

# Coding for Good - App Development (Juniors / Cadettes)

## **Welcome to the Coding for Good App Development Event!**

Before we get started, let's be sure we all know the following:

- The username and password of a GMAIL account
- How to connect your device (phone/tablet) to the event's WiFi network  
SSID: GSCOC-Guest Password: gs0cGuests (0 = zero)
- How to download an app from your device's app store, specifically the (free) app **Thunkable Live**

Please raise your hand if you are having problems with any of the above!

## **So what's the plan?**

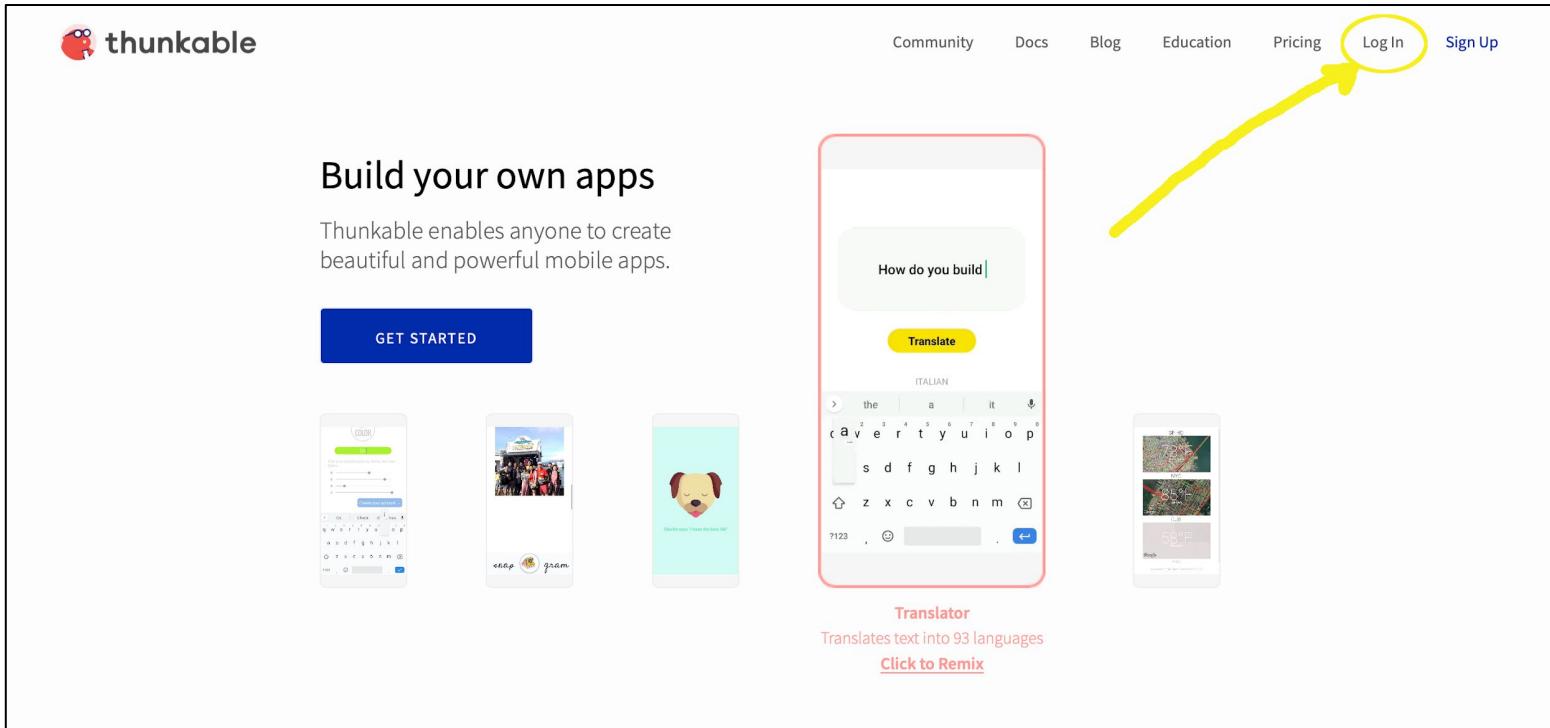
We're going to introduce the Thunkable X app development platform, then discuss specific requirements for the app we'll develop today. You can design your app as you'd like as long as it fits the event requirements, or you can just follow along with the example app that we will present. If you follow along, don't be afraid to "experiment" and pick your own button and logo images; add some of your own bedazzle to personalize your app!

Along the way, we'll talk about the process of app development, the tools app developers use, and the challenges they face.

I hope you will enjoy app development, and I encourage you to dream up your own app ideas and start building them! We'll spend a little time at the end of our event brainstorming ideas for apps that could help us solve problems in your neighbor or society.

# Getting Started - ThunkableX

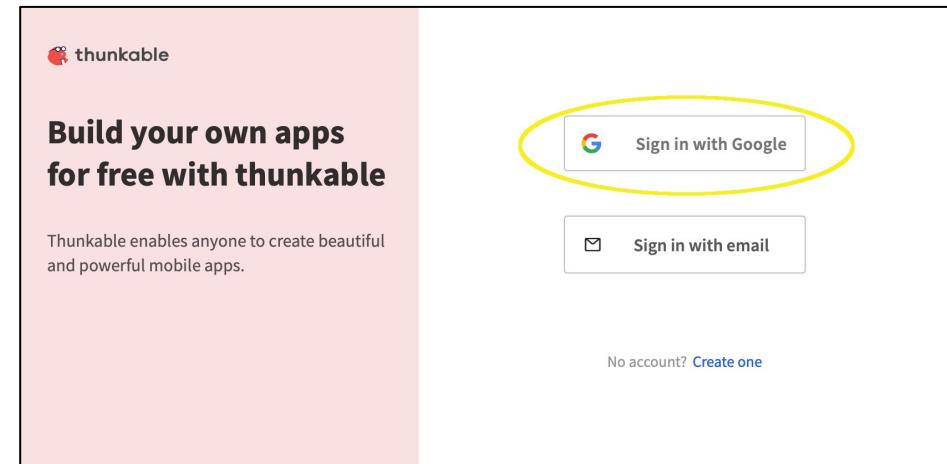
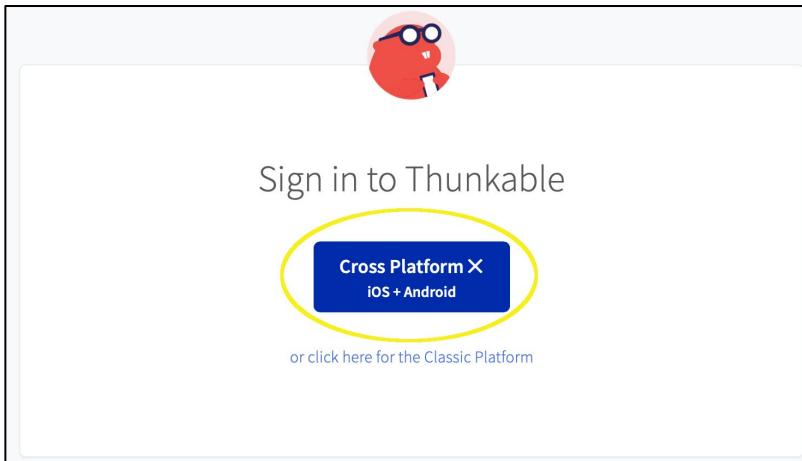
Using the **FIREFOX** browser on your laptop, navigate to the <https://www.thunkable.com> website and click the Login button in the upper right corner of the screen as shown below.



The screenshot shows the Thunkable website homepage. At the top, there is a navigation bar with links for Community, Docs, Blog, Education, Pricing, Log In (which is circled in yellow), and Sign Up. Below the navigation bar, the text "Build your own apps" is displayed, followed by the subtext "Thunkable enables anyone to create beautiful and powerful mobile apps." A large blue "GET STARTED" button is visible. On the right side of the page, there is a section titled "Translator" which includes a text input field containing "How do you build?", a "Translate" button, and a keyboard. A yellow arrow points from the bottom left towards the "Log In" button. Below the translator section, there is a link "Click to Remix".

# Getting Started - ThunkableX

Click on the Cross Platform X button, and then click on the Sign in with Google button. **Use the GMAIL username and password you were asked to bring to this event to log in.**

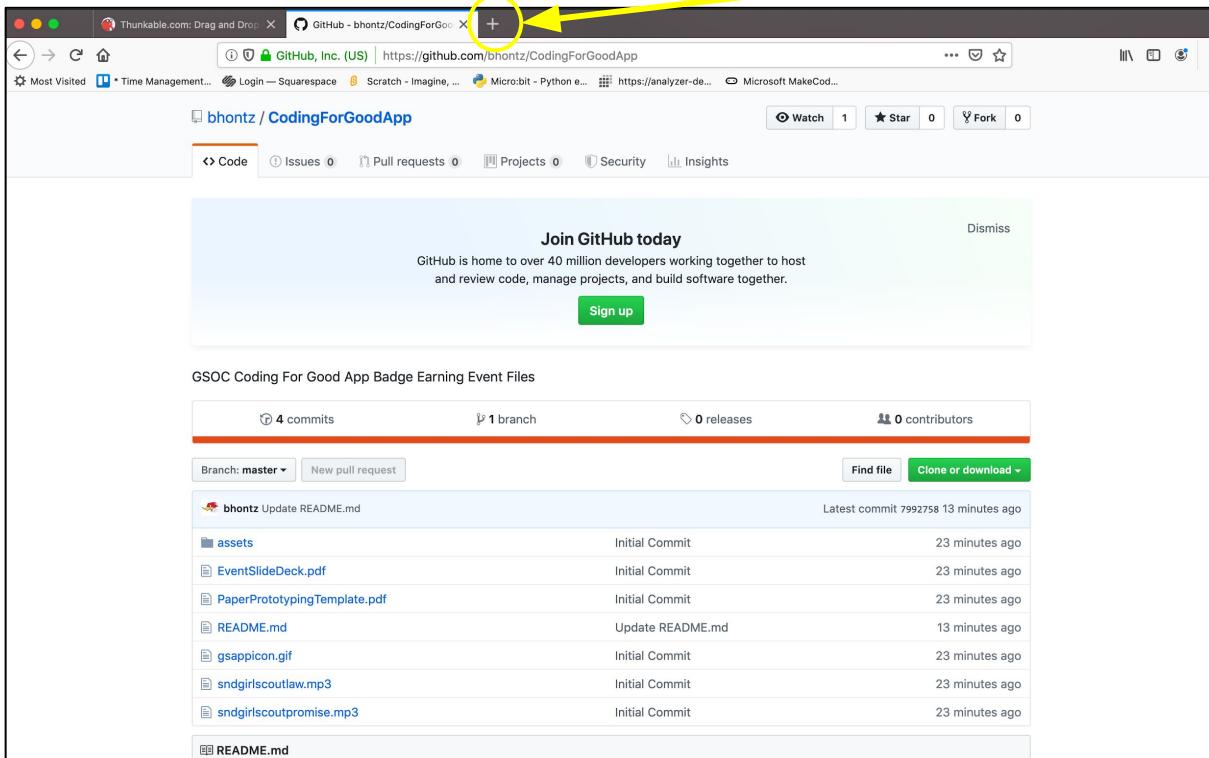


# GitHub - getting files we'll use today

Open a new browser window, and type in the URL:

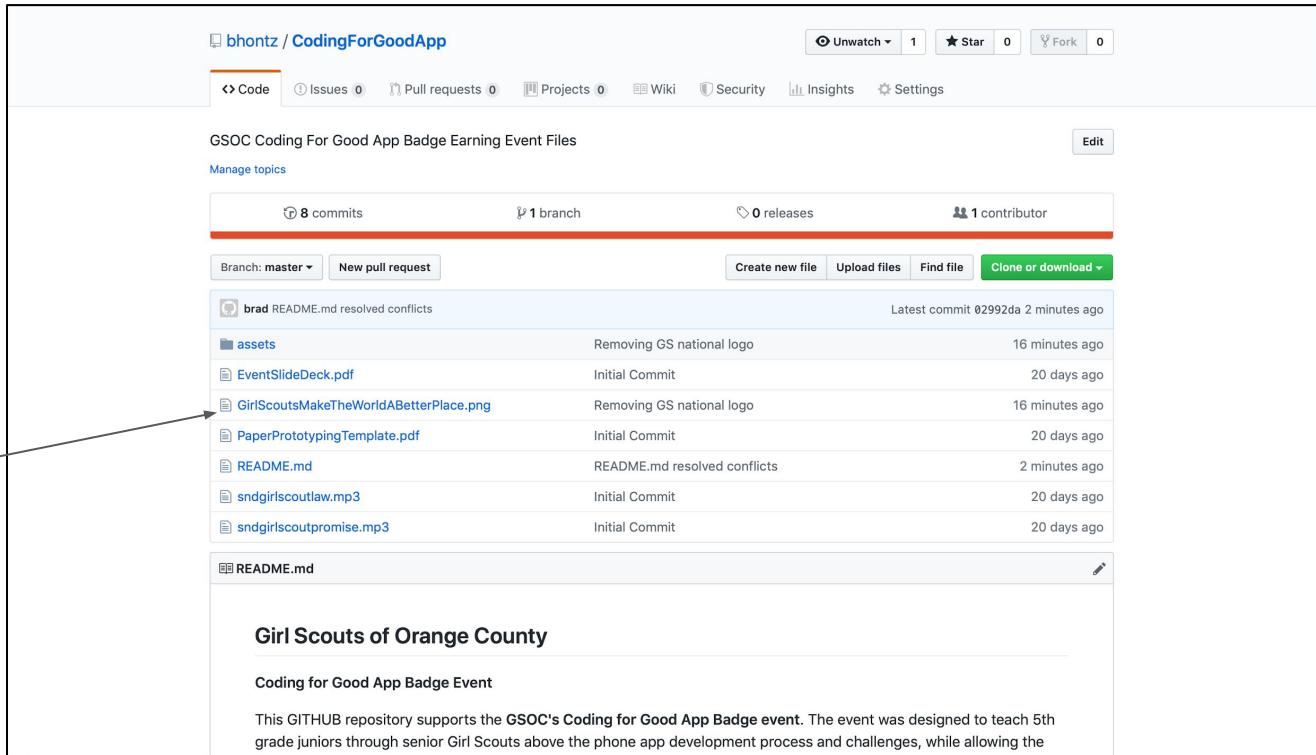
<https://github.com/bhontz/CodingForGoodApp>

Open a new  
Browser Window



# GitHub - getting files we'll use today

Double click on the file **GirlScoutsMakeTheWorldABetterPlace.png** as shown below.



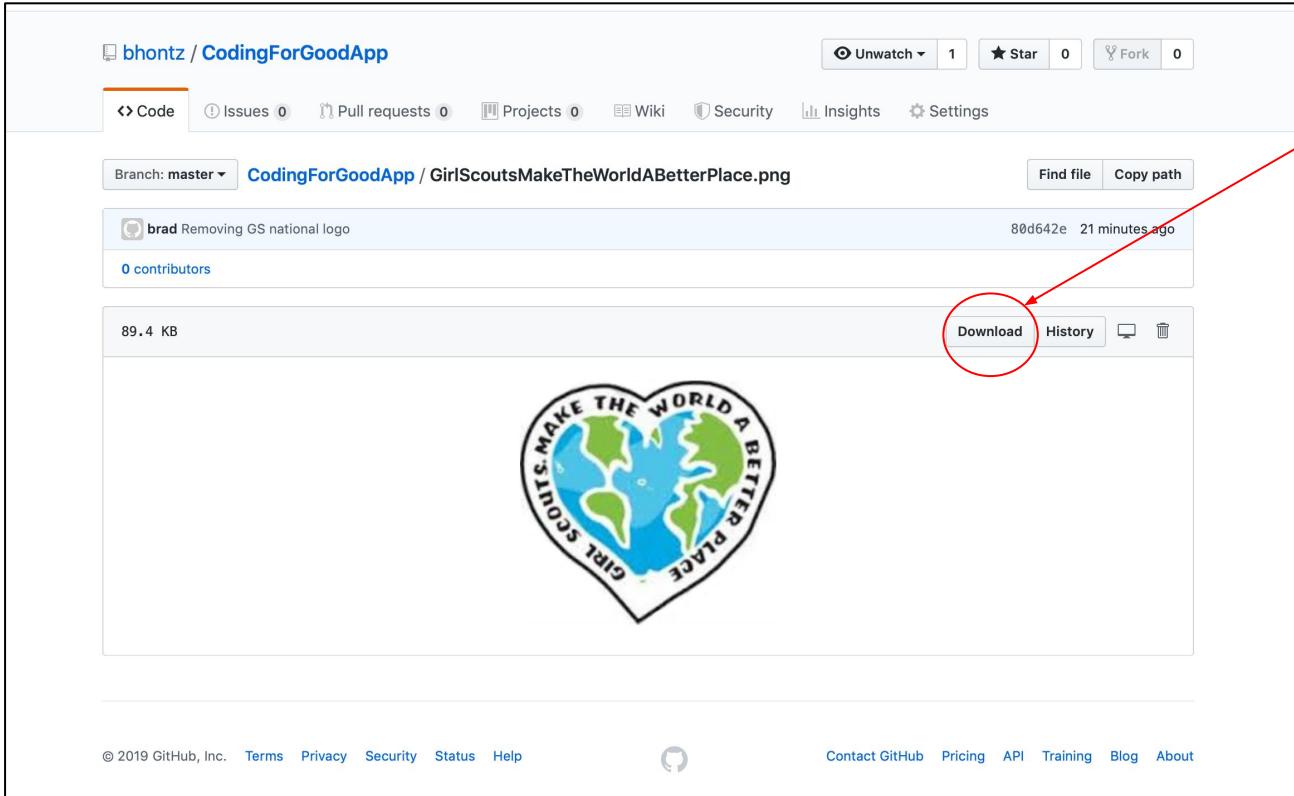
The screenshot shows a GitHub repository page for 'bhontz / CodingForGoodApp'. The repository has 8 commits, 1 branch, 0 releases, and 1 contributor. The latest commit was made 2 minutes ago. The repository contains several files, including 'README.md', 'EventSlideDeck.pdf', 'PaperPrototypingTemplate.pdf', 'README.md', 'sndgirlscoutlaw.mp3', 'sndgirlscoutpromise.mp3', and 'README.md'. A callout box with the text 'Click on this file' points to the file 'GirlScoutsMakeTheWorldABetterPlace.png'.

| File                                   | Description               | Time Ago       |
|--|---------------------------|----------------|
| README.md                              | resolved conflicts        | 2 minutes ago  |
| EventSlideDeck.pdf                     | Initial Commit            | 20 days ago    |
| GirlScoutsMakeTheWorldABetterPlace.png | Removing GS national logo | 16 minutes ago |
| PaperPrototypingTemplate.pdf           | Initial Commit            | 20 days ago    |
| README.md                              | resolved conflicts        | 20 days ago    |
| sndgirlscoutlaw.mp3                    | Initial Commit            | 20 days ago    |
| sndgirlscoutpromise.mp3                | Initial Commit            | 20 days ago    |
| README.md                              |                           |                |

**Click on this file**

# GitHub - getting files we'll use today

We need to save this graphic file to our DOWNLOADS folder. Click the Download button.



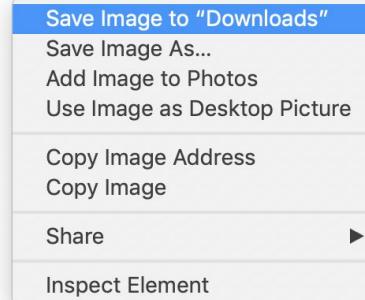
The screenshot shows a GitHub repository page for 'bhontz / CodingForGoodApp'. The repository has 1 watch, 0 stars, and 0 forks. The master branch contains a file named 'CodingForGoodApp / GirlScoutsMakeTheWorldABetterPlace.png'. The file was last modified by 'brad' 21 minutes ago, with the commit message 'Removing GS national logo'. The file size is 89.4 KB. At the bottom of the file preview, there is a 'Download' button, which is circled in red with a red arrow pointing to it from the text above. The GitHub footer includes links for Contact GitHub, Pricing, API, Training, Blog, and About.

© 2019 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub Pricing API Training Blog About

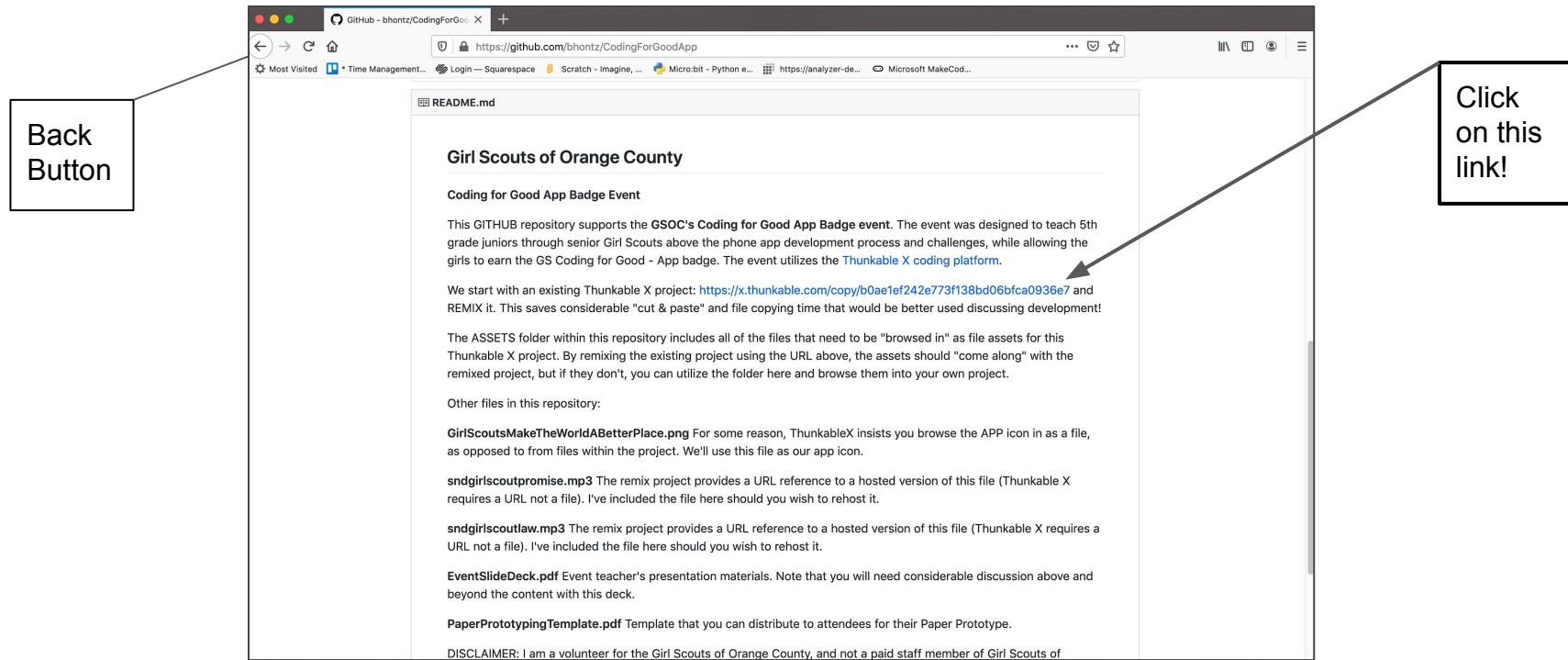
# GitHub - getting files we'll use today

Right Click on this screen and “Save Image to Downloads”. Your menu might look a little different than this! This step should save the file **GirlScoutsMakeTheWorldABetterPlace.png** to your computer’s DOWNLOADS file folder. We’ll use this file very soon.



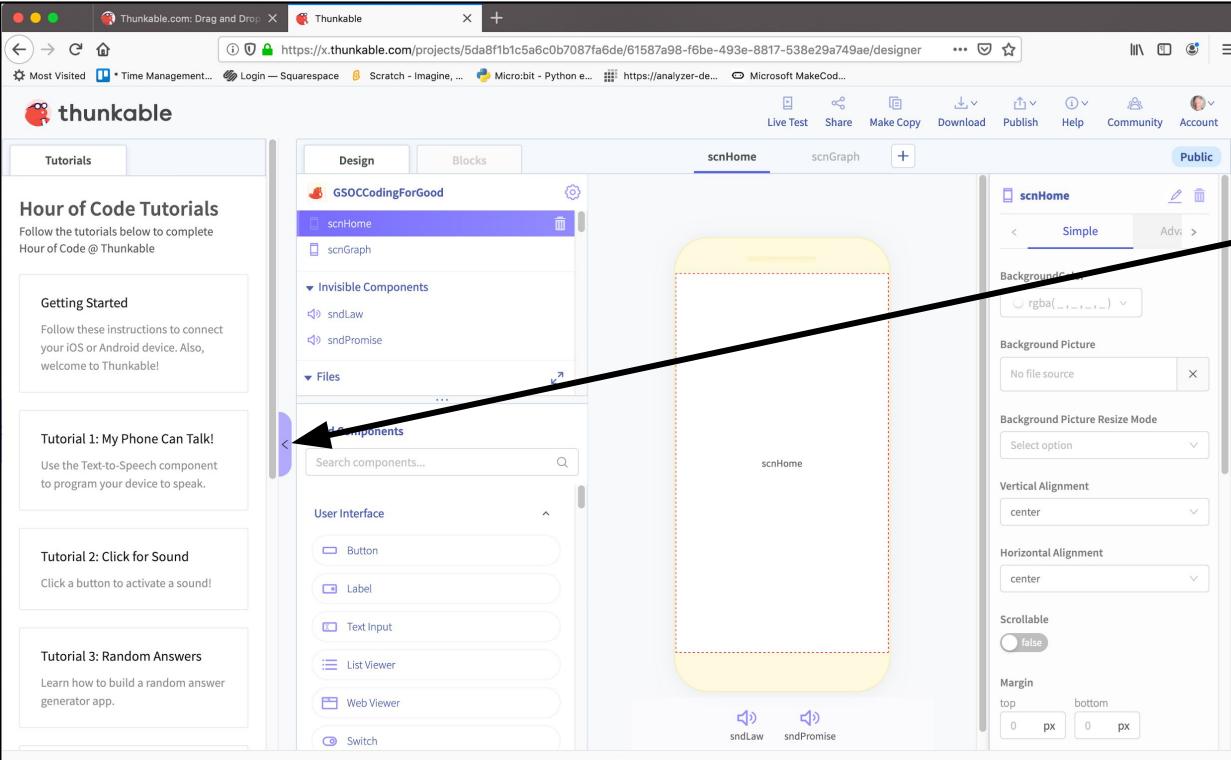
# GitHub - getting files we'll use today

Click the Back Button on your browser until you return to the list of files we started with, then scroll down until you can see the README.md document as shown below. Click on the link shown in the second paragraph.



# Getting Started - ThunkableX

**Congratulations!!!** Let's get started! Note that you can access TUTORIALS using the left side pane, but we'll close the tutorial pane for the remainder of our event.

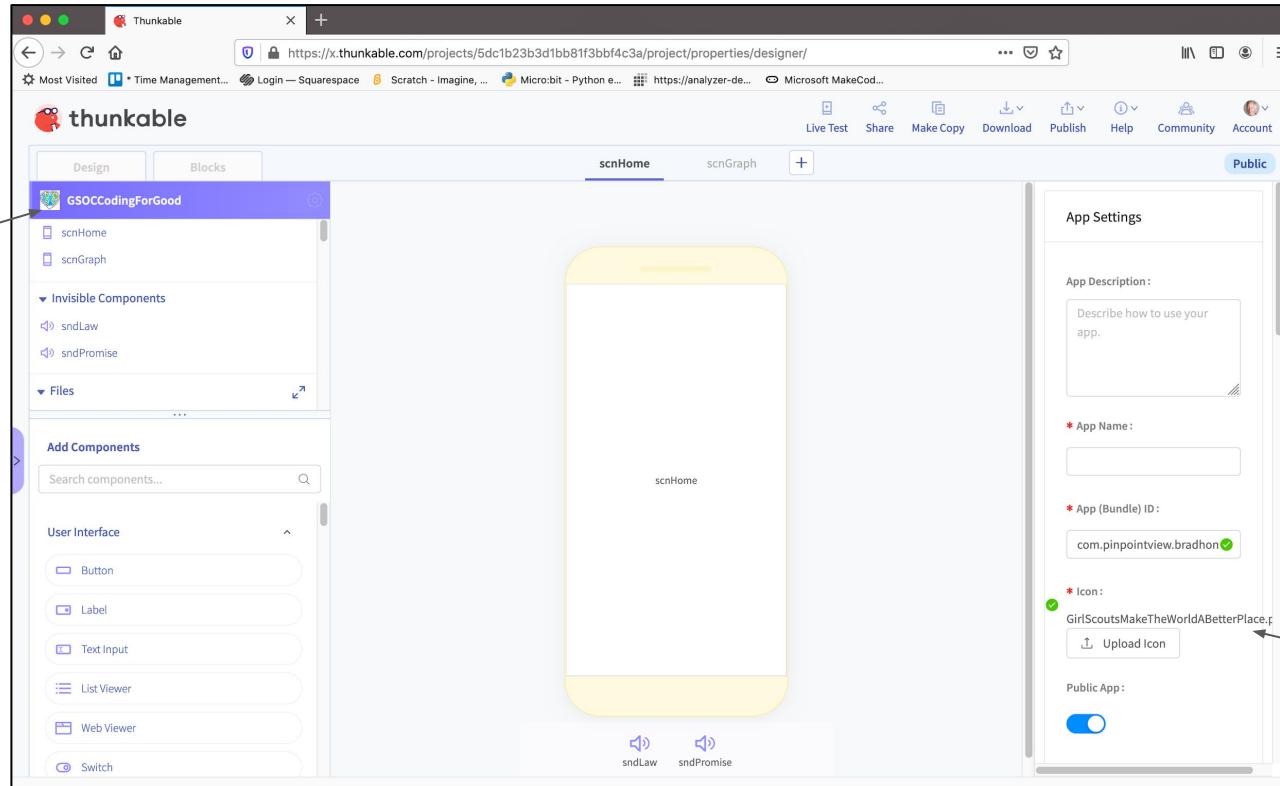


The screenshot shows the ThunkableX designer interface. On the left, there is a sidebar titled "Tutorials" containing several Hour of Code Tutorials: "Getting Started", "Tutorial 1: My Phone Can Talk!", "Tutorial 2: Click for Sound", and "Tutorial 3: Random Answers". A large black arrow points from the "Tutorial 1" section towards the left edge of the screen, indicating where to click to close the tutorial pane. The main workspace shows a scene titled "scnHome" with a yellow rounded rectangle component. The right panel displays the component's properties, including "Background Color" set to "Simple" with "rgba(255, 255, 0, 1)" and "Vertical Alignment" set to "center". Below the workspace, two sound effect icons are labeled "sndLaw" and "sndPromise".

Click here  
to close  
the  
tutorials  
pane

# Screens, Components and Blocks!

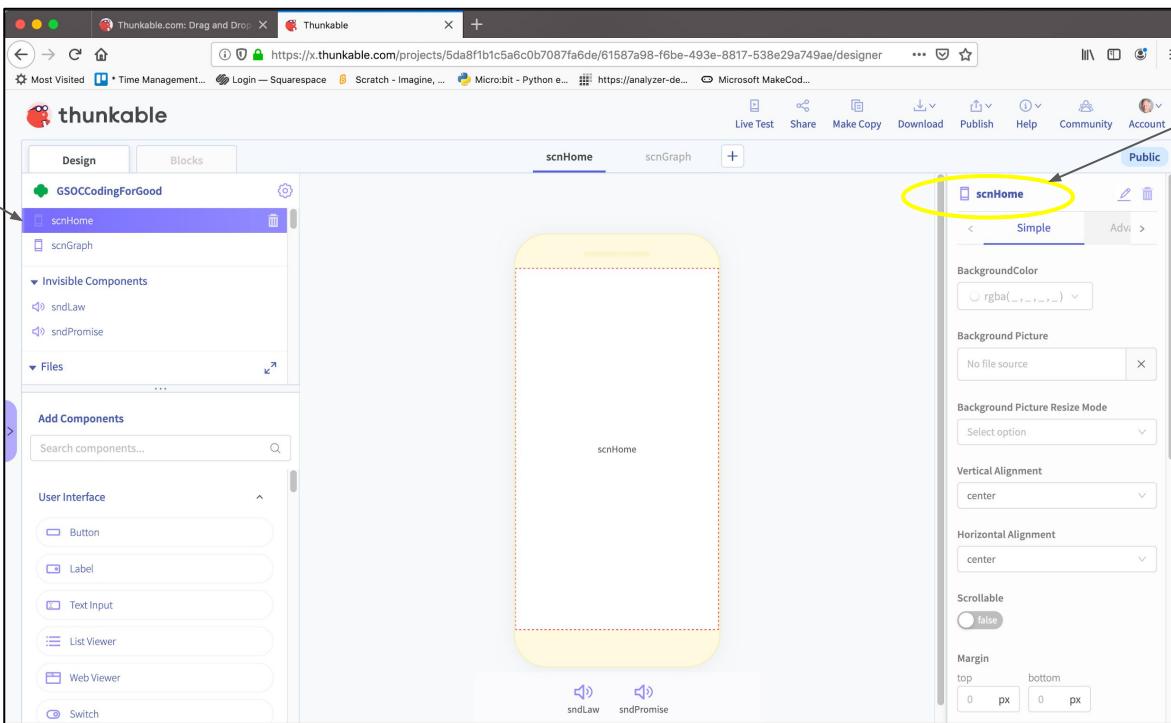
Let's talk about App Settings, Screens, Components, and Blocks! Use the ATTRIBUTE PANE on the right to assign our app an icon (icon that we'll see on our phone).



**GirlScoutsMakeTheWorldABetterPlace.png** should be in your DOWNLOADS folder!

# Naming conventions we'll use

The following slide will contain a list of naming conventions that we'll use for this project. App developers use conventions like this to make it easier to understand the screens and components they are referencing within their code.



