UCL DEPARTMENT OF MEDICAL PHYSICS AND BIOMEDICAL ENGINEERING

Research Software Engineering With Python

Refactoring Trees - helping Nicolas Flamel!

Module Code:	MPHY0021
Module Title:	Research Software Engineering With Python
Lecturer(s):	Dr Matt Clarkson
	Dr Anastasis Georgoulas
	Dr David Pérez-Suárez
Coursework Title:	Refactoring Trees: An exercise in Research Software Engineering
Date Handed out:	December 17th, 2018
Coursework Deadline:	January 21st, 2019
Submission Id:	454597

Total mark:
16.0/*25*

Description: This assignment asked to version control and refactor the existing code, use argparse to manage the input parameters and analyse the performance with and without vectorisation. Flamel's horrible code is given as a starting point.
Your job consisted in converting the code into **a cleaner and more readable script**, with a **a command line interface**, documenting each non-breaking change with **git commits**. Additionally, you needed to create a faster implementation using **numpy**, and **compare the performance** of both implementations.
This exercise has been marked manually with a tool that showed the changes and files needed for marking.

Marking legend:

Title of the grading section *points received*/*total*

General notes about this particular section

section being marked *auto* *manual* **total**

Feedback about this section
Comments here don't necessarily subtract points
Automatically graded marks either worked or not. Noted if they had to be run manually.

Refactoring of Flamel's Code (`tree.py`) *11.00* / *15*

Raw code initialised in git repository *Mark Awarded.*	*0*	*1*	**1.00**
1 mark for each git commit *Mark Awarded.*	*0*	*1*	**1.00**
1 mark for each git commit *Mark Awarded.*	*0*	*1*	**1.00**
1 mark for each git commit *Mark Awarded.*	*0*	*1*	**1.00**
1 mark for each git commit *Mark Awarded.*	*0*	*1*	**1.00**
1 mark for each git commit *Mark Awarded.*	*0*	*1*	**1.00**
1 mark for each refactoring *Mark Awarded.*	*0*	*1*	**1.00**
1 mark for each refactoring *Mark Awarded.*	*0*	*1*	**1.00**
1 mark for each refactoring *Could have refactored the for loop to iterate over the object rather than using range and len*	*0*	*0*	**0.00**
1 mark for each refactoring *There are still some lines of code that are repeated and could be moved into functions or otherwise streamlined*	*0*	*0*	**0.00**
1 mark for each refactoring *Lost the default values during the refactoring*	*0*	*0*	**0.00**
Use argparse to handle the branch angle parameter *Mark Awarded.*	*0*	*1*	**1.00**
Use argparse to handle the branch length parameter *Mark Awarded.*	*0*	*1*	**1.00**
Use argparse to handle the branch depth parameter *Mark Awarded.*	*0*	*1*	**1.00**
Use argparse to handle the branch scale parameter *It's not decreasing if you put a value larger than 1; None of the parameters include defaults values;* `argparse` *provides a way to set the type*	*0*	*0*	**0.00**

Performance programming

5.00 / *10*

Time to run code identified *Mark Awarded.*	*0*	*1*	**1.00**
Figure created *Mark Awarded.*	*0*	*1*	**1.00**
Figure correctly formatted *Mark Awarded.; Missing title*	*0*	*1*	**1.00**
Figure auto-generated from script *Mark Awarded.*	*0*	*1*	**1.00**
Performance law identified *No law identified*	*0*	*0*	**0.00**
Subtract the change angle from all angles in a single minus sign *Mark Awarded.*	*0*	*1*	**1.00**
Taking the sine of all angles using ‘np.sin‘ *This was not to replace ‘math.sin `to` np.sin‘*	*0*	*0*	**0.00**
Moving on all the positions with a single vector displacement addition. *Code is not fully vectorised*	*0*	*0*	**0.00**
Uses hstack or similar to create new arrays with twice the length *Use of multiple h/v/dstack when one would suffice*	*0*	*0*	**0.00**
Comparison is not meaningful *Comparison is not meaningful*	*0*	*0*	**0.00**