

MISSION 13: Sounds Fun

Time: 60-120 minutes

Overview:

In this project students will create a **user-friendly graphical user interface** for CodeX so that it can act as a race day controller for the start and end of a race. Students will learn how to play sounds and music in the background while other code is running, make sound effects for games and user feedback, and control the pitch and loop sounds.



Cross Curricular:

- **MATH:** A for loop can increment the loop control variable by a value other than 1. This is like skip-counting. Have students count by different amounts, like 5s, 10s, etc. and then write a for loop that does the same thing.
- **SCIENCE:** This mission discusses sounds by using a frequency. Have a lesson about sound and pitch or frequently. Try out the concepts on the CodeX in a program.
- Supports **language arts** through reflection writing.

Materials Included in the learning portal [Teacher Resources](#):

Mission 13 Sliddeck

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 13 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 13 Log

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 13 Lesson Plan

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

[Mission 13 Remix](#)

Following Mission 13 students should complete a remix of their code.

Links:

- [Mission 13 Solution \(Sounds Fun\)](#)
 - A code solution to Mission 13 in a text file
- [Review Kahoot \(Mission 13 Obj. 1-6\)](#)
- [Review Kahoot \(Mission 13 Obj. 7-11\)](#)

Formative Assessment Ideas:

- Exit ticket(s)
- Mission log completion
- Completed program
- [Obj. 1-6 Kahoot Review](#)
- [Obj. 7-11 Kahoot Review](#)
- Student Reflection

Vocabulary:

- **User Interface:** The area where a person interacts with a physical device, often through a screen.
- **Bitmap:** Graphics bits – drawing images and text. A bitmap is an object that can hold a 2D image of a given width and height; a list of pixel RGB values.
- **Local Variable:** A variable that is “private” to a function. It only exists while the function is running, and is separate from any other variable outside the function.
- **Global Variable:** Variables defined outside of a function. They exist the entire life of the program and can be accessed and used inside a function.
- **Initialization:** Set the initial or first value of a global variable when the program starts. Also, set the screen to its beginning look.
- **Soundlib module:** Functions for creating music and sound effects, including different types of tones.
- **For Loop:** Looping across a range of numbers, or iterating over a list.
- **Blocking Function:** Functions that block your code from continuing until they are finished. The code has to wait while a song plays, for example.
- **Non-blocking Function:** A function that doesn’t make your code wait for the function to finish. For example, other lines of code can execute while a song is playing.
- **Toggle:** Flip the state of a variable (True to False or False to True) that is used to either do or not do something.
- **Nested For Loop:** A for loop with a for loop inside, or nested.

Preparing for the lesson:

This mission will create a race day controller. Non-blocking functions will be used. You can compare the non-blocking sound functions to the blocking sound function from Mission 5.

Also, students will be drawing rectangles. You may want to first review the display screen and how to plot points on the screen. You can go over the parameters needed for the rectangle.

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, or upload digitally to your LMS.
- 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 2-4, page 1-2):

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 isn't a "right" answer 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 this mission you will use CodeX as a controller to start and end a race. With batteries CodeX can go anywhere! What is something you might want the CodeX to do or control?

Mission Activities:

Most of this lesson is on the computer, writing code to create the race day controller.

- 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 5-13, Pages 3-7

This objective is fairly involved. Students create a user interface for the race controller. They start with a function that will be used frequently to draw all the rectangles and text that will be used by a person. It uses a location, or position, on the screen to draw all the elements. You may need to have a short lesson or review the coordinates on the screen.

Students will answer one question in their mission log.


Students will create a new file, and then follow CodeTrek to code the user interface.


Teaching tip: Objective #2 -- Slides 14-22, Pages 8-12

The code will use a list for the y-coordinates. It will be used to move the rectangle. The x-coordinate never changes; it is always 0, so y is the only one that needs to be included in a list. The code will also use a variable for the list index.

CodeTrek will also use the max() and min() functions for the first time. You may want to review what these functions do, and give the students some practice.


Students will answer two questions in their mission log.

 **NOTE:** CodeTrek introduces a new variable to keep track of the previous menu selection. CodeTrek doesn't specifically stop on this line of code (line 33) but there is a comment about it. You might want to point it out. It is also a question on the mission log.


 **NOTE:** The code will cause a new error – with a global variable. It is discussed and fixed in the next objective. But students MUST throw the error to meet the goal of the objective.


Teaching tip: Objective #3 -- Slides 23-27, Pages 13-15


Fix another bug! This time it is that the rectangle moves, but the old one stays. So, erase your tracks!

 **NOTE:** Students need to be careful where they erase the rectangle. It needs to be BEFORE they draw the new rectangle.



 **Teaching tip: Quiz** -- Slide 28, Page 15

Students take a  short quiz. The 2 Quiz questions are below. Both are about local and global variables.

