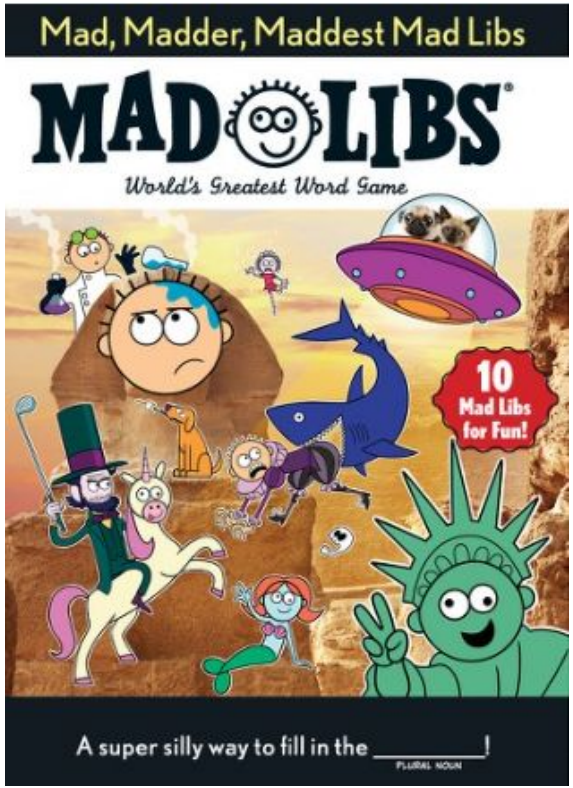
Madlibs

Language Arts

Madlibs

Mad Libs is a template word game created by Leonard Stern and Roger Price.

- It consists of one player prompting others for a list of words to substitute for blanks in a story before reading aloud.
- The game is frequently played as a party game or as a pastime.



Madlibs

The concept can also be used with short poems or nursery rhymes.

Madlibs with CodeX:

- Create a list for nouns, verbs, colors, etc.
- Use the random library to select a random item from the list
- Use the randomly selected words in the poem or nursery rhyme

FIRIA LABS

Step #1

Copy and paste the starter code into the CodeSpace text editor.

- Define variables for the verb and animal by getting a random item from their lists
- OPTIONAL: Add more items to each list

```
'''
Madlibs Program - starter code
'''
from codex import *
from time import sleep
import random

nouns = ["hill", "tuffet", "clock"]
verbs = ["broke", "ran", "sat"]
animals = ["spider", "mouse", "sloth"]
names = ["Jack", "Jill", "Miss"]

def bright_pixels():
    red = random.randrange(0, 255)
    green = random.randrange(0, 255)
    blue = random.randrange(0, 255)
    color = (red, green, blue)
    pixels.set(0, color)
    pixels.set(1, color)
    pixels.set(2, color)
    pixels.set(3, color)

def random_words():
    global name1, name2, noun, verb, animal
    name1 = random.choice(names)
    name2 = random.choice(names)
    noun = random.choice(nouns)
    #TODO: define variable for verb
    #TODO: define variable for animal
```

Step #2

Go to the main program and complete the **# TODOs**

- Clear the screen
  - This is a built-in function
- Call the bright pixels function
- Call the random words function

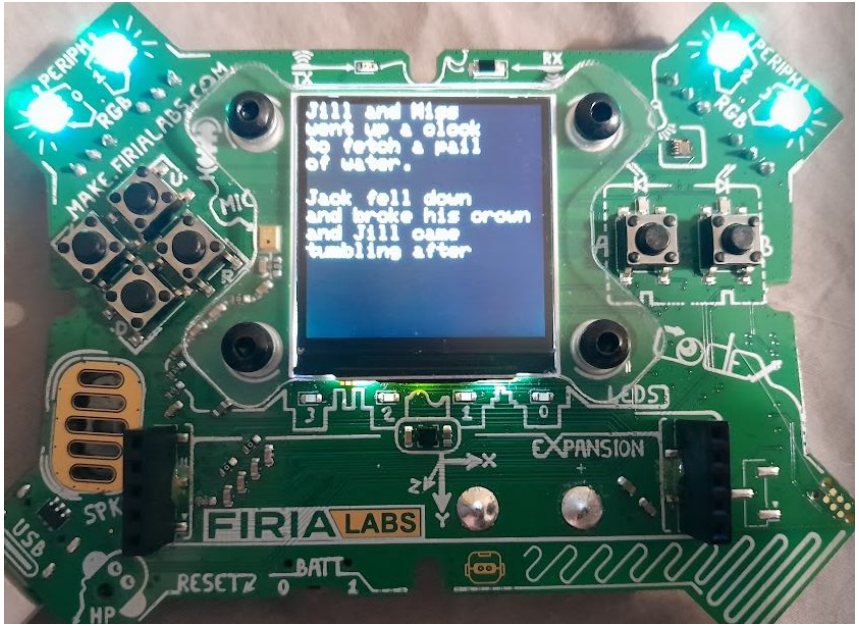
```
# === Main program ===
menu()
while True:
    if buttons.was_pressed(BTN_L):
        # TODO: Clear the screen
        # TODO: Call the bright pixels function
        # TODO: Call the random words function
        jack_and_jill()
```

FIRIA LABS

Step #3

Run the program

- You shouldn’t have any errors
- Press “L” multiple times to see the nursery rhyme with random words
- Press “B” to end



Step #4

Did you notice the program didn't end when you pressed "B"?

