Unit 6: Embedded Applications with MUzECS
Introduction

MUzECS is an alternative final module for the high-school level ECS curriculum. It uses the Arduino Leonardo board along with a customized block-based programming language and custom components to allow for an effective curriculum while remaining cost-effective. This curriculum was built using the Computer Science Teachers Association (CSTA) standards, which help focus the curriculum, as well as making it easier to adopt in a standards-focused education environment. Specifically, it uses standards 5.3.A.TR, 5.3.A.CL, 5.3.A.CPP, and 5.3.A.CI, which can be found in the standards handbook on the CSTA website [1]. This is a comprehensive and low-cost alternative to the ECS module 6, which will help make accessible the final module to schools where it was previously cost-prohibitive.

This module provides a physical application for the skills developed earlier in this course. MUzECS’s Ardublock-based drag-and-drop programming language has been designed to be easily understood by students who have previously worked with Scratch. Each hardware unit will consist of an Arduino Leonardo board and one of our custom circuitboard “shields”. Students will share hardware units (and computers, in order to program the Leonardo), allowing them to further develop both teamwork and programming skills.

The unit will begin with a discussion of how computers (specifically, embedded systems) have been so firmly established in our daily lives, in order to demonstrate to students just how important computers are to our society today. Students will be encouraged to discuss topics such as:

- Situations in which computers can potentially be better suited for a task than humans
- Applications of computers other than traditional “personal computers” and smartphones
- Potential hazards of ubiquitous computers
- Challenges, benefits, and downsides of non-PC computers

The unit consists of four main sections:

- Introduction to the concept of non-PC computers and “embedded systems” (Days 1-3)
- Algorithm development (Days 4-5)
- Introduction to Ardublock, Arduino Leonardo, and Shield (Days 6-10)
- Projects (Days 11-28)

Throughout the course of this unit, the instructor will need to assist students when the Ardublock software differs from Scratch.

Specifics for each instructional day are listed in the overview chart on the next page.

Special thanks to Robert Juranitch for his help in starting the curriculum design process, as well as the ECS team for developing an excellent foundation to build from.

References:

### Daily Overview Chart

<table>
<thead>
<tr>
<th>Instructional Day</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>What is an “Embedded System”?</td>
</tr>
<tr>
<td>4-5</td>
<td>Develop algorithmic thinking skills by designing a Tic-Tac-Toe algorithm</td>
</tr>
<tr>
<td>6-7</td>
<td>Introduction to the Arduino Leonardo and the shield</td>
</tr>
<tr>
<td>8-10</td>
<td>Introduction to Ardublock software and tutorials</td>
</tr>
<tr>
<td>11-15</td>
<td>Chip Tunes project</td>
</tr>
<tr>
<td>16</td>
<td>Recognize importance of computers in modern music</td>
</tr>
<tr>
<td>17-19</td>
<td>Theremin project</td>
</tr>
<tr>
<td>20-28</td>
<td>Piano project</td>
</tr>
</tbody>
</table>
Daily Lesson Plans

Instructional Day: 1-3

Topic Description: What is an “Embedded System”? 

Objectives:

Students will be able to:

- Identify situations where computing might be better than “doing it by hand”.
- Evaluate when and where to use computers.
- Provide a basic definition of an “embedded system”.

Outline of the Lesson:

- Brainstorm examples of non-PC computers (20 minutes)
- Small group discussion about why computers are used in those examples (15 minutes)
- Watch the “Day of Glass” video (5 minutes)
- Students volunteer some places they might want to introduce computers in their own lives (15 minutes)
- Small group discussion of the benefits of having computers in those places (15 minutes)
- Small group discussion on the challenges of getting computers in those places (15 minutes)
- Small group discussion on the potential hazards & downsides of computers everywhere (15 minutes)
- Whole group discussion: What is an “embedded system”? (15 minutes)
- Students volunteer examples of embedded systems (15 minutes)
- Small group discussion on differences between “computer” and “embedded system” (15 minutes)
- Students write down personal, simple definitions of what an embedded system is to each of them (10 minutes)

