

COMP 1010- Summer 2015 (A01)

Jim (James) Young

young@cs.umanitoba.ca

jimyoung.ca

Trucking along...

Understanding programs:

computers have tunnel vision

explicit, one by one instructions

simplest approach to getting it:

you need to know what happens, in
what order

re-visit the “hold up the roof” example

Hold the roof up!

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

Each time you draw, move the line, so it draws in a new place next time

Use min or max to be sure that the line never goes below the mouse.

Hold up the roof!

```
int lineTop = 0;
```

```
void setup()  
{  
  size(500,500);  
}
```

```
void draw()  
{  
  background(255);  
  lineTop = lineTop + 1;  
  
  lineTop = min(mouseY, lineTop);  
  
  line(0,lineTop,499,lineTop);  
}
```

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

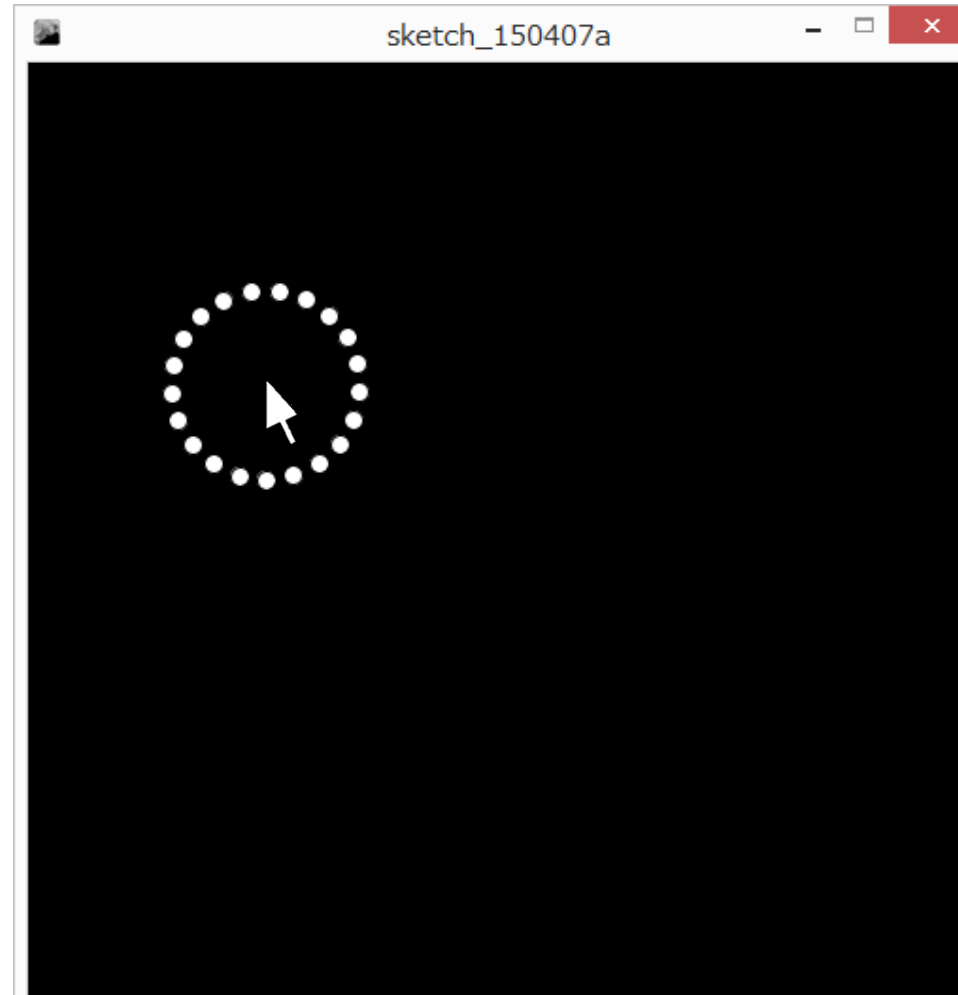
- Keep the variables global: break code into setup and draw
- And clear background command
- Link nose to mouse
- Eyes?

new example:

Make a ball circle around
the mouse

We need trigonometry

We need decimals



integer is fine.. but I *want* fractions!

I WANT $5/2$ to give me 2.5

I don't want to be restricted to integers...

what do I do???

you need a new data type that allows decimal portions...

Floating point!

why is it called floating point? (advanced)

its scientific notation:

first, we give a bunch of numbers:

e.g., 1234567

then, we tell it where to put the point:

e.g., after 4th digit: 1234.567

$$1.234567 \times 10^3$$

5 before first digit: 0.000001234567

$$1.234567 \times 10^{-6}$$

5 after last digit: 123456700000.0

$$1.234567 \times 10^{11}$$

we can store VERY small and VERY large numbers

aside: more globals

width

height

Carful! Only use those **AFTER** you set the canvas size.

How to make a floating point variable

Instead of:

```
int variableName;
```

```
float variableName;
```

Use floats to draw 30% way across the screen

```
float percent = 0.3;
```

floating point variables

```
size(500,500);
```

```
float percent = 0.3;
```

```
float targetX = percent*500;
```

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

Change the example

```
float percent = 30/100;
```

What happens?

Use `println` to investigate

Doesn't work!!

