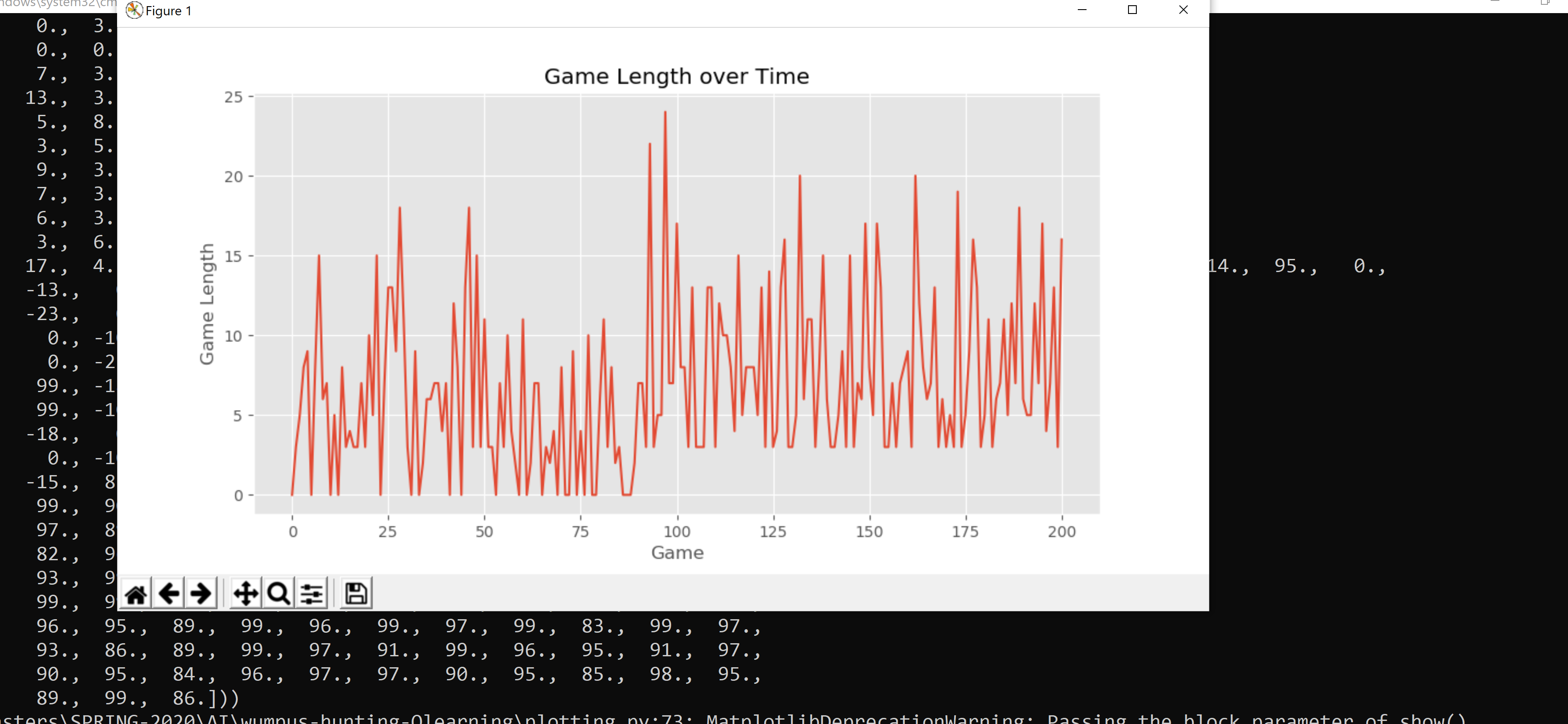
**python environment.py --ngames 150 --niter 40 --gridsize 4 4 --numwumpus 1 --numholes 3 --bullets 3**

