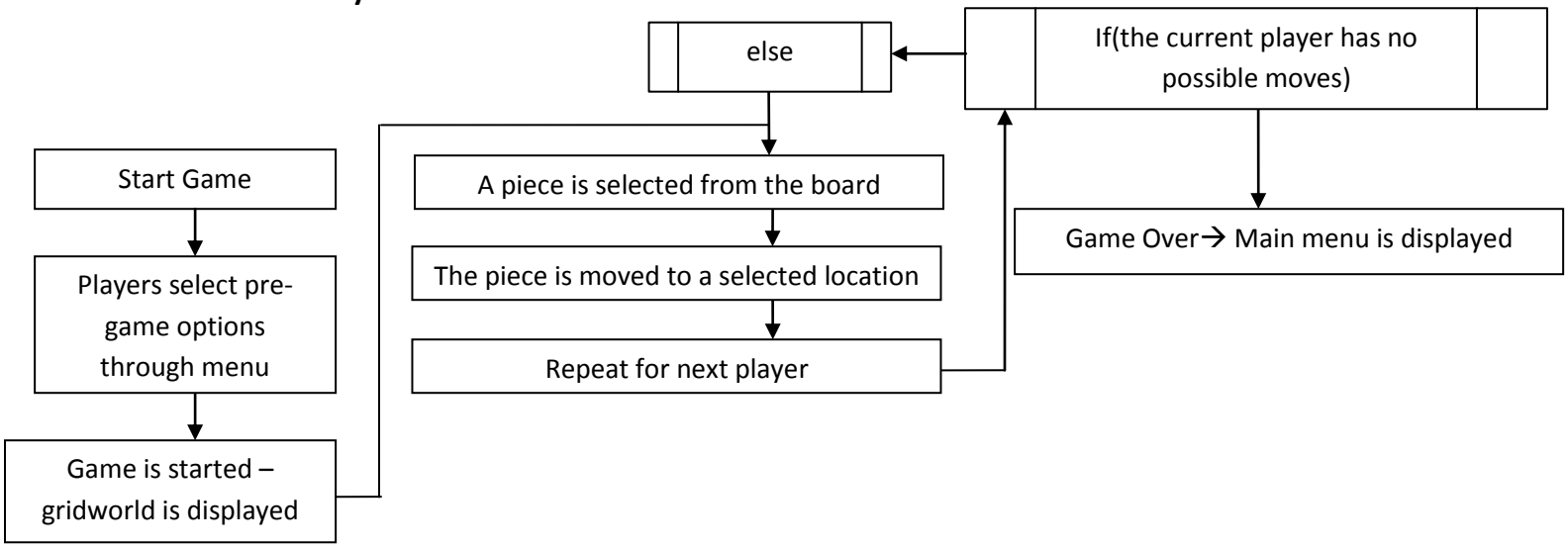
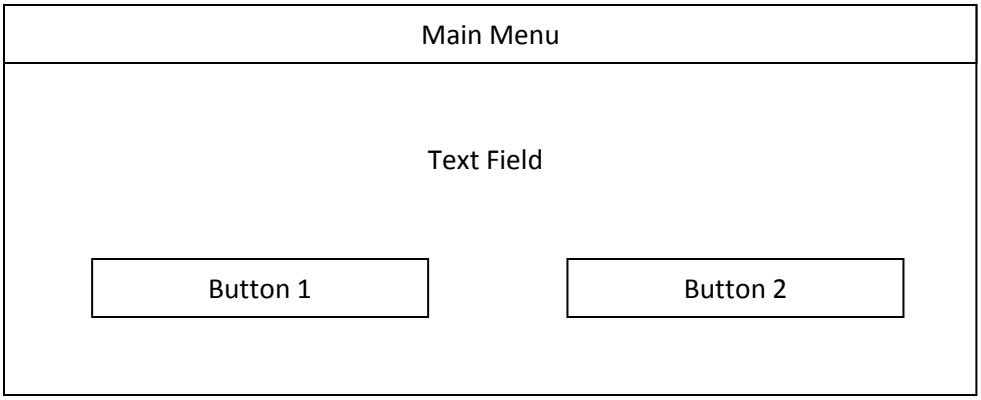
Criterion B - Design

There will be about 6 phases to the program:

