

Setting up the game

In [1]:

```
from MCTS import MCTS
from Board import Board
import numpy as np
```

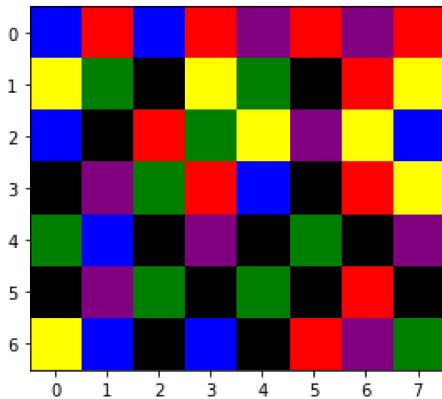
In [5]:

```
# Data from an actual iMessage Filler game
data = np.array([[3,0,3,0,4,0,4,0],[2,1,5,2,1,5,0,2],[3,5,0,1,2,4,2,3],
                 [5,4,1,0,3,5,0,2],[1,3,5,4,5,1,5,4],[5,4,1,5,1,5,0,5],[2,3,5,3,5,0,4,1]])

intelligence_parameter = 0.5
exploration_parameter = 1
num_iterations = 1000
```

In [6]:

```
test = Board(data=data)
test.display_board()
```



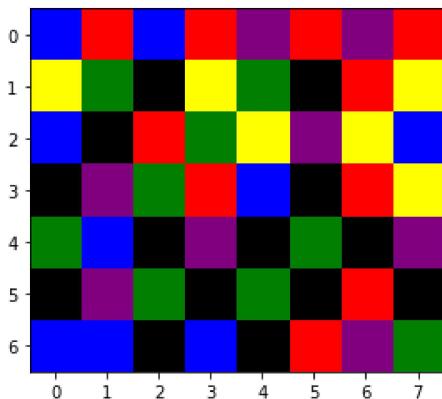
Playing the game

Round 1

In [7]:

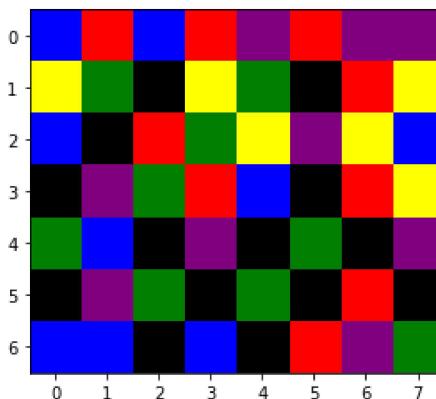
```
%%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

green has win percentage 0.32075471698113206 with an average score of 25.188679245283026
blue has win percentage 0.6682815616984896 with an average score of 30.868623539469965
purple has win percentage 0.3917525773195876 with an average score of 26.278350515463913
black has win percentage 0.4985507246376812 with an average score of 28.385507246376815
CPU times: user 9.75 s, sys: 59.6 ms, total: 9.81 s
Wall time: 9.65 s



In [8]:

```
test.update_board(2,4)
test.display_board()
```

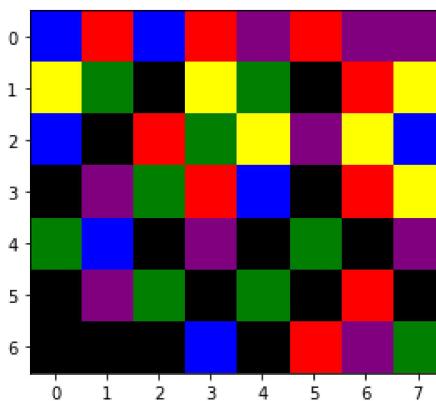


Round 2

In [9]:

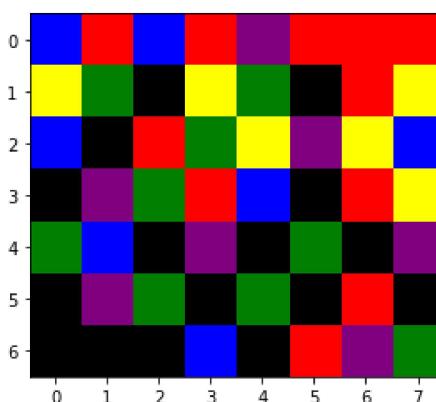
```
%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

red has win percentage 0.4666666666666667 with an average score of 27.644444444444424
green has win percentage 0.3508771929824561 with an average score of 24.94736842105263
yellow has win percentage 0.36923076923076925 with an average score of 25.215384615384615
black has win percentage 0.7139732020782061 with an average score of 31.337708504238503
CPU times: user 10.1 s, sys: 21.2 ms, total: 10.1 s
Wall time: 10 s



