README for Homework 5

Rong Kang Chew (rchew)

See index.html for a test of widgets
See editor.html for implementation of HW1 editor (contains non-text labels)

Integration of Widgets

- Lots of fixes required from Homework 4 with Behaviors, additional bugs with Line/Text bounding boxes, slowing down progress.
- But once this was completed, implementing widgets was not too bad.
- There was no place to fire the onSelectionChange event in the ChoiceBehavior, assigned it to the base GraphicalObject instead when its selected property changes
- Buttons, Checkboxes and Radiobuttons were similar, did not abstract but probably could.
- Certain re-factorizations were done after initially implementing for text-only labels, had to store values somewhere.
- Unfortunately did not get to implementing the slider. Sorry!

Implementation of Editor

- Current state of functionality took about 2.5 hours, and 130+ lines of code in the Editor.
    - Compared to original of about 4 hours and 230 lines of code (plain JavaScript, bootstrap for buttons)
    - This took about 40% of the time, with about the same number of lines of code.
    - Main savings in time was in terms of the new/choice/move behavior as it was already implemented.
    - Code savings were similarly from the behaviors for drawing and selecting rectangles. Previous implementation used bootstrap for buttons so that was already saved minimally.
- Re-implementing the editor using the new framework wasn't too difficult - the better improvements were in terms of behavior, which was already provided from the framework and didn’t have to be re-written. Hooks for selection changes made it easy to update state and attach to other buttons.
- Framework Issues
    - The selectionChange function is used for all items in a group, and this is a little weird for buttons as the function will have to check which button was pressed. Usually each button would have its own function unless explicitly decided.
    - The original implementation of the editor allowed for all actions to be performed with the left button without any modifiers. However, trying to have all 3 types of behaviors tie into just the left mouse button yielded strange behaviors that I did not have the time to debug. Probably something to do with the event chain or bubbling. Main improvements are required here.
    - Drawing of primitive objects were about the same, with the added benefit of the organized constraint system to establish relationships.

Programming Guide for Widgets

Widgets are pre-composed groups of graphical objects and behaviors for user interaction in a graphical user interface.

Pre-built widgets include ButtonPanel, CheckBoxPanel, RadioPanel, all of which extend LayoutGroup. Example of construction with parameters:

```
let buttonPanel = new ButtonPanel(
    labels = [], selectable = false, selectType = SELECT_SINGLE,
    onSelectionChange = function(object, value) {},
    x = 0, y = 0, width = 100, height = 100, layout = HORIZONTAL, offset =
    10);
```

Labels can be a string or a GraphicalObject. Each label becomes a selectable option in the group.

onSelectionChange is fired whenever any of the item's selection status changes. The function is given object - the graphical object whose selected status has changed, and the value it has changed to.
All of the objects will share the same function.

These widgets also have a value property that reflects the state of selection in each group. Widgets with single selections provide the value as a string (the label string itself, or the value property of the label object). Widgets with multiple selections provide the value as a Set of values.

To create your own widget, extend any Group and pre-populate it with your own set of GraphicalObjects and Behaviors to establish the interaction. You may also add properties to be constrained upon.