Here is a write up of things you may not see while you play the game. But a designer might see when he is setting up the game for the first time or adding new things.

Game Rules are programmed in the GameBoard.cs and PegSlotData.cs

Player Interactions with the game board are in PegSlot.cs with PegSlotData.cs as middle man.

BoardPresets

I used a ScriptableObject here as the intent was to list all the presets on the game setup screen for the player to choose from. Ran out of time to get that added, so only the game designer can set the preset by dropping a Board-Preset from Assets/PresetData into the MainMenu_Controller script located on the MainMenuCanvas object in the scene. Create new presets by right clicking on a folder and hit Create, Create Board Preset.

BoardPresets hold color styles for the board. I made two color style classes: PegColorStyle_RowByRow; and PegColorStyle_SingleColor. The board passes themselves into one of these styles held in a preset at runtime. The row by row style will color each separate row a different color. Order based on the order of the colors in the pre-set color list, cycles around when colors are all used. The single color just colors all pegs the same color.

BoardPresets Editor allows the designer to setup the options for each preset. They choose a style and setup that style. The chosen style is the one that gets used in the game. Styles that are not used are not shown in the editor. To view, change, or delete a preset, click the preset asset file in Assets/PresetData.

Problems Encountered

I created the board array data structure on paper first. The 2D array is sideways, but the game object in the world is not. I failed to see this difference and struggled with peg slot spacing for an extra hour. I was trying to draw 1 column at a time (at angle like the triangle) instead of drawing a row at a time. Eventually I got it working, but this was one of few problems I ran into.

Class shearing in my BoardPreset scriptable object was my next issue. Even though the main reason I was using a scriptable object was because it is suppose to handle class shearing when you stored derived classes in a base class list. Wasted a bunch of time on this bug, so I ended up just hardcoding the preset to have a copy of one of each style and storing an enum that keeps track of which one is in use, instead of an elegant dynamic solution. This creates a maintenance problem whenever a new style is programmed.

My biggest time consuming bug I encountered was (probably due to lack of sleep) a bug in the camera pan-ning when you grab the screen and drag. I was losing half the distance dragged because of it. It still worked but it just felt wrong, I wanted it so when you grabbed the screen no matter where you drag, your initial mouse placement in the world was always in the same world position when done dragging. This was a 2 lines of code bug fix that took around 4 to 5 hours! Yikes.

That's all. I had fun.

Reid

PS. I forgot to link to the commits page on github. I used trello and source control like I would in a team set-ting. So here is a link to that page.

https://github.com/rmalford2007/CrackerBarrel_TechnicalTest_Alford/commits/master