Lecture 8: Control Structures

CS 51G
Spring 2018
Kim Bruce

Announcements

- Discuss Exercises 7.9.1, 8.5.3
- Demo light balloon lab

Defs/Vars vs. Parameters

- Defs vars used in an object to remember information.
- Parameters are used to transmit information to a method, typically from another object.
  - Go out of existence when method body completes unless saved to instance variable.

Match Statement

- Won't talk about in class. See last lecture notes. Probably won't be used ...

Taking Advantage of Speed

- So far haven't taken advantage of computer's speed.
- Building drawings with lots of repetition can be numbingly boring.
  - Following example draws RR tracks, one with individual clicks, the other with **while** loop
  - http://www.cs.pomona.edu/classes/cs051G/demos/Railroad/

While Loop

- Similar to if-then: but execute number of times dependent on condition:
  - while **{**cond**}** do {...}
  - http://www.cs.pomona.edu/classes/cs051G/demos/LaundryBasket/LaundryBasket.grace
- General form:

```
while {condition} do {
    do some work
    update some variable so next time do
        something a bit different
}
```

Animations

- Want continuous motion, rather than triggered by clicks
  - And want multiple things happening simultaneously
- Use animation library
  - See “pathetic pong”
    - http://www.cs.pomona.edu/classes/cs051G/demos/PatheticPong/

Animations

- import “animation” as animator
- Provides methods:
  - while {cond} pausing (delay) do {...}
  - while {cond} pausing (delay) do {...} finally {...}
  - plus others.
- Methods are asynchronous
  - Following statements continue while animation is going.
  - If want to delay statements to end of loop, do in finally

Code Quality

- Code is high quality if it is easy to understand and efficient.
- See CS 51 Style guidelines for readability & comments/formatting
- Some bad code is unnecessarily wordy & inefficient.

Bad Examples

```
if (dragging == true) then {
   doSomeThing
} elseif {clicking == false} then {
   doSomethingElse
}
```

can be simplified to

```
if (dragging) then {
   doSomeThing
} elseif {!clicking} then {
   doSomethingElse
}
```

Bad Examples

```
if (theSwatch.contains (point)) then {
   dragging := true
} else {
   dragging := false
}
```

can be simplified to

```
dragging := theSwatch.contains (point)
```

Questions?