**Class**

Architecture: Model View Controller

**PlayerBean: model(the process behind setting up the data)**

**fxml: view(what user sees)**

**controller: interface between view and model**

**Main:** The stage/screen that you see when you first open the game and this is shown in the start method. The stop method is mainly for the high scores to be saved in the application even after exiting and formats it based off of three categories:Ranking, Name and Score. The main method is where the application is launched as demonstrated by the Launch(args) command in it. The FXML portion of it is controlled by Main.fxml file and it references the MainController class as well. On the front stage, you can see the title, a funny gif of letters bouncing around and the option to start game, show the high scores, how to play and also to quit. Pressing on quit will result in the interface closing completely. Pressing on How To Play redirects you to the How To Play stage (HowToPlay.fxml and the HowToPlayController class). Pressing on High Scores redirects you to the High Scores stage (HighScore.fxml and the HighScoreController class). Pressing on start game redirects you the actual game page (Game.fxml and the GameController class).

**MainController:** the controller that interacts with the Main.fxml file(view) and the data. The setupStages method is the most important method in the application in my opinion because this is where the logic is built for the transition and wireframing.

**GameController:** This class is meant for the actual game and sets up a list of images that recur and pop up on screen for user to answer. It also handles the quit game capability and go back capability. This class includes the method for setting up a player and the high score stage. It also has the list of all the images which, in the most efficient way possible that I know of, shuffles and randomizes the order of how they are displayed each time game is played. It also has the method for resetting the point counter at the top of the game stage. This class also includes the doEnter method which is another very important method where the high scores are sorted and points are added up.

**HighScoreController:** This class is meant for setting up the stage/screen for the High Scores. It utilizes a tableview to set up table and has go back capability. Very basic class that has setters for the stages and also utilizes javafx specific code for the tableview.

**HowToPlayController:** This class is meant for setting up the stage/screen for How To Play. It also has go back capability.

**Player:** This class sets up the profile for the player to later be entered into high scores if the player does achieve a high enough score. It sets up profile using encapsulation of the three variables ranking, username and score.

**HighScore.csv:** This file stores all of the high scores, ranking and usernames and helps in maintaining these scores even if application is exited.