Announcements

• Exam 1 Thursday, Oct 3
  – Closed book, closed notes, closed neighbor
  – Chaps 1-2, Chaps 4, 6, html
  – On Thursday, will give you an old exam to work on, then review it next Tuesday
• Assignment 4 storyboard due today
  – Alice world due Thursday

What we will do today

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

Functionality

• A function
  – Receives value(s)
  – Performs computation on value(s)
  – Returns (sends back) a value
  – “return value” – means “value” is the answer, you leave the function (don’t execute any more code in the function) and return the output
  – You should use the output value
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

• Last time used one of Alice’s built-in functions
  – skateAround method for the cleverSkater uses the distance to built-in function

• Let’s look at other examples.

Example of a function

• Write a function to calculate the perimeter of the bounding box of the function.
• I want to walk all the way around the chair, how far is that?
• Know width and depth of chair
How do we write this function?

• Name of function?
• Parameters and types?
• What type is returned?

• How do we call/use this function?
• Is this a class function or world function?
    – Would we use this function with other objects?

Built-in String functions

• String functions are under “world”, “functions tab”
  – “a joined with b” - join two strings together into a longer string
  – “what as a string” – converts a number into a string (6 into “6”)

Exercise: Write a function called phraseHowTall

• Given an object, this function should return a string “I am this tall: num” where num is the actual height of the object
• Then have objects say the result of calling this function
Longer Story Example

- Move ball to within 1 meter of net, then bounce ball over the net.
  - Bounce - Ball should move up and forward, then down and forward

Move Ball to 1 meter from Net

- Use “distance to” function and math

Height

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

Rolling the ball

- How do we roll the ball along the ground?
- Want a realistic motion rather than a slide
- The ball must simultaneously move and roll.
- The ball must roll “as seen by” ground
- The ball and ground must face the same direction

Demo – what happens?
Demo: A first attempt

```plaintext
toyBall1.test  No parameters

No variables

- toyBall1 = turn to face tennisNet = more... 
- ground = turn to face tennisNet = more... 

Do together
- toyBall1 = move forward = 2 meters = asSeenBy = ground = more... 
- toyBall1 = turn forward = 2 revolutions = 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 \( \frac{\text{distance}}{\pi \times \text{diameter}} \)

- There is no built-in function to return the number of revolutions
  - Must write it!

Parameters

- We want to return the value computed as \( \frac{\text{Distance}}{\pi \times \text{diameter}} \) where \( \pi = 3.14... \)

- Obviously, what is needed
  - The ball’s diameter
    - The ball object has a built-in width function
    - The distance the ball is to travel
      - Can be sent as a parameter to the function
**numberOfRevolutions function**

```java
toyball.numberOfRevolutions(distance)
```

- Do Nothing

  - Return `distance / (3.14 * subject = toyball)`

---

**Demo: Write method realisticRoll to Call the function**

- 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?
- Add a parameter for distance

---

**Now Ball roll to net?**

- Difficult…
- ToyBall turn to face TennisNet and roll, what happens?

---

**Tricky – Orient To**

```java
toyBall1.orientToGround more...
```

```java
toyBall1.turnToFaceTennisNet more...
```

```java
toyBall1.orientToWorld more...
```

```java
toyBall1.turnToFaceTennisNet more...
```

```java
do together
```

```java
toyBall1.move.forward 2 meters more...
```

```java
do in order
```

```java
toyBall1.move.up (subject = tennisNet) more...
```

```java
toyBall1.move.down (subject = tennisNet) more...
```
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
Calling Perimeter Function

- Code in myFirstMethod

```
world.perimeter (width, depth)
```

No variables

Do Nothing

```
(2 * width) + (2 * depth)
```

phraseHowTallFunction

Calling the function:

```
bunny say world.phraseHowTall creature = bunny duration = 2 seconds
camel say world.phraseHowTall creature = camel duration = 2 seconds
```