MPHYG001 Assignment 2: Refactoring Boids

Danial Dervovic
University College London
danial.dervovic.11@ucl.ac.uk

February 27, 2016

Introduction/Usage

For this assignment, the task was to take the Boids code[1], packaging it up into a form that can be `pip` installed and run from the command line. This code may be found at the GitHub repository `ddervs/bad-boids`. The usage of the packaged function is shown below (the output of `boids --help`).

```
usage: boids [-h] [--config CONFIG] [--example_config]

Runs the boids simulation.

optional arguments:
  -h, --help          show this help message and exit
  --config CONFIG     YAML file with simulation options. See README.md or use
                      --example-config option for specification.
  --example_config    Saves default config file to current working directory.
```

Refactorings

In this task, the refactoring approach was used. The common refactorings utilised in the code are shown in the table below.

Code Smell	**Commit number**
Repeated Loops	515da7c
Magic Numbers	87c90c7
Turn `boids` into a class	2ff425b
Repeated code/copypasta	0128753
Many constants → config file	a23e2fa
Large function into units	fa117b3
Too many loops → `numpy`	980ce16

The main advantage of such an approach is that at each stage, the code set still works as originally. This contrasts with the approach of rewriting code, which is both more bug-prone, and requires significant time until the new working code set makes itself available. Refactoring stops code from *rotting* without breaking it, both in the sense of ensuring it is resistant to new developments in the environment of its dependencies, and keeping it readable for future maintenance. The UML diagram for the `boids` package is shown in Figure 1.

Problems

The main problems encountered in the project were in integrating the configuration file with the command line interface, and in plotting the animation.

It was found that in `setup.py`, it was needed to specify the fact that there was a non-python config file in the package. This was because under `pip` installation, only python files are installed by default.

[1] http://development.rc.ucl.ac.uk/training/engineering/ch05construction/10boids.html

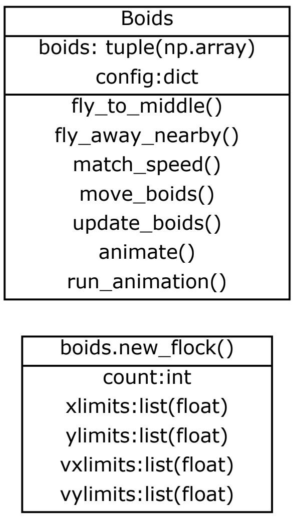

Figure 1: UML diagram of `boids` package.

Thus, when the config file was imported in the code, it would only work when executed in the package's root directory.

Another big problem was plotting the animation. The main issue is discussed at `http://stackoverflow.com/questions/21099121/python-matplotlib-unable-to-call-funcanimation-from-inside-a-function` and `matplotlib issue #1656`. For certain GUI backends, if the animation isn't stored in a persistent variable, it will quietly fail, as python's garbage collection will clear the plot. Thus the command line entry point scripts had to be restructured (commit `eecacbf`) to avoid this happening.