Contributing to an open source project

Ειδικά Θέματα Τεχνολογίας Λογισμικού 2015

Barcelonetta

Final Presentation

Description of contribution

Where we were left last time:

- Bug was found, reported and assigned
- I was getting accustomed to the various technologies
- I joined the aspiring coders guild.
- My next steps were:
  - Set up Habitrpg locally
  - Implement the deletion of empty parties

Details of steps taken (1)

- I set up MongoDB and got accustomed to it:

  - Had many difficulties installing on Windows, had to create my own YAML (a human friendly data serialization standard for all programming languages) config file , and other directories

  - In the end I still did not have access to the http interface, and did not create a Windows service, but did not have any problems with creating my own database

  - Set of collections → Set of documents → Set of key-value pairs. Documents have dynamic schema, which means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

Details of steps taken (2)

- Got accustomed to node.js

  - Definition: Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

  - Installed all the required modules and also used

  http://nodeschool.io/ for interactive lessons.

Details of steps taken (3)

- Forked and cloned the project, and then

  - Also had to install some npm packages globally,

  - And some npm packages specific to HabitRPG.

  - npm is a NodeJS package manager. As its name would imply, you can use it to install node programs.

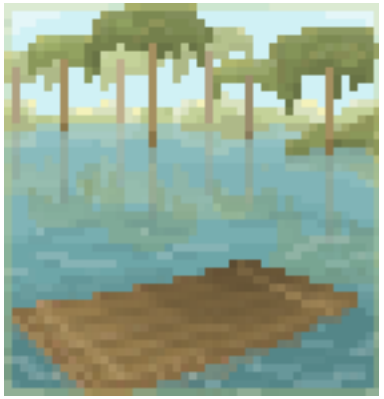
- Bower was also a prerequisite. It is a front-end package management, that exposes the package dependency model via an API. It depends on Node.js and npm.

Details of steps taken (4)

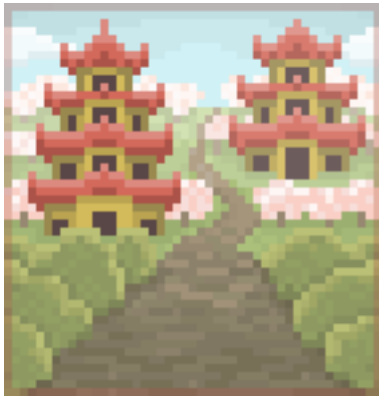
- And then with MongoDB running I was also able to start HabitRPG on my terminal and examine my changes through localhost!
- Because the master branch is continuously updated, I had to add the remote upstream and fetch and rebase my local develop branch often.
- The next and most important step was to study the code and find a way to implement the automatic deletion of empty parties.

Details of steps taken (5)

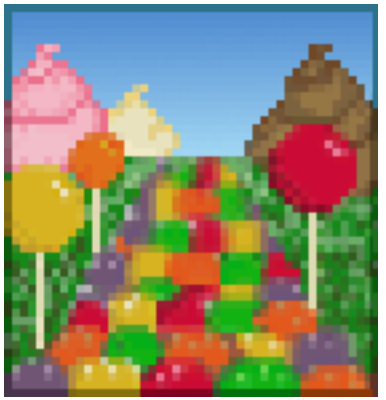
- After thoroughly checking the code, from an holistic point of view, I found that the file I should change was a controller (represents the logic behind the website functions) of the source code named groups.js
- I created two test users through localhost, and managed their parties and guilds through MongoDB, in order to find how to implement their automatic deletion.



Final Steps!



- Implemented and tested the code, it worked perfectly. After discussing with the developers, they suggested not only parties but also private guilds to be deleted. This was also implemented.
- Added and committed my changes, cleaned some untracked files, pushed my commit and sent a pull request!
- My commit passed the automatic testing applied by Travis CI.



Is it over?

- The feedback from the developers is positive.
- Since my commit is changing API behaviour, changes to the api tests for groups need to be applied. They offered to inform me regarding the next steps that need to be taken, and I will probably also implement that test.

Communicating with the developing team and contributing to this project was an overall positive experience.

Thank you for your time!~ Any questions?

HABITRPG