

CO3015 Computer Science Project

DOTA2 Web and Android Assistant

Interim Report

Matthew Bennett - mb508@student.le.ac.uk
December 2014

Table of Contents

Aims & Objectives

Planning & Timescales

Prototype Description

- *Captains Mode Draft*
- *Android Application*

Software Architecture, Algorithms and Data Structures

- *Architecture*
- *Data Structures & Algorithms*

Bibliography

DECLARATION

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s).

Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s).

I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Matthew Bennett

Signed:

A handwritten signature in black ink, appearing to read 'Matthew Bennett', with a horizontal line underneath it.

Date: 11/12/2014

Interim Report

Aims & Objectives

I feel like the aims and objectives set out in my project plan are still fairly accurate, there are a few optional aims that I now feel don't really add anything to the functionality or complexity of the project, such as:

- A twitter/reddit news page that shows updates surrounding the game and personalities
- A hero records page that shows records recorded for each hero, GPM, XPM, KDA, LH, Denies, Hero Damage, Hero Healing, Tower Damage, Rapiers Purchased ect.

I feel like neither of them provide anything other than filler to the project and I will probably not be including them. There are also a few optional aims that I would like to move up to Recommended as I feel like they add a large amount of useful functionality and complexity, they are as follows:

- Advanced statistical pages on your games, hero trends, the standard deviation of hero win rates ect.
- Additional AI in the drafting assistant that looks at your recent games and automatically takes into account the heroes you play when deciding suggestions

I feel like these two add hugely to my project, the second one will be added on top of the personalised log in system so that it is automated in the way it effects the users advised heroes. The objectives and challenges of the system remain the same as the ones I wrote in my project plan.

Planning and Timescales

My original plan of how I would execute my project has changed vastly, as I am now building a web app and then attempting to transfer the functionality of the drafting assistant over to an android application. For my prototype I have a web application that has a functional drafting assistant. My plans over christmas and the next semester are as follows:

Task	Completion Date
Fully functional web based drafting assistant with good usability	12/01/2015
Simple android GUI ready to take functionality of the drafting assistant	12/01/2015
Log in system that effects your advised heroes by using your personal heroes played	28/02/2015
Fully functional statistics page	28/02/2015
Fully functional android based drafting assistant	28/02/2015

I feel like this plan is both attainable and challenging, it will also allow for time afterwards to introduce new functionality and complexity to the project if it is required. Having my web functionality done before exam period along with a base GUI for my android app will allow me to dive into the additional functionality once my exams are finished.

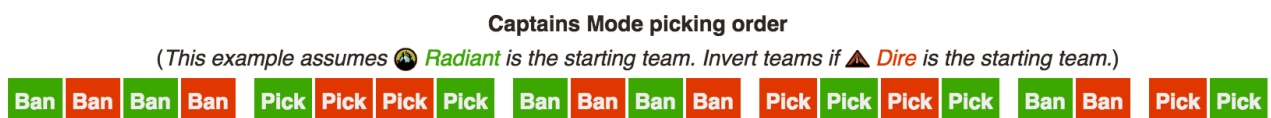
Prototype Description

My prototype is relatively complex and has a number of components. It utilises a DOM Parser that is written in Java to gather data from valves API, this parser inserts the data into a MySQL database that tracks how many times each hero has beaten each other hero. I then have a java document that contains methods that gather data from the database and processes them into data used by the web app to display the correct heroes to counter pick against specific heroes.



Captains Mode Draft

The main feature of my system is to have a fully functional assistant for a user to utilise to guide them through a complete draft in a captains mode game of DOTA 2. Currently I only have implemented the option to select between 1 and 5 enemy heroes and the system will provide the top 10 heroes in terms of mean win rate against the heroes inputted. Over the christmas break I will implement a full draft, including both teams picks and bans. Each team gets 5 bans and 5 picks, they then play with the heroes they picked. The draft order is as follows [1]:



My final system needs to guide a user through the draft, so it needs to take ban/pick inputs when appropriate, the user needs to input whether they are picking first or second so that the system knows which 'side' of the draft belongs to the user. The system will calculate advised picks by taking all the data inputted into account. For advising the user which heroes to pick it will calculate which heroes 'counter' heroes the enemy has picked, which heroes synergise well with the heroes that your team has picked and ignore all heroes that have been banned; resulting in a set of heroes that statistically work in the current situation. For advising heroes to ban it will effectively do all of the opposite calculations, working out synergy with the enemy heroes and 'counter' against the friendly teams

heroes. The system will update as new information is entered so at whichever stage of a draft the user is, it produces advice for the next few decisions that need to be made.

Currently I am just using mean averages to combine hero data, for example if a user inputs that the enemy team has selected Zeus and Dark Seer the system will find all other heroes win percentages for these two heroes and return the 10 heroes with the highest mean win rate. I plan to improve both the accuracy of this statistic and the complexity, it will eventually take into account more factors such as recent changes to the game and certain heroes only 'countering' others at a high skill level. In the finished product the system will be easy to use and produce accurate and useful information.

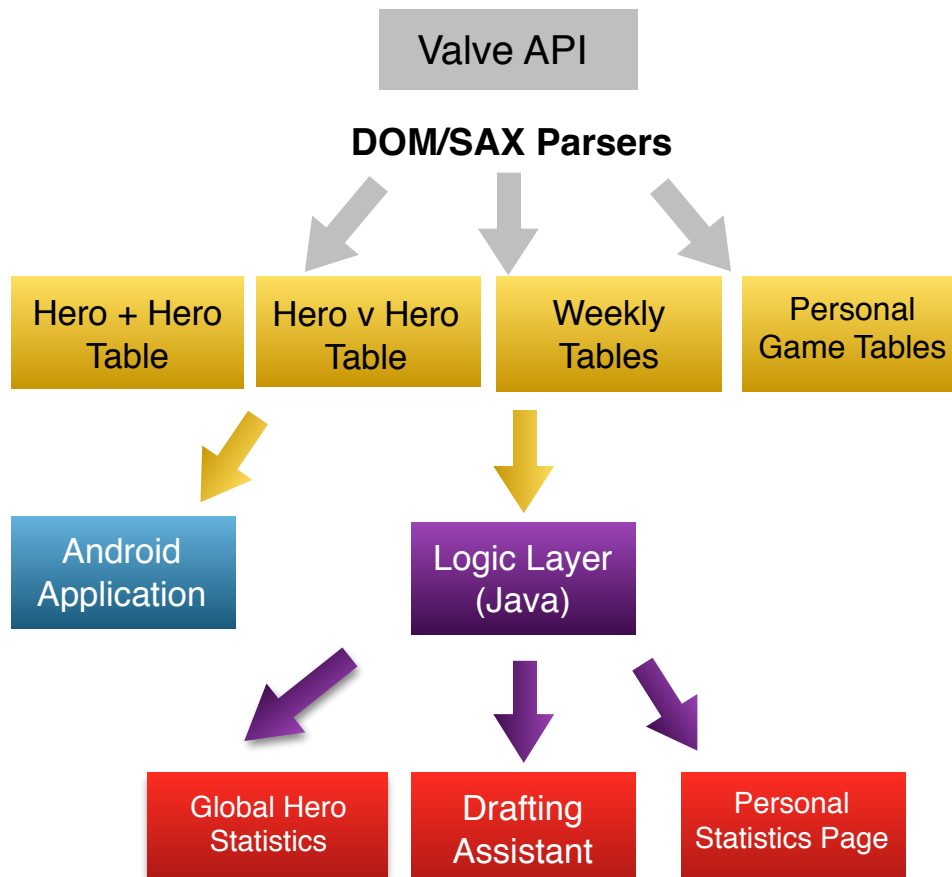
Android Application

Currently I have only messed around in android development and not got any functionality working, my intention is to create an application that mirrors the functionality of the web app but uses a user interface that is more applicable for a smaller screen. It obviously needs to be easy to use, I am expecting complications and issues with the HCI and UI development of the app but I am sure I will be able to overcome them in the time I have given myself.