Student Activities:

- Brainstorm examples of computers beyond personal computers, smartphones, and tablets.
- Students use their imaginations for how they could implement computers in their own lives, guided by the video.
- Small group discussions on the benefits, challenges, and potential hazards of using non-PC computers.
- Brainstorm examples of embedded systems.
- Students discuss what makes a system either a “computer” or an “embedded system”, identify examples of embedded systems, and determine what the concept of an embedded system means to them.

Teaching/Learning Strategies:

- Brainstorm: Ask students about non-“personal computer” computers. Responses may include:
- Industrial computers, such as assembly line controls
- Electronic instruments, such as an electronic keyboard
- Robots
- Embedded systems, such as an electronic thermostat or a traffic light controller
- Complex electromechanical systems controlled by computers, such as airplanes

- Ask students why they think that computers are used in those examples. Answers might include:
  - Computer can do some things easier, more consistently, or more repetitively than humans.
  - Computers can make things smaller, lighter, or more portable; for example, a keyboard is easier to wield than a full piano.
  - Computers can do things more safely, as in the example of a bomb disposal robot.
  - Computer can do things more cheaply, as in the example of an ATM vs. a full-time teller.

- Watch the “Day of Glass” video.

- Encourage students to volunteer situations in which they could implement computers in their own lives. The answers need not be realistic; the purpose is to spur discussion on the benefits and challenges of using computers. Some answers might include:
  - The Jetsons, or other home automation as seen in the “Day of Glass” video
  - Self-driving cars
  - Various forms of robotics

- Lead the discussion on the benefits of computers. Some benefits to keep in mind are:
  - Computers help reduce work
  - Computers can make things cheaper and safer
  - Automation can simply be cool

- Lead the discussion on potential challenges. Several that may be suggested:
  - Computers can be expensive to start with
  - Computers cannot solve aesthetic problems nearly as well as humans
  - Computers require considerable technical knowledge and effort, between programming and maintenance

- Ask students about potential downsides of having computers everywhere. Examples include:
  - Hacking and other security vulnerabilities
  - System bugs and faults
  - Be certain to emphasize that these benefits and challenges should be weighed against each other when considering if a computer should be used to solve a problem (encourage students to do basic “cost-benefit analyses”).

- Guide a discussion on what an embedded system actually is. Some example aspects:
  - Limited resources (cost, size, space, memory, speed, etc.)
  - Real-time operation and/or hard deadlines
  - Should run without constant attention from an operator (automated)
  - “Embedded” because it is usually part of a larger system
  - Dedicated to specific tasks

- Lead into small group discussion on what differentiates a computer from an embedded system.

- Have students each individually write down a simple, one to two sentence definition of what an embedded system is, as they see it.
Resources:

- Corning’s Day of Glass 2: [https://www.youtube.com/watch?v=jZkHpNnXLBO](https://www.youtube.com/watch?v=jZkHpNnXLBO)
- More “futuristic technology” commercials (there are countless commercials, these is just a sampling):
  - Toyota Fun-Vii: [http://www.youtube.com/watch?v=UkS29w0V0Wo](http://www.youtube.com/watch?v=UkS29w0V0Wo)
  - Coca-Cola: [http://www.ispot.tv/ad/7Y6D/coca-cola-futuristic-technology](http://www.ispot.tv/ad/7Y6D/coca-cola-futuristic-technology)
  - Droid: [http://www.youtube.com/watch?v=ArJ_mcDSrSo](http://www.youtube.com/watch?v=ArJ_mcDSrSo)
Instructional Day: 4-5

Topic Description: Tic-Tac-Toe Algorithms

Objectives:
Students will be able to:

- Write and debug an algorithm which describes a Tic-Tac-Toe strategy.

Outline of the Lesson:

- In pairs, students create a strategy algorithm for a human to play Tic-Tac-Toe (65 minutes)
- In their teams, students have a Tic-Tac-Toe tournament based on their algorithms (35 minutes)
- Group discussion on what algorithms worked, and why (10 minutes)

Student Activities:

- In pairs, students discuss what strategies exist to play Tic-Tac-Toe, figure out how to make these strategies systematic, and then test these strategies.
- The class has a Tic-Tac-Toe tournament, with each pair playing based on their own algorithm.
- Group discussion on why some algorithms worked better than others

Teaching/Learning Strategies:

- Organize students into pairs, and have them design an algorithm for playing Tic-Tac-Toe. This process starts with writing down a first draft of the algorithm. Then, each pair should test their algorithm, and determine where there were bugs in their first draft. With the revised algorithm, they continue to test and revise until they think they have a good strategy.
- Have each student pair play other student pairs in a best-of-3 single elimination tournament.
- Lead a discussion on which algorithms worked, and why some methods of revising algorithms may have worked better than others.

Resources:

- http://xkcd.com/832/ (for teacher interest; probably too in-depth for class)
- http://en.wikipedia.org/wiki/Game_tree (for teacher interest; probably too in-depth for class)
- Watson Playing Jeopardy: https://www.youtube.com/watch?v=WFR3IOm_xhE
- Deep Blue winning a chess game in 1997 by “learning” its opponent’s strategies
- Robot Soccer World Cup: https://www.youtube.com/watch?v=NbE4 davLuh8
Instructional Day: 6-7

Topic Description: Introduce the Arduino Leonardo and the shield

Objectives:

Students will be able to:

- Recognize and describe the components of the Leonardo and the shield
- Be able to describe how one might connect a generic component to the Leonardo

Outline of the Lesson:

- MUzECS Getting Started video (5 minutes)
- Explain what the Leonardo is (15 minutes)
- Go over the handout which describes the various components of the Leonardo (35 minutes)
- Discuss what components could be put on the Leonardo, and how (20 minutes)
- Explain the concept of the shield, and what all the components are (35 minutes)

Student Activities:

- Whole group discussion of the hardware that the students will be using for this unit

Teaching/Learning Strategies:

- Watch video introducing students to MUzECS
- Explain what the Arduino Leonardo is. Be sure to mention how this is only one of many types of microcontrollers, and relate this discussion back to the “what is an embedded system” discussion from before.
- Using the handout, go over the components of the Leonardo. This vocabulary can be tricky, especially if the students are unfamiliar with hardware, so be certain that students ask questions when they are unsure what they are looking at.
- Lead a discussion of components that could be attached to a simple board like this. Because this is many students’ first experience with hands-on electronics, be ready to prompt discussion with examples of your own. Some examples include:
  - Speaker
  - Motors
  - Lights
  - Buttons/Switches
  - Sensors
- Explain what a shield is, and go over the components of the shield used for this unit.

Resources:

- MUzECS “Getting Started” video
- “Hardware Setup” Teacher Resource
- “Component Identification” Handout
Instructional Day: 8-10

Topic Description: Introduce Ardublock software and tutorials

Objectives:
Students will be able to:

- Write a program in Ardublock
- Upload and run an Ardublock program on the Leonardo

Outline of the Lesson:

- Discuss differences between Scratch & Ardublock (25 minutes)
- MUzECS “Deploying Code” Video & “MUzECS and Arduino Code” Video (5 minutes)
- Run a sample program on the Leonardo (10 minutes)
- Run first tutorial (15 minutes)
- Run second tutorial (25 minutes)
- Run third tutorial (30 minutes)
- Run fourth tutorial (25 minutes)
- Run fifth tutorial (30 minutes)

Student Activities:

- In pairs or small groups:
  - Upload the sample program to the Leonardo, and watch it run
  - Using the tutorials, write the demo programs, and run them on the Leonardo

Teaching/Learning Strategies:

- Explain that Ardublock is very similar to Scratch, but that programs written in Ardublock will control the Leonardo and the peripherals on the shield used for this unit
  - Distribute the “Block Types” Handout
- Divide the students into pairs (or groups), and guide them through opening a sample program and running it on the Leonardo.
- Have the students, in the same groups as before, follow the instructions in the five tutorials. Assist as necessary.

