

Widget Toolkit: (Homework 5)

By Ramya Balaraman

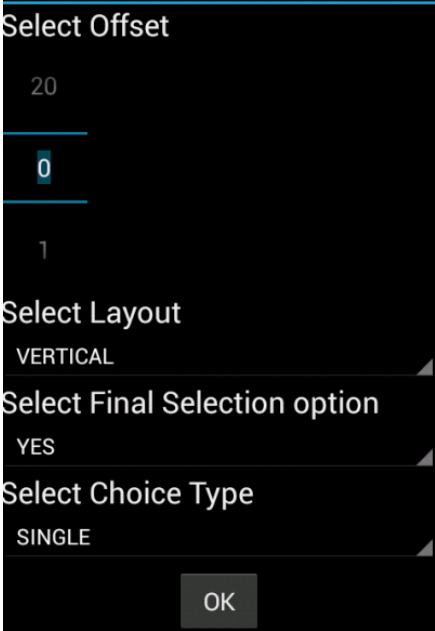
Widgets implemented:

(Separate classes designed for Button, CheckBox, RadioButton and NumberSlider)

Click on TestWidgetToolkit button to test.

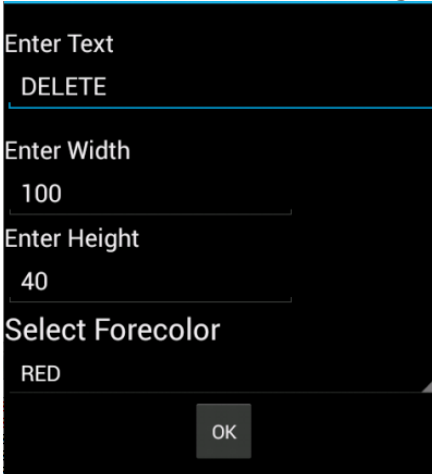
1) Button Panel:

- a. Long click(press and hold) the Panel widget to set options for Panel. Default options are Layout:VERTICAL, Offset:5, ChoiceBehavior SINGLE, and finalSelection TRUE (In case you don't set parameters at all)



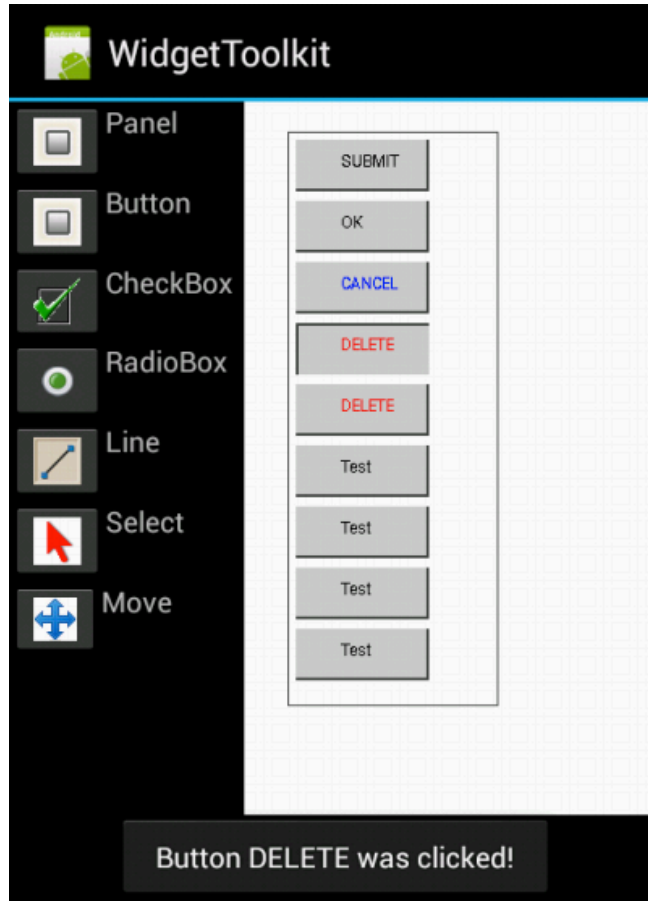
A screenshot of a configuration dialog box for a Panel widget. The dialog has a dark background with white text. It contains four sections: 'Select Offset' with a numeric input field showing '0' and a range from 20 to 1; 'Select Layout' with a dropdown menu showing 'VERTICAL'; 'Select Final Selection option' with a dropdown menu showing 'YES'; and 'Select Choice Type' with a dropdown menu showing 'SINGLE'. An 'OK' button is at the bottom right.

