CGI

Thunkable Mobile App – Virtual STEM Camp Course

STEM@CGI AT HOME

What you need

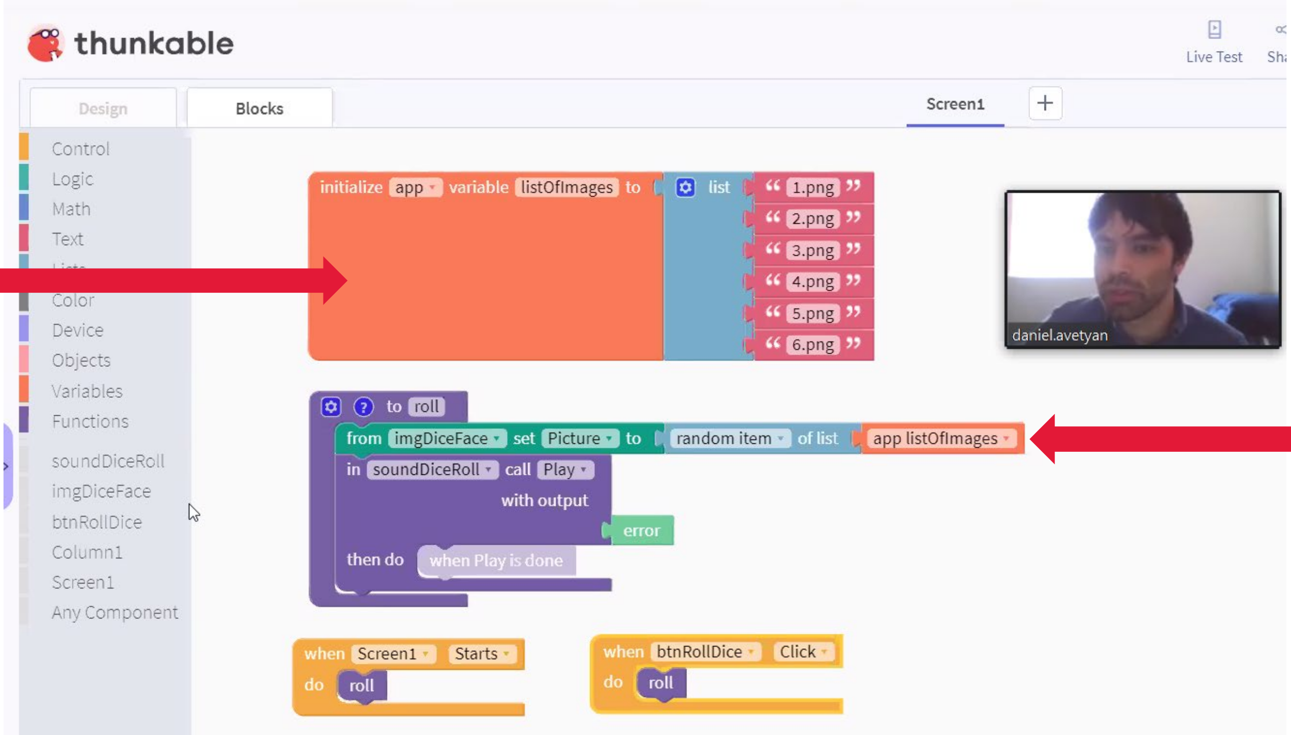
- **Time Needed**
  - A total of 90 minutes is recommended for this course

- **Materials needed**
  - 1 computer with internet access
  - Optional
    - 1 smartphone for downloading the mobile app
    - Second monitor
    - Printed out copy of the lesson

Intro to coding concepts

Variables

A variable allows you to store data in the form of a **string** of text, a **number**, a true/false **boolean**, or a collection of data (e.g. a list of names and addresses like a phonebook).



```
initialize app variable listOfImages to  list  "1.png"
                                               "2.png"
                                               "3.png"
                                               "4.png"
                                               "5.png"
                                               "6.png"

to roll
    from imgDiceFace set Picture to  random item of list  app listOfImages
    in soundDiceRoll call Play
                         with output
                                     error
    then do  when Play is done

when Screen1 Starts
do  roll

when btnRollDice Click
do  roll
```

Variable being created

Variable being used

Basic flow control (if statement, while, for, etc.)

- **If statements** – An “if statement” allows you to make a choice in what executes based off of logic that’s passed into the if statement. (.e.g. *If* your fridge has food, *then* you will make dinner, *else* you’ll go to the market)

- **For statements** – A “for statement” allows you to iterate through a list of items such as going through the phone book example mentioned above contact by contact.

- **While statements** – A “while statement” will continue to execute until a condition is met, these are useful when you want to have the same piece of logic executed multiple times for a set amount. (e.g. continue to send a message until a counter has reached a certain number)

- **Syntax** – Different languages (e.g. HTML, CSS, Python, Java, C#) are all different forms of writing instructions.

Various platforms

What is a platform? A platform is the environment a program is run on. Some examples of platforms are operating systems, web browsers, or a program called Thunkable!

