BYTES AND BEATS

An Introduction to Programming with MATLAB

Instructor Guide

Module 8: Using Functions to Create Music

Prerequisite Domain Knowledge: Functions

Expected Completion Time: 50 minutes

Table of Contents

Happy Birthday Song

Expected Duration: 25 minutes

Learning Objectives

- Call functions with appropriate input arguments
- Work with **Live Scripts** – edit and modify

Motivation

Students will use MATLAB to play the Happy Birthday song!

Materials

- MATLAB®
- Worksheet "Playing Happy Birthday Song"
- Handout "Music Note Frequencies"

Solution

```
open HappyBirthdaySong_solution.mlx
```

Steps

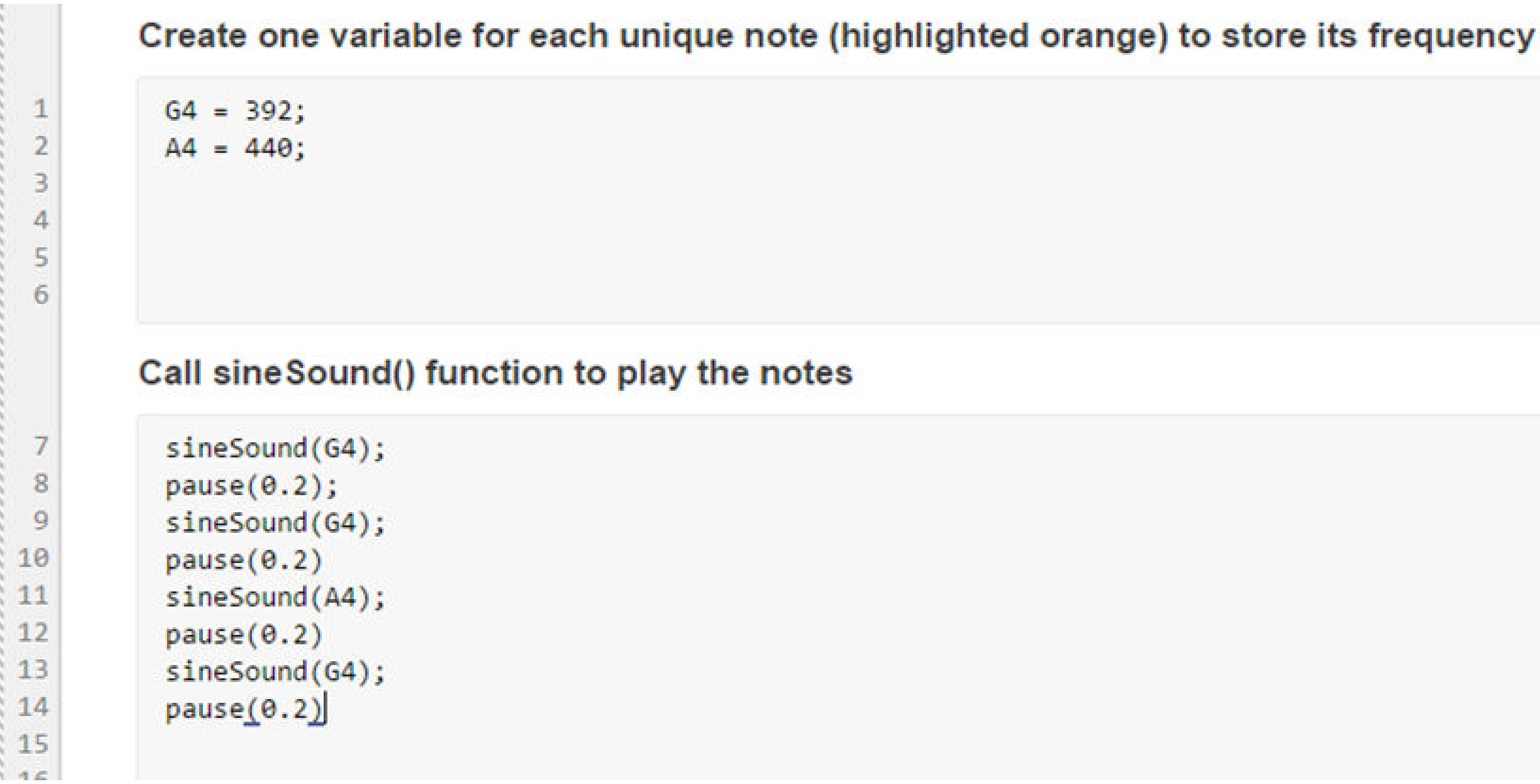
- Print and distribute the worksheet "Playing Happy Birthday Song" and the handout "Music Note Frequencies"
- Tell the students to open the Live Script "`HappyBirthdaySong.mlx`" by executing the following command in the **Command Window:**

```
open HappyBirthdaySong.mlx
```

- This will open the Live Script for them to complete.
- The script shows the notes for all four lines of the song. Some lines of code have already been filled for starting. Go over each line and discuss what they think the code does:
- The first two lines create variables to save frequencies for two notes: G4 and A4. These frequencies can be looked up in the handout.
- The next few lines are calling a function `'sineSound'`. The input to this function is a frequency for the note to be played. The four calls to `'sineSound'` are for the first four notes of the song.
- Tell the students to run this script. What they hear should sound wrong. Ask them why they think it is so.
- Because all the notes are being played back-to-back without letting each complete. MATLAB is attempting to play the notes over each other. This cannot be done online currently.
- Students can correct this by using the 'pause' function. The input to this function is time in seconds. Ask the students to enter the following line of code after each `'sineSound`' call:

```
pause(0.2)
```

- This is what the code should like at this point:



Create one variable for each unique note (highlighted orange) to store its frequency

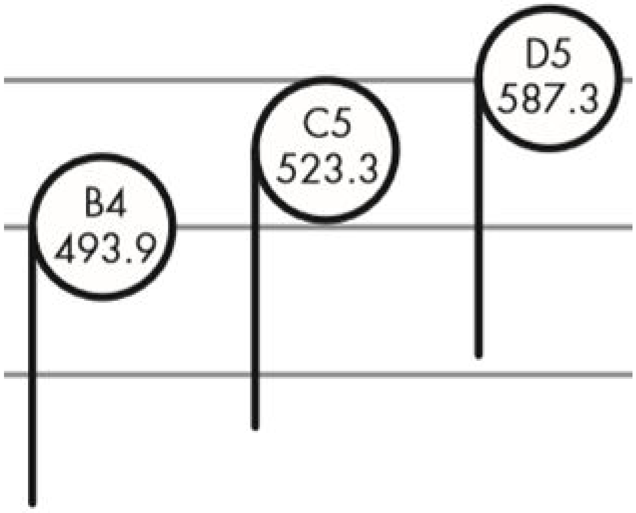
```
G4 = 392;
A4 = 440;
```

Call sineSound() function to play the notes

```
sineSound(G4);
pause(0.2);
sineSound(G4);
pause(0.2)
sineSound(A4);
pause(0.2)
sineSound(G4);
pause(0.2)
```

