**Log Book**

**An application of Machine Learning to the Card Game Hearthstone**

**am2074 – jrfc2 – fz41**

# Jan 12

Searched the best way of getting data from the game. Concurred that reading log files was better than listening on the network.

Made a Proof Of Concept using python to tinker around Hearthstone log files, read them and retrieve useful data then display some of those data on a GUI.

# Jan 20

Put together a Roadmap.

Wrote a description for the project, found a supervisor, confirmed subject of the project.

# Mar 21

Experimented different frameworks to write the GUI. Tested NW.js, Electron and HTML Application (HTA).

Decided to use Electron.

# Mar 25

Gathered static data from Hearthstone.

Fetched cards data with hearthstonejson, images of the cards with wow.zamimg.com api and generated cropped pictures of the cards with Python, using a library called Pillow.

# Mar 26

Built a rough interface with a GUI and CSS files.

Worked on the design of the GUI.

# Apr 20

Searched a library to handle and process data to display on the GUI. Tried ReactJS and AngularJS. Opted for AngularJS.

# May 4

Found a very good Python library to parse Hearthstone logs: python-hearthstone.

Since NodeJS is used to run our application, had to find a way to make Python communicate with NodeJS.

# May 15

Discussed about what kind of algorithm to use in order to recognize enemy deck (opted for Instance-based learning for now). Made hypotheses about how to implement it.

# May 20

Started finding a way to retrieve most popular decks of the moment from hearthstone dedicated websites.

# May 25-27

Worked on classifying enemy deck: associate it with an existing deck by looking at the cards that have been played so far.

# May 30

Deployment of a MongoDB server to store all data.

# Jun 2-3

Started working on play prediction.

# Jun 4-5

Gathered most popular decks for each class and wrote advices for each of them.

# Jun 6-8

Added a way for the user to manually enter the cards he uses in order to see the cards remaining in his deck during a game.

# Jun 9-10

Started playing games while using the program in order to gather data and get more accurate results.

# Jun 13-17

Fixed bugs, improved classification/prediction algorithms, improved communication Python/NodeJS, improved GUI (design, “always on top” option, images …).

# July

Expanded database by playing numerous games, registering new decks, refining and correcting predictions results.

Created a video presenting and demonstrating the project.