Announcements

- Midterm exam on Tuesday
  - Closed book, closed notes, closed neighbor
- Chaps 1-2, Chaps 4-5, html
  - Classes, objects, inheritance
  - Events and event handling
  - Methods, instructions, parameters, properties
  - Storyboards, design

Functionality

- A question (or function)
  - Receives value(s)
  - Performs computation on value(s)
  - Returns (sends back) a value

What we will do today

- Lecture on Chap 6, Sec 1 - Functions
- Classwork

Note: thanks to Wanda Dann and Steve Cooper for slide ideas
Types of functions

• The type of a function depends on the type of value it returns
  – a calculated value (a number)
  – a specific object
  – a color
  – etc.

Built-in functions

• Used one of Alice’s built-in functions
  – skateAround method for the cleverSkater

Example

• How do we bounce a ball? Let’s bounce a ball over a net
  – Ball is 1 meter from the net to start
  – Ball should move up and forward, then down and forward

  – Note: Looks easy – but do not be deceived!

Design Storyboard

• Design for a world-level method

  World.ballOverNet:
  Do in order
  toyball turn to face the net
  Do together
  toyball move up
toyball move forward
  Do together
  toyball move down
toyball move forward

• To reach the top of the net
  – Ball should move forward 1 meter (we positioned it 1 meter in front of the net)
  – How far upward should the ball move to clear the net?
Height

• Can use the built-in height function to determine the height of the net and move the ball up that distance

Problem

• The ball does not bounce over the net
• The problem – cannot tell “which way is up” from the perspective of the ball

Demo – ballOverNetWork – what happens?

Solution

• We think “up” and “down” relative to the ground – so can orient the ball (and net) with the ground
• (0,0,0) 1
• Now, ball will bounce over the net

Rolling the ball

• How do we roll a ball along the ground?
• Want a realistic motion rather than a slide
• The ball must simultaneously move and roll.
Demo: A first attempt

| toyball | move right 1 meter asSeenBy = ground more...
| toyball | turn right 1 revolution more...

Revising the Approach

- The ball is made to roll 1 revolution.
- What if we want the ball to roll a certain distance?
- How can we make the ball roll the correct number of revolutions to cover a given distance along the ground?

Number of Revolutions

- The number of revolutions depends on the size of the ball
  - The number of revolutions is distance / (Pi * diameter)
- There is no built-in function to return the number of revolutions
  - Must write it!

Parameters

- We want to return the value computed as Distance / Pi * diameter
- Obviously, what is needed
  - The ball’s diameter
    - The ball object has a built-in width question
  - The distance the ball is to travel
    - Can be sent as a parameter to the question
**numberOfRevolutions** function

```plaintext
toyball.numberOfRevolutions : 122 distance

No variables

Do Nothing

Return (distance / (3.14 * subject = toyball * s.width))
```

---

**Demo: Calling the function**

```plaintext
toyball.realisticRoll 0 parameters

No variables

Do together

- toyball move forward 10 meters asSeemly = ground duration = 3 seconds more...
- toyball turn forward = toyball numberOfRevolutions distance = 10 duration = 3 seconds more...
```

This is a test value

- Run the animation with several test values
- Make sure it works as expected
- What happens if you use a negative value?

---

**Levels of functions**

- As with methods, you can write functions as either class-level or world-level. (what was the function we just wrote?)
- Guidelines for class-level methods apply to class-level functions:
  - No references to other objects
  - No references to world-level functions
  - Built-in world-level functions are ok

---

**Classwork today**