MISSION 9: Game Spinner	Time: 60-90 minutes

Overview:

In this project, students will make a Game Spinner that can make a selection, just like a spinner in a game. The game spinner will show a spinning arrow on the LCD display when Button A or B is pressed, and then slow down and stop in one of 8 random directions.

Cross Curricular:

- Supports **language arts** through reflection writing.
- See **Mission 9 Remix** Lesson Prep for several cross curricular ideas and projects.

Materials Included in the learning portal Teacher Resources:

Mission 9 Slidedeck

The slide deck is for teacher-led instructions that let you guide students through the material using the slides. It is an alternative to the students reading a lot of instructions in CodeSpace. The slides mirror the instructions, with simplified language that is chunked into smaller sections at a time. The information is shown on slides with “Objective”. The tasks to complete are on slides with “Mission Activity".

Mission 9 Workbook

The workbook can be used instead of slides for student-led or independent work. It is an alternative to students reading a lot of instructions in CodeSpace. It mirrors the instructions (and the slide deck), with simplified language that is chunked into smaller sections at a time. Each objective is on its own page. The tasks to complete are labeled “DO THIS” and have a robot icon next to it.

Mission 9 Log (and answers)

This mission log is the worksheet for students to complete as they work through the mission. It should be printed and given to each student before the mission starts. They write on the mission log during the assignment and turn it in at the completion of the mission (assignment).

Mission 9 Lesson Plan

The lesson plan comes from the CodeX Teacher Manual and is included here for easy reference.

Mission 9 Remix Folder

Following Mission 9, students should complete a remix of their code. Get supplemental materials from the folder.

Additional Resources:

- **Mission 9 Solution (Game Spinner)** – in Answers section
- **Kahoot** (Mission 9)

Formative Assessment Ideas:

- Exit ticket
- Mission log completion
- Completed program
- Kahoot Mission 9 Review

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
- **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.
- **Control Variable**: A variable used in a condition that determines when a loop will end; must be incremented or changed inside the loop.

Preparing for the lesson:

Students will use the Codex throughout the lesson. Decide if they will work in pairs or individually.

- Look through the slide deck and workbook. Decide what materials you want to use for presenting the lesson. The slide deck can be projected on a large screen. The workbook (if used) can be printed or remain digital through your LMS.
- Be familiar with the Mission Log (assignment) and the questions they will answer.
- Print the Mission Log for each student.
- The mission program does not need to be portable. If you want students to use the CodeX without a cable, then have batteries available.

Lesson Tips and Tricks:

💡 **Teaching tip:**

You can use a variety of discussion strategies to get the most engagement from your students. For example, you can have students write their answers before asking anyone for an answer. You can use one of many think-pair-share methods. You can have students write their answer and share with someone, and then have other students share answers they heard from their peers. You can randomly select students to answer.

👫 **Pre-Mission Discussion (Slide 2, page 1):**

Students can write in their log first and then share, or discuss first and then write in their log.

There is one question for the pre-mission. There aren't any “right” answers here. The purpose is to get them thinking about the need for selecting something random. Also, there are real-world applications to what they are learning.

- In the last mission, you learned about random numbers. Other than a game, give an example of when you want to select a random person:

💻 **Mission Activities:**

Most of this lesson is on the computer, writing code to make a digital game spinner with arrows.

- Each student will complete a Mission Log.
- Students could work in pairs through the lesson, or can work individually.
- Students will need the CodeX and USB cable.

💡 **Teaching tip: Objective #1** -- Slides 3-5, Pages 2-3

This objective is review. Students use a random number to display an arrow from a predefined list.

🗝️ **NOTE:** The list of arrows DOES NOT need to be typed.

💡 **Teaching tip: Quiz** -- Slide 6, Page 4

Students take a ❓ short quiz. The 2 Quiz questions are below. You can decide if you need to go over the question with your students.

💡 **Teaching tip: Objective #2** -- Slides -13, Pages 4-6
Two concepts are introduced in this objective.

- First, students learn there is a buttons.is_pressed, different from the buttons.was_pressed. This concept was actually introduced in Mission 4 but not used until this mission.
- Second, logical operators are introduced. The embedded instructions mention and, or and not. In the workbook and slide deck, only and and or are shown. Only OR is used in the code.
- Two activities are given for students to complete. They will write in the mission log and complete code in CodeSpace.

💡 **Teaching tip: Objective #3** -- Slides 14-22, Pages 7-11
Students learn about writing their own functions. This is an important concept. The slide deck and workbook give an example from Mission 3 and Mission 4 where students could write a function. Then the instructions go into more detail with the example from Mission 3. You could have students open the program and do the example for extra function practice.

