CS 51 Test Program #1

Due: Friday, October 17, at 4 PM

A test program is a laboratory that you complete on your own, without the help of others. It is a form of take-home exam. You may consult your text, your notes, your lab work, or our on-line examples and web pages, but use of any other source for code is forbidden. You may not discuss these problems with anyone aside from the course instructor. You may only ask the TA's for help with hardware problems or difficulties in retrieving your program from a disk or network.

If you get an error message that mentions in internal compiler error, or another error message that makes no sense (e.g., refers to a line of your program that does not exist), you may send a copy of the program (along with a copy of the error statement) to the instructor at kim@cs.pomona.edu. If I find that the error statement is not as good as would normally be expected, I will provide you with a better error statement.

Complete each of the following three problems, documenting your code well. You are encouraged to reuse the code from your labs or our class examples. Submit your code in the usual way by dragging it into the Dropoff folder. Please do *not* submit three separate folders. Instead, place all three of your complete programs into one folder, make sure that your name appears in the title of the folder, and then drag the folder into our dropoff folder.

To receive full credit, your programs must compile correctly using the `rtobjectdraw` dialect.

Problem 1: A new broom sweeps clean

For the first problem, you are to write a program that emulates a broom that can sweep a rectangle around the screen. The broom will be represented by a small gray ball with diameter 20 that appears when the mouse is pressed and disappears when the mouse is released. While the mouse is depressed, the ball is dragged along by the mouse (the upperleft corner of the ball is always at the mouse location). There is also a filled square (30 pixels to a side) showing on the screen. If the broom bumps into the square during a drag, then the square will move the same distance (and in the same direction) as the broom, as indeed it would if you had a real broom.

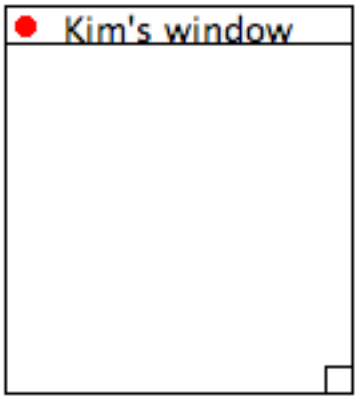
Notice that we are asking you to have the broom push the square, not just drag it. Thus if the mouse is depressed when it is not over the square, but then is dragged over the square, the square should be pushed ahead of the mouse (that is, after the drag, the mouse will still not be on top of the square). Of course if the mouse is depressed on top of the square, then the broom will drag the square, but this program should also be able to push (and the code to do the pushing will automatically do that dragging). Please use the method `overlap` to determine if the broom has been moved over the square.

One warning: If you are pushing the square diagonally, the square will have a tendency to slide off of the broom. This is because movements parallel to the side of the square will not affect the square

(and the computer is so fast that some of your diagonal moves will end up parallel to the square). This is a "feature" not a "bug" (this is what happens with real brooms), so don't worry about it as long as the square generally is moved along with the broom.

Problem 2: Playing with Windows

For this program you are to create a class to represent a window on a computer screen. The diagram below shows what such a window should look like (except that your window should have a title with your name replacing "Kim").



As you can see, the window has a title bar (including title), an outline, a "go-away circle", and a resize box. The title bar should be 18 pixels high and extend across the entire top of the window, the "go-away circle" should have diameter 12 pixels while the resize box should be about 15 pixels square and should occur in the lower right corner of the window.

Like most computer windows, windows from your class can be dragged by grabbing them on the title bar and dragging. They can also be resized by grabbing the small square in the lower right hand corner and dragging that around. The window is closed if the user clicks on the "go-away circle".

We have supplied you with an object, windowManager, inheriting graphicApplication, that creates a window on the screen and then determines whether the window should be dragged or resized (or indeed do nothing at all) in response to user actions, and sends appropriate messages to the window.