Resources:

- Student Resources
  - MUzECS “Deploying Code” Video
  - “MUzECS and Arduino Code” Video
  - “Block Types” Handout
  - Sample program (blinking the LED, same as tutorial 1)
  - Tutorial 1: “Blinking the Lights”
  - Tutorial 2: “Playing Notes”
  - Tutorial 3: “Using the Buttons”
  - Tutorial 4: “Using the Distance Sensor”
• Tutorial 5: “Using the Keyboard”
  o Advanced Handouts (for students who are curious/could use extra challenge)
    ▪ “Advanced Pins” Handout
    ▪ “Arduino Code” Handout
• Teacher Resources
  o “List of Note Names”
  o “Software Setup”
  o “Helping with Error Messages”
• Additional Resources
  o Ardublock: http://blog.ardublock.com/
Instructional Day: 11-15

Topic Description: Project 1 – Chip Tunes

Objectives:

Students will be able to:

- Write and run a program on the Leonardo which plays a song on the shield’s speaker

Outline of the Lesson:

- Explanation of project guidelines (15 minutes)
- Design and write a chip tune (200 minutes)
- Present chip tunes (40 minutes)
- Reflection (20 minutes)

Student Activities:

- In pairs or small groups:
  o Agree on which song to write
  o Outlines where control structures, such as loops, could be used
  o Write a first draft of the program
  o Test and improve the program
- Participate in song presentations to whole class
- Complete the project reflection

Teaching/Learning Strategies:

- Hand out requirements and rubric, taking time to answer questions.
- Keep students on task and asking questions when necessary.
- After each group’s presentation, ask what they found helpful in terms of control structures
- Have the students complete the project reflections individually

Resources:

- Chip Tunes Project Description
- Chip Tunes Project Reflection
- Chip Tunes Project Sample Rubric
- Famous chip tune examples:
  o Mario theme song: [https://www.youtube.com/watch?v=mnipB_8Br8U](https://www.youtube.com/watch?v=mnipB_8Br8U)
  o Tetris theme song: [https://www.youtube.com/watch?v=NmCCQxVBfyM](https://www.youtube.com/watch?v=NmCCQxVBfyM)
  o Teenage Mutant Ninja Turtles theme song: [https://www.youtube.com/watch?v=Sd87aUe78Rk](https://www.youtube.com/watch?v=Sd87aUe78Rk)
- “Sample Project 1 – Chip Tunes” Teacher Resource
- “Project 1” Videos
Chip Tunes Project Description

Chip tunes (8-bit music) are a form of electronic music made by programming a simple song into a computer, which plays the music on a speaker. Some well-known examples of chip tunes are the original Mario theme song, the Tetris theme song, and other themes from older video games.

- **Task:**
  - Using Ardublock, write a chip tune, upload it to the Leonardo, and play it on the shield’s speaker.

- **Requirements:**
  - The program should not take any input, and should only output audio on the speaker.
  - The program should be autonomous: After starting the chip tune, it should run to completion on its own.
  - The song should be appropriate for school.
  - The chip tune should resemble a well-known song, or should be a complete original song.
  - The program should use control structures, such as loops or if-then statements.
  - Students must be respectful of each other and other students’ chip tunes.

- **Process:**
  - Decide amongst your group what song you want to write.
  - Start exploring the “Play Note” and “Play Note Time” blocks.
  - Write the code for your song.
  - Test and revise your song.
  - Show off your song!

