Project Documentation

* Classes
  + OnePlayerGameModel, TwoPlayerGameModel are classes that are not subclasses or abstract classes.
* Subclasses
  + GameController, TwoPlayerGameController, and StartPageController all extend Switchable making them subclasses.
* Abstract Class
  + My program uses a public abstract class called Switchable. This abstract class was written in class for our scene switching example.
* Interface
  + My program uses a public interface called GameInterface. I wanted my one player and two player game models to have some of the same functions but also wanted the ability to modify implementation between the two models so I created an interface to ensure that my models had similar functionality but could be different when needed.
* Collection Classes
  + My program uses a Collection classes, specifically a HashMap and Stack. The HashMap is located in Switchable and is created on line 23. It is used to store the names of controllers to allow my program to switch scenes. I used a stack to further demonstrate by ability to use Collection classes since the Switchable code was largely written in class. I used two stacks in my TwoPlayerGameModel on lines 40 and 41 to hold the x and y coordinates of each move made. The user can select undo from the options menu to pop the latest move off of the stack and undo their last move. Moves can be undone until the stack is empty, and the stacks are popped on lines 377 and 378 of the TwoPlayerGameModel and the board and ellipses are adjusted accordingly.
* Exception Handling
  + My program uses exception Handling in a few locations:
    - Game Controller- there is a try catch statement on lines 324-340 that catch exceptions if the user is trying to resume a game and there is an error. There is another try catch on lines 354-369 catching an exception if the user tries to save their game to a file and it cannot be done.
    - TwoPlayerGameController- There are two try catch statements in this class also used to catch exceptions when the user trues to resume a game (lines 299-315) or save a game(lines 329-344).
* MVC/Clearly defined Model
  + Models: OnePlayerGameModel, TwoPlayerGameModel
    - These models hold the 2d array board and data that keep track of which moves have been made and whether or not there is a winner. These models let the controller know which elements of the view need to be updated.
  + Views: StartPage.fxml, Game.fxml, TwoPlayerGame.fxml
    - These are the views which are updated by the controllers.
  + Controllers: GameController, StartPageController, TwoPlayerGameController
    - These controllers update the views based on the data from the models
* Multiple Scenes
  + The three scenes I used are Game.fxml, StartPage.fxml and TwoPlayerGame.fxml
  + The abstract class switchable was used to switch scenes.
    - When the user initially selects one or two players from the StartPage the scene switches to the appropriate scene on lines 40 and 47 of the StartPageController.
    - If the user is in the Game.fxml scene and they select menu from the Options section of the menu the scene switches back to StartPage.fxml on line 137 of the GameController.
    - If the user is in the TwoPlayerGame.fxml scene and they select menu from the Options section of the menu the scene switches back to StartPage.fxml on line 127 of the TwoPlayerGameController.
* “About information”
  + There is about information available in the menuBar of all three FXML scenes. If the user selects “About” they will see an alert box that displays information about me. If the user selects “Rules” they will see an option to view the one player game rules or the two player game rules.
  + Game Controller code for:
    - One player directions- line 82
    - Two player directions- line 96
    - About-line 110
  + StartPageController code for:
    - One player directions- line 57
    - Two player directions- line 70
    - About-line 83
  + TwoPlayerGameController code for:
    - One player directions- line 73
    - Two player directions-line 87
    - About-line 101
* Save and Load data
  + My program allows the user to save a current game to a file or resume a game by opening a previously saved file. This code was largely taken from the serialization example in class. There are two locations in my code where this occurs.
    - GameController
      * Loading data begins on line 313
      * Saving Data begins on line 346
    - TwoPlayerGameController
      * Loading data begins on line 288
      * Saving Data begins on line 321