FIRIA LABS

MISSION 9 Remix

Time: 60-90 minutes

Overview:

This remix will allow students to use their creativity to create their own project. They will use concepts from mission 3 through mission 9. They will create a program that incorporates loops, button presses, if statements, variables, and output such as images, lights or sound.

Cross Curricular:

- **CROSS CURRICULAR:** The remix can be like a Jeopardy game, with categories on the wheel. Use it for review or discussion in any subject.
- **CROSS CURRICULAR:** Spin a wheel of student names. The selected student answers the question, is team captain, etc.
- **MATH:** Spin a wheel of numbers 1 through 10. The number it stops is the answer, and students come up with an expression that equals the answer.
- **MATH:** Use the dice roll or game spinner for a lesson on probability. Then see how partial or impartial the wheel is.
- **LANGUAGE ARTS:** Spin a wheel of words – types of speech, sentence starters, parts of a story, etc. Use the wheel spin for discussion, review, etc.
- **VISUAL ARTS:** Spin a wheel of colors. Use the wheel to discuss color, or use the designated color in a project, etc.
- Supports **language arts** through peer review and reflection.

Materials Included in the learning portal Elementary Remix Projects:

Mission 9 Remix Slidedeck

The slide deck is for teacher-led instructions that let you guide students through completing a remix using the slides. There are no instructions for a remix in CodeSpace. The slides give instructions, with simplified language that is chunked into small sections at a time. The information is shown on slides with "Step #". The tasks to complete are on slides with "Do This".

Mission 9 Remix Workbook

The workbook can be used instead of slides for student-led or independent work. It is an alternative to the slide deck, with simplified language that is chunked into small sections at a time. Each step is on its own page. The tasks to complete are labeled "DO THIS" and have a robot icon next to it.

Mission 9 Remix Log (print and digital)

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

Mission 9 Remix Lesson Plan

This is the original lesson plan for the remix. It includes a suggestion for the remix and rubric. For additional ideas, hints and helps, use the information below.

Mission 9 Remix Solutions Videos

Ten suggestions are given for remix projects. Solutions are given for each suggestion (mild, medium, spicy)

Additional Resources:

- **Remix Videos:**
  Mild-1A, Mild-1B, Mild-1C, Med-2A, Med-2B, Spicy-3A, Spicy-3B, Spicy-3C, Folder with all coding solutions
  Extra Spicy-4A, Extra Spicy-4B
- Mission 9 Review Kahoot

Formative Assessment Ideas:

- Exit ticket
- Remix log completion
- Completed program
- Gallery walk
- Mission 9 Review Kahoot
- Student Reflection

Vocabulary: No new vocabulary for this remix.
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 or Google. ● Be familiar with the Remix Log (assignment) and the questions they will answer. ● Print a copy of the Remix Log for each student or prepare it digitally. ● Look over the remix solutions so you can help students as they create their own remix.

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-Remix Discussion: Slide 2 (slides), Page 1 (workbook)

There are two questions. Students can write in their log first and then share, or discuss first and then write in their log. The purpose of the pre-remix questions is to get students thinking about their own programming journey.

- What is a function?
- What is the difference between a parameter and an argument?

💻 Remix Project: Slides 3-4, Page 2

These slides and pages discuss the benefits of creating a remix.

💻 Remix Steps:

The remix is organized into 5 steps. Each step has a corresponding section on their log assignment. After every mission, there will be a remix opportunity, and each remix will go through the same five steps. This follows the design process used in many career fields.

- Each student will complete a Remix Log.
- Students can work in pairs through the lesson, or can work individually.
- Students will need the CodeX and USB cable, and optionally, 4 AAA batteries.

💡 **Teaching tip: Step #1 -- Review projects and concepts** (slide 5, page 3)

Students open their program from the last mission and review what the program does and the concepts they learned and used. They fill out the information on their log. Take your time on this part. Let the students discuss or share their answers. OPTION: Review other missions as well; it could help the students with their creativity.