In [10]:

```
test.update_board(2,0)
test.display_board()
```



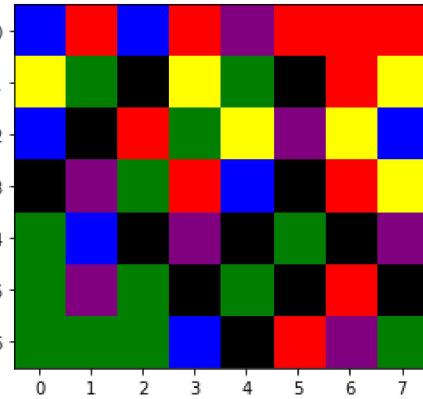
Round 3

In [11]:

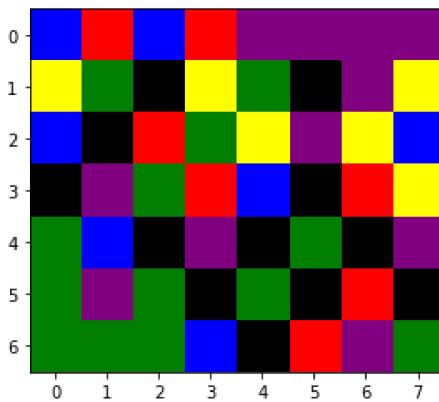
```
%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

green has win percentage 0.7490226739640344 with an average score of 31.792285639822847
yellow has win percentage 0.41509433962264153 with an average score of 27.000000000000007
blue has win percentage 0.49504950495049505 with an average score of 27.76237623762377

```
purple has win percentage 0.07692307692307693 with an average score of 20.076923076923077
CPU times: user 10.3 s, sys: 12.8 ms, total: 10.3 s
Wall time: 10.3 s
```



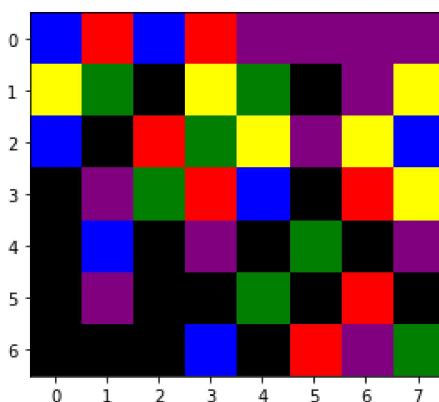
```
In [12]: test.update_board(2,4)
test.display_board()
```



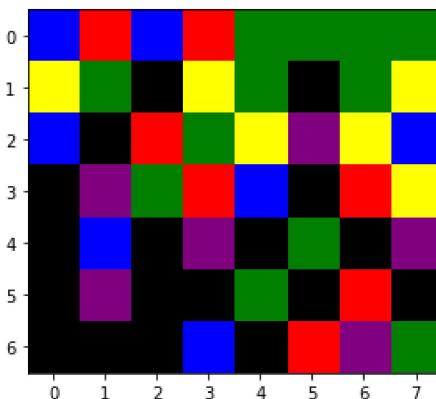
Round 4

```
In [13]: %%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

```
red has win percentage 0.4146341463414634 with an average score of 26.121951219512198
yellow has win percentage 0.36 with an average score of 23.68
blue has win percentage 0.3448275862068966 with an average score of 26.17241379310345
black has win percentage 0.8649270913277053 with an average score of 33.27858787413639
CPU times: user 9.46 s, sys: 67.2 ms, total: 9.52 s
Wall time: 9.39 s
```



```
In [14]: test.update_board(2,1)
test.display_board()
```

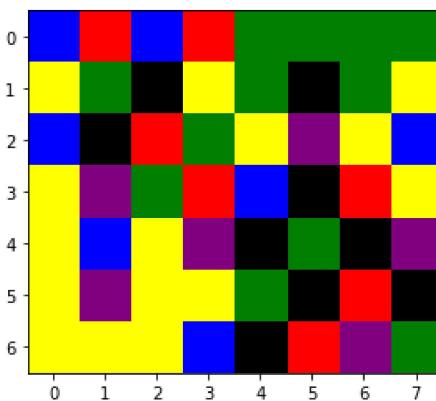


Round 5

In [15]:

