Nathan Westover

I SYS 520

Prof. Allen

Wednesday, December 5, 2012

Final Project Write-up

Executive Summary:

Movies have become one of the biggest sources of entertainment in today's society. They allow an individual to escape reality for a time and experience the impossible. In the past, the best movie experience was at the movie theater. Home solutions were enjoyable, but not very similar. However, as technology has advanced and the prices of home theater equipment have steadily declined, many people have purchases a home entertainment solution that, while not as impressive as a movie theater, can be comparable.

With the home technology in place to adequately supplement movie theaters, people have begun to expand their collections of movies. In fact some have even become such avid movie fans that they have begun to collect an extra-ordinary amount of movies, constantly building their home collect. With the changing formats from VHS to DVD to Blu-Ray, some have even purchased duplicates of their favorite movies in the latest format. The problem that this creates, is that someone who has purchased so many movies, often times begins to forget what they own, sometimes resulting in duplicates.

This project is designed to assist avid home movie fans to log their movie collection in an excel sheet. Instead of having to manually input information such as the title, the director, the rating, the year, etc., they only have to type in the name, and the program does the rest of the work. In addition, it pulls down a synopsis of the movie and the movie poster.

Implementation

In order to complete this project, I determined that there were several elements that this program needed. The following are some of these components and their purpose:

1. User form – The form would be used to allow the user to input the movie to search
2. A web-query - Gather all the desire information and put it into a spreadsheet
3. Transfer Procedure – This is a sub procedure to transfer the desired information to the correct sheet in the desired order.
4. Format Procedure – This procedure is designed to take the data that has been gathered and format it so that it looks the desired way.
5. Sort Procedure – This procedure is designed to sort all of the data so that you can add new movies and then sort them

6. Edit Procedure – This procedure is designed to allow the user to take the data and edit anything they want.
7. Ribbon Control - Buttons were placed in the ribbon to control the program

While originally I thought this was not going to be that difficult, several challenges arose that made it much more difficult than expected.

User Form

Creating the user from was really quite simple. I created a form that allowed the user to input the Movie name and then click a button that would begin the search. (Figure 1)

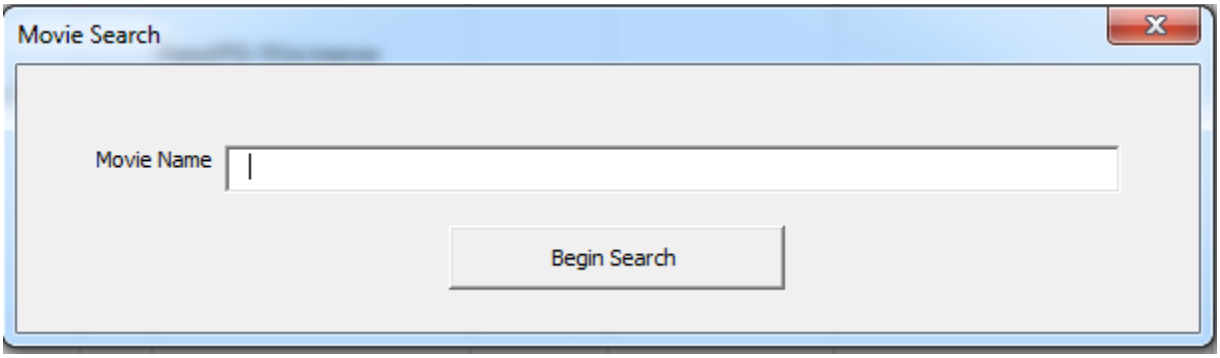

Figure 1 When the ribbon button "Add Movie" is pressed, this form appears

Web Query

Originally for the Web-query I thought that all I would have to do would be to record a macro using the web-query wizard and just substitute the name that I wanted just as if I was getting stock data from Yahoo. I tried several different websites with no success. No website categorized movies in that fashion. To solve this problem, I decided to use the Prof. Allen's agent code and control the browser to get what I wanted done. I was able to edit the code and direct the agent to use the website's search bar to look for a movie. After trying several websites, I determined that IMDB.com had the data I wanted and began to work with that. (Figure 2)

What I had not planned for was the fact that several movie names would come up with lots of results. In order to solve this, I had to get some help from Prof. Allen on how to pause the search until the user clicked on which movie they wanted.

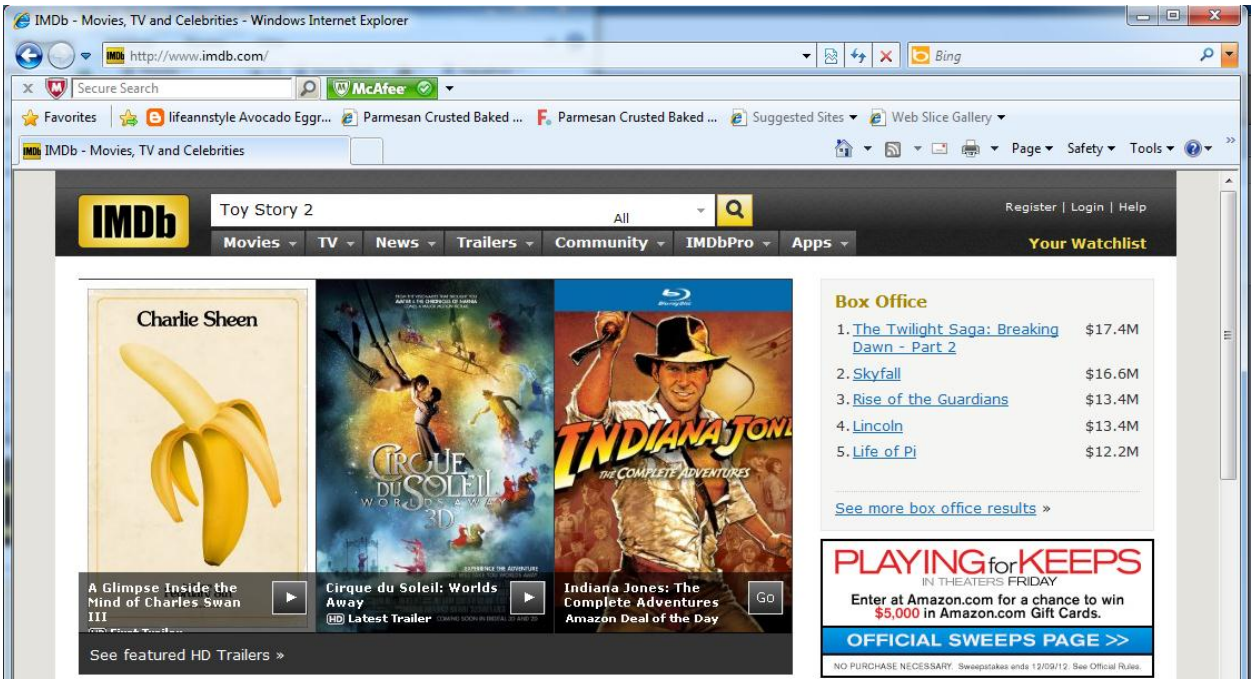

Figure 2 imdb.com with search bar and input. Not seen by the user.

