Name: Christian Pierre Paul & Roy Avrett

\*Any details regarding instructions for the game that are not obvious from the set of standard known instructions (include any rules variations used) & A description of how to use the interface

-When starting the game, click “File”, then click “New Game” pop the screen, next click the “Play” then “Move” button below the board to start playing. Once the “Play” and “Move” button (in that order) is clicked, use the arrow keys to move the tiles in the direction desired (Up key to move all tiles up, Down key to move all tiles down, etc. ). Once you win or lose, navigate to “File”, then “View High Scores” to check the list of high scores. The “about” tab has the rules listed under the “2048 Rules” tab and information about Roy and I in the “Development Team” tab. Once there are no more available moves or the user has a block on the board that is of the value 2048, the game is automatically over and the user has the option to restart by clicking the play button.

\*Any other details about your implementation that you feel are important

-The GameIFrame.java file contains all of the GUI of the game itself, this includes the JPanels and JButtons to play the game such as “Move” (JButton) and “Board” (JPanel). It also contains the implementation to set up the board and moves for the arrow key that is pressed.

-The RulesIFrame.java file and TeamIFrame.java are both JInternalFrames that contain information about the rules of the game and information about Roy and I.

-The HSIFrame.java file contains the code utilized to grab the top ten highscores and sends it to the “highscores.txt” which will continuously be updated after each game (only if the user gets a new highscore). After finishing a game, close the previous file by clicking “X” on the upper left corner and reopen a new one so that the file updates.

-The AppWindow.java file contains the code the puts every class and GUI together in the main JFrame.

\*Descriptions of any extra features implemented

-We added a file containing the top 10 highscores which is updated after every game.

-We replaced all actionlistener events for buttons with keyboard listener events for the move button which allows the user to move the board with arrow keys.

For partners: include a description of the separation of the work (who was responsible for what pieces of the program)

-Roy:

Created the original code for all the GUI (AppWindow.java, GameIFrame.java, HSIFrame.java, RulesIFrame.java, TeamIFrame.java). This was a critical portion of the assignment as it gave the program its beautiful aesthetic. Created the code for updating the board after a move.

-Chris:

Created the original code for initializing and setting up the game board which is seen in GameIFrame.java. Contributed to making the code related to moves (I struggled on this portion but Roy assisted me heavily in finishing this part). Updated GameIFrame.java, AppWindow.java, and HSIFrame.java to utilize elements of each other such as calculating and maintaining the top 10 scores. Also added extra implementation for KeyListener allowing us to use arrow keys for a ove.