

Go Guitar User Guide

Main Menu

`boolean started = false;`: This is a boolean variable named `started`, which is initialized with the value `false`. It is used to track whether the game has started or not.

`textAlign(CENTER);`: This sets the text alignment to the center, which means that text displayed using the `text()` function will be centered horizontally.

`if (started) { ... } else { ... }`: This is a conditional statement. If the `started` variable is `true`, the code inside the first block will be executed (for the game code). Otherwise, the code inside the second block will be executed (for displaying the guitar image and the "press anything to continue" text).

Inside the `else` block, the code is responsible for drawing a guitar using various shapes and colors. Here's a breakdown of the guitar elements:

- Guitar Body and Neck: An ellipse and a rectangle are drawn to represent the guitar's body and neck.
- Bridge: A black rectangle is drawn to represent the guitar's bridge.
- Sound Hole: A black ellipse is drawn to represent the sound hole of the guitar.
- Headstock: A black rectangle is drawn to represent the guitar's headstock.
- Strings: Several thin rectangles are drawn to represent the guitar strings.

`void keyPressed()`: This is an event handler function in Processing that gets called whenever a key is pressed.

`if (keyPressed) { started = true; }`: Inside the `keyPressed()` function, there's a conditional statement. If any key is pressed, the `started` variable is set to `true`. This means that when the user presses any key, the game will start, and the **guitar fretboard screen** will be displayed.

Sound Setup

First, the Sound Library tool is imported into the sketch with

`"import processing.sound.*;"`

After the sound library is imported you have to set **soundFiles**. For example, **"SoundFile E;"** is setting a soundFile to be recognized as E. Then in setup you attach your sound to the soundFile.

`E = new SoundFile(this, "filename.type");`

Then later on in the code you use **`E.play();`** to play the sound you set.

Fretboard Screen

When the Fretboard screen is entered you will see the functional strings where you can play sounds. To play a sound you click a highlighted string and it will make its respective sound. (Sounds are a little delayed unfortunately because of where i got the files).

How The Fretboard Works

The fretboard's code has been divided up into separate functions. Those functions are, **"neck", "frets", "strings", "tracking", and "activateSound."** We will take a deep dive into what code each function has.

neck(); – The "neck" function job is to draw the neck of the guitar to do that, it has draws a brown rectangle.

frets(); – The "frets" function draws the black lines you see going under the strings. To do this the function has a while loop where "As long as the variable 'fretX' is less than the width of the screen", a line will be drawn every 50 pixels making the frets.

The loop is written here →

```
while(fretX < width){  
  line(fretX,1*height/8,fretX,7*height/8);  
  fretX = fretX + 50;  
}
```

strings(); – "strings" just draws lines every 8th of the screen height starting from "one 8th" going to "seven 8ths".

tracking(); – The “tracking” function keeps track of your curser and highlights the string you will activate if you click. The mouse is tracked with “**if statements**” that include **mouseX & mouseY**. To highlight the closest string **mouseY** is set greater than the value of the previous string and less than the value of the next string start. When your mouse falls within those values it adds a different color line directly on top of the already existing one; giving the appearance of a highlighted string.

activateSound(); – This function is responsible for playing the sounds of the strings when you click. It has similar coding to what was in the tracking function but with added **mousePressed** and the “**.play**” & “**!.isPlaying()**” This allows for the strings dedicated sound to be played when clicked. “**!.isPlaying()**” is used to make sure the sound plays only once to prevent loud and distorted sounds.