

## CISC106 Spring 2013 Final Project

Your job is to write a Python solution to the packing problem. The basic program is already designed, and the functions enumerated, so that you will be given certain functions to write. You will also be writing unit tests<sup>1</sup>.

### Readings

- Review your class notes about lists, classes, pretty much everything we've done so far, and (if we go over it in class) the project.
- [http://en.wikipedia.org/wiki/Packing\\_problem](http://en.wikipedia.org/wiki/Packing_problem)

### April 15

Team rosters due. Team must submit declaration of team name and member names and UdelNetIDs via a single email (see email instructions below) to professor by 11:59 p.m.

Example:

Subject: 106 Final Project, The Wombats

The Wombats:

Xenon Raqzx, xraqz

Zorpon Minzq, zminzq

Pweebl Xanz, maryx

### May 9

Individual submission date. Submit any code you (an individual) have written as of the deadline, just to prove you are doing something.

### May 14

Complete Project 2 and testing due midnight.

Recommended timeline:<sup>2</sup>

1. By **April 21**, have your shape constructor and `get_shape` tested and implemented.
2. By **April 24**, have a *text based* version of `GameView` implemented using strings and print statements. Test it informally by creating one, handing it a hand-coded list of shapes, and then calling `draw_game_board` on it to make sure the correct board is printed out.
3. By **April 26**, Have `show_and_rotate` tested and implemented.
4. By **April 29**, Have your board constructor tested and implemented.

---

<sup>1</sup>And if you have learned anything in this course, you'll complete them thoughtfully and use them to test your code as you develop it!

<sup>2</sup>Note that you can work on whatever timeline you like so long as the entire project is handed in by the due date. However, this timeline was devised by someone who has lots of experience pacing software projects! If instead you decide to wait until the last minute to start your project, you will get no sympathy from the professor or TAs.

5. By **May 1**, have your test for `fit` written with a good set of assertions testing any cases you can think of. Implement `fit` on **May 2**. **NB:** You'll need to have `fit` implemented before you can write the other board functions, since they make use of it. This is why it's recommended you do `fit` first.
6. By **May 3**, have your test for `find_one_place` written. As with `fit`, you'll want to test any cases you can imagine.
7. By **May 5**, have `find_one_place` fully implemented.
8. By **May 8** you should have `find_best_place` tested and fully implemented.
9. By **May 10** you should have `update` tested and fully implemented
10. You should be able to implement `start` and `run` at this point and watch your program pack the shapes on the board.
11. By **May 14**, you should have implemented your graphical `GameView` and replaced the text based one with it. I will provide the changes necessary to `start` to make the graphical version work.

## How will we be graded?

You will be part of a coding team. You will be individually responsible for certain specific parts of the project, and **one half** of your grade will be based on your achievement of those parts. The **other half** will be a grade for the achievements of the team. The team grade will be apportioned among team members by the team, using a blind rating system where each team member rates the contributions of their peers.

## Email

Any email about the project to the professor or your TA must include [106 PROJECT *yourteamname*] in the subject line, including the square brackets. Emails without this run the risk of being lost or overlooked <sup>3</sup>.

## What must be done before March 21?

Nothing, that was weeks ago! See deadlines for this project above.

## Teams

Teams will be 2-4 people only.

NOTE: *You* form teams, but only the professor can break a team apart. Individuals are *not* allowed to leave a team without the professor's permission. Teams are *not* empowered to "vote people off". Form carefully - breaking up is hard, and nobody wins.

## What will the project be graded on?

The first eighty-five percent of the project will be simple completion of the existing functions (and test code) so that your game performs correctly and fills the square with moderate efficiency (I will release the data file I used for the class demonstration so that you can measure your code against it).

The last fifteen percent will be based on your adherence to the deadlines listed in this writeup.

I'll also be uploading a document with some extra credit opportunities within the next few days.

---

<sup>3</sup>Adding this information to your subject line allows us to categorize emails automatically. Help us help you!

## Communication

**Work on the project within your team only.** Seek help and explanations from your team members, not from members of other teams. In particular, DO NOT communicate with another team or outside person about how they wrote a particular function or test. Feel free to send your code or thoughts to other members of your team, but not outside the team. Violations of this policy will be considered academic dishonesty (see the class web page).

**All team-related emails<sup>4</sup> to another member of the team must be cc'ed to every team member.** This means that everyone will need to be especially careful when choosing their words to comment on someone else's code, timeliness, new shoes, etc.

## Team Member Evaluations

I will try to upload the questionnaire you will all be asked to fill out on your teammates in the coming week or so.

---

<sup>4</sup>This rule applies to any written words, including text messaging, paper, etc.