Processing and Synthesizer Lab
Computational Media

Description:

Using Processing and the ‘synth_simulator’ code you will run several experiments with modifying and creating ‘Oscillator’ objects that vibrate at different pitches and waveforms. The ‘synth_simulator’ code is designed to simulate analog synthesizer functions such as playing pitches, shifting octaves, and adding a low frequency ‘wobble’ to the sound. You can start with this program and continue to add features to increase quantity and variety of sounds and inputs.

After downloading and running the synth_simulator, you will make some adaptions to the code and record your observations of the changes in waveforms and sounds on this document. The purpose is to observe how the shape of the waveform creates different timbres of sound that we hear.

This lab will prepare us for the Georgia Tech Werkstatt project we will work on in the following weeks.

Part 1: Downloading the Code and Experimenting with the Synthesizer

Directions:

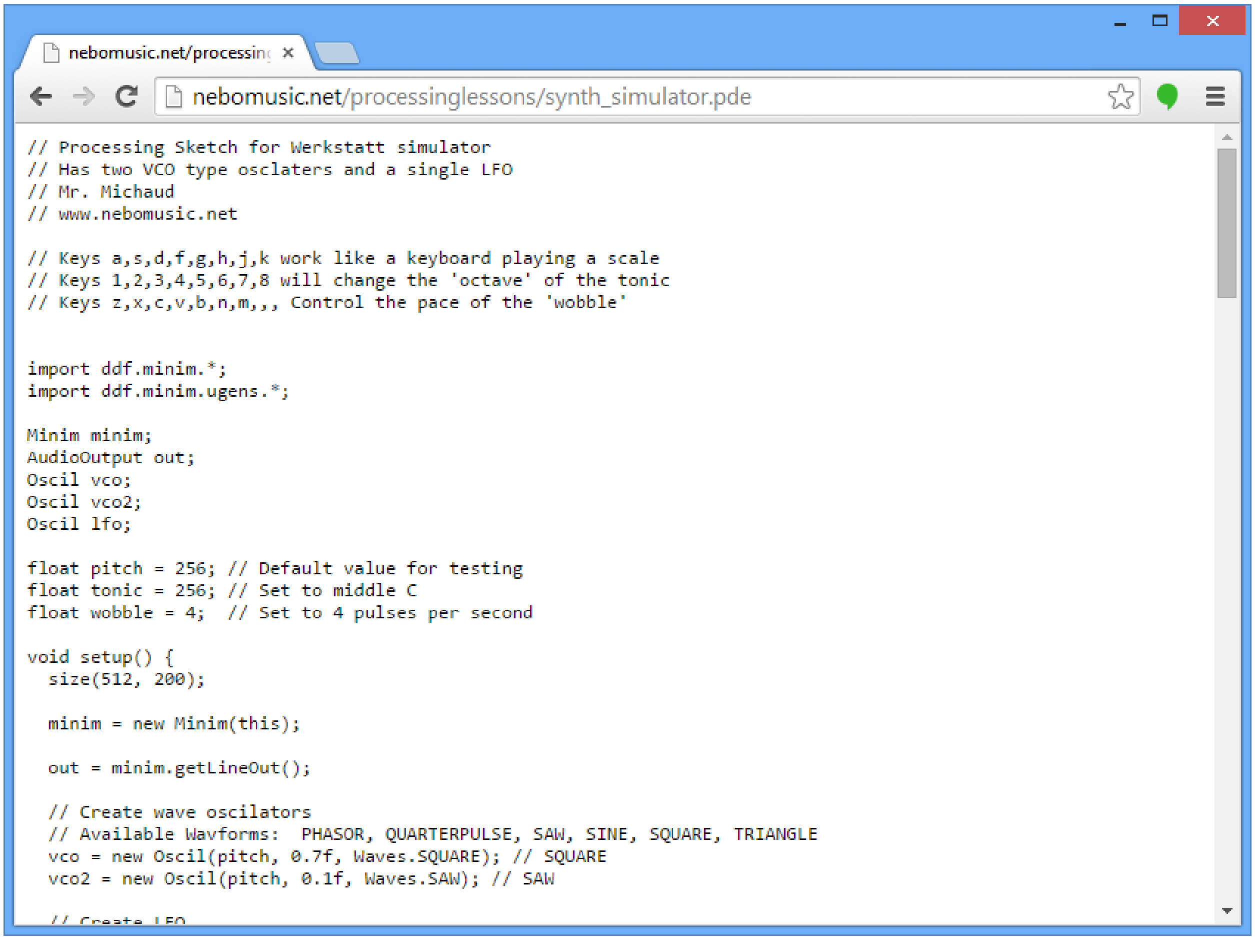
1. Download this document and save to your Processing Folder as “Lastname Processing Synth”

