Asignment 3

Zkon584

Ziyin Kong

I. OBJECT ORIENTED

object oriented typically means program will run time create objects and objects are communicating by sending messages to each other. To be an object oriented program, first, objects must be clearly defined. In the case of Assignments 3, kalah game is divided into two components: gameController and IOhandler which are created during run time.

IOhandler is in charge of dealing with input and output but nothing else. As for gameController, it is functioning to control the game flow and give orders to other objects that how they should function. for example, before player starts their turn, gameController will message IOhandler required data and IOhandler prints out the game board. gameController then requires to obtain input from player via telling IOhandler to print out the prompt for user and get the input and pass it back to gameController.

Instances held by classes all have the same behaviors. For instance, every game has its own independent gameController and IOhandler. gameControllers all have the same behaviors such as player capturing seeds from opponent's house, but different features that contains distinguishing data and game flows.

There are other features that contribute kalah to be an object oriented program such that objects are well encapsulated. One gameController is created during run time, the game starts and its accessibility is limited from outside the object. No other object can affect or alter its behavior or features referring to game data.

Kalah utilizes the concept of polymorphism which it considered an extraordinarily significant feature for object oriented program. gameController takes interface IO when constructing so that if game is alter that behaviors of how to take input is reshaped. Passing a different concrete class implements interface IO could cater for its changing. A different IO class similar to mock which is passed to gameController in kalah could meets the requirement without changing the behavior gameController.

II. GOOD DESIGN

Before discussing whether kalah has good object oriented design, how to define what good object oriented design means should be clarified. Appropriate selection of objects contributes to good design and different behaviors are performed separately within the objects. Different objects meet their own obligation rather than cross over with others. Furthermore, A good design is only good because it fit right into the context that it could be a bad design in context schema.

As mentioned in the last section, kalah is separated into two objects: gameController and IOhandler. Two classes have distinguishing obligations that gameController contains game rules and game flow that control how game is operating and IOhandler is only for printing out the messages for players and taking input from players. Two modules are fairly independent which means every module perform its own obligation with negligible cross over. It contributes to high cohesion and low coupling.

Two classes all have distinct behaviors clearly separated into methods. For example taking input and printing out message are two behaviors corresponding to methods printPlayboard and requesInput. As for gameController, actions for the game such as capture, sow seeds are separated into methods rather than one method contains the flow of entire game. It increases modifiability of the game since if game rules are changed only the corresponding method requires to be modified not the entire class.

Uses of interface IO also increase reusability, if the behavior of how input and output handled in kalah is altered then a different IO class could solve such changes which increases reusability.