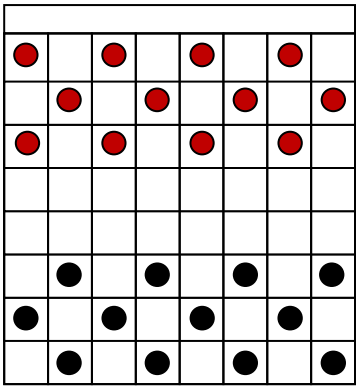
1. Selecting the pregame options
2. Selecting a piece
3. Selecting a location to move it
4. Checking if the move is legal
5. Executing/cancelling the move

Basic Game Layout





A menu will be displayed when the program is run with options that must be filled out before the game starts.



Once all the pre-game options have been selected, a GridWorld window with pieces will appear which will be used to play the game.

Global Variables:

Name(s)	Type	Description
isValid	Boolean	Whether the move made was legal or not.
TakeAgain	Boolean	Whether another piece can be taken with the same piece.
Black	Array of Actors*	Player 2's pieces.
White	Array of Actors*	Player 1's pieces.
Piece	Actor*	Piece currently selected.
Loccount	Location*	There are 2 phases to each move, clicking a piece (loccount=0) and then a location to move it (loccount=1).
count	int	The number of moves made – starts out as 0.
C1, c2	Color	The colors of players 1 and 2, respectively.
Name1, name2	String	The names of the colors; used because when c1 and c2 are converted to strings, their r-g-b values, rather than their names, are displayed.
Take	Boolean	Whether the move selected was a capturing move.
hasClicked	Boolean	Whether the piece has actually clicked a piece.