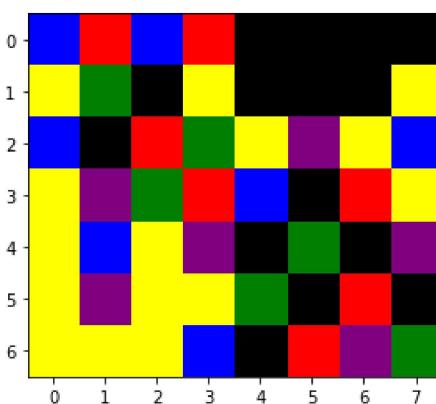
```
%%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

red has win percentage 0.32 with an average score of 26.480000000000004
 yellow has win percentage 0.8999708369787109 with an average score of 33.05278506853295
 blue has win percentage 0.6947040498442367 with an average score of 30.816199376947008
 purple has win percentage 0.6681222707423581 with an average score of 30.66812227074236
 CPU times: user 8.62 s, sys: 48.1 ms, total: 8.67 s
 Wall time: 8.58 s



In [16]:

```
test.update_board(2,5)
test.display_board()
```



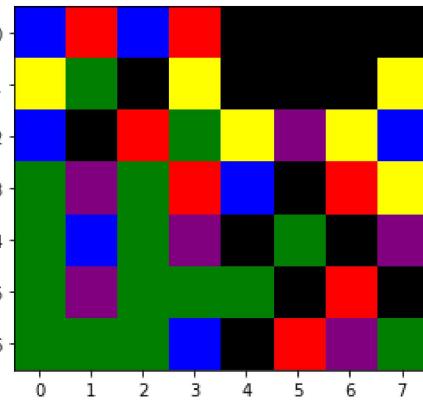
Round 6

In [17]:

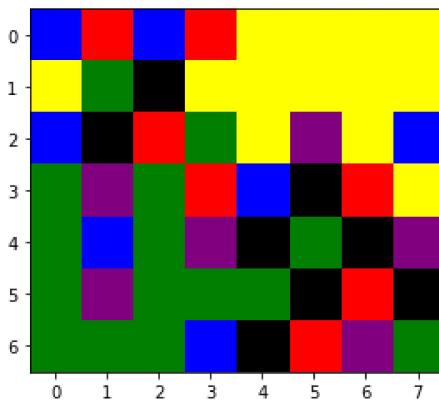
```
%%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

red has win percentage 0.42857142857142855 with an average score of 27.380952380952387
 green has win percentage 0.9410407587798001 with an average score of 33.96949500128179
 blue has win percentage 0.5172413793103449 with an average score of 28.137931034482754

```
purple has win percentage 0.6226415094339622 with an average score of 29.47169811320755
CPU times: user 7.71 s, sys: 40.8 ms, total: 7.76 s
Wall time: 7.65 s
```

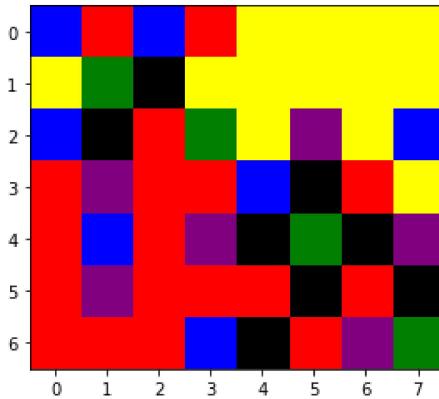


```
In [18]: test.update_board(2,2)
test.display_board()
```

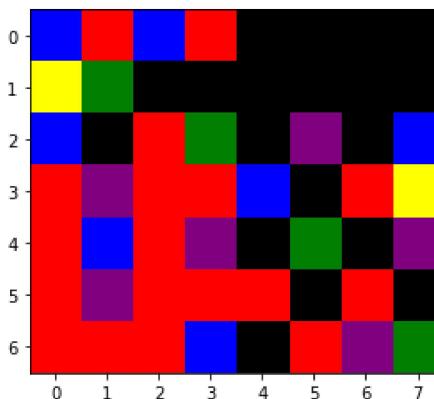


Round 7

```
red has win percentage 0.9654996166624074 with an average score of 33.871454127268294
blue has win percentage 0.6097560975609756 with an average score of 28.975609756097573
purple has win percentage 0.48 with an average score of 27.28
black has win percentage 0.52 with an average score of 29.04
CPU times: user 6.48 s, sys: 74.6 ms, total: 6.55 s
Wall time: 6.42 s
```



```
In [20]: test.update_board(2,5)
test.display_board()
```