A screenshot of the Thunkable app interface. On the left, there's a sidebar with 'Design' and 'Blocks' tabs, a project list ('GSOCCodingForGood'), and component categories like 'User Interface' (Button, Label, Text Input, List Viewer, Web Viewer, Switch). In the center, a mobile phone-shaped canvas has a yellow rounded rectangle labeled 'scnHome'. Below the canvas are two sound icons labeled 'sndLaw' and 'sndPromise'. On the right, a panel shows the screen's properties: 'Public' tab selected, 'BackgroundColor' set to 'rgba(255, 255, 255)', 'Vertical Alignment' and 'Horizontal Alignment' both set to 'center', and 'Margin' values of '0 px'. A yellow circle highlights the 'scnHome' label in the properties panel. A callout box with the text 'I used this name (click on the pencil)' points to the pencil icon next to the 'scnHome' label. Another callout box with the text 'I clicked here' points to the 'scnHome' label in the project list on the left.

I clicked here

I used this name (click on the pencil)

# Naming Conventions - Reference

---

You will find it much easier to code your app when using meaningful PREFIXes when we name our app's screens and components:

**scn** = Screen (e.g. scnHome, scnLaw, scnPromise ...)

**btn** = Button (e.g. btnHomeLaw, btnLawReturn, btnPromiseSpeak ...)

**lst** = List (e.g. lstTroopMembers)

**row** = Row layout container component (e.g. rowHomeBotttom)

**col** = Column layout container component (e.g. colHomeTop)

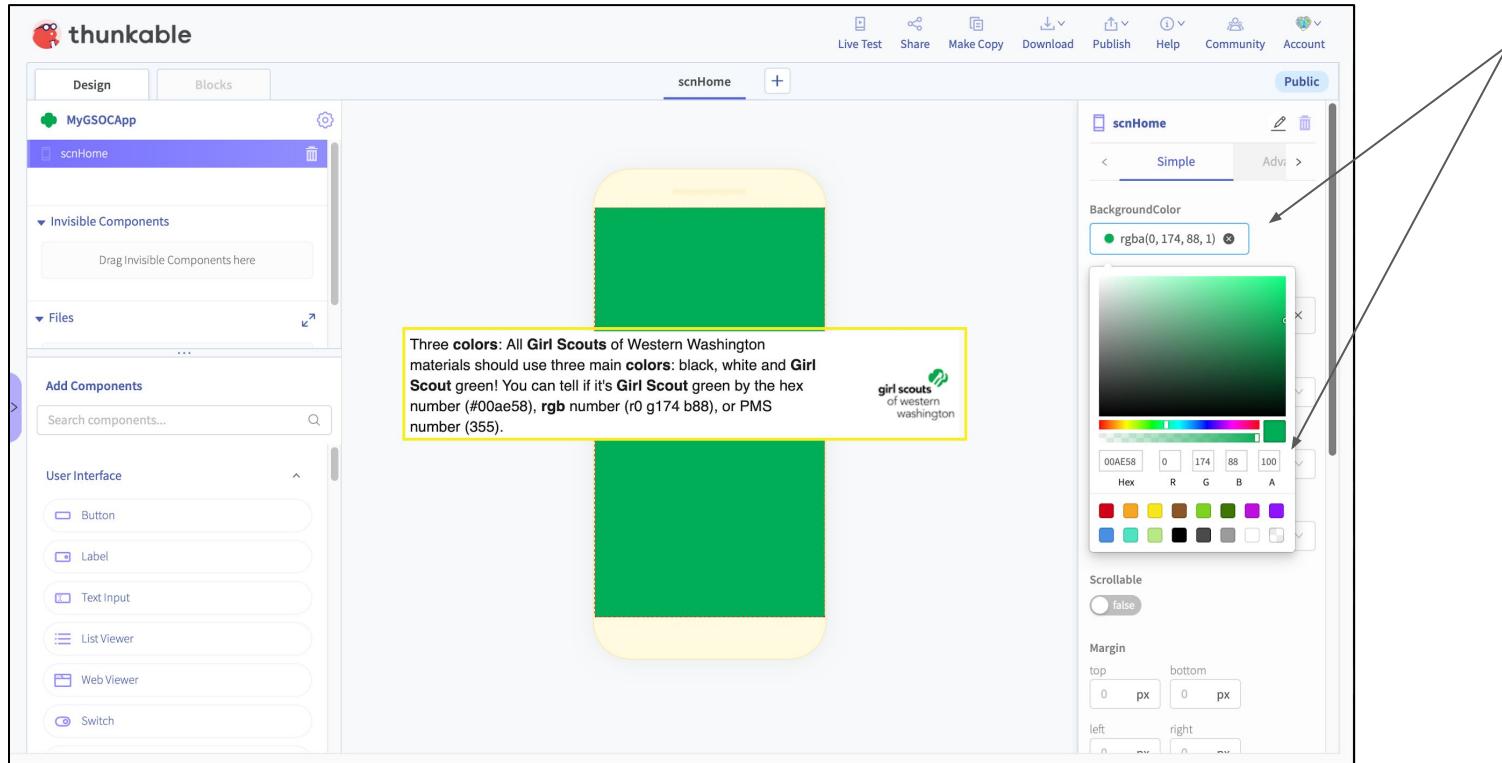
**snd** = Sound file object (e.g. sndPromise, sndLaw)

**img** = Image file object (e.g. imgGSOCLogo)

**wv** = “Web Viewer” (e.g. wvPromise, wvGraph...)

# Home Screen Background Color

Let's use Girl Scout colors for our home screen background! The RGB(A) representation of "Girl Scout Green" is R = 0, G = 174, B = 88. Set our background as shown below:



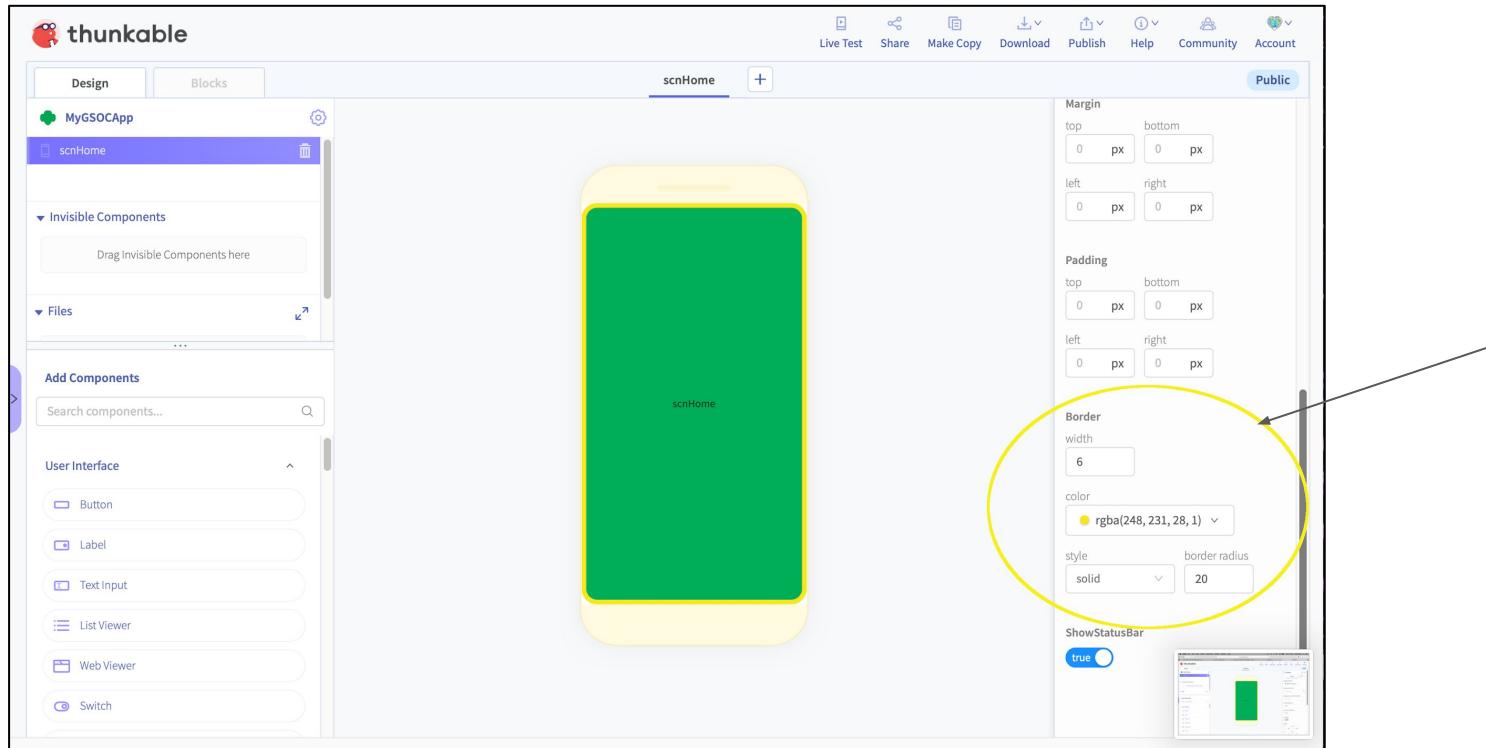
The screenshot shows the Thunkable app interface for a project titled "MyGSOCApp". The scene is named "scnHome". On the right, the scene editor shows the "BackgroundColor" property set to "rgba(0, 174, 88, 1)". A color picker panel is open, displaying a gradient from black to green, with the green color selected. The color code "#00AE58" is also visible. The scene itself is a yellow rounded rectangle containing a solid green square.

**Three colors:** All Girl Scouts of Western Washington materials should use three main colors: black, white and **Girl Scout green!** You can tell if it's Girl Scout green by the hex number (#00ae58), rgb number (r0 g174 b88), or PMS number (355).

girl scouts  
of western  
washington

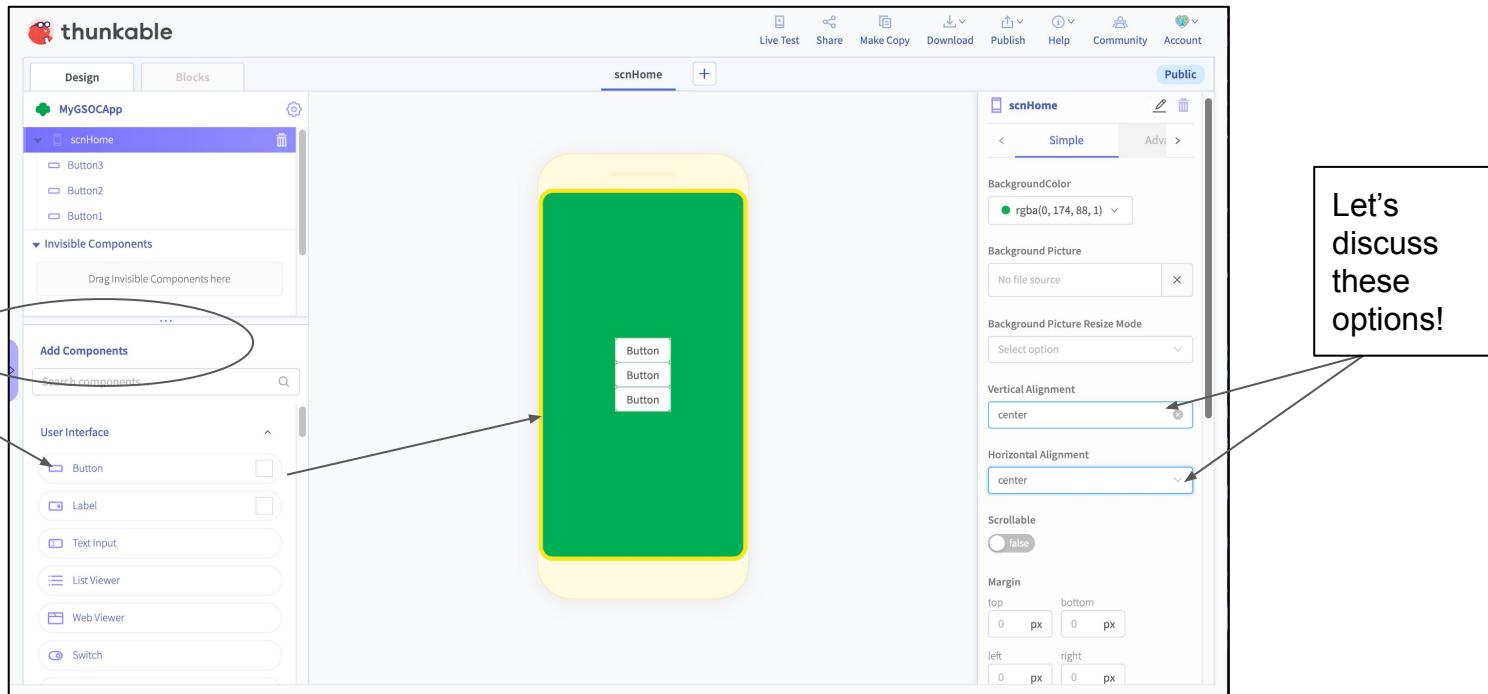
# Home Screen: Set a Border

A border might look cool on our home screen. Scroll the scnHome ATTRIBUTE WINDOW down a bit and set the border width, color, and radius. Radius determines how round the border is.



# So what is “Layout” anyway?

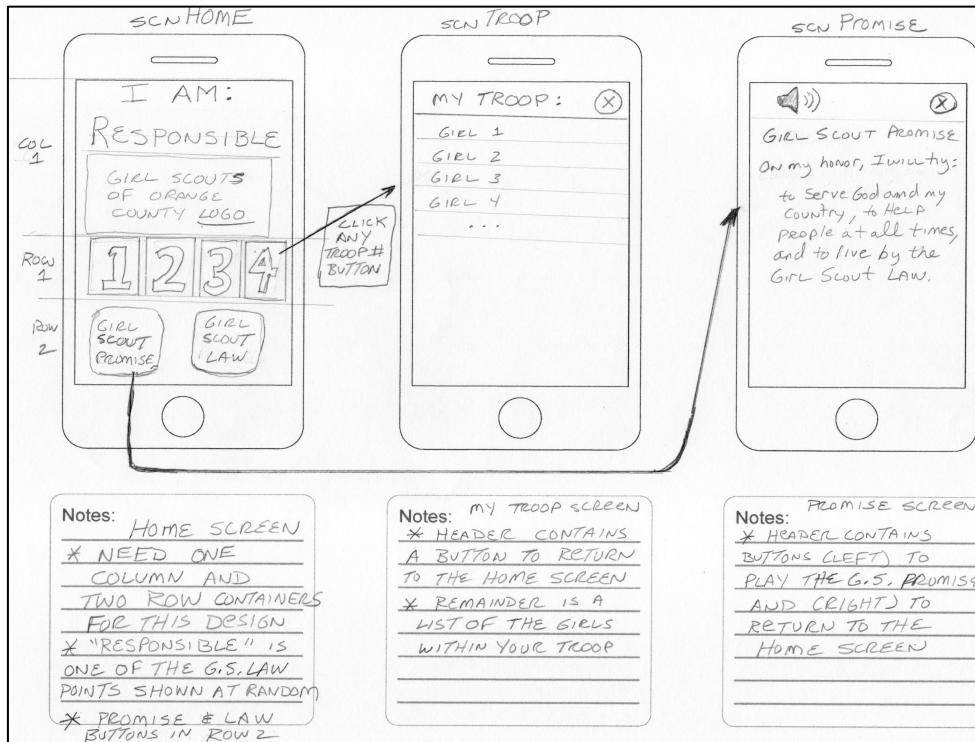
An app designer gives considerable thought to the user interface (UI) of the app. App development tools generally provide “containers” that hold UI components like buttons and sliders. The container determines the **alignment and orientation** of the components placed within it. Let’s drag three buttons onto our home screen in order to explore how layout works!



The screenshot shows the Thunkable app editor interface. On the left, the sidebar lists "Design" and "Blocks", the project name "MyGSOCAPP", and the scene "scnHome". Inside "scnHome", there are three buttons labeled "Button1", "Button2", and "Button3". Below them is a placeholder for "Invisible Components". The "User Interface" section includes components like "Button", "Label", "Text Input", "List Viewer", "Web Viewer", and "Switch". In the center, a smartphone-shaped canvas displays the three buttons stacked vertically. To the right, the "scnHome" properties panel is open, showing settings for "BackgroundColor" (set to `rgba(0, 174, 88, 1)`), "Vertical Alignment" (set to "center"), and "Horizontal Alignment" (set to "center"). A callout box on the right says "Let's discuss these options!" with arrows pointing to the alignment settings.

# Help! Take a step back - we need a plan!

App designers use **Paper Prototyping** to plan their app. An example is shown below. Here we've captured the functionality of the design, but not the final look (e.g. colors, graphic images, etc.), which we might capture in a second version of our paper prototype. Now create your own paper prototype!



# YOUR paper prototype - goals for our app

---

## Our desired functionality:

- Our app allows us to read (and listen to) the Girl Scout Law and Promise.
- Each time we open our app, we're reminded of one of the points of the Girl Scout Law (e.g. "Courageous").
- Our app graphs cookies sales (or estimates) for each of the members of our troop.

We will need a Home Screen, Graph Screen, Law Screen and Promise Screen to accomplish the goals above! [e.g. scnHome, scnGraph, scnLaw, scnPromise]

Our Home Screen should display random Girl Scout Law points, contain our troop number (clicking on our troop number navigates to the Graph Screen), as well as buttons for navigating to the Girl Scout Law and Promise screens.

