COMP 1010- Summer 2015 (A01)

Jim (James) Young

young@cs.umanitoba.ca

jimyoung.ca

Hello!

James (Jim) Young
young@cs.umanitoba.ca
jimyoung.ca

office hours T / Th: 17:00 – 18:00
EITC-E2-582
(or by appointment, arrange by email)

Remember the cat?

```
/*******************
* Cat Face! Draw a cat face on the screen
* author: Teo the dog
* version: try #awesome
* purpose: to show how a cat can be drawn
*******************/

size(500,500); // make a 500x500 canvas

//draw the head
ellipse(250,250,300,300);

//draw the ears
triangle(375,80,300,150,400,200);
triangle(125,80,200,150,100,200);

//draw the eyes
ellipse(175,225,60,30); // left eye
ellipse(175,225,15,30);
ellipse(325,225,60,30); // right eye
ellipse(325,225,15,30);

//whiskers!
line(250,300,200,275);
line(250,300,300,275);
line(250,300,190,300);
line(250,300,310,300);
line(250,300,200,325);
line(250,300,300,325);

// draw the nose. draw after whiskers for nice overlap effect
ellipse(250,300,30,30);
```

Cat whisker:

line(250,300,300,325);

→line(noseCenterX,noseCenterY,300,325);

Notice: the line end point is..

50 pixels to the right (250→300)

25 pixels below (300→325)

line(noseCenterX, noseCenterY,

noseCenterX+50, noseCenterY+25);

integer “operators” - subtraction

the “-” symbol:

<integer> - <integer>

10-2, 5-10, 40-30, 34243401-1312322

Cat whisker:

line(250,300,200,275);

→line(noseCenterX,noseCenterY,200,275);

Notice: the line end point is..

50 pixels to the left(250→200)

25 pixels above (300→275)

line(noseCenterX, noseCenterY,

noseCenterX-50, noseCenterY-25);

Let’s look back at the cat code: the eyes

```
//draw the eyes
ellipse(175,225,60,30); // left eye
ellipse(175,225,15,30);
ellipse(325,225,60,30); // right eye
ellipse(325,225,15,30);
```

Pupil is 15 wide
Pupil is twice high as it is wide (30 high)
eye height = pupil height (they just touch!)
eye width is twice the eye height

integer “operators” - multiplication

the “*” symbol:

<integer> * <integer>

5*5, 10*2, 2*2, 231421341*12341234

Setup the code for the pupil size

integer “operators” - division

the “/” symbol:

<integer> / <integer>

$\frac{10}{5}$ ½

10/5 , 50/10, 9/3, 12/4

We can use this to reverse the eye example

Instead, do everything with respect to the width of the eye.

Eye ratios:

Eye width is 60

Eye height is half width

Pupil height is half eye width

Pupil width is quarter eye height

Cat example – practice update with variables (at home)

We already have nose/whiskers and eye sizes done

- Head center
  - Update head ellipse
  - Update Eye locations
  - Update ear locations
  - Update nose center

Back to division….

Let’s make a line that goes X percent across the screen

```
int percent = 33;
int targetX = percent/100*500;

line(0,250,targetX,250);
```

what happened?

Reality check – calculate by hand

33/100*500 = ?

165 – try it

Why did we get a different answer?

IMPORTANT HELPER TOOL

Remember the console in processing?

You can toss data out there for a reality check

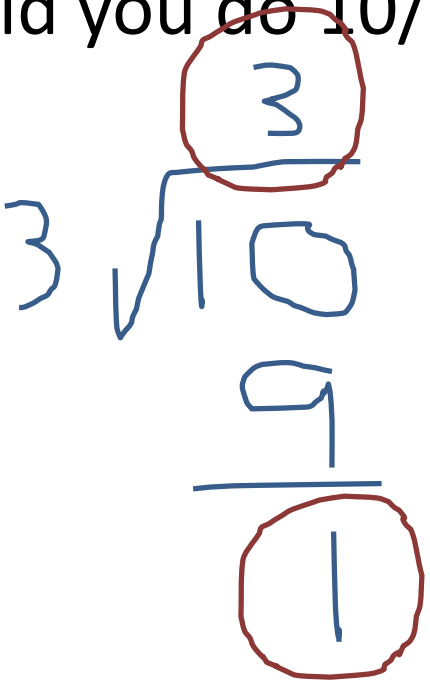
New processing command:

println(data);

Use this to debug our problem

before highschool:

How did you do 10/3 in elementary school?



The answer is 3 remainder 1

In processing: 3 is the result

The remainder is discarded

division – integers never give a fraction amount. (seriously)

1/2 = ?

11/3 = ?

100/26 = ?

integer division always discards the fraction amount and gives you the whole amount.

does it always round down?

-9/10 = ?

remainder: (also called modulo, mod)

difficult but useful - highly recommend you practice this

use the “%” symbol

10%2

remainder when you do 10/2

10/2 = 5 R 0

10%2 = 0

5%2?

5/2 = 2 R 1

5%2 = 1

11%3?

11/3 = 3 R 2

11%3 = 2

Order of operations!

order of operations!

complex statements:

3+2*6/3%4

what is the answer?

order of operations!! BEDMAS

Brackets!

Exponents (and roots)!