- b. Click the Panel button after returning to the Widget screen and click and drag on drawing area to draw the panel. (Draw a panel with height greater than 300 to test)
- c. Long-click Button widget to set Button parameters. Default parameters are Text:Button, Width: 100, Height: 40, Forecolor: Black.



A screenshot of a configuration dialog box for a Button widget. The dialog has a dark background with white text. It contains four sections: 'Enter Text' with a text input field showing 'DELETE'; 'Enter Width' with a numeric input field showing '100'; 'Enter Height' with a numeric input field showing '40'; and 'Select Forecolor' with a dropdown menu showing 'RED'. An 'OK' button is at the bottom right.

- d. **Click-to-add:** Click on the button widget. The button gets added to the Panel. Multiple clicks will add successive buttons according to the layout selected.
- e. **Click on Select widget.** Thereafter test ChoiceBehavior on the Buttons. A toast message is displayed on button click from the onClick() event of the button.

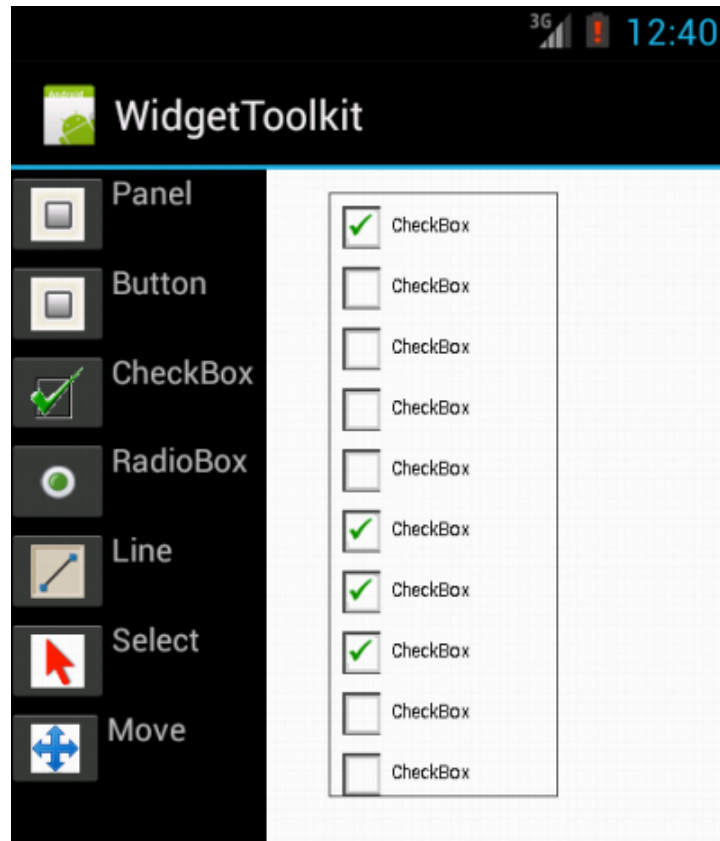


- f. Feedback for the clicked button is shown by making the button appear engraved/depressed.
- g. The button can also take a GraphicalObject parameter and this is demonstrated in NumberSlider where the button takes an Image.

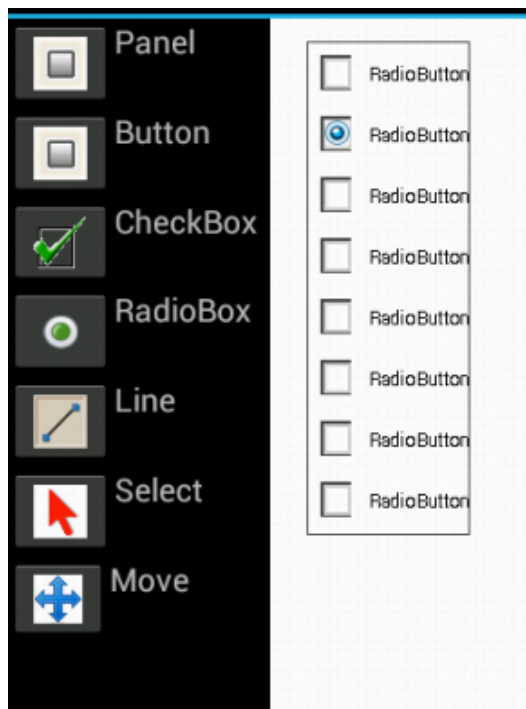
NOTE: Although it is possible to add different widgets, **add only one type of widgets to the Panel** to properly test ChoiceBehavior. In the sense, do not add buttons, checkboxes and radio buttons together to the same panel.