2. Start Processing.

3. Go to the following link:

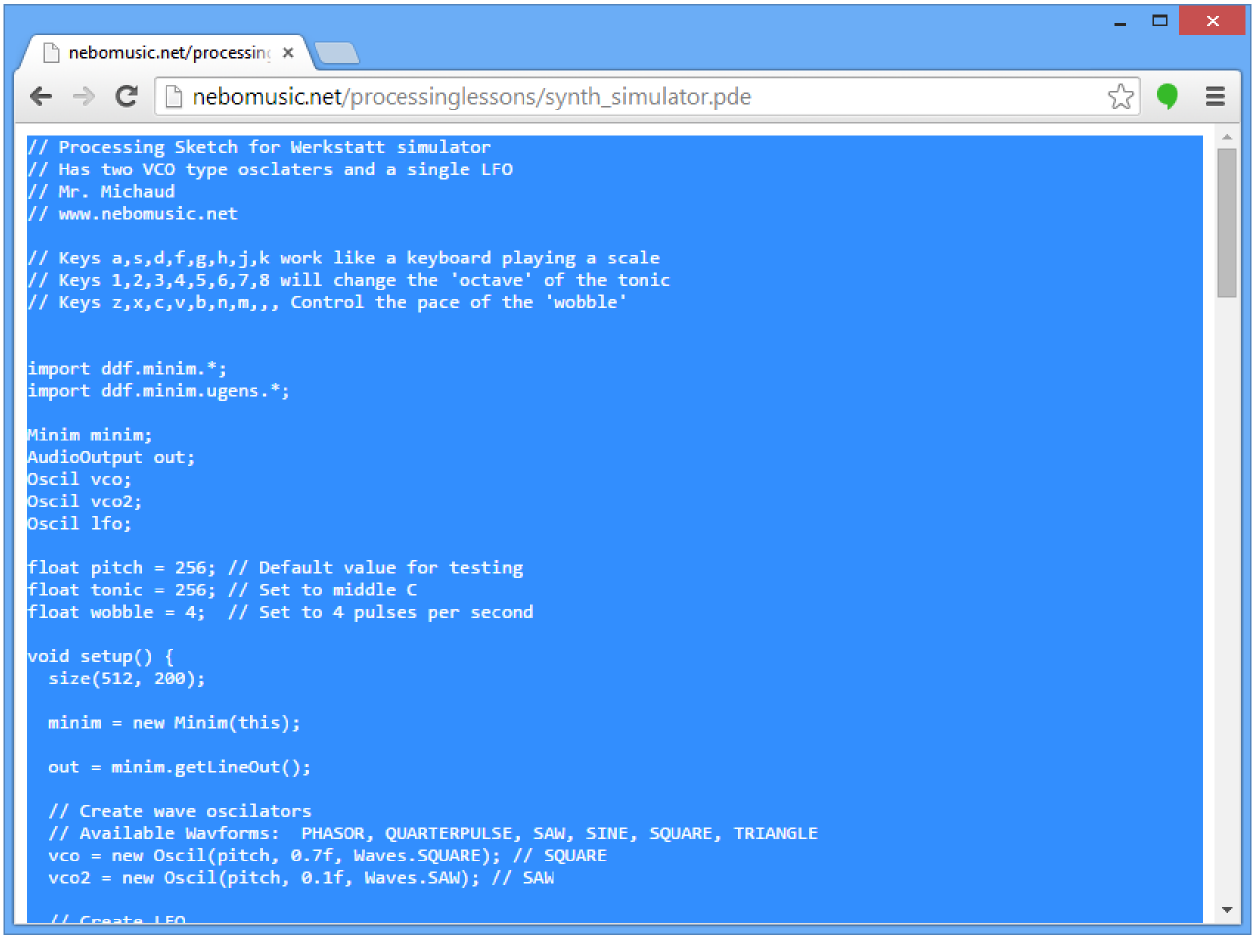
http://nebomusic.net/processinglessons/synth_simulator.pde

