Unit 1: Getting Started

Mission 3: Lightshow & Mission 4: Display Games

Intro and Discussion Points:

These missions explore the CodeX's LCD display and push-buttons. We're jumping in head-first with some real Python coding.

Discuss the following before getting started:

- Always start a new program by creating a new file and naming it appropriately. If you don't, you will lose all your previous work! Using descriptive file names is essential to finding the program later!
- You are making a project – not just working random problems. Focus on the *project-based* objectives and avoid rushing through the material too quickly.
- Test your understanding along the way by "coloring outside the lines". Try stuff!
- Collect all the Tools you find! (They're indicated with a wrench icon)
- Read carefully – usually the answer is right there in front of you!
- Code Tracing charts are used to document bugs and fixes. Documentation is not just for YOU, it's for the TEAM of people working on a project together. (Any time a student gets an error message or the code does not do what they believe it should, they should use a code tracing chart to write their corrections and the steps they took to fix the errors. Programmers go back to these quite often when they come to the same errors months down the road)
- Flowcharts are used to make a graphical representation of their code before they begin coding. Like a rough draft in writing.

Preparation and Materials:

Each student will need a CodeX, and a USB-C cable.

You may want to copy one example of the code tracing chart for each student, or have them duplicate it in their notebook.

CodeX Lesson Plans

UNIT 1 : Getting Started	**MISSION 3: Lightshow & MISSION 4: Display Games**	**# DAYS:** 2
UNIT GOALS: Students will learn the basics of Python.	**ADDITIONAL MATERIALS:** ● none	**VOCABULARY:** ● Data types (integer, string) ● Variable ● Conditional Statement

CSTA FOCUS STANDARDS: 1B-CS-01, 1B-CS-02, 1B-CS-03, 1B-AP-09, 1B-AP-10, 1B-AP-12, 1B-AP-15, 2-AP-11

LEARNING TARGETS:

- I can use the "Step" feature to debug a program.
- I can assign **data** to a **variable**.
- I can use **variables** to make code more efficient.
- I can explain the difference between ***integer*** and ***string*** data types and use each appropriately.
- I can convert between string and integer data types.
- I can write an **if:else** conditional statement.

SUCCESS CRITERIA:

- ❑ Show a sequence of Images on the CodeX''s LCD display.
- ❑ Change the animation speed of the Images.
- ❑ Display text message strings.
- ❑ Crunch some numbers with a micro-calculator.
- ❑ Program a push-button to make a fast-click game.

KEY CONCEPTS:

- Computers execute code in **sequential** steps, initially starting at the top of your file and proceeding down the page.
- The Codespace **debugger** lets you *step* through the code one line at a time to understand what the computer is doing.
- Built-in functions: `sleep()`, `display.show()`, `buttons.is_pressed(BTN_U)`
- Values have different **data types**, like *integer* and *string*.
- **Variables** can be defined to hold changing values.
- Branching with `if : else` statements.
- **Indenting** after a colon is very important!
- **Batteries** Included! The CodeX can make your code portable.

DISCUSS REAL WORLD APPLICATIONS:

The skills used in the project are used by professional software developers to build:

- Traffic Lights and People Counters
- Sports Event Scoreboards
- Games

ASSESSMENT STRATEGIES:

2.8 Checkpoint - use as entrance/exit slip

Remix suggestions (set aside 0.5-1 period to complete):

- Show your name on the display
- Make a 2-player game, where player A and B press their respective buttons, and the winner is decided by which player has the highest number of clicks. A neat idea here is to use positive and negative numbers, with one player counting up and the other counting down!
- Other starting or winning animation sequences.
- Next page has an example rubric and lesson plan for a remix

TEACHER NOTES:

Always refer to **Answer Keys by Mission** if you get stuck. All coding solutions are available, in alphabetical order.