I began by laying the groundwork for the game by setting up the colour and settings objects, then I created a step-by-step outline of the main game loop before implementing it. I knew I wanted to be able to easily add and remove colours, and alter the number of rounds and colour options, so I made sure to design around that from the beginning (although the specifications could be interpreted to exclude such configurability, I chose to interpret the four colours and fixed rounds as minimum requirements).

I started with just a basic UI for the game and title screen to ensure I got the requirements completed and refined first, before going back and making them fancier later. I only needed to make one minor change to the actual game code later after noticing edge cases where the same colour would be selected several times in a row, making it seem as though the selection code wasn't functioning.

All the code is my own beside the wave script used to animate the background; I considered making it myself, but decided it wasn't important enough to spend excessive effort on when it's simply aesthetic. I've implemented similar code before, but not in Unity or C#. The screen wipe and loading screen scripts and assets are reused from some of my previous projects, with slight modifications.

For the scoring, I went for a simple overall time and an average per question. Upon testing, the average seems to not really get above three seconds in my testing, unless you intentionally leave it for awhile, so it mightn't have been the best choice.

Also, I've had issue regarding the editor viewport moving the camera if the dimensions aren't set to 1920x1080, which I wasn't able to solve by fiddling with the camera settings. I've changed the build settings to default to that resolution as well, and that seems to have worked fine. I'm unsure how to fix this, and I'm not sure if it'll remember my editor settings when you open it, so if you don't see the title and wavy background, that's why.