***The rest of YOUR app design is up to you!!***

# YOUR design - our UI Assets



asterisk.png



asterisk2.png



Banana.png



btnGSLaw.jpg



btnGSpromise.jp  
g



cat1.png



cat4.png



clock.png



CopyAndPasteLi  
nks.txt



explosion.png



GirlScoutsMakeT  
heWor...lace.png



GS\_GIRL\_logo.jp  
g



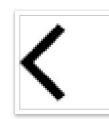
GSLaw.html



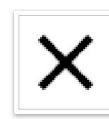
GSOC\_ServiceM  
ark.jpg



GSPromise.html



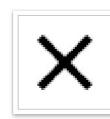
icon\_back\_arrow  
.png



icon\_close\_clear  
bg.png



icon\_close\_w\_cir  
cle.png



icon\_close\_x.png



icon\_face.png



icon\_happy.png



icon\_listening.bn  
g



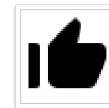
icon\_phone\_talki  
ng.png



icon\_speaker.bn  
g



icon\_talk\_30px.p  
ng



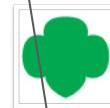
icon\_thumbs\_up.p  
ng



icon\_wrench.png



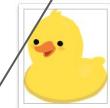
imgGSOCSTEMP  
atrol.jpg



imgGSTrefoil.png



lightningbolt.bn  
g



rubber-duck.png



taco.png



TroopNbr\_0.jpg



TroopNbr\_1.jpg



TroopNbr\_2.jpg



TroopNbr\_3.jpg



TroopNbr\_4.jpg



TroopNbr\_5.jpg



TroopNbr\_6.jpg



TroopNbr\_7.jpg



TroopNbr\_8.jpg



TroopNbr\_9.jpg

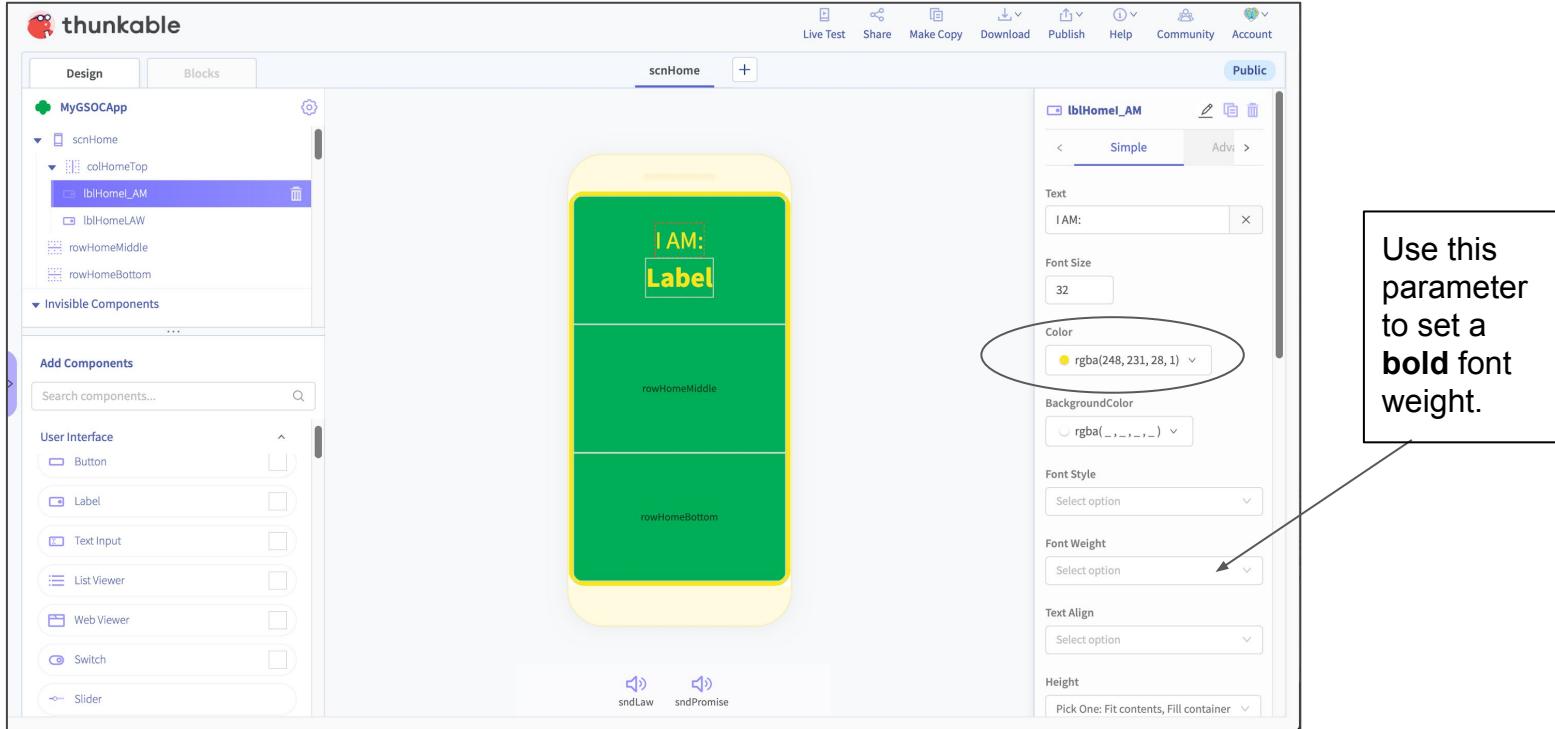


unicorn.png

Here are some great “close button” graphics!

# Guided Design - scnHome

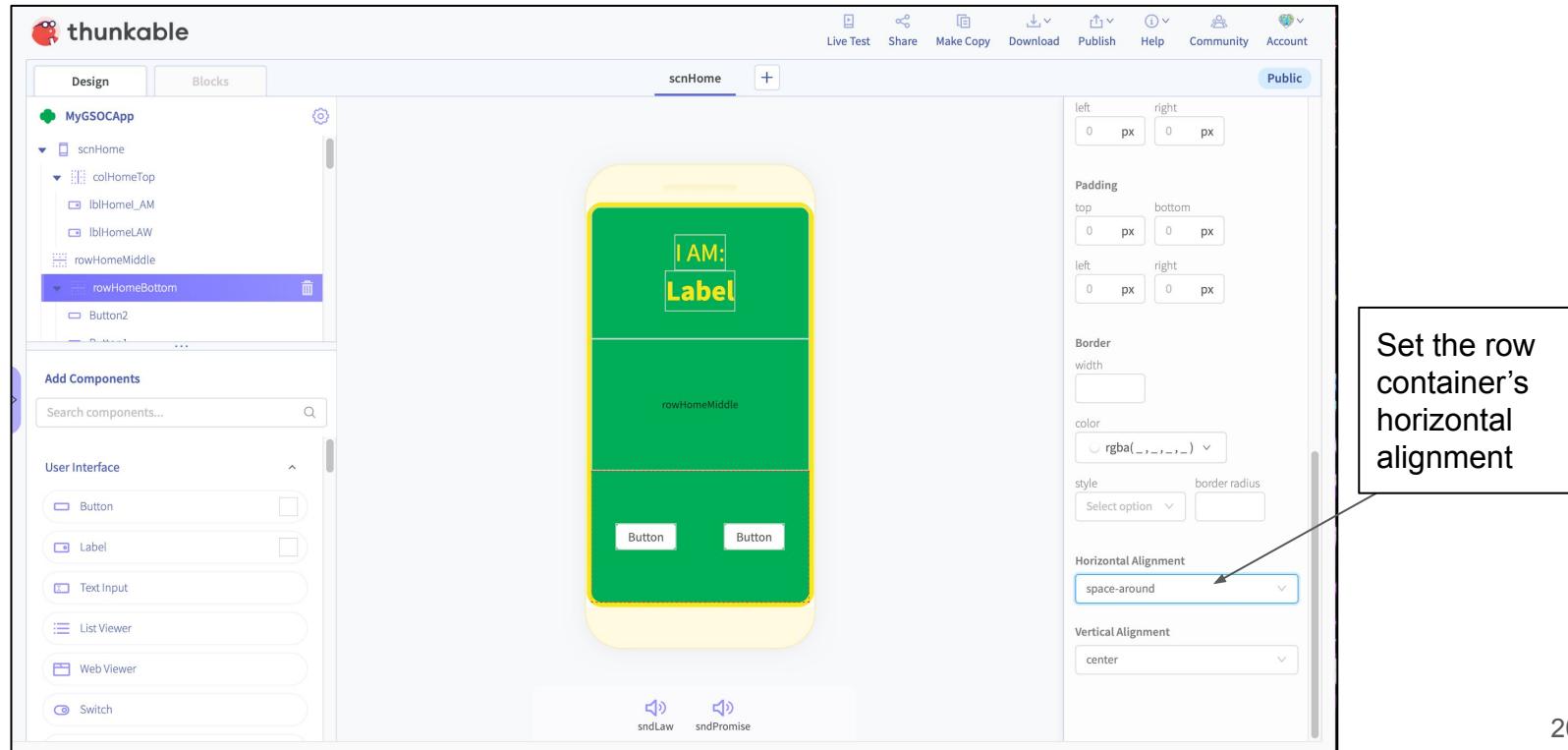
Add a column container and two row containers to the Home screen as shown, then add two Label components to the column container. Set the font sizes of the labels to 32, and use the same “yellow” color for the labels as our border. Bold the bottom label (lblHomeLAW).



Use this parameter to set a **bold** font weight.

# Guided Design - Law and Promise Buttons

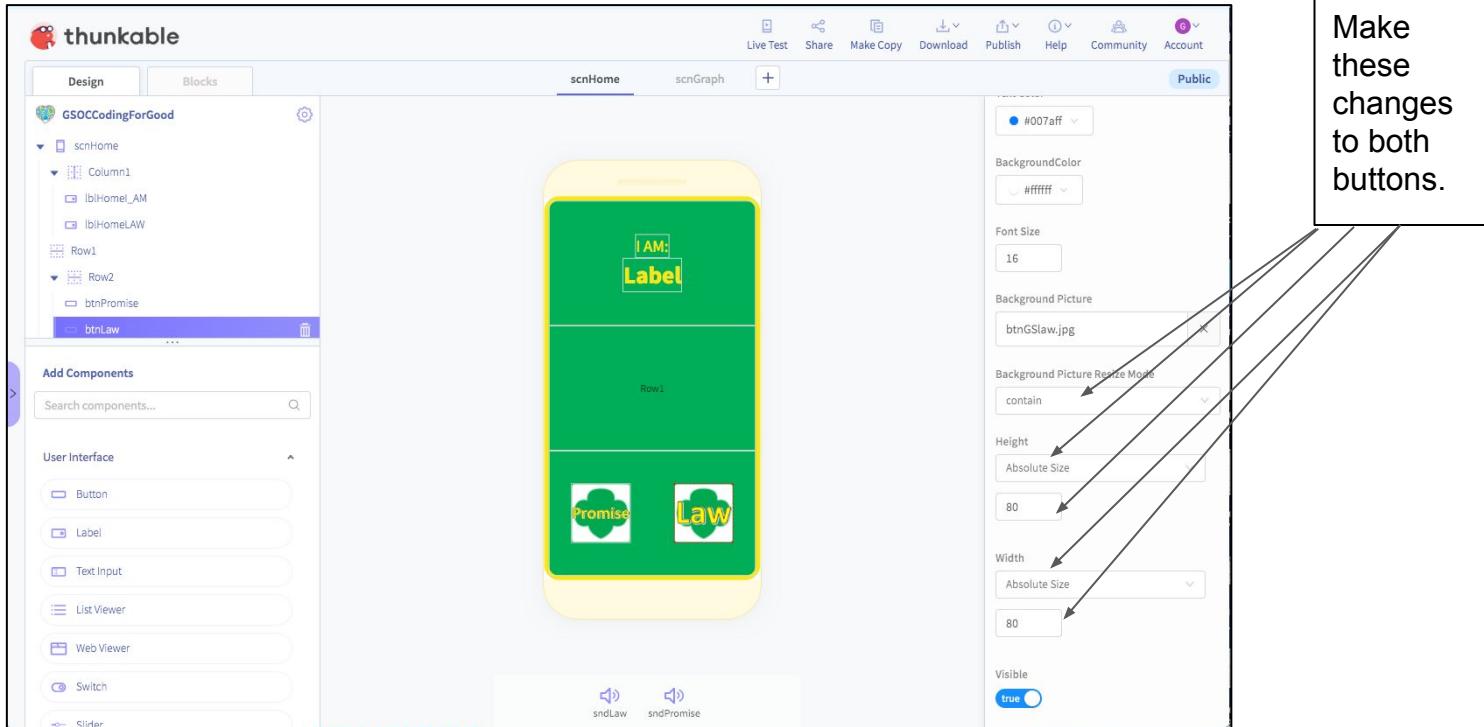
Drag two buttons onto the rowHomeBottom container. Click on the container (outside of the buttons) to select it, then set the Horizontal Alignment in the ATTRIBUTE PANE of the row container to space-around.



Set the row container's horizontal alignment

# Guided Design - Law and Promise Buttons

Select each button in order, changing the name (e.g. btnLaw) and selecting a Background Picture for the button face. Resize the button so the picture fits better, and change the background resize mode to “contain”. We used a value of 80 for the height and width of the button, but you may wish to adjust that after viewing your app on your phone.



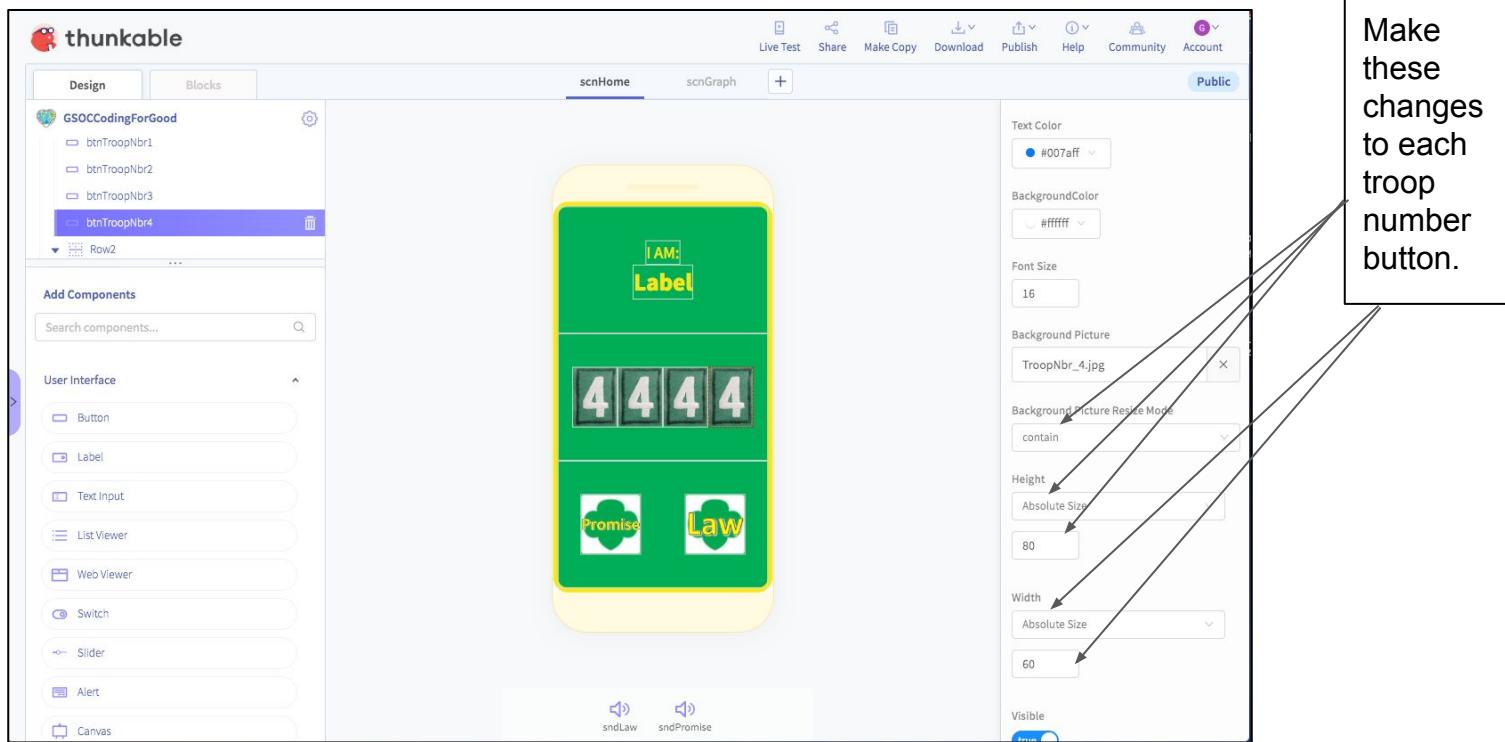
The screenshot shows the Thunkable app editor interface. On the left, the project tree shows a scene named 'scnHome' with components: Column1, Row1, Row2, btnPromise, and btnLaw. The main canvas displays a smartphone screen with a green background. Inside the screen, there is a yellow rectangular area labeled 'Row1'. At the top of 'Row1', there is a small white box containing the text 'I AM: Label'. Below this, in the center of 'Row1', are two green buttons with white text: 'Promise' on the left and 'Law' on the right. At the bottom of the screen, there are two small speaker icons labeled 'sndLaw' and 'sndPromise'. On the right side of the editor, there is a detailed view of the 'btnLaw' component settings. The settings include:

- Background Color: #007aff
- Background Picture: btnGSLaw.jpg
- Background Picture Resize Mode: contain
- Height: 80 (Absolute Size)
- Width: 80 (Absolute Size)
- Visible: true

A callout box with the text "Make these changes to both buttons." has arrows pointing to the 'Background Picture', 'Background Picture Resize Mode', 'Height', and 'Width' settings.

