

CompSci 4
Chap 7 Sec 2
Nov 1, 2005



Prof. Susan Rodger

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

Announcements

- Read Chapter 9.1 for next time
- Assignment 6 due Nov 8
 - Finish classwork from Thursday first...

What we will do today

- Lecture on Chap 7 Sec 2 and Tips and Tech.
 - While loop – indefinite loop
 - Event Loops
- Classwork

Last time -Loop – definite number

- What is the output?

world.my first method *No parameters*

123 number = 1

number set value to 6 more...

Loop number times time show complicated version

Chicken move up number meters more...

Chicken move down number meters more...

Loop 123 index from 0 up to (but not including) number times incrementing by 1

Chicken move up index meters more...

Chicken move down index meters more...

Loop 123 index from 0 up to (but not including) number times incrementing by 2

Chicken move up index meters more...

Chicken move down index meters more...

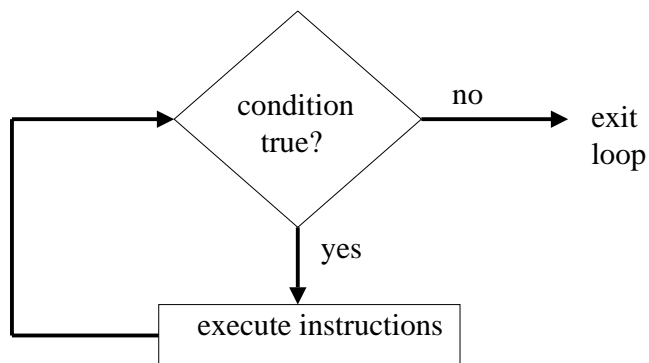
Repetition

- Sometimes don't know exactly how many times a set of instructions are repeated.
- Stopping is based on a condition
- Example:
 - Game of Chess, how many moves until win
 - Stop: when markers are in check mate position

Indefinite Repetition

- In programs where number of repetitions not known in advance, can use
 - While statement (today)
 - Recursion (Chap 8) - later

While statement



- While some condition is true
 - execute instructions

Example

- Common feature in popular “action films” is a chase scene
- Example: hungry shark chasing fleeing goldfish
 - Repeat: fish swim away from shark, and shark swim toward fish
 - Shark swim distance a little more than fish swim distance
 - Eventually, shark will catch up with fish and eat fish

Storyboard

World.chase

While goldfish more than .5 meters from shark

Do in order

shark point at goldfish

Do together

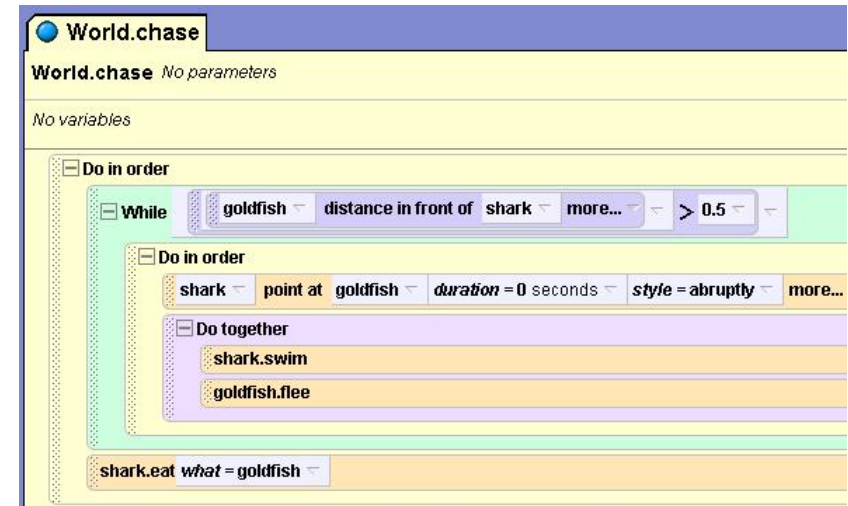
shark swim (toward goldfish)

goldfish flee (away from shark)

shark eat goldfish

shark.Swim, shark.eat and goldfish.flee in book

World.chase



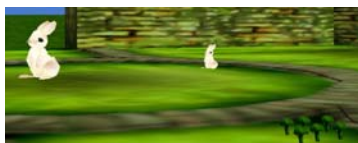
Shark will catch goldfish

- How do you know the shark will eventually catch the goldfish?
 - Shark always moves 0.4 meters towards goldfish
 - Goldfish moves randomly away from shark at most .2 meters
 - Shark will eventually catch up, **the loop will end**