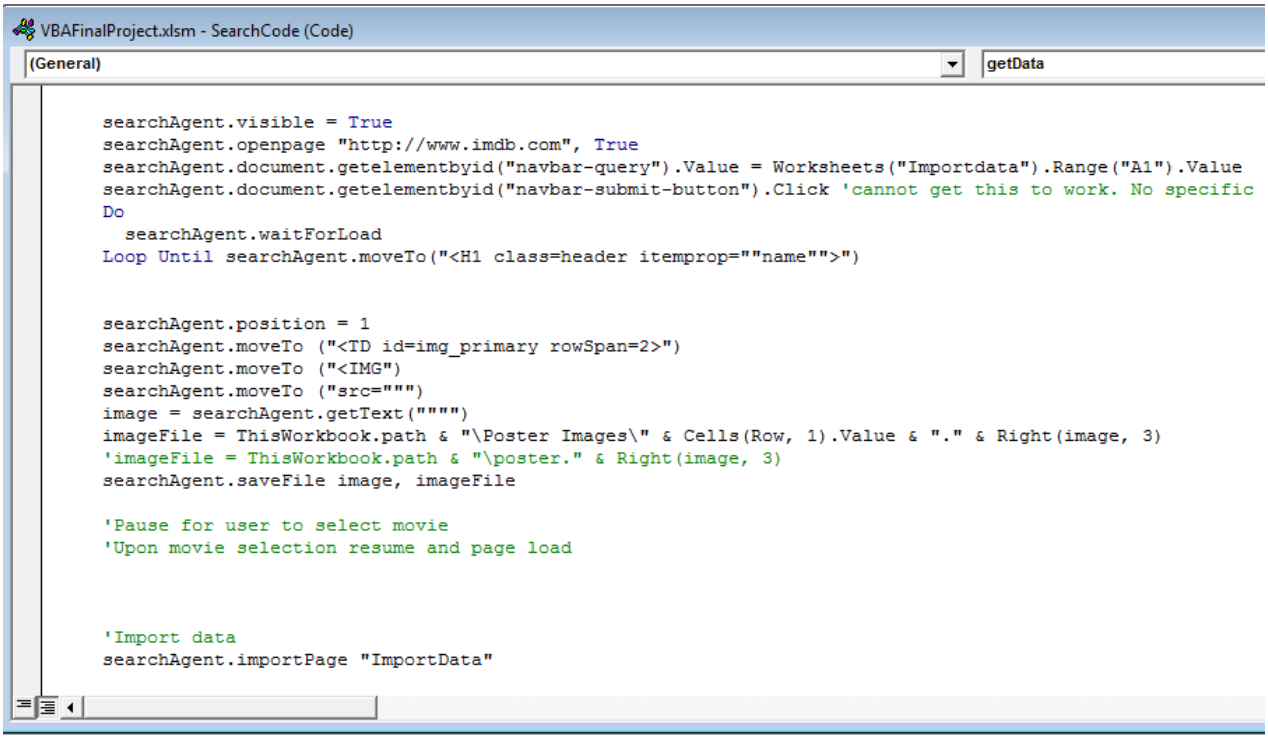

Figure 3 This is the reference to that agent code to control IMDB and to get the desired data. This shows the DO… Loop that was created to stop the program until the specific info was found on the page allowing the user to select a movie from the search results

What we ended up doing was setting up a loop where the program would stay in the loop until the user selected the movie that they were searching for from the search results. (Figure 3) A certain html line was found on the page. That html code however was unique to all movie display pages. So until the user selects a movie the program would not continue. However, once a movie is selected and the page moves on and loads, the program continues. It then downloads all of the page information into a separate sheet that is created to hold that information.

In addition to gathering the information about the web-query, with the assistance of Professor Allen, I was able to add code into my program that also downloaded the poster that matched up with that movie so that I could insert it into the spreadsheet later on.

Transfer Procedure

I wrote this procedure, so that it would search through all of the data that was imported through the web query and transfer it to the database spreadsheet in the desired order.

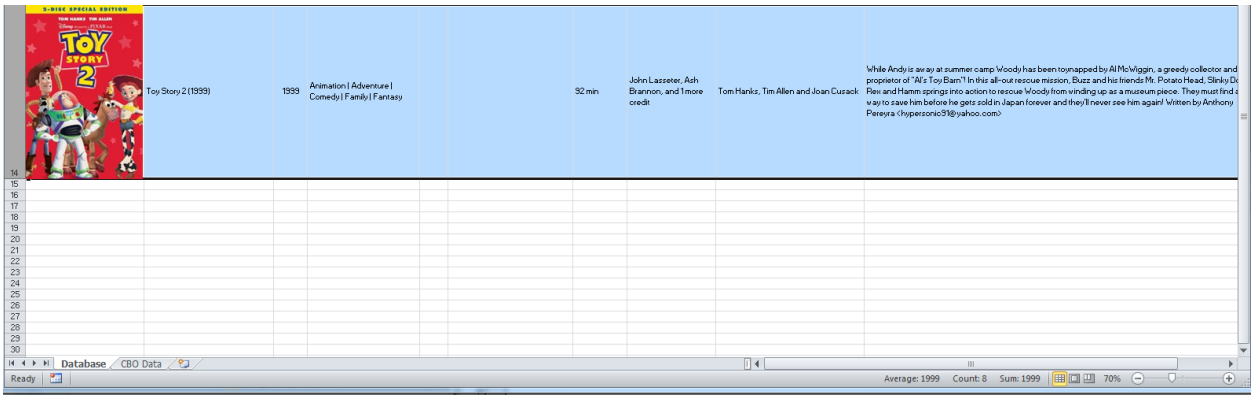

Figure 4 Data Transferred using the Transfer Procedure into the correct location in the database sheet.

