

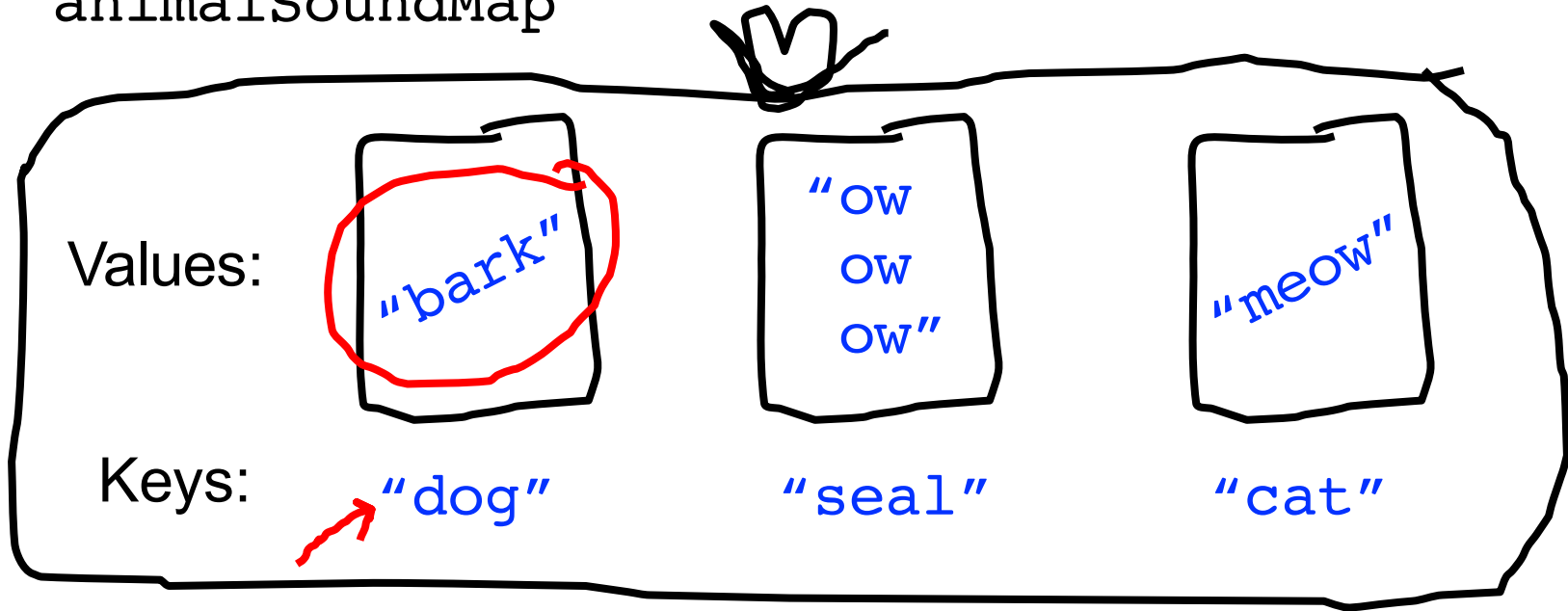
Advanced Maps

Chris Piech

CS106A, Stanford University

My First Map

animalSoundMap



```
// 1. Make a new map  
HashMap<String, String> animalSoundMap =  
    new HashMap<String, String>();
```

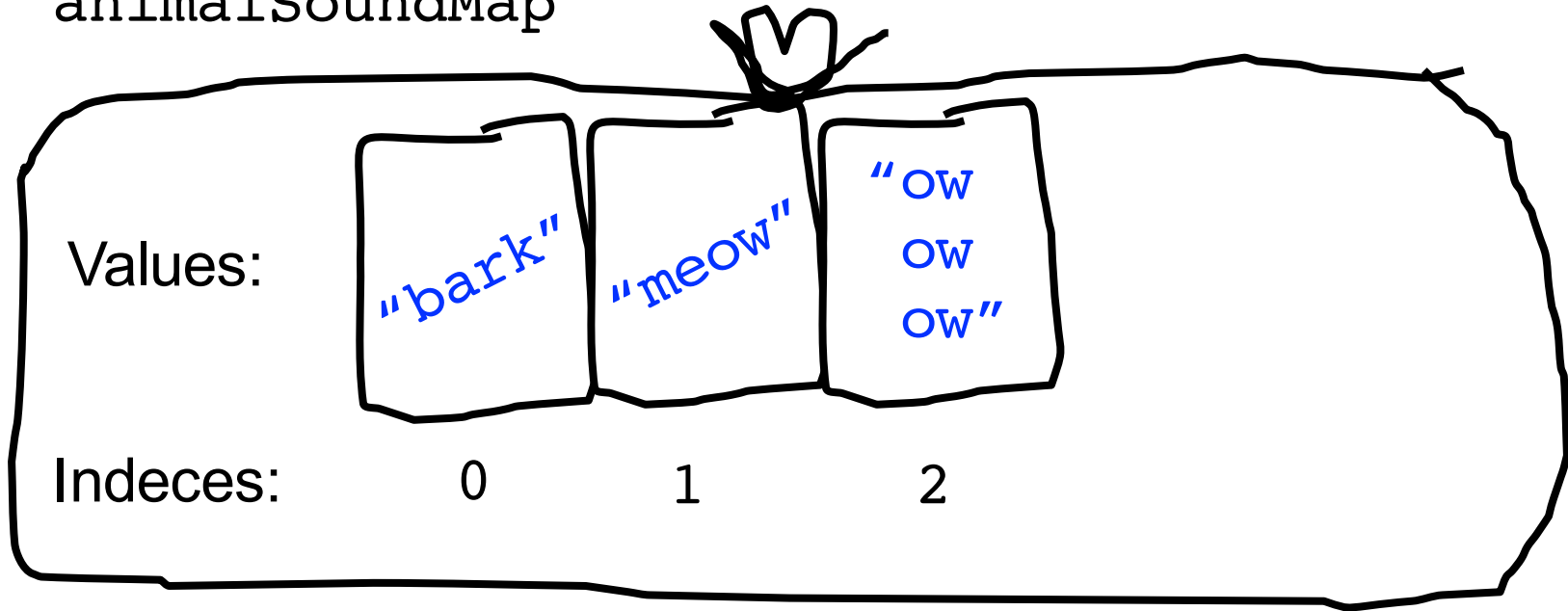
```
// 2. Put things into the map  
animalSoundMap.put("dog", "woof");  
animalSoundMap.put("cat", "meow");  
animalSoundMap.put("seal", "ow ow ow");
```

```
// 3. Get things out of the map  
animalSoundMap.get("dog"); // "woof"  
animalSoundMap.get("fox"); // null
```