# Guided Design - Troop Number Buttons

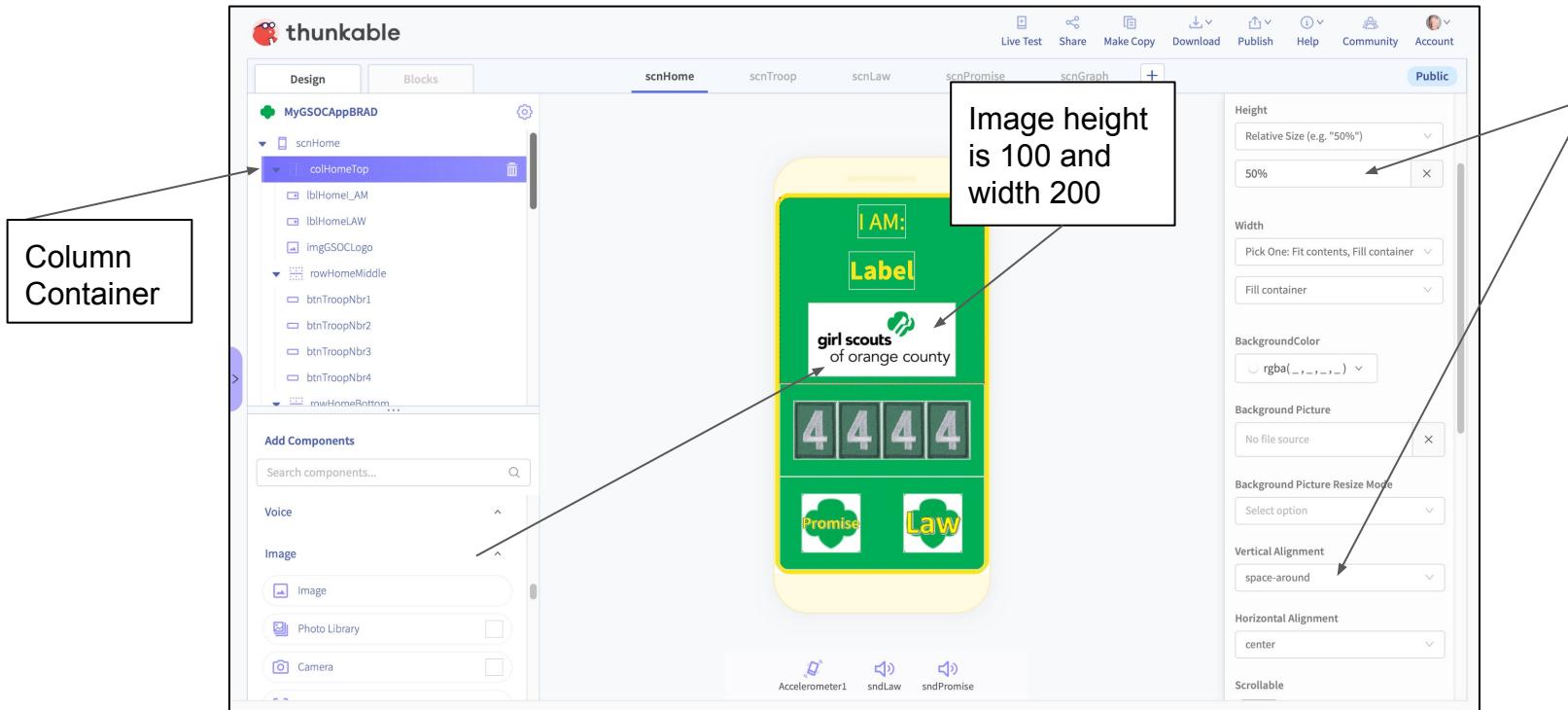
How many digits are in your troop number? Add that many buttons into the middle row container, then set the background picture for each button appropriately. Set the Background resize mode to contain, height to 80 and width to 60. Did you name your buttons btnTroopNbr1 (etc.)?



Make these changes to each troop number button.

# Guided Design - Add a logo!

Drag an Image component onto the column container, being sure to place it beneath our lblHomeLaw label. **Wow, too tight!** Click on the column container and set its height to 50% and its vertical alignment to space-around. Add a picture from our assets to the image component and rename it!

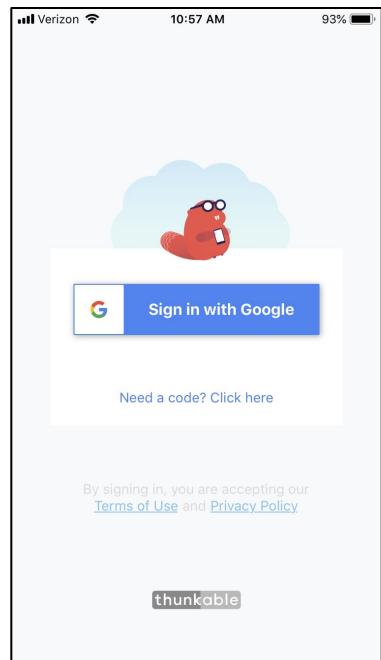


The screenshot shows the Thunkable app interface with the following elements:

- Column Container:** A callout box points to the "colHomeTop" column container in the project tree.
- Image Component:** An image component is placed on the "colHomeTop" container, containing the "girl scouts of orange county" logo.
- Properties Panel:** A callout box highlights the properties panel on the right, which includes:
  - Height:** Set to "50%".
  - Width:** Set to "Fill container".
  - Background Color:** Set to "rgba(255, 255, 255)".
  - Background Picture:** Set to "No file source".
  - Background Picture Resize Mode:** Set to "Select option".
  - Vertical Alignment:** Set to "space-around".
  - Horizontal Alignment:** Set to "center".
  - Scrollable:** Set to "false".
- Text Overlay:** A callout box contains the text "Image height is 100 and width 200".

# See how it looks (so far) on our phone!

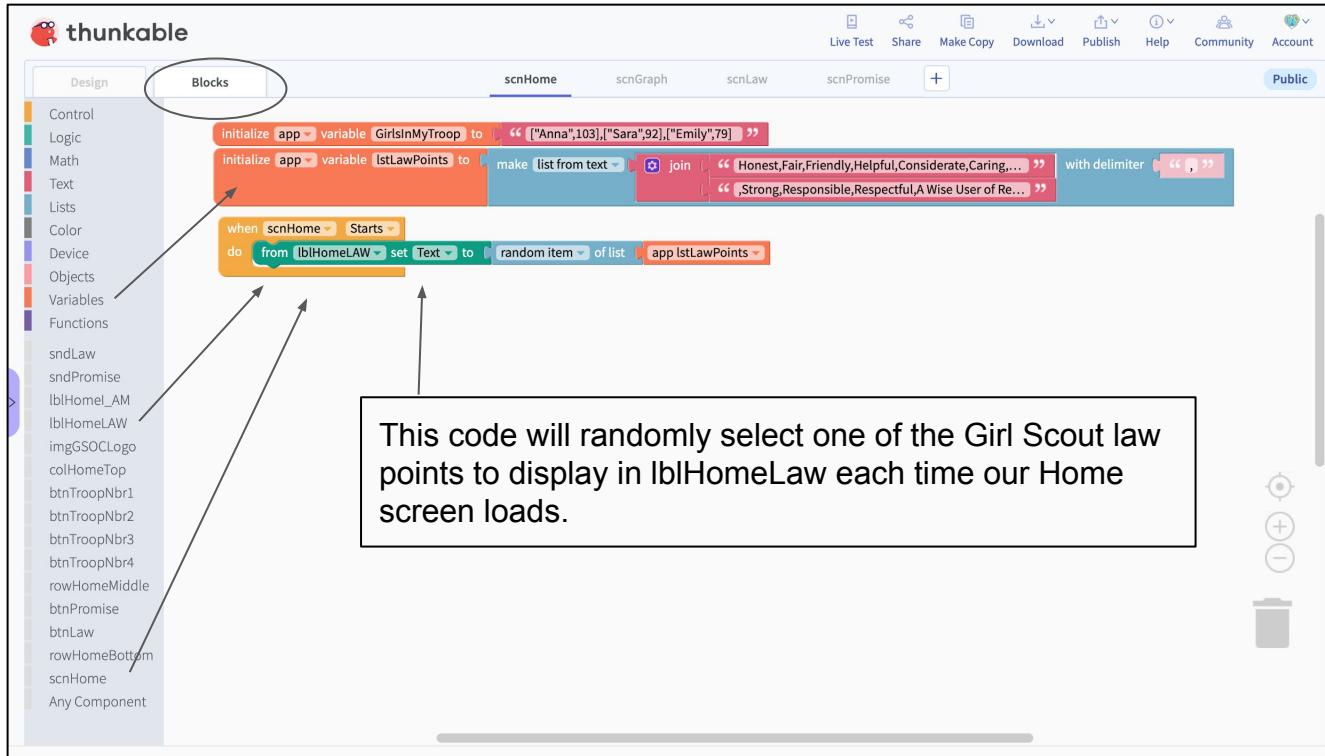
After installing the Thunkable Live app (App Store or Google Play), log in using your GMAIL ACCOUNT, then take a peek at your phone app's screen - it's coming along nicely!



Wait! How did  
THIS get on  
there??

# Guided Design - Let's Code!

Initialize a variable to a list which we'll build from text that we copy from our assets file CopyAndPasteLinks.txt. Click on scnHome and add a Starts block for scnHome, setting our lblHomeLaw component to a random item within our newly created list variable.



The screenshot shows the Thunkable app editor interface. The top navigation bar includes 'Live Test', 'Share', 'Make Copy', 'Download', 'Publish', 'Help', 'Community', and 'Account'. The current project is titled 'scnHome'.

The left sidebar lists components: Control, Logic, Math, Text, Lists, Color, Device, Objects, Variables, Functions, sndLaw, sndPromise, lblHome\_AM, lblHomeLAW, imgGSOCLogo, colHomeTop, btnTroopNbr1, btnTroopNbr2, btnTroopNbr3, btnTroopNbr4, rowHomeMiddle, btnPromise, btnLaw, rowHomeBottom, scnHome, Any Component. The 'Variables' section is highlighted with a red oval.

The main workspace contains the following blocks:

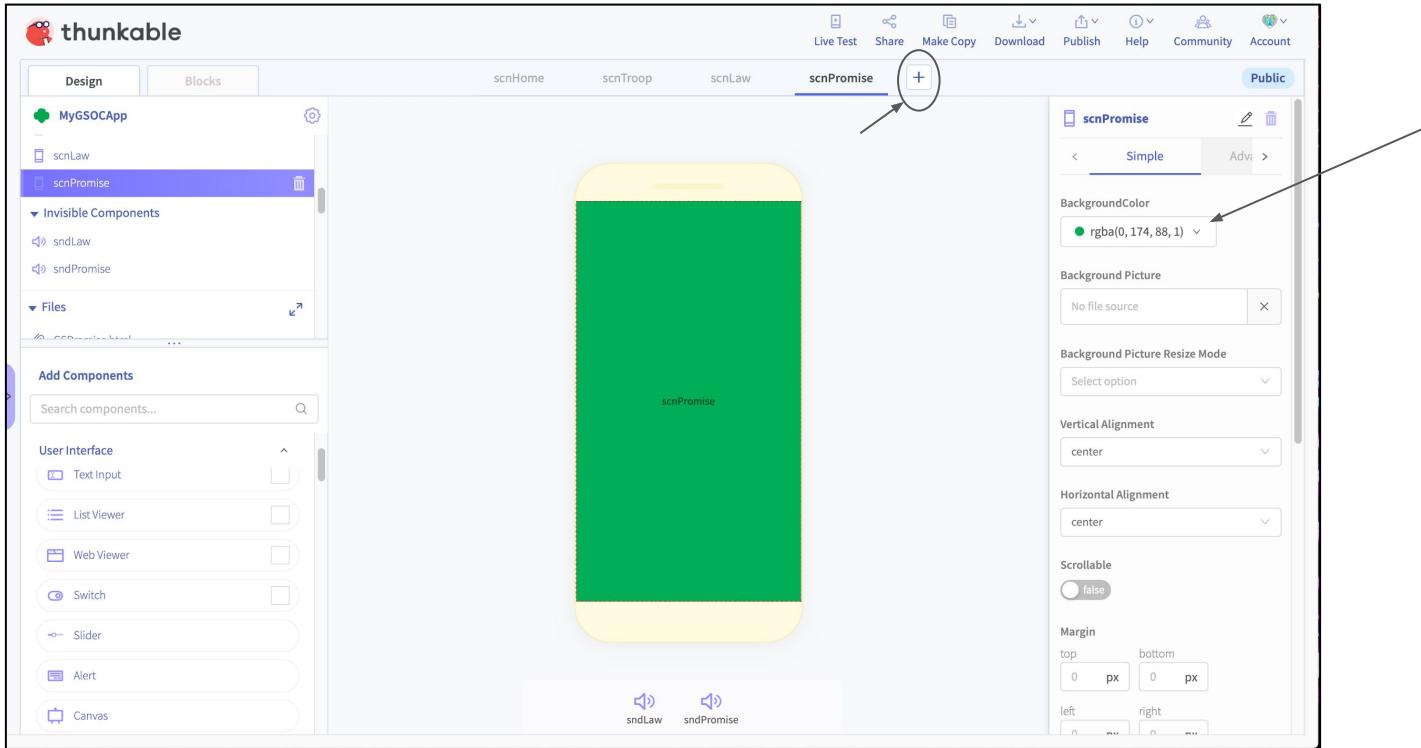
- An orange 'initialize app variable' block sets 'GirlsInMyTroop' to a list of three names: Anna, Sara, and Emily.
- A blue 'initialize app variable' block sets 'lstLawPoints' to a list of law points: Honest, Fair, Friendly, Helpful, Considerate, Caring, Strong, Responsible, Respectful, A Wise User of Re...
- A blue 'make list from text' block is part of the 'lstLawPoints' initialization.
- A pink 'join' block is used to combine the text elements of the list.
- A pink 'with delimiter' block specifies a comma as the delimiter for the list.
- An orange 'when scnHome Starts' control block triggers the following code.
- A green 'do' control block contains a 'from lblHomeLAW set Text to' block and a 'random item of list app lstLawPoints' block.

Arrows point from the explanatory text below to the 'Variables' sidebar, the 'initialize app variable' blocks, and the 'when scnHome Starts' block.

This code will randomly select one of the Girl Scout law points to display in lblHomeLaw each time our Home screen loads.

# Guided Design - adding screens!

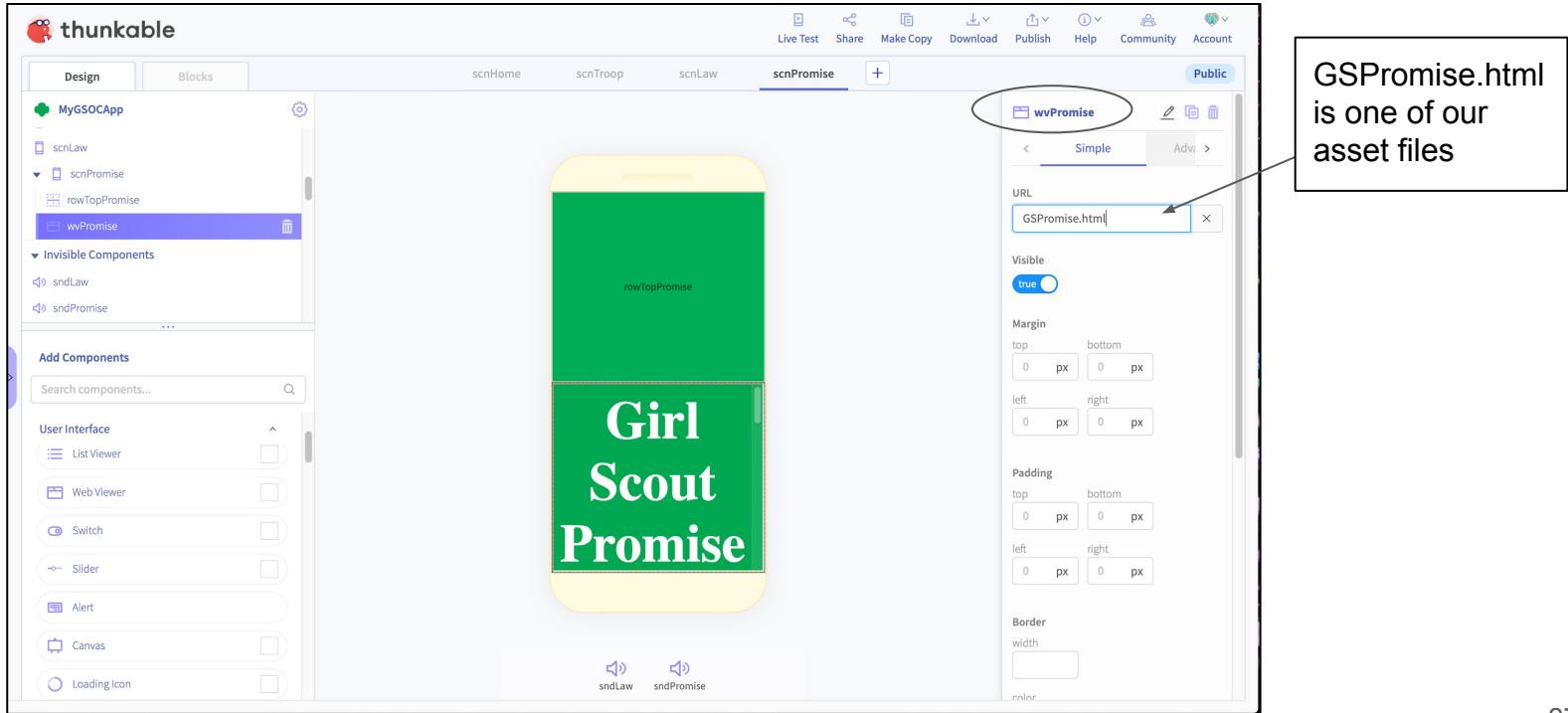
Use the + button to the right of the screen names to add **two** new screens, scnLaw and scnPromise. Change the background color on each of the three screens to the Girl Scout Colors, R = 0, G = 174 and B = 88 just as we did with scnHome.