- **Grading:**
  - Quality and duration of your chip tune
  - Programming complexity, such as use of conditionals and loops
  - Cooperation with your group and other groups

You will have three and a half class periods to design, build, and test your chip tune, and you will present it on the fifth day.
**Chip Tunes Project Reflection**

For each member in your group, evaluate their performance as a team member:

<table>
<thead>
<tr>
<th>Name:</th>
<th>Circle one word to describe his or her performance:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Excellent</td>
</tr>
</tbody>
</table>

Why? ____________________________________________________________

<table>
<thead>
<tr>
<th></th>
<th>Excellent</th>
<th>Good</th>
<th>Average</th>
<th>Poor</th>
</tr>
</thead>
</table>

Why? ____________________________________________________________

<table>
<thead>
<tr>
<th></th>
<th>Excellent</th>
<th>Good</th>
<th>Average</th>
<th>Poor</th>
</tr>
</thead>
</table>

Why? ____________________________________________________________

<table>
<thead>
<tr>
<th></th>
<th>Excellent</th>
<th>Good</th>
<th>Average</th>
<th>Poor</th>
</tr>
</thead>
</table>

Why? ____________________________________________________________

What was your favorite thing about this project?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

If you could do this project over, what would you do differently?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________
# Chip Tunes Project Sample Rubric

<table>
<thead>
<tr>
<th>Extra Credit</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Song Complexity</strong></td>
<td>The song is clearly recognizable, and plays for the full duration of the original.</td>
<td>The program clearly plays an enjoyable song.</td>
<td>The program produces a melody, but it is short or cacophonous.</td>
<td>Notes play, but they are simple and do not form a melody.</td>
</tr>
<tr>
<td><strong>Programming</strong></td>
<td>The program is extremely easy to read, and uses a number of control blocks to make the code efficient.</td>
<td>The program is legible, and uses control blocks.</td>
<td>The program is somewhat difficult to read or inefficient, and could benefit from control structures.</td>
<td>The program is slipshod, and demonstrates a lack of effort. It has no control structures.</td>
</tr>
<tr>
<td><strong>Cooperation</strong></td>
<td>Student(s) helped other groups.</td>
<td>Student worked well with group, and actively participated in all parts of the project.</td>
<td>Student worked somewhat well with group, and participated in most parts of the project.</td>
<td>Student had trouble working with the group, and participated in few parts of the project.</td>
</tr>
</tbody>
</table>
Instructional Day: 16

Topic Description: Music and Computers

Objectives:

Students will be able to:

- Recognize how some contemporary music is made using computers
- Recognize the role of computers in electronic instruments

Outline of the Lesson:

- Whole group discussion on electronic music as a whole (35 minutes)
- Watch an example of purely electronic music (5 minutes)
- Watch an example of an electronically enhanced music (5 minutes)
- Watch an example of a Theremin and discuss concept (10 minutes)

Student Activities:

- Whole group discussion on electronic music. This discussion should cover:
  - Purely electronic music (dubstep, for example)
  - Computer-modified music (auto-tune, remixing, etc.)
  - Electronic instruments (synthesizer, Theremin, etc.)

Teaching/Learning Strategies:

- Lead the discussion on electronic music. Be sure to mention that there are many forms of computer-generated music, from chip tunes (like the students had made in the first project), to computer-enhanced music (such as auto-tune), to purely electronic music.
- Have the students watch an example of purely electronic music
- Have the students watch an example of electronically enhanced music
- Have the students watch an example of a Theremin and lead a brief discussion of what a Theremin is to prepare them for the next project

Resources:

- Electronic music samples (most modern dance music applies):
  - Darude – Sandstorm: [http://www.youtube.com/watch?v=2HQaBWziYvY](http://www.youtube.com/watch?v=2HQaBWziYvY)
  - Skrillex – Summit: [http://www.youtube.com/watch?v=OR6AV9yJPoM](http://www.youtube.com/watch?v=OR6AV9yJPoM)
  - Faunts – Das Malefitz: [http://www.youtube.com/watch?v=27yRNPEQ5q8](http://www.youtube.com/watch?v=27yRNPEQ5q8)
  - Deadmau5 – I Remember: [http://www.youtube.com/watch?v=zK1mL1eXwsQ](http://www.youtube.com/watch?v=zK1mL1eXwsQ)
  - Calvin Harris – Summer: [http://www.youtube.com/watch?v=McETlqoRkk](http://www.youtube.com/watch?v=McETlqoRkk)
  - Blake Lewis – Binary Love: [http://www.youtube.com/watch?v=qBFQmGvWjEc](http://www.youtube.com/watch?v=qBFQmGvWjEc)

- Computer-modified music samples (much of modern pop music applies):
  - Sia – Clap Your Hands: [http://www.youtube.com/watch?v=HLA0ofsu0Qg](http://www.youtube.com/watch?v=HLA0ofsu0Qg)
  - Justin Timberlake – What Goes Around...Comes Around: [http://www.youtube.com/watch?v=IC8qPpnD0uE](http://www.youtube.com/watch?v=IC8qPpnD0uE)
  - Ariana Grande – Problem: [http://www.youtube.com/watch?v=iS1g8G_njx8](http://www.youtube.com/watch?v=iS1g8G_njx8)
- Pharrell Williams – Happy: http://www.youtube.com/watch?v=y6Sxv-sUYtM
- Electronic instruments samples (many live concerts apply):
  - Theremin Performance: http://www.youtube.com/watch?v=nJYho56lINKU
  - Synthesizer Performance: http://www.youtube.com/watch?v=uvDfHhuFF5w
  - Calabria Remix Live: http://www.youtube.com/watch?v=6pfr39FAuRY
- TED Talk: http://www.ted.com/talks/ge_wang_the_diy_orchestra_of_the_future
**Instructional Day:** 17-19

**Topic Description:** Project 2 – Theremin

**Objectives:**

Students will be able to:

- Create a Theremin using the Leonardo and shield

**Outline of the Lesson:**

- Explanation of project guidelines and rubric (10 minutes)
- Design and program Theremin (125 minutes)
- Demonstrate the Theremin (15 minutes)
- Reflection (15 minutes)

**Student Activities:**

- In pairs or small groups:
  - Develop an initial design for the Theremin program
  - Write the initial program
  - Test, debug, and refine the program
- Demonstrate the final product to class
- Complete project reflection

**Teaching/Learning Strategies:**

- Divide students into groups, hand out project description and rubric
- Circulate and answer questions
- View students’ Theremin demonstrations
- Have students individually complete project reflections

**Resources:**

- Theremin Project Description
- Theremin Project Reflection
- Theremin Project Sample Rubric
- TED Talk: [https://www.ted.com/talks/pamelia_kurstin_plays_the_theremin](https://www.ted.com/talks/pamelia_kurstin_plays_the_theremin)
- “Sample Project 2 – Theremin” Teacher Resource
- “Project 2” Videos
Theremin Project Description

A Theremin is an electronic instrument that is played using two distance sensors, one of which controls the volume, and the other controls the frequency (or note) that is being played. For this project, you will be constructing a simplified Theremin that does not have a volume sensor, and only a frequency sensor.

- **Task:**
  - Using Ardublock, write code for a Theremin, upload it to the Leonardo, and play it for the class.

- **Requirements:**
  - The program should only take input from the distance sensor, and should only output on the speaker.
  - The program should be autonomous: After starting the program, it should run on its own save for input via the distance sensor.
  - The program should use control structures, such as loops or if-then statements.
  - Students must be respectful of each other and other students’ Theremins.

- **Process:**
  - Start exploring the “Get Distance” block.
  - Write the code for the Theremin.
  - Test and revise your code.
  - Show off your Theremin!