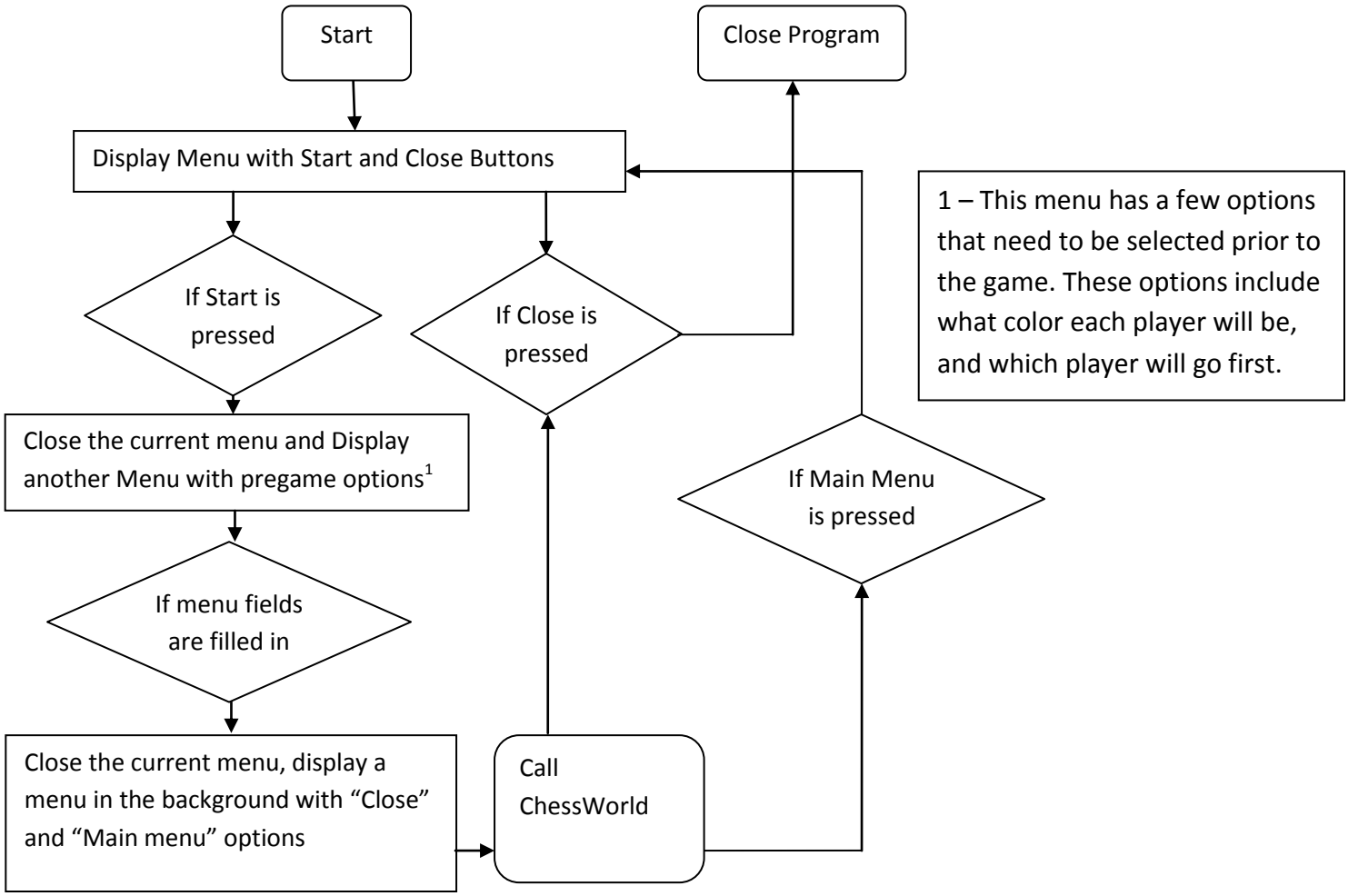
* - A GridWorld Data type

Methods:

Name	Type	Description
isValidinCheckers(Actor Piece, Location start, Location loc, String Type, String color)	Boolean	Determines whether the move being made is legal or not.
canMove(Actor piece,String color)	Boolean	Determines whether that piece has any possible moves.
canMoveAtAll()	Boolean	Determines whether any of the current player's pieces have any possible moves.
canTake(Color color)	Boolean	Determines if the player of the given color

		has any possible capturing moves.
canTake(Actor piece)	Boolean	Determines if that piece has any possible capturing moves.
getTaken(Actor Piece, Location start, Location loc, String Type, String color)	Location	Returns the location of the piece being captured.
keyPressed(String ans, Location loc)	Boolean	Is a given GridWorld method that was extended. Detects when a key is pressed and either confirms or cancel a move that was selected. Also carries out post-move tasks, like removing a captured piece, etc.
LocationClicked(Location loc)	Boolean	Detects locations and uses them to select the piece to be moved and the location to move it to.
movePiece(Actor piece, Location start, Location loc)	Void	Links LocationClicked to isValidinCheckers, and checks a few conditions. Then asks the user to confirm or cancel their move.

