SOFTENG 701: Assignment 5

Ahmad Umar, auma020
The University of Auckland
auma020@aucklanduni.ac.nz

I am using the files submitted for Assignment 4.

I. Plan

I plan on introducing a new method to automate a Player's move. I need to include two variables in *Kalah* class: *robotMode* to store whether computer mode is enabled or not and *automatedPlayer* to check if player 1 or 2 needs to be automated. I need new a method, *checkBestMove*, to calculate the automated Player's next move. The *robotMode* variable will be used in the method *gamelogic* written in class *Kalah*. In my current code whenever it is Player 2's turn, I need to add an if statement to check if robot mode is enabled; if enabled then we call the method *checkBestMove* to get Player 2's input.

The minimal changes I think are needed to implement the change case are summarized below:

1) Find the class *Kalah* and create a static boolean variable *robotMode*; value of true = enabled and false = disabled. The value is initialised at the very beginning of the *Kalah* class.
2) create a new method in *Kalah* class named *public int checkBestMove(int[] array,int automatedPlayer)*. This will take in the *boardArray* which has the scores stored and takes in the player number that is automated; for this assignment automatedPlayer =2.
3) *checkBestMove* will interate through player 2's pits in the *boardArray* i.e indexes 8 to 13. It will first check for a move that leads to an extra move, else a capture, else the lowest legal move. The method sets the *playerInput* to the index which leads to one of the options. The method will return an integer; the possibilities are, 1=extra move, 2=a capture or 3=legal move (1 with the highest priority).
4) Go to the method *gamelogic* in class *Kalah*. For my code, each time I am taking input and printing *string2*, that means it is Player 2's turn. Thus, before each input statement I need to add an *if* statement to check if *robotMode* is enabled.
5) If robotMode is enabled, then call the *checkBestMove* method.

II. Implementation

Start time: 10.30am ; End Time: 12.15pm

I followed out all the steps as stated in my plan. I felt the steps were well detailed and I faced no difficulty. However, I forgot to mention in my plan that I want to add comments. Comments makes it easier when the developer is looking at his code again and makes it easier to understand for another person looking at the code. In my plan, I missed out details for the print statements for each of the cases for *robotMode*; we can either have an extra move, or a capture or a legal move. For each scenario different statements with calculated Player 2 input need to be printed. These statements were copied from the Assignment brief.

III. Changeability

I believe my code included changeability well; nothing had to be removed from my previous submission. To incorporate the new computer mode feature, I was easily able to add logic for the new feature and it fit in nicely with my old code. The effort I put in was consistent with what I had expected when planning. Initially I thought this will be very hard but once I planned it out, it seemed easier and I believe planning it beforehand allowed swift implementation.

For the method checkBestMove, I have used a variable *automatedPlayer* to allow me to easily include automation for Player 1 easily, if need be..

The impact of the change case is about 0.2. All of previous code is still there, with some additions, making it completely recognisable as a variation of Kalah.