Creating your own Questions

Alice
Overview

- Review of built-in questions
- Need for other information
- Writing Boolean questions
- Types of questions
- Writing Numeric questions
Review: Built-in questions

In our programs, we have been using many of Alice’s built-in questions to get information about objects and the world:
- LilFish’s distance to the FireCoral
- Ball’s height
- Cloud’s position
- who = = Homer
Need for other Information

But, sometimes you will want to get other information not provided by the built-in questions

Which is the tallest character in a world?
Which bowling-pin is closest to the bowling-ball?
A first example: Is the first object closer than the second?

A bee is hunting for pollen. He is going to fly over to either the red flower or the pink flower, depending on which flower is closest to him.

How can we know which flower is closer to the bee?
Creating a Boolean question

We can write our own question to ask “Is this object closer than another object?”

The question needs three parameters

Parameters: comparingObject, FirstObject, secondObject

If comparingObject’s distance to firstObject is less than comparingObject’s distance to secondObject
    return true
Else
    return false

This is a **Boolean question** because it returns either a *true* or *false* value
The question in Alice

world.isFirstCloser

If/Else, Loop, While, For all in order, print, //, Return, +, -, *, /

comparingObject, firstObject, secondObject

create new parameter

create new variable

If
comparingObject, distance to firstObject, comparingObject, distance to secondObject

Return true

Else
Return false
Since the question returns a Boolean value, it can be used as the conditional expression in an if statement.
A more complicated example: Avoiding a collision

The biplane and helicopter are flying in the same airspace and a collision is possible.

We want to write a question that returns true or false, depending on whether or not the two objects are at approximately the same altitude (in danger of collision).
The heightDifference question

```plaintext
World.heightDifference [first, second]

No variables

If first distance above Ground more... == second distance above Ground more...
Return true

Else
  If both first distance above Ground more... > second distance above Ground more... and first distance above second more...
  Return true

  Else
    If both second distance above Ground more... > first distance above Ground more... and second distance above first more...
    Return true

    Else
      Return false
```

create new
Demo: Invoking the question

If the heightDifference question returns true, a Danger sign (3D text object) blinks.
Types of questions

We have looked at two Boolean questions. But, questions may also be written to return:

- a calculated value (a number)
- a specific object
- a color
- etc.
Example: Number Question

We want to animate a ball rolling to the right along the ground in a realistic motion (NOT just slide along as if it were skating on ice).

To do so, the ball must simultaneously move and roll
Demo: A first attempt

Why is the *asSeenBy* parameter needed?
Revising the approach

The ball is made to roll 1 revolution. Suppose we want the ball to roll a certain distance along the ground.

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
\[ \frac{\text{distance}}{P \times \text{diameter}} \]

But, there is no built-in Question to return the number of revolutions.

So, we will write one!
Parameters

We want to return the value computed as

\[ \text{distance} / (\pi \times \text{diameter}) \]

Obviously, what is needed is

- 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
howManyRotations Question

The diagram shows a calculation for the number of rotations a toy ball makes. The calculation involves multiplying the distance by a constant factor and then dividing by the width of the toy ball. The variable names and values are:

- `toyball`: The subject name is `toyball`.
- `distance`: The distance is represented by `distance`.
- `width`: The width is represented by the subject's width.

The formula can be summarized as:

\[
\text{Number of rotations} = \frac{\text{distance} \times 3.14}{\text{subject}=\text{toyball}'s\ width}
\]
Invoking the Question

This is a test value.

We should run the animation with several test values to be sure it works as expected.

What happens if you use a negative value?
Levels of Questions

As with methods, programmer-defined Questions can be categorized as either character-level or world-level.

The Question just presented was written as a character-level method.

If we wanted the question to be applicable to any circular object, it could be written as a world-level method.
World-level Question

Note that the diameter of the object is now a parameter – because we do not know which object is being used.
Guidelines for Character-level Questions

The guidelines for character-level methods also apply to character-level questions:
- No references to other objects
- No references to world-level programmer-defined Questions
  - but, built-in world-level questions are fine to use
Assignment

Read Chapter 6 Sections 2 and 3

- Creating a new question
- Calling the question
- Types of questions
- Testing
- Character-level and world-level