4. You will see the code in the browser:



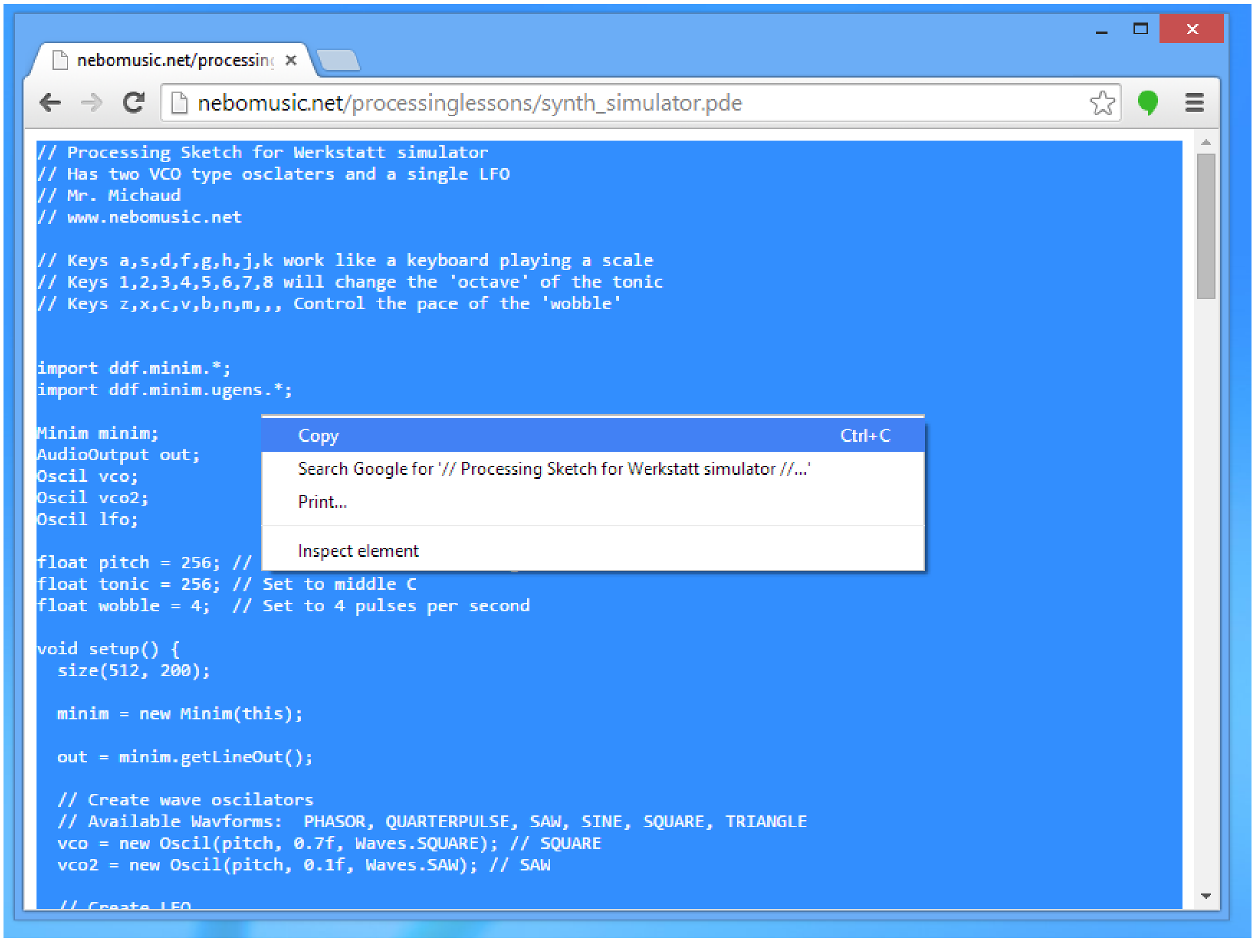
```
// Processing Sketch for Werkstatt simulator
// Has two VCO type osclaters and a single LFO
// Mr. Michaud
// www.nebomusic.net

// Keys a,s,d,f,g,h,j,k work like a keyboard playing a scale
// Keys 1,2,3,4,5,6,7,8 will change the 'octave' of the tonic
// Keys z,x,c,v,b,n,m,,, Control the pace of the 'wobble'


import ddf.minim.*;
import ddf.minim.ugens.*;

Minim minim;
AudioOutput out;
Oscil vco;
Oscil vco2;
Oscil lfo;

float pitch = 256; // Default value for testing
float tonic = 256; // Set to middle C
float wobble = 4;  // Set to 4 pulses per second

void setup() {
  size(512, 200);

  minim = new Minim(this);

  out = minim.getLineOut();

  // Create wave oscilators
  // Available Wavforms:  PHASOR, QUARTERPULSE, SAW, SINE, SQUARE, TRIANGLE
  vco = new Oscil(pitch, 0.7f, Waves.SQUARE); // SQUARE
  vco2 = new Oscil(pitch, 0.1f, Waves.SAW); // SAW

  // Create LFO
```