Format Procedure

The next procedure went through all the data that was transferred into the database and formatted it so that all of the information from all movies that are imported is formatted the same. It also separated some of the data out into its own line, such as the year the movie was released.

Sort Procedure

This procedure was created so that once the user has added the movies to the database they can go through and sort them alphabetically with just one click. I thought that this would be really simple to do being that excel has a feature built in to sort, but this turned out to be really quite difficult because of the images. The way that excels treats images is different than text. So when I would sort using vba, the images would not sort with the data so they no longer lined up with the correct movie

```
Sub Sort()

'does not work properly. Try and fix
        ActiveWorkbook.Worksheets("Database").AutoFilter.Sort.SortFields.Clear
        ActiveWorkbook.Worksheets("Database").AutoFilter.Sort.SortFields.Add Key:= _
        ActiveCell.Offset(-1, 0).Range("A1"), SortOn:=xlSortOnValues, Order:= _
        xlAscending, DataOption:=xlSortNormal
    With ActiveWorkbook.Worksheets("Database").AutoFilter.Sort
        .Header = xlYes
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With

    Row = 2

    Do Until Cells(Row, 2).Value = ""

    ActiveSheet.Shapes.Range(Array(Cells(Row, 1))).Select

    Selection.Cut

    Cells(Row, 1).Activate

    ActiveSheet.Paste
     ' look for value
     'move to Range Location

    Row = Row + 1

    Loop

End Sub
```

Figure 5 Sort procedure that was created to sort the data and keep the images linked to the files.

To solve this problem, I created a new procedure (Figure 5) that first sorted the data, and then linked the picture up to the data and then repositioned the picture to where the data moved to. In order to make sure this could happen, I had to change how the pictures were named when they were inserted into the database sheet because excel just assigns then the next sequential number. I then was able to link that name up with the row that the data moved to.(Figure 6 Before, Figure 7 After)

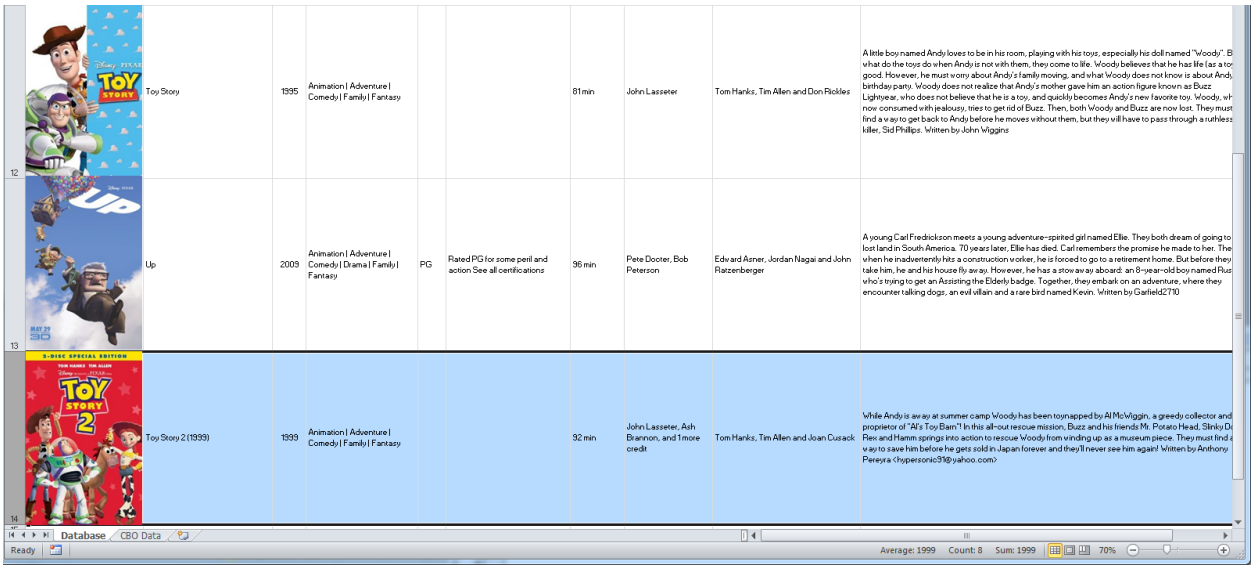

Figure 6 Before sort button was pressed.

