Documentation for Permutation Game

By Justin Carpenter

Design process:

Given the limitations imposed – only 1 week to complete the game, and the player must finish the game in under 4 minutes – I decided I wouldn’t be able to craft a full set of “game” mechanics the user can play while still teaching them about permutations in under 4 minutes, so I decided to opt for a guided learning experience instead. The goal was to show visual examples of permutations as the user answers questions so they can see how permutation works; each time they answer the number of total permutations, there will also be those permutations shown on screen visualized as colored blocks.

The previous learning experiences I made tended to have too much going on on-screen at any one time. For this one I tried to limit the amount of text shown on screen at one time, but I had to strike a balance where the user wasn’t also left clueless when asked to answer a question. I chose to keep certain elements, such as the basic explanation of permutations and the question being asked, on-screen at all times.

Code:

I decided to use Unity for this project even though WPF would have been simpler; I just wanted more experience with Unity. Given the small scope of this project, I used some pretty inefficient practices like having a single game object be the “manager” for all scripting, and have all other interactable game objects communicate directly with it. For larger projects I’ll have to find more efficient ways to manage how the various objects communicate so that the project doesn’t become too unwieldy.

Testing:

The most valuable input I got from having others test the game was rewording the explanations and questions. I originally wrote them assuming too much initial knowledge, such as assuming the user understands factorials. The amount of explanation provided now should be suitable for the target audience of college students aged 19-24.