5. Click once on the code and then type ‘Ctrl-a’ (Control – a) This should select all the code.



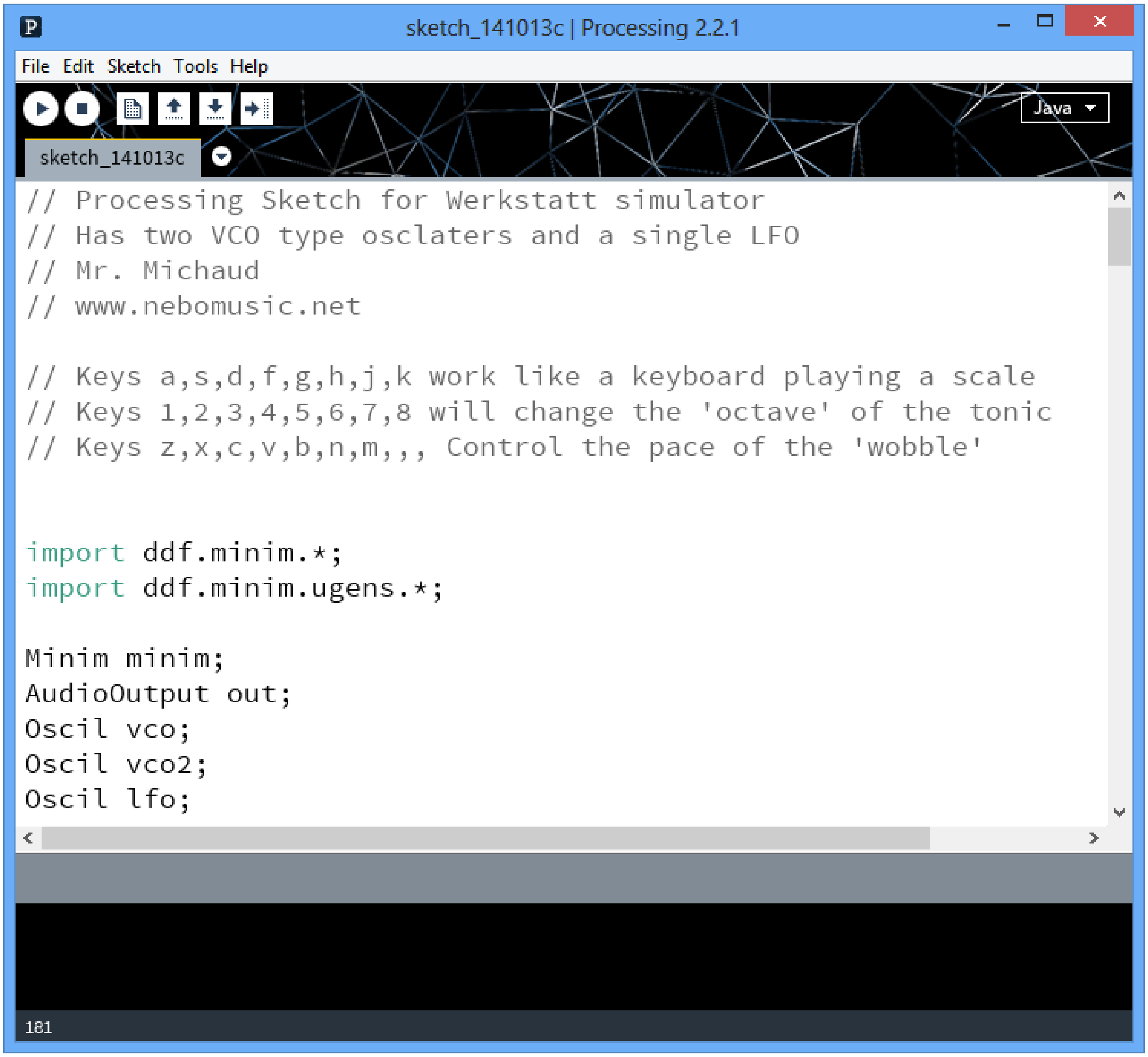
```
// Processing Sketch for Werkstatt simulator
// Has two VCO type osclaters and a single LFO
// Mr. Michaud
// www.nebomusic.net

// Keys a,s,d,f,g,h,j,k work like a keyboard playing a scale
// Keys 1,2,3,4,5,6,7,8 will change the 'octave' of the tonic
// Keys z,x,c,v,b,n,m,,, Control the pace of the 'wobble'


import ddf.minim.*;
import ddf.minim.ugens.*;

Minim minim;
AudioOutput out;
Oscil vco;
Oscil vco2;
Oscil lfo;

float pitch = 256; // Default value for testing
float tonic = 256; // Set to middle C
float wobble = 4;  // Set to 4 pulses per second

void setup() {
  size(512, 200);

  minim = new Minim(this);

  out = minim.getLineOut();

  // Create wave oscilators
  // Available Wavforms:  PHASOR, QUARTERPULSE, SAW, SINE, SQUARE, TRIANGLE
  vco = new Oscil(pitch, 0.7f, Waves.SQUARE); // SQUARE
  vco2 = new Oscil(pitch, 0.1f, Waves.SAW); // SAW

  // Create LFO
```

6. Right Click and select ‘Copy’. (Or Control-c)


nebomusic.net/processin
nebomusic.net/processinglessons/synth_simulator.pde
Copy
Ctrl+C
Search Google for '// Processing Sketch for Werkstatt simulator //...'
Print...
Inspect element