2) Checkbox Panel:

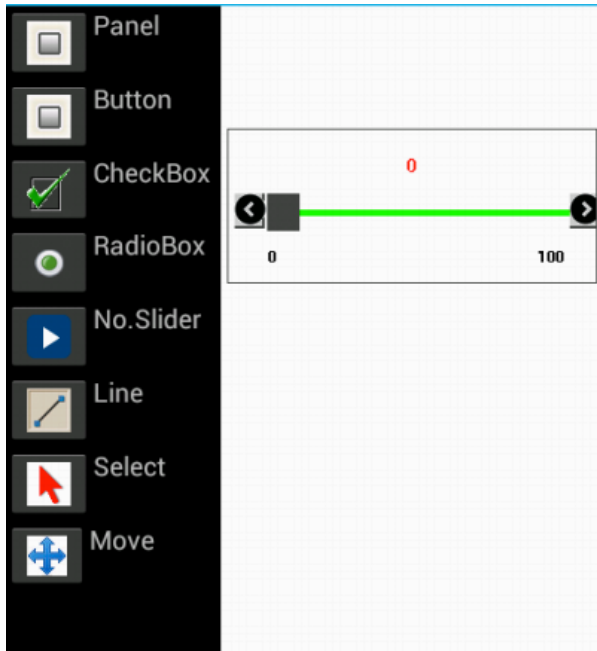
- a. Add a panel as explained above.
- b. Click-to-add: Click on the CheckBox widget to add checkboxes to the panel according to the layout set. (Options similar to Button options for CheckBox label can be given but could not implement due to time constraints)
- c. **Click on Select widget** to test choice behavior.



3) Radio Button panel : Similar to CheckBox Panel



- 4) NumberSlider (Horizontal): Only design; not functional.
- Add a Panel (Horizontal)
 - Click on No. Slider widget



As I was stuck on assignments from other courses and had converted this course to Pass/Fail, I could only begin working on this assignment of late and hence was unable to attempt all required features. However, like every other assignment in this course, this one was also immensely valuable and I learnt a lot. I aim to complete the remaining features over this week or summer break.

Constraint System: (Homework 4)

By Ramya Balaraman

- 1) Every property (integer) of every class can be constrained. Every such property is represented by a corresponding instance of the 'Variable' class.
- 2) Variable holds the value of the property, the list of variables it depends on and the list of variables which depend on it. It also has a flag to indicate if the value is up-to-date or out-of-date.
- 3) To express your constraint, create a class which implements interface Solvable and provides the evaluate() method implementation.

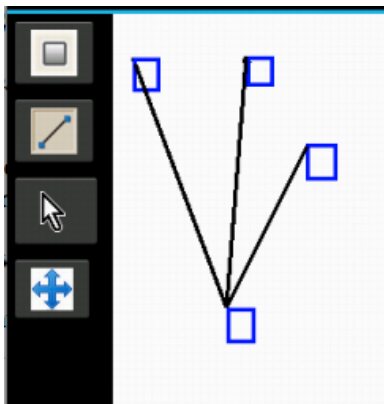
```
public int evaluate(ArrayList<Variable> params, GraphicalObject parent);
```

params is the list of variables the property depends on & parent is the Graphical Object the constrained property belongs to.

- 4) A few examples are provided in TestHomework4.java with comments.
- 5) Call ConstraintSolver.evaluate() before calling redraw() to evaluate all existing constraints on all variables.

Note and Edge Editor: (Only partially done)

1. Test using TestNoteEdgeEditor from main screen
2. Create multiple nodes using rectangle tool on the palette. Create edges between nodes using line tool on the palette.



3. All edges associated with a node move when the node is moved. However sometimes edges leave specks while moving.
4. The system does not allow multiple edges between the same nodes. However instead of canceling the event the program currently throws an exception and exits.
5. Edges cannot be moved by clicking and dragging them directly. (Bug: However they sometimes randomly move when the corresponding node is clicked)