The screenshot shows the thunkable app interface for a project titled "MyGSOCApp". On the left, the "Design" tab is selected, displaying a list of screens: "scnHome" (selected), "scnTroop", "scnLaw", "scnPromise" (highlighted in blue), and "Invisible Components" which include "sndLaw" and "sndPromise". Below this is a "Files" section. A sidebar on the left contains a "Add Components" section with various UI components like "Text Input", "List Viewer", "Web Viewer", "Switch", "Slider", "Alert", and "Canvas". The main workspace shows a smartphone-shaped screen with a solid green background labeled "scnPromise". Above the workspace are navigation and sharing buttons: "Live Test", "Share", "Make Copy", "Download", "Publish", "Help", "Community", and "Account". The "Public" status is shown. To the right of the screen are the screen settings: "BackgroundColor" set to "rgba(0, 174, 88, 1)", "Background Picture" (no file source), "Background Picture Resize Mode" (select option), "Vertical Alignment" (center), "Horizontal Alignment" (center), "Scrollable" (false), and "Margin" (top: 0px, bottom: 0px, left: 0px, right: 0px). An arrow points from the text "Change the background color on each of the three screens to the Girl Scout Colors, R = 0, G = 174 and B = 88 just as we did with scnHome." to the "BackgroundColor" dropdown. Another arrow points from the text "Use the + button to the right of the screen names to add two new screens, scnLaw and scnPromise." to the "+" button next to "scnPromise".

# Guided Design - Web Viewer Component

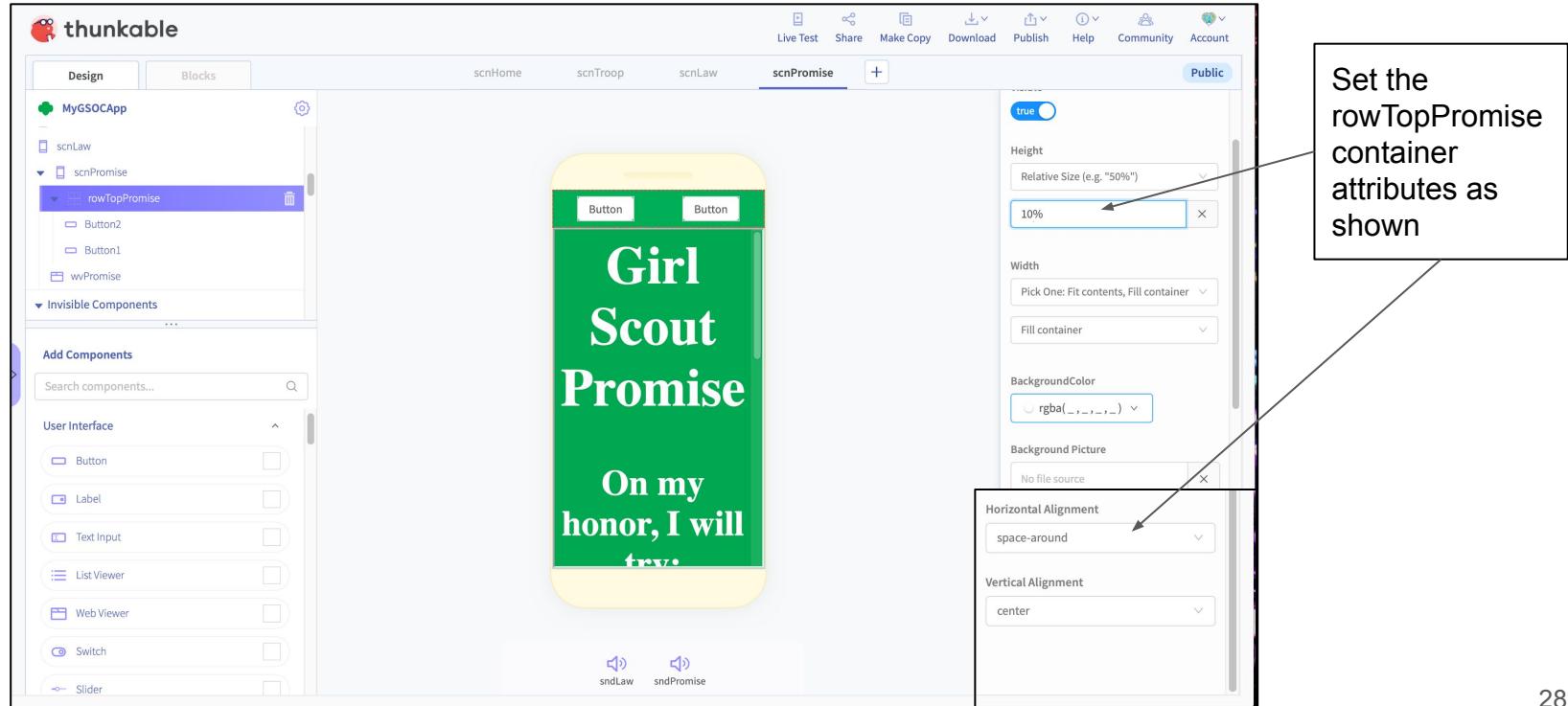
Drag a row container and a Web Viewer component onto the Promise screen. Type in the asset filename “GSPromise.html” into the web viewer component’s URL parameter on the ATTRIBUTE PANE of the web viewer. Note that we renamed the row and web viewer components.



The screenshot shows the Thunkable app interface for mobile application development. On the left, the project tree for "MyGSOCApp" shows a "scnPromise" scene with a "rowTopPromise" row and a "wvPromise" Web Viewer component. The main canvas displays a smartphone screen with a green background and the text "rowTopPromise" at the top and "Girl Scout Promise" in large white letters below. Below the phone are two sound effect icons labeled "sndLaw" and "sndPromise". The top navigation bar includes "Live Test", "Share", "Make Copy", "Download", "Publish", "Help", "Community", and "Account". The right side of the screen is the "ATTRIBUTE PANE" for the "wvPromise" component, which is set to "Simple" mode. The "URL" field contains the value "GSPromise.html". A callout box on the right points to this URL field with the text "GSPromise.html is one of our asset files".

# Guided Design - Promise Screen UI

Resize the row container so it represents only 10% of the height of our screen. Set the horizontal alignment of the row container to “space-around”, then add two buttons to the row container. We’ll use these buttons to play our promise sound file and to close the promise screen.



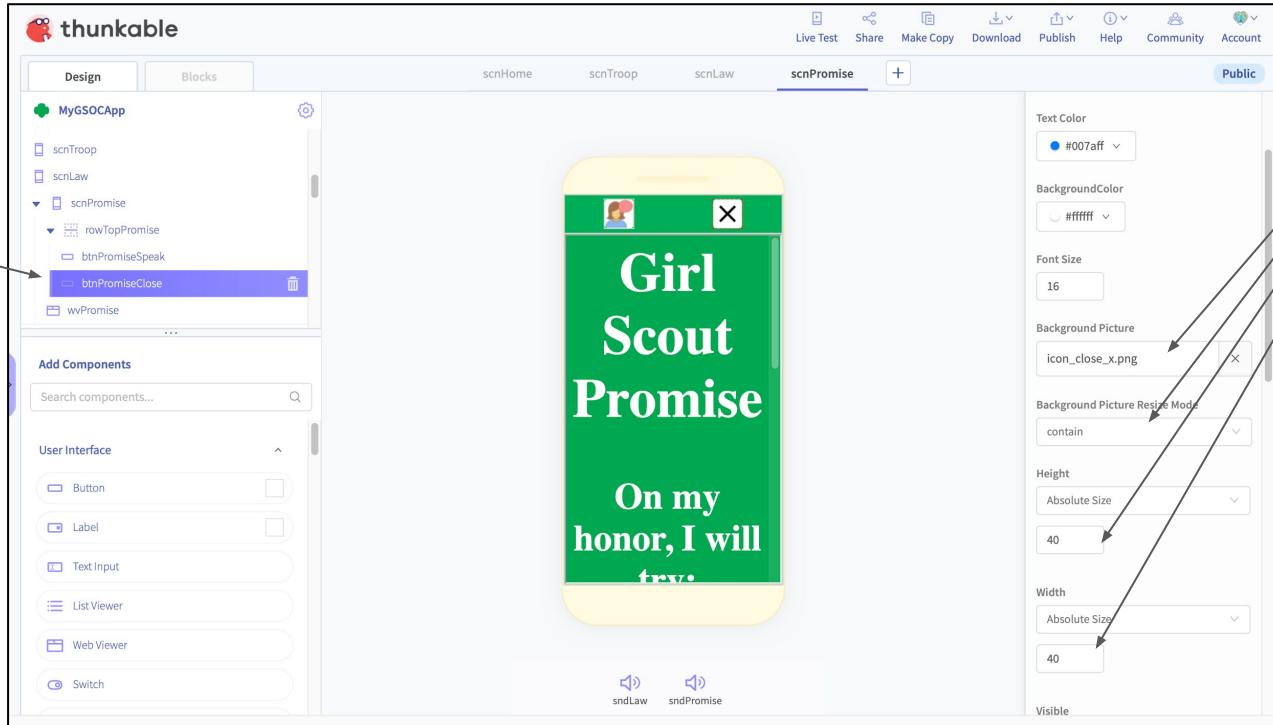
The screenshot shows the thunkable app interface with the following details:

- Left Sidebar:** Shows the project structure under "MyGSOCApp". The "scnPromise" screen is selected. Inside "scnPromise", there is a "rowTopPromise" component containing "Button2" and "Button1", and a "wvPromise" component.
- Center Preview:** Displays a mobile phone-like preview of the "scnPromise" screen. The title "Girl Scout Promise" is at the top, followed by the text "On my honor, I will trv.". At the bottom are two buttons labeled "sndLaw" and "sndPromise" with speaker icons.
- Right Panel (Properties):** Shows the properties for the "rowTopPromise" component.
  - Height:** Set to "10%" (highlighted with a red arrow).
  - Width:** Set to "Fit contents, Fill container".
  - BackgroundColor:** Set to "rgba(0, 128, 0)".
  - Horizontal Alignment:** Set to "space-around".
  - Vertical Alignment:** Set to "center".
- Callout Box:** A callout box on the right side contains the text: "Set the rowTopPromise container attributes as shown". It points to the "Height" and "Horizontal Alignment" settings in the properties panel.

# Guided Design - Promise Screen UI

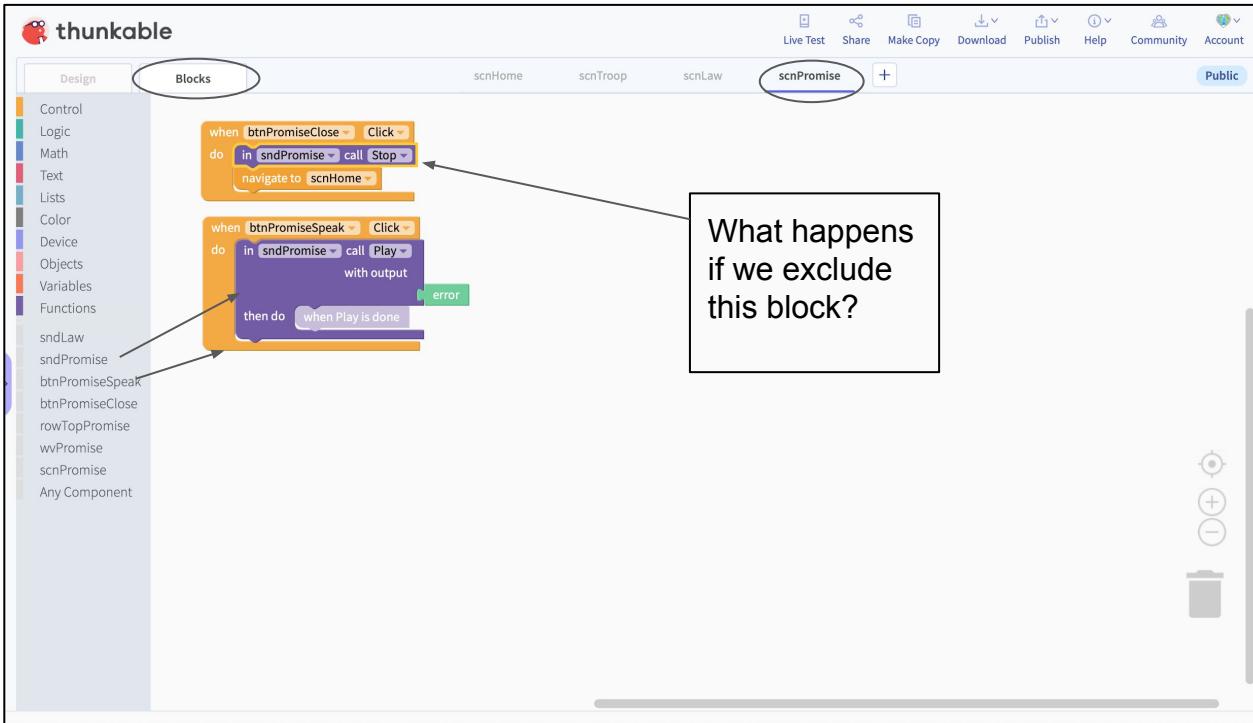
Name the two buttons based upon our conventions, and then add background pictures. Be sure to set the Background Picture parameter of each button to “contain”. A height and width of 40 looks good on the phone for the background pictures we have selected in this example.

Be sure to  
rename your  
buttons!



# Guided Design - Let's Code! (scnPromise)

Click on Blocks, then add “Click” event blocks for our Promise screen’s Speak and Close buttons. When we click the Close button, we return to the home screen. When we click the Speak button, we play our Promise sound file asset.