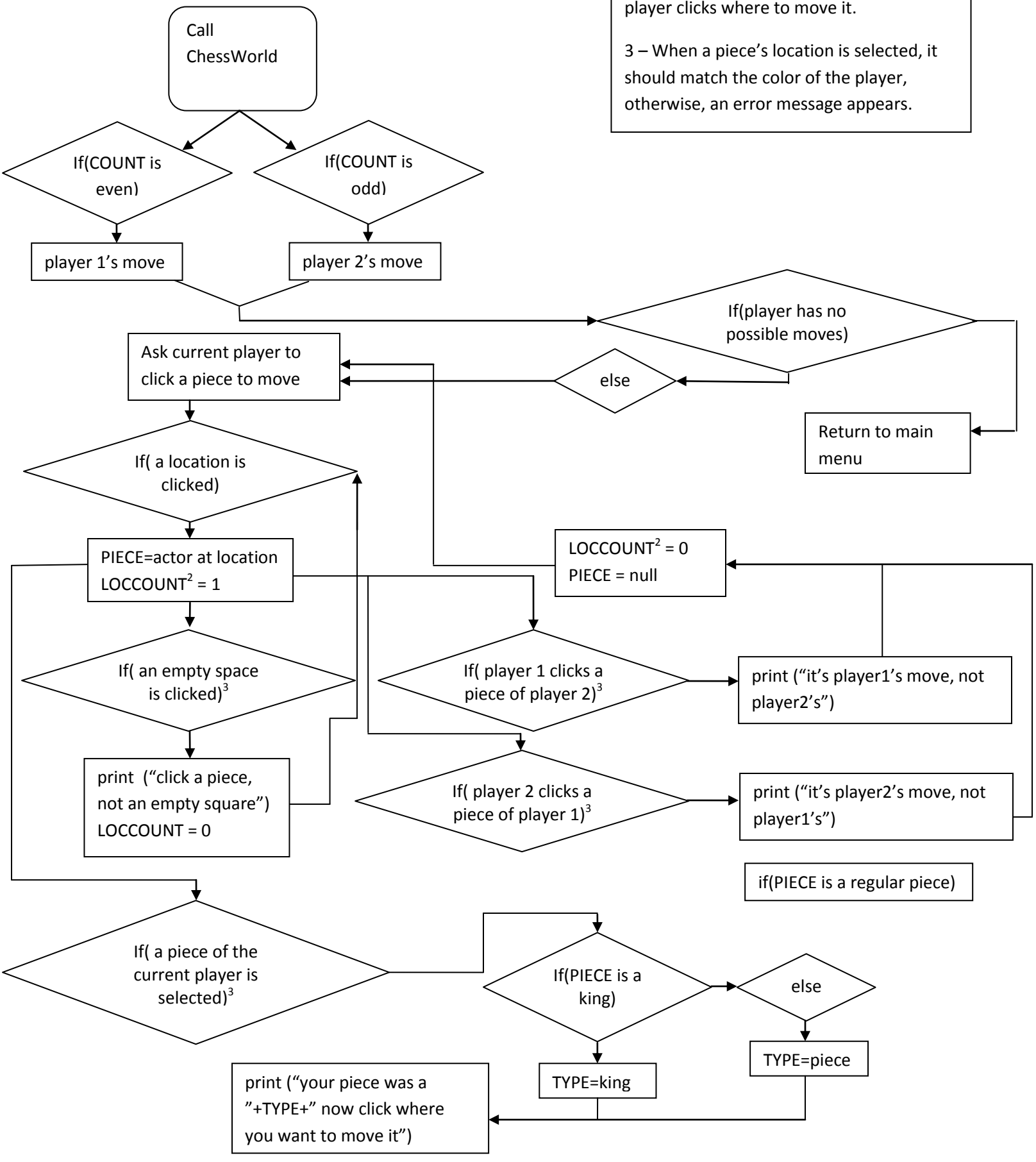
Menu Layout



Phase 2 – Selecting pieces

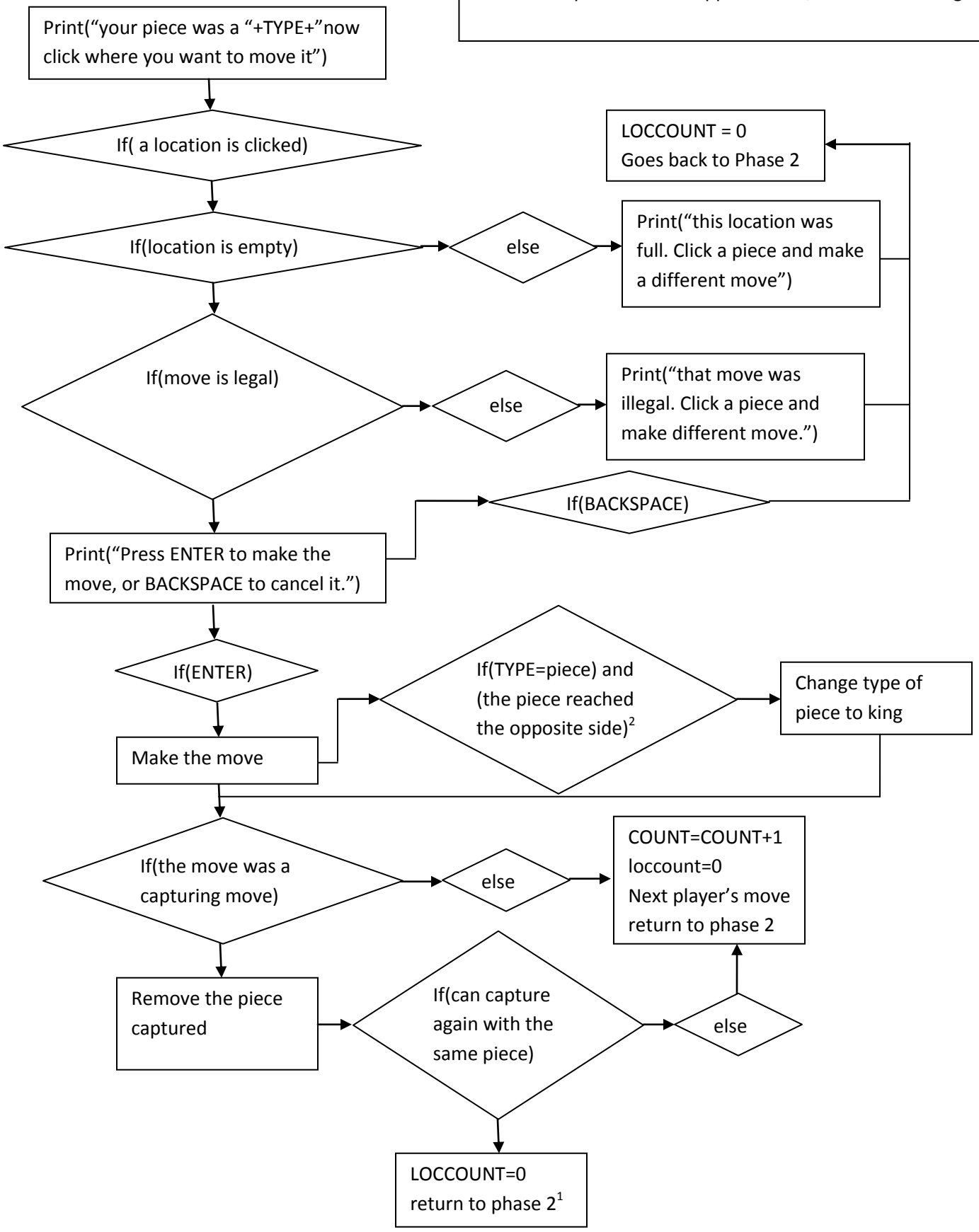
2- When LOCCOUNT is 0, the player clicks a piece to move. When LOCCOUNT is 1, the player clicks where to move it.