- Write the code in the main program that will break out of the loop if button B was pressed
- Then run the program again and test your "B" button

```
# === Main program ===
menu()
while True:
    if buttons.was_pressed(BTN_L):
        display.clear()
        bright_pixels()
        random_words()
        jack_and_jill()

    if buttons.was_pressed(BTN_B):
        |
```

FIRIA LABS

Step #5

Complete the madlib for the first nursery rhyme.

```
def jack_and_jill():
    display.print(name1 + " and " +  name2)
    display.print("went up a " + noun)
    display.print("to fetch a pail")
    display.print("of water.")
    display.print("")
    display.print("Jack fell down")
    display.print("and broke his crown")
    display.print("and Jill came")
    display.print("tumbling after")
```

- Look at the function for “Jack and Jill”
- Right now it uses three variables
- Study the code for the three variables and notice the syntax

FIRIA LABS

Step #5

Add three (or more) variables to complete the madlib for the nursery rhyme.

- The image shows how to add name1 and name2 in the second half
- Add a verb variable as well
- Run the program again

```
def jack_and_jill():
    display.print(name1 + " and " +  name2)
    display.print("went up a " + noun)
    display.print("to fetch a pail")
    display.print("of water.")
    display.print("")
    display.print(name1 + " fell down")
    display.print("and broke his crown")
    display.print("and " + name2 + " came")
    display.print("tumbling after")
```

FIRIA LABS

Step #6

Add variables to one or two of the other nursery rhymes

- You do not need to use all the variables in every nursery rhyme – just use the ones that make sense

```
def hickory_dickory():
    display.print("Hickory dickory ")
    display.print("dock")
    display.print("The " + animal + " went up")
    display.print("the " + noun)
    display.print("The clock struck 1")
    display.print("The mouse ran down")
    display.print("Hickory dickory ")
    display.print("dock")
```

```
def little_miss_muffet():
    display.print("Little " + name1 + " Muffet")
    display.print("sat on a " + noun)
    display.print("eating her curds")
    display.print("and whey")
    display.print("Along came a spider")
    display.print("and sat down ")
    display.print("beside her")
    display.print("And frightened Miss")
    display.print("Muffet away")
```

Step #7

For every nursery rhyme you make into a madlib, add a button press to call the function

- Follow the same steps as the button press for “Jack and Jill”
- Run your code and make sure it is error-free

```
menu()
while True:
    if buttons.was_pressed(BTN_L):
        display.clear()
        bright_pixels()
        random_words()
        jack_and_jill()

    if buttons.was_pressed(BTN_R):
        display.clear()
        bright_pixels()
        random_words()
        hickory_dickory()

    if buttons.was_pressed(BTN_B):
        break
```

FIRIA LABS

Step #8 - Optional

Right now all the pixels are set to the same random color.

- Add variation to the display by creating a random color for each pixel.
- You can do this without duplicating code by using a loop!

```
def bright_pixels():
    red = random.randrange(0, 255)
    green = random.randrange(0, 255)
    blue = random.randrange(0, 255)
    color = (red, green, blue)
    pixels.set(0, color)
    pixels.set(1, color)
    pixels.set(2, color)
    pixels.set(3, color)
```

FIRIA LABS

Step #8 - Optional

You used a for loop in Mission 9 with the game spinner.

- A for loop lets you repeat steps multiple times
- In this case, you want to get a random number for each pixel

```
def bright_pixels():
    red = random.randrange(0, 255)
    green = random.randrange(0, 255)
    blue = random.randrange(0, 255)
    color = (red, green, blue)
    pixels.set(0, color)
    pixels.set(1, color)
    pixels.set(2, color)
    pixels.set(3, color)
```

FIRIA LABS

Step #8 - Optional

Use a for loop to get a random color for each individual pixel

- Use the control variable to indicate each pixel
- Be careful with the indenting
- Run the code and make sure it is error-free

```
def bright_pixels():
    for pixel in range(4):
        red = random.randrange(0, 255)
        green = random.randrange(0, 255)
        blue = random.randrange(0, 255)
        color = (red, green, blue)
        pixels.set(pixel, color)
```

FIRIA LABS

Extensions

Language Arts

Language Arts Extensions to Madlibs

- Add more items to each list for more variety
- Add more lists and variables
  - Remember to make any additional variables global in the random_words() function
- Add another nursery rhymes
- Write your own short poem and add it to the program as a madlib
- Discuss rhyming words with your students

FIRIA LABS

Math Extensions to Madlibs

Use the Madlibs program to discuss probability. Here are some sample problems:

- What is the probability of getting the original nursery rhyme?
- How many madlib variations are possible?
- What is the probability of getting the same madlib twice?

Have your students come up with other problems and then do the math.

FIRIA LABS

Coding Extensions to Madlibs

- Have multiple lists to choose from.
  - Button A chooses from one set, and Button B chooses from the other
- Add a button that randomly selects a poem or nursery rhyme.
- Add an image and sound to the ending function.
- Add a sound file that plays for each poem / nursery rhyme.