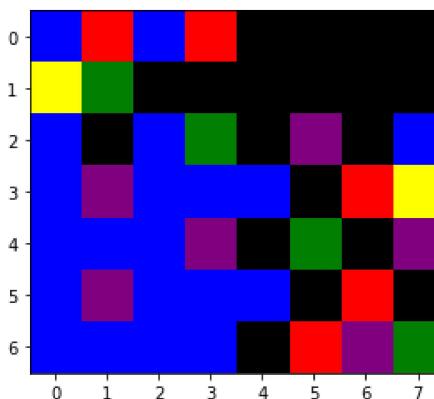


Round 8

In [21]:

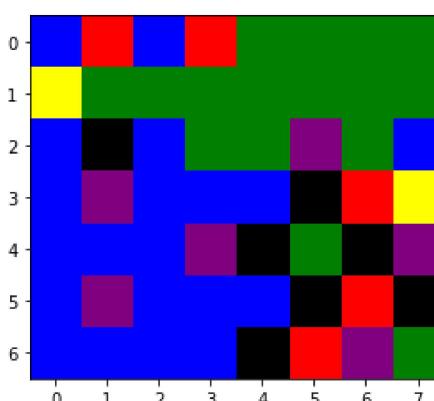
```
%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

green has win percentage 0.5405405405405406 with an average score of 28.21621621621622
 yellow has win percentage 0.6491228070175439 with an average score of 28.947368421052623
 blue has win percentage 0.9517081941947084 with an average score of 33.16516825070669
 purple has win percentage 0.4117647058823529 with an average score of 25.64705882352941
 CPU times: user 4.54 s, sys: 98.4 ms, total: 4.64 s
 Wall time: 4.43 s



In [22]:

```
test.update_board(2,1)
test.display_board()
```



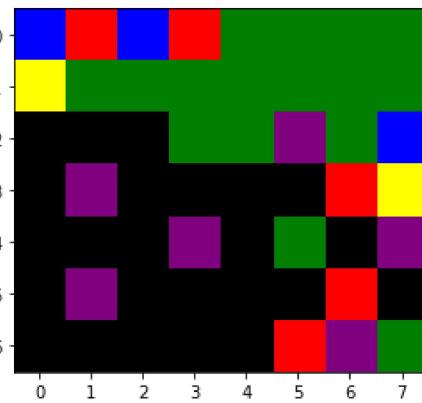
Round 9

In [23]:

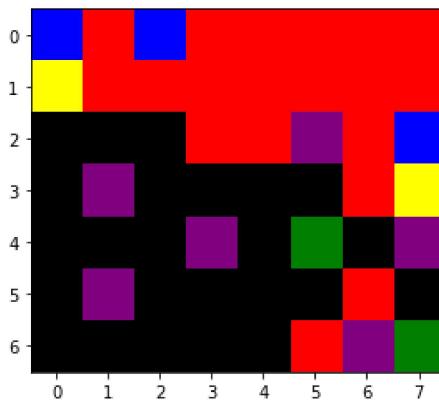
```
%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

red has win percentage 0.7883211678832117 with an average score of 30.941605839416066
 yellow has win percentage 0.7283950617283951 with an average score of 30.1358024691358
 purple has win percentage 0.7692307692307693 with an average score of 30.316239316239308

```
black has win percentage 0.9577541564458981 with an average score of 32.78550013627719
CPU times: user 4.19 s, sys: 59.4 ms, total: 4.25 s
Wall time: 4.13 s
```



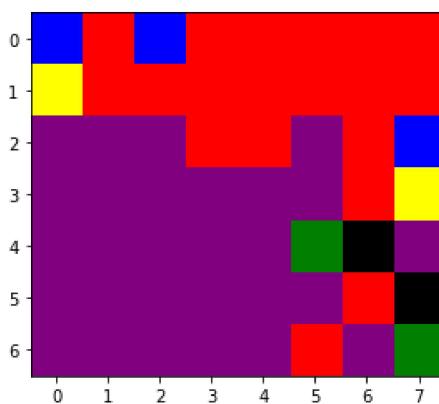
```
In [24]: test.update_board(2,0)
test.display_board()
```



