Hex

Instructions: Unless it has been done for you, print a copy of this document for your team. As you work through it, write answers to any questions or prompts that are in **boldface**.

Team:

Manager:

Recorder:

Presenter:

Analyst:

Analyst: Pay special attention to your team's performance in the area of *critical thinking*. How well does your team identify the problem at hand, spot similarities and differences, and make analogies?

Learning Objectives

Content Objectives

Parentheses below correspond to part of the knowledge units in the ACM's *Computer Science Curricula 2013*.

After completing this activity, students should be able to:

- Perform depth-first search in a graph (DS/Graphs and trees).
- Use Java's ArrayList class (SDF/Fundamental data structures).

Process Objectives

After completing this activity, students should have improved their ability to:

- Determine what a class does from seeing it used. [Information processing]
- Reason about how a program's behavior would change as a result of various code changes. [Critical thinking]
- Explain nontrivial algorithms. [Communication]

Model 1: Playing the Game

Open the Hex project in Eclipse. Run Eclipse.java to play the game. It’s up to you whether you want to read the rules or play the game first, but do both.

Hex was invented independently in 1942 by the Danish mathematician Piet Hein and in 1947 by the American mathematician John Nash (of A Beautiful Mind fame). Two players take turns placing pieces on a board made of hexagons, trying to connect two opposite sides of the board.

Rules to the game can be found here:

http://en.wikipedia.org/wiki/Hex_(board_game)

1. **What is the black player trying to accomplish?**

2. **What is the white player trying to accomplish?**

3. **What does the program do if you try to make an illegal move?**

4. **Is a tie possible? If so, give an example. If not, explain why not.**

5. **What strategy advice would you give?**

Model 2: HexNode

Examine HexNode.java.

7. **What is the type of the field `neighbors`?**

8. **In what package is the ArrayList class found?**

9. **What two methods of ArrayList are called in HexNode?**

Model 3: HexModel

Examine HexModel.java.

1. **If `size` is 3, how many HexNodes does the constructor create?**

2. **After the constructor finishes, which HexNodes are neighbors of `grid[0][0]`?**

3. **What two things does `playAt` accomplish?**

4. **The nontrivial method `toString` is never called in the running game, only in the tests. Why bother writing such a method?**

5. **Aside from tests, where is `search` called?**

6. **How could the method that calls `search` be implemented without the sentinels `north`, `south`, `east`, and `west`?**

7. **What are the base cases in the recursive method `search`?**

8. **How does the game fail if you remove the check `visited.contains(node)`? (It might not fail for several moves.)**

9. **Explain the purpose of the list `visited`.**

10. **Why does `findWinner` need to create a new ArrayList for each call to `search`? In other words, what would go wrong if `findWinner` just created one ArrayList and used it for both calls?**

11. **Explain, in plain English, the algorithm implemented by `search`.**

Reflection

Analyst: Reflect on your team’s performance in the area of *critical thinking*.

25. **What were your team’s strengths?**

26. **What were potential areas of improvement?**

27. **What insights did you gain?**