# AniMate: An Animation Toolkit for Java Swing

Khelan Modi (kmodi@gatech.edu), Tulika Banerjee(tbanerjee33@gatech.edu)

The final deliverable of our project is a java swing package that contains the function definitions for the animations mentioned below. External users will be able to import this package to apply the animation functions from our toolkit to their own swing applications. In developing the package, we referred to Swing Hacks by Joshua Marinacci and Chris Adamson and the official Oracle Java Documentation for features 2, 5.2, 8, 12 and 13.

In order to successfully run the application, the development environment should be set to JDK 17.0.5. No additional 3rd party libraries are required to run the toolkit.

We discuss the specific animation functions that we implemented in our toolkit along with the principles of animation they correspond to.

1. **Highlight on hover**

Class: Highlight.java
Function: The selected component grows in size when the mouse pointer hovers over it.
Parameters passed to the function:
1) JButton button - The button to which the animation will be applied
2) Dimension size - The dimensions of the button
3) int numIterationsLimit - The no. of iterations it takes for the animation to complete
4) int sec - The duration the timer runs for
5) MouseEvent e - The event that calls Highlight i.e. Mouse_Entered or Mouse_Exited
Components: Selectable components- JButton, JRadioButton, JPanel, etc.
Principles: Solid Drawing, Anticipation

2. **Animated Sheet Dialogs**

Class: AniSheetableJFrame.java
Function: A dialog or modal rolls down from the title bar and rolls back up.
Components: JDialog
Principles: Solid Drawing, Motion Blur

3. **Smooth Menu motion**

Class: SlidingMenu.java
Function: A Menu appears in a continuous motion from a start point and returns to the same point on close.
Parameters passed to the function:
1) JComponent component - The menu component to which the animation will be applied
2) String type - Specifies whether the animation is Horizontal or Vertical
3) String direction - Specifies the direction in which the component needs to move (Folding or Expanding)
4) int numIterationsLimit - The no. of iterations it takes for the animation to complete
5) int sec - The duration that the timer runs for
Components: JComponent, JMenu
Principles: Solid Drawing, Motion Blur, Arcs

4. **Cyclic Rotation of components**

Class: CyclicRotation.java
Function: When multiple instances of the same component are to be displayed in a limited space, it allows the user to rotate through the components. The components slide left or right to make space for the next component.
Parameters passed to the function:
1) Arraylist<JComponent> components - An array containing all the components that the cyclic rotation should be applied to
2) int numPanes - The number of Panes to be displayed at a time
3) int numIterationsLimit - The no. of iterations it takes for the animation to complete
4) int sec - The duration that the timer runs for
Components: JLabel, JTextArea, JPanel, etc.
Principles: Slow in and slow out, Motion Blur

5. **Shake/Wiggle in place**

Class: Wiggle.java, DialogEarthquakeCenter.java
Function: The component wiggles/shakes along the horizontal axis to show invalid input e.g incorrect password.
Parameters passed to the function:
1) JComponent component - The component to which the animation will be applied
2) Point location - The current coordinates of the component
3) int numIterationsLimit - The no. of iterations it takes for the animation to complete
4) int sec - The duration the timer runs for
Components: Any component extending JComponent but especially relevant for JTextField, JPasswordField, JDialog, etc.
Principles: Solid Drawing

6. **Bounce in place**

Class: Bounce.java
Function: Once clicked, a component follows through on the click and ‘bounces’ back into its original position.
Parameters passed to the function:
1) JComponent component - The component to which the animation will be applied
2) Point location - The current coordinates of the component
3) int numIterationsLimit - The no. of iterations it takes for the animation to complete
4) int sec - The duration the timer runs for
Components: Any component extending JComponent
Principles: Solid Drawing, Follow Through

7. **Smooth Scaling**

Class: Scale.java
Function: Components grow and shrink in size in a smooth motion
Parameters passed to the function:
1) JComponent component - The component to which the animation will be applied
2) Dimension size - The dimensions of the component
3) Dimension new_size - The new dimensions for the component that it will be scaled to
4) int numIterationsLimit - The no. of iterations it takes for the animation to complete
5) int sec - The duration the timer runs for
Components: Any component extending JComponent
Principles: Motion Blur, Arcs

### 8. Morphing through Tabbed Transitions

Class: TabTransitions.java
Function: When transitioning between two tabs of a tabbed pane, the tabs
fade out and fade in through the use of horizontal or vertical blinds in a smooth manner
Components: Multiple components- JLabel, JPanel, JButton, JTextField, etc.
Principles: Motion Blur, Dissolves

### 9. Drag and Drop

Class: DragAndDrop.java
Function: Dragging a component prompts the destination component to appear on the screen
Parameters passed to the function:
1) JComponent component - The destination component to which the animation will be applied
2) String type - Specifies whether the animation is Horizontal or Vertical
3) String direction - Specifies the direction in which the component needs to move (Folding or Expanding)
4) int numIterationsLimit - The no. of iterations it takes for the animation to complete
5) int sec - The duration that the timer runs for
Components: Multiple components-JLabel, JPanel, etc.
Principles: Motion Blur, Slow in and Slow out, Solid Drawing

### 10. Float

Class: Float.java
Function: A component follows a smooth motion to a new specified location
Parameters passed to the function:
1) JComponent component - The component to which the animation will be applied
2) Point new_location - The new coordinates of the component that it will shift to
3) int numIterationsLimit - The no. of iterations it takes for the animation to complete
4) int sec - The duration the timer runs for
Components: Any component extending JComponent
Animation Principles: Motion Blur, Arcs

### 11. Button Gradient

Class: ButtonGradient.java
Function: When clicked, the gradient background of the button fades out and fades back in, resembling a ripple effect
Components: JButton
Animation Principles: Solid Drawing, Motion Blur, Dissolves, Follow Through

### 12. Transparent Background

Class: TransparentBackground.java
Function: A JFrame is made transparent and its background is continuously updated to maintain transparency as the frame is resized and dragged around
Components: JFrame
Animation Principles: Solid Drawing, Motion Blur

### 13. Animated JList Selection

Class: AnimatedJList.java, ListTransferHandler.java
Function: A JList item, when selected, undergoes a slow fade before being Highlighted
Components: JList
Animation Principles: Solid Drawing, Motion Blur, Dissolves

Each of these classes has a Main class for testing purposes which is not included in the final package. We have commented out the main class in each file but retained it for ease of testing for grading and future reiterations of the package.

We also created a Demo Application to integrate and test the various animations mentioned above in a coherent real-world use case. (DemoApp.java)