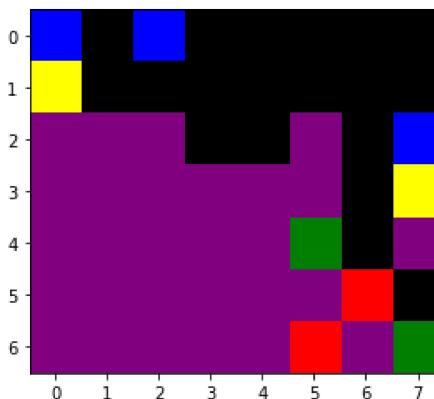
Round 10

```
%%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

```
green has win percentage 0.9477806788511749 with an average score of 31.90339425587459
yellow has win percentage 0.9531123686337915 with an average score of 31.814874696847056
blue has win percentage 0.8638132295719845 with an average score of 31.206225680933866
purple has win percentage 0.9551800146950772 with an average score of 31.86260102865516
CPU times: user 3.54 s, sys: 21.8 ms, total: 3.57 s
Wall time: 3.5 s
```



```
In [26]: test.update_board(2,5)
test.display_board()
```

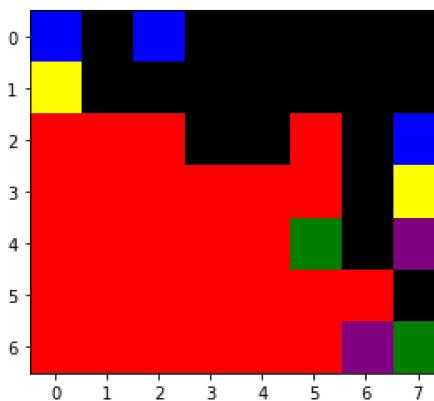


Round 11

In [27]:

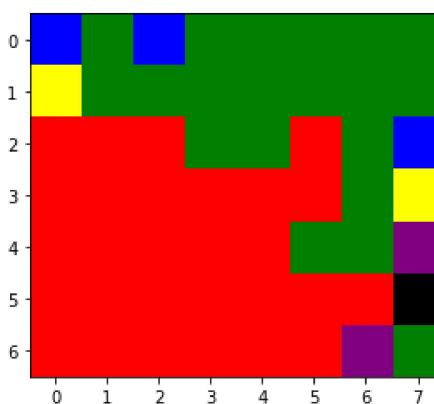
```
%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

red has win percentage 0.9989410518884575 with an average score of 31.6201906106660412
green has win percentage 0.9193154034229829 with an average score of 31.134474327628396
yellow has win percentage 0.9520451339915373 with an average score of 31.210155148095915
blue has win percentage 0.660377358490566 with an average score of 29.47169811320755
CPU times: user 3.07 s, sys: 61.5 ms, total: 3.13 s
Wall time: 3.03 s



In [28]:

```
test.update_board(2,1)
test.display_board()
```



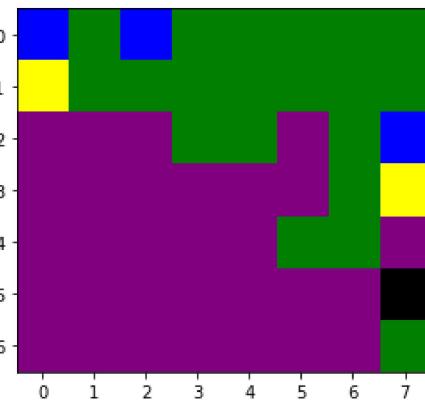
Round 12

In [29]:

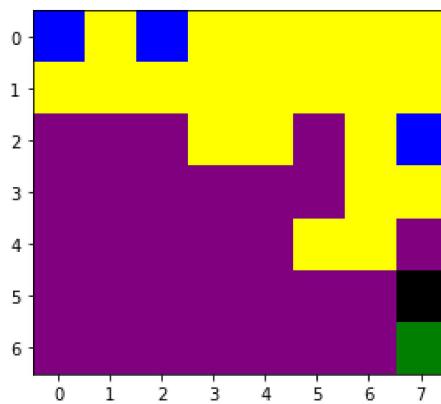
```
%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

yellow has win percentage 0.9991645781119465 with an average score of 31.934001670843607
blue has win percentage 0.9890859481582538 with an average score of 31.19508867667123
purple has win percentage 0.9989047097480832 with an average score of 31.266155531215826

