FIRIA LABS

MISSION 11: Spirit Level		Time: 50 minutes
Overview: In this project students will create a digital level using the CodeX's built-in accelerometer and LCD display. Physically rotating the CodeX will move the digital "bubble" they create on the display, with code! This is more than just a fun project – it's a useful tool with practical applications. Students can discuss or compare a physical level with their digital level.		**Objectives:** ● I can use comments to explain and document the purpose of each line of code. ● I can use variables to calculate and convert measurements.
Standards: **2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. **3A-AP-17** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. **3A-IC-26** Demonstrate ways a given algorithm applies to problems across disciplines.	**CSP Framework:** **Computational Thinking Practices:** 3.B Use abstraction to manage complexity in a program. 4.C Identify and correct errors in algorithms and programs, including error discovery through testing. 6.A Collaborate in the development of solutions.	**Key Concepts:** ● Meet the accelerometer. There's one in your cell phone, and in many other devices we use. ● Make sure students read the accelerometer toolbox entry! ● Convert units generated by the accelerometer into degrees. ● Use a bit o' math for scaling the degrees to a range suitable for the moving "bubble".
Preparation: **Bring a physical level** to show to students and discuss how it works **Make a copy** of the assignment or put it in the LMS. **Prepare** formative assessments	**Links:** ● Assignment ● Reflection form ● Mission 11 - basic ● Mission 11 - with extensions ● Mission 11 - with functions	**Agenda:** ● Warm-up (5 minutes) ● Mission 11 (30 minutes) ● Extensions (10 minutes) ● Wrap-up (5 minutes)
Vocabulary: ● **Accelerometer:** A device that measures proper acceleration; a sensor chip that detects motion, impacts, and orientation ● **Tuple:** An *immutable* sequence of items that you can access with an *index*, or a list with values that don't change. A read-only version of a list.		
Assessment: ● Daily reflection journal or form ● Completion of assignment and mission ● Exit ticket on vocabulary ● Group review on vocabulary		

Teaching Guide

Warm-up (5 minutes)

The actual coding part of this Mission is about one normal class period. The extensions for this mission are fairly easy and don't take much time. Encourage the students to do them if they have time. Three challenges are included only if some students work very quickly. The first one is fairly easy and is good for students to try if there is time. The other two challenges are OPTIONAL!

👫 **Discuss** – Use a discussion strategy, like journaling, working at boards, selecting random students, or a form of think-pair-share.

- **Topic:** Discuss how a real *mechanical* **spirit level** works. The "spirit" is a liquid with space for a bubble, which will be in the center of the tube when it's in a **horizontal** position. Show them a physical, or mechanical, level.
- **Say:** Today you will make your CodeX into a digital spirit level. This is more than just a fun project – it's a useful tool with practical applications.

Activity – Mission #11 with extensions (30 minutes)

💻 Randomly group students into pairs for pair programming.

Students log in to one computer. Two computers can be used if they want to see instructions on one computer and work on the other computer. However, the assignment document requires snippets, so it will need to be open on the same computer as CodeSpace.

💡 **Teaching tip – Before they start:**

Optional: Review the Mission Reminders slides.

Important! Remind students that they need to document their errors and how they fixed them. There is a table at the end of the assignment document for this.

Students go to sims.firialabs.com and should be at the beginning of Mission 10

💡 **Teaching tip – Objective 1:**

The objective introduces the accelerometer. It returns three values for 3 dimensions – x, y and z. Students will work with x (and y in the challenges). There is a bit of reading for this objective so students understand what the accelerometer is and what it does, and the values it will return.

💡 **Teaching tip – Objective 2:**

Using the accelerometer in code.

💡 **Teaching tip – Objective 4:**

This objective introduces line and circle for the first time. Students can fill the screen with any color and then draw lines and circles on the screen. There is a bit of reading on this objective so students can understand how the simple graphics work and the orientation of the screen. Here are all the commands students could use. These will be used in a future lesson. (https://docs.firialabs.com/codex/canvas.html)

```
draw_line(x1, y1, x2, y2, color=(255, 255, 255))
```

```
draw_rect(x1, y1, width, height, color=(255, 255, 255))
```

```
draw_text(text, scale=1, color=(255, 255, 255), background=None, x=0, y=0)
```

```
fill(color)
```

```
fill_circle(x, y, radius, color=(255, 255, 255))
```

```
fill_rect(x1, y1, width, height, color=(255, 255, 255))
```

💻 **Teaching tip – Extension #1:**

Complete this mission if at all possible. It doesn't take very long. Students create a "wait" function like Mission 10 where the program waits until button A is pressed before becoming a level.

💻 **Teaching tip – Extension #2:**

Complete this mission if at all possible. It doesn't take very long. Students create a "kill switch" for the program. Also, encourage the students to include a message after the loop – not indented – that displays the program has ended.

💻 **Teaching tip – Challenge #1:**

This is completely optional. It is only for students who work very fast so they have something to work on throughout the class period. The spirit level only uses the x value. Change the code to use the y value. The line will need to be vertical instead of horizontal, and when drawing the circle, switch the x and y. Not much else is different.

💻 **Teaching tip – Challenge #2:**

This is completely optional. This challenge combines the original code with challenge #2. The level should draw a vertical and horizontal line and use both x and y values. (see code for Mission 11 with challenges)

💻 **Teaching tip – Challenge #3:**

This is more of a remix and can be done instead of challenge #1 and challenge #2. The students will write an if statement with logical operators to determine if the level is level. If so, draw a rectangle in the center instead of a circle. (see the code for Mission 11 with challenges) I used one long if statement, but students could do a nested if statement, or anything that works.

✅ Assignment is complete and ready to turn in. Both students should include their names on the document.

✅ Determine how you want to check-off the student program (turn in text file, submit through LMS, observe on student computer, etc.)

✅ **IMPORTANT!!**

Students should clear their CodeX by running their ClearCodeX program.

Wrap-Up (5 minutes)

👩‍🏫 **Vocabulary** – Review the vocabulary for today's lesson:

- **Accelerometer:** A device that measures proper acceleration; a sensor chip that detects motion, impacts, and orientation
- **Tuple (pronounced TOO-ple):** An *immutable* sequence of items that you can access with an *index*, or a list with values that don't change. A read-only version of a list.

👫 **Discuss** – Use a discussion strategy, like journaling, working at boards, selecting random students, or a form of think-pair-share.

> 👫 **REVIEW the TOPIC:** There isn't a specific real-world applications question to review with students. They could discuss what else the accelerometer might be used for. Or you can review what they learned, or debugging practices.

Formative Assessment:

- Daily reflection journal or form
- Completion of assignment and mission
- Programming journal
- Exit ticket on vocabulary
- Group review on vocabulary
- Students create a vocabulary canvas with vocabulary words.

SUCCESS CRITERIA:

- ❑ Display a numeric "tilt" value from the accelerometer.
- ❑ Scale the raw tilt value to show 0-9, indicating 0° to 90° incline.
- ❑ Replace the number display with a graphical bubble simulation!
- ❑ Use at least one function in your code.