This student is registered as dyslexic / dyspraxic with UCL Student Disability Services.

Please refer to Guidance for Academic Staff: Assessment of the work of students with specific learning difficulties – Dyslexia / Dyspraxia

http://www.ucl.ac.uk/disability/info-for-staff/dyslexia-marking-guidelines

Bad Boids
Assessment 2
MPHYG001

Leo Carlos-Sandberg
16101556

6/1/2017

Code Smells

Create folder structure (V1.0.2-V1.0.3):

Smell: No folder structure

Details: Created folders to put the tests, fixtures and code in

Replace magic numbers with constants (V1.0.4):

Smell: Raw numbers appear multiple times in the code

Details: Create named variables to replace the numbers in the code

Replaced repeated coded with a function (V1.1.0):

Smell: Repeated code used to initialize positions and velocities

Details: Create function new_flock to replace it

Replaced multiple variable with one variable (V1.1.0):

Smell: Two identically shaped arrays used to hold x and y values of positions and velocities

Details: Create numpy array to hold both x and y arrays

Replaced for loops with numpy array calculations (V1.1.1-V1.1.5):

Smell: Large number of nested loops iterating through arrays

Details: Used numpy arrays and numpy calculations

Introduced functions and classes (V1.2.0-V1.6.0):

Smell: Large amount of code with minimal structure and little separation

Details: Created classes to hold different sections of the code and functions to break these classes up further

Introduced different files (V2.0.0):

Smell: Number of large classes in one file

Details: Created a file to hold each class that can then be imported into other files

Add unit testing (V3.0.0-V5.0.0):

Smell: No testing for now broken up code

Details: Created unit tests for each definition which were broken up into different files depending on the class

Add command line input (V6.0.0-V6.1.0):

Smell: All values are hard coded

Details: Added a parser to allow command line entry of values

Add input file (V6.2.0):

Smell: Large number of values to input by hand

Details: Added the option to parse a file containing the input values

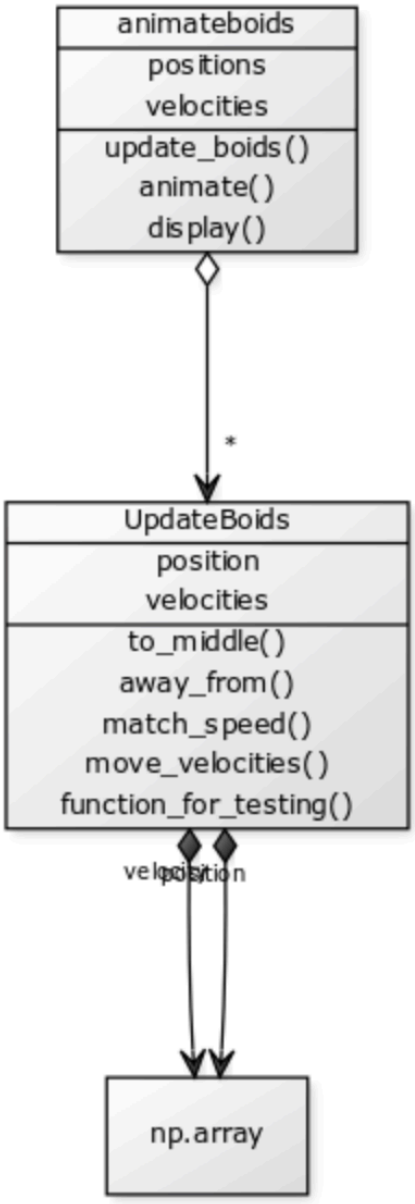
Add supporting files (V6.5.0):

Smell: No documentation on how to use the code or what it is

Details: Added supporting files to explain the code and its use

UML

UML diagrams are a way to display the class structure of code in a visual format.



Refactoring

The advantages of refactoring, which is the act of making many small changes to code to produces a much larger cumulative effect, are usually a reduction in the complexity of the code and an increase in readability by dealing with code smells. This processes of slowly changing the code in small chunks while keeping the code functional, is a lot safer then making large changes, as if errors do occur it is easier to identify the source and revert to the original code if necessary. This processes also adds in testing as a natural step as each part of the code is changed and as functions and structure emerges tests become easier to identify. Testing allows this processes to maintain the codes functionality and have faith in the changes being made to the code, allowing old often-unwieldy code to be improved with out the fear of breaking it. Often the aim of refactoring will be to help improve the maintainability, reusability and efficiency of the code. Over all refactoring is a safe method of improving code, especially if the code is in use during this process.

Problems

During the initial changes to update_boids() the for loops were exchanged for numpy array operations, these have a different operation order and can work in parallel giving back a slightly different answer then the for loops. However this does not strongly affect the actual program.

I encountered some issues preparing my package, including my package fail to be able to properly make calls between folders, this stopped tests being able to call classes and the script from calling the starting class.