Patterns Document

Design patterns used:

1. Observer:
    - When bid was made or when a robot captured the target piece. Then, scoreboard and players information had to be updated.
    - When a robot was moved on a board then we had to notify the 'game logic' that was storing/checking the current state of the game to check if the move made was legal so that it can change the state of the board.
    - Timer gave user 1 min to make bid. When timer reached its time. The user then had to stop making bid and play his/her move.
2. Command:
    - Control button like 'New Game' was used to set game settings, 'Exit Button' to quit game, 'Start Game' to start game according to user settings, etc.
3. Adapter:
    - We had to connect different methods in board class. Like when we were creating board design, different methods took parameter related to simple board design type. Hence, we created few methods as adapter to fulfill the need for that method so that we can use it for complex board.

Design pattern we might would have used if we had more time:

1. Command: 'save game' and 'load game' button to store/load the current state of game, 'Get hint' button to help user make his/her move.
2. Factory method: to implement AI which provides an interface for creating objects in a superclass and allows subclasses like easy AI and hard AI to alter the type.