

Done	To-Do
<ul style="list-style-type: none"> - Make it executable everywhere DONE! <p>#Board</p> <ul style="list-style-type: none"> - input of team names DONE! - initial positions can be added NO NEED! - Get enough information a about pygame and start building it ABORTED! - flip baord method and fix working DONE! - To string methods for all the classes DONE! - Complete validation of moves DONE! <p>#Main</p> <ul style="list-style-type: none"> - Initiate DONE! -- Connect GUI also DONE! - Connect with Board and use methods DONE! <p>#AI</p> <ul style="list-style-type: none"> - toStrng no need! - Plan out minimax of chess board SOME! - returns names and teams done! 	<ul style="list-style-type: none"> - Just use board direct updation - Start Controller method of input and output <p>#Main</p> <ul style="list-style-type: none"> - Then act this as a tunnel between Board, AI and GUI - start finishing player to player <p>#AI</p> <ul style="list-style-type: none"> - Just use board direct updation - Complete the non working minimax - Design minimiax

Time-line:

Task	October	November	December
<u>Main</u> -(10)	-	10%	-
<u>Game</u> -(20)	20%	-	-
<u>GUI</u> -(10)	2%	10%	-
<u>AI</u> -(60)	5%	50%	60%

Notes:

- Don't just think about losing pieces, Think about losing specific pieces. Give priorities or preferences.
 - Think about other factor like positions of the pieces
 - Define your winning state and /or close to winning state
-
- Check and go to recurse for Alpha and beta.
 - Whole Board would be passed as parameter in Ai class
 - we might need to construct the gui inside board which would be moving methods because it is just printing nothing

