# Code Structure

Other than the 4 custom subclasses for Critter, no other classes were created. The Critter collection remained in the Critter class as a static ArrayList variable. Any assumptions are written below

## Main Class

Holds the controller. Parses the input and runs it through a switch statement. Assumes the following:

- Any command not listed in list of commands will result in invalid input message
- Any valid command that results in errors via any means will result in error processing message
- Any class or subclass used as an input is case sensitive ('Craig' is not the same as 'craig')
- Run stats will only process valid subclasses of Critters and not Critter class itself

## Critter Class

Contains all the basic functionalities of all Critters as well as the view of the world. It also contains static variable holding the entire Critter population as an ArrayList collection. In addition to our implementation of the starter code, the following private methods were created as helper methods for the class:

> move() : used in both the walk and run methods to shift Critter position
> negModulo(): used to account for moving Critter to a negative position in grid
> critterEncounter(): wrapper function that deals with encounter resolution
> locOccupied(): used in worldTimeStep to determine if location is already occupied by another

## Custom Critters

All of the following Critters implement doTimeStep(), fight(), and toString(). Some have the private variable moved to indicate whether the Critter had already moved in the given time step. All Critters have the capability of reproducing but each has differing minimum thresholds for when they want to reproduce.

### Critter1 : Lazy Critter

> Critter that spends its time doing nothing unless it's so low in energy that it needs to actually move to get food. It will reproduce at twice the designated start energy.

### Critter2 : Fighter Critter

> Energetic and hotheaded critter that is constantly seeking a fight. This critter spends all its time running and looking for encounters. It never runs away from a fight and always faces it head on. It will reproduce at twice the designated start energy.

### Critter3 : Straight Laced Critter

> This Critter is the most varied out of all the custom Critters. It decides to either walk, run , or rest depending on how it feels at the moment. The catch is that it refuses to walk diagonally and will only moved either completely vertically or completely horizontally. I guess it's just OCD in that way. It will reproduce at twice the designated start energy.

### Critter4 : Mommy Critter

> This Critter is pretty straightforward. It will keep reproducing so long as it has enough energy to do so.