Alex Klavens

This program allows the user to manipulate a simulated macroeconomy. The program provides buttons that adjust the levels of certain economic variables and displays changes in numerical and graphical form.

Major components:

The program consists of 4 major components:

The first is a class called MacroClass. This class provides the mathmateical basis for the macroeconomy, giving the user starting values and providing calculations.

The second is two classes that compose the aggregate demand graph. The first class is called Demand and sets up a line and a clone of that line to represent the initial and manipulated states of aggregate demand, respectively. This class takes the parameters of a GraphWin, as well as the MPC of the economy, a determiner of the line’s slope. This class also provides the basis for various movements of the manipulated line. This will be employed by the class ADGraph and also takes the GraphWin and MPC parameters. This class provides an X and Y axises, as well as a 45˚ line to symbolize the points in the economy where demand is equal to supply. This class acts on commands for movements and also establishes markers on the axis as to where the lines intersect with the 45˚ line.

The third component is a class called ADASGraph which reflects the relationship between aggregate demand and aggregate supply in the original and manipulated economies. This takes the parameters of a GraphWin as well as one called “s” which allowed me to use similar coordinates from the ADGraph class to make this graph but position it in a different location. The ’s’ parameter is used to change graphical X variables. Similarly to the ADGraph class, the ADASGraph class provides methods to move, clone, and place labeling based off of intersection points for certain lines.

The fourth component is a class called EcoButton. This variation of the Button class will later be used to adjust various economic variable levels. It provides the user with buttons to increase or decrease the value of a given variable. It also displays the current value of that variable.

The last major component is a class called MacroGUI. MacroGUI takes no parameters and is the operational GUI that the user will interact with. It provides with adjustor buttons for various economic variables. It also displays an instance of the ADGraph and the ADASGraph classes. This class takes user input via mouse clicks and calls on the various classes to graphically and mathematically adjust their manipulated macroeconomy. This class also displays the current GDP level in the economy and provides an explanation as to what any abbreviated variables displayed represent. The main method within this class, called ‘update’, is based on a quit-button while loop. Within the while loop, the program checks for mouse clicks and adjusts mathematical variables and calls another method (called \_updateGraph) to make graphical changes.

This program has been rigorously tested, though still contains what some might call a bug. At this point, changes in a given variable result in certain graphical changes that must be represented by a significant enough shift to be noticeable. Unfortunately, because the graph is relatively small, a user can quickly force lines to move off the graphs. However, I chose not to make lines stop moving once something hits the bottom or end of a graph because a movement to the end of a graph doesn’t necessarily mean that GDP is at zero. For example, if investment is decreased from its starting value by 4 clicks, the demand line will hit zero. This is because i wanted to make changes in these seem significant. However, even as it hits zero, the real GDP is actually still quite high. In a future, modified version of this program, I would like to figure out away to get around this issue. Additionally, I would enhance the economic model to include a third graph that represents the state of money in the economy. With this graph and additional buttons, the user would be able to adjust interest rates (by adjusting variables that cause changes in the interest rate) and see the subsequent visual changes. Additionally, a future version would more accurately account for price level changes while allowing the user to set a full employment level and give the user the option of seeing a demonstration of how the economy would self correct, or allowing the user to see if he or she can use monetary or fiscal policy to return the economy to full employment. Ideally, this program could be enhanced enough that it would give students of macroeconomics a user-friendly and accurate interactive representation of a macroeconomy.

Original GUI idea: