CMPT 381 – Example exam questions

NOTE: this is not a study guide – these practice questions illustrate some of the forms that the exam questions will take, but they do not cover all of the material that will be on the test.

Draw the diagram of the "interaction cycle" that specifies the architecture of most interactive systems. Label each part of the diagram.

Imagine that you want to connect two mice to a single computer so that two people can collaboratively use any application. Using the 'layered approach' to user interfaces introduced in class, describe what would be required to make this new input scheme work at each layer.

Trace a touch event through the layers of the layered model for the Android system, stating what occurs at each layer.

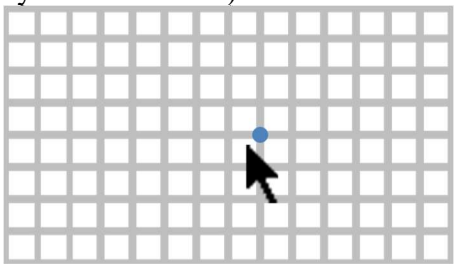
Draw the Android interface that results from the following XML specification.

State the difference between a windowing system and a window manager, and give examples of each.

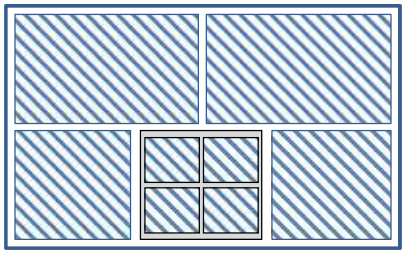
? Imagine you are building a sliding touch button that can slide to the right or to the left. Describe the steps needed to integrate this widget into an existing GUI toolkit.

If you were implementing Tomer Moscovich's "touch area contact widgets", how would hit detection differ when checking to see whether the user has touched a widget?

Write a JavaFX that demonstrates 'grid snapping' for graphical interaction. The system should show a dot at the closest grid location to the current position of the mouse. (Use a 10-pixel grid size; the grid does not need to be shown on the screen; you do not need to snap the mouse cursor to the grid, only the marker dot).

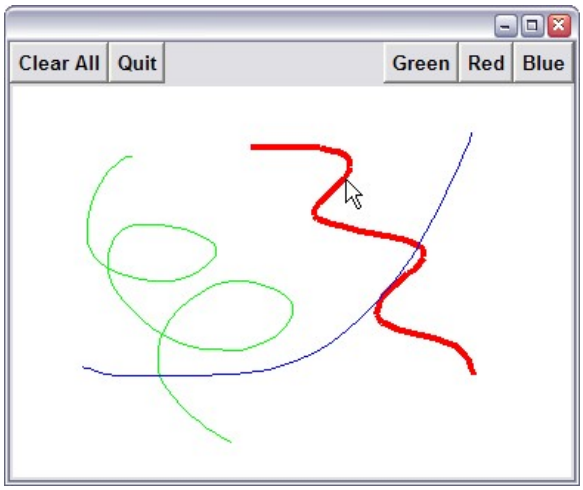


State the steps needed to create the layout shown below. Indicate all widgets and layout managers you would use. Assume that the individual shaded rectangles are buttons. Code is not required.



Define the following types of widgets and give an example of a JavaFX widget that is in the type: virtual input device, container widget, abstract model widget

Write the complete code for the interactive system pictured below. You should build the system to be as close to the picture as possible. If you cannot remember particular language elements, make up a term and write a note to say what your term refers to. Use the back of the page for more space if needed.



Specification:

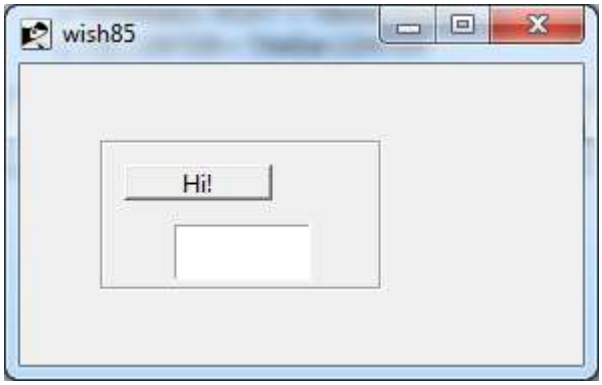
- The system is a basic whiteboard.
- When the user moves the mouse with the mouse button down, lines are drawn.
- Line colour can be changed by clicking the buttons at top right.
- The whiteboard can be erased by clicking on the 'Clear All' button
- The system exits once the 'Quit' button is pressed.
- When the mouse is over an existing line (with no buttons down), it highlights (e.g. by becoming thicker)
- You do not need to show a full MVC approach

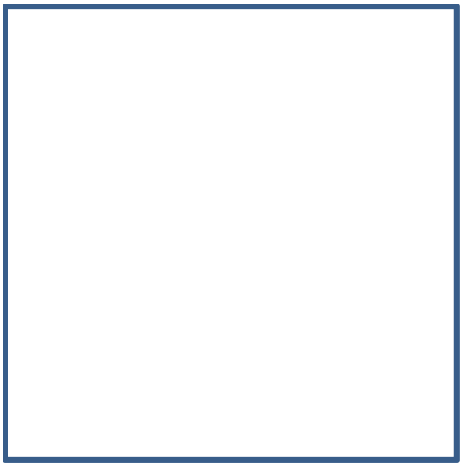
Describe and compare the way that *callbacks* and *listeners* handle events, and indicate one main advantage of each approach.

Draw a possible containment hierarchy for a given picture of an interface.

Given a "before" and an "after" picture, state the sequence of 2D transformations (including translate, scale, and rotate) that would lead to the "after" picture.

The two pictures below show the same interface in two different window sizes. Using the box at right as the main window, draw a diagram of the interface showing the struts and springs (for all components) that would lead to this resizing behaviour.





State the main differences in the capabilities of Android and JavaFX for programming custom 2D graphics.

State the main advantage of a publish-subscribe approach between models and views in MVC, compared to a direct connection between the model and the view.

What is an interaction model in MVC, how does it differ from the regular model, and what is stored in it?

Draw the state diagram for the following Controller code. Specify all events, context checks, and side effects.

Write a Controller class that implements the following state diagram, using a switch statement.

What will be printed when the user clicks on Button A in the pictured interface, given the event-handling and event-filtering code shown below?