Homework 1

Implementing Search to solve a "Challenge"

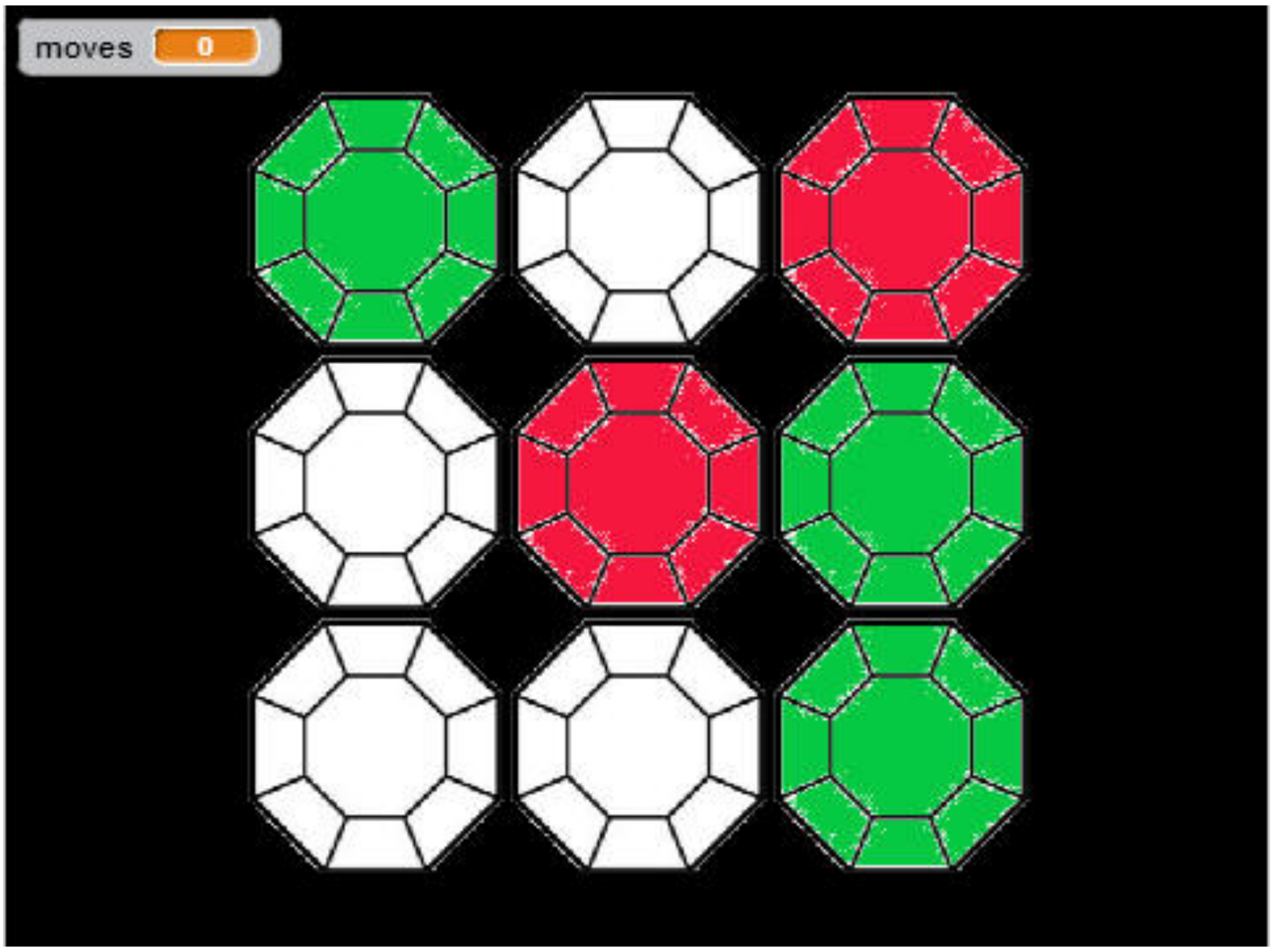
Due: Monday, September 17th, 10:00 AM