You are to write a class window that can be used with class windowManager and that has the appropriate behavior. As well as the constructor, the class should support methods given by the type Window:

```
type Window = {
  // Resize to make lowerRight the new bottom right hand corner of the window
  resizeTo(lowerRight: Point) -> Done

  // Move window to right by dx and down by dy
  moveBy(dx: Number, dy: Number) -> Done

  // Is pt in the resize box of the window
  inResizeBox(pt: Point) -> Boolean
```

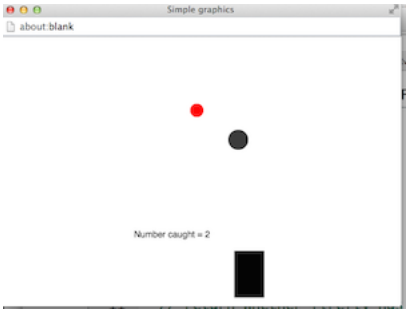
```
  // Is pt in the title bar of the window
  inTitleBar(pt: Point) -> Boolean

  // Is pt in the go-away circle in the upper left of the window
  inGoAwayCircle(pt: Point) -> Boolean

  // Remove the window from the canvas
  removeFromCanvas -> Done
}
```

You should not modify the windowManager class in any way to accomplish the required parts of this program. Feel free to add other graphic items to the Window class to make it look more attractive.

Problem 3: Catching Fireflies

For this problem, you will write a program that simulates the fun of catching fireflies. When the game starts, a firefly will start flying across the screen at a constant speed. (You need not worry about computing elapsed time for this project, just always move the firefly by the same amount each step.)

The user will control a "net" by moving a mouse around the screen. The center of the net will correspond to the current mouse location. If the firefly is inside the net, and the user presses the mouse button, then the firefly will be captured inside the net (and the net will "light up" to the color of the firefly to show this). If the mouse is released inside the jar when the firefly is caught in the net, then the count of the number of fireflies captured should increase. If the mouse is released anywhere else, then the firefly "escapes", and the count is not updated. Either way, when the firefly is released, a new firefly begins going across the window. The game will end if a firefly makes it all the way across the screen without ever being caught in the net.



You will need a fireflyGame object that **inherits** graphicApplication, and a firefly class to represent the firefly. The jar and the net can be represented as simple geometric objects, and do not need separate classes. You may set the speed of the firefly as you wish, but remember that fireflies are easy to catch!

The fireflyGame class will need methods onMousePress, onMouseMove, onMouseDrag, and onMouseRelease, as well as initialization code. Note that to catch the firefly when the mouse is pressed, we only need for the center of the firefly to be inside the net.

The firefly class, which will use the "animation" module, will be represented by a small red circle. It will need to provide the methods given in type Bug:

```
type Bug = {
  // start the animation
  start -> Done

  // returns center of bug
```

```
  center -> Point

  // return whether firefly has been caught
  caught -> Boolean

  // Reset whether caught is true
  caught:=(isCaught:Boolean) -> Done

  // remove the bug from the canvas
  removeFromCanvas -> Done
}
```

After the bug stops moving, its representation should be removed from the canvas. Your program window should be 600 pixels wide by 400 pixels tall.

Table 1: Grading Guidelines

Value	Feature
	Code Quality & Readability (16 pts for each of 3 programs)
2 pts.	use of boolean conditions
2 pts.	ifs/whiles
2 pts.	appropriate vble (instance/local, public/private)
2 pts.	Descriptive comments
2 pts.	Good names
2 pts.	Good use of constants
2 pts.	Appropriate formatting
2 pts.	Parameters used appropriately
	Correctness (16 pts for each of 3 programs)
Broom	
4 pts.	Drawing the screen initially
4 pts.	Dragging the broom
4 pts.	Pusing the "dirt"
4 pts.	Showing/hiding the broom
Windows	
4 pts.	Drawing a window
3 pts.	Moving the window
3 pts.	Resizing the window
3 pts.	Making the window go away
3 pts.	Centering title at all sizes
Firefly	
2 pts.	Net moves with the mouse
3 pts.	Firefly moves smoothly across the screen
3 pts.	Firefly captured when mouse pressed over firefly
3 pts.	Releasing firefly in jar adds to score
3 pts.	Releasing firefly anywhere results in new firefly
2 pts.	Game ends when firefly gets across screen
	Miscellaneous (4 pts total)
	Extra Credit (4 pts maximum)