- **Grading:**
  - Quality, consistency, and responsiveness of the Theremin
  - Programming, such as use of conditionals and loops
  - Cooperation with your group and other groups

You will have two and a half class periods to design, build, and test your Theremin, and you will present it at the end of the third day.
**Theremin Project Reflection**

For each member in your group, evaluate their performance as a team member:

<table>
<thead>
<tr>
<th>Name:</th>
<th>Circle one word to describe his or her performance:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Excellent  Good  Average  Poor</td>
</tr>
</tbody>
</table>

**Why?**

<table>
<thead>
<tr>
<th>Name:</th>
<th>Circle one word to describe his or her performance:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Excellent  Good  Average  Poor</td>
</tr>
</tbody>
</table>

**Why?**

<table>
<thead>
<tr>
<th>Name:</th>
<th>Circle one word to describe his or her performance:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Excellent  Good  Average  Poor</td>
</tr>
</tbody>
</table>

**Why?**

<table>
<thead>
<tr>
<th>Name:</th>
<th>Circle one word to describe his or her performance:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Excellent  Good  Average  Poor</td>
</tr>
</tbody>
</table>

**Why?**

What was your favorite thing about this project?

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

If you could do this project over, what would you do differently?

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

|                                           |
|                                           |
|                                           |
|                                           |
## Theremin Project Sample Rubric

<table>
<thead>
<tr>
<th>Extra Credit</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Responsiveness</strong></td>
<td>The Theremin reacts quickly, and reacts to small changes in distance from the sensor to the hand.</td>
<td>The Theremin is responsive, and reacts to changes in distance.</td>
<td>The Theremin has some lag time between movement of the hand and a change in pitch.</td>
<td>There is significant lag time between movement of the hand and a change in pitch.</td>
</tr>
<tr>
<td><strong>Sound Quality</strong></td>
<td>The Theremin plays clear, musical tones.</td>
<td>The Theremin plays clearly distinct tones.</td>
<td>The tones are somewhat muddled.</td>
<td>The tones are not at all distinct, and might be better described as noise.</td>
</tr>
<tr>
<td><strong>Programming</strong></td>
<td>The program is extremely easy to read, and uses a number of control blocks to make the code efficient.</td>
<td>The program is legible, and uses control blocks.</td>
<td>The program is somewhat difficult to read or inefficient, and could benefit from control structures.</td>
<td>The program is slipshod, and demonstrates a lack of effort. It has no control structures.</td>
</tr>
<tr>
<td><strong>Cooperation</strong></td>
<td>Student(s) helped other groups.</td>
<td>Student worked well with group, and actively participated in all parts of the project.</td>
<td>Student worked somewhat well with group, and participated in most parts of the project.</td>
<td>Student had trouble working with the group, and participated in few parts of the project.</td>
</tr>
</tbody>
</table>
Instructional Day: 20-28

Topic Description: Project 3 – Piano

Objectives:

Students will be able to:

- Create a piano using the Leonardo and shield

Outline of the Lesson:

- Explanation of project guidelines and rubric (15 minutes)
- Design and program piano (400 minutes)
- Demonstrate the piano (25 minutes)
- Project reflection (15 minutes)
- Reflection on unit and class (40 minutes)

Student Activities:

- In pairs or small groups:
  - Brainstorm an initial design for the piano software
  - Write the initial program
  - Test, debug, and refine the program
- Demonstrate the final product to class
- Complete project reflection
- Complete reflection on unit and class

Teaching/Learning Strategies:

- Divide students into groups, hand out project description and rubric
- Circulate and answer questions
  - If students find working with the buttons on the shield too easy, encourage them to use the computer’s keyboard as input.
- View students’ piano demonstrations
- Have students individually complete project reflections
- Have students complete the final reflection

Resources:

- Piano Project Description
- Piano Project Reflection
- Piano Project Sample Rubric
- Final Reflection
- “Sample Project 3 – Piano” Teacher Resource
- “Project 3” Video
Piano Project Description

In this project, you will be modelling a piano using the Leonardo and the shield. One handy feature of the Leonardo is that it can send and receive information to the computer while it is connected via USB. You will be using this feature to make your computer keyboard into a piano keyboard.

