

Lab: GridWorld Critters

Exercise: Critters

Create the `Critters` specified by:

- subclassing `Critter` and overriding one or more of the methods `getActors`, `processActors`, `getMoveLocations`, `selectMoveLocation`, and `makeMove`.

For each of the classes you design, you'll have to decide which methods are most appropriate to override. **You may not modify the `Critter` class, or any class given to you in the case study.** *And if you find yourself copying and pasting code, you've missed the point of inheritance!* You will be graded on both the successful implementation of your new bugs and critters ***and on the design and proper documentation of your code.*** Projects that do not adhere to the style guide will receive a score of zero (0) for style and zero (0) for functionality.

If you begin coding before you have completed the design and documentation, you will lose 10% on this portion of the lab **and** you will be ineligible to earn the 15% additional credit for the optional exercise.

Implement the following `Critters`. If you like, try displaying appropriate images to represent your new actors.

Wolf

A `Wolf` might eat others in front of it, breeding more wolves when it has eaten enough, and dying when it hasn't eaten in a while.

Follower

A `Follower` might follow a particular unsuspecting actor around, perhaps picking a new actor to follow after a certain number of timesteps.

Stealth Critter

A `Stealth Critter` might behave like one of the other critters, except that it might randomly teleport.

Exercise: Design Your Own Game or Simulation

You may complete this section if you wish to earn a score above 85%. However, the additional 15% may not be used to substitute for the above required exercises. And, if the above exercises do not follow the style guide and/or do not function, no additional credit will be awarded. You will also forfeit this 15% plus an additional 10% if you begin coding prior to completing the design and documentation.

You may create a game or a simulation that uses a grid. Your project must be **non-trivial, it must be properly documented, and it must use only constructs in the AP Java subset**. You must show that you can use and extend the `Actor` class and that you can use mouse clicks and button presses to control some aspect of your game/simulation.

Some ideas (you may also, with permission, choose one that is not on this list):

- Flocking Birds
- Battleship
- Ant Farm
- Sudoku puzzle
- Checkers
- Frogger
- Jeopardy
- Mario
- Monopoly
- Maze escape