## HexNodeTraversal

### CRC

|  |  |
| --- | --- |
| HexNodeTraversal | |
| Responsibilities: Traverses through a structure of HexNodes connected by links to one another without modifying their data. | Collaborators: HexNodeTraversal obviously requires a HexNode ADT to exist for it to work. It also needs a root HexNode to begin a traversal and each HexNode should have at least three links (3,4, or 6 are the possibilities in a Hexagon Board). |

### Description / Overview

HexNodeTraversal is used to iterate through the graph-like structure of the HexNodes. As shown in the CRC card, each HexNode can have either 3,4, or 6 links too other HexNodes depending on their position on the Board. It keeps track of the root HexNode (where the traverse began), the current HexNode that is being traversed, and which linked node it is facing to (stored as an integer, 0 being the HexNode it is currently facing). It provides movement through the Hexagon board by moving from HexNode to HexNode, layer by layer while preventing the traverse from leaving the boundaries. It begins from the center and starts from the global position of 0 and works around that layer, then moves to the next layer until the entire HexBoard has been traversed. It is an inside-out approach at traversing the Board. First layer will just be the middle so it will have the node 0. The second layer will have 0-5, the third layer will have 0-11, the fourth 0-17, the fifth 0-23, and in the case the board size is 7 the sixth layer will have 0-35.

### Instance Variables

#### root

Data Type:HexNode

The root contains reference a HexNode that is the starting point of the traversal which is the center HexNode on the Board.

#### currentNode

Data Type:HexNode

CurrentNode is the CurrentNode that the iterator is on. All traversing onwards depends on this node and which HexNodes are adjacent to it.

#### facing

Data Type:Integer

Facing is an integer that contains the direction to start the traversal according to global direction.

### Method Overview

*public HexNodeTraversal(HexNode root, int facing)*

*public void go(int distanceOut, int distanceAround)*

*public HexNode getCurrentNode();*

*public void next();*

*private void out();*

*private void rotate();*

### Method Writeups

#### public HexNodeTraversal(HexNode root, int facing)

This is the constructor for the HexNodeTraversal class. It simply creates a HexNodeTraversal object, initially sets the root HexNode and the CurrentNode to the middle Node on the Board.

#### public void go(int distanceOut, int distanceAround)

Go is a method that chooses a HexNode based on the global positioning. DistanceOut is how many layers away from the center layer the traverse will next move to. DistanceAround is which global positioning number reference of the HexNode the traverse will go to. DistanceAround has constraints depending on which layer is chosen. For example, DistanceOut = 0, then DistanceAround can only equal 0 since there is only one element in that layer, which is the center HexNode.

#### public void next();

Moves the CurrentNode to the node beside it on the current layer. Example: If the currentNode is at 1 on the Global position, then it will move to the number 2 HexNode.

#### private void out();

Out simply moves the currentNode one layer outwards from the current position.

#### private void rotate();

Turns the facing direction one more to the right (Adds 1). It needs to check that it is facing a valid HexNode and that it does not rotate more than 6 times (0-5), it should reset back to zero in this instance.