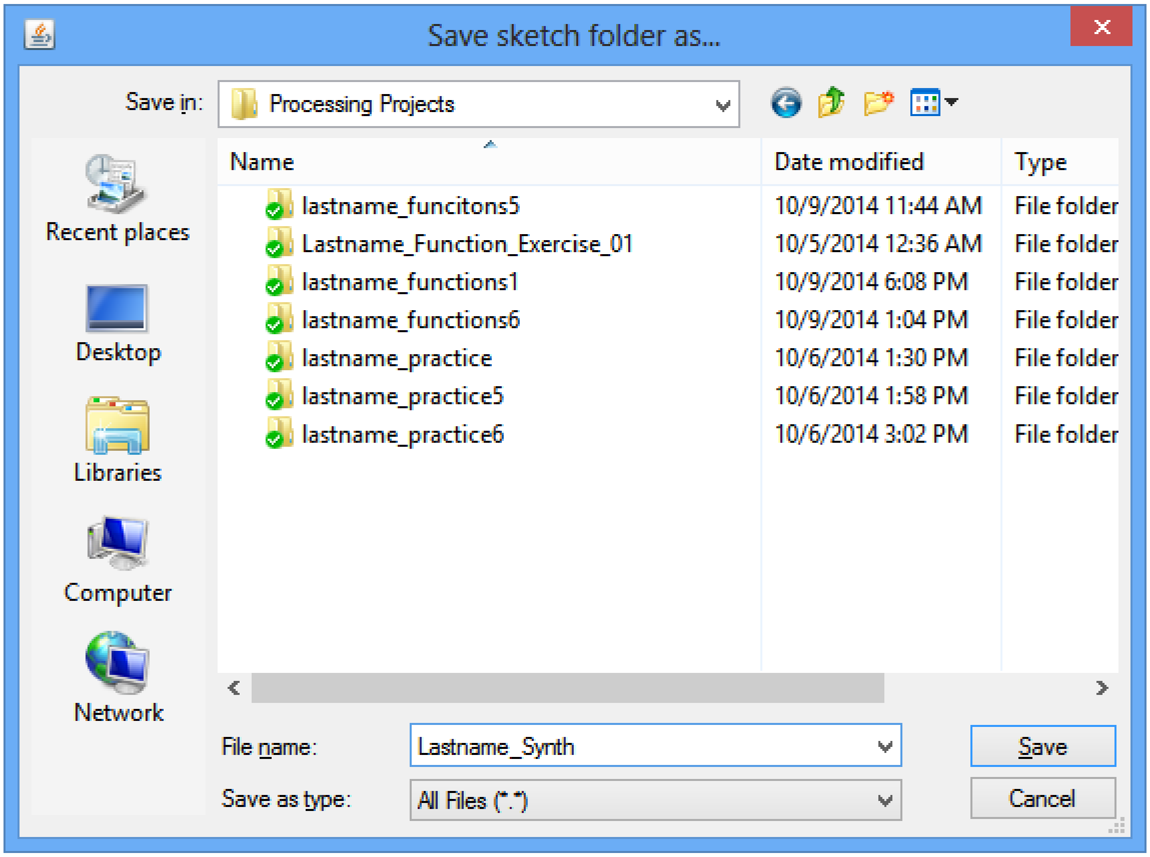
```
// Processing Sketch for Werkstatt simulator
// Has two VCO type osclaters and a single LFO
// Mr. Michaud
// www.nebomusic.net

// Keys a,s,d,f,g,h,j,k work like a keyboard playing a scale
// Keys 1,2,3,4,5,6,7,8 will change the 'octave' of the tonic
// Keys z,x,c,v,b,n,m,,, Control the pace of the 'wobble'


import ddf.minim.*;
import ddf.minim.ugens.*;

Minim minim;
AudioOutput out;
Oscil vco;
Oscil vco2;
Oscil lfo;

float pitch = 256; //
float tonic = 256; // Set to middle C
float wobble = 4;  // Set to 4 pulses per second

void setup() {
  size(512, 200);

  minim = new Minim(this);

  out = minim.getLineOut();

  // Create wave oscilators
  // Available Wavforms:  PHASOR, QUARTERPULSE, SAW, SINE, SQUARE, TRIANGLE
  vco = new Oscil(pitch, 0.7f, Waves.SQUARE); // SQUARE
  vco2 = new Oscil(pitch, 0.1f, Waves.SAW); // SAW

  // Create LFO
```

7. Go to your blank Processing Document and paste the code into Processing.


sketch_141013c | Processing 2.2.1
File Edit Sketch Tools Help
Java
sketch_141013c
181

```
// Processing Sketch for Werkstatt simulator
// Has two VCO type osclaters and a single LFO
// Mr. Michaud
// www.nebomusic.net

// Keys a,s,d,f,g,h,j,k work like a keyboard playing a scale
// Keys 1,2,3,4,5,6,7,8 will change the 'octave' of the tonic
// Keys z,x,c,v,b,n,m,,, Control the pace of the 'wobble'


import ddf.minim.*;
import ddf.minim.ugens.*;

Minim minim;
AudioOutput out;
Oscil vco;
Oscil vco2;
Oscil lfo;
```

8. Select File-Save and save your Processing as “Lastname_Synth” in your Computational Media Processing Folder.



Save in: Processing Projects

Name	Date modified	Type
lastname_funcitons5	10/9/2014 11:44 AM	File folder
Lastname_Function_Exercise_01	10/5/2014 12:36 AM	File folder
lastname_functions1	10/9/2014 6:08 PM	File folder
lastname_functions6	10/9/2014 1:04 PM	File folder
lastname_practice	10/6/2014 1:30 PM	File folder
lastname_practice5	10/6/2014 1:58 PM	File folder
lastname_practice6	10/6/2014 3:02 PM	File folder