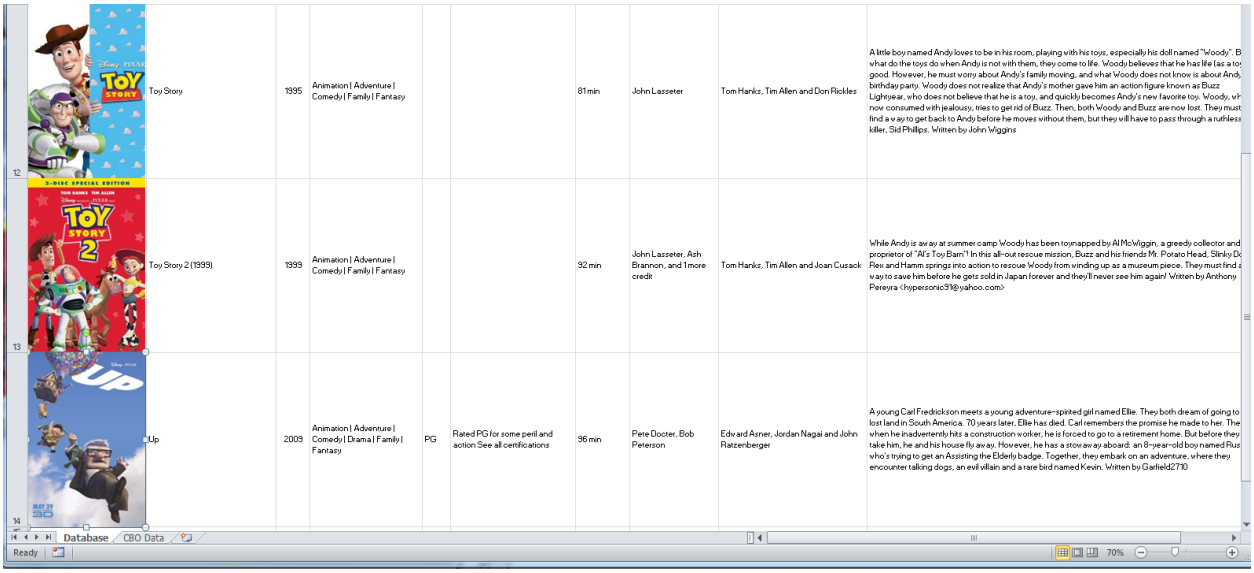

Figure 7 After the sort button was pressed

Edit Procedure

I wanted to make sure that if some of the data imported incorrectly or was blank, that the user could go in and change what they wanted to. In order to facilitate this, I had to create a new form that, when called, would reflect the information of the desired movie. To keep it simple, I set it up so that whatever was in the active cell would be the row of information that the user would see. The form then allowed users to go in and edit the data and save it back to the database.

