**RockHound Specifics**

**Data Fields –** *hungry* (boolean) *target* (Rock) *allRocks* (ArrayList<Rock>)

public ArrayList<Actor> getActors()

1. Make him hungry
2. Find all the rocks in the grid. If there are no Rocks, RockHound should remove himself from the grid.
3. Find all the rocks in front, left, right and behind the RockHound . Put in an ArrayList<Actor> (actors)
4. return actors

public void processActors(ArrayList<Actor> actors)

1. If there is at least 1 Actor (Rock) in the received ArrayList,
	1. Change the color of the RockHound to match the first Actor (Rock) in the ArrayList.
	2. Check to see if that is the target. If it is, make sure target is changed to null.
	3. Remove the first Actor from the Grid.
	4. Make the RockHound not hungry.
	5. Check to see if you just removed the last Rock, in which case, the RockHound should remove itself from the grid.
	6. If we don’t have a target, pick a new target.
		1. Find all the remaining rocks
		2. Find the northern most rock (lowest row)
		3. Set the target to that northern most Rock. (Change its color to white to check at run time)

\*\*\*\*Only do the movement code if the RockHound is still hungry

public ArrayList<Location getMoveLocations()

1. Create an ArrayList<Location> that are all empty adjacent spaces around the RockHound.
2. return that ArrayList

public Location selectMoveLocation(ArrayList<Location> locs)

1. If the received ArrayList is empty, return the RockHound’s current Location. (He isn’t moving)
2. To start assume his current Location is his *bestMove*.
3. Calculate the distance between each Location in the ArrayList and the target’s Location. Find the Location that is closest to the target.
4. Set the direction of the RockHound to point toward that calculated “bestMove” Location.
5. return that *bestMove* Location

public void makeMove(Location loc)

1. Move the RockHound to the received Location.

**Helper Methods**

findRocks – Finds all the rocks in the Grid, populates the allRocks data field. If there are no Rocks, remove RockHound.

findTarget – Finds all the rocks, locates the northernmost rock, sets that Rock as the target.

findDistance – Receives 2 Locations. Returns the double value of their distance within the Grid.