Week 04 Workshop

EXERCISE:

Workshop 4 - And Now, For Something Completely Different

PRE-WORKSHOP
PREPARATION:

Apart from watching the lectures, there is no specific preparation to do this week! This leaves you plenty of time to read over the assignment.

IN-WORKSHOP REVISION:

Your tutor will briefly discuss the assignment, and any good advice they have about it. If you have any questions about content from this week, this is a great time to ask them!

This week's workshop will work slightly differently to previous weeks. This is because the content for this week is more gentle, and we want to give people time for thinking about the assignment.

We expect that we will split the class in half, with one tutor for each half of the class leading a discussion.

First, we will do a short code review. This will take the form of choosing an exercise (probably `christmas_tree`), and having everyone pull up their code. Much like in a pull-request, we will get students to swap computers and leave comments on things they like, don't like, or have questions about. Then, in a larger group, we will look through some of the interesting code and have a group discussion.

Second, we will have a discussion on the theoretical answers given to questions in Week 2 and Week 3's exercises. This will be the main opportunity for tutors to give feedback on that theoretical work. If you cannot make this weeks workshop, you will still get marks for submitting the theoretical work, but you will not get feedback (unless you ask us seperately). If you can turn up, you will get detailed feedback on those answers.

After the first two parts of the workshop, we will split into small groups where you can work on `directions_lib` (which is provided as partially working starter code). If you can get it working, it will be useful for your assignment. **Unlike other code, you may use this code in your assignment even if somebody else in your group contributed to it**

Directions lib implements three types: `Direction`, `Coordinate` and `CardinalDirection`. Try and create a library which:

- Can add (including +=) Directions and Coordinates.
- Can create Directions from CardinalDirections.
- Can find if a Coordinate is within a rectange defined by two other Coordinates.
- Has documentation for every public type.
- Has at least one doctest for every public type.
- Has tests to check internal functionality. We suggest one test per method/function.
- Bonus: Does anything other features you think might be useful for the assignment.
- Bonus: Finds the distance between two points.
- Bonus: Implements scalar multiplication on Directions.

Note: Some of this code may be useful in next week's workshop, so keep a copy of it for next week.