💡 **Teaching tip:Step #2 -- Brainstorm** (slides 6-11, pages 4-7)

Students brainstorm their remix project. Ten suggestions for a remix are given: three mild, two medium, two spicy and two extra spicy. Students can look at a video of each finished remix. They can choose any of these, or come up with their own ideas. They can combine ideas from the suggestions to make their own, as well. They will write about their idea in the log assignment.

💡 **Teaching tip: Step #3 -- Make a plan** (slide 12, page 8)

Students plan the variables they need, functions for the code, lists they will create and use, and the buttons they will write code for. Students don't always want to plan, or see the value of planning, but it really will help them code the project. Emphasize this with the students, that this is an important part of the design process.

💡 **Teaching tip: Step #4 -- Code your project** (slides 13-14, page 9)

Students start a new project in CodeSpace. There isn't a mission for the remix, so students will use the sandbox. The icon for the sandbox is in the lower right-hand corner above the toolbox. Students should write just a few lines of code at a time and test frequently. They can use their code from any mission or remix, as well as the instructions from any of the missions or this lesson. They don't have to have anything memorized.

💡 **Teaching tip: Step #5 -- Documentation and feedback** (slides 15-16, page 10)

This step has two parts: documentation and feedback. For documentation, students should make their code readable by adding blank lines and comments. Some students are naturally good at this and may have already done it. Other students may need the reminder. The second part is to get a peer to look over the code and give feedback. The student also reviews his/her project and gives feedback. Students are encouraged to read the feedback and use it to improve their project.

🧑‍🤝‍🧑 **Post-Remix Reflection:** (slides 17-18, page 11)

The project is complete and students are asked to reflect using three questions. These are thought questions, and you may want students to share their responses. This is an excellent opportunity to have a gallery walk of all the projects, or have presentations.

Solution code is given in the folder for eight remix suggestions.

💻 Mission Complete:

This mission ends with a completed, working program. This is an excellent time to have students present their project, or have students do a gallery walk around the room and play the other students' projects.

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 words for you

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

A Kahoot! that reviews Mission 9 is available (questions below).

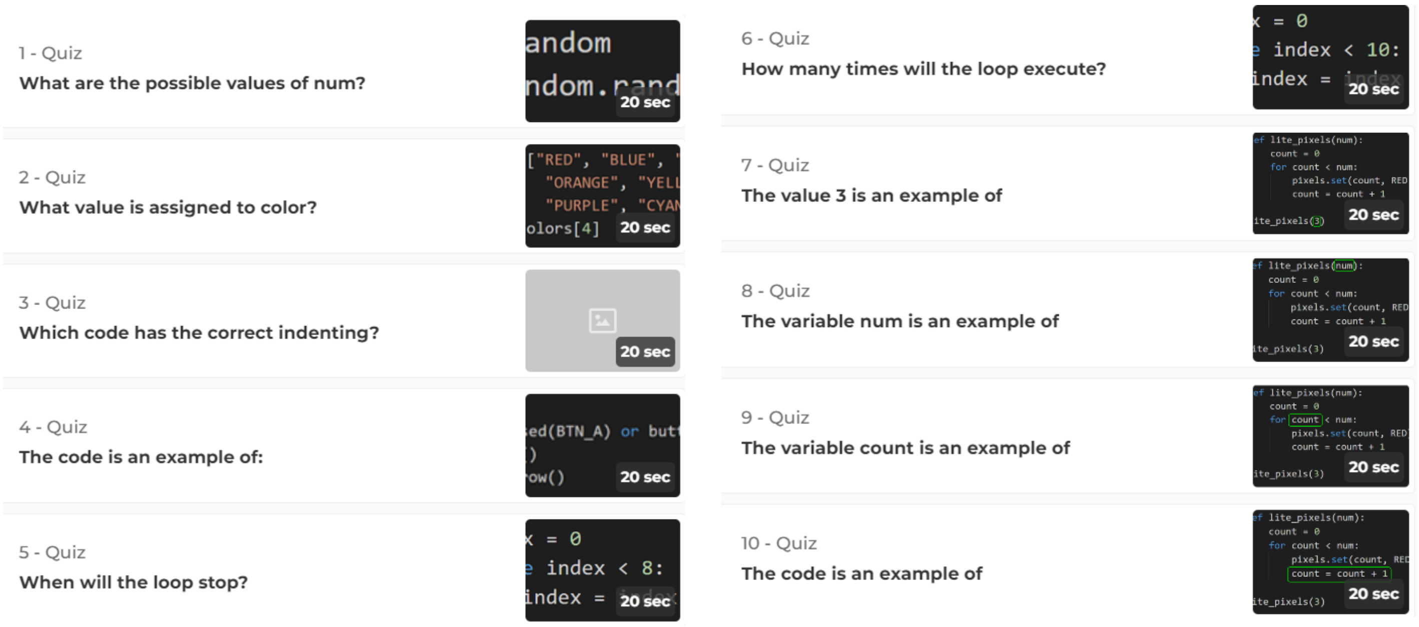
💻 IMPORTANT Clearing the CodeX:

Students should run their “Clear” program at the end of each day before returning the CodeX.

SUCCESS CRITERIA:

- ❏ Write an original program, run it, and save it to the CodeX
- ❏ Follow the design process and document their work in the log assignment
- ❏ Add readability to your program by adding blank lines and comments
- ❏ Use at least one list
- ❏ Use at least one button as input to affect the code
- ❏ Use a random function at least once
- ❏ Debug any errors in the code
- ❏ Clear the CodeX of meaningful code

❓ Kahoot! Review



1 - Quiz
What are the possible values of num?

```
andom
ndom.rand
```
20 sec

2 - Quiz
What value is assigned to color?

```
["RED", "BLUE", "
  "ORANGE", "YELL
  "PURPLE", "CYAN
olors[4]
```
20 sec

3 - Quiz
Which code has the correct indenting?

20 sec

4 - Quiz
The code is an example of:

```
sed(BTN_A) or butt
)
row()
```
20 sec

5 - Quiz
When will the loop stop?

```
x = 0
e index < 8:
index = index
```
20 sec

6 - Quiz
How many times will the loop execute?

```
x = 0
e index < 10:
index = index
```
20 sec

7 - Quiz
The value 3 is an example of

```
ef lite_pixels(num):
   count = 0
   for count < num:
       pixels.set(count, RED
       count = count + 1

ite_pixels(3)
```
20 sec

8 - Quiz
The variable num is an example of

```
ef lite_pixels(num):
   count = 0
   for count < num:
       pixels.set(count, RED
       count = count + 1

ite_pixels(3)
```
20 sec

9 - Quiz
The variable count is an example of

```
ef lite_pixels(num):
   count = 0
   for count < num:
       pixels.set(count, RED)
       count = count + 1

ite_pixels(3)
```
20 sec

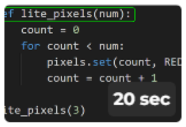
10 - Quiz
The code is an example of

```
ef lite_pixels(num):
   count = 0
   for count < num:
       pixels.set(count, RED
       count = count + 1

ite_pixels(3)
```
20 sec

11 - Quiz

The code is an example of



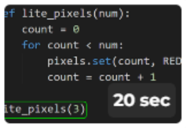
```
ef lite_pixels(num):
    count = 0
    for count < num:
        pixels.set(count, RED
        count = count + 1

ite_pixels(3)
```

20 sec

12 - Quiz

The code is an example of



```
ef lite_pixels(num):
    count = 0
    for count < num:
        pixels.set(count, RED
        count = count + 1

ite_pixels(3)
```

20 sec

13 - Quiz

What code correctly defines a function?

20 sec

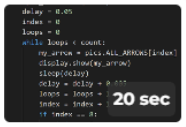
14 - Quiz

What code correctly calls a function?

20 sec

15 - Quiz

What variable is the loop control variable?



```
    delay = 0.05
    index = 0
    loops = 0
    while loops < count:
        my_arrow = pics.ALL_ARROWS[index]
        display.show(my_arrow)
        sleep(delay)
        delay = delay + 0.005
        loops = loops +
        index = index +
        if index == 0:
```

20 sec