Architectural Style:

To design this chess game, I think the best architectural style would be to use the Model View Control (MVC) style. With this architectural style, the model would be able to manage the storage of the data of the game, the view would be able to display the information (the game) to the user, and the control would aid in the management and the control of the information between the view and the model. These three tiers focus a lot on the user interaction which is essential to playing the chess game.

The model of the game would showcase the specific data on how the chess game operates. This includes the location of the pieces in the game (for example, information on the starting position and documentation of the game to keep track of the current position), the rules of the game, and more. The view part would show the state of the game (location of the pieces on the chessboard) to the players and a list of the moves to be made (and previous moves). The controller would link the view and the model. The controller would also collect the responses from the user's mouse, keyboard, etc. and trigger a reaction by having model objects interact based on the response. An advantage to this style is since a game user interface (GUI) is being used, a change to the view does not need a change to the model. It also makes it easier to break down, implement, and test each part.

UML Class Diagram:

This architectural style for the chess game would follow somewhat of a state machine diagram. The model and controller would control the game, the states (the location of pieces on the chessboard), the transitions of the states and other events. Specifically, the controller would

use the model and the controller would update the view.