DigitalTechnologies™

Unity test project

Overview

Create a simulation of the famous Cracker Barrel wooden peg game, with a way to replay moves and design and use triangular board shapes. See here for general rules and discussion.

Requirements

In rough priority order:

1. Game should run both in Editor and preferably on Android phone build.
2. Menu to select difficulty level -- easy = 4 pegs each triangle side, normal = 5 pegs, hard = 6 pegs
3. On mouseover of peg (in Editor) or single tap (on phone), valid pegs (i.e. those you can actually move) should be highlighted somehow as an indication to user.
4. On click (in Editor) or double tap (on phone), animate the peg to move out of the board into the space above the board -- like an invisible hand is picking it up, ideally above and over to the side so it's not blocking the board.
5. Now, on mouseover (in Editor) or single tap (on phone), only the empty holes that are valid for you to play should be highlighted somehow as an indication to user.
6. On click (in Editor) or double tap (on phone), the previously selected peg should animate into new peg and the jumped peg should disappear.
7. When there are no more legal moves, it's game over and there should be either win or lose indication.
8. Have a real-time clock counting down with 3 minutes -- game is also over when time runs out.
9. **Bonus requirement**: After the game, save the sequence of moves played to a save file. Next time you run, you can choose to play new game, or choose to replay the previously saved game (loaded from the save file). In replay mode now you have forward and backward buttons that let you step through the game played previously step by step. In replay, the clock should show the actual time left at the time of the moves recorded.

Guidelines

- The requirements give you a general outline of the functionality desired. Use your own game design judgement on the UI and gameplay and art style (including audio and visual feedback).
- For art and sound assets, be resourceful to get whatever looks good to you.
- **Correct and intuitive functionality** is more important than a beautiful art style.
- Readability matters
- Good code architecture matters
- Good scene organization matters
- Do not worry about performance.
- Prefer hooking things up in Inspector rather than using Find or GetComponent. Don't use SendMessage.

Code style

1. Be consistent in your code style
2. Use clear naming -- no unnecessary abbreviations

Process

- Create a git repo on bitbucket or github to store your work, make commits regularly as is typical good git discipline.
- After the test, we should be able to follow your progress throughout the two days by looking at the commit history. Do NOT commit everything at once at the end.
- At the end of the test we should be able to clone your repo, load the scene in Unity v5.5, and hit play in Editor, and also build to my Android phone (or iPhone) and play.
- The best is to complete the full assignment, including the bonus, but if you can't, then use your judgement about what to implement and what to leave out so you still have something to show.