Lecture 16: Inheritance

CS 51G
Spring 2018
Kim Bruce

Announcements

- Lab for next few Fridays will be moved upstairs in Edmunds 219
  - I will be covering Java lab as well as Grace at same time!
- Homework:
  - 10.8.1
  - 10.8.3

Test Programs

- One on Friday in lab
  - Must turn in by 4 p.m.
- Two others available by one week from today
  - maybe earlier
- No coverage of GUI components
  - Dragging and interacting w/objects
  - Designing classes
  - Animations

Responding to Key Presses

- Can tell application to pay attention to key presses.

    -
      ```
      onKeyPressDo{keyEvt: KeyEvent ->
          // do something
        }
      ```
- Can ask KeyEvent for code of key:
    - http://www.cs.pomona.edu/classes/cs051G/demos/KeyDemo/KeyDemo.grace
- How can you control frog with arrow keys?
    - http://www.cs.pomona.edu/classes/cs051G/demos/MoveBoxWKey/MoveBoxWKey.grace

Mysteries of Objectdraw

- Where did canvas come from?
    - Why do we have to pass it to some classes and not others?
- What about startGraphics, append, prepend?
    - Why is there no receiver? Where are they defined?
- What about onMousePressed, etc.?
    - Why are they special?

Similar Things

- What is a strobe light?
  - a light that flashes — extra behavior
- What is a poisonous snake?
  - A snake whose bite is poisonous — new effect of behavior
- What is a text book?
  - Book with extra features — problems
  - … and a ridiculous price!

Inheritance & Similarity

- Creating GraphicApplications:

```
def fallingLeaves: GraphicApplication = object {
        inherit graphicApplicationSize (500 @ 500)
```

- All have similar behavior except what we specify.
    - E.g. how it draws display and reacts to mouse actions
- fallingLeaves is subobject of graphicApplication

Compare

- Falling leaves and falling ball from pong
    - What do they have in common
        - Start at top of window and fall
        - Fall at constant speed
        - Disappear at bottom of screen
    - What is different?
        - Appearance
        - Rate they fall
        - What creates them

Creating Falling Objects

- Create FallingObject that has all the common characteristics
    - http://www.cs.pomona.edu/classes/cs051G/demos/FallingObjects/FallingObject.grace
    - Constructor takes parameters to decide
        - where it should fall to
        - speed it falls

Classes

- FallingObject
  - has instance var theObject:Graphic2D and start method
- FallingLeaf
  - inherit fallingObjectAt... (so gets start method!)
  - Initializes theObject
- Hail
  - inherit fallingObjectAt... (so gets start method!)
  - Initializes theObject

Inheritance Example

- Anything that inherits fallingObject need only initialize theObject with Graphic2D

More Example

- FallingObjectsWithTomatoes?
  - http://www.cs.pomona.edu/classes/cs051G/demos/FallingObjectsWithTomatoes/FallingObjectsWithTomatoes.grace
  - Differing behavior when hit the ground
  - Define method hitBottom to specify behavior after fall.
- FallingObject is "abstract classes"
  - Only defined to inherit from!

Questions?