Processing still does integer division!!! why??

note: the right side of the assignment does not change based on the variable on the left!!

so.. why is $30/100$ giving an integer result?

note: if the numbers in an operation are integers, the computer does integer arithmetic:

in programming speak: if the operands are integers, processing uses an integer operator

how to do floating point math?

note: if the numbers in an operation are integers, processing does integer arithmetic:

note: if either number in an operation is floating point, Java does floating point arithmetic

how do we force $30/100$ to be floating point?

make either the 30 or the 100 a floating point:

$30.0/100$ or

$30/100.0$ or

$30.0/100.0$

Processing works one step at a time... (confusing but important)

what about $1.0+1/2*3.0$?

lets try it

why? – processing does it one step at a time.

Order of operations!

$1.0+1/2*3.0$

$1/2 \rightarrow \text{int} / \text{int} \rightarrow \text{integer division} \rightarrow 0$

$1+0*3.0$

(this is floating point, since $\text{int} * \text{double}$)

$1+0.0$

Trigonometry and processing

Processing uses **radians** not degrees

Circle goes from 0..2PI

PI?

-> 180 degrees

PI/4?

-> 45 degrees

Trigonometry and processing

New constant: PI (all caps)

New way to show you commands

```
type commandName(type parameter);
```

```
float sin(float radians)
```

```
float cos(float radians)
```

```
float tan(float radians)
```

More trigonometry

We have inverse functions $\rightarrow \sin^{-1}, \cos^{-1}, \tan^{-1}$

Also called arccos, arcsin, arctan.

float asin(float ratio)

float acos(float ratio)

float atan(float ratio)

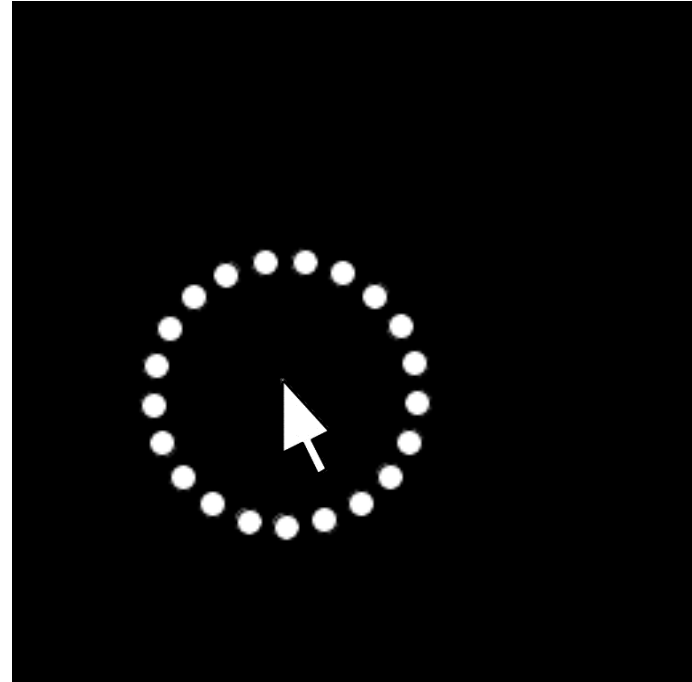
ADVANCED: atan2 (see notes)

Back to our mouse orbiter!

Start with simple static case

Let's pick an angle, θ , and start at some point

We also need a radius – how far the ball will circle



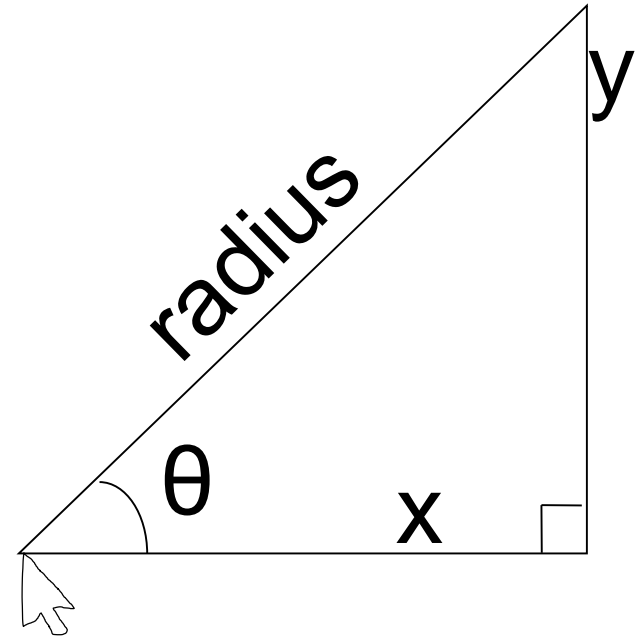
Given a radius and an angle, what is the x, and y?

$$\sin(\theta) = y/\text{radius}$$

$$y = \sin(\theta) * \text{radius}$$

$$\cos(\theta) = x/\text{radius}$$

$$x = \cos(\theta) * \text{radius}$$



It worked!! We can now place a ball at some distance and angle from the mouse

How to make it animate?

What did we do for mouse bubbles?

Every time we draw, adjust angle by a little delta

What about the angle getting too big?

mod and floating point is messy...

angles wrap around we're safe!

At home exercise:

Here is an extension you can try at home: expand this to having multiple planets orbiting the mouse at different speeds. Here are some helping steps:

- rename the variables to start with p1 for planet 1
- copy all your variables for each planet. Try three planets
- make each planet move at a different speed.
What if one moves at a negative speed??