Lecture 33: More Files in Grace

CS 51G
Spring 2018
Kim Bruce

Announcements

- Nibbles lab Friday

Nibbles

- Nibbles: *done*
- Position, Direction: *both done (in same file)*
  - *Look for translate method on position!*
- NibbleField: *Mostly done*
  - *cells in field are filled rects or empty (type FieldObject)*
- Snake: You write!
  - Use short-circuit boolean operators:
    - p && {q}

Writing Files

- Must import io library from standard Grace
  - import “io” as inout // use whatever name you like!
- Must open file for writing:
  - inout.open (path,”w") // object of type inout.FileStream
  - def myFile: inout.FileStream =
    inout.open (“Lec32/bookmarks.txt","w")
- Writing:
  - myFile.write(stuff) // where stuff is string
- When done: Must close or won’t write
  - myFile.close

Reading files

- Open for reading:
  - inout.open ("Lec32/bookmarks.txt", “r")
  - returns value of type inout.FileStream
- Can get path location (in files in left panel)
  - myReadFile.pathname
- read gives whole file, getline gives one line
- eof determines if at end of file
- Don’t forget to close!!

Example

- Find words of given length in dictionary:
  - http://www.cs.pomona.edu/classes/cs051G/demos/FindShortWords/FindShortWords.grace

Writing a big Program

- Identify the objects to be modeled in your program.
- For each type of object identified:
  (a) List its properties. (b) List its behaviors.
- Model properties with instance variables.
- Model behaviors with methods. Init. code too.
  - Focus on the method headers & parameters.
  - What will be the result of each method invocation?

Testing & Debugging

- Once find location of bug, relatively easy to fix.
- Test small (simple) pieces of code.
  - When put together, confident details work

Writing & Testing Simon

- Pop up window with buttons
- Does pressing one button work (no song)
- Create and play songs with 1 or 2 notes.
  - Add new note after play
- Start onMousePressed

Testing

- After each part, thoroughly test
  - If can't see what happens, add print statements to show what has changed.
  - Often best strategy is to write a test harness which only designed to test program pieces.
    - As I did for NibbleField!!
  - If can't find error, comment out parts until find where error arises

Nibbles Development

- Write missing methods (one at a time) in NibbleField and test with NibbleFieldTester
- Construct snake of length 1
  - Write shrink and stretch methods
  - Makes snake of length 1 move in straight line
  - Dies when hits edge
  - Steer snake using direction:=() method
  - Do same with snake of length 3

Nibbles Development

- Worry about eating!
  - Detect food before running over it.
  - Skip shrinking 3 times for each food item eaten.
    - I.e., if eat 2 in a row will stretch 6 times before shrinking.
  - If you decide to stretch an extra 3 times, you will NOT succeed because you must watch out for walls, etc while stretching (and might eat again)
    - *Follow this advice or waste many hours of your time!!!*
- Worry about dying by biting itself.
  - check out-of-bounds before checking biting itself

Questions?