Maze Program

TOPICS

• Graphical Programming
• Using Classes (Objects)
• Multiple Files (Eclipse)
• Maze Logistics

Graphical Programming (1)

Graphical Programming (2)

▪ No, we’re not going to show the code for the Puzzle program! (yet)
▪ 150 lines of Java Swing code
▪ What kinds of things does it do?
  ▪ Set window size, title, and location
  ▪ Setup a frame and panel, add buttons
  ▪ Read a photo and extract parts of it
  ▪ Listen for mouse and keyboard events

Graphical Programming (3)

▪ But, why not have you write code that controls a graphical program?
▪ Maze program:
  ▪ Move student around a maze
  ▪ ~260 lines of graphical programming
  ▪ You write the main method
  ▪ You instantiate the Maze
  ▪ You control the character
Maze Program: Objectives

- Use objects (classes) developed by someone else
  - Create an instance of a class
  - Call methods on the object
  - For example: graphical programming!
- Multiple source files
- Resource and data files

Maze Program: Initial Code

```java
// Create maze
String fileName = args[0];
Maze maze = new Maze(fileName);
System.out.println("Maze name: " + fileName);

// Get dimensions
mazeWidth = maze.getWidth();
mazeHeight = maze.getHeight();
System.out.println("Maze width: " + mazeWidth);
System.out.println("Maze height: " + mazeHeight);
```

Maze Program: Interface

```java
// Moving character
maze.moveRight(); // move right, no return value
maze.moveLeft(); // move left, no return value
maze.moveUp(); // move up, no return value
maze.moveDown(); // move down, no return value

// Getting position
maze.getColumn(); // get column of character
maze.getRow(); // get row of character

// Found Haku?
if (maze.foundHaku()) …
```

Maze Program: Output

```
Maze name: maze1.txt
Maze width: 6
Maze height: 6
Moved to row 0, column 1
Moved to row 1, column 1
…
Chihiro found Haku, congratulations!

Error Messages
Chihiro went outside the maze!
Chihiro ran into the witch Yubaba!
```
Maze Program: Algorithm

- Rules for moving Chihiro around maze:
  - If not on edge, move up to edge
  - Then move counter-clockwise around edge
  - Check for Haku every time you move
  - Does not require 16 loops!

Guaranteed to find Haku, will never meet Yubaba.

Maze Program: Method

```java
// Returns true if on any edge, false otherwise
public static boolean onEdge();
```

Method not required, but very useful!

Maze Program: Setup

- ~/workspace/P6
  - Chihiro.png
  - Haku.png
  - Yubaba.png
  - Success.png
  - Maze*.txt
- ~/workspace/P6/src
  - Maze.java
  - P6.java (write this!)