 **Teaching tip: Objective #4** -- Slides 29-30, Pages 16

This objective uses the previous selection variable from Objective 2 to “erase” the rectangle before drawing the new rectangle at “menu index”. Review as needed.

 **Teaching tip: Objective #5** -- Slides 31-36, Page 17-20

Students will create two more functions. One function takes a parameter for a message and displays it at the top. The other function checks which button is pressed and calls the first function, passing in a message for the parameter.

Students will answer one question in their mission log.

 **Teaching tip: Objective #6** -- Slides 37-41, Page 21-22


Students make the UI more user-friendly by initializing the screen. Uses a global Boolean variable “init” and modifies the if statement in the while True loop.

Students will answer one question in their mission log.

 **Teaching tip: Objective #7** -- Slides 42-47, Page 23-25

A lot of information here about the soundlib module. It has functions for creating music and sound effects. First a tone will be selected, and then set a pitch and play. Use stop to quit playing the tone at the given pitch. Review as needed.


Students will answer two questions in their mission log.

 **Teaching tip: Objective #8** -- Slides 48-51, Page 26-27

This objective introduces the for loop as an alternative to the while loop. You might want to practice with your students. Have them convert some while loops to for loops, and vice versa to help them understand the concept.

Students will answer two questions in their mission log.

 **Teaching tip: Quiz** -- Slide 52, Page 27

Students take a  short quiz. The 2 Quiz questions are below. Both are about the results of a for loop.


 **Teaching tip: Objective #9** -- Slides 53-59, Pages 28-30

This objective discusses blocking and non-blocking functions. You can go back to Mission 5 and look at the code for playing a sound. It is blocking! Here students will learn a different code for non-blocking. In the hint, it shows an argument to add to the “get_mp3” statement so the music doesn’t start automatically.




This objective also uses a “toggle” Boolean variable. If this is a difficult concept for your students, you will want to review and practice it. CodeTrek does not specifically stop on this line of code (#45) so you may need to specifically point it out.


Students will answer one question in their mission log.

 **Teaching tip: Quiz** -- Slide 60, Page 31

Students take a **?** short quiz. The 3 Quiz questions are below. Two questions are about blocking / non-blocking. The other question is about a toggle variable using “not”.

 **Teaching tip: Objective #10** -- Slides 61-63, Pages 31-32

Students create another sound using a loop within a loop. You might want to step through this part of the code using the debugger to track the variables, or work it out on a white board.

 **Teaching tip: Objective #11** -- Slides 64-67, Pages 33-34

Students create another sound using the glide function.

Mission Complete:

This mission ends with a completed, working program that will act as a race day controller. 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 making a GUI on the CodeX for other programs. Also, review the concept of abstraction by discussing how a GUI is an abstraction. You can change the questions if there is something else you want to emphasize with your students.

- You can now make a GUI on CodeX! This is a very useful skill that you can apply to many other programs. What are some other programs you have already created, or might create, that could use a GUI?
- A GUI is an example of abstraction. Explain how a GUI is an abstraction.
- What are three things you did to work through struggles and complete the program?

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 a GUI with four menu items
- Use a rectangle to show which menu item is selected
- Scroll the selection rectangle up and down to the different menu items
- Select a menu item using BTN A
- Play music in the background
- Play music for starting the race
- Play music for finishing the race
- Play music as a warning sound

? Quiz Questions

Quiz 1

 Globals and Locals

Which of the following is true? +5 XP

- Any variable used inside of a function is considered **local** to that function, unless it is explicitly named in a `global` statement.
- A variable which is **assigned** to inside of a **function** is considered **local**, unless it is explicitly named in a `global` statement..
- A **global** variable is not visible inside a function, unless it is explicitly named in a `global` statement.

The following function reports an error on Line 2. +5 XP

```
1 def add_counter(amount, limit):
2     if counter < limit:
3         counter = counter + amount
```

What is the expected error message?

- Undefined variable: counter
- Global variable cannot be used in comparison.
- Local variable referenced before assignment

Quiz 2

? For Loops

What is displayed by the following code?

```
for i in range(4):  
    display.print(i, end=',')
```

1,2,3,4 0,1,2,3,4 0,1,2,3

What is displayed by the following code?

```
for i in range(1, 5):  
    display.print(i, end=',')
```

1,2,3,4,5 1,2,3,4 0,1,2,3,4,5

Quiz 3

? Blocking and Toggling

What is a **blocking** function? +5 XP

A function composed of blocks. A function that creates square 2D shapes or 3D cubes.

A function that returns immediately, even if that would block it from completing its goal.

A function that blocks program execution and does not return until it has completed its goal.

`sleep(5)` will delay the program for 5 seconds. +5 XP

Is it a "blocking function"?

Yes No

What is the final value of `toggle` after this code runs? +5 XP

```
toggle = False  
  
toggle = not toggle  
toggle = not toggle  
toggle = not toggle
```

3 True False