Objectives

- Nondeterminism
- Problem solving using Graphics API
  - Animation

Nondeterministic Decisions

- Sometimes, we don't want to necessarily know that a specific decision is always made
- For example, games often use randomness to make decisions
  - Roll dice
  - Coin flips
  - Location and behavior of baddies

How can we simulate coin flips?

Flipping Coins

- Simulate by randomly selecting between 0 (heads) and 1 (tails)
- Program: `coinFlip.py`

Defined Constant

```
# flip the coin
if randint(0,1) == HEADS:
    print "heads"
else:
    print "tails"
```

- Problem: How many flips does it take to get 3 consecutive heads?

`consecutiveHeads.py`

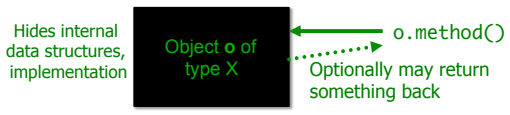
Review: Object-Oriented Programming

- How do we create a new object?
- How do we give commands to/do operations on objects?
- What is the syntax for calling a method on an object?
- What are two types of methods we talked about?
  - How do they work differently?

Review: Object-Oriented Programming

- Objects combine data and methods together

Provides **interface** (the methods) that users interact with



Use an Application Programming Interface (**API**) to interact with a set of classes.

OO Terminology Summary

Term	Definition	Examples
Class	A data type. Defines the data and operations for members of the class	string, TV, GraphWin
Object	An *instance* of a specific class	animal, myTV, 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)

Always need to create/construct an object before using it.

Review: Benefits of Object-Oriented Programming

- **Abstraction**
  - Hides details of underlying implementation
  - Easier to change implementation
- Easy reuse of code
  - We used the `graphics.py` package
- Collects related data/methods together
  - Easier to reason about data
- Reduces code in program

Review: What Does This Code Do?

- Use OO terminology previously defined

```
from graphics import *

win = GraphWin("My Circle", 100, 100)
point = Point(50,50)
c = Circle(point, 10)
c.draw(win)
win.getMouse()
```

Constructor

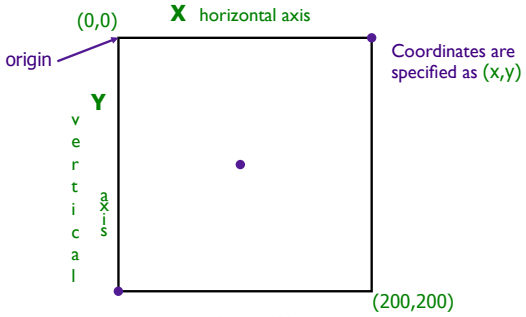
GraphWin object

Also known as an **instance of** the **GraphWin class**

Method called on GraphWin object

Note: Class names start with capital letters, Method names start with lowercase letters

Review: A GraphWin Object's Canvas



Problem: Draw a Full-Canvas Tic-Tac-Toe Board

- Using the Graphics API
- Make lines purple and line width 3
  - Keep it general, regardless of GraphWin width, height

Modification to Tic-Tac-Toe

- **clone** a vertical line and horizontal line and shift appropriately
- Why clone?
  - Maintain the same properties (color, line-width, length)
  - Simplifies code

Broader Issue: DARPA Urban Challenge

- Challenge: automated cars in an urban setting
  - Deal with human drivers, automated drivers
  - Correctly obey traffic laws
  - Winners: 1st - $2Mill, 2nd - $1Mill, 3rd - $500K
  - Apply for $1Million in "seed money"

Ola, Minh, Colin, Meng

Anh, Lida, Yates, Nick

Jean Paul, Callie, Will

DARPA Urban Challenge

- Will you feel safe (safer?) with an automated driver in the lane next to you?
- What guarantees about the cars would you want from the company/government?
- Are there situations that would be particularly difficult for software to handle that a person would be better equipped to handle?
- Was there any part of the DARPA Urban Challenge in the other article?
- What should the next DARPA Challenge be?
  - In a year?
  - In 5 years?

Relation To Our Class

- One IF statement from victory
  - Programming you're learning applied to major tasks
- Importance of testing
  - Need robustness, reliability of systems
  - Test the "small"
- Likely, infinite loop waiting for signals
- Imagine an API to car
  - `move(direction, amount)`
  - `getSpeed()`
  - …

Looking Forward…

- Tomorrow: Lab
  - Due Friday
- Wednesday: Bring your questions about the exam
- Friday: Exam
  - See preparation document online