CS 2334
Project 1: Elemental Finch Actions

Object Oriented Design

Class Hierarchies and Abstraction

Generating Proper Documentation

Plan for Today

- Using javadoc effectively
- Project 1 overview:
  - What to do
  - When is it due
  - What to turn in
- Begin configuring your computers for the Finches (these will be available for check out early next week)
- Begin design of your project

Documentation

Javadoc will include some of your in-line documentation into the html files that are generated

- See “Documentation Requirements.pdf” for details of use (in projects/general)

An Example: Top of Your Java Source File

```
/**
 * @author Andrew H. Fagg
 * @version 1.0
 *
 *<P>
 *
 *  Class represents the action of
 *    producing a tone.
 */

public class FinchTone extends FinchAction {
:
```

An Example: Function Documentation

```
/**
 * Primary constructor
 *
 * @param name    String describing
 *                the name of the action.
 * @param durationDuration of the action
 * @param frequency    Frequency of the
 *                tone to be generator
 */
FinchTone(String name, int duration, int
frequency){
```

An Example: Function Documentation

```
/**
 * Accessor: Frequency of the
 *tone to be generated
 *
 * @return The tone frequency in Hz
 */
double getFrequency() {
       :
       :
```

Project 1 Components

- FinchAction: superclass for general actions
- Subclasses: FinchJog, FinchTone, and FinchBeakFlash
  - For each: execute() method performs the action with the Finch
- FinchActionList: creating and manipulating a list of actions
- Reading an action list from a file
- Allowing a user to specify which actions are displayed and executed

Example File

```
forward_short      JOG      2000    10.0    10.0
forward_long       JOG      5000    20.0    20.0
forward_left       JOG      2000    10.0    20.0
forward_right      JOG      2000    20.0    10.0
dance              RGB      100     255     30      0       0
dance              JOG      2000    15.0    15.0
dance              RGB      100 0   30      255     1
dance              SOUND    1000    200
dance              JOG      4000    -20.0   20.0
dance              SOUND    500     400
dance              RGB      100     0       255     20      0
```

User Interaction

- User can specify a certain name or “All”
- Actions with matching name will be displayed and executed in order

Finch Support
(Do these steps today)

Download: FinchSoftwarev3_1.zip

- Note the update on 9/9/9
- Extract the contents of the zip file into your cs2334 directory (or wherever you are keeping your class work)

See Finch Driver Setup.pdf (projects/general)

- Configure your computer to talk to the Finch

Creating a Finch Project

- Go to File->New->Java Project
- Name it something like Finch
- Choose "Create project from existing source" and point it at the Source directory
- Finish
- Right click on the project and go to properties
- In the Java Build Path->Libraries tab, hit "Add External Jar" and select the finch.jar listed
- Hit Ok

Main method not found error

- So far, the only known solution is to go to File->Switch Workspace->Other, create a Finch workspace, and try again

Example Programs

<install dir>/FinchSoftwarev03/SourceCode/ contains a set of example programs that demonstrate the use of the Finch object

Referencing the Finch Support

At the top of your source java file(s) – may need to be included in several files:

```
import finch.*;
```

Opening the connection to the Finch:

```
Finch myFinch = new Finch();
```

Used only at the end of the main method:

```
 myFinch.quit();
```

Design: On Paper

- Cover page: Group members, work contributed, and outside citations
- The UML on engineering paper: should be neatly arranged and easily readable

Hardcopy Due: Sept 17th @ 5:00pm

Design: On Computer

Submit **project1_design**.zip

- Documentation and stubs only (no “working” code)
  - Class variables and method prototypes
- Javadocs-produced html made by using proper documentation in the source files

Electronic Copy Due: Sept 17th @ 5:00pm

Final: On Paper

Update from stage 1

- Cover page: Group members, work contributed, and outside citations
- The UML on engineering paper: should be neatly arranged and easily readable

Hardcopy Due: Sept 24th @ 5:00pm

Final: On Computer

Submit project1.zip:

- Complete working code
- Proper documentation (internal and Javadoc)

Electronic Copy Due: Sept 24th @ 5:00pm

And for the rest of lab

Form groups and start your UML…

Updates to handout

- Milestone 1, step 2: menu option may be called “Run dialog as” or “Configuration options”