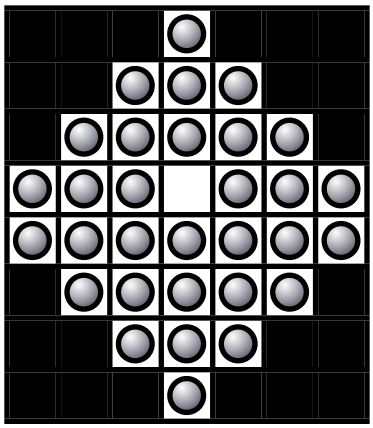
COMP 2355
Winter 2012
Homework #1
Due at midnight, January 31, 2012

For this assignment you need to implement the functions for simulating Peg Solitaire and then perform a Breadth-First Search (BFS) on the game counting the legal states and finding the best solution. The board we will use has 32 locations, and is shown to the right (see next page for more). The full rules for the game can be found on wikipedia:



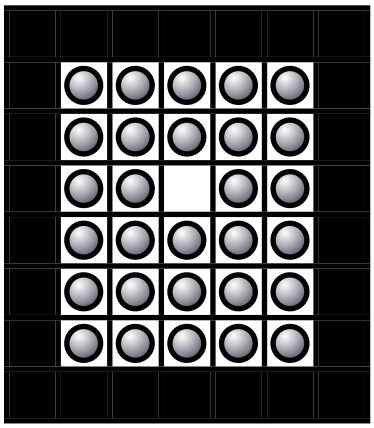
http://en.wikipedia.org/wiki/Peg_solitaire

Your solution should be submitted via SVN in a folder called "HW1".

Suggested functions for the homework are listed below:

```
struct PegState {
      bool board[56];
};

struct PegAction {
      int from, to, middle;
};

PegState GetStartState();
uint32_t GetHashFromState(PegState s);
PegState GetStateFromHash(uint32_t hash);
uint32_t GetMaxHashValue();
vector<PegAction> GetLegalActions(PegState s);
PegState ApplyAction(PegState s, PegAction a);

void BFS()
{
      // initialize memory, 1 byte per state, to store the depth of the state

      // reset all memory to -1 (unseen)

      // Set the start state to depth 0

      // Loop through all states; any at the current depth should:
      //   * Have their legal actions generated
      //   * Find all possible successor states
      //   * Write the depth of the successors
      // Repeat the loop until no states are updated in the loop
      // Print statistics about the search
}
```

Some students might have computers with only 2 or 4 GB or RAM. (Most likely 4GB.) The approach we are using for the assignment requires 4GB of *free* RAM, so might not run well on all students' computers. (We will deal with this issue by assignment 3.) The board on the right only requires 1GB of RAM to run (30 spaces), and so this can be used if your computer has insufficient RAM.



Depth	States at Depth (Board 1 - 32 locations)	States at Depth (Board 2 - 30 locations)
0	1	1
1	4	4
2	17	20
3	83	95
4	390	435
5	1,773	1,883
6	7,323	7,712
7	27,405	28,574
8	92,219	94,559
9	274,351	277,778
10	716,180	712,683
11	1,635,337	1,576,725
12	3,247,371	2,973,318
13	5,608,786	4,730,969
14	8,410,884	6,300,614
15	10,933,456	6,991,763
16	12,329,090	6,473,524
17	12,049,268	5,031,301
18	10,185,731	3,309,853
19	7,445,245	1,856,508
20	4,694,013	890,344
21	2,537,608	365,375
22	1,177,741	127,539
23	465,865	36,980
24	154,626	8,790
25	44,077	1,764
26	10,231	268
27	1,766	20
28	327	2
29	18	
30	5	