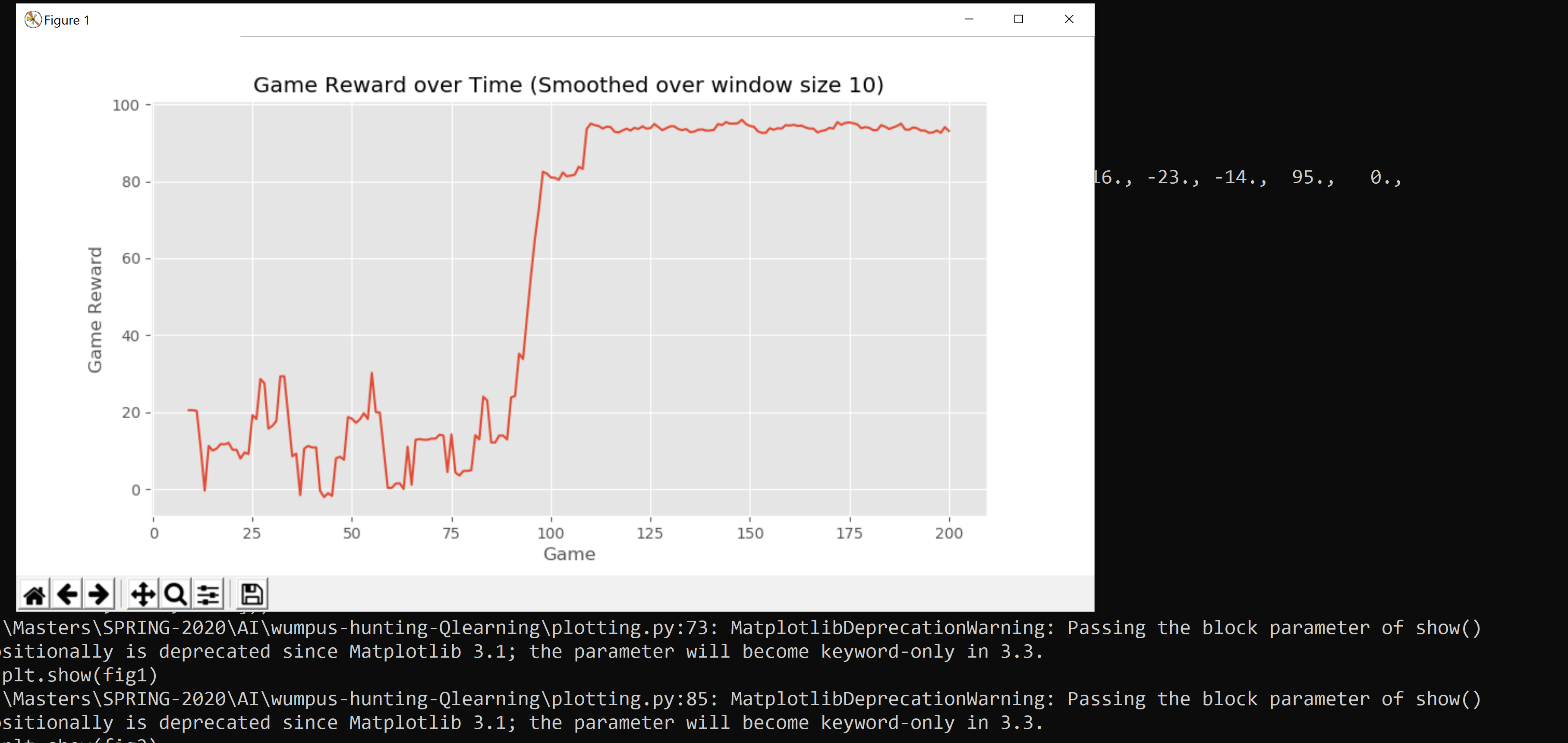
The cumulative reward obtained is 10984.0

On an average considering that we have played 125 games. Average reward per game would be 87.872.

Game length indicates the number of steps that it took to end the game.

End of game includes falling into the pit, getting swallowed by Wumpus or capturing the gold.





The grid structure for the current scenario is(4x4).

|  |  |  |  |
| --- | --- | --- | --- |
| Hole |  |  |  |
|  |  | Agent | Hole |
|  |  | Gold |  |
|  |  | Hole | Wumpus |

We can observe that once the initial 90 games are done the agents reward for every subsequent is approximately equal its previous and next games.

The reward depends on the position of gold and the Agent. In general we can observe that the reward obtained is high once 90 games are finished.

In our situation

**\*** The treasure is found: reward +100 and the game ends **\*** You kill the Wumpus: reward +10 **\*** The Wumpus catches you: reward −10 and the game ends **\*** You fall into a hole: reward −10 and the game ends **\*** Nothing happens: reward −1

It seems that convergence is happening after around 100 games

Execution command:

python environment.py --ngames 150 --niter 40 --gridsize 4 4 --numwumpus 1 --numholes 3 --bullets 3

None of the arguments are compulsory.

ngames indicates are number of games

niter indicates number of iterations

gridsize 4 4 indicates rowsize columnsize

numwumpus indicates of Wumpus in grid

numholes indicate number of holes in the grid

bullets indicate the number of bullets that agent possesses