**Testing and Evidence Log**

|  |  |
| --- | --- |
| Before getting into the programming, I had to do some planning and set up some project management tools. Shown to the right is a screenshot of my trello board, which will help me keep track of this program and my progress on it. The planning I have done is in the Planning document. |  |
| Here is the testing plan I set up for the first section of the program that I want to get working. For now, I will just code it so that the program prints to the shell, so I know that it is rating. This will not be in the final program, but I need a way to tell the rating of the movie before I set up the search option. | Test plan for setting up the rating frame:   * Press no rating   + The program should print no rating into the shell * Press rating 1   + The program should print rating 1 into the shell * Press rating 2   + The program should print rating 2 into the shell * Press rating 3   + The program should print rating 3 into the shell * Press rating 4   + The program should print rating 4 into the shell * Press rating 5   + The program should print rating 5 into the shell |
| In version one of my program, I set up the variables and the tkinter for just the rating frame. For testing purposes, I coded it so that the rating that the user chooses out of the radio buttons will print to the shell. However, a few problems presented themselves. First, regardless of what button is pressed the program will only output “No Rating”. Secondly, the layout of the radio buttons is not in line. The No Rating button is not aligned with the others, and it is confusing to look at.  I will fix these problems in version two. |  |
| In version two, I got the radio buttons to work as expected, and fixed some issues with the alignment in the layout.  The first problem was that pressing different radio buttons did not change the variable they were assigned to. This was because they were assigned to a regular variable, and not a StringVar. This was simple fix, I changed self.rating to a StringVar and when I printed it in the printrate command, I put .get() at the end of it so it would actually print out the value I wanted.  The second problem was that the layout of the program was confusing. I fixed this by adding padding to the radio button frame and the “Your Rating:” label, and aligning that label to the north and the radio buttons to the west. |  |
| To check that everything was working before I moved onto the next component of the program, I went through the small testing plan I created (two log entries ago). The video for this is in the videos folder, entitled “TESTING 1 – RADIO BUTTONS”.  With the testing done, I had finished the first programming component of the program. I could now get to work on the next one. |  |
| Here is the loose testing plan I set up for the second section of the program. Essentially, it’s the same as with one movie, but here I have listed some other things that I need to be aware of. This is to do with the next and back buttons, and when they should be disabled, and how movie ratings should be able to be kept as the default if they haven’t been rated, and how ratings for each movie should not be forgotten. | Test Plan for Multiple Movies:   * Press next button repeatedly   + Should switch movies until it reaches the last one in the list, where the button will then disable. * Press back button repeatedly   + Should switch movies until it reaches the first one in the list, where the button will then disable * Rate all but one movie   + Should keep the rating for the unrated movie as No Rating, but the others should have ratings * [Note that the ratings for each movie should save – if I rate one movie as three and then rate another movie and return to that first one, it should still be rated as three, with the three radio button selected] |
| In version three, I added some more movies so there were four movies in total to cycle through. I added back and next buttons, though at this point the back button does not do anything. The next button, however, calls a method called “go\_next” when it is pressed. This adds one to the “position” variable, and changes the relevant label and variable so that the user can rate the next movie. It also calls another method called “check\_pos”. This method checks the position variable and, if it is on the last movie in the list, will disable the next button to prevent the user pressing it again and prompting an error message. | Testing video in videos folder as “TESTING 2A – NEXT BUTTON” |
| In version 4, I got the back button to work. The back button calls the “go\_back” method. This functions almost identically the same as “go\_next”, but it puts the position back by one. I added some code to “check\_pos” so that it also checks if it is on the first movie in the list, and then disables the back button to prevent errors. Because the program starts on the first movie in the list, the back button is disabled by default at creation. |  |
| Now the multiple movies component was done, I tested to make sure it was working (using the test plan detailed a few log entries ago). The video for this is in the videos folder, named “TESTING 2B – MULTIPLE MOVIES”. Note that in the shell, each rating when you press a radio button prints out twice because I am changing the rating attribute for the relevant movie classes and want to make sure that it is the same as the rating assigned to the rating variable. After this testing, I was done with the multiple movies component. |  |
| Here is the testing plan for the next component, setting up the search frame. This component will not actually have any functionality for the searching, so for testing purposes the rating that is meant to be searched for will just be printed out into the shell. | Test plan for search frame:   * Select to search for movies with no rating   + Prints out no rating (it cannot actually search yet) * Select to search for movies with a rating of 1   + Prints out 1 in the shell * Select to search for movies with a rating of 2   + Prints out 2 in the shell * Select to search for movies with a rating of 3   + Prints out 3 in the shell * Select to search for movies with a rating of 4   + Prints out 4 in the shell * Select to search for movies with a rating of 5   + Prints out 5 in the shell |
