CS 10:
Problem solving via Object Oriented Programming
Winter 2017

Tim Pierson
260 (255) Sudikoff

Agenda

1. Multiple blobs: lists

2. Images

3. Animated images

Java provides an ArrayList that can hold a collection of multiple items

ArrayList

- Stores list of objects in order
- Variable length – don't specify size; can grow (unlike C array, but like Python list)
- Random access – get item by index (starting at 0)
- Must be imported from java.util (Eclipse can help!)
- Provides methods to add or remove elements
- Specify what type of object it holds in angle brackets <> (e.g., ArrayList<Blob> or ArrayList<String>)
- ArrayList called a *generic* container because it can hold any type of object
- All objects must be of same type (unlike Python)

ArrayList methods provide a consistent means of interaction

ArrayList methods

- *add (E elmt)* – appends element *elmt* to end of list

- *add (int index, E elmt)* – inserts element *elmt* at position *index*

- *get (int index)* – returns the element at position *index*

- *remove (int index)* – removes (and returns) the element at position *index*

- *set(int index, E elmt)* – sets item at position *index* to *elmt*

- *size()* – returns the number of elements in the ArrayList

- Others on Oracle website

4

Show me some code!

BlobsDriver.java
- "is-a" nature of classes allows multiple subclasses to be in the same ArrayList

BlobsGUI.java
- ArrayList instance variable can hold many blobs
- List created in constructor, initially empty, don't forget to call new()
- New draw() method to draw each blob using for-each loop
- Same for handleTimer() – loop over each blob and call step() for each blob
- handleMousePress() also looks over each blob

Agenda

1. Multiple blobs: lists

2. Images

3. Animated images

Images can be drawn instead of Blob's ovals

SimleGUI.java
- main() loads an image called "simley.png" into BufferedImage class
- Creates new SimleGUI class, passing BufferedImage to constructor
- draw() displays image at coordinates (0,0) using drawImage()

WanderingImage.java
- Extends Wanderer
- Constructor takes BufferedImage
- draw() shows image centered at (x,y) instead of oval

BlobsGUI2.java
- Add BufferedImage blobImage, load in constructor
- Add WanderingImage type to collection of blobs
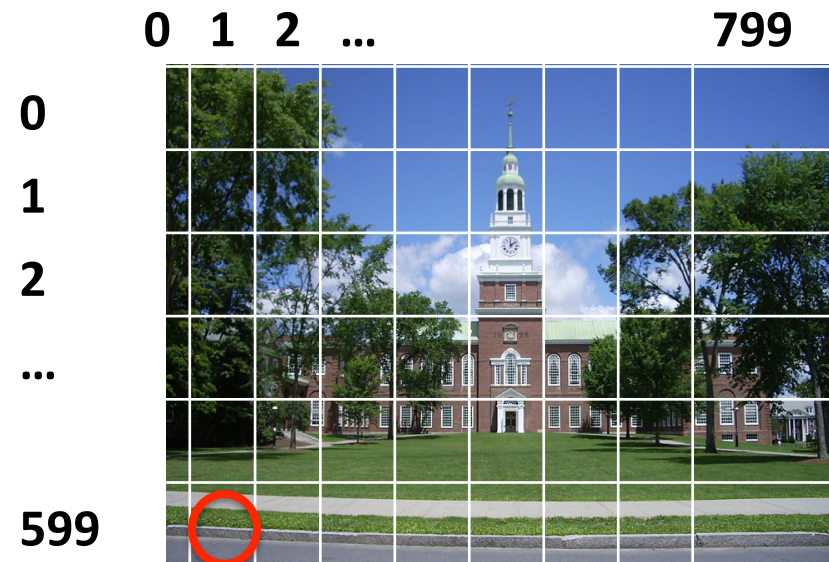
Agenda

1. Multiple blobs: lists

2. Images

3. Animated images

Images are made up of pixels, each with a (x,y) location and a color

NOTE Y axis counts downward!

800 x 600 image

Methods:
getRGB(x,y)
setColor(Color)
setRGB(Color)

Each pixel color is an integer where bits:
16-23 = red component
8-15 = green component
0-7 = blue component

New Blob type WanderingPixel draws a blob with a given Color

WanderingPixel.java
- Inherits from Wander
- Constructor takes Color
- draw()
  - First set drawing color to be this blob's color
  - Draw oval of that color

Now we can animate the image

AnimateImage.java
- Image loaded in main() side call to constructor
- Constructor
  - Chooses random (x,y) locations
  - Picks up color from image at that location
  - Adds new WanderingPixel with color from (x,y) location and random radius
- draw() handles drawing multiple blobs
- handleTimer() picks a random blob to step, does this numToMove times

Another option is for the blobs to trace out the image

PaintedImage.java
- As blobs move around, they leave a trail, painting the color from the underlying image
- Constructor
  - Save original images
  - Create a new blank image called "result" size of original
  - Create a bunch of Wanders at random locations
- draw() plots blobs on "result" image (not original)
- handleTimer()
  - Picks blob at random (does this numToMove times)
  - Gets (x,y) location of blob
  - Picks up Color from original image with getRGB(x,y)
  - Sets color on result image (if on screen)
  - Steps blob and repaints