Project Proposal Components:

- **Project Description**: Magic Makeup is a makeup application game where the user applies different makeup products to a face, and a magical fairy teaches them about which placements look best. The goal is to try to get the highest score possible, based on quality of application.

- **Similar projects**: One game that is similar to mine is Beauty Battle by Kelly Cha, who created a makeup battling game where the user tries to apply time-based makeup based on what the client requested, and compete with three levels of difficulty AI. The user's side was on the right, and they could simultaneously see the AI's application on the left. My project will be similar because it also uses the application of makeup, with guidelines on where the product should be placed. Also, both of our games include scores. However, my game does not include AI, is not time-based, and is more about learning about the application of makeup rather than beating an AI. Additionally, I am including the overall theme of a magical fairy that teaches the user how to apply makeup. My game will also include pop-ups that educate the user on the product they have selected. For instance, when the user selects eyeliner, there will be a speech bubble that comes out of the fairy that says "Eyeliner is used to accentuate the eyes. Place this product on top of the upper lash line."

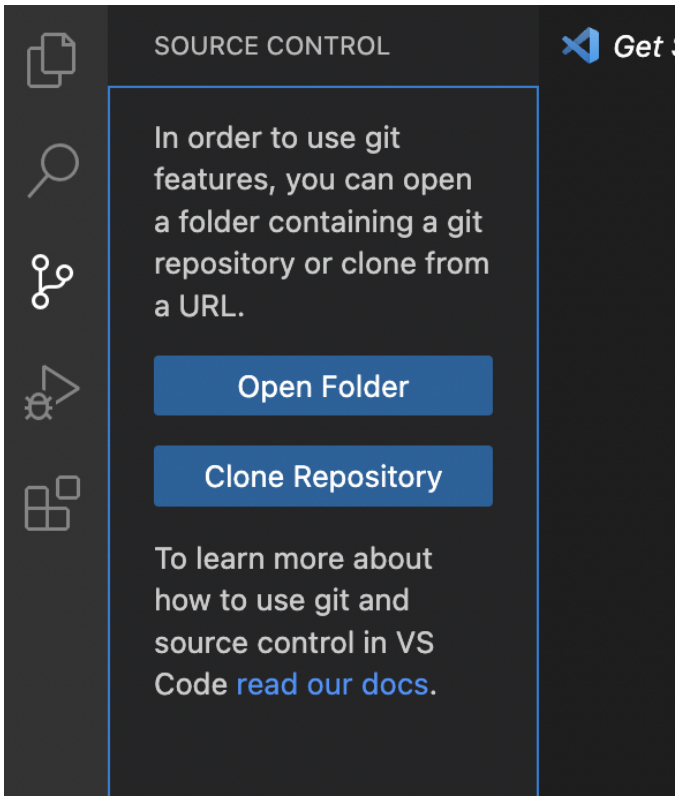
- **Structural Plan**: I am planning to have a makeup class, which has many self variables that are referenced throughout the class in different functions. I also need to use animations heavily, where I deposit color onto images, display the fairy on the screen, and print labels. I also need to have a score calculation function, and a feedback printing function so that users can see their scores and learn from their mistakes.

- **Algorithmic Plan**: The more algorithmically difficult part of my project will be to gather the scores based on how far the user places the product from the guidelines. I am also considering outputting a log at the end of what the user lost points on, so that they can improve in the future. For instance if they placed the lipstick too far out of the lines, they would get a message at the end that says "make sure to apply lipstick inside of the lip lines." I am planning on implementing this by checking if the user's application is within the "perfect application" dimensions. If it is not, I will deduct points and store the reason in a "feedback" variable which would then be printed at the end.

- **Timeline Plan**:

04/09/23 → Turn in TP0
04/16/23 → Finish most of user's game elements and starting screen, scores calculation not included, turn in TP1
04/21/23 → MVP complete, scores are calculated and printed at the end
04/25/23 → add more products to choose from, polish UI, possibly save scoreboard info?

- **Version Control Plan**: I will be using GitHub as my primary use of version control. I will be pushing new changes from VSCode to GitHub after each major feature is implemented. Below is a photo of the source control portion of VS code, where I can easily clone repositories and push directly to Github.



- **Module List** : I will not be using any modules.

- **TP1 Update:** Majority of my design plan has not changed, I am still planning to implement the algorithmic complexity previously discussed. However, I am now planning to include a flood fill portion to the makeup application where the user draws the outline of a shape, then clicks inside to fill it. This will add to my algorithmic complexity because it uses backtracking to figure out which pixels to fill next.

- **TP2 Update:** Majority of my design plan has not changed, I am still planning to implement the algorithmic complexity previously discussed. I am no longer thinking about the flood fill algorithm though because it more complex than anticipated. Also, I have manually implemented a text input field for the user to type their name in.

- **TP3 Update:** My design plan has changed, I have scrapped the idea of both scores and flood fill, and decided to more closely focus on the UI/UX portions of the project. I have added a slider so that the user can choose different brush sizes for their products, and the user will be able to see their final product at the end of the game. I have also implemented a smooth drawing feature and a makeup eraser which erases one product at a time. I have also added the blush feature which has a lower opacity and larger brush size than the other products.