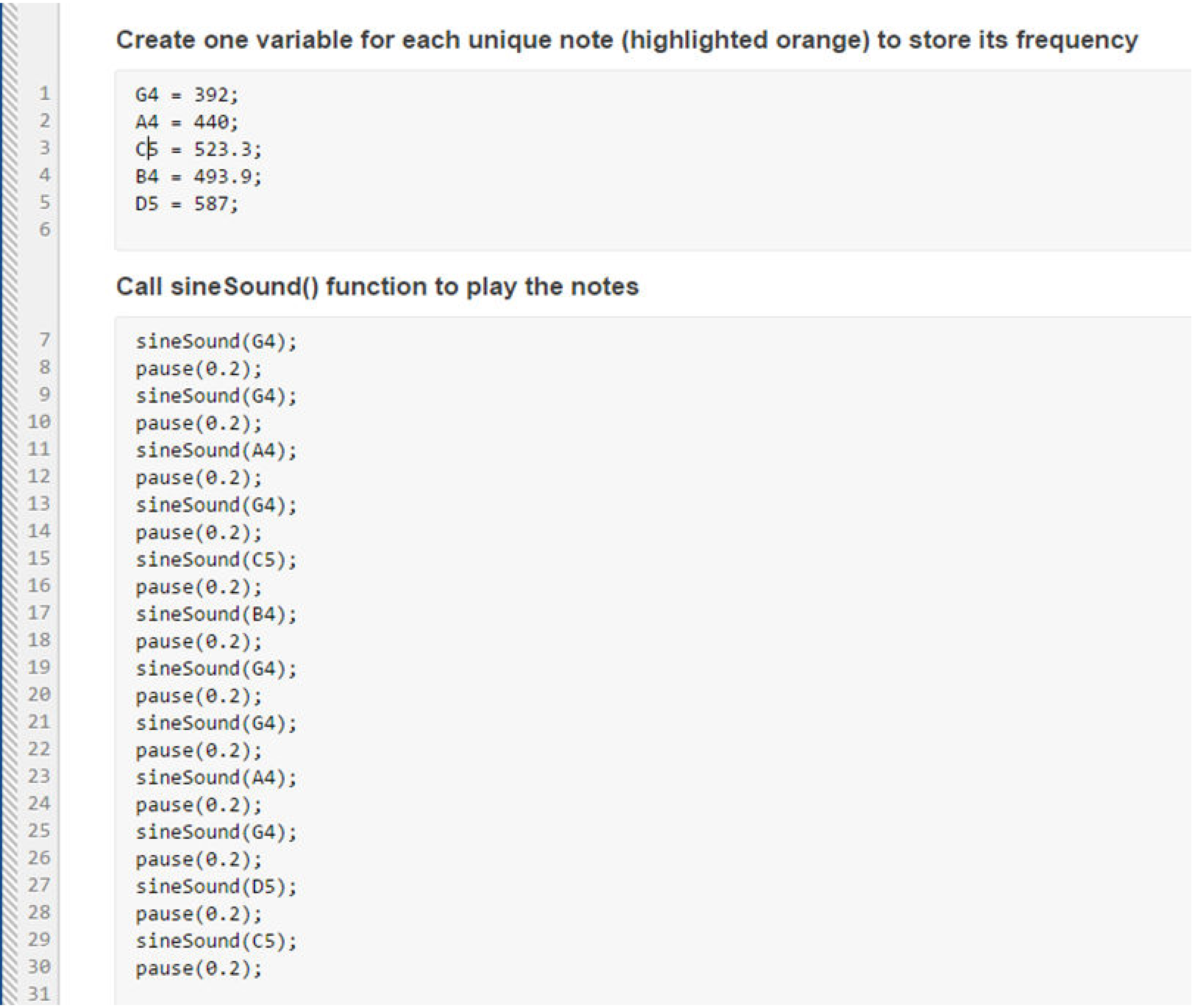
- Run the script again. These are the first four notes of the song.
- Now, the students must complete the first line and second line of the song.
- Ask them what other variables they will need for this.
- Each unique note in the song will need a variable. These are highlighted in orange. The first two lines use the notes: G4, A4, C5, B4, and D5.

They can look up frequencies in the handout and first fill them in the Worksheet. If students have trouble with this step, you can provide these to them.



Then they can create the variables for the music notes.

- Finally, they will add lines to call the function `'sineSound'` with appropriate input variables and `'pause'`with input of 0.2. This is what the script will look like at this point:



Create one variable for each unique note (highlighted orange) to store its frequency

```
G4 = 392;
A4 = 440;
C5 = 523.3;
B4 = 493.9;
D5 = 587;

```

Call sineSound() function to play the notes

```
sineSound(G4);
pause(0.2);
sineSound(G4);
pause(0.2);
sineSound(A4);
pause(0.2);
sineSound(G4);
pause(0.2);
sineSound(C5);
pause(0.2);
sineSound(B4);
pause(0.2);
sineSound(G4);
pause(0.2);
sineSound(G4);
pause(0.2);
sineSound(A4);
pause(0.2);
sineSound(G4);
pause(0.2);
sineSound(D5);
pause(0.2);
sineSound(C5);
pause(0.2);
```

- Run the script and if some students complete it faster, they can attempt to play the entire song! They can try adding longer pause in between two lines.

