

# Gameloop.py

GameLoop.py									
Requirements									
				i+m<12 j+n<12	i+m<11 j+n=12	i+m=12 j+n<11	i+m=12 j+n=11	i+m=11 j+n=12	
Inputs			Empty (Initial State)	iRP/jBP/m RK/nBK	iRP/jBP/ mRK/nBK	iRP/jBP/ mRK/nBK	iRP/jBP/ mRK/nBK	iRP/jBP/ mRK/nBK	Updates
Win Board	Place Red Piece	White	error1						Board -tile matrix -piece matrix
		Black(x)	-	error1	error1	error1	error1	error1	
		Black(o)	1RP	(i+1)RP jBP mRK nBK	(i+1)RP jBP mRK nBK	error2	error2	(i+1)RP jBP mRK nBK & Start	
	place Blue Piece	White	error1	error1	error1	error1	error1	error1	
		Black(x)	-	error1	error1	error1	error1	error1	
		Black(o)	1BP	iRP/ (j+1)BP mRK nBK	error2	iRP/ (j+1)BP mRK nBK	iRP/ (j+1)BP mRK nBK & Start	error2	
	Place Red King	White	error1	error1	error1	error1	error1	error1	
		Black(x)	-	error1	error1	error1	error1	error1	
		Black(o)	1RK	iRP jBP (m+1)RK nBK	iRP jBP (m+1)RK nBK	error2	error2	iRP jBP (m+1)RK nBK & Start	
	Place Blue King	White	error	error1	error1	error1	error1	error1	
		Black(x)	-	error1	error1	error1	error1	error1	
		Black(o)	1BK	iRP jBP mRK (n+1)BK	error2	iRP jBP mRK (n+1)BK	iRP jBP mRK (n+1)BK & Start	error2	
ForceStart			Start	Start	Start	Start	Start	Start	
Reset			-	empty	empty	empty	empty	empty	

Legend

\*Table for assignment 3 below

White	Place this on the white tile
Black(x)	Place this on the occupied black tile
Black(o)	Place this on the unoccupied black tile
error1	Show message "Can't place here" + current state
error2	Show message "Out of pieces" + current state
<p>"board", then updates "Board" with tile matrix and piece matrix. <u>P.Redpiece</u> means clicking the button of 'Place Red piece, and <u>P.BluePiece</u>, <u>P.RedKing</u> and <u>P.BlueKing</u> mean clicking each button in order. Then, <u>'ForceStart'</u> and <u>'Reset'</u> mean clicking Force Start button, and Reset button respectively. Also, each variable 'i', 'j', 'm', 'n' is the number of each RP(number of Red piece on the board), BP(Blue piece), RK(Red King), and BK(Blue King). White(or Black) means placing the piece(or king) on the 'White'(or Black) tile. The meaning of Black(x), Black(o), error1, and error2 is explained on the right side of table. As player choose <u>P.RedPiece-Black(o)</u> + Empty(Initial State), player clicks <u>'Place RedPiece'button</u> and click on unoccupied tile of empty board. Then there is one Red piece on the board and it is written as 1RP in the table. In any state, pressing the 'force start' button starts the game with current state, and</p>	

\* The table is only a screenshot as we had trouble exporting it.

The table above illustrates the requirements that must be achieved by each module in terms of a simple interface, note that variables in this chart are not actual variables in the program and instead variables that are simply names used to represent certain functions or states.

Draw.py									
Requirements									
Inputs									Updates
win	As player clicks any button of 'place RED piece', 'place Blue piece', 'place Red king' and 'place Blue King', then the text of the button change to yellow, otherwise it remains as white.								text
b1									
b2									
b3									
b4									

The table above provides a general description of what the main function of the draw module.

Note only these interface contains trace backs to requirements as other as they are the only one with huge amount of logic . Other modules simply create additional changes to the interface.

**Classes** module simply defines behaviour to important objects and will represent the objects in the checkers game

**Screen** creates the window that the user will use

**Main** simply links everything together

---

The table below shows the requirements of module after the game has started.

The table is to be read from left to right. First, the player supposed to drag one of the Red piece, Blue piece, Red king, and Blue king. Then the player choose which tile to drag, for example 'lower white tile on the same row' from the first line of the table below. Depends on whether the tile is empty, empty & bottom/top, or occupied, the game shows different options. Any white tile has error that the player can't drag his/her piece to the tile. Also, any piece or king only can move diagonally to the closest tiles. Red pieces can't be moved to same column or higher column tile and Blue pieces can't be moved to same column or lower column. However Kings can go either lower or higher unless it's occupied or white. For one special case, as the Red piece arrives to empty black bottom tile, it becomes a Red king, also Blue piece becomes Blue king when it's at empty black top tile. The other case is when there is a blue piece where red piece tries to get(which is diagonal for sure), the red piece can jump the blue piece so it arrives to the next diagonal black tile and the blue piece is removed(it is stated as "Good move" on the table). However when there is same red piece, it can't jump with error4( "same team"). It's stated on the right side of the table "between the piece and the tile where the player dragged to, there is a red/blue piece/king. 'error1' means the player can't move the piece/king since it's not diagonal. 'error2' means the player can't move the piece/king it's white tile. 'error3' means the player can't move the piece/king since there's no one to kill.

# Gameloop.py

		tile					Between the piece and the tile where the player dragged to, there is a					
inputs		column		empty	empty and the bottom	occupied	red pice/king	blue piece/king	Updates			
win board	Drag Red Piece to	lower	white	on the same row	error1		nothing	-				
				else	error2							
			black	1 diagonal	move	Red King		error4	Good move			
				2 diagonal	error3	Red King						
				on the same row	error1							
				else	nothing							
		same or higher	white	nothing					-			
			black									
	Drag Red King to	lower	white	on the same row	error1		nothing	-				
				else	error2							
			black	1 diagonal	move			error4	Good move			
				2 diagonal	error3							
				on the same row	error1							
				else	nothing							
			same	white	nothing					-		
				black								
		higher	white	on the same row	error1		nothing	error4		Good move		
				else	error2							
			black	1 diagonal	move							
				2 diagonal	error3							
				on the same row	error1							
				else	nothing							
				tile					Between the piece and the tile where the player dragged to, there is a		Board -tile matrix -piece matrix	
		column		empty	empty and the top	the tile is occupied	red pice	blue piece				
Drag Blue Piece to	lower or same	white	nothing					-				
		black										
	higher	white	on the same row	error1		nothing	good move		error4			
			else	error2								
		black	1 diagonal	move	Blue King							
			2 diagonal	error3	Blue King							
			on the same row	error1								
			else	nothing								
Drag Blue King to	lower	white	on the same row	error1		nothing	-					
			else	error2								
		black	1 diagonal	move			good move	error4				
			2 diagonal	error3								
			on the same row	error1								
			else	nothing								
		same	white	nothing					-			
			black									
	higher	white	on the same row	error1		nothing	good move		error4			
			else	error2								
		black	1 diagonal	move								
			2 diagonal	error3								
			on the same row	error1								
			else	nothing								

Board  
-tile  
matrix  
-piece  
matrix

# Gameloop.py

				tile			Between the piece and the tile where the player dragged to, there is a					
inputs		column		empty	empty and the bottom	occupied	Red Piece	Blue Piece (Only one blue piece/king left on the board)	Blue Piece (There are more than one blue left piece/king on the board)	Updates		
Initial state/ Red piece/king just moved	Blue piece/king just moved	Drag Red Piece to	lower/same/higher	white/black	any tile		nothing					
			lower	white	on the same row	error1		nothing	-			
					else	error2						
					1 diagonal	move	Red King					
					2 diagonal	error3	Red King					
					on the same row	error1						
			black	on the same row	error1		nothing	-				
				else	nothing							
				nothing								
			same or higher	white/black	nothing							
			There is at least one Red King on the board									
Red piece/king just moved	Blue piece/king just moved	Drag Red King to	lower/same/higher	white/black	any tile		nothing					
			lower	white	on same row	error1		nothing	-			
					else	error2						
					1 diagonal	move						
					2 diagonal	error3						
					on the same row	error1						
			black	on the same row	error1		nothing	-				
				else	nothing							
				nothing								
			same	white/black	nothing							
			higher	white	on same row	error1		nothing	-			
else	error2											
1 diagonal	move											
2 diagonal	error3											
on the same row	error1											
black	on the same row	error1		nothing	-							
else	nothing											
There is at least one Red King on the board												

w in board				tile			Between the piece and the tile where the player dragged to, there is a			Board -tile matrix -piece matrix					
		column		empty	empty and the top	the tile is occupied	Red piece (Only one Red piece/king left on the board)	Red Piece (There are more than one Red piece/king left on the board)	Blue Piece						
	Blue piece/king just moved	Drag Blue Piece to	lower/same/higher	white/black	any tile		nothing								
			lower or same	white	nothing		nothing	-							
					black	nothing									
					higher	black						on the same row	error1		
												else	error2		
												1 diagonal	move	Blue King	
			2 diagonal	error3	Blue King										
			on the same row	error1											
			else	nothing											
There is at least one Blue King on the board															
Red piece/king just moved			Drag Blue King to	lower/same/higher	white/black	any tile		nothing							
	lower	white		on the same row	error1		nothing	-							
				else	error2										
				1 diagonal	move										
				2 diagonal	error3										
				on the same row	error1										
	black	on the same row		error1		nothing	-								
		else		nothing											
		nothing													
	same	white/black		nothing											
	higher	white		on the same row	error1		nothing	-							
else			error2												
1 diagonal			move												
2 diagonal			error3												
on the same row			error1												
black	on the same row	error1		nothing	-										
else	nothing														
There is at least one Blue King on the board															

Board  
- tile matrix  
- piece matrix

The table above is pretty much same as the table from assignment 2 which describes the movement of pieces on the board. However this one also shows 1)only Blue piece can start from the initial state (the state than none of the piece made a movement), 2) when every Blue piece/King removed, Red wins, and 3) when every Red piece/King removed, then Blue wins. If it is PVP mode then whole table is describing the game, or if it is Blue(or Red) vs computer, part of the colour which player is playing is describing the player's movement.