Announcements

- Start Java next time
  - We will use Eclipse with Java
- Extended Office Hours.
  - Tue (today): 10:15-11:15, 1:30-2:15pm
  - Wed: 11:00-11:40am, 1-2:15pm

Topics for the Exam

- No HTML
- Alice Chapters 1-2, 4-7, 8.1, 9-10
  - Focus on Chap 5-7, 8.1, 9-10
- Topics
  - Random numbers
  - Looping structures
  - Variables
  - If/else
  - Lists/Arrays
  - Recursion

Random Numbers

- Assume they work
  - Can use == to compare integers
  - Use “integer only” if you want an integer
  - Use “random number” to generate random numbers
    - Maximum is “up to but not including”
    - Default values
Looping Structures

• Loop
  – Simple and complicated
• While
• When to use “while” vs. “if”
• BDE

Conditionals – If/Else

• Format
  if (condition)
    do if true
  else
    do if false
• Nested if's
• 3 Logical and 6 relational operators
• When do you use an “if” vs “while”

Variables

• Types
  – Local variable
  – Parameter
  – World variable
  – Class variable
• What do you do with a variable?
  – It stores a value
    • Initialize it
    • Use its value
    • Update it

Lists

• Create a list
• Process a list
  – For all in order
  – For all together
  – Item-in-list
  – Do something to a part of an object
• List search
Array

- Create an Array
- Process an array
  - Loop – complicated version
  - Use value from array
  - Set value in array
- Search for item in an array

Recursion

- Function or method call a “clone” of itself
- Must get closer to ending with each step
- Must be a way out
- Recursion is a loop, but you don’t use “loop” command
- You will need to know recursion at “reading level”

How to Study for the Exam

- Write methods from old classworks
- Lecture notes
- Reading quizzes – will make available
- Reading in book
- Understand the topics

Problem

- Objects are a Bunny and a bunch of frogs. They are all facing the same direction. The Bunny is behind the frogs.
- The frogs are all in a list called “frogs”
- The Bunny and frogs will be moving each time an event occurs. The bunny will move 0.5 meters and the frogs 0.15 meters each time.
- When the Bunny is at least the “Bunny’s depth” in front of all the frogs, it should turn around and say “Stop”
Write function IsBunnyInFront

- Returns true if Bunny is at least Bunny’s depth in front of all the frogs in the list frogs
- Else returns false

Write method moveAndCheck

- Moves all the frogs in the list frog 0.15m
- Moves the bunny 0.5m
- If the bunny is in front of all the frogs, turns around and says Stop.
```plaintext
world.moveAndCheck

world.moveAndCheck  No parameters

No variables

- Do together
  - For all world.frogs, every item_from_frogs together
    - item_from_frogs move forward 0.15 meters
  - bunny move forward 0.5 meters
  - If world.isBunnyInFront
    - bunny turn right 0.5 revolutions
    - bunny say stop
  - Else
    - Do Nothing

- world.isBunnyInFront
  - No parameters
  - No variables
  - For all world.frogs => one (0) item_from_frogs at a time
    - If bunny is behind item_from_frogs => more...
      - Return false
    - Else
      - Do Nothing
  - For all world.frogs => one (0) item_from_frogs at a time
    - If bunny distance in front of item_from_frogs => more...
      - subject - bunny's depth
    - Else
      - Return false
  - Else
    - Do Nothing
  - Return true
```

The text describes two AI procedures, `world.moveAndCheck` and `world.isBunnyInFront`. `world.moveAndCheck` includes a `Do together` segment where actions are taken for all world.frogs, and another segment that checks for a bunny's presence and executes actions accordingly. `world.isBunnyInFront` checks if a bunny is in front of a specific item and returns a boolean value.