- **Task:**
  - Using Ardublock, write code for a piano, upload it to the Leonardo, and play it for the class.

- **Requirements:**
  - The program should only take input from the buttons on the shield, and should only output on the speaker.
  - The program should be autonomous: After starting the program, it should run on its own save for input via the buttons.
  - The program should use control structures, such as loops or if-then statements.
  - Students must be respectful of each other and other students’ pianos.

- **Process:**
  - Write the code for your piano.
  - Test and revise your code.
  - Show off your piano!

- **Grading:**
  - Quality, consistency, and responsiveness of the piano
  - Programming, such as use of conditionals and loops
  - Cooperation with your group and other groups

- **Extra Challenge:**
  - One handy feature of the Leonardo is that it can send and receive information to the computer while it is connected via USB. Use this feature to make your computer keyboard into a piano keyboard!
  - In order to accomplish this, you will need to make use of the “Keyboard Setup”, “KeyPressed”, and “Update” blocks.

You will have seven and a half class periods to design, build, and test your piano, and you will present it at the end of the eighth day.
Piano Project Reflection

For each member in your group, evaluate their performance as a team member:

Name: __________________________________________

Circle one word to describe his or her performance:

<table>
<thead>
<tr>
<th>Excellent</th>
<th>Good</th>
<th>Average</th>
<th>Poor</th>
</tr>
</thead>
</table>

Why? __________________________________________

Name: __________________________________________

Circle one word to describe his or her performance:

<table>
<thead>
<tr>
<th>Excellent</th>
<th>Good</th>
<th>Average</th>
<th>Poor</th>
</tr>
</thead>
</table>

Why? __________________________________________

Name: __________________________________________

Circle one word to describe his or her performance:

<table>
<thead>
<tr>
<th>Excellent</th>
<th>Good</th>
<th>Average</th>
<th>Poor</th>
</tr>
</thead>
</table>

Why? __________________________________________

What was your favorite thing about this project?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

If you could do this project over, what would you do differently?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
### Piano Project Sample Rubric

<table>
<thead>
<tr>
<th></th>
<th>Extra Credit</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Successful Solution</strong></td>
<td>The piano can be successfully played from the USB keyboard.</td>
<td>The piano can be played from the buttons with no erratic behavior, and features a range of notes.</td>
<td>Each button on the shield corresponds to a distinct note being played.</td>
<td>Button input corresponds to sound playing from the speakers.</td>
<td>The piano cannot be played.</td>
</tr>
<tr>
<td><strong>Programming</strong></td>
<td>The program is extremely easy to read, and uses a number of control blocks to make the code efficient.</td>
<td>The program is legible, and uses control blocks.</td>
<td>The program is somewhat difficult to read or inefficient, and could benefit from control structures.</td>
<td>The program is slipshod, and demonstrates a lack of effort. It has no control structures.</td>
<td>The program does not work.</td>
</tr>
<tr>
<td><strong>Cooperation</strong></td>
<td>Student(s) helped other groups.</td>
<td>Student worked well with group, and actively participated in all parts of the project.</td>
<td>Student worked somewhat well with group, and participated in most parts of the project.</td>
<td>Student had trouble working with the group, and participated in few parts of the project.</td>
<td>Student did not participate in project, and sabotaged others’ work.</td>
</tr>
</tbody>
</table>
Final Reflection

What was your favorite thing about this unit?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

What was your favorite thing about this class?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

What would you like to do differently in this unit?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

What would you like to do differently in this class?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Do you have any other comments or suggestions for this class in the future?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________