My First Map

animalSoundMap



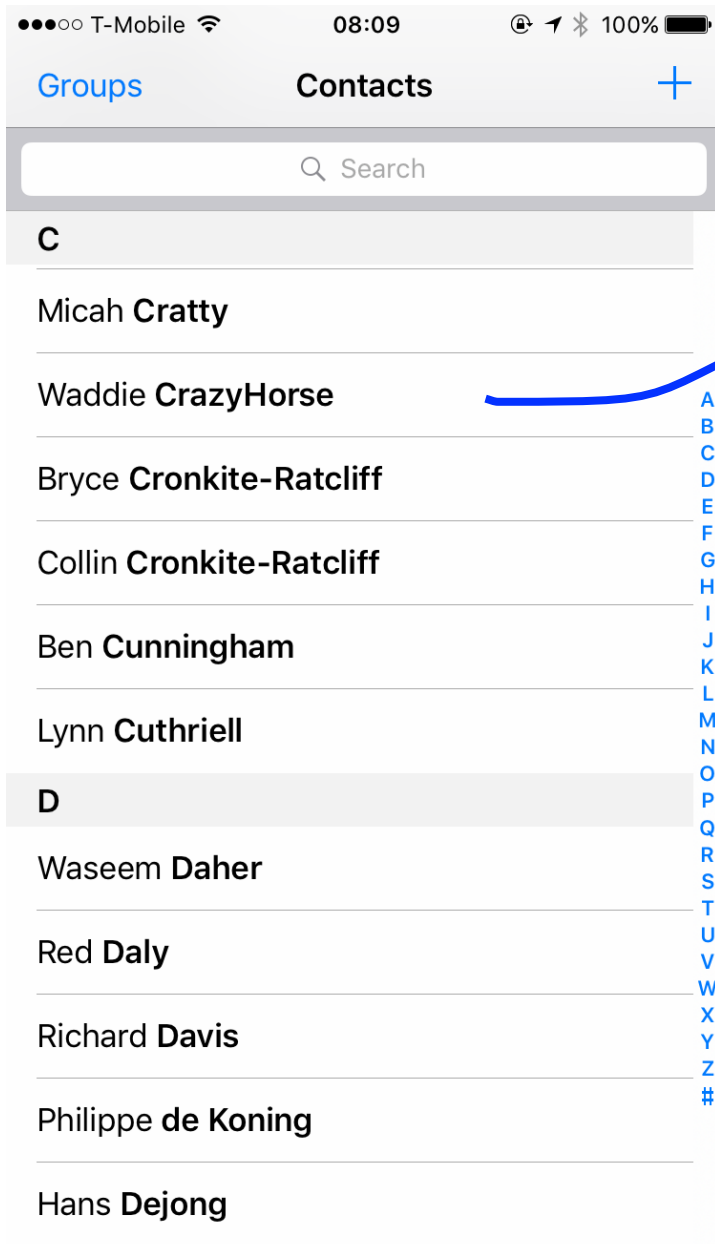
```
// 1. Make a new list
ArrayList<String> animalSounds =
    new ArrayList<String>();
```

```
// 2. Put things into the list
animalSoundMap.add("woof");
animalSoundMap.add("meow");
animalSoundMap.add("ow ow ow");
```

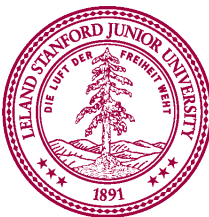
```
// 3. Get things out of the list
animalSoundMap.get(0); // "woof"
```



Phone Book



6701678



Phone Book

```
PhoneBookSolution
Enter command (printAll, add, lookup): add
Key: Waddie
Phone number: 1234567
Added Waddie

Enter command (printAll, add, lookup): lookup
Key: Waddie
Waddie: 1234567

Enter command (printAll, add, lookup): printAll
Chris: 8666586
Nick: 5551212
Jenny: 8675309
Waddie: 1234567

Enter command (printAll, add, lookup): |
```



HashMaps on one slide

1. Make a HashMap

```
HashMap<KeyType, ValueType> myMap =  
    new HashMap<KeyType, ValueType>();
```

2. Put and get values into a map

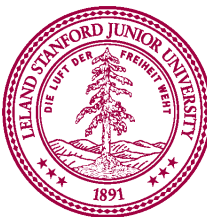
```
myMap.put(key, value);  
myMap.get(key) // returns the corresponding value
```

3. Some useful other methods

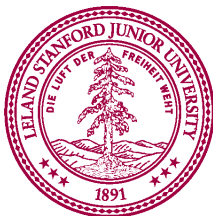
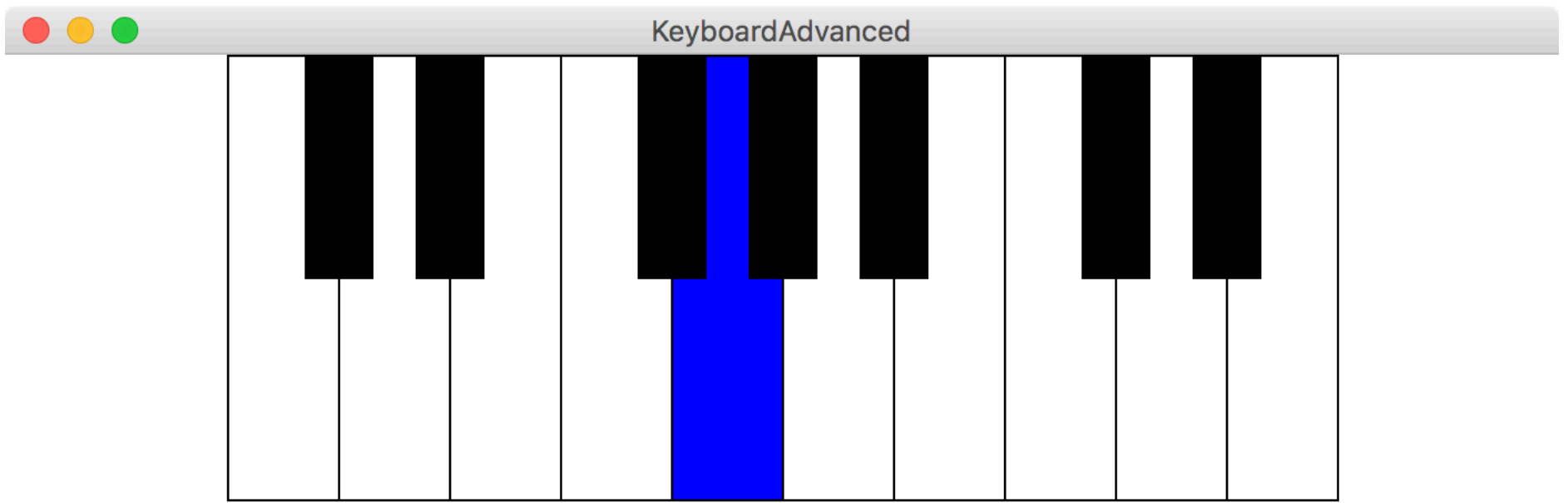
```
int size = myMap.size();  
myMap.containsKey(key); // returns true or false if key is in map  
myMap.keySet();  
myMap.remove(key); // make like a tree and leave!
```

4. Iterate using a foreach loop

```
for(keyType key : myMap.keySet()){ // not ordered  
    myMap.get(key); // do something with the key/value pair  
}
```

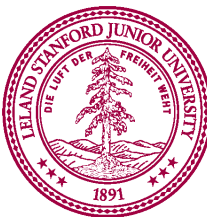


Make a keyboard



Aside: AudioClips

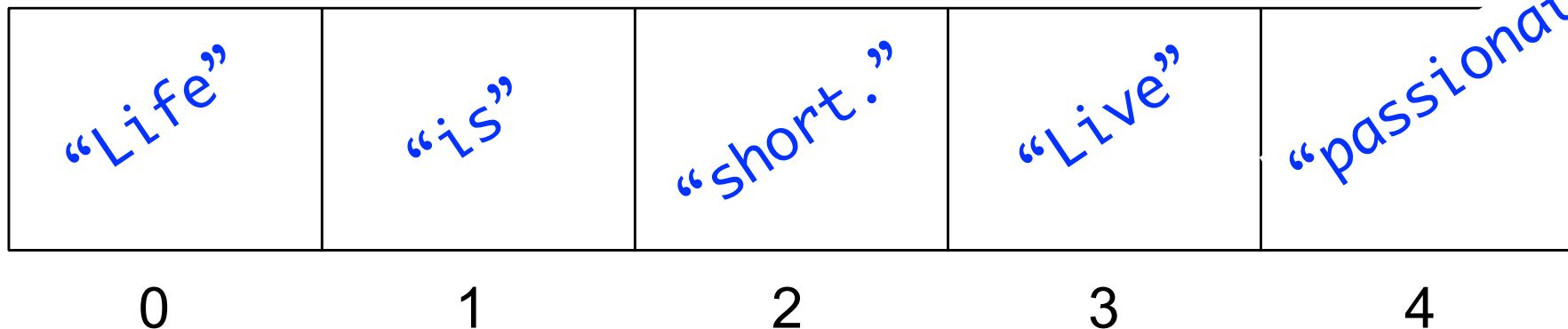
```
AudioClip soundFile =  
    MediaTools.loadAudioClip(fileName);  
  
soundFile.play();
```



Aside: Split

```
String str  
    = "Life is short. Live passionately."  
String[] words = str.split(" ");
```

words



Aside: Split

```
String str  
    = "Life is short. Live passionately."  
String[] words = str.split(".")
```

words

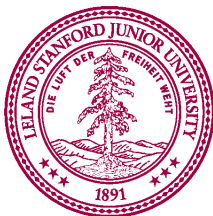
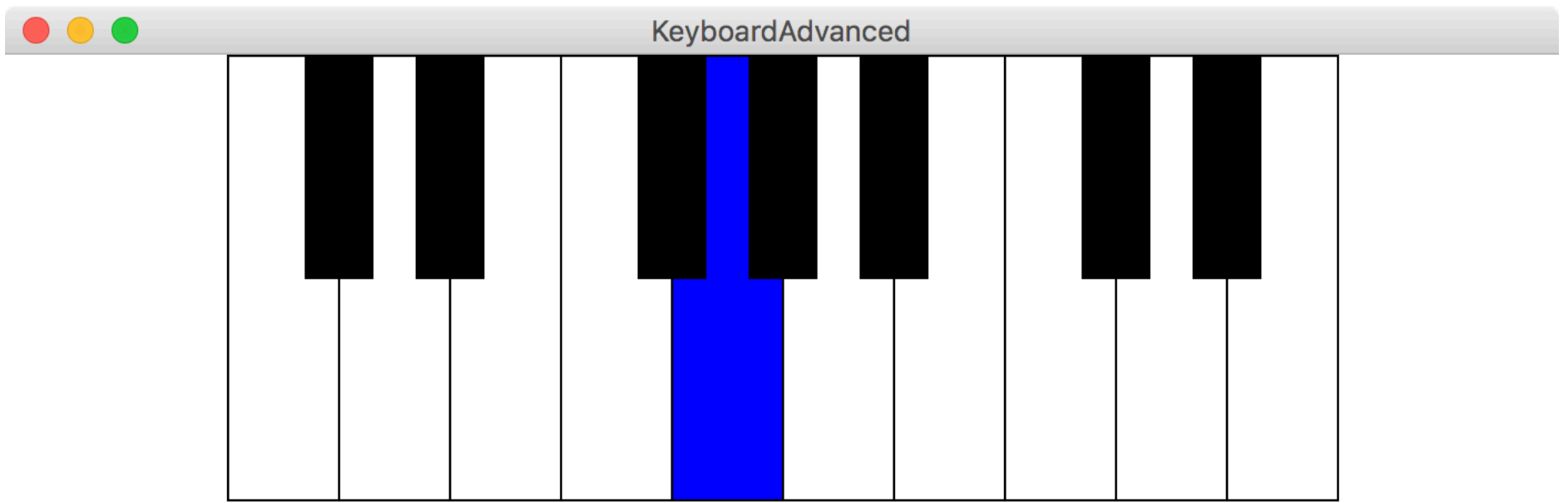
| | |
|-----------------|---------------------|
| "Life is short" | "Live passionately" |
|-----------------|---------------------|

0

1



Make a keyboard



Why is this so fast?



mantis shrimp colors



All

Videos

Shopping

Images

News

More

Settings

Tools

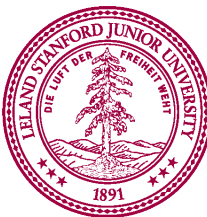
About 1,870,000 results (0.54 seconds)

Humans and many other primates have three; some birds and reptiles have four photoreceptors. Certain butterflies can even have six. But the mantis shrimp has **12** different types of photoreceptors in their eyes – and scientists haven't understood why until now. Jan 27, 2014



[Study Offers Insights into Unique Color Vision of Mantis Shrimp ...](#)

www.sci-news.com/biology/science-color-vision-mantis-shrimp-01719.html

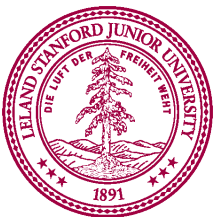


Why is this so fast?



```
int hash(string key);
```

* Learn more in CS106B



Why is this so fast?



mantis shrimp colors



All

Videos

Shopping

Images

News

More

Settings

Tools

About 1,870,000 results (0.54 seconds)

Humans and many other primates have three; some birds and reptiles have four photoreceptors. Certain butterflies can even have six. But the mantis shrimp have different types of photoreceptors in their eyes that scientists haven't understood why until July 7, 2014

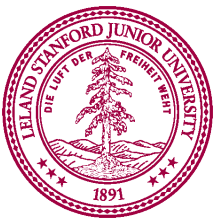
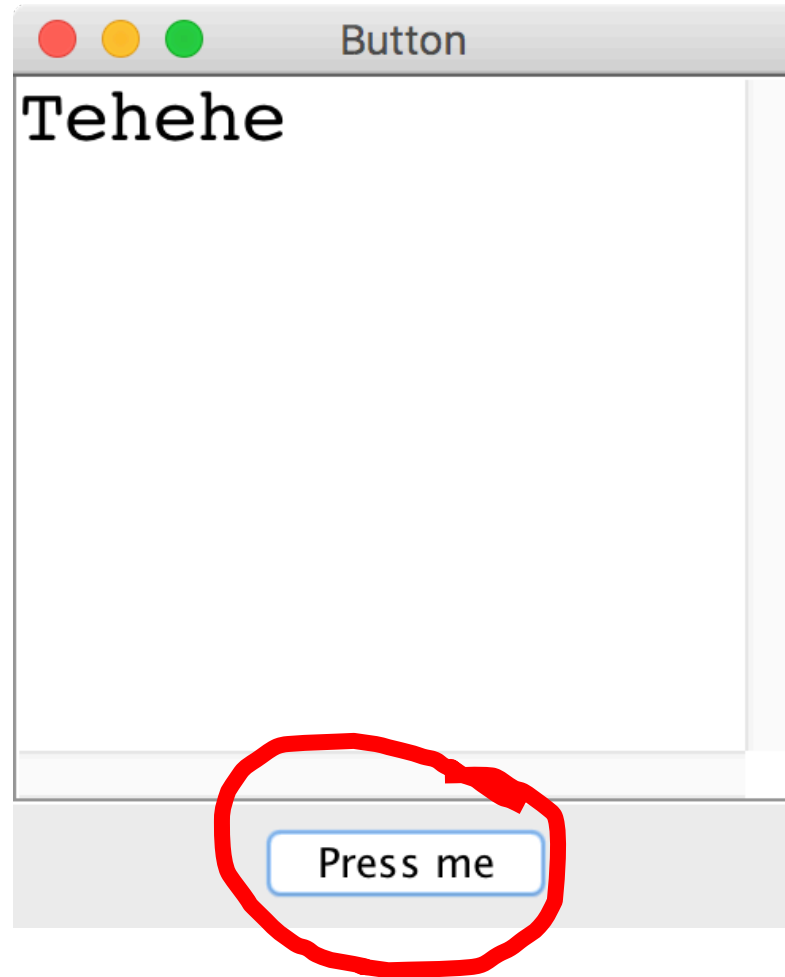


ers Insights into Unique Color Vision of Mantis Shrimp ...
www.sciencemag.com/biology/science-color-vision-mantis-shrimp-01719.html



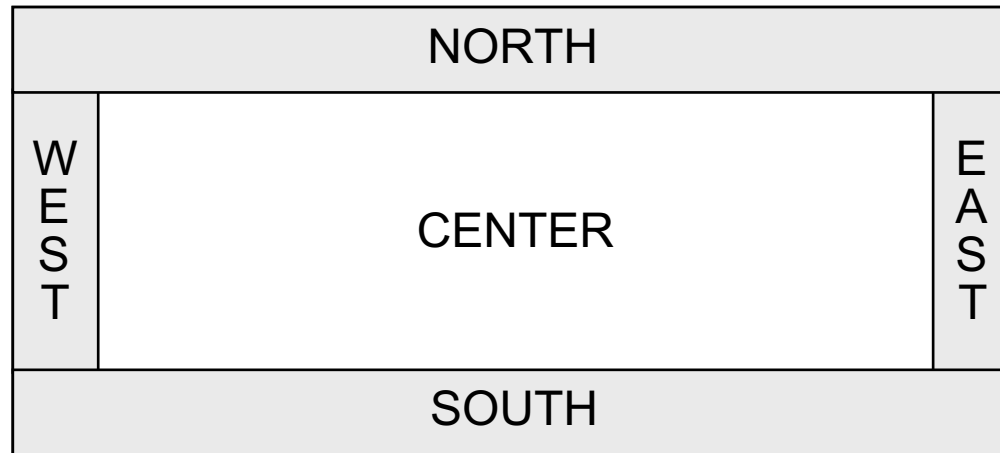
Interactors

Button

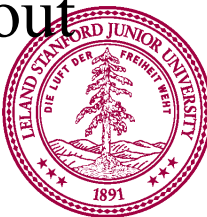


Adding Interactors

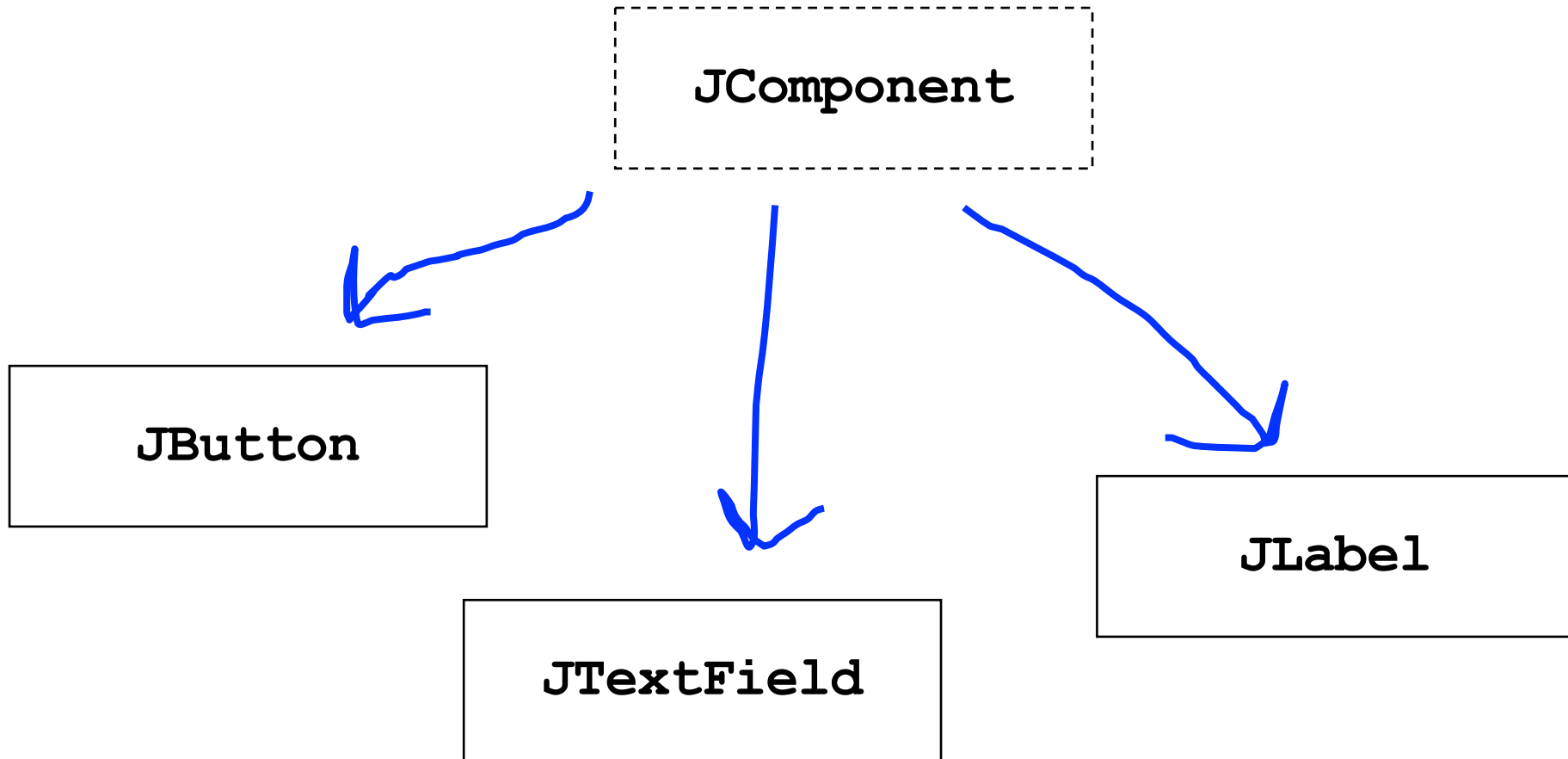
- When you create an instance of any **Program** subclass, Java divides the window area into five regions as follows:



- The **CENTER** region is typically where the action takes place. A **ConsoleProgram** adds a console to the **CENTER** region, and a **GraphicsProgram** puts a **GCanvas** there.
- The other regions are visible only if you add an interactor to them. The examples in the text use the **SOUTH** region as a control strip containing a set of interactors, which are laid out from left to right in the order in which they were added.



JComponents



JButton

JButton

```
JButton button = new JButton("Press me");
```



JButton

Button Text

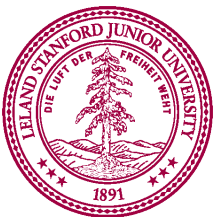
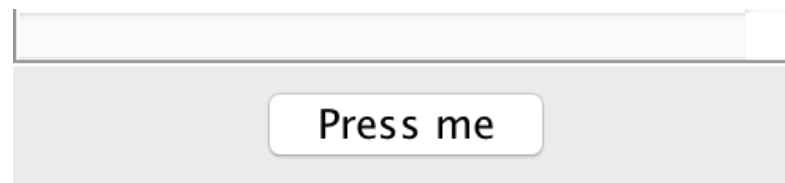


```
.JButton button = new JButton("Press me");
```



JButton

```
JButton button = new JButton("Press me");  
add(button, SOUTH);
```



JButton

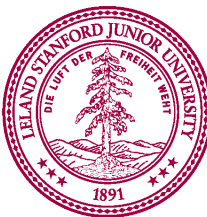
```
public void actionPerformed(ActionEvent e) {  
    println(e.getActionCommand());  
}
```



JLabel

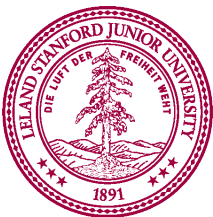
JLabel

```
JLabel label = new JLabel("Hi");
```



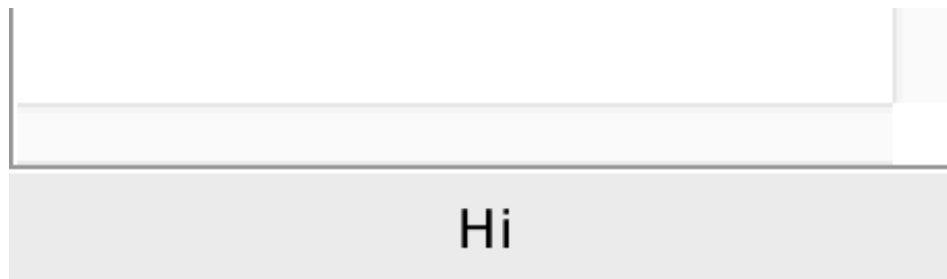
JLabel

```
JLabel label = new JLabel("Hi");
```



JLabel

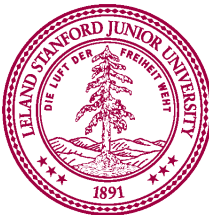
```
JLabel label = new JLabel("Hi");  
add(label, SOUTH);
```



JTextField

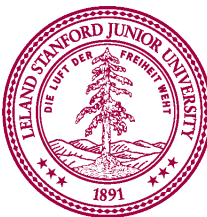
JTextField

```
JTextField field = new JTextField(10);  
add(field, SOUTH);  
field.getText(); // returns string in field  
field.setText("Marry me");
```



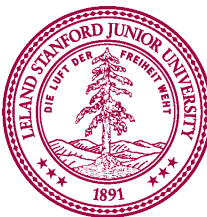
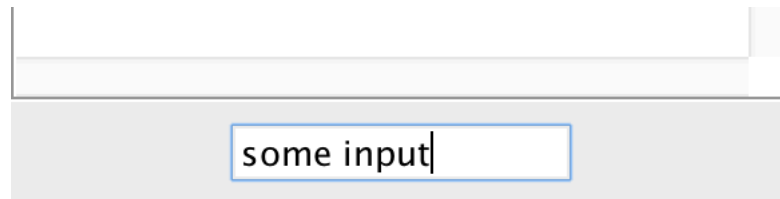
JTextField

```
JTextField field = new JTextField(10);  
add(field, SOUTH);  
field.getText(); // returns string in field  
field.setText("Marry me");
```



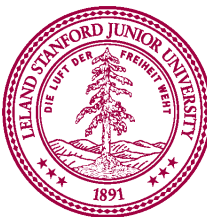
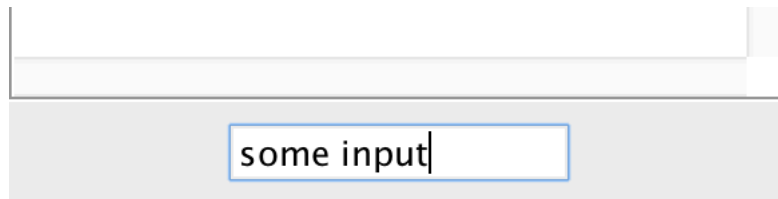
JTextField

```
JTextField field = new JTextField(10);  
add(field, SOUTH);  
field.getText(); // returns string in field  
field.setText("Marry me");
```



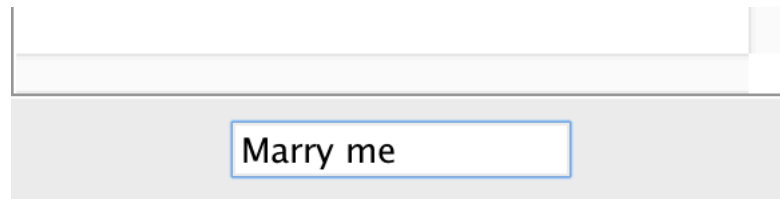
JTextField

```
JTextField field = new JTextField(10);  
add(field, SOUTH);  
field.getText(); // returns "some input"  
field.setText("Marry me");
```

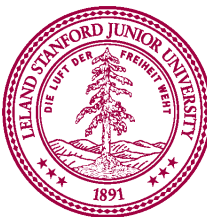


JTextField

```
JTextField field = new JTextField(10);  
add(field, SOUTH);  
field.getText(); // returns "some input"  
field.setText("Marry me");
```



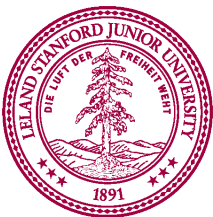
*in honor of Carlos, my freshman and sophomore roommate who just proposed



Recall the Dancing Children

Normal Program

Run Method

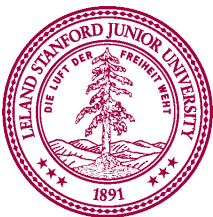


Normal Program

Run Method



```
public void run() {  
    for(int i = 0; i < N_DRIBBLES; i++) {  
        dropOneDrizzle();  
    }  
}
```

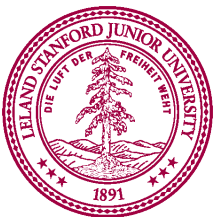


Normal Program

Run Method

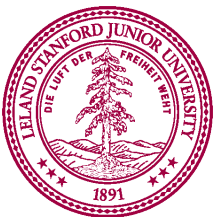


```
public void run() {  
    for(int i = 0; i < N_DRIBBLES; i++) {  
        dropOneDribble();  
    }  
}
```



Normal Program

Run Method



New Listener Characters

Action Listener



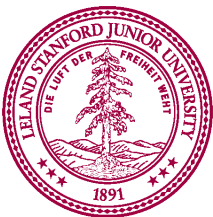
Action Performed



Program Starts Running

Run Method

Action Performed



Add Action Listeners

Run Method

Action Performed

Action Listener



```
addActionListeners();
```

© 2009, 2010, Stanford University



Program Runs as Usual

Run Method

Action Performed

Action Listener



Button Clicked!

Run Method

Action Performed

Action Listener

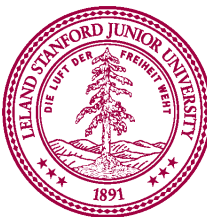


Calls Action Performed Method

Run Method

Action Performed

Action Listener



When done, Run continues.

Run Method

Action Performed

Action Listener



Keeps Doing Its Thing...

Run Method

Action Performed

Action Listener



Button Clicked!

Run Method

Action Performed

Action Listener

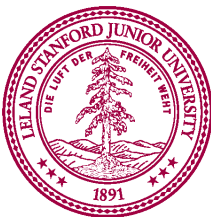


Calls Action Performed Method

Run Method

Action Performed

Action Listener

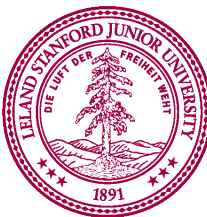


When done, Run continues.

Run Method

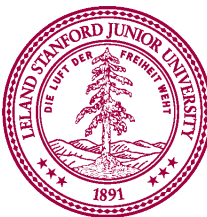
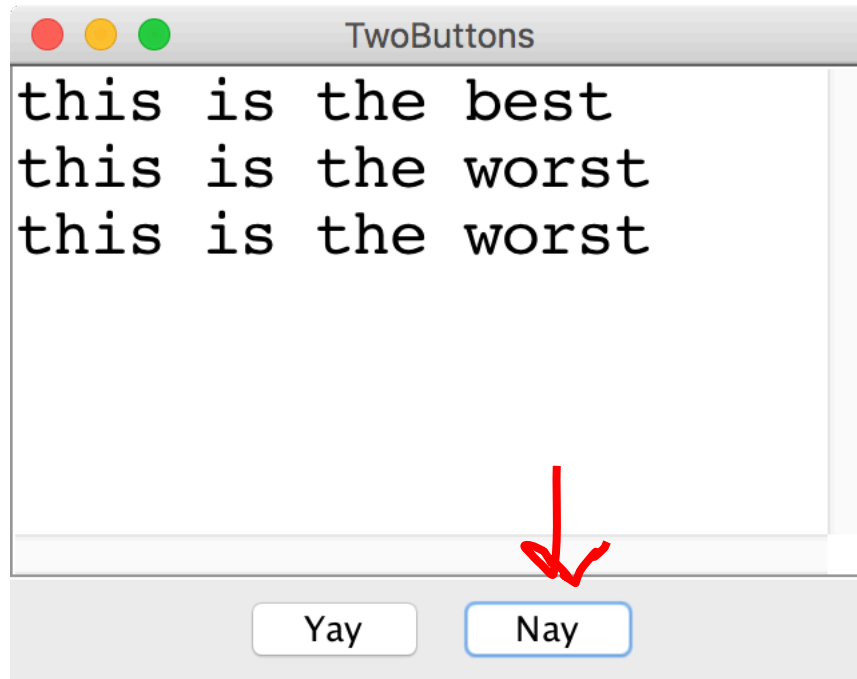
Action Performed

Action Listener



Recall the Dancing Children

Two Buttons



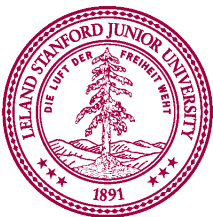
Text Field



Something awesome

*thanks Keith for the idea

The XKCD Color Survey



The XKCD Color Survey

- Volunteers (online) were shown a randomly-chosen color and asked to name the color.
- The result is (after filtering) about 2.8 million RGB triplets and their names.
- What do people think the colors are?