| In version 5 I create the search frame. At the moment this frame can’t actually search for movies, but when you select a rating to search for and press go, it will print out into the shell. This is mostly set-up for when I create the summary frame, and this will prompt an actual search for movies. Unlike the other frames, this is only a local variable because it will be visible on both screens and doesn’t need to be hidden. |  |
| Once I had finished the search frame, I followed the testing plan I made for it. The video for this is in the videos folder, named “TESTING 3 – SEARCH FRAME”. Next, I had to combine the two frames I’ve made into one program. |  |
| In version 6, I combined both the rating frame and the search frame into one program. I put the variables needed for both frames at the top, then all the code for the rating frame GUI, then all the code for the search frame GUI in the init method of the MovieRaterGUI class.  Everything still functioned as before when I moved them into this program, but I had to make a few formatting changes. I changed the background colour of the searching frame to a light grey to differentiate it from the rating frame, and I added some padding – most notably between the two frames – to space things out. |  |
| I tested that the combined frames worked as expected. There was no testing plan needed for this, as it is the same functionality as in previous tests. The video is in the videos folder named “TESTING 4 – COMBINING FRAMES”. Next up was to get the search to actually work. |  |
| Here is the test plan for the summary frame part of the program, which will add functionality to the search aspect. Upon selecting a rating and pressing go in the search frame, the rating frame should hide and the user should be taken to the summary frame which will show all the movies with that rating in a scrolled text box. If there are no movies with that rating, there should be a message along the lines of “There are no movies with this rating!”. | Test plan for summary frame:   * Search for no rating, and then a few different ratings after having rated no movies   + All of the movies should show up under the no rating category, and the others should be empty. * Rate two or more movies the same thing, e.g. 3   + Both of those movies should show up under the relevant rating, while the other movies should not have changed. * Change the rating of each movie so they all have a unique rating   + Searching for each rating should bring up the relevant movie, they should have all changed their rating and movies should not show up in the lists of their old rating. * Trying to search without selecting a rating   + Should not open the summary frame unless there is a rating chosen to search for. * Pressing the back to rating button on the summary frame   + At any time on the summary frame you should be able to press this button to return to the rating frame. Going to and from frames should be able to be done repeatedly. |
| In version 7, I set up the GUI for the summary frame. At the moment the button does not work, I have hardcoded some example text in the scrolled text widget, and there is a gap in the information label. Once this is integrated into the main program, this frame will only show up once search\_rate has a value so the label will no longer be a problem, there will be a method to put movies in the scrolled text, and the button will call a method that switches back to the rating frame. For now, this is just to sort out the layout of the frame. |  |
| In version 8, I added the summary frame to the program. I also changed the method for the “go!” button in the search frame and added a function for the go back button in the summary frame so that the rating and summary frames can be switched between.  I put the GUI setup code for the summary frame in the init method for the GUI class, but did not grid the frame. This is because the frame only gets gridded when the go button is pressed, after the rating frame has been removed. Then when the back to rating button is pressed, the summary frame is removed and the rating frame is re-gridded. |  |
| In version 9, I added to the search method so that the names of the movies with the rating being searched for shows up in the scrolled text on the summary frame. I added a list for search results, and when the search method is called all of the movies with the rating being searched for are added. If no movies have that rating, a message explaining this is added instead.  When I first tried this out, there were some problems. I did not rate any of the movies and immediately searched for movies with a rating of 2. This came up with the message as expected. However, when I went to search for movies with no rating, the message was still there. Then, when I pressed the go button again the movies printed out again, despite the fact that the scrolled text clears itself each time the method is called.  Eventually, I figured out what I needed to do was clear the search results list each time the method is called. After adding “self.search\_results.clear()” to the method, everything worked as expected. |  |
| Version 10 was only a couple of small changes. First, I made it so that the info label on the summary frame actually displays what the user is searching for. There is also a custom message if they search for No Rating, otherwise it doesn’t make grammatical sense.  I also added some padding to the summary frame so it wasn’t pressed up against the search frame, and I made the scrolled text not as wide because it was way wider than it needed to be. |  |
| Now that I had finished this, I needed to test to make sure that everything was working before I moved on. *Almost* everything worked as I wanted, except for at the very beginning when I pressed the go button to search before having selected a rating to search for. It didn’t break the program, but I would prefer if the user wasn’t able to do this. I will fix this before I continue with the project. | The testing video is in the videos folder, under the file name “TESTING 5 – SUMMARY FRAME”. |
