

MUSIC 208 Homework 7-8

This project has four steps.

1. metronom.ck

Create a metronome class using the Impulse with the BiQuad filter from Lab 8 to create the clicks. Beat one in every measure will be louder and use different pitch/timbre to distinguishing it from the other beats. Choose sounds that would make good metronome background clicks and are not annoying. There can be any number of beats in a measure and that number will not change once the metronome is running.

2. metroControls.ck

Add keyboard controls to the metronome. Repurpose the keyboard control code from Twinkle in Lab 8 and paste it after the end of the Metronome class. Delete. Use the up/down arrow keys to increase/decrease the speed of the beats. Add space bar code to toggle the metronome sound on and off.

3. twinkleComplete.ck

Complete the Twinkle Twinkle Little Star song so the entire song plays.

4. twinkleMetro.ck

Copy the metronome code, the keyboard code and the Twinkle code to twinkleMetro.ck. Modify the Twinkle code to use the metronome beat instead of the global variables used in Lab 8. Test and debug until it works.

Add comments to your code so I can follow what you're trying to do.

Turn In These Four Chuck Files

You'll turn them in to the course Hand-in folder.

1. metronome.ck

2. metroControls.ck

3. twinkleComplete.ck

4. twinkleMetro.ck

1. metronome.ck

Download homework78Files.zip from the course web page for Unit 8.

Save metronomeTemplate.ck as metronome.ck.

Create a metronome class using the Impulse with the BiQuad filter from Lab 8 to create the clicks. Beat one in every measure will be louder and use different pitch/timbre to distinguishing it from the other beats. Choose sounds that would make good metronome background clicks and are not annoying. There can be any number of beats in a measure and that number will not change once the metronome is running.

Console Monitor Output

This is what you should see when it's working. Tempo is 60 with 4 beats per measure. Measure and beat numbers are zero based.

```
44100.000000 60.000000 0 0
44100.000000 60.000000 0 1
44100.000000 60.000000 0 2
44100.000000 60.000000 0 3
44100.000000 60.000000 1 0
44100.000000 60.000000 1 1
44100.000000 60.000000 1 2
44100.000000 60.000000 1 3
44100.000000 60.000000 2 0
44100.000000 60.000000 2 1
44100.000000 60.000000 2 2
44100.000000 60.000000 2 3
44100.000000 60.000000 3 0
```

The above output is the result of this code statement.

```
<<< getBeatDuration(),getTempo(),getMeasNum(),getBeatNum() >>>;
```

2. metroControls.ck

Download homework78Files.zip from the course web page for Unit 8.

Save metroControlsTemplate.ck as metroControls.ck.

Add keyboard controls to the metronome. Repurpose the keyboard control code from Twinkle in Lab 8 and paste it after the end of the Metronome class. Delete. Use the up/down arrow keys to increase/decrease the speed of the beats. Add code for the space bar to toggle the metronome clicks on and off.

Use the kb.ck example to find the msg.which values for the up arrow key, down arrow key and the space bar.

Copy/paste the keyboard controls section of the following code after the end of the Metronome class in the new metroControls.ck file.

Save As metroControls.ck.

Test and Debug metroControls.ck

You should be hear the metronome respond to up and down arrow key tempo changes. The space bar should toggle the sound on and off. Add this code a

3 twinkleComplete.ck

Open a new twinkleComplete.ck file. Complete the Twinkle Twinkle Little Star song so the entire song plays. Copy the Twinkle code from Lab 8 into a new file called twinkleComplete.ck. You'll need to add measures 5 to 12 to finish the song.

Musical notation for measures 1-4 of Twinkle Twinkle Little Star. The piece is in 4/4 time. The treble clef staff contains the melody, and the bass clef staff contains the accompaniment. Chord labels are provided below each note.

Measure	Treble Clef Notes	Bass Clef Notes
1	C4, G4	C3, C3
2	A4, G4	F2, G2
3	F4, E4	D3, C3
4	D4, C4	G2, C3

5

Musical notation for measures 5-8 of Twinkle Twinkle Little Star. The piece is in 4/4 time. The treble clef staff contains the melody, and the bass clef staff contains the accompaniment. Chord labels are provided below each note.

Measure	Treble Clef Notes	Bass Clef Notes
5	G4, F4	E3, D3
6	E4, D4	C3, G2
7	G4, F4	E3, D3
8	E4, D4	C3, G2

9

Musical notation for measures 9-12 of Twinkle Twinkle Little Star. The piece is in 4/4 time. The treble clef staff contains the melody, and the bass clef staff contains the accompaniment. Chord labels are provided below each note. A double bar line is present at the end of measure 12.

Measure	Treble Clef Notes	Bass Clef Notes
9	C4, G4	C3, G2
10	A4, G4	F2, G2
11	F4, E4	D3, C3
12	D4, C4	G2, C3

4 twinkleMetro.ck

Open a new twinkleMetro.ck file. Copy the metronome code, the keyboard code and the finished Twinkle code into twinkleMetro.ck. Modify the Twinkle section from step 3 to use the metronome beat instead of the global variables beat used in Lab 8. Test and debug until it works.

Twinkle should play in sync with the metronome clicks. The up arrow should increase the tempo. The down arrow should decrease the tempo. The space bar should toggle the metronome sound on and off and Twinkle should keep playing.

twinkleMetro.ck Code Outline

```
// twinkleMetro.ck
// FROM STEP 1
class Metronome
{
    // a lot of code
} // end Metronome class

// create metro variable
Metronome metro;
metro.init( 100.0, 4 );

// FROM STEP 2
// Standard Keyboard stuff
Hid hi;
HidMsg msg;
// change for your keyboard, these are Mac codes
82 => int upArrowCode;
81 => int downArrowCode;
44 => int spaceBarCode;

// which keyboard
0 => int device;
// get from command line
if( me.args() ) me.arg(0) => Std.atoi => device;

// open keyboard (get device number from command line)
if( !hi.openKeyboard( device ) ) me.exit();
<<< "keyboard '" + hi.name() + "' ready", "" >>>;

function void handleKeyboard()
{
    // handle msg.isButtonDown() for upArrowCode, downArrowCode,
```

```
and spaceBarCode
}

// FROM STEP 3
// complete Twinkle song goes here

// STEP 4 make it work
spork ~ metro.doClicks();
spork ~ handleKeyboard();
spork ~ playClarinetMelody();
spork ~ playMoogieBass();

3::minute => now;
```