Assignment 5F will be optional, so this assignment is required for all students.

PIC 16, Winter 2018 – Assignment 5W

Assigned 2/7/2018. Code (a single `.py` file) due by the end of class 2/12/2018 on CCLE. Hand in a printout of this document with the self-assessment portion completed by the end of class on 2/12/2018.

In this assignment, you will explore response to signals and events. You will create a GUI with a square that can be dragged around the window. Double clicking the square will bring up a color picker dialog that allows you to change square's color. Example video: https://youtu.be/uco2uxgBs1Q

Task

Create a window 600px wide by 400px tall with a white background (not the default gray) and a 50px square, initially of any color and at any position of your choice.

Write code that enables the user to click and drag the square. That is, if the user presses the mouse within the bounds of the square, then the square is being dragged and should follow the mouse around the screen. When the user releases the mouse button, the square is no longer being dragged and should not move with the mouse. If the user presses the mouse outside the bounds of the square, the square is not being dragged and should not move with the mouse. In order to accomplish this you will have to override the methods that are called when the mouse is pressed, when the mouse is released, and when the mouse moves. You will need to figure out the logic of how to detect whether the mouse press is inside the square, and you will need to figure out how to keep track of whether the square is currently being dragged or not. You can use the same technique as the last assignment to animate the movement of the square. Note that dealing with the coordinates can be a bit tricky. The most common mistake will be that as soon as you start dragging the square, the top left corner (possibly the center, depending on your code) will jump to the point of the mouse. This is not ideal; the square should move smoothly with the mouse and should not move relative to the mouse while being dragged. Debugging this is a good exercise!

Students sometimes ignore the wording of the instructions above and try to follow the ZetCode Drag and Drop tutorial or similar to use `QDrag`. These approaches are quite different. For the sake of learning the basics of GUIs, please do not use `QDrag`. The square should just be a rectangle painted on a "canvas" QWidget (like last time), you should use regular mouse events / mouse move events, and the animation should be accomplished like last time.

Next write code that shows a color selection dialog (there is a `QWidget` for this purpose; you need to find it) when the user double clicks within the square. When the user selects a color, change the color of the square. We haven't covered color selection dialogs, but we have covered how to learn about them from the documentation!

Self-Assessment

Does your code draw a red square? 10 points

Can the user click and drag the square? 50 points

Does the square follow the mouse smoothly, without jumping relative to the cursor when clicked or when the mouse starts to move? (It's OK for it to lag a tiny bit behind the cursor; that's normal. I want you to avoid a particular bug where the square jumps as soon as it is clicked or starts being dragged.) 20 points

Does your code show the color selection dialog only when the user double-clicks in the square? 10 points

Does picking a color in the color selection dialog change the square color? 10 points

Circle the points you deserve above and indicate your total score here: