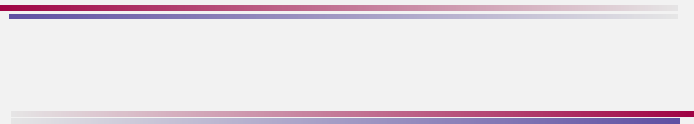


Interaction



Static Sketches

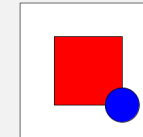
- in a static sketch, nothing changes over time
- a static sketch in Processing is just a list of statements, executed from top to bottom – each statement is executed once

- template

```
// open window  
// clear the background  
// draw stuff
```

- example

```
size(200,200);  
background(255);  
rectMode(CENTER);  
ellipseMode(CENTER);  
fill(255,0,0);  
rect(100,100,100,100);  
fill(0,0,255);  
ellipse(150,150,50,50);
```



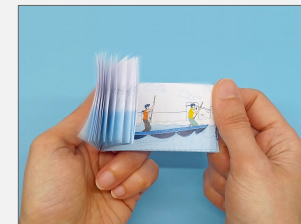
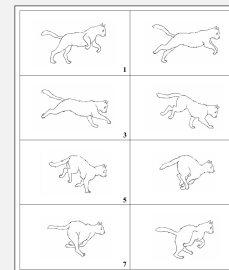
3

Active Sketches

- in an active sketch, things change (or have the potential to change) over time

Time in Processing

- time in the real world is continuous
- time in Processing is discrete
 - an active sketch consists of a series of *frames* drawn one after the other



Active Sketches

- an active sketch in Processing has two lists of statements instead of the single list in a static sketch
 - things that happen once when the sketch first starts – such as opening a window – go into the `setup()` function
 - things that happen to draw one frame go into the `draw()` function

- **template**

```
void setup () {  
    // open window  
    // other things done only once at the beginning  
}  
  
void draw () {  
    // draw one frame  
    // update what changes for next frame  
}
```

- when the sketch is run, first all of the statements in `setup()` are executed once, then the statements in `draw()` are executed over and over and over and ...

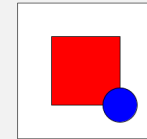
Comparison – Static vs Active

static

```
size(200,200);  
background(255);  
rectMode(CENTER);  
ellipseMode(CENTER);  
fill(255,0,0);  
rect(100,100,100,100);  
fill(0,0,255);  
ellipse(150,150,50,50);
```

active

```
void setup () {  
    size(200,200);  
}  
  
void draw () {  
    background(255);  
    rectMode(CENTER);  
    ellipseMode(CENTER);  
    fill(255,0,0);  
    rect(100,100,100,100);  
    fill(0,0,255);  
    ellipse(150,150,50,50);  
}
```



Active Sketches: Interaction vs Animation

In *interaction*, the changes over time are driven by user actions.

- mouse or keyboard actions

In *animation*, the changes over time are driven (only) by the passage of time.

A sketch can include both interaction and animation at the same time.

Variables

In a static sketch, the position, size, and color of each shape is hardcoded.

```
fill(255,0,0);  
rect(100,100,100,100);
```

- changing the value means changing the code itself and running the program again

For values that change over time, we need variables.

- a *variable* is a name which has an associated value

- can be thought of as a box which can hold one value at a time



- when a statement containing the name of a variable is executed, the computer looks up and substitutes in the current value of the variable

Mouse Position

The current position of the mouse is represented by the system variables `mouseX` and `mouseY`.

```
// draw a rectangle whose upper left corner is at the
// mouse's current position
rectMode(CORNER);
rect(mouseX,mouseY,50,50);
```

Handling Events

Mouse clicks and key presses are *events* – user-initiated things that happen at some moment in time.

Fill in the body of an *event handler* function to specify what to do when a given kind of event occurs.

- system calls the appropriate function when that type of event happens
- statements in the function body are executed from top to bottom

```
void mouseClicked () {
  ...
}
void mousePressed () {
  ...
}
void mouseReleased () {
  ...
}

void keyPressed () {
  ...
}
```