Introduction

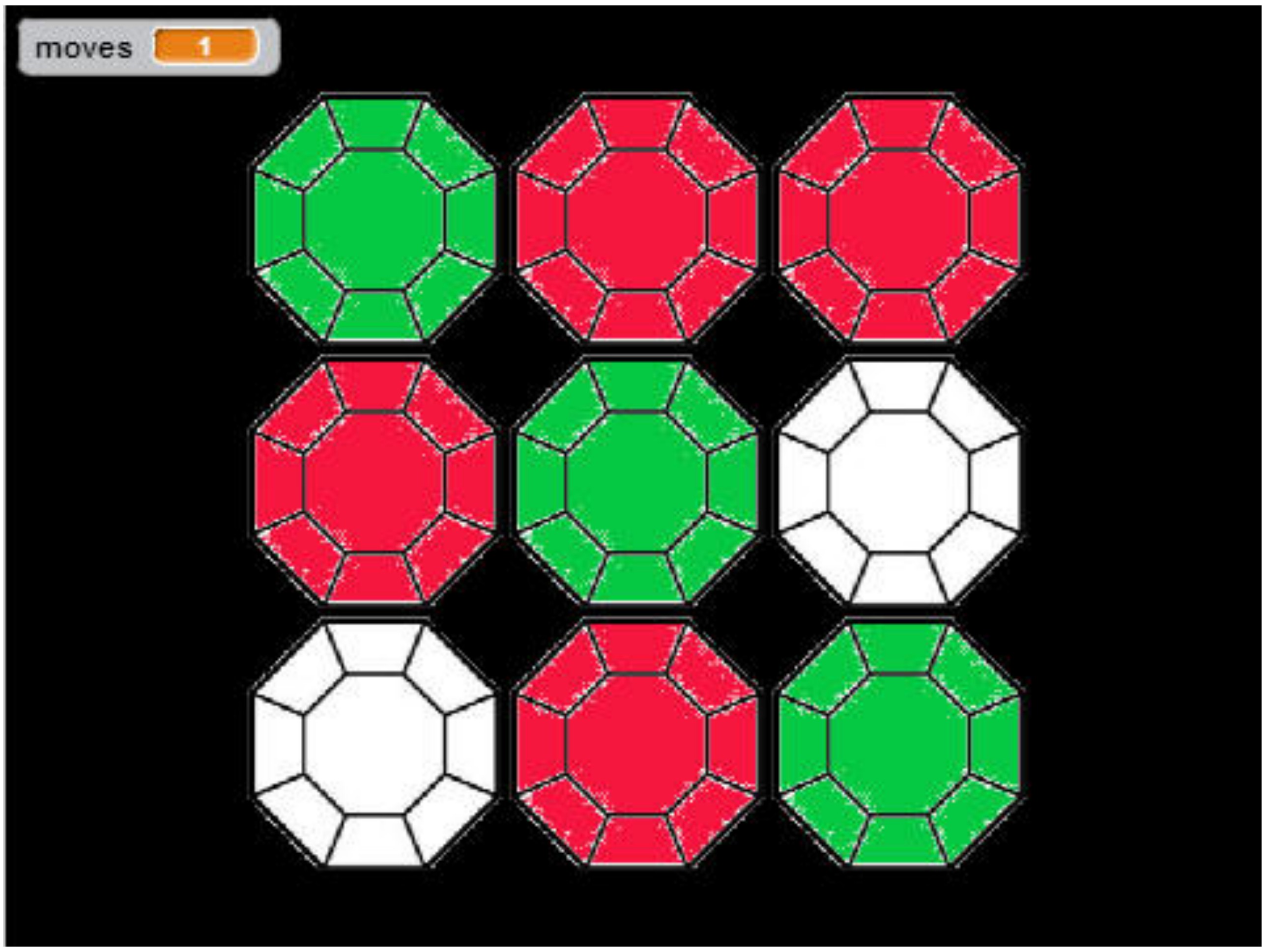
[Note: This problem was inspired by Dr. Shaw in the mathematics department.]