The screenshot shows the Thunkable app interface with the following details:

- Project Title:** thunkable
- Screen Selected:** scnPromise (highlighted with a blue oval)
- Blocks Tab:** Active tab (highlighted with a blue oval)
- Block Preview:** A callout box contains the question: "What happens if we exclude this block?" with an arrow pointing to the "when Play is done" block.
- Code Preview:**

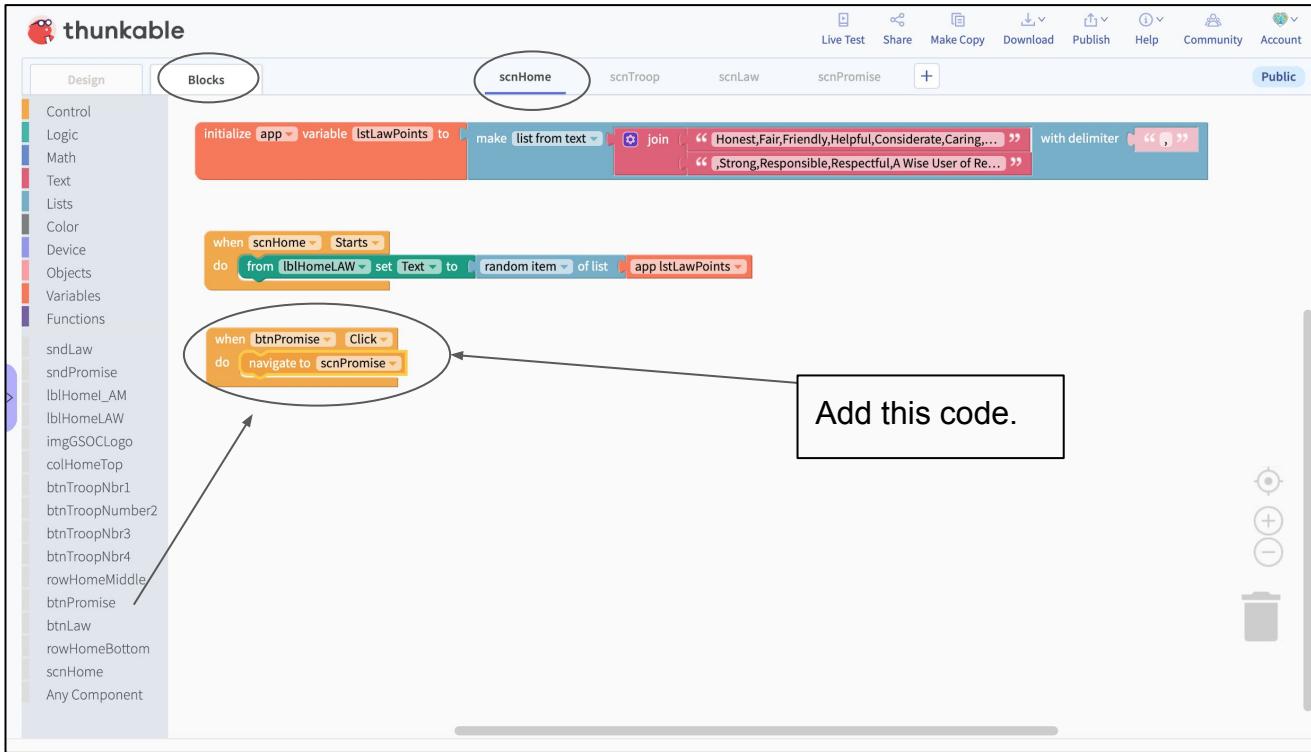
```

when btnPromiseClose Click
do in sndPromise call Stop
navigate to scnHome

when btnPromiseSpeak Click
do in sndPromise call Play
with output
then do when Play is done
  
```
- Block Palette:** On the left, under the "Control" category, the "when [button] Click" blocks for "btnPromiseClose" and "btnPromiseSpeak" are selected.
- Object List:** On the left, the objects "sndLaw" and "sndPromise" are listed.
- Toolbar:** Top right includes buttons for Live Test, Share, Make Copy, Download, Publish, Help, Community, and Account.

# Guided Design - Let's Code! (scnHome)

But how do we reach our new Promise screen? Click back on scnHome and then Blocks and add a Click event block for our btnPromise. Now we can click on bthPromise and navigate to scnPromise.



The screenshot shows the Thunkable app editor interface. The scene is named "scnHome". In the Blocks tab, there are three main blocks:

- An orange "Control" block: `initialize app variable lstLawPoints to make list from text [make list from text [join [“Honest,Fair,Friendly,Helpful,Considerate,Caring...”] with delimiter [“,”]]]`
- An orange "Control" block: `when scnHome Starts do [from lblHomeLaw set Text to [random item of list app lstLawPoints]]`
- A blue "Control" block: `when btnPromise Click do [navigate to scnPromise]`

A callout box with the text "Add this code." has an arrow pointing to the blue "Control" block.

# Guided Design - Repeat for scnLaw

---

## For scnLaw:

Slide 26: Add the Girl Scout Background Color (R = 0, G =174, B = 88)

Slide 27: Add a row and web viewer component, and set the web viewer's URL parameter to the asset file **GSLaw.html**.

Slide 28: Change the row container's height to 10% of our screen space, and set the horizontal alignment of the row container to “space around”.

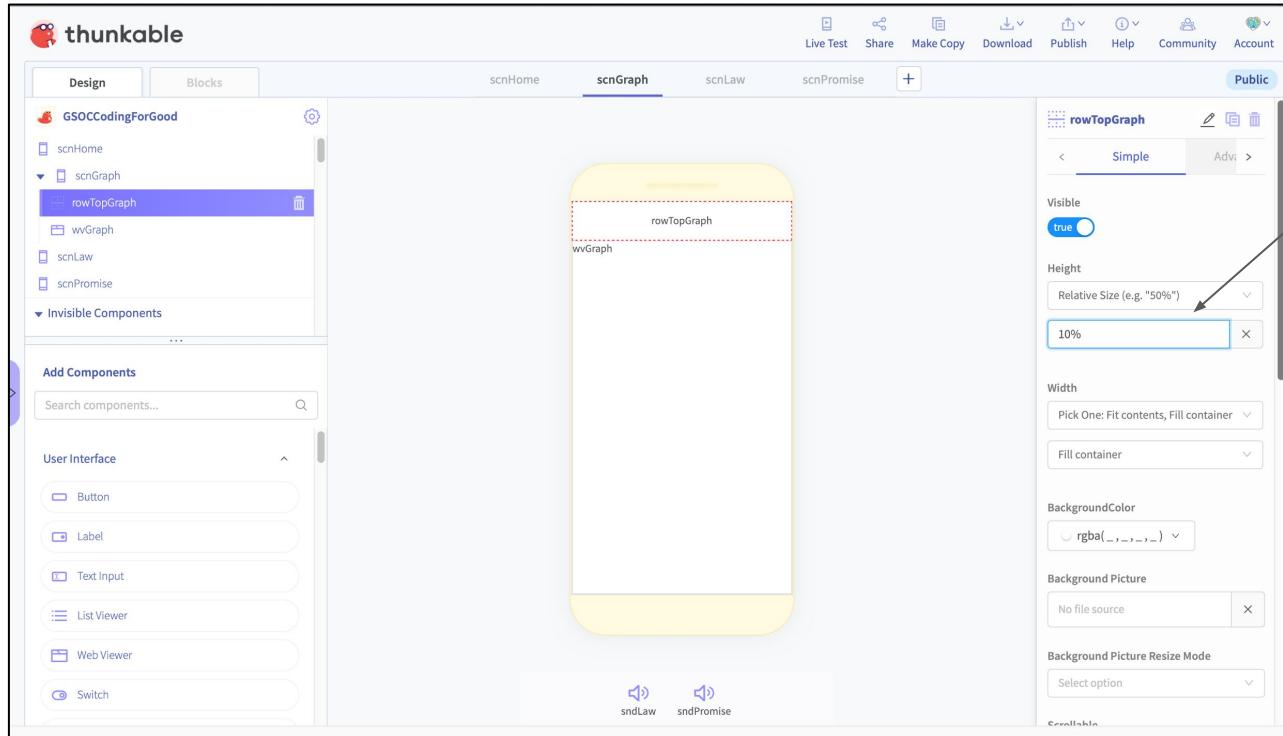
Slide 29: Add images to the row container's buttons and set their height / width to 40.

Slide 30: Add code to scnLaw, causing btnLawClose to navigate to scnHome and btnLawSound to play the sndLaw sound file asset.

Slide 31: Add code to scnHome, causing btnLaw to navigate to scnLaw.

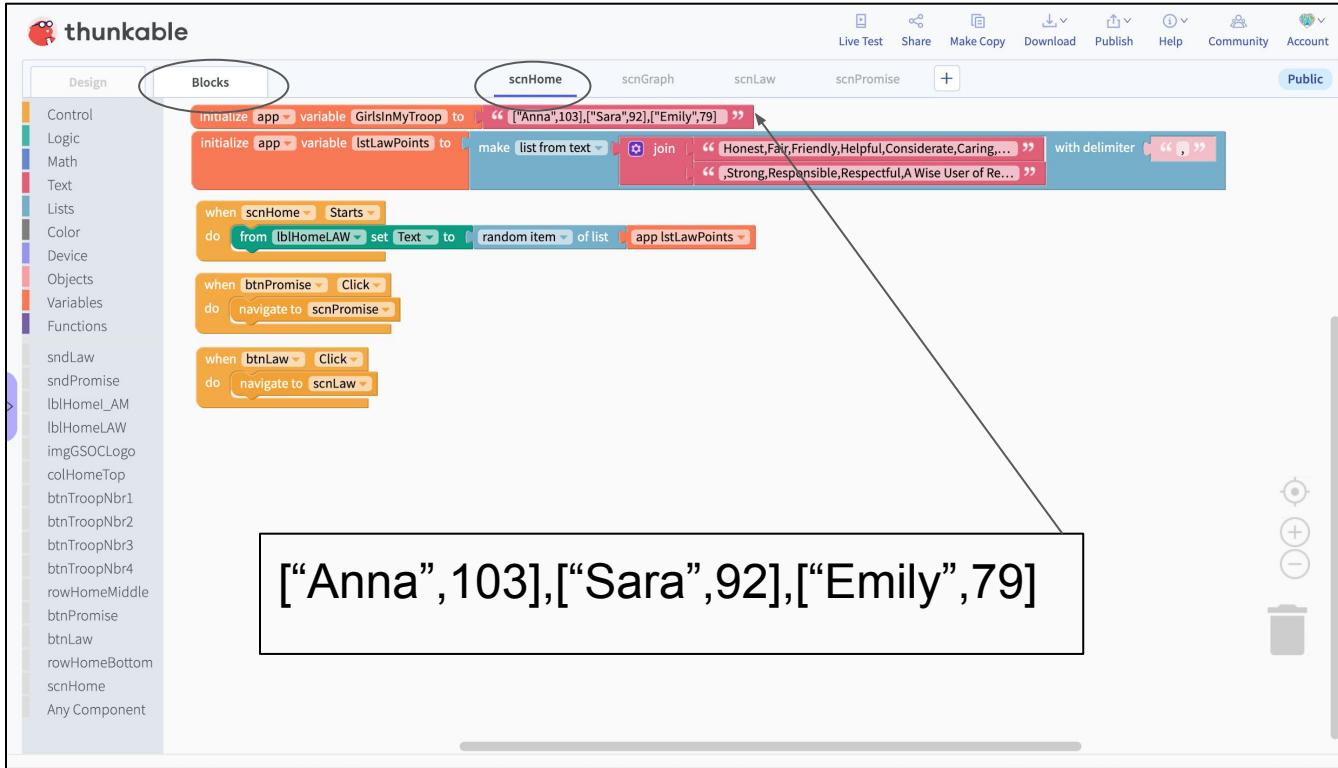
# Guided Design - Graph your cookie sales!

Select existing scnGraph, then add a row container (top) and Web Viewer component (bottom) to scnGraph, renaming the components using our conventions. Set the row container's relative height to 10%.



# Guided Design - Graph your cookie sales!

Select scnHome and then Blocks to edit our code. The existing variable GirlsInMyTroop represents an ARRAY of the girls in your troop and their cookie sales. Let's edit that array with your troop's data!



The screenshot shows the thunkable app interface with the following details:

- Blocks Tab:** Selected tab.
- Variables:**
  - scnHome: An array variable containing the data: `[{"Anna":103}, {"Sara":92}, {"Emily":79}]`
  - scnGraph
  - scnLaw
  - scnPromise
- Code Preview:** A large text box displays the JSON array: `[“Anna”,103],[“Sara”,92],[“Emily”,79]`.
- Code Editor:** Shows the following Scratch-style script blocks:
  - Initialize app variable GirlsInMyTroop to `[{"Anna":103}, {"Sara":92}, {"Emily":79}]`
  - Initialize app variable lstLawPoints to make list from text `join “Honest,Fair,Friendly,Helpful,Considerate,Caring,...” “Strong,Responsible,Respectful,A Wise User of Re...”` with delimiter `,`
  - when scnHome Starts
    - do from lblHomeLAW set Text to random item of list app lstLawPoints
  - when btnPromise Click
    - do navigate to scnPromise
  - when btnLaw Click
    - do navigate to scnLaw

# So what is an ARRAY?

Developers use many different **data structures** as variables when developing their applications, and **ARRAYS** are one of the most common data structures.

Simply put, an array is a list. In our guided design's scnHome blocks, we have an array of the Girl Scout Law points and an array of three girls in a troop along with their cookie sales for the year.

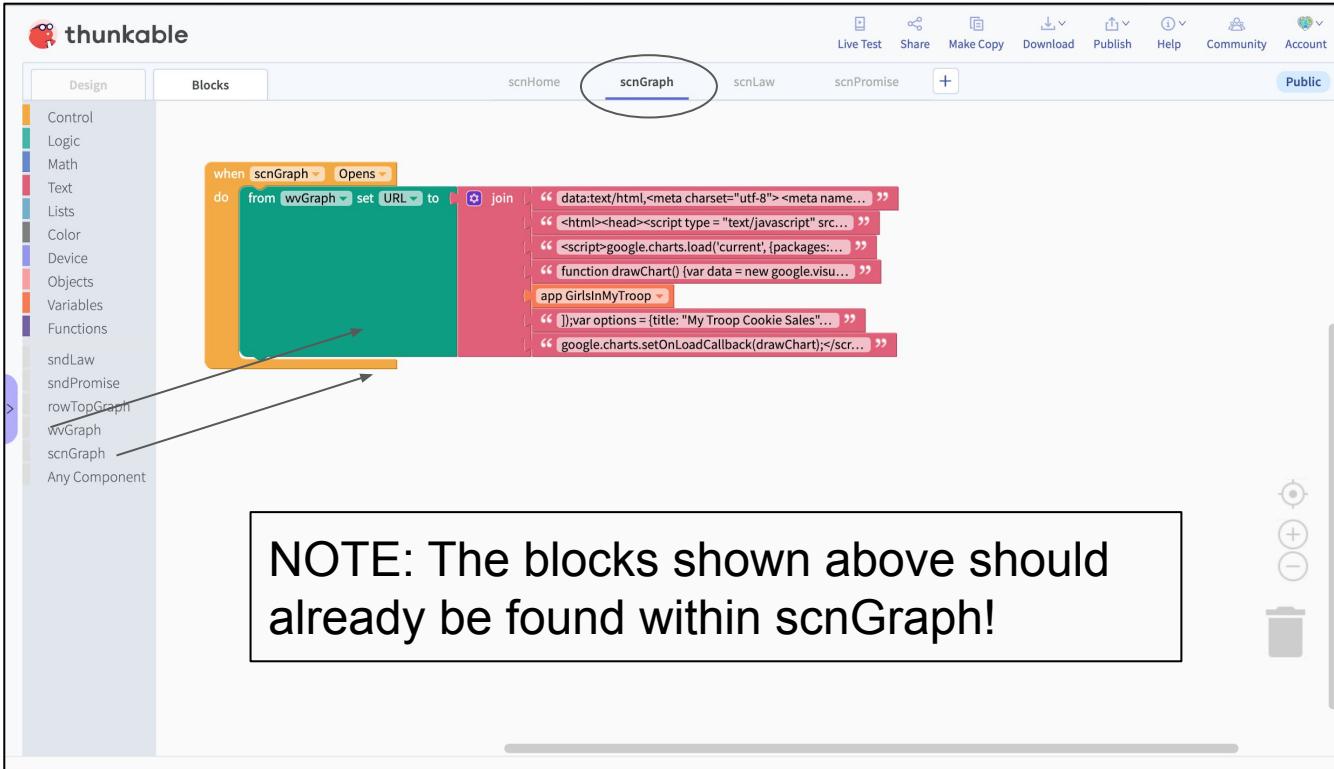
Each array **element** is separated by a comma, so ["Anna",103] and ["Sara",92] are separate elements.

In this example, each of our array elements is also an array! "Anna" is the name of a girl in our troop and 103 is Anna's cookie sales. We use square braces to delimit each individual element within our array variable that we are calling **GirlsInMyTroop**. If we wanted to reference an array element in our code, we would use the syntax GirlsInMyTroop[index number]. Most computer languages start counting array elements at index number 0, so GirlsInMyTroop[0] would return the array ["Anna",103].

**Now edit the GirlsInMyTroop variable and create an array like this one reflective of the girls in your own troop. Just make up sales numbers for each girl if you don't know them!**

# Guided Design - Graph your cookie sales!

Now select scnGraph. Add the wvGraph block (set URL) as shown, adding in the existing “join” block. Then add a scnGraph Opens block and place the wvGraph block inside.



NOTE: The blocks shown above should already be found within scnGraph!