The File Format

color-name,

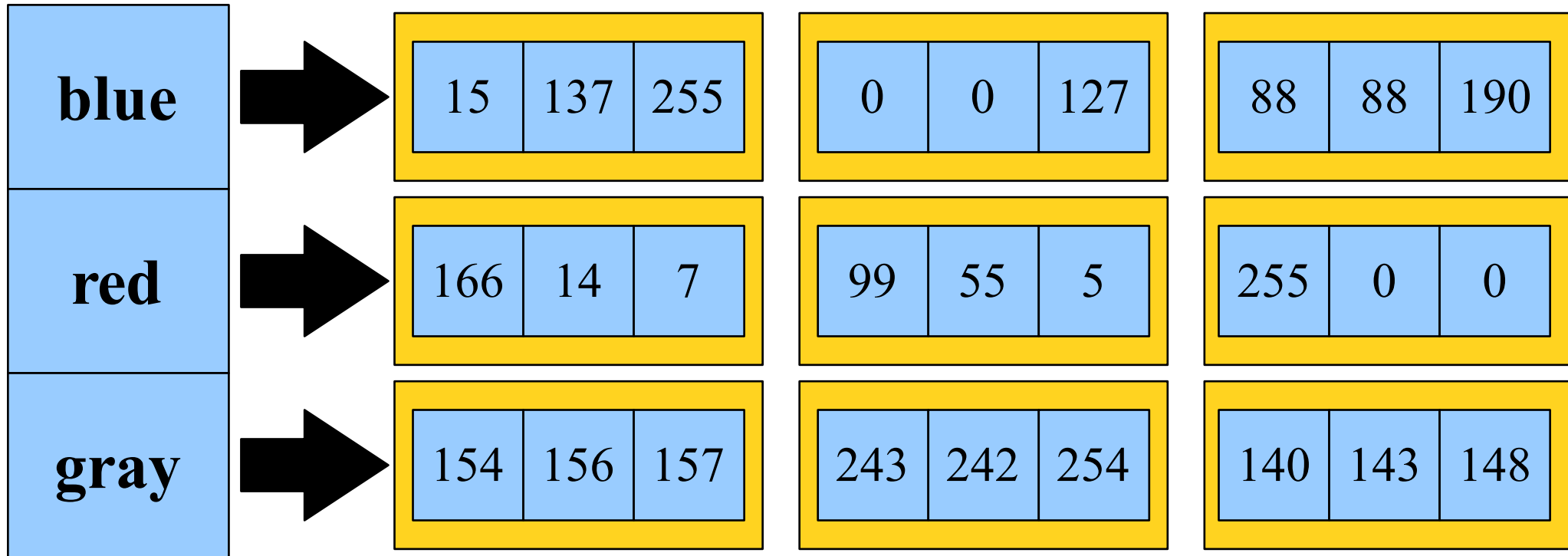
red,

green,

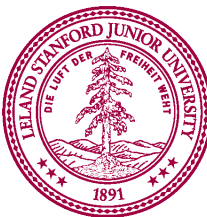
blue



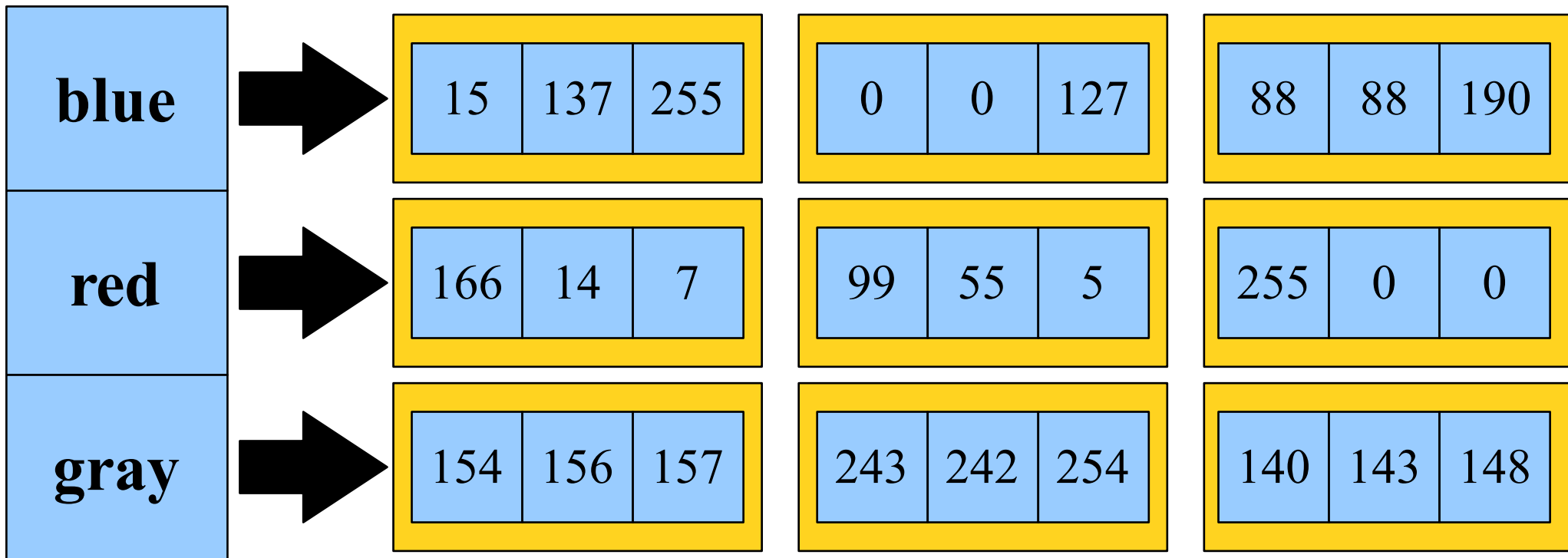
How to Structure Data



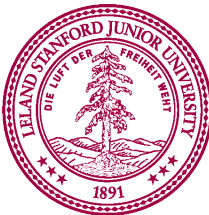
***associate each color name
with a list of colors***



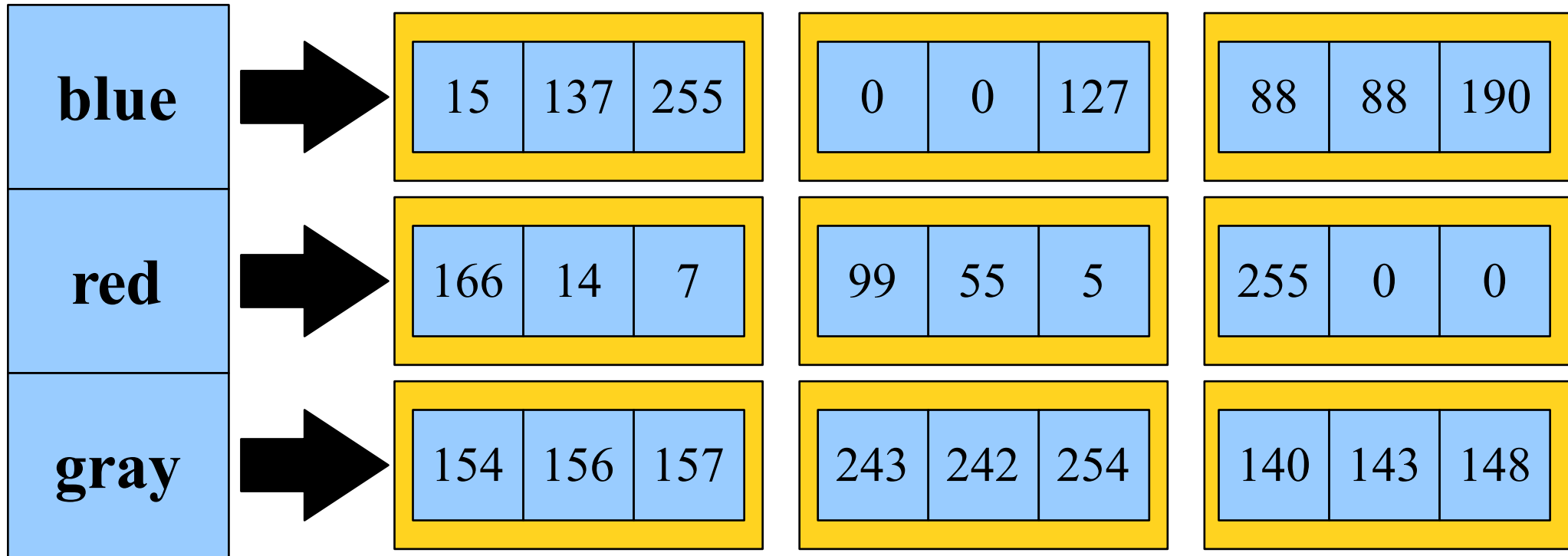
How to Structure Data



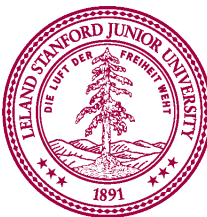
HashMap<*color name* , *list of colors*>



