

Architectural Pattern for Betrayal Online   
By Nobody

Team 3: Caleb Corlett, Ryan Nodarse, Patrick Storer, Ethan Wyman

## Description:

The Architectural Pattern we are using for Betrayal Online is a Layered Architecture. It is made up of a User interacting with a Game UI Layer, which will trigger game logic in the Game Logic Layer which can, in response, modify the output from the Game UI Layer. The Game Logic Layer can also receive information from other Users connected to the game via LAN to influence the Game UI Layer’s output to the original User. And, if required, the Game Logic Layer can also place requests to store and retrieve data inside the Local Database for game logic tasks.

## Why Layered Architecture for Betrayal Online (Justification):

The functionality of our program in regards to User input as well as interactions with the database would be well represented by the separated layers of a Layered Architectural pattern. The User does not need to interact directly with the data stored in the Local Database or the Game Logic Layer, so the Game UI Layer would be a helpful middle-man to trigger game logic in the Game Logic Layer which would handle calls to the Local Database, so that the User doesn’t have to interact with anything besides the Game UI Layer for the game to do its thing. Working in a unidirectional way that is often associated with the Layered Architectural pattern.