Period: 3
Group Members: Benjamin Nudelman
Group Name: Team 4096
Project Title: Project 4096

Project Description: This project aims to transform the original 2048 game by adding more modes and a start and end screen to it. The original game begins with one square and the ability to move in any direction, and with each move all the squares move in one direction, and a new square is created in the grid. The objective would be to combine equal squares as much as possible to achieve one square with the highest number. There is a score to keep track of progress in the game, as well as a timer for the user.

Functionalities:

- **Start screen**: The music starts playing and the user is given two possible options (buttons) to choose from.
- One, **the settings icon,** brings the user to a separate screen in which the general **rules** of the game are displayed, and there is a **"go back" button** that brings the user back to the start screen, a **"reset"** button that resets the colorway to the original 2048 colorway for the squares, and a **"pink"** and **"blue"** palette that each utilize their respective color palettes to create the colorways for the squares.
- **"Start" button:** Once it is pressed the user is brought to the game screen and their **timer** starts. The **best score** is also displayed, if the user plays the game multiple times in one run of the code, and the **score** constantly updates, starting at zero.
- The user can use all **four arrow keys** to move all of the Squares on the grid in a specific direction
- After each click a **"postMove" method** is executed which both adds either a 2 or 4 to an empty grid box (if there is one) and **checks for a loss** by going through each Square and looking if it is possible to combine it with any of its "neighbors".
- If there eventually is a loss, the **loss time** and score are kept track of, the best score will update if it is beat, and the user is brought to the **loss screen**, where this information is displayed. In the loss screen, the user has the option to click a **"try again" button** that brings the user back to the start screen, resets all of the variables, and starts up the music again.

Libraries:

- java.util: I am using this for my ArrayList variable in the game
- processing.sound.*: This is used for my music in the start screen.

How does it work: The actual game works with the objective of combining equal-valued squares to achieve the maximum valued square without filling out your grid into an un-combinable position. What this means is that every move requires an arrow key press, which moves every Square in that specific direction. If a row has more than one square in that column/row, then all the squares are moved in that direction in the same order. If the row/column is full, nothing is moved. Since a new 2 or 4 randomly is placed on the grid after every move, one must attempt

to constantly combine squares to avoid filling up the grid. A square is combinable with another square if it is, without squares in between (there can be empty spaces), next to another equal-valued square in the specific direction of the key press. Once the grid is full and no possible moves remain, the game is lost. Your objective can either be to get the highest number, get the highest score, or get the largest time spent playing the game, though usually the objective is to attain the greatest value for a Square in the grid.