CS 251 Intermediate Programming
Space Invaders: Part 1 – Game Objects

Brooke Chenoweth

Spring 2014

Please note: This is an individual assignment. It is designed for you to start working on inheritance models, and to see how useful they can be. In this assignment we'll start working on a space invaders game that we'll complete in one or more later assignments.

Background

Space Invaders is one of the earliest shooting games and the aim is to defeat waves of aliens with a laser cannon to earn as many points as possible.

Problem Specification

You are going to define an inheritance heirarchy for four types of games objects:

- The `Ship` controlled by the player
- The `Laser` fired by the ship
- The `Aliens` attacking
- `Missiles` fired by aliens

In order to do this, you will define an `abstract` class called `GameObject`. The reason this class is abstract is because we don't want to be able to create instances of this class. However, we want it to be able to define certain functionalities for space invaders game objects. To make sure that this is done, your class must implement the supplied `Object2D` interface. The `Object2D` interface specifies seven methods. These methods must be implemented for each type of game object, either through inheritance from the abstract class, or in the concrete classes.

The ship and aliens are able to fire lasers and missiles, respectively. In order to do this, `Ship` and `Alien` must appropriately implement the provided `Shooter` interface. When a ship fires and creates a laser, the `Laser` should be centered horizontally above the ship. It's dimensions should be as specified in the `GameData` interface, which defines some common constants for the game. Similarly, a new `Missile` should be centered below the alien that fired it.

I have provided you with a `TestDriver` class to test your classes and some sample output to expect from that class.

Turning in your assignment

Once you are done with your assignment, use UNM Learn to turn in the java files that you have created. You should turn in a total of five files (one abstract `GameObject` class and four concrete implementations). Do *not* turn in the `GameData`, `Object2D`, or `Shooter` files, since you should not have changed them.