FIRIA LABS

MISSION 9: Game Spinner

Time: 50-90 minutes

Overview:

Your game spinner will show a spinning arrow on the LCD display when you press Button A or B, and then slow down and stop in one of 8 random directions. The game spinner can be used for a variety of applications, such as choosing the next person to tell a story in a group of friends, navigating a crazy compass game, or deciding which pizza slice to eat next. It can provide an element for any game you create.

Objectives:

- I can apply properties and uses of an index to a new program.
- I can define and call a function.
- I can utilize multiple variables to a new program and describe their purposes.
- I can utilize loops to make my code more efficient.
- I can create simulations using computer hardware and software.

Standards:

2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.

3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

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:

- Logical operators like 'and', 'or', and 'not' allow your code to test for multiple conditions.
- Define functions to break complex code into smaller, easy to use (and re-use) pieces.
- Detect instantaneous button presses with button.is_pressed()
- Make your own animated sequences with loops and index variables.
- Loop for a determined number of iterations: while index < N

Preparation:

Make a copy of the assignment or put it in the LMS.
Prepare formative assessments

Links:

- Assignment
- Daily reflection form
- Mission 8 code

Agenda:

- Warm-up (5 minutes)
- Mission 9 (40 minutes)
- Wrap-up (5 minutes)

Vocabulary:

- **Logical Operator:** Operators that handle combinations of Boolean results; not, and, or
- **Function:** A named chunk of code you can run anytime just by calling its name; also called a procedure
- **Simulation:** Code that builds a *model* of something, and lets you play with that model. Simulations let you explore "virtual" situations, both realistic and imaginary, that might be difficult or impossible to do in the real world.
- **Parameter**: A local variable in a function that receives a value passed into the function when it is called; information the function needs to complete its task
- **Argument**: The value passed into a function – information the function needs to complete its task. An argument can be a literal value, a variable, or an expression.
- **Local variables**: Variables defined inside a function, and can only be used within that function.

Assessment:

- Daily reflection form
- Exit ticket or group discussion

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 only if a team works very quickly and has extra time. All of the extensions can be part of a remix, so do not go longer on this mission to do the extensions. They are OPTIONAL!

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

- **Topic:** There isn't a specific warm-up for this lesson. As a teacher, decide on a topic that may need review with your students before starting the mission. You may want to review lists and indexes. Or you can go over common programming errors. Spend five minutes or less doing a quick review and then get right to the mission.

Activity – Mission #9 (40 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 9

💡 **Teaching tip – Objective 1:**

The objective uses a built-in arrow list. If students don't read the instructions carefully, they will create the list. They do not need to! They should use their knowledge (and code if necessary) to display a random arrow from the built-in list.

💡 **Teaching tip – Objective 2:**

Students are asked to fill in a chart with three logical operators and what they are used for. They may need help with this; the information in the toolbox is kind of general. The students could give an example instead of a definition if that is what makes more sense to them.

Not	Used with only one value; gives the opposite	not(True) = False
And	Combines two Boolean values where both must be true for the expression to be true	True and False = False
Or	Combines two Boolean values where only one must be true for the expression to be true	True or False = True

💡 **Teaching tip – Objective 3:**

This objective is all about functions. This is a new function for CodeX. Students may have been exposed to functions in previous computer classes, but it is going to be new for this class. They will need to do a lot of reading about functions. Students can do the best they can with the questions. More assignments will follow this one that will go into depth on functions, so this isn't their only chance to understand them.

💡 **Teaching tip – Objective 5:**

This objective introduces more vocabulary: simulation, parameter, and argument. Formal definitions of parameter and argument are not given in the toolbox. You can help the students with a definition, or let them give examples instead of definitions.

💻 **Teaching tip – Extensions:**

There are three extensions included in the lesson, but they DO NOT need to be complete. They are included only in case some students work quickly and complete the mission before time is up. The mission has a lot of reading and thinking involved in it, so the mission should take an entire class period. All of the extensions can be included in future remix programs, or the Create PT, so don't extend the lesson to include the extensions. Just use as needed.

✅ 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.)

Wrap-Up (5 minutes)

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

- **Logical Operator:** Operators that handle combinations of Boolean results; not, and, or
- **Function:** A named chunk of code you can run anytime just by calling its name; also called a procedure
- **Simulation:** Code that builds a *model* of something, and lets you play with that model. Simulations let you explore "virtual" situations, both realistic and imaginary, that might be difficult or impossible to do in the real world.
- **Parameter**: A local variable in a function that receives a value passed into the function when it is called; information the function needs to complete its task
- **Argument**: The value passed into a function – information the function needs to complete its task. An argument can be a literal value, a variable, or an expression.
- **Local variables**: Variables defined inside a function, and can only be used within that function.

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

👫 **REVIEW the TOPIC:** After completion of this project, it's a great time to have a group of students gather in a circle to play with what they've created. Disconnect the CodeX and run it on batteries. Put it on the floor or a table in the middle of the group, so the random-arrow selects a student each time the button is pressed. Maybe the chosen one gets to name something different they could build using the CodeX and the code they've learned! Or talk about a debugging strategy or programming term.

Fast button inputs, animation, and simulation! Those are essential ingredients for lots of interesting applications:

- Video games
- Flight Simulators
- Virtual Reality

✅ **IMPORTANT!!**

Students should clear their CodeX by running their ClearCodeX program.

Formative Assessment:

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

SUCCESS CRITERIA:

- ❑ Display an Arrow in a random direction
- ❑ Detect an input- button A or B - to trigger the Arrow spin
- ❑ Animate an Arrow spinning around
- ❑ Make the Arrow gradually slow rather than stopping abruptly