**ActorWars**

**Instructions:**

Add GridWorld and ActorWars jar files as libraries in your project  
To test your Actors, instantiate an ActorWarWorld and add your Actors, like so:

public void main() {  
 ActorWarWorld place = new ActorWarWorld();  
 place.add(new MyActorThing());  
 place.show();  
}

The ActorWarWorld is a 42 by 42 grid, containing a number of rocks, trees, and wheat.  
The contents of the grid are distributed randomly across the space.  
Wheat is the most important resource, food.  
Wheat can be eaten, giving your Actors energy.  
Wheat can be planted from your inventory.  
Wheat is added to an ActiveActor’s inventory when it is destroyed by Action.attack()  
Wheat has an age, age 0-9 drops no wheat, 10-19 drops one wheat, and 20+ drops 2 wheat (profitable farming).  
Wheat ages increase by 1 per tick.

Your Actors must at least extend ActiveActor, however it is recommended to extend Peon (An extension of ActiveActor).  
Peons have an ArrayList as an Action queue, executing actions from index 0, then removing them.  
Actions are how your Actor interacts with the world, either by adding them to the action queue, or by using the method {ActiveActor}.perform(Action a)  
Actions can be exclusive, meaning that only one such action may be performed per tick; such actions include spawning, attacking, and moving.   
As many non-exclusive actions as you want can be performed.  
All ActiveActors start with an amount of energy, which can be referenced with getEnergy(); in addition, they extend DestructibleActor, and have 15 health.  
Actions are referenced from the Action class [eg. Action.attack(1), Action.move(), Action.turn(180) etc.]  
In addition, Peons have additional Actions designed to use their queues; such actions can only be performed by Peons [Peon.moveToGradual(new Location(0, 0)), Peon.conditionalAct(new ModifiableBoolen(true), Action.move()) etc.]  
Utils contains several functions, most of which are useful to Peon users, but can be used by anyone

**Rules:**

Actor.moveTo(), Actor.putSelfInGrid() ,Grid.put() are not permitted

grid.remove() and Actor.removeSelfFromGrid() are permitted if it’s your own actor

Any location on the grid is considered 'visible' and you can get data from any actors on the grid, although you may not invoke any methods on an actor that is not yours.

No manually creating new instances of your actors, use Action.spawn()

Actor.setDirection() is allowed, but Action.turn() is useful if you are extending Peon, as it allows passing of ModifiableIntegers

No Actions with negative costs.