The last next couple slides (page) go through the function creation for the current mission.

Two activities are given for students to complete. They will complete code in CodeSpace by writing and calling a function. Then they will write in their mission log what they learned about functions.

💡 **Teaching tip: Objective #4** -- Slides 23-26, Pages 12-14
A loop that isn't infinite is the new concept. Students will use a condition, like they do for if statements, to control a loop. The control variable will be the index. It will also be used for accessing an item from the arrows list. The images on the slides/pages give an example of how this will work in the current program.

🗝️ **NOTE:** Do not move too quickly on this concept. You could even have students practice on extra examples if you are comfortable creating some. ***See "Examples of practice problems" below.***

💡 **Teaching tip: Quiz** -- Slide 27, Page 14

Students take a ❓ short quiz. The 3 Quiz questions are below. You can decide if you need to go over the question with your students.

💡 **Teaching tip: Objective #5** -- Slides 28-33, Pages 15-17
Two key concepts are introduced here: parameters and arguments. An example of each is given for this mission. This is a pretty important concept, so students will also write their own definitions in their mission log as part of the objective activities.

💡 **Teaching tip: Objective #6** -- Slides 34-36, Pages 18-19
This objective introduces an error in the code. Students change the parameter to something greater than 8. When the code is run, they will get an error. This will be fixed in the next objective.

🗝️ **NOTE:** The activity for the objective uses the debugger. It has been awhile since students used the debugger. The instructions are divided into two slides. You may want to do a refresher with them, or do this part together as a class.

🗝️ **NOTE:** To really see what is happening, students will need to view the local variables in the console. They can get everything checked off without opening the console, but they won't learn anything from the exercise unless they observe the changes in the console.

🗝️ **NOTE:** Remind the students as they are working to observe the value of index in the console log when the error occurs. They will write down the value in their mission log.

💡 **Teaching tip: Objective #7** -- Slides 37-40, Pages 20-21
Two concepts are given in this objective.

- Using index to wrap around a list (from mission 7)
- Using a different variable to control the loop. In this case, a variable called "loops".

Completing these two concepts will correct the error from objective #6. You may need to take time to go over the changes, so students have a clear understanding of the variables being used in the while loop, and the purpose of each variable.

💡 **Teaching tip: Objective #8** -- Slides 41-42, Page 22
Students will add one more variable to the function. This time the variable will be used to slow down the arrow spinning, which means it will be used in the sleep() command and students will name it delay, like they did in earlier missions. A detailed image of the changes is given.

💻 Mission Complete:

This mission ends with a completed, working program that will act as a digital game spinner with 8 possible arrows, spinning when either A or B is pressed. You need to decide how you will use the program for assessment. You could:

- Go to each student and check-off their code
- Have the students download their code to a text file and turn it in using your LMS
- Have students print their code (either download and then print the text file, or print a screenshot)
- Have students switch computers and run each other's code. Fill out a simple rubric and turn in to teacher
- Any other way that works for you

👫 Post-Mission Reflection:

The post-mission reflection asks students to think about real-world applications for animation, and also to reflect on their coding experience during the mission. You can change the questions if there is something else you want to emphasize with your students.

- What are some coding projects or applications that could use animation?
- What is something you did well during this mission? (Answers to this question could be coding related, or having to do with a soft quality like persistence, working together, etc.)

End by collecting the Mission Log and any formative assessment you want to include.

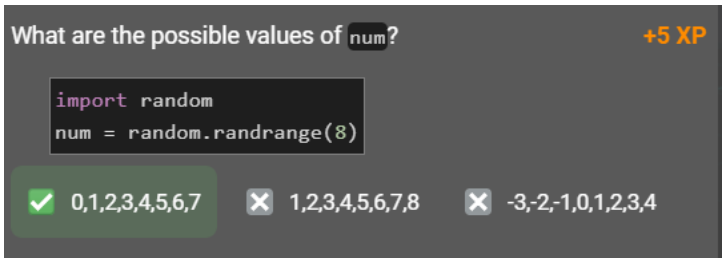
💻 IMPORTANT Clearing the CodeX:

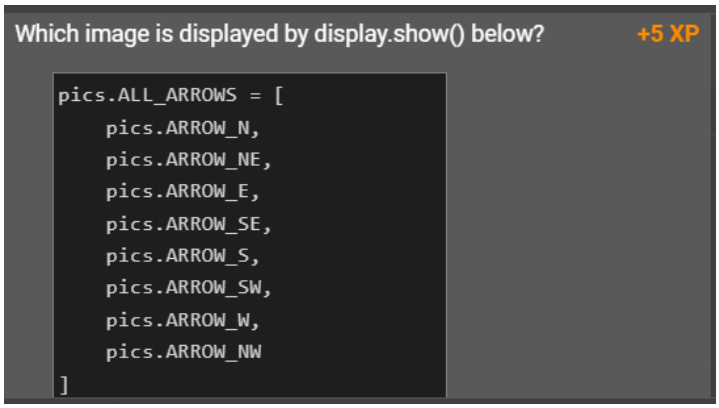
The students have already created a “Clear” program. Students should open and run “Clear” at the end of each class period.

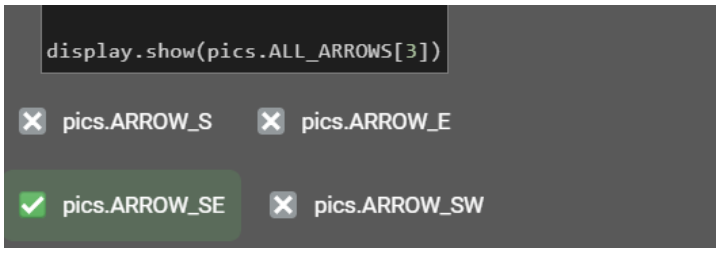
SUCCESS CRITERIA:

- ❏ Display an Arrow in a random direction
- ❏ Detect an input- button A or B - to trigger the Arrow spin
- ❏ Use a loop to animate an arrow spinning around
- ❏ Define a function for spinning arrows
- ❏ Use a parameter in the function and an argument in the function call
- ❏ Define a function that displays one random arrow
- ❏ Use three variables:
  - ❏ Variable for the index
  - ❏ Variable to control the loop
  - ❏ Variable to vary the speed of the animation