Software Architecture, Algorithms and Data Structures

Architecture

My final system will be much larger than my prototype along with being vastly more complex to create. It follows a three tier architecture model, with a few additions. It will contain many more MySQL tables that contain data I extract from the Valve API, multiple jsp/html pages will utilise this data to display useful and accurate information to the user. I have developed a mockup diagram to describe my current vision for my final system that is here:

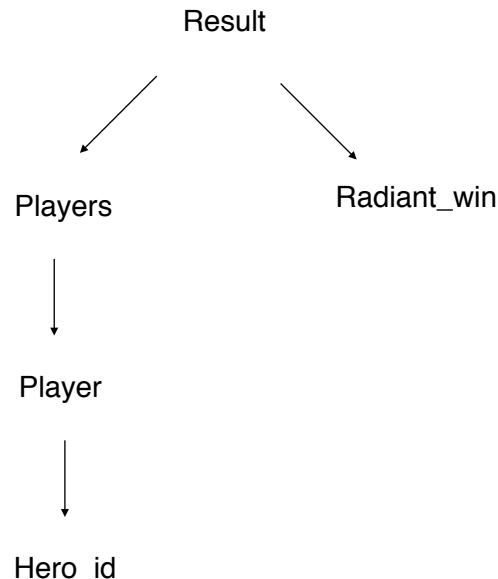


As you can see I also plan on introducing tables for each hero, so that I can track how win rates both with and against each hero changes over week. With this I can track changes in how heroes matchup and work together, I can calculate the standard deviation of the hero rates and calculate how changes in the game impact the numbers. My logic layer processes all of the data from the database and sends the required data to the web app, the android application will have logic within it to extract required data.

Data Structures & Algorithms

In this section I will go my main methods in my logic sections and discussing the main algorithms and data structures I have used in them, along with why I decided to use them or write them like that.

My *Data Processor* class is my parser that gathers data from Valves API and sorts it down into useful information which is inputted into the appropriate tables in my database. The *parseparticulargame* method contains the algorithm I use for processing the data I receive from valve for any game I want. The XML file that is outputted by the API has a tree similar to the following, without parts that I am not concerned with [2]:



My method takes the Game ID and saves that as the first element in a String array of length 12, it then checks who won the game and saves either Radiant or Dire as the second element in the array. It then parses through the players and gets each hero id, the first 5 heroes are Radiant and the last 5 are Dire. It produces an array similar to the following

[Game_ID, Winning Team, RH1, RH2, RH3, RH4, RH5, DH1, DH2, DH3, DH4, DH5]

Where RH/DH represent Radiant/Dire Hero. It is then fairly simple to enter the correct data into the database as we know which heroes have beaten which other heroes and we can process each of the winning teams heroes beating each of the losing teams heroes. A similar method will be used when processing hero synergy as it will parse the data in the same way, it will just process the 5 winning heroes winning with each other and the opposite for the losing heroes.

Drafting Assistant

My *Database Manager* class is my main logic layer, it does all of the calculations to turn raw data from my database into appropriate data to output to a client who is using the website. When a user inputs a hero to the form it goes through a few methods to produce advised heroes based off the databases statistics.

The *getparticularhero* method takes a hero id and returns its win % against each other hero as an array of int arrays of the following format :

[[id,%],[id,%],[id,%],.....]

This array is then input into the *getworstheroes* method which returns only the 10 heroes with the lowest %, ie the heroes that beat the particular hero most. It does this by sorting

the original array and putting the first 10 elements into a new array. This can then be outputted for the website to display.

When a user enters more than one hero the *getgrouphero* method is called in which it runs *getparticularhero* for each hero entered and then combines the arrays generated by mean averaging the win percents, it then runs *getworstheroes* to get the best heroes to pick versus the multiple heroes entered.