```
black has win percentage 0.9991386735572783 with an average score of 31.861326442721577
CPU times: user 2.75 s, sys: 82.4 ms, total: 2.83 s
Wall time: 2.71 s
```



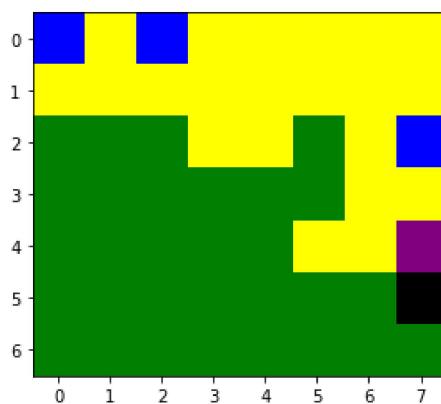
```
In [30]: test.update_board(2,2)
test.display_board()
```



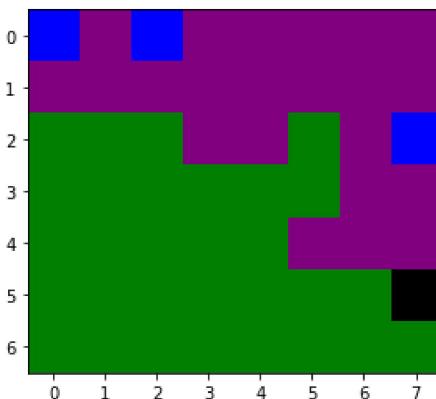
Round 13

```
In [31]: %%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

```
red has win percentage 0.9989281886387996 with an average score of 30.70418006430879
green has win percentage 0.9989550679205852 with an average score of 30.769070010449415
blue has win percentage 0.9988801791713325 with an average score of 30.5856662933932
black has win percentage 0.9991809991809992 with an average score of 31.35380835380846
CPU times: user 2.15 s, sys: 44.6 ms, total: 2.19 s
Wall time: 2.13 s
```



```
In [32]: test.update_board(2,4)
test.display_board()
```

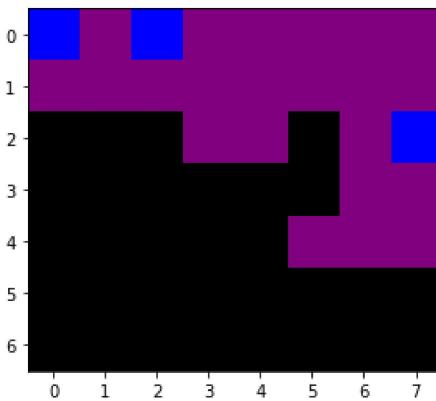


Round 14

In [33]:

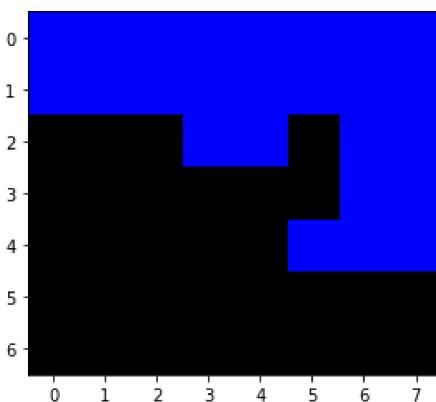
```
%time
tester = MCTS(test, 1, exploration_parameter = exploration_parameter, intelligence_parameter = intelligence_parameter)
test.update_board(1, tester.select_move(num_iterations = num_iterations, verbose = True))
test.display_board()
```

red has win percentage 0.9989417989417989 with an average score of 30.36719576719571
yellow has win percentage 0.9989327641408752 with an average score of 30.350053361792877
blue has win percentage 0.9988998899889989 with an average score of 30.27062706270619
black has win percentage 0.9991755976916735 with an average score of 30.97444352844233
CPU times: user 1.7 s, sys: 44.1 ms, total: 1.75 s
Wall time: 1.7 s



In [34]:

```
test.update_board(2,3)
test.display_board()
```



Final Score

In [35]:

```
test.get_score()
```

Out[35]: (31, 25)