Adulthood has been much more laid back for Harry Potter. Now that he has defeated Voldemort and settled into raising his family he has accepted a fairly banal job - that of a jewelry smith. His specialty is creating customized pendants consisting of a 3x3 arrangement of jewels. Each of the 9 jewels is either a diamond, a ruby, or an emerald. These jewels can be changed by casting a spell at a particular jewel. This spell will change a diamond into a ruby, a ruby into an emerald, or an emerald into a diamond. Unfortunately, you may recall that Harry, while being "the chosen one" wasn't necessarily the greatest spell caster - in particular with simple spells. So unfortunately for Harry, any time he casts a spell at a particular jewel he not only modifies that jewel, but he also modifies the jewels touching that jewel orthogonally (that's a fancy word meaning both horizontally and vertically!)

Thus, if Harry started with the pendent described by the string "EDRDREDDE" or :



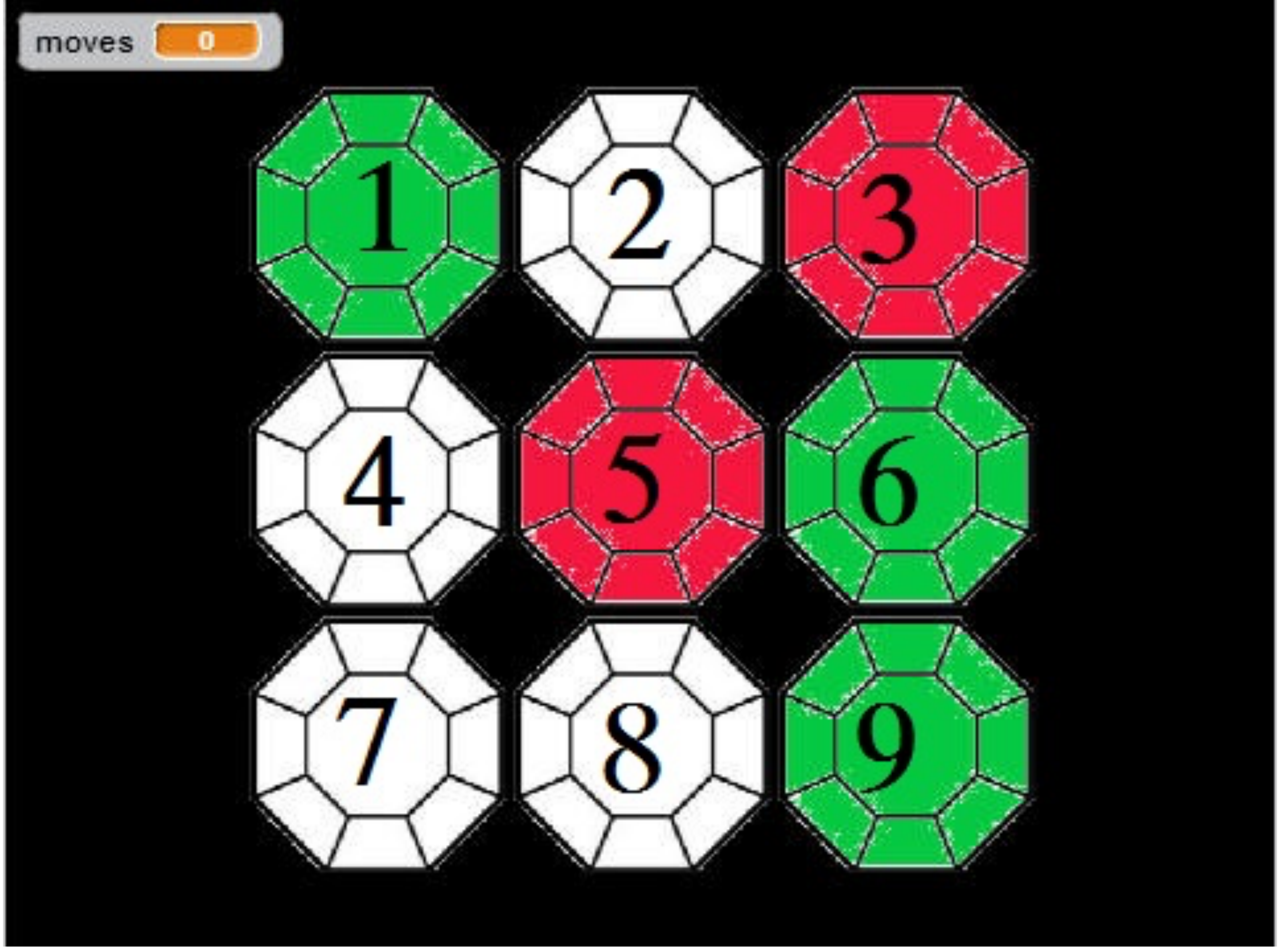
and cast a spell at the center ruby (jewel #5), he would end up with the pendent described by the string "ERRREDDRE" or :