The screenshot shows the thunkable app interface. The top navigation bar includes 'Live Test', 'Share', 'Make Copy', 'Download', 'Publish', 'Help', 'Community', and 'Account'. The main workspace shows a script with the following blocks:

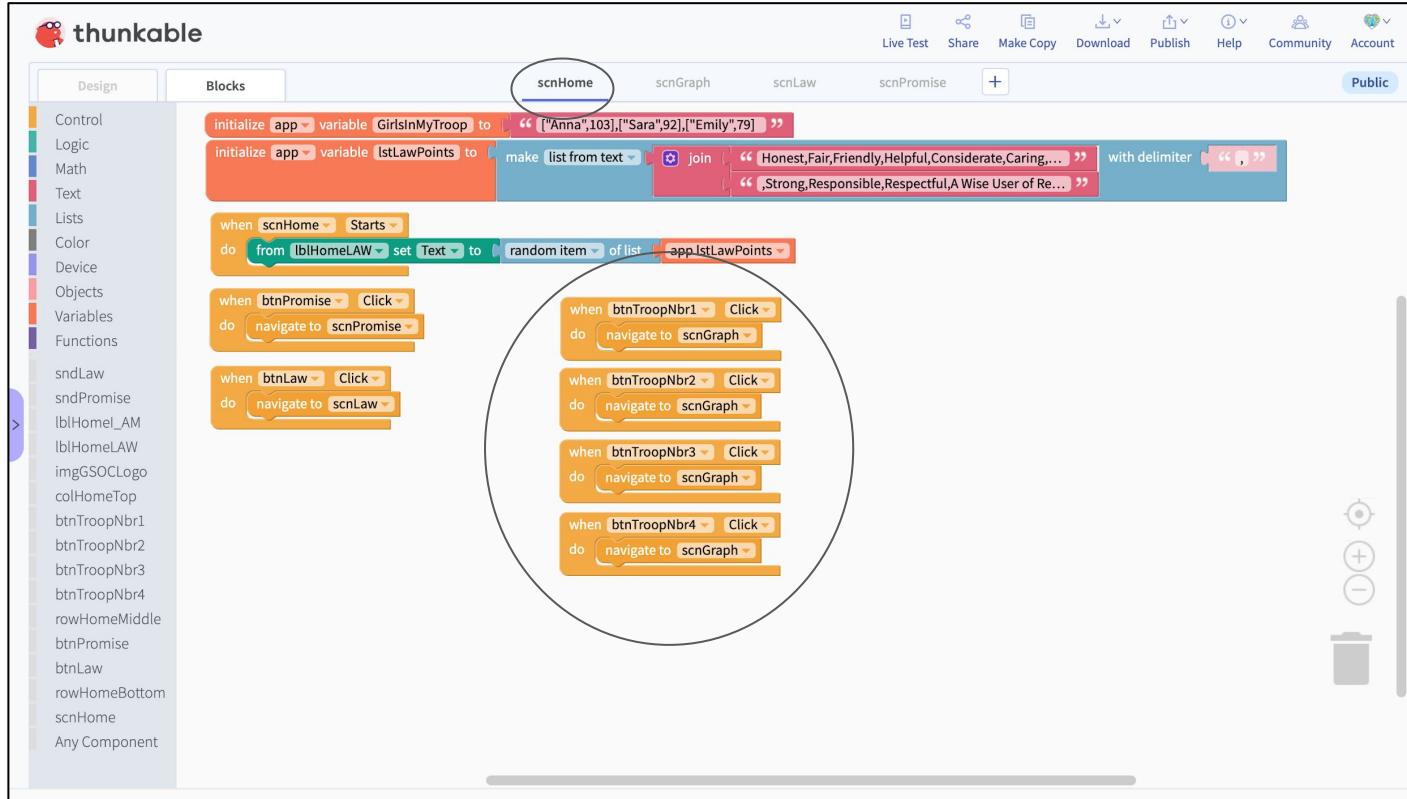
- A green 'when scnGraph Opens' hat block.
- An orange 'do' block containing:
  - 'from wvGraph set URL to' block with a red arrow pointing to it from the note below.
  - 'join' block with several lines of JavaScript code:

```
data:text/html,<meta charset="utf-8"> <meta name...>
<html><head><script type = "text/javascript" src...>
<script>google.charts.load('current', {packages:...});
function drawChart() {var data = new google.visu...
app GirlsInMyTroop
});var options = {title: "My Troop Cookie Sales"};
google.visualization.setOnLoadCallback(drawChart);</scr...
```

The left sidebar lists components: Control, Logic, Math, Text, Lists, Color, Device, Objects, Variables, Functions, sndLaw, sndPromise, rowTopGraph, wvGraph, scnGraph, and Any Component. The 'scnGraph' component is selected, highlighted with a blue underline.

# Guided Design - Graph your cookie sales!

We need a way to open scnGraph! Return to scnHome and add a block for each of your troop number buttons as shown below. Clicking on any troop number will open scnGraph.



The screenshot shows the Thunkable app editor interface. The top navigation bar includes 'Live Test', 'Share', 'Make Copy', 'Download', 'Publish', 'Help', 'Community', and 'Account'. The left sidebar lists various components: Control, Logic, Math, Text, Lists, Color, Device, Objects, Variables, Functions, sndLaw, sndPromise, lblHome\_AM, lblHomeLaw, imgGSOCLogo, colHomeTop, btnTroopNbr1, btnTroopNbr2, btnTroopNbr3, btnTroopNbr4, rowHomeMiddle, btnPromise, btnLaw, rowHomeBottom, scnHome, and Any Component. The main workspace displays the following code:

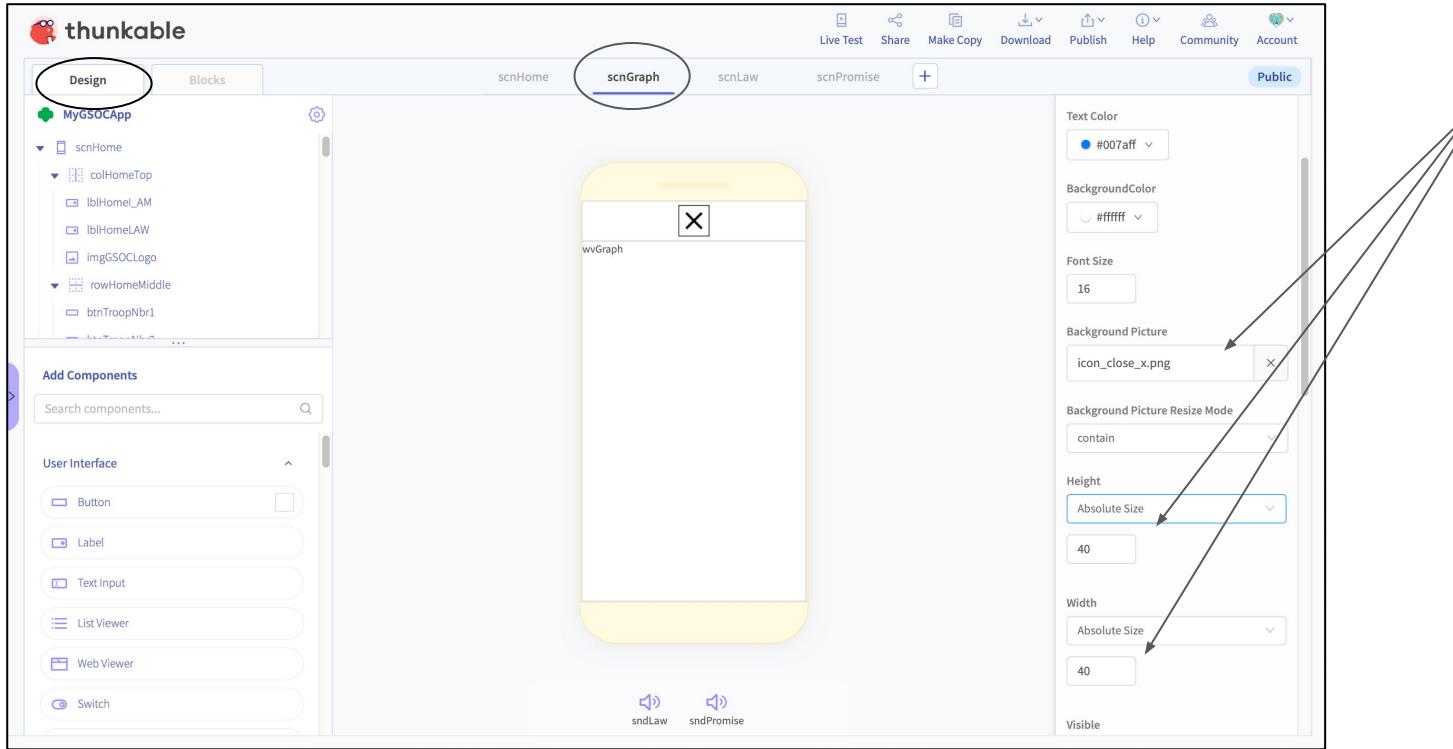
```

scnHome
  initialize app variable GirlsInMyTroop to ["Anna",103],["Sara",92],["Emily",79]
  initialize app variable lstLawPoints to make list from text join "Honest,Fair,Friendly,Helpful,Considerate,Caring,... with delimiter ","
  when scnHome Starts
    do from lblHomeLaw set Text to random item of list app lstLawPoints
  when btnPromise Click
    do navigate to scnPromise
  when btnLaw Click
    do navigate to scnLaw
  when btnTroopNbr1 Click
    do navigate to scnGraph
  when btnTroopNbr2 Click
    do navigate to scnGraph
  when btnTroopNbr3 Click
    do navigate to scnGraph
  when btnTroopNbr4 Click
    do navigate to scnGraph
  
```

A large circle highlights the four 'when btnTroopNbr Click' blocks.

# Guided Design - Graph your cookie sales!

We'll also need to return to scnHome from scnGraph. Let's add a close button to our design. Click on Design and then scnGraph, adding a button into the row container. Let's use the same graphic and sizes as our scnLaw and scnPromise close buttons. Here I've also added a border to the button!

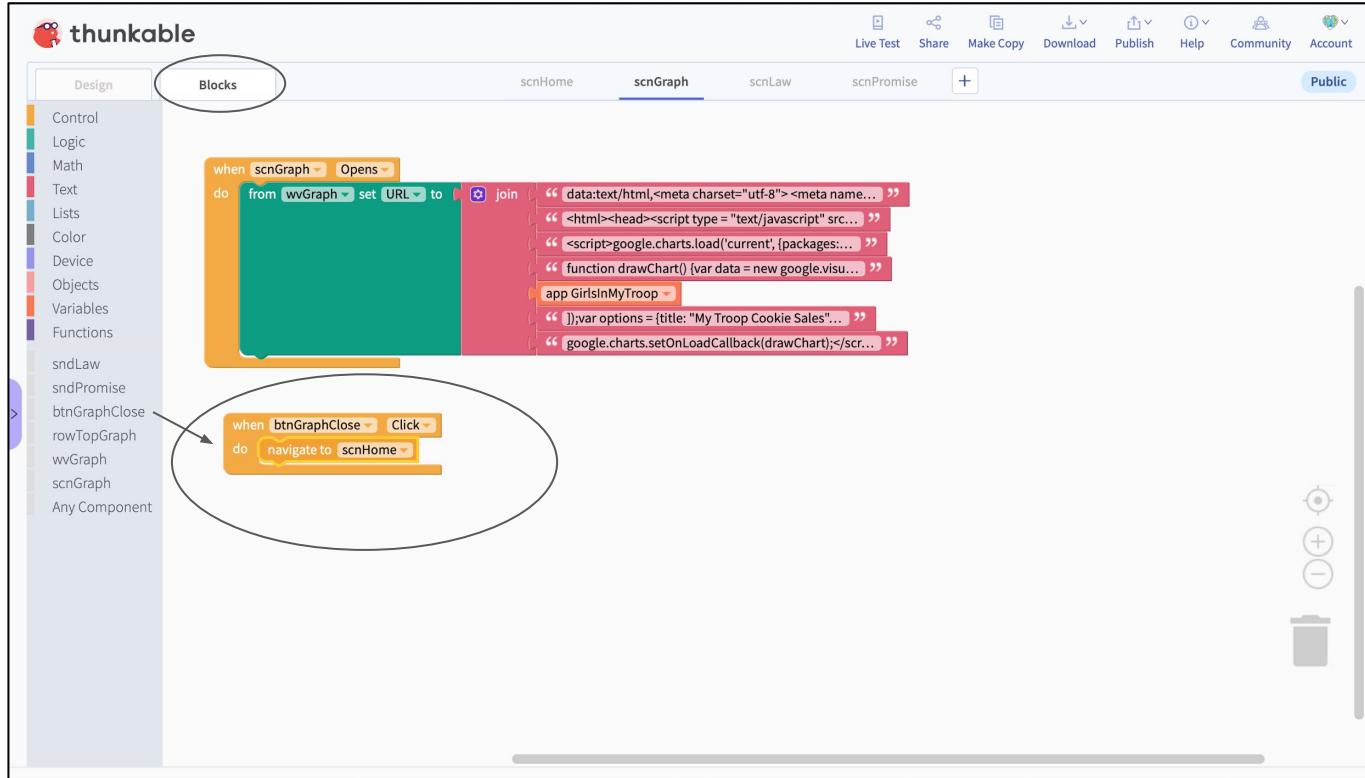


The screenshot shows the Thunkable app interface with the following details:

- Top Bar:** Includes "Live Test", "Share", "Make Copy", "Download", "Publish", "Help", "Community", and "Account". A "Public" button is also present.
- Left Sidebar:** Shows the project structure under "MyGSOCApp". The "scnHome" screen is selected, displaying its components: "colHomeTop", "lblHome\_AM", "lblHome\_AW", "imgGSOCLogo", "rowHomeMiddle", and "btnTroopnbr1".
- Design View:** The "scnGraph" screen is currently active. It features a yellow smartphone-shaped viewfinder labeled "vvGraph". Inside the viewfinder, there is a white rectangular area with a black "X" icon in the top right corner, representing a close button.
- Properties Panel (on the right):** Provides settings for the selected component:
  - Text Color:** #007aff
  - Background Color:** #ffffff
  - Font Size:** 16
  - Background Picture:** icon\_close\_x.png
  - Background Picture Resize Mode:** contain
  - Height:** Absolute Size, set to 40
  - Width:** Absolute Size, set to 40
  - Visible:** checked
- Bottom Buttons:** "sndLaw" and "sndPromise" are visible at the bottom of the screen.

# Guided Design - Graph your cookie sales!

Click on Blocks to add the code for our new close button (did you name yours btnGraphClose?). Add a Click block that will navigate back to scnHome.



The screenshot shows the thunkable app interface with the following details:

- Blocks Tab:** The "Blocks" tab is selected in the top navigation bar.
- Scenes:** The scenes are listed as tabs: scnHome, **scnGraph**, scnLaw, and scnPromise.
- Scratch-like Script:**
  - Top Script:** "when scnGraph Opens" - "do [from wvGraph set URL to [join "data:text/html,<meta charset="utf-8"><meta name..." "]]" - "script type = "text/javascript" src="]" - "google.charts.load('current', {packages:[...]};" - "function drawChart() {var data = new google.visu..." - "app GirlsInMyTroop" - "var options = {title: "My Troop Cookie Sales", width: 600, height: 400};" - "google.charts.setOnLoadCallback(drawChart);<scr..."
  - Bottom Script:** "when btnGraphClose Click" - "do [navigate to scnHome]"
- Left Sidebar:** A sidebar lists components: Control, Logic, Math, Text, Lists, Color, Device, Objects, Variables, Functions, sndLaw, sndPromise, btnGraphClose, rowTopGraph, wvGraph, scnGraph, Any Component.
- Top Bar:** Includes "Live Test", "Share", "Make Copy", "Download", "Publish", "Help", "Community", and "Account".
- Right Side:** Includes a zoom-in/out icon, a plus sign, and a trash bin.

# Guided Design - viewing on our phone

Open the Thunkable Live app on your phone, then click the Live Test button at the top of the Thunkable X development environment. You should be able to navigate through your screens now!



# Challenges for extending your app ...

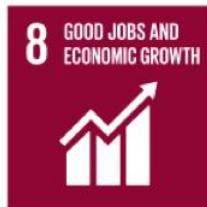
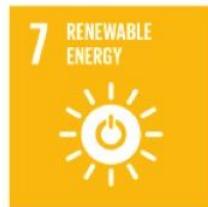
---

- Add a note-taking screen to capture ideas for your Bronze, Silver or Gold Award projects.
- Add a timer component and a label to display a countdown timer within a new screen. Help your troop leader keep your meetings running on time!
- Create a “photos” screen and add a camera and photo library component to capture photos taken at Girl Scout Events.
- Create a “compass” screen and use the magnetometer component and a graphical asset (picture of a compass) from [google.com/images](http://google.com/images) to create a compass that you can use on hikes.

Find more cool assets! [google.com/images](http://google.com/images) and [material.io](http://material.io) websites are good sources of pictures you could use for button backgrounds.

# Brainstorming App ideas

Let's discuss app ideas that could address a problem in society, in our neighborhood, or in our school. Here are some "global goals" to consider to help you get started!



# Technovation Girls and the Congressional App Challenges

***Did you enjoy today's event? 🙌🙌🙌***

Girl Scouts of Orange County would like to support teams to enter the Technovation Girls (Feb/Mar 2020) and the very similar Congressional App (Oct/Nov 2020) app development challenges!

Here are links to these events:

<https://technovationchallenge.org>

<https://www.congressionalappchallenge.us>

You must be **a junior or senior high school student** in order to enter these two competitions. Since the event requirements are very similar, you can enter the identical project in both challenges. There isn't an entry fee, and you'll have a chance to travel and win prizes for your team if your app is selected as a finalist!

If you're not in junior high school just yet, please enter the "Coolest Projects" event in March 2020: <https://coolestprojects.org/usa/> with your super cool app idea!

**The GSOC can help you connect with mentors that can assist your team!**

# What to do next...

---

## ***Continue to develop your app at home!***

Now that you know how to log onto ThunkableX and develop an app, start building some cool apps! Be sure to take this slide deck home with you as a reference.

You can start by adding onto the app you created today, or you can create a complete new app!

If you'd like to investigate other app development tools using text based programming (rather than block coding), I'd recommend that you look into these two development environments:

Swift: <https://developer.apple.com/swift/>

Flutter / Dart: <https://flutter.dev>

If you'd like to consider entering the Technovation or Congressional App Challenges, use these two search terms within YouTube to review prior year's entries: "congressional app challenge" "technovation challenge"

**Congratulations on earning your Coding for Good - App Development Badge!**