Application Development Diary

Kieran Warren  
100332570  
University of Derby  
kieranwarrenuk@yahoo.co.uk

TOPIC: Planning Approved Idea

DATE: 02/04/16 START: 13:00 END: 17:00

Today was the full planning phase for the project shortly after getting approval which included experimenting with a few different technologies and software to find which would suit myself and the product best. For the client side I experimented with Phaser and Cocos2D-JS game engines and I did not have good results in trying to get a grasp on either. Having no previous experience in Javascript would make this a big task to learn a new language that varies quite greatly from the languages I currently know, as well as having to pick up a new game engine in the next 3 weeks. I did however discover Intel XDK which so far I’ve found to be a great IDE for web based development and even Javascript mobile app development.

I also did some further research into web APIs relating to news and found the perfect solution. Prior to idea submission, it appeared my only options would be the APIs offered by BBC and The Guardian. However I have found Webhose.io which appears to be ideal. Compared to the other APIs, it offers the ability for far more specific queries and rather than being limited to a single website, I would get access to a huge number of different news websites.

An additional benefit to Webhose, which has influenced my final decision in the language used on my server, is that it offers a free SDK to use with a few different languages. That along with built in compatibility with SQLite and Sockets has made Python an ideal choice for the server side of the application.

TOPIC: Server Setup and Using Webhose

DATE: 08/04/16 START: 17:00 END: 22:00

I began to set up the server side of the application and attempt to understand the Webhose API and how I could use it to benefit the project. I’ve been able to successfully form queries that return valid results and I am storing the latest piece of news from each query and now I just need to use that data to find the most recent news topic which can be returned to the main server component to be distributed to players and also stored within the database for a player to return to if they particularly enjoyed the matchup of characters, locations and game modes.

I’ve also set up the supported topics to be from a text file to allow the game to support completely custom modes. For example, the system could be set up to use The Derby Telegraph as the news source and have people and locations relevant to the area, delivering a potentially very unique experience. Having custom people and locations should be relatively simple, the client must contain the relevant image files to the custom game and the code must make sure to load them correctly. Custom game modes on the other hand would not be particularly feasible without the users having to write some of their own code on both the server and the client.

TOPIC: Further Server Development

DATE: 10/04/16 START: 09:00 END: 15:00

Continuing on from yesterday’s positive results, I continued working on the server and communication with Webhose and setting it up so that it can be used effectively with the remainder of the server. This included creating functions that would be called from the main portion of the server which would return only the appropriate data. I implemented a InvokeWebhose function which will be called every time the game needs to be updated. This function then retrieves the date and time of the latest story related to each of the keywords from the file implemented yesterday from the Webhose results. Then from those dates, the latest game mode, characters and location are calculated separately and then returned to the core portion of the server.

I also began work on the main portion of the server. So that the server can potentially be run 24/7, I had to make a few design tweaks to make sure it would work. I’ve chosen to divide the server into multiple threads so that it can certain parts can be run concurrently. A minimum of three threads will be required for the server to function as expected. Two will be used to handle single player and multiplayer users separately, as each would require different data. The third will be used to communicate with new clients and then send them into the correct thread based on their choice through the client. A timer, which also executes on its own thread will also be used to call the InvokeWebhose function mentioned previously to get the latest data after a specified amount of time. The updated info would then need to be sent to the clients so that it can also be updated.

The biggest challenge I faced was getting the format of the data received from Webhose into a format that I could use with the rest of the system. The first part was getting the latest date and times from each of the search queries. The SDK offers a class item for query results which makes the date easily accessible. Then to find the newest item on the list, I had to remove non numerical characters and from that, I can simply compare all of the items to find the largest number for the date, and then narrow it down to time if there are multiple queries that have the same newest date.