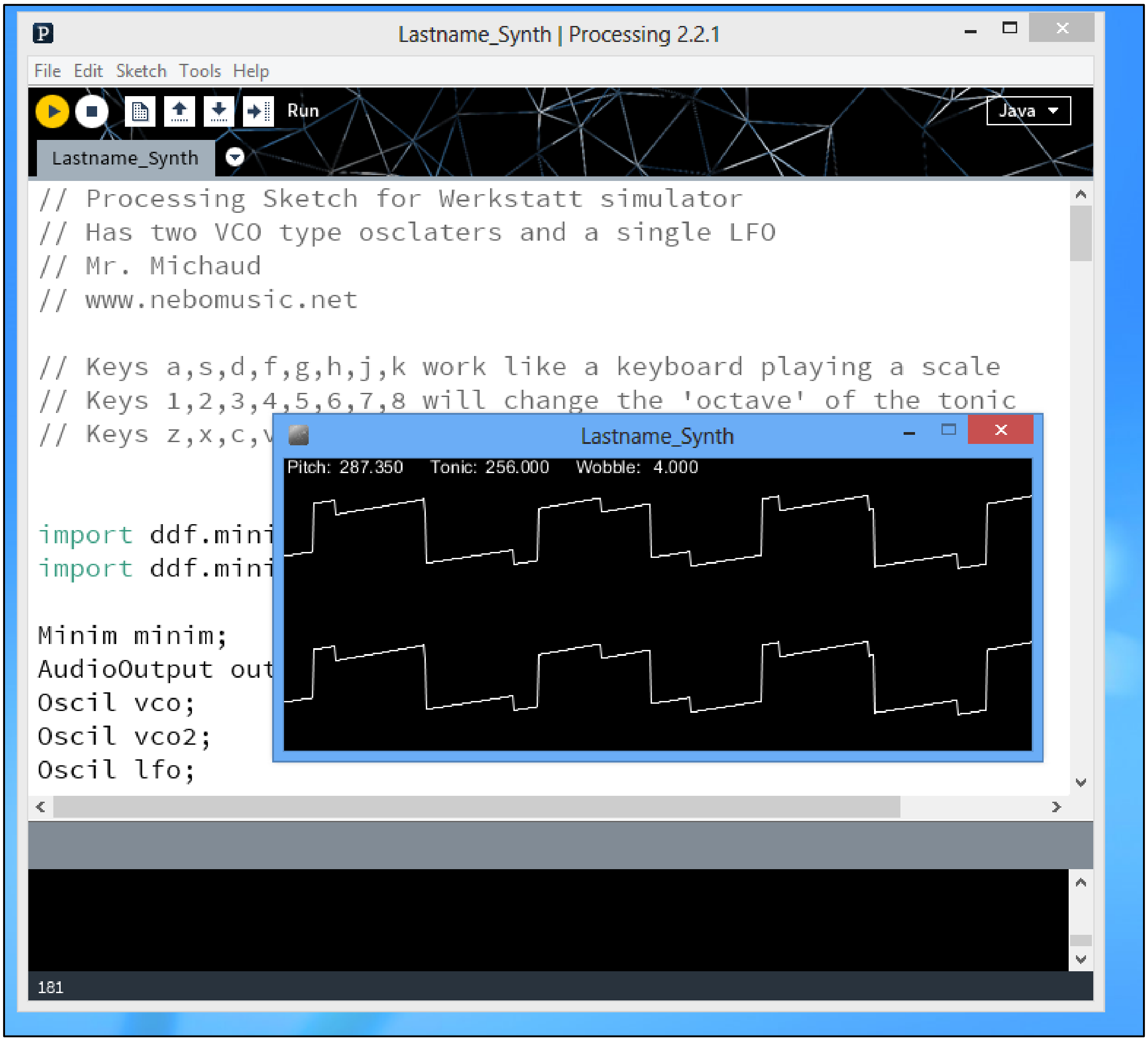
File name: Lastname_Synth

Save as type: All Files (*.*)

9. You now have the synth installed. Click Play and you should see the waves and hear the sound. Experiment with the following keys:

```
// Keys a,s,d,f,g,h,j,k work like a keyboard playing a scale
// Keys 1,2,3,4,5,6,7,8 will change the 'octave' of the tonic
// Keys z,x,c,v,b,n,m,,, Control the pace of the 'wobble'
```

The view should look like:


Lastname_Synth | Processing 2.2.1
File Edit Sketch Tools Help
Run
Java
Lastname_Synth
Lastname_Synth
181

```
// Processing Sketch for Werkstatt simulator
// Has two VCO type osclaters and a single LFO
// Mr. Michaud
// www.nebomusic.net

// Keys a,s,d,f,g,h,j,k work like a keyboard playing a scale
// Keys 1,2,3,4,5,6,7,8 will change the 'octave' of the tonic
// Keys z,x,c,v


import ddf.mini
import ddf.mini

Minim minim;
AudioOutput out
Oscil vco;
Oscil vco2;
Oscil lfo;
```

