

Software Engineering Large Practical

s1235260 Francisco Vargas

December 15, 2014

1 Development Process

1.1 Original Specification

The Original Specification comprised of a web app in which users could rank paths. Paths being streets edges walks accross parks etc. Besides the orginal path ranking non users/users should be able to obtain a random walk biased on the ranking of previous users. The idea behind this is to through a dice which is bias on users weightings at each decision step. A decision step in this sense is simply moving from one edge to another aiming to reach a final destination. So input a start point input an end point and you have a random path which probably may by plesant. This specification was met fully.

1.2 Challenges and Solutions

the biggest challenge was working with spherical polar coordinates (The earth can be approximated to a sphere) and ensuring the path stayed sort of on track. The movements here are probabilistic (pseudo random) and hence constraining such movements becomes very difficult. Understanding the cosine law for spherical polars and implementing it to give constraints on the queries was a major challenge in this project.

Another significant challenge was finding a dynamic map. This took about 2 weeks of research before the start of the project. Used Mapbox and not leaflet like I originally propposed nonetheless both their API's are connected. In terms of the data which provides addresses (geocoding and reverse geocoding) I used openstreetmaps originally started using google maps yet encountered loads of traffic problems.

Working in the web development framework was remarkably challenging since it is something I am new to. Due to this lack of experience I made several design and choice errors which I will expand on later chapters of this report.

1.3 Source Control (git)

As required git was used for source control. More specifically I used I private repository in github since I enjoy the graphs they provide. I find visual aid useful when dealing with branches.

The general approach was to add elements with their relevant commits as time progressed commits became shorter and less clauses where used within them. When confronting a majoy bug fix I would branc away from master and merge after the bug had been solvented.

1.4 Unit Testing

I carried out unit tests formally towards the end of the project to verify if the custom mathematical functions I used worked as expected. Something important to note is that I should have carried out unit tests earlier on in the development process and along with branches and bugfixes, I did not manage to do so since I am new to this development process and workflow which let me to struggle in managing everything perfectly.

I would like to point that I made an attempt in replacing view tests by clicking on buttons and querying the database using sqlite3.

There is an interesting test inside the unit test folder which I think is worth mentioning. The test *visual_dice_test.py* computes histograms of two different weight distributions. One distributed in a symmetric normal (but discrete) like manner and the other uniformly distributed. The dice is rolled from 100 to 9000 times and its histograms at different number of rolls are computed and plotted as png files. By inspecting these images one can confirm that the biased dice does indeed work. A formal unit test may not be relevant here since its probabilistic behaviour does not give us information that can be asserted.

1.5 Documentation and Comments

During the development process small informative comments and debug logs were made. Towards the end more formal and dense documentation was delivered.

2 App Dissection