Team Cookies	self-evaluation score	comments
working game (5)	5	Servers handle clients and peers battle peers.
network architecture (5)	5	The matchmaker runs on the server which clients can connect to and create, view, and join games. Games in progress are set to a P2P connection on top of two matchmade clients.
no crash in play (5)	5	Our game has been tested for edge cases throroughly.
register (5)	5	The only option available for an unregistered user. The game will register the player's terminal name together with a randomly generated port number and their computer's hostname.
list games (5)	5	Players can view the server's database of peers waiting for opponents.
create game (5)	5	Players can create games and wait opponent for an opponents to join in. Once an opponent joins, the game will start.
join game (5)	5	Lists games, and then allows the player to type in an opponent's username to enter into a game.
exit game (5)	5	In a game, pressing "ctrl-c" will return the player to the menu. In the menu, selecting "Exit Game" or pressing "ctrl-c" will return the player to the terminal.
unregister (5)	5	The Player can unregister the game and go to the register page.
application specific protocol (20)	20	We have a server that listens to the clients request on the different options the client can select such as register, list games, create games, join game, exit game, and unregister. To maintain client vs client (i.e. player vs player), we use P2P application or Peer.cpp. We also have a class that generates the the battleship board and handles coordinate pair attacks and lets the players know whether they hit or miss their opponent.
bonus feature - chat (3)	0	Did not implement.
scoreboard (3)	0	Did not implement.

:he clients, and unique protocols are enforc

protocol allows each peer to enter one coc

in-play game video (3)	**0**	**Did not implement.**
document (10)	**10**	**Created documentation that documents the overview of the game, how to compile the game, instructions for game, and detailed description about how our methods and protocols were implemented.**
Total (75 + 6 = 81)	**75**	

Team Cookies
working game (5)
network architecture (5)
no crash in play (5)
register (5)
list games (5)
create game (5)
join game (5)
exit game (5)
unregister (5)
application specific protocol (20)
bonus feature - chat (3)
scoreboard (3)

ced with error checks for each option.

ordinate pair to send to the other. They may only enter one pair at a time.

in-play game video (3)
document (10)
Total (75 + 6 = 81)