3 – When a piece's location is selected, it should match the color of the player, otherwise, an error message appears.



Phase 3 – Moving pieces

1- if a player captures one piece and can capture another with the same piece as before, the players turn continues, so COUNT stays constant.
2 – When a piece reaches opposite side, it becomes a king.



Phase 4 – Checking the Legality of Moves

```
TYPE = piece's type(king/piece)
If(TYPE=piece)
    If(the move is backwards)
        Return false
        End if
    If(the move is(1 to the right or left) and forward)
        Return true
        End if
    If(the move is (2 to the right or left) and forward)
        TAKE=the piece taken
        End if
    if(TAKE is a piece)
        return true
        end if
    return false
end if
If(TYPE=king)
    If(the move is(1 to the right or left) and (1 forward or backwards))
        Return true
        End if
    If(the move is (2 to the right or left) and (2 forward or backwards))
        TAKE=the piece taken
        End if
    if(TAKE is not null)
        return true
        end if
    return false
end if
```

Test Plan

Component Tested	Method(s) of Testing
Clicking an empty space when picking a piece.	Click an empty space when asked to click a piece(N).
If a piece can be captured, it must be captured.	When a piece can be captured, make a non-capturing move(N).
Regular pieces cannot move backwards.	Move a regular piece 1 step diagonally backwards(N), and also 2 steps diagonally back(N).
Multiple captures can be made in one turn.	Make multiple captures using the same piece in one turn(Y).
When a player has no valid moves, a congratulatory message will appear.	Play through an entire game and reach a win. Check for a congratulatory message(Y). Continue the game even after reaching a win(N).
Pieces can only capture other pieces, not kings.	Capture a king using a regular piece(N).
Capturing moves go 2 spaces forward or backwards and 2 spaces right or left.	Capture a piece by clicking on it(N). Move 2 spaces diagonally when there is no piece in between(N).
Kings can move forwards and backwards.	Move a king in all 4 directions, for capturing and non-capturing moves(Y).
Pieces of one's own color cannot be captured.	Capture a piece of one's own color(N). Capture a piece of the opposite color(Y).
ENTER confirms a move only when a valid move has been selected.	Press enter when an invalid move was selected(N). Press enter when no move was selected (N). Press enter when a valid move is selected(Y).
Backspace cancels a move that has been selected, and returns to the piece-selection-phase	Press backspace when a piece has been selected(Y), and when a move has been selected(Y).

(Y) – Returns **no** error message

(N) –Returns an error message