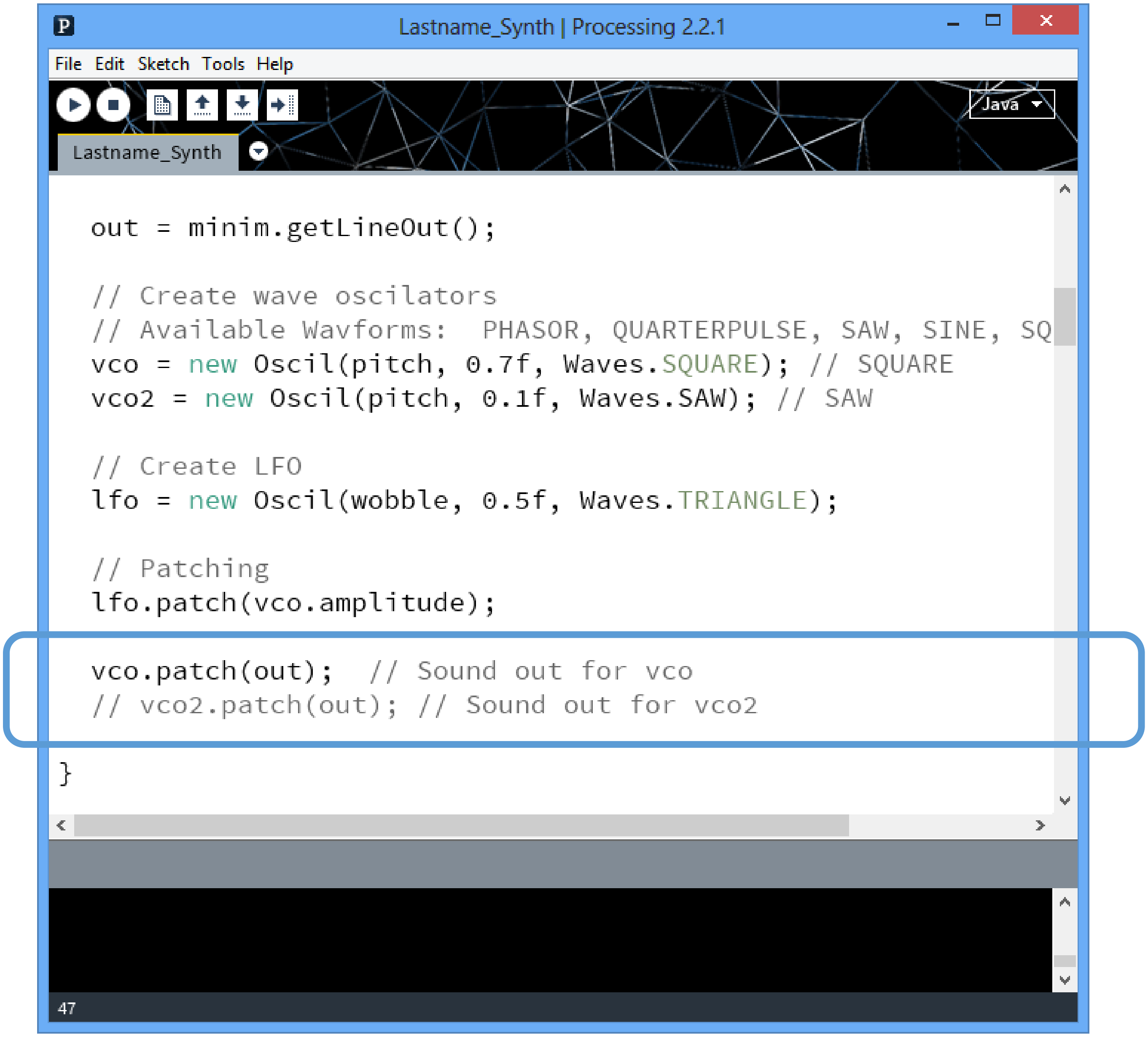
```
Pitch: 287.350   Tonic: 256.000   Wobble: 4.000
```

10. After experimenting with the keyboard inputs and sounds, move on to Part 2 where we will change the waveforms in the code.

Part 2: Lab on Waveshape and Timbre

1. Go to line 43 in the code and comment out the line:

```
vco2.patch(out);
```



Lastname_Synth | Processing 2.2.1
File Edit Sketch Tools Help
Java
Lastname_Synth
47

