Miles Gillis
CS5200 – Final Project Writeup
12/11/14

Problem statement. Tell us what were you trying to solve

Dota 2 is a a popular computer game of a new genre known as ARTS (Action Real-Time Strategy) or MOBA (Multiplayer Online Battle Arena). These types of games are incredibly complex, consisting of over one hundred unique characters each with their own unique abilities. Any given match typically pits five characters against five characters, which translates into an astronomical number of combinations of these characters and the abilities. The properties of the elements in this game are governed by a pattern of rules and mechanics not uncommon to fantastic video games (health and mana, move speed, attack damage, cooldowns, etc).

Herein lies the problem: the data about the elements of this game, as they are currently found online, do not allow comparison, filtering, or ordering along the values of these mechanics. Heroes and their abilities are presented in an all encompassing form, and there is no way to query them based on the common criteria. This is what my app seeks to do.

Proposed solution.

By putting all of the elements in the game into a relational database, the data will be forced into appropriate patterns. These patterns will allow execution of queries based on any level of depth of the related elements.

APIs

I actually pulled my data directly from the game files of Dota 2. There's a suite called Nem's Tools with a program GCFScape that makes it possible to extract text and other files from the compiled game file itself. There were four text files relevant to the game elements of interest and they were in VDF format (Valve Data Format), a type of key value pairing used by Valve software. A third party library by Github user “nosoop” was used to translate these data into JSON elements.

Technology stack that you used

MySQL database
EclipseLink JPA
Apache Tomcat server
Unimplemented JWS endpoints?
Unimplemented AngularJS website?

Use cases. List and discuss the use cases

Filter or compare heroes based on any of their inherent attributes.
Calculate cost benefit ratios of different items for different heroes (because attributes may affect heroes

differently)
Calculate the mana efficiency of spells at different ranks
Calculate the cumulative cooldowns of each hero across their three non-ultimate spells (to determine who benefits the most from Phoenix's Aghanim's upgrade)
Find all abilities that cause any of a specific type of disables
Find all abilities that produce a certain effect cross referenced with a possessing hero who meets certain criteria
Display cast range of item abilities (not displayed in-game!)
Sort items based on mana cost of abilities
Calculate the amount of XP and gold gained by certain jungle camps
Find the neutral creep with the highest HP (best target for Clinkz's ultimate)

And many many more!

Data model