**Design Paradigm, Software Architecture, and Design Patterns**

**Identify the design paradigm you (i.e., the team) chose for your prototype and explain why you chose that paradigm.**

**Within the context of your chosen design paradigm, describe the software architecture of your prototype.**

Within the context of our chosen design paradigm, object-oriented design, the software architecture of

our prototype will be N-tier. For our design paradigm, object-oriented design was chosen because we will organize the methods for conducting the gameplay based on both players and the rules of connect 4. For our software architecture, N-tier was chosen because there will be 3 layers: a presentation layer, a game logic layer where this may be subdivided into player interactions and rules. The presentation layer will contain information about front-end user interaction such as user click-ability as well as user visual and audio feedback. The player interactions layer will contain information about the possible disk placements a user can make. And from each placement, the player interactions layer triggers an event for a visual/audio cue to mark the end of player 1’s turn and the start of player 2’s turn. The rules layer will outline when our game should end. At the beginning of the game, we can create an option for a random player to start first. As the game is taking place between player 1 and player 2, once a connect 4 has been made the rules layer will trigger an event to mark the end of the game and declare who the winner and loser of that round of connect 4 is. However, if there is a tie between two players, the rules layer will also indicate this and make sure that this information is displayed to both players of the round (and give both users the option to play the game once again).

**Identify the design patterns you used in your design and explain how you applied them.**

We will use a mediator, which is a behavioral design pattern, because our Connect.Js file will encapsulate all of the objects and functionality of the program. We will also use the User Interface Design Pattern because our project will be done using HTML and Javascript. This will allow the user to interact with the game in a web browser. The game should be intuitive to use for the user, and the interface should look professional and aesthetically pleasing. This allows for a web design pattern to also be used in our game. We will use the typical colors used in a connect four game, such as a blue grid and red and yellow chips. This will make our online version of the game more similar to the physical version of the game. We will also be building a prototype, which is a type of creational design pattern. Our prototype is a Use-Case Diagram because it shows all of the inputs and outputs involved for both players during the game. There are multiple functions that will be used by both of the players in the program, so we believe a Use-Case Diagram is the best option. We will also be using an iterator, which is a behavioral design pattern. We will have an updateGrid method that will be constantly updating the grid whenever a player clicks on a button to change its color to either red or yellow. It will also check if a player has won to end the game.

**Prototype- Use Case Diagram**

**图示

描述已自动生成**