Use Case Name: Run Simulation (as it will appear in final implementation)

Use Case Description: This use case runs and displays a simulation of simple interacting objects on an open field

Actors: User, GUI, and MeleeSim

Triggers: This use case begins executing as soon as the program begins running.

Preconditions: The program is running, and the GUI is displaying the simulation window.

Postconditions: The simulation runs until the user resets, pauses, or closes the program.

Normal Flow:

1. The MeleeSim class randomly places the shapes in the window according to stored default values.
2. The user clicks the “Start” button in the GUI window.
3. The MeleeSim begins running, stepping through states in the simulation in real time.
4. In each step, the simulation generates projectiles from existing shapes, determines the movement each shape and projectile will make in the next step, and resolves collision detection between projectiles and shapes. Shapes struck by a projectile will have an individual number decremented, and any shape with a number lowered to 0 or below is removed from the simulation.
5. The GUI is sent to position and orientation of each object and renders them to the simulation window.
6. The user can click a button, “Pause”, to pause the game. This halts the stepping process of the simulation and refreshes the window with the last step before the pause.
7. After pausing, the user can unpause the game. This resumes the stepping process.
8. The user can click a button, “Reset”, which returns the program flow to Step 1, placing shapes in a different random placement.
9. The user can click a button, “Close”, to close the simulation. This tells the program to stop the simulation, run class destructors for all objects, close the GUI window, and exit the program.

Use Case Name: Edit Shapes (to be implemented)

Use Case Description: Allows the user to alter the numerical values that control the shapes' behavior in the running simulation. A list of text boxes that allow keyboard input are displayed next to an image of the shape being edited. One button on screen allows changes to be saved and the program returns to the simulation mode.

Actors: User, GUI, and MeleeSim

Triggers: The user clicks a button labeled "Edit Shapes" in the simulation window.

Preconditions: The program is running, and the GUI is displaying the simulation window.

Postconditions: The program saves the numerical values to the MeleeSim's list of objects and the program returns to the simulation window.

Normal Flow:

1. The user clicks on the "Edit Shapes" button.
2. The MeleeSim is paused.
3. The GUI switches to displaying the Edit Shapes window.
4. The user makes changes to numbers in the text boxes.
5. The program changes any invalid input to a default value and notifies the user.
6. The user may switch to other shapes and edit their values, going back to step 4.
7. The user clicks the Save Changes button.
8. The GUI calls a MeleeSim function that changes the variable values in each class of SimObjects.
9. The program switches back to the simulation window.
10. The MeleeSim is unpaused.