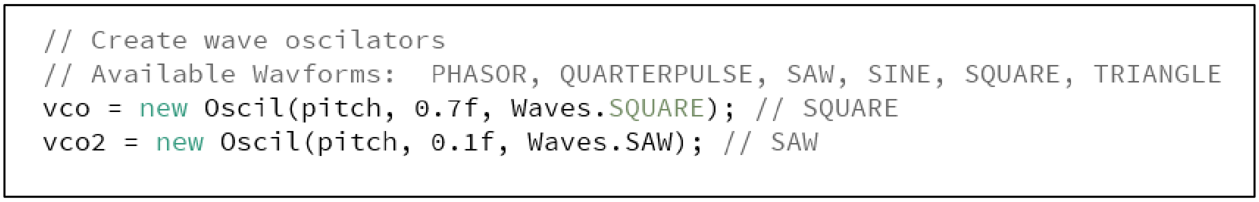
```
  out = minim.getLineOut();

  // Create wave oscilators
  // Available Wavforms:  PHASOR, QUARTERPULSE, SAW, SINE, SQ
  vco = new Oscil(pitch, 0.7f, Waves.SQUARE); // SQUARE
  vco2 = new Oscil(pitch, 0.1f, Waves.SAW); // SAW

  // Create LFO
  lfo = new Oscil(wobble, 0.5f, Waves.TRIANGLE);

  // Patching
  lfo.patch(vco.amplitude);

  vco.patch(out);  // Sound out for vco
  // vco2.patch(out); // Sound out for vco2

}
```

This turns off the 'overtone' so you will hear only one pitch. Run the program and note the difference in sound.

2. Go to line 33 in the Code and experiment by changing the Waveform for the vco to the following shapes. For each shape, describe the quality of sound - or where you have heard this type of sound before. Write at least one sentence for each waveform. Take a screenshot of the waveform and paste with your sentence. The SQUARE wave is done as an example:

Available Waveforms:
SQUARE
PHASOR
QUARTERPULSE
SAW
SINE
TRIANGLE

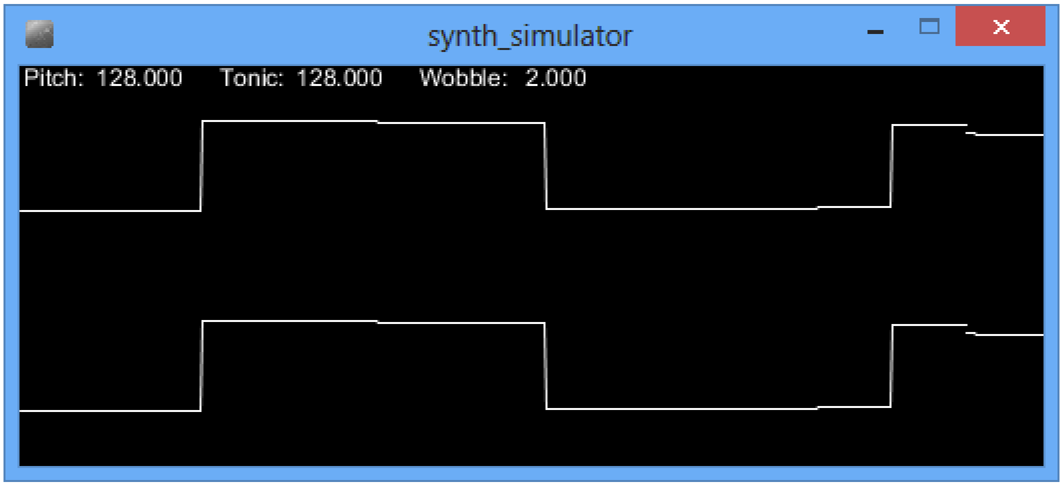
Example Code for Square: Line 33



```
// Create wave oscilators
// Available Wavforms:  PHASOR, QUARTERPULSE, SAW, SINE, SQUARE, TRIANGLE
vco = new Oscil(pitch, 0.7f, Waves.SQUARE); // SQUARE
vco2 = new Oscil(pitch, 0.1f, Waves.SAW); // SAW
```

Wave Descriptions: (Complete this Section)

SQUARE: The SQUARE waves sounds edgy and somewhat hollow. The waveform looks like blocks of shapes with ½ being high and ½ being low.



```
Pitch:  128.000    Tonic:  128.000    Wobble:  2.000
```

PHASOR:

QUARTERPULSE:

SAW:

SINE:

TRIANGLE:

3. Save this document and upload to your Google Site Processing Page as at attachment.

Part 3: Installing the Arduino Software and working with Arduino Board

1. Go to the following Link:
http://www.nebomusic.net/processinglessons/Directions_Arduino_Synth_Controller.pdf

2. Follow directions for Sections 2, 3, and 4.

3. Connect Arduino Board to your computer and use it to drive the Synth Program as shown in the directions. Take a picture of the Arduino board connected to your computer while the synth program is running.

Requirements: (100 Points)

1. Complete the Lab as directed and upload this document to your Google Site Processing Page as “Lastname Processing Synth”.
(85 Points)

2. Upload your ‘Lastname_Synth’ Processing Code to your Google Site Processing Page.
(5 Points)

3. Complete the *Sections 2, 3, and 4* of the directions at http://www.nebomusic.net/processinglessons/Directions_Arduino_Synth_Controller.pdf and take a picture of the Arduino Board connected to your computer while running the code from that lesson. Upload this picture to your Google Site Processing Page.
(10 Points)