The Mouse, Keys & Text

Lawrence Snyder
University of Washington, Seattle
An important part of computing is the input to the program and the output it produces.

We consider three types of I/O:

- Mouse Input
- Key Input
- Text Input
So that it is all out there, here’s the situation on the mouse (you’ve seen most of this):

- `mouseX` and `mouseY` give the coordinates of the mouse on the canvas … recall:
  
  ```
  rect(mouseX, mouseY, 20, 20);
  ```

- `void mousePressed ( ) {`
  
  ```
  dir = 0 – dir;
  ```

- There’s also `mouseReleased` that “fires” when the mouse is released after being pressed
Control “looping” with the mouse

```java
int x=0;
void setup( ) {
    size(300, 100);
    background(0);
}
void draw( ) {
    background(0);
    smoke( );
    fill(255);
    rect(40+x, 40, 100, 20);
    triangle(140+x, 40, 155+x, 50, 140+x, 60);
    triangle(35+x, 40, 35+x, 30, 60+x, 40);
    triangle(35+x, 60, 35+x, 70, 60+x, 60);
    x = x+1;
}
void smoke( ) {
    float d;
    fill(255, 200, 0);
    ellipse((x+40)-(x%10), 50, max(10, x%30), max(15, x%30));
}
```
Stop Animation Action

- Control “looping” with the mouse

```cpp
int x=0;
void setup( ) {
    size(300, 100);
    background(0);
    noLoop();
}
...
void smoke( ) {
    float d;
    fill(255, 200, 0);
    ellipse((x+40)-(x%10), 50, max(10, x%30), max(15, x%30));
}
void mousePressed( ) {
    loop( );
}
void mouseReleased( ) {
    noLoop( );
}
```

Looping off
Turn looping on
Turn looping off
Pressing a key is like pressing mouse button ...

Result of typing g y m m y g

```java
int pos = 0;

void setup( ) {
    size(400, 100);
    background(0);
    fill(0);
}

void draw( ) {
    ellipse(pos, 40, 40, 40);
}

void keyPressed( ) {
    if (key == 'g') {
        fill(0, 255, 0);
    }
    if (key == 'y') {
        fill(255, 255, 0);
    }
    if (key == 'm') {
        fill(255, 0, 255);
    }
    pos = pos + 50;
}
```
Datatype Information

- The **key** keyword has the value of the key just pressed; it has the datatype of a character, that is, **char**
- Notice that characters are enclosed in single quotes:

```c
void keyPressed() {
    if (key == 'g') {
        fill(0, 255, 0);
    }
    if (key == 'y') {
        fill(255, 255, 0);
    }
    if (key == 'm') {
        fill(255, 0, 255);
    }
    pos = pos + 50;
}
```
So, What Does This Code Do?

```plaintext
char last = ' ';

void setup() {
    size(100, 100);
    background(0);
    fill(0);
}

void draw() {
    ellipse(50, 40, 40, 40);
}

void keyPressed() {
    if (key == last) {
        fill(0, 255, 0);
    } else {
        fill(255, 0, 0);
    }
    last = key;
}
```
Processing is great for graphics and images, but it is a little more cumbersome for text.

Follow these steps:

1) Go to tools and locate the font you want
2) Load font into the data directory of your program ... this happens automatically when you load
3) In the code, load the font into the computation (get name and size perfect); specify its use
4) Use text( ) to print text; color using fill( )
Find Fave Font

- “Create Font …” is under Tools
Pick Font, Size

- Try to pick common fonts
- Click to load font into the data directory
Pick Font, Size

- Try to pick common fonts
- Click to load font into the data directory
Declare Font Var, Load, Select

- Need to declare font name(s)

```cpp
PFont typeface1, typeface2;

void setup ( ) {
  size(400,100);
  background(0);
  typeface1 = loadFont("Desdemona-48.vlw");
  textFont(typeface1);
}

void draw( ) {
  fill(255);
  text("A cool font!", 20, 80);
}
```
Declare Font Var, Load, Select

- Need to declare font name(s)
- Need to load named font

```
PFont typeface1, typeface2;

void setup () {
  size(400,100);
  background(0);
  typeface1 = loadFont("Desdemona-48.vlw");
  textFont(typeface1);
}

void draw() {
  fill(255);
  text("A cool font!", 20, 80);
}
```
Declaring a Font Variable, Loading, and Selecting

- Need to declare font name(s)
- Need to load named font
- Need to select named font as “in use”

```cpp
PFont typeface1, typeface2;

void setup () {
  size(400,100);
  background(0);
  typeface1 = loadFont("Desdemona-48.vlw");
  textFont(typeface1);
}

void draw( ) {
  fill(255);
  text("A cool font!", 20, 80);
}
```
Declare Font Var, Load, Select

- Need to declare font name(s)
- Need to load named font
- Need to select named font as “in use”
- Then, fill( ) and write text( ... );
Switching Fonts ...

```cpp
PFont typeface1, typeface2;
void setup () {
  size(400, 100);
  background(0);
  typeface1 = loadFont("Desdemona-48.vlw");
  typeface2 = loadFont("AppleCasual-24.vlw");
}
void draw() {
  fill(255);
 setFont(typeface1);
  text("A cool font!", 20, 80);
  fill(255, 0, 0);
  setFont(typeface2);
  text("really", 28, 35);
  triangle(50, 75, 40, 90, 60, 90);
}
```
A String is a datatype of a letter sequence. The sequence must be surrounded by (double) quotes. "" is the empty String.
A String is a datatype of a letter sequence. The sequence must be surrounded by (double) quotes. "" is the empty String.

A character can be added to a String (it’s called *concatenation*) using a + sign.

```
void keyPressed() {
    st = st + key;
}
```
A String is a datatype of a letter sequence. The sequence must be surrounded by (double) quotes. "" is the empty String.

A character can be added to a String (it’s called concatenation) using a + sign.
A String is a datatype of a letter sequence. The sequence must be surrounded by (double) quotes. "" is the empty String.

```java
PFont typeface1;
String st = "";

void setup () {
    size(400,100);
    background(0);
    typeface1 = loadFont("Desdemona-48.vlw");
    smooth( );
}

void draw () {
    fill(255);
    textFont(typeface1);
    text(st, 20, 80);
}

void keyPressed () {
    st = st + key;
}
```

A character can be added to a String (it’s called *concatenation*) using a + sign.
A String is a datatype of a letter sequence. The sequence must be surrounded by (double) quotes. "" is the empty String.

A character can be added to a String (it’s called concatenation) using a + sign.