This ended up being an awfully long script. Tell the students that there is a way to make this code much shorter and play the entire song with few lines of code. We are repeatedly using the same two functions with different inputs. The faster way to do it is called a 'loop'. We will see that later.

Composing a Song

Expected Duration: 25 minutes

Learning Objectives

- Compose tunes using what was learned.
- Play chords and vary time duration of notes.

Motivation

Let students have fun experimenting.

Materials

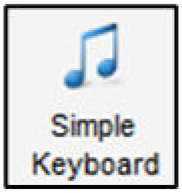
- MATLAB
- Worksheet "Playing Happy Birthday Song"
- Handout "Music Note Frequencies"
- Simple Keyboard App

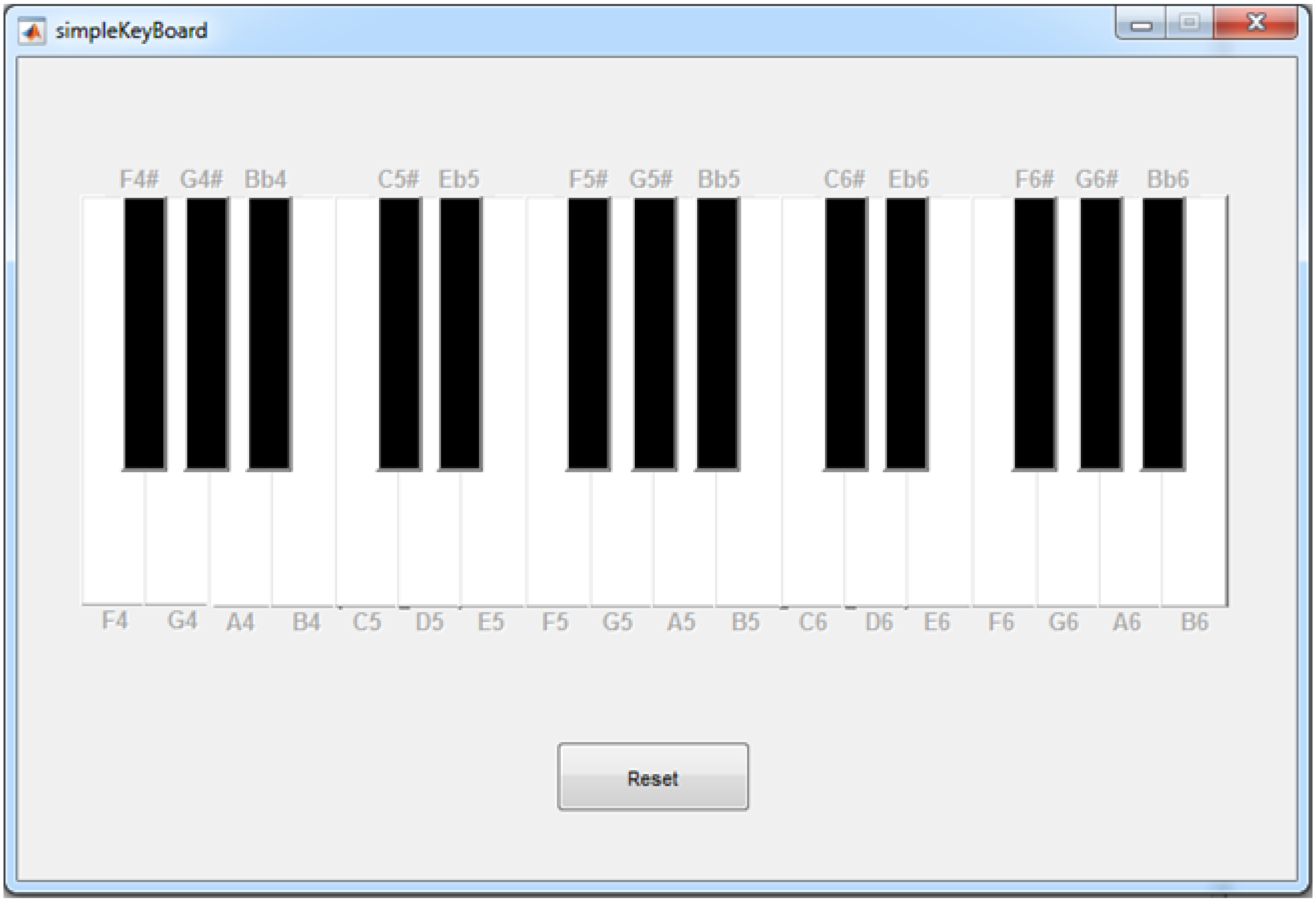
Solution

```
open HappyBirthdaySong_solution.mlx
```