General “Rule of Thumb”

- As a general rule, a While loop should be written so the loop will eventually end
 - Requires statements inside the loop change the conditions of the world such that condition for While eventually becomes false
- If While loop never ends
 - Infinite while loop

Practice – From Bunny eats Broccoli



- Replace with while loop, bunny hops until close to closestBroccoli (< 1 meter)

Practice – Eat Broccoli until all invisible

Assume method
bunny.hopToClosestBroccoliAndEatIt exists



ReWrite with while loop

- Eat broccoli (broc1, broc2, and broc3) until gone (all invisible)
- Assume at least one of the brocs is visible when start

Classwork – Can Cinderella and Handsome Prince ever get together?

- Objects
 - Cinderella, HandsomePrince, ballCinderella, ballPrince
- Setup
 - Cinderella and Prince 10 units each from middle
- Game
 - both Cinderella and Handsome prince move forward randomly
 - If both reach the middle, they connect!
 - If either reach the middle alone, then that one starts over

Setup

- They start by touching hands.
 - ballCinderella (invisible) is at same location as Cinderella
 - ballPrince (invisible) is at same location as Prince
- They both turn and walk 10 units
- They turn and face.



Play

Repeat forever

- They both move randomly forward
- Cinderella gets to the middle first alone.
 - She turns around
 - Goes forward 10 units
 - She turns back around
- They are both in middle at same time
 - Message displayed and stop



World. My first method

Note: variable, where is it set and used?

What is repeated a lot?

- Turn around
- Move forward 10 units
- Turn back around

Setup

```
world.setup
world.setup 123 unit
No variables
// Move cinderella and prince to starting positions at the same time
// They should move one unit at a time
Do together
  world.turnAround personToTurn = handsomePrince unit = unit
  world.turnAround personToTurn = cinderella unit = unit
```

turnAround

- Turn halfway around
- Move forward 10 units, 1 unit at a time
- Turn back around

```
world.turnAround obj personToTurn , 123 unit
No variables
// Turn person around, move forward 10 times in 1 unit increments, then turn back around
personToTurn turn left 0.5 revolutions duration = 0.5 seconds more...
Loop 10 times times show complicated version
  personToTurn move forward unit meters duration = 0.25 seconds more...
personToTurn turn left 0.5 revolutions duration = 0.5 seconds more...
```

randomInt

```
123 world.randomInt
world.randomInt No parameters
123 randint = 1
// returns integer from 1 to 3
randint set value to random number minimum = 1 maximum = 3.99 more...
randint set value to floor randint
Return randint
```

How to move forward randomly?

- Move forward randomly 1 to 3 units, but not past marker ball?
- Use randomInt

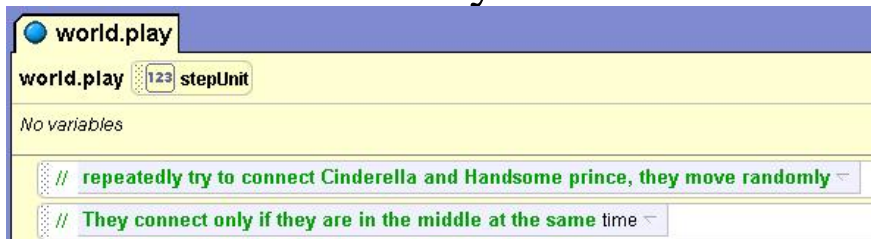


```
123 world.findRanInt
world.findRanInt Obj person , Obj marker create
create
// returns the smaller of a random integer from 1 to 3, and the number of steps to the middle
```

More Detailed Play

- Repeat over and over
 - Cinderella moves forward 1 to 3 units, but not past the ballCinderella, her spot in the middle
 - Prince moves forward 1 to 3 units, but not past the ballPrince, his spot in the middle
 - If both are in the middle, display a message, then stop.
 - If just one in the middle, then that one starts over

Play



```
world.play
world.play 123 stepUnit
No variables
// repeatedly try to connect Cinderella and Handsome prince, they move randomly
// They connect only if they are in the middle at the same time
```