# W200 Project 1 - Gomoku Design Document

**Description**
Implement a game of Gomuku, i.e. Five in a Row.
The game can be played between humans, or human and computer. Humans can even design computer players to play against each other.

**Objects**
Class: Stone
Attributes: color
Methods: __repr__

Class: Board
Attributes: size, dict where keys are tuples (row, col) and values are Stone objects
Methods: __repr__

Class: Player
Comments: This is a generic parent class, and children classes are human players and different computer players
Attributes: name, stone
Methods: play (pass)

Class: Player_H
Attributes: inherit
Methods: play - prompt user to play, evaluate the move

Class: Player_C
Attributes: inherit
Methods: play – decide the move using a sequence of Strategy classes. block opponent > extend its own (make 5 > make 4 w/ liberty > make 2 3s w/ liberty > extend)

Class: Strategy
Attributes: name, description
Method: consider – takes current board and stone as position, return list of viable moves according to the strategy.

Class: Strategy_X(Strategy)
Attributes: inherit
Methods: code each strategy

Class: Gomoku
Attributes: turn (which player's turn), status (on/end), move_sequence, move_time
Methods: game, eval_move, eval_win, eval_end, game_stats

Gomoku.game()
1. Ask: player 1, player 2; type (human or computer); names; stones

# W200 Project 1 - Gomoku Design Document

2. Board()
3. While(status=='on'): players alternate plays & eval_move, then eval_win and eval_end
4. Announce winner & game_stats

**User Interaction**

User just needs to run Gomoku().game() to play the game.
User will be prompted for game information (players & type, names & stone color).
User will play the game by inputting row and column numbers to place the stone.
Game finishes and announce winner & game stats.