

Jeff Malone (malonejm@cs.washington.edu)
University of Washington, CSE 403, Winter 07 (Instructor: Stepp)
Project Proposal

Operational Concepts:

On the conceptual level my project is a search database for video clips on the Internet. The catch is that all keywords and descriptions for the video clips will be generated by users playing a matching game. Ideally, at least two people will watch a video at the same time and as they notice things in the video such as actors, actions, or even themes they type them into the application. The more matching descriptions you achieve with your partner the higher your overall score will be. When the clip is over, the matching values of participants are tallied up and added to a database along with the url of the clip. Once the game has been played enough, the search system should be fairly robust in terms of keywords and descriptions and since more than one person agrees on each of the keywords and descriptions, we can be somewhat more assured of their accuracy. So the high level idea is a video clip search system that uses keywords that are generated by actual people. This is interesting because no known algorithm can identify images as well as the human brain. The incentive to actually label the clips comes from the fact that the system is set up like a game with high scores and what not. Word matches in the game portion of this project would have to be somewhat loose, as spelling correctly can be somewhat problematic when you're trying to watch the rest of the clip. The clips themselves can be pulled from various sources on the Internet, such as YouTube or Google. My overall hope with this project is to see a more accurate form of video search on the Internet. This idea is inspired from existing web games like peekaboom (www.peekaboom.org) where this idea is used to label images.

System Requirements

There are two rather separate portions to this project, one is the search database and the other is the game to label clips in the database. Essential features of the database include a search and feature setup as well as different ways to format the output. People should be able to type in an actor or theme and have different results brought back to them in the form of a title, list of keywords, and a link to the actual video clip.

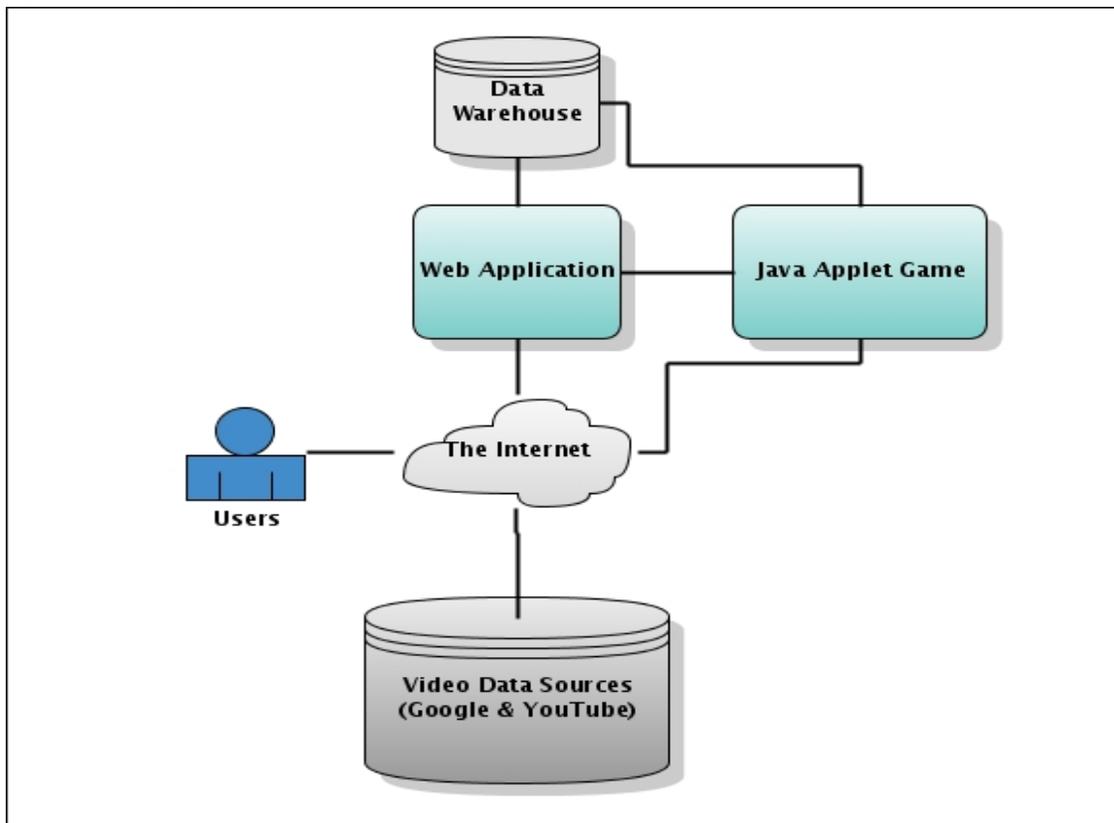
While the database portion is a pretty standard search set up the game portion of the project is much different. It needs to have a friendly interface where the users can log in and search for a partner to play the game with. Once the game starts they need to be able to clearly see the clip as well as where they're supposed to input their keywords and descriptions. Ideally the clips should run at the same time for all players, but clearly bandwidth will be a bottleneck to that idea. This will require a server that keeps track of players connections. When players are done viewing the clip, their keywords/descriptions will be compared for