When I implement the second database and set of methods to take into account how many times a hero has won/lost when on a team with a particular hero the output will take into account both hero vs hero data and hero + hero data. The system will calculate advise hero picks based on the win rates against the heroes the enemy has picked and the win rates when played with the heroes your team has picked, it will work out the mean average and present the best heroes. I plan for my Android App to do all of the logic in a similar way.

Advised bans are effectively opposite to advised picks, it will calculate how well a hero performs when against the heroes your team has picked and how well it performs with heroes the enemy team has picked and advise the top heroes. This way you can see which heroes the enemy team is likely to pick and ban them!

I chose to use arrays for storing my data as it is the easiest way to store many types of data and be able to access any element at any time, I can store arrays inside arrays to group hero id's and percentages inside one data structure. Overall arrays were the obvious choice as opposed to other data structures that store group data, easier to navigate and take specific elements and useable with many data types.

Personal Statistics Page

The plan for my personal statistics page is to have the system storing the games a player has participated in. It will hold all of the data from that game that we need, so the winning team, the hero the player played with and all the other heroes. It will be able to display the players win rates with heroes, win rates against heroes, win rates with particular roles, win rates with particular players as well as more advanced statistics around these such as the changes in win rates of a hero globally when related to the hero and the standard deviation of the players win rates. I plan to insert steam login functionality so a player can log in via their steam account and that will automatically send the system their account id so the system can build the table containing their games.

Global Statistics Page

I plan to display global statistics page that uses the weekly hero tables to display changes in hero win rates and games played, it will display the standard deviation of a heroes win rate over a long period of time. The user will be able to enter a hero and the page will then show tables and graphs displaying statistics on the hero and how they have changed over

time. It will also show as a base the heroes that have dropped and risen in terms of win rate that week along with the heroes that have the highest and the lowest deviation on their win rates.

Android Application

My android application will mainly be the drafting assistant, it will be the same functionality as my web based drafting assistant with a cleaner UI designed for finger use on a small screen. I have a concept of a drafting 'game' where you pick heroes against different skill setting AI's that pick against you, if you end up having higher than a 50% win rate you win the game, if you have lower you lose. This game is a possible functionality that will take a fair bit of effort but is something I am passionate about including.

Bibliography

- [1] Curse, Wiki entry on game modes - http://dota2.gamepedia.com/Game_modes
- [2] Valve, Particular game XML - https://api.steampowered.com/IDOTA2Match_570/GetMatchDetails/V001/?match_id=1040766302&format=XML&key=F132852481FC966F6B0185C7EA6BE5A3
- [3] Valve, Hero ID's - https://api.steampowered.com/IEconDOTA2_570/GetHeroes/v0001/?format=XML&key=F132852481FC966F6B0185C7EA6BE5A3
- [4] Valve, DOTA2 API info - https://developer.valvesoftware.com/wiki/Dota_2_Workshop_Tools/Scripting/API
- [5] Zoid (Valve Dev), info on using the API - <http://dev.dota2.com/showthread.php?t=47115>

Appendix

CO3015/CO3016/CO3120 Career Plan

Matthew Bennett

1. Where do I want to go after graduation?

I want to get into the video games industry via developer jobs, I am incredibly keen in moving to the USA as the majority of companies I want to work for are based over there. My dream job is to be a developer at Blizzard Entertainment, having played their games for over 10 years I am both incredibly passionate about their games and about learning to develop them.

2. What will I do this academic year to get there?

I have started preparing a CV and cover letters for a few job applications in various areas, I am mainly looking at Java Development posts but am interested in applying for a wide variety of positions. I am planning to apply for around 10 jobs by christmas, I plan to apply to a load of different positions so that I have a higher chance of going straight into work after graduation. My priority is becoming financially independent as soon as I can.

3. How does my project contribute to my career?

My project is helping me to enhance my skills of making a complete system swell as generating evidence of what I am capable of. It shows off my both my Java and HTML skills and my ability to understand the user and make my system highly usable for a client. My project being video games related will also show my passion for the industry and my desire to create not only games but the software that surrounds them.