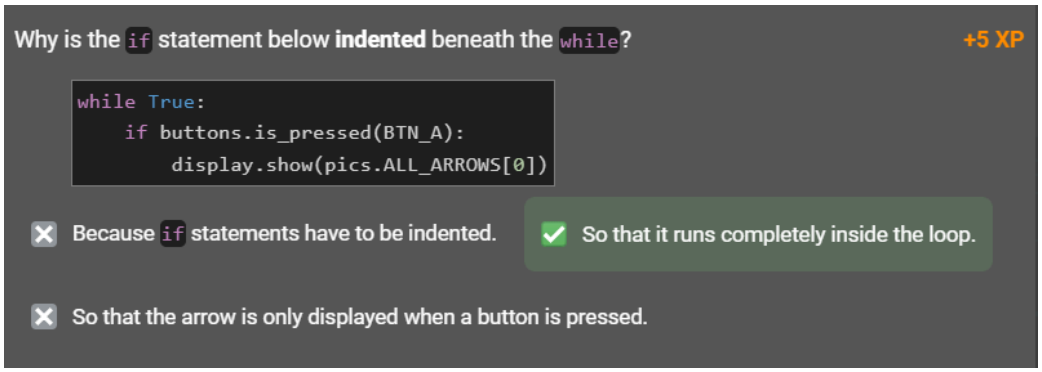
❓ Quiz #1 Questions

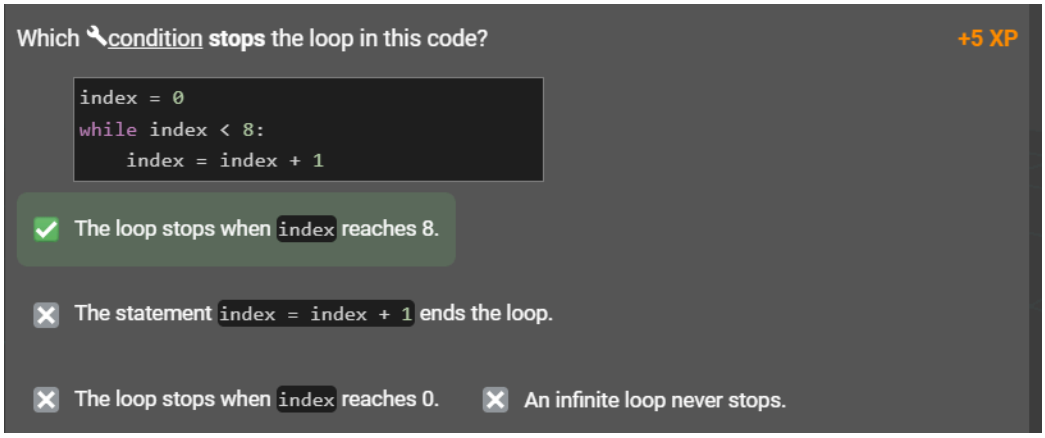


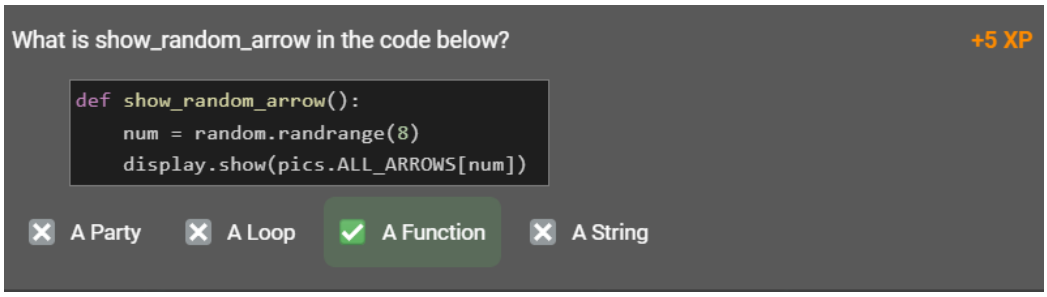




❓ Quiz #2 Questions



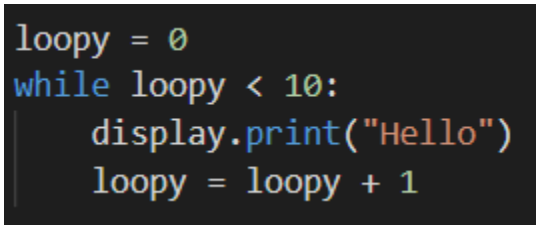
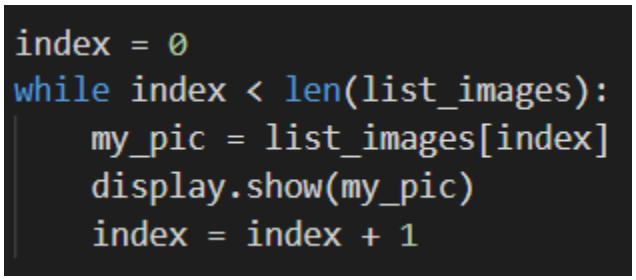
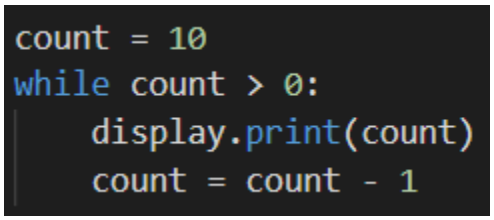
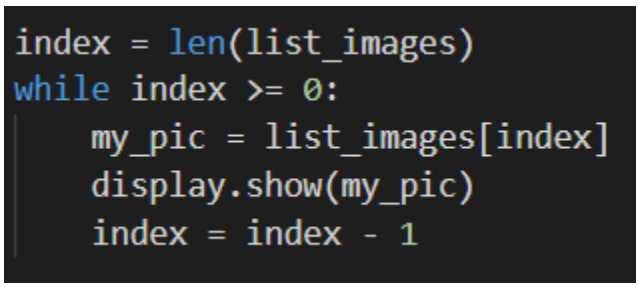




Objective 4 Examples of Practice Problems

Write a loop that will loop 5 times. Use the variable count as the control variable. Print the value of count inside the loop.	`count = 0`
	`while count < 5:`
	`  display.print(count)`
	`  count = count + 1`

FIRIA LABS

Write a loop that will loop 10 times. Use the variable loopy as the control variable. Print “hello” inside the loop.	
Your program has a list: list_images. Write a loop that will display the images one at a time until the last image. Use index as the control variable. Use the length of the list to know when to stop.	
CHALLENGE: Write a loop that starts at 10 and counts down to 1. Print the value of the control variable inside the loop.	
CHALLENGE: Write a loop that displays the last image in a list, and ends with the first image in the list. Use index as the control variable, and the length of the list as the first value.	

```
loopy = 0
while loopy < 10:
    display.print("Hello")
    loopy = loopy + 1
```

```
index = 0
while index < len(list_images):
    my_pic = list_images[index]
    display.show(my_pic)
    index = index + 1
```

```
count = 10
while count > 0:
    display.print(count)
    count = count - 1
```

```
index = len(list_images)
while index >= 0:
    my_pic = list_images[index]
    display.show(my_pic)
    index = index - 1
```