Name:

Bad Guy Attack!

CSC-490, Winter 2020

For this worksheet you will need to recall our previous discussions of the `BadGuy` class from the Strategy and Abstract Factory worksheets.

1. Suppose we have a class `Game` which has a method called `attack()`. When that method is called the following sequence occurs: a bad guys is created, the bad guy chases (the hero), and then the bad guy fights (the hero). The goal of this exercise is to understand how one could implement this method.

   (a) (2pts) As we discussed before, the bad guy that will be created will depend on the level of the game, so we should introduce some levels. Let's have classes `Level1`, `Level2`, `Level3`,... representing the different levels of the game. Suppose that in addition to getting from `Level1` to `Level2` by completing the first level, you can also jump multiple levels (maybe by warping, or maybe the game is not linear). Why is the State pattern appropriate here?

   (b) (5pts) Use the State pattern to draw a class diagram involving `Game`, `Level` and various concrete level classes. Leave room to add one more class and pseudocode.

   (c) (2pts) Add the pseudocode for the implementation of `attack()` in the class `Game`.

   (d) (2pts) In order to implement `attack()` in `Level` let's give `Level` a data member of type `BadGuyFactory` (the same class that we discussed in the Abstract Factory worksheet). Include this one extra class to your diagram (you don't need to draw all the concrete factories here...).

   (e) (4pts) Add detailed pseudocode for the implementation of `attack()` in the class `Level`.

2. (10pts) Draw a big class diagram for all of the classes associated to `BadGuy` that we've discussed this term. You don't need to include any data members or methods in your picture, just all the has-a, is-a, and creates-a arrows. To save space, you might want to draw collections of similar concrete classes so that one appears behind the other (as I do in class sometimes). You might also want to turn this page on its side. Finally, label the appropriate regions with Strategy, State, and Abstract Factory.