Melisa Cecilia - 362044 - Asif Mayilli - 362164
DIS2 Team 2

Window System (Part 2/3): Now You See Me, Now You Don't

Windows System

Add WindowManager as an object in *WindowSystem*. Pass *WindowSystem* on *WindowManager* constructor. This way *WindowSystem* can communicate with *WindowManager*, vice versa.

Do all drawings in *handlePaint*. On adding a new window, it will call *WindowManager* if exists to redraw all the elements for a window. This way, *WindowSystem* can run alone with or without *WindowManager* and still draw all the windows properly. Components drawing also simple since it just draw everything without differentiating between button or ordinary square, *WindowManager* will be the one that handles that. This add flexibility when we want to add a new component to a window. Just add it to the list, no need to change anything in *handlePaint*.

Windows Manager

Set all the colors in here and make it *final* since it should be something that comes in with windows manager, and it's not logical to give ability to change them programmatically. This makes it easier if we want to change any color value in the code. Handle all mouse events from *WindowSystem*. Detect and store selected window if any on mouse pressed. This will make it easier on checking when mouse is dragged and also provides a flag to tell other thread that there's an active window currently chosen.

Redraw window. This is to draw all the elements in a window (buttons, header, strings). All elements are of type *RectangleComponent.* The class supports flexibility by giving freedom to use it as label, rectangle, or button. All the components x&y positions are calculated based on the window's x&y position. We make title bar part of the window so its easier to draw window's border.

Move window based on x&y difference. We need to use the x&y difference parameters to move all the component objects inside a *SimpleWindow.* Since we passed those parameters on move method, we also use them to move the window position.

Expert Implementation

Windows Reordering - On clicking a window, it will automatically be on the top
Technical : On mouse pressed, detect which window is selected. Store the window object in a variable. Delete window object from the windows list, add again on the tail of the list.

Windows Border - Add a border around each windows
Technical: Draw a rectangle on x-1 and y-1 position from the original window square with same width and height.
Windows Shadow - Add a shadow under each windows to give out 2.5D feeling
Technical: Draw a gray 0.5 opacity rectangle before drawing the window. The position of the half transparent rectangle would be slightly on bigger y and bigger x than the window's rectangle.

Added Stuffs

SimpleWindow - Now supports list of components. On moving the window, automatically moves all the components inside.
Rectangle Component - Extending from Rectangle (we need the basic functions of rectangle such as x, y, widht, height, move). This class is the basic class for each components added to simple window. Can be added as a button with value, also supports string inside the square. If need only string, simply set the color to transparent.
Button Value Enumeration - To store window's button values, for now there are only CLOSE but its possible to add MINIMIZE or MAXIMIZE. Use enum to prevent adding buttons out of those 3 values in a simple window.