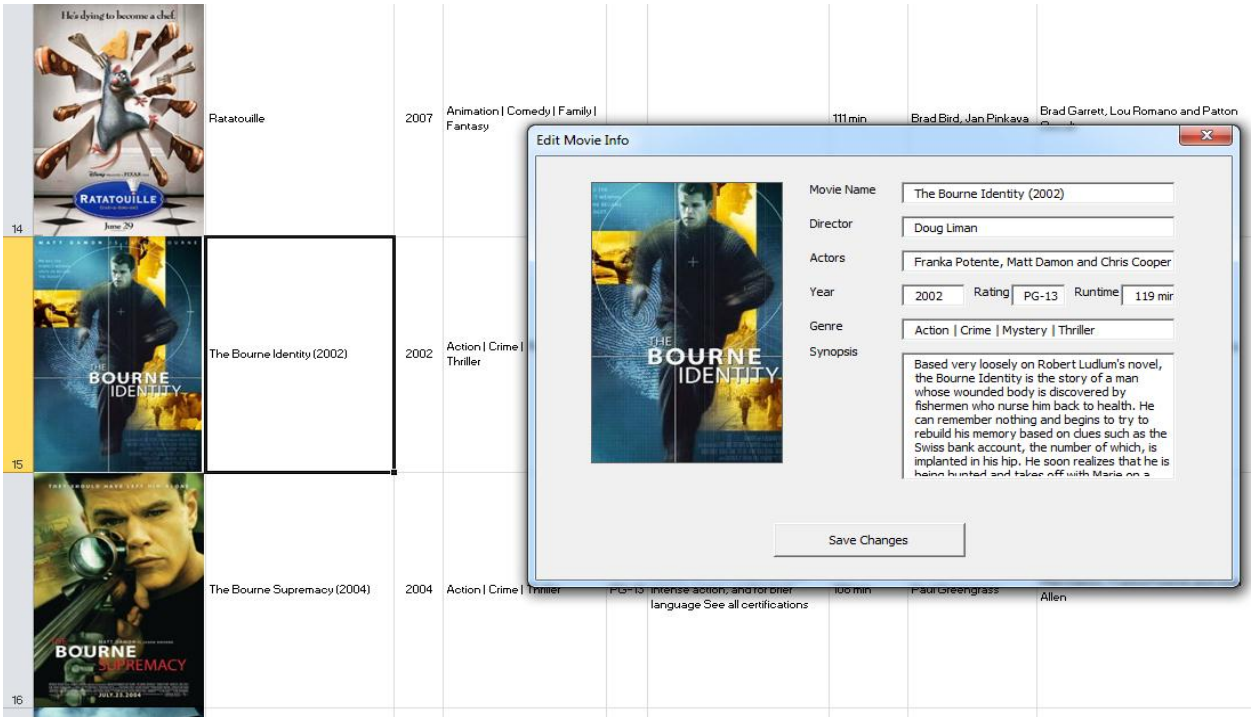

Figure 8 This is the Edit Movie Form. It was called while "The Bourne Identity" row was active.

Ribbon Control

To improve how the project looks and feels, I added a new tab at the top called project and then added buttons to control the program. (Figure 9)

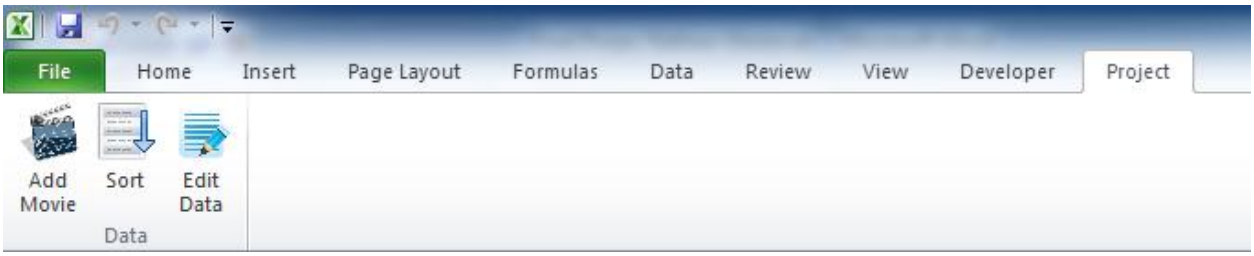

Figure 9 Project Tab with three ribbon buttons

The three buttons do precisely what they say:

- Add Movie: Calls the Movie Search form and allows the user to add a movie into the database
- Sort: This automatically sorts all of the data and links the images to them.
- Edit Data: This calls the Edit Data form and allows the user to edit any of the data

Learning

Throughout the process of writing this program I encountered several challenges that I was not expecting. I thought that a simple web-query would achieve my desired result, but I was very wrong. As I began to write the code and encountered problems, I had to draw upon the things we had learned in class and expand upon it. One of the biggest things that I learned was how to better use and control the agent. When we learned about the agent in class and how it can be used to control webpages, I struggled to understand exactly how it was done. For this project, it became the key element that I need to use. I learned about its uses, and how to look at and understand the source code of a webpage, and use the agent to find various parts.

Another area that I learned a lot when developing this project was how to work with the images that I was inserting into the project. With regards to the sort procedure, I had a very challenging time figuring out how I could link the images to the data since they sit on top of the cells. I had to figure out how excel treats the images and then how to move them to the proper location after the data had been sorted.

In general, all of what I had learned in class, which I pretty much knew none of it before, seemed to come into play in some form or fashion helping me to connect how it is used in programming with VBA.

Assistance

For this project I did receive some help, although it was minimal. The help that I did receive was from Prof. Allen and was specifically related to the agent and how to manipulate it to achieve my desired result.