TOPIC: Network Communication Experimentation

DATE: 15/04/16 START: 12:00 END: 17:00

Today I did some work between a basic C# command line based client and the current server using Sockets. Sockets appears to be an ideal solution as all of the data going back and forth between the client and server can be as part of strings. I decided to implement a few commands server side which can be received from the client and acted upon. After a few setup issues I was able to communicate and get the correct results being displayed on both the server and client.

I expected to receive more issues than I did using two different programming languages. I have used Sockets previously but only between the same languages. I expected some formatting issues when creating the data to be sent however this was not the case.

TOPIC: Building up the Client Side

DATE: 17/04/16 START: 16:00 END: 21:00

I have attempted to recreate my results in my originally chosen language, JavaScript. However, I had great trouble getting it set up and communicating and I feel that learning a new language, game engine and more may be too much to achieve within the next two or so weeks. For that reason, I decided to continue working with the code I have already written in C# and transfer the current networking code to a WPF application. That way I can still learn many new technologies while creating the client such as WPF, networked gaming and more while keeping some level of familiarity so that I don’t feel completely lost during the remainder of development.

After changing everything over to WPF, I started to see what WPF had to offer and how I could implement a game with it. I experimented with moving different objects with the keyboard and it requires a few extra steps compared to a game engine but having a language syntax that I’m comfortable with makes learning new tools a much more comfortable experience and don’t feel that I am trying to learn too much. The game mechanics will be relatively simple so that I can incorporate multiple game modes triggered by their topic being active in the news. With time being a big restriction, I want it to be more of a demo of having unique games and experiences being created from the API and the news instead of having a game that is of higher quality but doesn’t really experiment with anything new and unique.

I’ve already added some features to make game development using WPF a little bit easier including a sprite class that currently holds just an X & Y position and an image, and a game loop so that the window can be refreshed with new item positions a number of times per second.

TOPIC: Adding a database to the server

DATE: 22/04/16 START: 12:00 END: 15:00

Using SQLite 3, I have added a database into the server with 3 separate tables. The first table is a list of users, containing a unique username, a password and unique email address. SQLite offers the keyword ‘UNIQUE’ meaning that a column can still contain only unique items without being a primary key. This table will be queried from the user when logging in, or inserted when a user registers, first checking if the item exists and returning a response depending on its existence. The second table is a High score list, it uses the username as a foreign key from the username table, which is part of a primary key along with the score. Each user will only be able to have one high score in the table. The final table is to store previous combinations. These will contain the date the combination occurred as well as its settings. This list will be added to whenever the game combination is updated and the user can choose a combination in the database if they wish to play a custom game mode.

Manually querying these tables works great. All that needs to be done now is figuring out how to dynamically create entries and query the database from the client and send the results back in a format useful to the client.

TOPIC: Huge Progress

DATE 25/04/16 START: 10:00 END: 21:00

I managed to make some great progress on all aspects of the application today. I built up both the client and the server and defined some more communications between each other, to the point where the user is able to successfully query and choose from the list of previous game modes from the database and select an option or retrieve just the current settings for single player. Now that the settings are successfully transferred, all that needs to be done here is to create some simple games to utilize these settings. It was particularly satisfying successfully transferring multiple entries of the table and having them successfully appear within the window using WPF. I was expecting a lot of issues and bugs because I was sending each row of the table in a single string separated by commas similar to a CSV storage method. However, C# contains a string splitting method

The user is now also able to register or login using the application which is also set up to take advantage of the database stored with the server. I had some issues with the formatting of SQLite, particularly with how to pass variables into it from Python but I was able to solve this issue.

After that, I discovered some new issues with my implementation of using the Webhose API. When I enabled the API calls, it was crashing. I discovered that this was because some API queries were returning no results. E.g. the keyword had not been in the news lately, so I had to implement some error exceptions to be able to remove that keyword from the remainder of the code for when it is passed on to find the latest date between all of the keywords per category. Re-enabling the Webhose API also helped me to find a few other minor issues with my code that can be difficult to detect. When running within the same time period, it’s very likely that the search results will always return the same so I assumed all was well when it was returning the same results, however it seems that some values were set incorrectly and so the program was simply defaulting to the first item in each of the categories lists.

TOPIC: Some Gameplay

DATE: 26/04/16 START: 16:00 END: 20:00

Along with some generic bug fixes, my main aim today was to wire up some of the settings data into some kind of display with some user input. There were still some bugs with the data being sent over which were pretty easily fixed. Then to begin adding some on screen feedback to the received settings I had to add some background images appropriate to the game mode. For example, a football pitch and a boxing ring. Currently I’ve set just the text of a location to be displayed, providing it is one supported by the game currently. I have to work out a way in which to include better visual feedback based on the location that does not conflict with the backgrounds which are based on the game mode.

I was able to also get the character images into the game as well as a football and was able to add some controls to move the character around which works great, although is a lot more hassle when using WPF than doing it through a game engine. The minimum I hope to reach is to get a game working and some kind of high score system that will be stored in the server database and also accessed in the form of a leader board.

TOPIC: Game Development

DATE: 27/04/16 START: 10:00 END: 16:00

The actual game mechanics was the last major I needed to implement and I have done so, although very basic as I am limited by what WPF allows me to do. I’ve implemented some collisions that are working ok and currently, the aim is to get the ball inside the goal past the moving enemy. Every time you score a goal your high score increases and all that needs to be included now is a timer that ends the game and submits the score, although only updates if it is higher than the previous score of the user. I need to somehow adapt the game to reflect on the other available game mode settings. The games themselves are still not my biggest target I just want to be able to show how an application could be adapted by the latest news.

As well as introducing some gameplay and a high score system, I always fixed a number of bugs and issues mainly server side. I started testing multiple users on the server and realised that I had not set up each new client to move themselves to a new thread. I did some brief testing using multiple clients (albeit through the same machine) and all functions appeared to work as expected through both clients such as logging in, getting the latest game settings and more.

Tomorrow or this evening I will make a few final additions, some final testing and also include a conclusion on the application, my personal performance, what went well and what didn’t

TOPIC: Final Additions and Thoughts

DATE: 27/04/16 START: 19:00 END: 21:00

I’ve chosen to, instead of creating separate game modes, to just keep the same game but change the background. As the assignment is largely based on independent learning, I feel that adding more basic games would be redundant as I feel I’m already very confident with the basics of video game mechanics. Instead, I want the application to really show that it is making use of the Web API and the database which are both parts where I really feel I was able to grow the most. I’d never used a Web API before so it was great to see what I was able to produce from the service. I also had my first opportunity to use SQLite which introduces some different keywords and I also used it within a language to generate some commands dynamically which brought up some interesting issues with some data being passed to the database. This seemed largely down to the dynamic variable types in Python.

Using WPF for the client was also a new experience for me. I had never used it before and only used Windows Forms very briefly. It gave me the opportunity to use XAML, which was my first experience with a Data Description Language. XAML actually really helped with building the client and made some things easier than if I were to do it in C#. Although it does not allow for any kind of dynamic updating without some extra steps which is a bit of an issue when developing a game that requires regular updating. I was somewhat aware of this possibly being an issue when changing from JavaScript for the client but I think it was a good decision as I’d have needed to do much more programming in a language that was almost completely new to me.

The final application was one that I felt did a good job of showcasing the main aim of the application, which was to demo that unique combinations of games could be created from the news and getting what I wanted out of Webhose certainly took a lot of time. I did purposely choose not to spend too much on developing the main game as the real aim of the assignment was to gain new skills and I feel very confident that a game even twice the size would not have really pushed me to my limits and the time spent on the game would have only taken away time from putting the main idea for the game into practice.

I really think this is a potentially great idea for a game and I will very likely be continuing development of the game and will mostly concentrate on expanding the game component of the application with the use of a game engine as I feel the server component is already of a high quality.