

Lecture 14: GUI Components

CS 51G
Spring 2018
Kim Bruce

Announcements

- Frogger Questions?
 - Download planning document at
 - <http://www.cs.pomona.edu/classes/cs051G/labs/frogger/plan.docx>
- Pay attention to red error messages at bottom of window.
 - Go to top and look for first error indicating line in your code!

Let's Do GUI!

GUI Items

- Components so programmer can interact with system.
 - Text fields (for input!)
 - Labels
 - Buttons
 - Menus
 - etc.
- Can add to screen before or after canvas

First Examples

- Input from a text field
 - <http://www.cs.pomona.edu/classes/cs051G/demos/TextController/TextController.grace>
 - Create with class `textFieldLabeled(startString)`
 - It is NOT on canvas, so no canvas parameter!
 - Access value by sending the method request "text"
 - `var curValue: String := input.text`

Buttons & Events

- Add a button:

- <http://www.cs.pomona.edu/classes/cs051G/demos/TextButtonController/TextButtonController.grace>
- Create with class `buttonLabeled(startString)`

- If button pressed, mouse event is generated

- Associate action to a button by sending an `onMousePressedDo` message to button with action to be performed:

```
clearButton.onMousePressDo {mevt: MouseEvent ->
    canvas.clear
}
```

- `mevt` is the event generated. Action after arrow is performed after each button press

Adding GUI Items

1. Create the item. E.g.,

- `def shapes: Choice = menuWithOptions
 ["FramedSquare", "FramedCircle", "FilledSquare"]`
- `def clearButton: Button = buttonLabeled ("Clear screen")`

2. Add the items to window with `append` or `prepend`. E.g.,

- `append (shapes) // add after other existing items`
- `prepend (clearButton) // add before other existing items`

Example

- Program using a menu
 - <http://www.cs.pomona.edu/classes/cs051G/demos/DrawingProgram/DrawingProgramSimple.grace>
 - `menuWithOptions[, , ,]`
 - takes list of options — must use square brackets.
 - access label with method selected
 - Notice match `case{...} case{...}...`

Handling Events

- Variant where no more onMouseDown method!
 - <http://www.cs.pomona.edu/classes/cs051G/demos/DrawingProgram/DrawingProgram.grace>
 - Notice canvas.onMouseDown {mevt -> ...}
 - When a press event happens on canvas, take mouse event and perform an action.
- Two different menus, two different approaches:
 - <http://www.cs.pomona.edu/classes/cs051G/demos/DoubleDrawing/DoubleDrawing.grace>
 - Shape menu only used when new object constructed
 - Color menu triggers action immediately
 - colorMenu.onChangeDo { evt: Event -> ... }

Organizing GUI Components

- Can put components in Containers
 - Start with emptyContainer and append components
 - Can even put containers in other containers!
 - Example:
 - <http://www.cs.pomona.edu/classes/cs051G/demos/FallingBall/FallingBall.grace>

Questions?