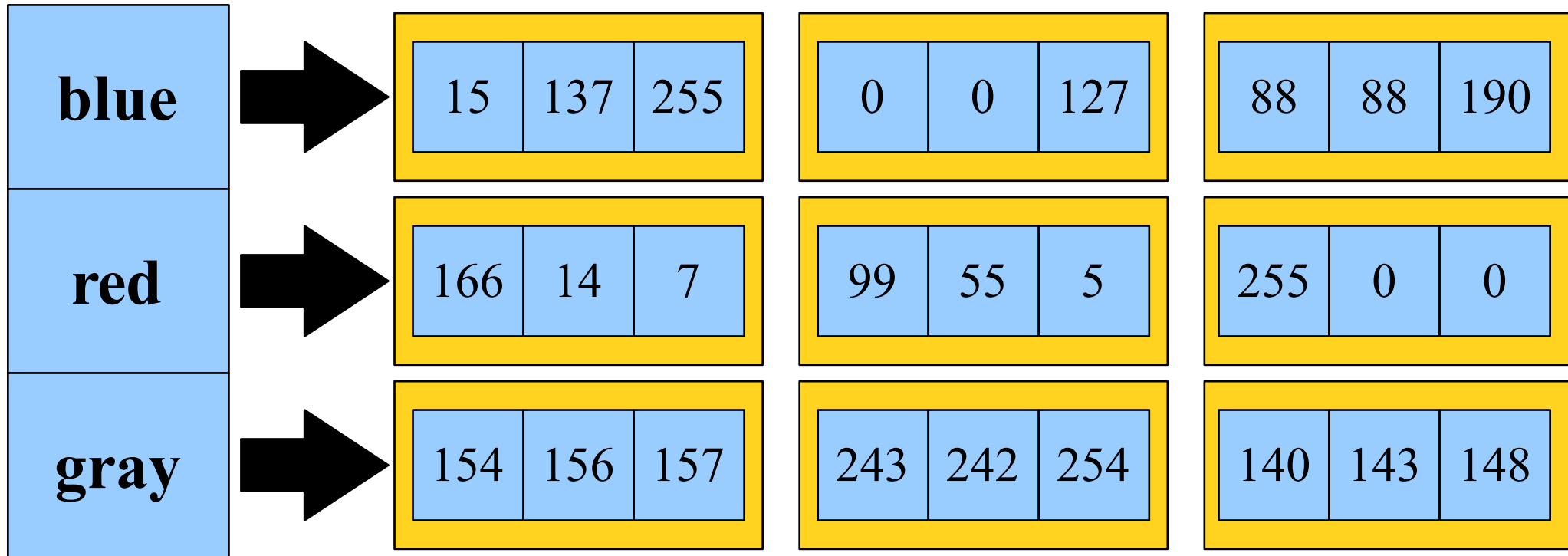
How to Structure Data



HashMap<String, *list of colors*>



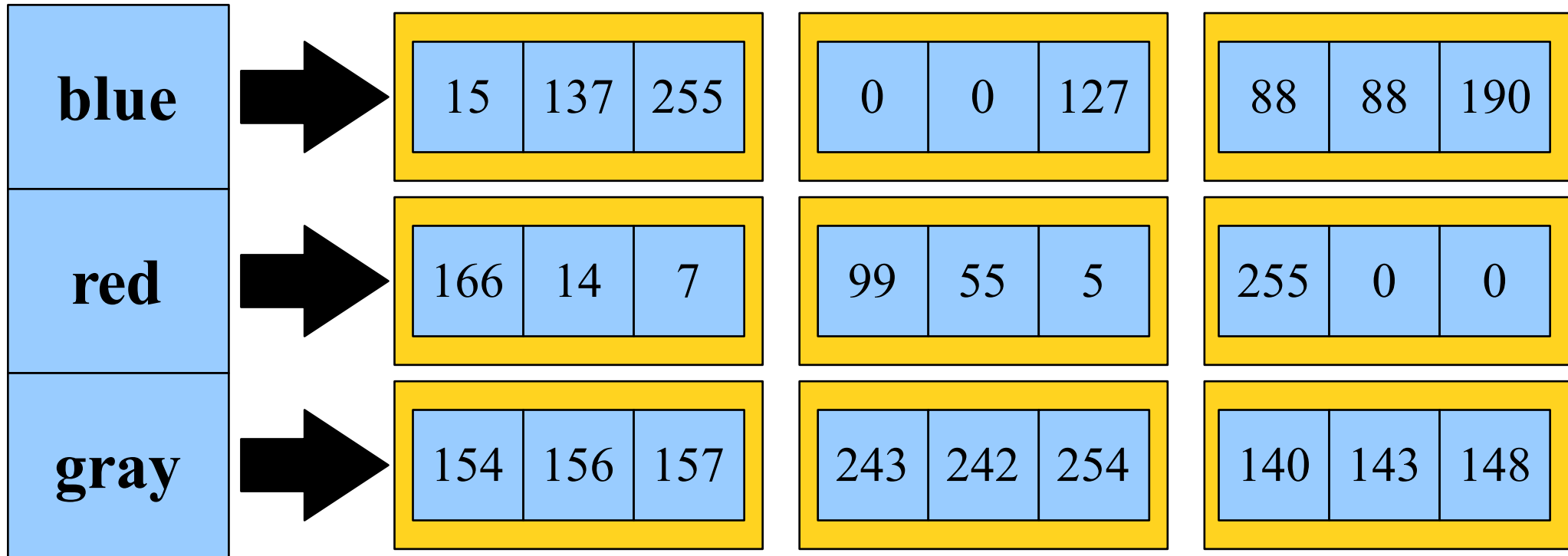
How to Structure Data



HashMap<String, ArrayList<color>>



How to Structure Data

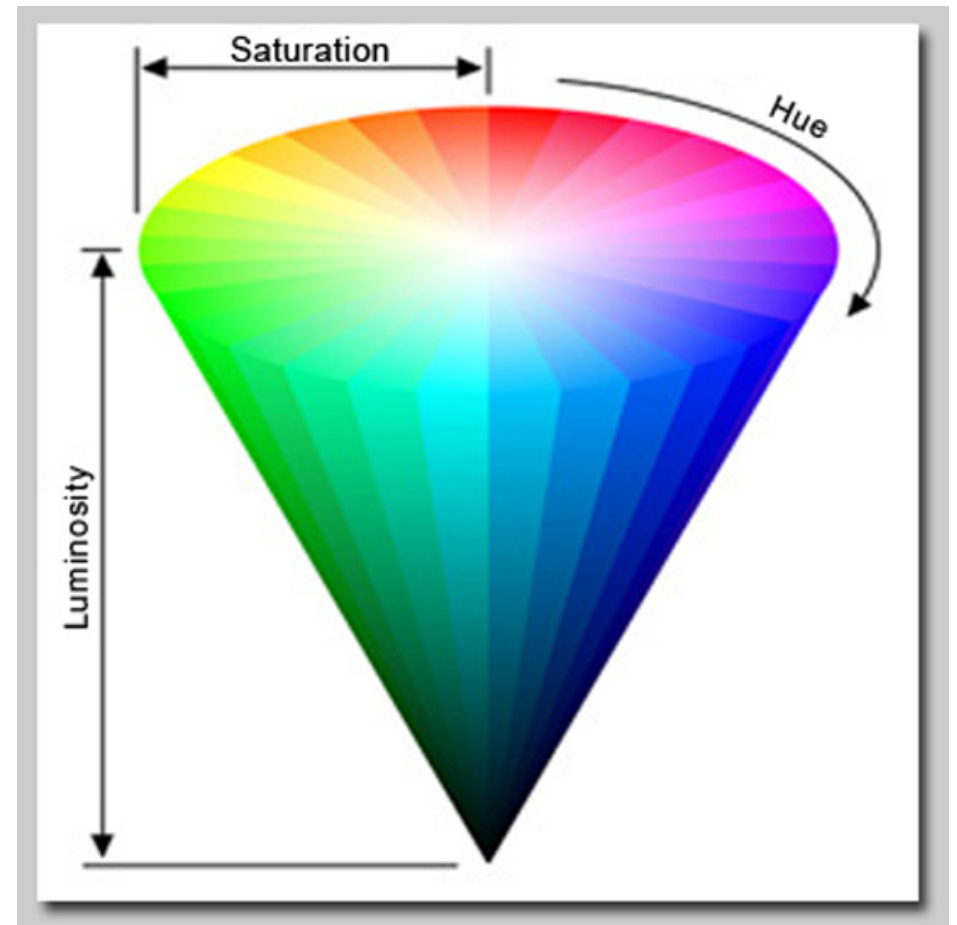


HashMap<String, ArrayList<Color>>



Displaying Colors

- HSB color format:
 - Choose the *hue* (which color), *saturation* (how intense), and *brightness* (absolute brightness).
 - Each choice in the range from 0.0 to 1.0.



Further Reading

- <http://blog.xkcd.com/2010/05/03/color-survey-results/>