Division and Multiplication (and remainder)

Addition and Subtraction

order of operations!

not 100% sure? just use brackets to enforce what you mean:

3+2*6/3%4 → 3 + (2*6/3)%4 = 3

Active processing

Active Processing

So far – *static processing*

one run through, shows the result at the end

Active processing

the program starts, and keeps running!!

things can animate

respond to keyboard, mouse!

New command: random

Generate a number [0..high)

from 0 to high but excluding high.

random(high);

e.g.,

random(5); // generate a random number 0..4

Boring static program:

Line from center to random spot

```
line(250,250,random(500),random(500));
```

Moving to active..

Active programs have two regions, or blocks.

- setup – run once
- draw – run over and over!!

60 times a second!

Processing has special syntax to specify blocks

Processing *block* syntax

```
{
    // this is a code block
}
```

How to setup your active blocks

You can give blocks names – these are called functions (later). Kind of like making your own commands

Some commands take parameters and some give you data. Some do both.

line(1,2,3,4);

random(500)

We setup our regions in a similar way

How to setup your active blocks

resultType blockName(parameters)

{

}

for now, don’t worry too much about the result or parameters…

No result? Put “void”

Active blocks – setup and draw

```
void setup()
{
   // run once
}

void draw()
{
   // run 60 times every second
}
```

NO COMMANDS OUTSIDE THESE BLOCKS

(only create variables.. Later)

First active program:

```
void setup()
{
  size(500,500);
  line(250,250,random(500),random(500));
}
void draw()
{
}
```

Make it active – move to the draw

```
void setup()
{
   size(500,500);
}
void draw()
{
 line(250,250,random(500),random(500));
}
```

What do you expect to happen in this program

```
void setup()
{
  size(500,500);
}
void draw()
{
  int linePosition = 0;
  line(250,0,linePosition,linePosition);
  linePosition = linePosition + 1;
}
```

Variables and blocks **RULE #1**

If you define a variable inside a block, it only exists until the block is finished. You can do it, but once the block is over, the variable gets destroyed.

The next time into the block, the variable gets re-created and re-initialized

Called a **local variable**

Maybe move the declaration to the startup?

```
void setup()
{
  size(500,500);
  int linePosition = 0;
}
void draw()
{
  line(250,0,linePosition,linePosition);
  linePosition = linePosition + 1;
}
```

Variables and blocks **RULE #2**

A variable created inside one block is only visible within that block. Other blocks cannot see them.

This is called a scope rule – it defines the area where a variable is visible

Solution: global variables

Put variables at the top of the program, outside the blocks

- visible in all blocks
- only created once and not destroyed on every redraw!

result

```
int linePosition = 0;
void setup()
{
  size(500,500);
}
void draw()
{
  line(250,0,linePosition,linePosition);
  linePosition = linePosition + 1;
}
```

MOUSE!!! YAY!!

Processing globals

// current mouse position at beginning of DRAW

mouseX

mouseY

// previous mouse position, at last draw

pmouseX

pmouseY

Example 1: line to mouse position

Example 1: line from last position to current

Example 3: don’t erase background

Hold the roof up!

Make a line that is falling down, like a roof, and hold it up with the mouse.

First, make a falling line

- Variable for current line position
- Draw across the screen at that y
- Increase y each time it draws

Hold it up with the mouse?

If the line is below the mouse (bigger y), we should bring it back up to where the mouse is.

New commands!

max(a,b) – gives the bigger of the two

min(a,b) – gives the smaller of the two

Let's update the cat face program

```
/*******************
* Cat Face! Draw a cat face on the screen
* author: Teo the dog
* version: try #awesome
* purpose: to show how a cat can be drawn
********************/

// variables
int headCenterX = 250;
int headCenterY = 250;
int noseSize = 30;
int pupilWidth = 15;
int noseCenterX = headCenterX;
int noseCenterY = headCenterY+50;

size(500,500); // make a 500x500 canvas

//draw the head
ellipse(headCenterX,headCenterY,300,300);

//draw the ears
triangle(headCenterX+125,headCenterY-170,
        headCenterX+50,headCenterY-100,
        headCenterX+150,headCenterY-50);
triangle(headCenterX-125,headCenterY-170,
       headCenterX-50,headCenterY-100,
       headCenterX-150,headCenterY-50);

//draw the eyes
ellipse(headCenterX-75,headCenterY-25,
      pupilWidth*4,pupilWidth*2); // left eye
ellipse(headCenterX-75,headCenterY-25,
      pupilWidth,pupilWidth*2);
ellipse(headCenterX+75,headCenterY-25,
      pupilWidth*4,pupilWidth*2); // right eye
ellipse(headCenterX+75,headCenterY-25,
      pupilWidth,pupilWidth*2);

//whiskers!
line(noseCenterX,noseCenterY,noseCenterX-50,noseCenterY-25);
line(noseCenterX,noseCenterY,noseCenterX+50,noseCenterY-25);
line(noseCenterX,noseCenterY,noseCenterX-60,noseCenterY);
line(noseCenterX,noseCenterY,noseCenterX+60,noseCenterY);
line(noseCenterX,noseCenterY,noseCenterX-50,noseCenterY+25);
line(noseCenterX,noseCenterY,noseCenterX+50,noseCenterY+25);

// draw the nose. draw after whiskers for nice overlap effect
ellipse(noseCenterX,noseCenterY,noseSize,noseSize);
```

Let’s update the cat face program

- Keep the variables global: break code into setup and draw
- And clear background command
- Link nose to mouse
- Eyes?