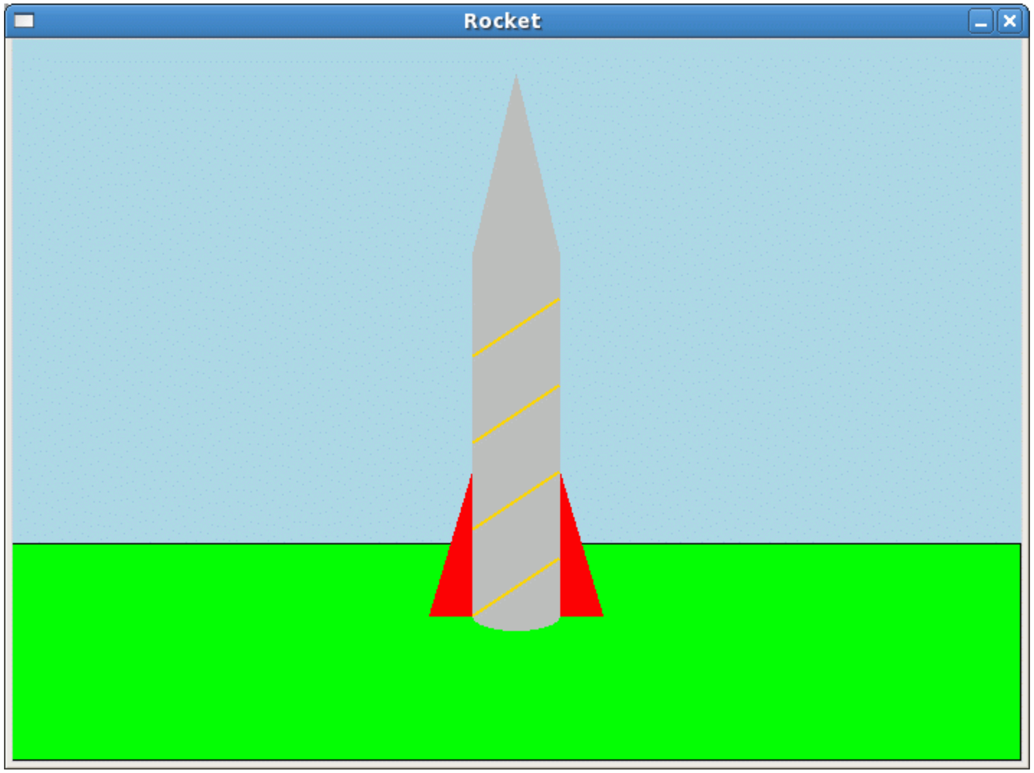
Objectives

- Continue Object-Oriented Programming with Graphics
- Broader Issue: Algorithms (postponed from Friday)

Review: Object-Oriented Programming

- What is the term for how we create a new object?
  - What is the syntax for that?
- What is the term for how we give commands to/do operations on objects?
  - What is the syntax for that?
  - What are two types of those operations we talked about?
    - What is the difference?
    - How does that effect how we use them?
- How do we get access to the code in `graphics.py` in our code?
- How can we find out what we can do to an object?
  - How can we make a duplicate of a drawable object using the Graphics API?

What objects make up this scene?



Colors

- Strings, such as "blue", "red"
  - Add numbers for darker colors, e.g., "red2", "red3", "red4"
- Can also create colors using the *function* `color_rgb(<red>,<green>,<blue>)`
  - Returns a color, as specified by its amount of red, green, and blue
    - Parameters in the range [0,255]
- Example use:

```
darkBlueGreen = color_rgb(10, 100, 100)
win.setBackground(darkBlueGreen)
```

  - Background is a dark blue/green color
- Example color codes: `http://en.wikipedia.org/wiki/List_of_colors`

Using the Graphics Library

- Create an instance of a Rectangle that is blue and 50x100 pixels in the upper left of the window

- Draw the rectangle

- Shift the instance of the Rectangle class to the **right** 10 pixels

- Find out the x- and y- coordinates of the upper-left corner of the Rectangle now

`rectangle.py`

Getting Input from the User

- `<GraphWinObj>.getMouse()`
  - Returns the user’s mouse click as a **`Point`** object
- Entry objects
  - Get text from user

OO Terminology Summary

Term	Definition	Examples
Class	A data type. Defines the data and operations for members of the class	`str, SmartPhone, GraphWin`
Object	An instance of a specific class	`animal, myPhone, window`

Method	Operations you can call on an object	`setBackground(<color>),`
		`getWidth()`

Constructor	Special method to create an object of a certain type/class	`GraphWin(), str(1234)`

Benefits of Object-Oriented Programming

- **Abstraction**
  - Hides details of underlying implementation
  - Easier to change implementation
- Collects related data/methods together
  - Easier to reason about data
- Less code in main program
  - Our program code is relatively simple

Broader Issue: Typical Process

1. Break into assigned groups
2. Introduce yourselves
3. Answer questions in groups
4. Discuss in class

Broader CS Issues

- Good summaries!
  - Good English, complete sentences
  - Followed the specifications
- Good, thoughtful questions
- Interest scale is 0 to 9
  - Recall: Lab 0
  - Why we start at 0 will be clearer soon…

Algorithms Everywhere

- How does knowing how your brain thinks about code affect how you think about code?
- Comment on these from articles:
  - “Because it’s less familiar, *algorithm* tends to emphasize our uncertainty.”
  - “An algorithm is, essentially, a brainless way of doing clever things.”
- What are examples of algorithms that you use every day?
- What is machine learning useful for?
- What aren’t algorithms useful for?
- What would be some useful algorithms, specific to W&L students?
  - What are problems that are difficult—but useful—to solve?

My Corrections to Articles

- “In his book *The Master Algorithm*, Pedro Domingos offers a masterfully simple definition: ‘An algorithm is,’ Domingos writes, ‘a sequence of instructions telling ~~a computer~~ what to do.’”
- “An algorithm is, essentially, a ~~brainless~~ way of doing clever things.”

This Week

- Pre Lab 2 – due tomorrow before lab
- Broader Issue due Thursday at 11:59 p.m.