The "goal" of this puzzle is, given a starting pendent, find the *shortest* sequence of spells that will turn the pendent into one containing 9 of the same jewel.

If you want to try this out you can play this game at: http://scratch.mit.edu/projects/26515800/

BEAUTIFUL! A straightforward search problem where at each node in the search you have 9 possible next moves - a single spell at each of the 9 jewels/possible spell locations:



A search problem for this game would attempt to find a goal state (9 matching jewels of any type) in as few moves as possible.

Requirements

For this homework you should write a program that fits the following requirements:

- You may write your code in either Python or Java.
- Your search code should be in a file called either HW1.py or HW1.java
- Your primary search should be in a function calleed search() which takes in a single parameter. This parameter should be the "starting state" of the puzzle as a SINGLE, string of all capital letters such as "EDRDREDDE" which would be the pendent described above.
- You must use one of the five uninformed search techniques we studied in session 5 or session 6 to find an optimal solution. The challenge is picking which of the 5 strategies we might use. We will talk about this in class, but I think there are two which make the most sense.
- When a solution is found you need to
    - Print out the sequence of moves that will get you from the initial state to the goal state. [Note, while the solution is actually valid in either order, I will expect you to print the path in the order it was actually discovered and I will pay attention to this.]
    - Print out the the *total* number of nodes expanded so far (gives us an idea of time complexity) and the number of nodes remaining in the fringe (gives us an idea of memory usage). Note that for the total number of nodes expanded so far this includes BOTH the nodes currently in memory

    and the nodes already looked at and discarded.
    - Return the sequence of moves that will get your from the initial state to the goal state. [Same thing you printed earlier, but include this as a return statement as well.
- For a few bonus points, you may implement a sedond search strategy (again, I think that only two of these REALLY make sense). If you do this it should be named searchV2() and follow all the prior requirements.
- If you use Java you may produce additional classes which support your primary class as outlined above. But I need to be able to launch it from the file/function as described.

Hints from past experience with this assignment...

Be careful when you "copy" your data structures. This is true in both python and java.

Remember what happens when you "copy" a referential data type in Java... you only get a copy of the reference, not a copy of the object. This means that if you have a node that you want to copy several times and modify to create the children nodes, you need to be careful what you do to copy the node. Students have a habit of saying something like:

```
//Assume that Node parent was created elsewhere
Node child1 = parent;
Node child2 = parent;
Node child3 = parent;
child1.modifyOneWay();
child2.modifyAnotherWay();
child3.modifyYETAnotherWay();
```

This just gives you FOUR variables all of which point to the same single instant. Students end up forgetting this and then being surprised when child1, child2, and child3 are identical and TOTALLY screwed up (in essence, they just cast 3 different spells on the same single object.

You will need to write code which produces a true, second *copy* of whatever data structure you use in your tree.

Similar "copy by value" vs. "copy by reference" problems exist in Python. Make sure you pay attention to this issue.

2) When it gets to be time to test your code, start with simple problems. For example,

```
DED
EEE
DED
```

can EASILY be turned into a pendent of all diamonds by casting the spell on the center emerald. Shortest solution is ONE move. If your code can't solve this, than you have a significant problem. Similarly, your code should detect that

```
EEE
EEE
EEE
```

is already a solution and requires no moves at all.

Submitting Your Work

Prior to the deadline, use the homework submission system

https://www.cs.uni.edu/~schafer/submit/which_course.cgi

to upload any files you have produced.

In addition to online copies, please submit hard copies of this lab at the start of class on the day this lab is due.