Steps

- Students should form groups of three to do this activity.
- The worksheet has a section for them to compose their own song the same way the notes of the Happy Birthday song were given to them.
- To aid the students in this activity, a simple keyboard app has been provided. In the **APPS** tab, click on the music app Simple Keyboard




simpleKeyBoard
F4# G4# Bb4 C5# Eb5 F5# G5# Bb5 C6# Eb6 F6# G6# Bb6
F4 G4 A4 B4 C5 D5 E5 F5 G5 A5 B5 C6 D6 E6 F6 G6 A6 B6
Reset

This will open a keyboard which has the notes written next to the corresponding keys. They can try creating different tunes using this keyboard. In the group:

- One student will play the keyboard to come up with a tune.
- The second student will write down the notes of the keys being played.
- The third student will find the corresponding frequencies from the handout

Students can take turns switching the activities within their groups.

- Once students have the sequence of music notes they want to play and their corresponding frequencies, they can write a script for it using the `'sineSound'` function in the same way they did for the Happy Birthday song.
- They can open the Live Script for this using the below command in the **Command Window:**

```
open ComposingSong.mlx
```

- For convenience, the MAT file which contains frequencies for all the notes is included at the beginning of the Live Script so that they do not have to create variables.

```
load allNotes.mat
```

Encourage the students to make more complicated compositions by:

Changing the tone duration - The `'sineSound'` function can take an additional input to specify the time duration for which the tone is played on the speaker.

```
        sineSound(G4)
```

will play a frequency of 392 for 0.2 seconds (default)

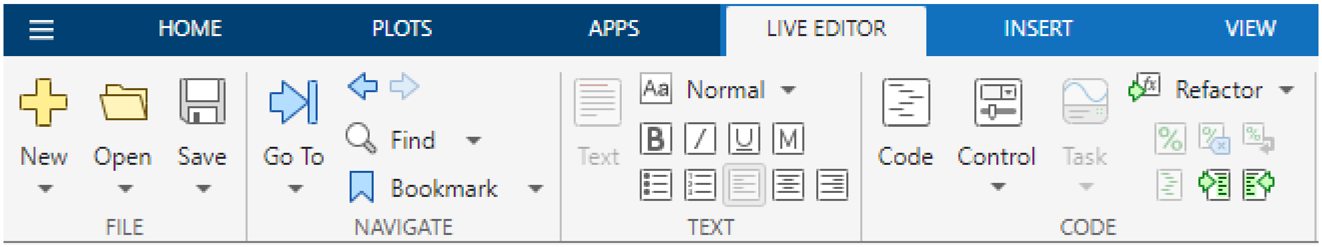
```
        sineSound(G4,0.5)
```

will play the frequency for 0.5 seconds.

You can see all the ways of using sineSound by typing:

```
        help sineSound
```

- Additionally, students can also find the music sheets for their favorite songs, get the frequencies of the music notes and use `'sineSound'` to play the songs.
- Students can save the script by clicking on the "Save" button in the Live Editor tab:


HOME
PLOTS
APPS
LIVE EDITOR
INSERT
VIEW
New
Open
Save
FILE
Go To
Find
Bookmark
NAVIGATE
Text
Normal
B
I
U
M
TEXT
Code
Control
Task
Refactor
CODE