RockCreek

Skills Assessment

Your challenge: build a 2-player Connect 4 browser-based game using Javascript (as well as any variations of HTML and CSS you like).

Summary: Two players should each be able to hit a URL. Each player should be prompted for their name as well as the game name. If the game name for two players match, they are playing each other. Each player should be able to click or otherwise interact with a visual grid and see their pieces fall into place–as well as the pieces their opponent plays. Anyone who lays four of their pieces in a row–horizontally, vertically, or diagonally–wins. Both players are notified, and the game ends.

Functional Requirements:

- Runs normally under Chrome or Safari on Macs and PCs.
- Support both 1- and 2-player mode.
- Asks for player's name and uses it sensibly during play.
- Connect 4 board and pieces are represented visually (but we don't count off if you're not a designer)
- 1-player mode has computer play pieces randomly (but bonus points if you come up with something less ignorant for the computer)
- 2-player mode requires both players enter the same "room" or "game" name at startup.
- In 2-player mode, each player is given a different color piece.
- Any player's move appears reasonably immediately on both screens.
- Winning is 4 pieces in a row horizontally, vertically, or diagonally.
- Illegal moves are not allowed (playing outside the board, trying to play twice in a row, etc.).
- A "You win!" or "Your opponent won!" or similar message appears on the correct screens at game close.

Technical Requirements:

- You can use all the HTML, CSS, imagery, and toolkits you like for the UI.
- All interaction must be written in JavaScript, although you can use any flavor or framework you like. By the way, we personally love React and Vue.js, but the choice is ultimately up to you!
- You can handle two-player interactions using any JavaScript toolkit you like. We tend to be fans of serverless components, but the key is to build a solid demo for us, so again, this choice is up to you.

- Don’t stress out over hosting. A whitelisted IP address is fine, as is a fully-qualified domain running on AWS. Just make sure we can test it out with two players simultaneously.
- Assume a laptop, not a phone or tablet. If you want to get ambitious and build a responsive app, you’ll probably get some bonus points, but it’s very much not expected or required.

A Few Tips:

- If you’re not sure about something, ask. We like interaction! No, we won’t point you to a GitHub with React code for the game, but we will clarify, make sure you know what we’re looking for, and generally help you in any way we can. We *want* you to rock this!
- Know *why* you made the choices you did. We’re *at least* as interested in your reasoning as we are with your actual finished code. If you can’t explain why you did something, that will not go over that well.
- Use the Internet. Seriously. We Google everything, and so should you. No, you should not find someone else’s finished game and rip it off shamelessly. But yes, you should reuse code, use frameworks and toolkits instead of re-inventing wheels, and generally take advantage of what’s out there… not rely wholly on your memory.
- This isn’t a designer role, but we’ll admit it: we *love* a good UI. And even stock images can be used in ways that feel right. Take some time and put it into the interactions of the app, and it will almost certainly pay off.