matches. This match needs to check some sort of thesaurus if possible because I highly doubt people will use the same exact words for certain situations. These matching values and a corresponding score then need to be displayed to the users. The keywords can then be added to the search database and the scores can be added to a saved user name.

System and Software Architecture

This program will be a web- application with a database running behind it. For the database I'm most familiar with Microsoft SQL server, but I'm open to suggestions. The web- application part can be done with JSP or tapestry, both use Java (again I list these only because it's what I'm familiar with). The game itself can be a Java applet that launches off of the web- application itself. The applet will need to communicate with either a server or the other users directly, so it's going to need some networking code. Within this applet we can show a video clip that is stolen from some various data source, such as Google Video. So in overview, we need a web server hosting our search stuff, a database for storing keywords and links, and a Java applet that actually plays the videos and tallies scores. Each of these components will need to interact with each other. I've attached a diagram of how this interaction should work. I've also included a flow chart of how I expect users to interact with the program as figure 2.

Figure 1: High Level Overview



Lifecycle Plan

This project still has plenty of room for design and analysis, so much so, that I would dedicate at least a week, maybe two, to planning out all of the separate modules. The database, the web application, and the Java applet will each need some design time; however they should be able to be developed in parallel. Designing in parallel will require a hefty group of people and I would want at least two people on the design of each separate component. Once the design of each component was complete the groups would need to get together and analyze the designs to make sure that they can interface together. Once all the designs are in place and analyzed I would have everyone begin coding on the same component. This way we can ensure progress and hopefully finish each of the components in a timely manner. As far as testing goes I'd like to have Junit tests set up for all the Java components. Since all of the modules have Java components I would really like for all of the workers to have a good feel for Java. Any experience with web based applications is a plus. I think people who are actually interested and want to learn some of this stuff are just as welcomed as the people who already know how to do most of it. I would say that this project is going to be roughly thirty percent planning and seventy percent implementation. Instead of slipping into code- and- fix as soon as the design is done, I'd like to have a few releases in which the team tests the product and finds as many bugs as possible. This way we could have something to work with but constantly be making improvements.

Feasibility Rational

I believe that this project can be completed successfully because everyone in the class has a fairly good grasp on Java and it seems like it can be broken up into fairly big chunks of work. I'm assuming that we can get a machine to run our web server on as well as all of the software to run a database. I'm also assuming that Java has some built in media functionality, IE the ability to play video clips. Some risks include problems like bandwidth and lag. What happens when one of the players has to buffer the video? Is it realistic to think that users wouldn't quickly find a way to abuse the game? How much computation is needed to run a the matching algorithm at the end of the game if we including things like look ups to a thesaurus? What length of clips are we looking at loading? I'd really like to have a feature where you could click on a spot in a certain frame and add your own description right there but that's fairly unrealistic given our time frame. Other feature cuts include thumb nailing videos in our database and other fancy stuff that really just adds polish to the overall project.

Figure 2: User Interaction

