

# Task sheet 2 comp 1 2015 pre release

## Task 11

Function	Explanation	Parameters
DisplayMenu()	This prints the menu of options for the user	No parameters or return
GetMenuSelection()	This asks the user what menu option they want to do	No parameters, return = menu choice
MakeSelection()	Carries out the option entered by the user	Menu choice is passed in, No return
PlayGame()	Plays the game as usual if the option entered is 'PlayGame' or an option that plays a game	Menu choice, No return

5. For PlayGame() function, I have taken the InitialiseBoard function apart by taking out the sample game part because it was ill placed. This was put into its own temporary function for the time being. Also took the main program from 'StartSquare = 0' or just above and refactored it into the PlayGame function. Also took out two lines of code about the sample game because they are not needed in the PlayGame() function. I also had to change the MakeSelection() function so that the options entered have a possible program to work through by passing into each whether a sample game was wanted.

## Task 12

-The functions that need to be altered are GetMove() and PlayGame(). These two need to be changed because the rogue value needs to be entered into the GetMove() and then the options functions need to be called later on. A new function would have to be made to print the options menu, as well as making the selection and carrying out the task.

## Task 13

- The functions that will need modification are: OptionsMenu(), OptionsSelection() and MakeOptionSelection() - because a new option needs to be able to be carried out. Also PlayGame() would need to be altered because all of the options functions are called and their response to surrendering will need to be changed. Changed the playGame() function so that it returns once the value of 4 has been entered to surrender, this takes you out of the current game and into the main menu.

## Task 14

- Refactoring is altering the current program by re-organising it in order to change its logical flow and understandability.

Changed the three functions so that InitialiseBoard() calls the other two functions and initialises the correct board that has been asked for.

#### Task 15

Role	Explanation	Example
Fixed Value	A value which will not be changed. This number is set from the start.	BOARDDIMENSIONS
Most Wanted Holder	The most appropriate value in a series of values.	PieceType
Stepper	Used to keep track of something	Count
Most Recent Holder	Holds the latest value encountered	StartSquare
Gatherer	Gathers the sum of all values and keeps a running total	None in this program
Transformation	The result of a calculation of other values	StartFile
Follower	A variable that is updated each time the old value is changed from another data item.	None in this program
Temporary	The variable that only holds a value for a short time	WhoseTurn

#### Task 16

-Made sure that the abs of FinishFile - StartFile = 1 and the FinishRank - StartRank = 1 because it is checking for the immediately adjacent squares diagonally.

#### Task 17

- Made sure abs of FinishFile - StartFile == FinishRank - StartRank because you are checking all of the diagonal spaces from the original that can be moved to.

#### Task 18

- Made it so the legal moves of Etlul allow the abs of FinishFile-StartFile to change by 2 **and** the abs of the FinishRank-StartRank to change by 1. OR the abs of FinishFile-StartFile to change by 1 **and** the abs of the FinishRank-StartRank to change by 2.

#### Task 19

- Changed the Redum legal moves so that if its starting rank is = 7(White) or = 2(Black) and

the FinishRank == to the StartRank + or - 2 (Depending on black or white) then it is able to move two spaces forward on the first go only due to the original start rank being one of the two values stated. Also I have left in the check for one space so that after the first move the Redum can only move one space forward from then on.

#### Task 20

Passing by reference: - A parameter is being passed in where it is stored so any changes to the parameter are accessible to the function that has called it.

Passing by value: - A copy is made of the variable to the parameter so any changes have no effect on the original value.

Function	Parameter Pass
CreateBoard	Neither
DisplayWhoseTurnItIs	Value
DisplayMenu	Neither
GetMenuSelection	Neither
MakeSelection	Value
PlayGame	Value
OptionsMenu	Neither
OptionsSelection	Neither
MakeOptionSelection	Value
GetTypeOfGame	Neither
DisplayWinner	Value
CheckIfGameWillBeWon	Reference and Value
DisplayBoard	Reference
CheckRedumMovelsLegal	Reference and Value
CheckSarrumMoveLegal	Reference and Value
CheckGisgigirMovelsLegal	Reference and value
CheckNabuMovelsLegal	Reference and alue

Function	Parameter Pass
CheckMarzazPaniMovelsLegal	Reference and Value
CheckEtluMovelsLegal	Reference and Value
CheckMovelsLegal	Reference and Value
InitialiseBoard	Reference and Value
InitialiseNewBoard	Reference
InitialiseSampleBoard	Reference
GetMove	Value
ConfirmMove	Value
GetPieceName	Value
MakeMove	Reference and Value

Written with [StackEdit](#).