**Capstone Programming Project**

**Due Date:** Submit all documents and files as a zip file by 11:59 April 27th through canvas.

**Grading**: This project is worth 15 points of your final grade. It will be graded on the following criteria:

* 5 points – Meets required criteria
* 5 points – Style and quality of the code
* 5 points – Write up

**Project description**: Use the Goldfield library to create your own game. Any game is acceptable, but keep in mind that a simpler game will be easier to program. Even a relatively simple concept may end up being more complicated than you thought. Write a project description (see below). The following elements are required and will be weighted for half the project’s points. The project must have:

* Some event from player.
* An end goal.
* A reset function.
* Make at least one new mover.
* Share a library with a friend.

**Write Up:** Write a minimum of 1 full page, not much more than 2. Cover the following topics.

* Instructions to use your game.
* What library you shared.
* What libraries you borrowed. Please comment in the code to show which libraries came from who?
* Problems you encountered and how you overcame them.
* Things you like or don’t like about your project. What would you change or keep?

**Tips and suggestions**:

1. Start with a simple concept using the existing movers.
2. Figure out how to create the basic capabilities you need.
   1. Physics capabilities can be hard. If you figure out something like “bounce off a wall” you can share that library with your friends.
   2. Movement tracking can also be hard. Goldfield prevents some impossible moves, but this is another library to share with your friends.
3. Create the run through from start to end of the game.
4. Figure out how to reset to the start once you finish.
5. Develop whatever movers you need to match the game’s operation.
6. If you create a mover for the “Bad guy” that makes a great library to share with your friend. You might also share movers which are obstacles (wall?) or objectives (house?).
7. ONLY at this point should you consider something more complicated.
   1. Add small things like animation/GIFs for the images.
   2. Think about what makes your project work or look better.
   3. Focus on making your game right. Test, test, test.
   4. It is often better to test by letting someone else use the game. You know what you expect it to do. They will try to do something else entirely.
8. Look at your code and make sure it is properly formatted and has good comments, style, etc.