| In version 11, I fixed the problem with being able to search before pressing any of the radio buttons. In the init method of the GUI class I added a line of code to set the search\_rate variable to the no rating option, so it selects this by default as soon as the program starts. This means that there is always an option chosen, so the searching before choosing something to search for is no longer a problem.  (This commit in github was also accidentally entered as version 10. It is version 11, not version 10). |  |
| In version 12, I was getting my program to a point where a user could use it for testing. I removed all of the print statements that were there for testing (I also renamed the printrate method to set\_rate, as that is more accurate to what it does now). I also added some more movies to the list because before there were only four, and a user would expect a wider range of movies. |  |
| Before I continued, I wanted to catch as many bugs as possible before doing the end user testing. So, I ran through the test plan with my program. I did not catch any bugs but I did see a way to improve the program, which I will explain in more detail in the next log entry. | Video is in videos folder, titled “(voiced over) TESTING 6”. |
| Something I noticed when I was testing is that sometimes, when you have movies rated multiple different things, the user may want to search for more than one rating at once – e.g. movies rated 4 and 5. Radio buttons would not allow for this. So, I made a copy of the program and started on version 13 – turning the search radio buttons into check buttons.  I altered the GUI code to allow for check buttons. Then I had to update the search method. I turned the search\_rate variable into a list and looped through the check buttons – if they had their on value the rating would be added to search\_rate. I also made some changes to the go button and the movie display. Because I didn’t want the user to be able to search for nothing, I made a method linked to the check buttons that disabled the go button if no check buttons are selected. Also, because multiple ratings can now be searched for, it wasn’t clear what rating each movie was when you searched for more than one rating at once. So, I changed the movie display to show the ratings as well as the movie titles.  This was all done in a copy of the program so I can show these two different versions – the check buttons and the radio buttons to the end users and get feedback on which they prefer. |  |
| To get feedback on aspects of the program through fresh eyes, and to potentially pick up bugs/errors that I have missed, I got two end users to test my program and give me feedback. I showed them both the radio button and the check button versions of the programs and asked them which they preferred, as well as any other feedback that they had. Shown on the image to the right is the notes I took while they used the program.  The consensus between the testers was that the check button program was better and it needed to be clearer that their ratings were being saved.  There were also some other suggestions, like a more orderly and descriptive (i.e. sorting by rating and showing number of search results) summary frame, and a welcome screen at the start with instructions.  Tester 1 also found a bug within the check button program that needs to be fixed. |  |
| In version 14, I added a label that shows the user that their ratings of each movie are being saved. It shows the rating selected and the movie that was rated. It shows up each time a rating radio button is pressed, and is set back to empty when the user goes to the next movie or when they go to the summary frame and then back to the rating frame. This solved one of the biggest problems raised by both testers – that they were confused when the list of movies ended as they thought they needed to press next to submit their rating. |  |
| In version 15, I fixed the error message found by a user during end user testing. When they tried to search for a rating that none of the movies were, an error message occurred. I went to the line of code that had the problem and wrote a print statement for the relevant variable to see what was the problem. I then started the program, didn’t rate any movies, searched for no rating and then for rating 1. In the shell when I searched for no rating it printed out the objects, which is what I expected, but when I searched for rating 1 it printed out the “no movies with this rating” string. While this was not a problem before, now that I was printing out the rating of each movie it was trying to treat that string like an object.  I deleted the line that appended the no movies with this rating string to the list and instead moved the if length of list smaller than one statement to where the movie results are displayed. If the list is less than one then that string is inserted directly into the scrolled text. Otherwise, it works as normal.  I tested the search on ratings with and without movies and now everything worked as expected, and no error messages showed up. (I also deleted the print statement that was used to find the error). |  |
| In version 16, I added a label that would show how many movies were found in each search. I gridded the label above the back to rating button, and added a line in the search method that configures the label to show how many search results there were. For example, there are 8 movies in the movies list and when I searched for movies with no rating before I had rated any of the movies it showed up as 8 movies having been found. |  |
| In version 17, I added a welcome screen to the program so that the user understands how the program works before they start using it. I removed the lines of code that gridded the rating and searching frames, and instead put those in a method that is called when a button on the welcome frame is pressed. The welcome frame is forgotten (because it will not be needed in the program again) and is replaced with the normal program GUI. |  |
|  |  |
|  |  |