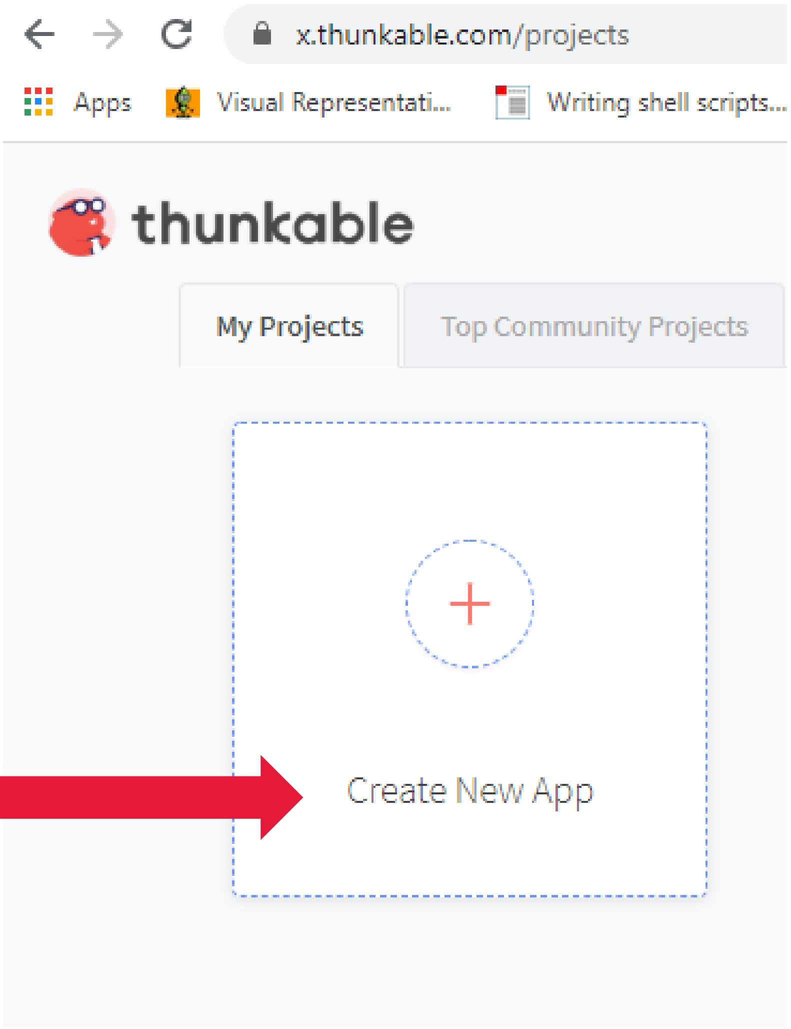
Platforms today like Salesforce are more user friendly and do not always require code. Thunkable is another example of a platform that does not necessarily require coding.

Thunkable is a “drag and drop” programming platform developed at MIT to help students focus on the structure of coding without having to get into other details.

How-To for the Thunkable platform

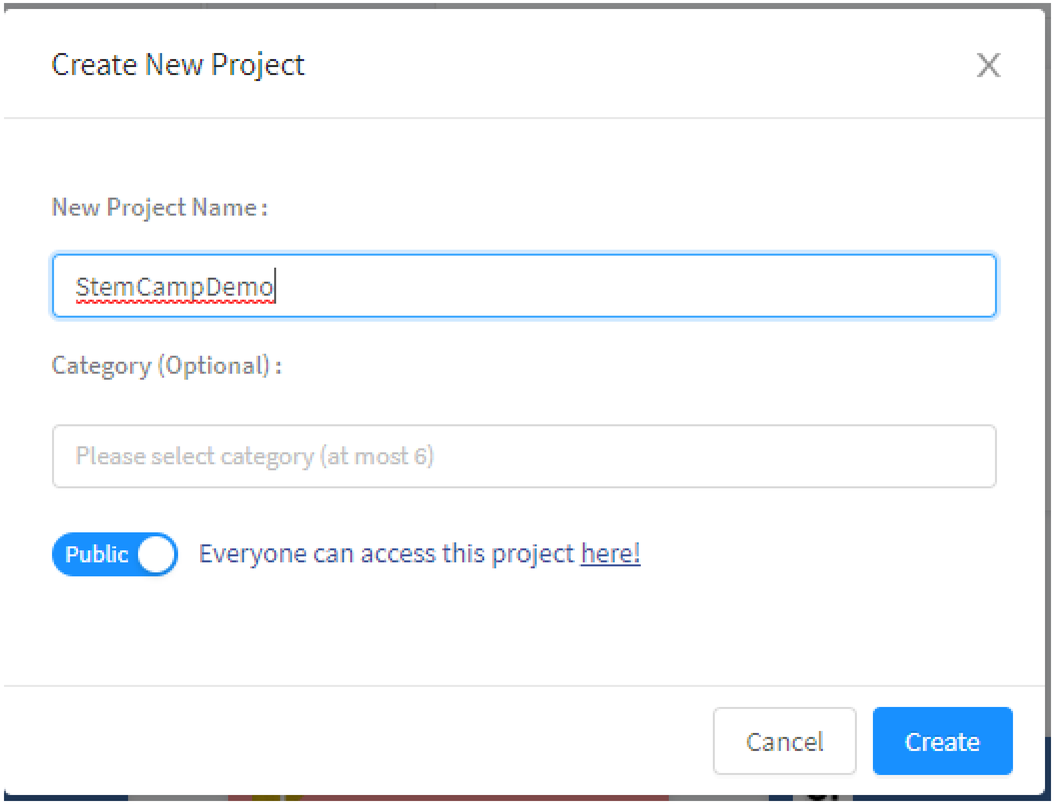
How-To for the Thunkable platform – Part 1

Navigate to the Thunkable projects page and click *Create New App*. (Additionally now is a good time to go download the Thunkable Live App so you can play around with the App we're making in the How-To in real time.)



How-To for the Thunkable platform – Part 2

Name your new project something unique and click the *Create* button:



Create New Project

New Project Name:

StemCampDemo

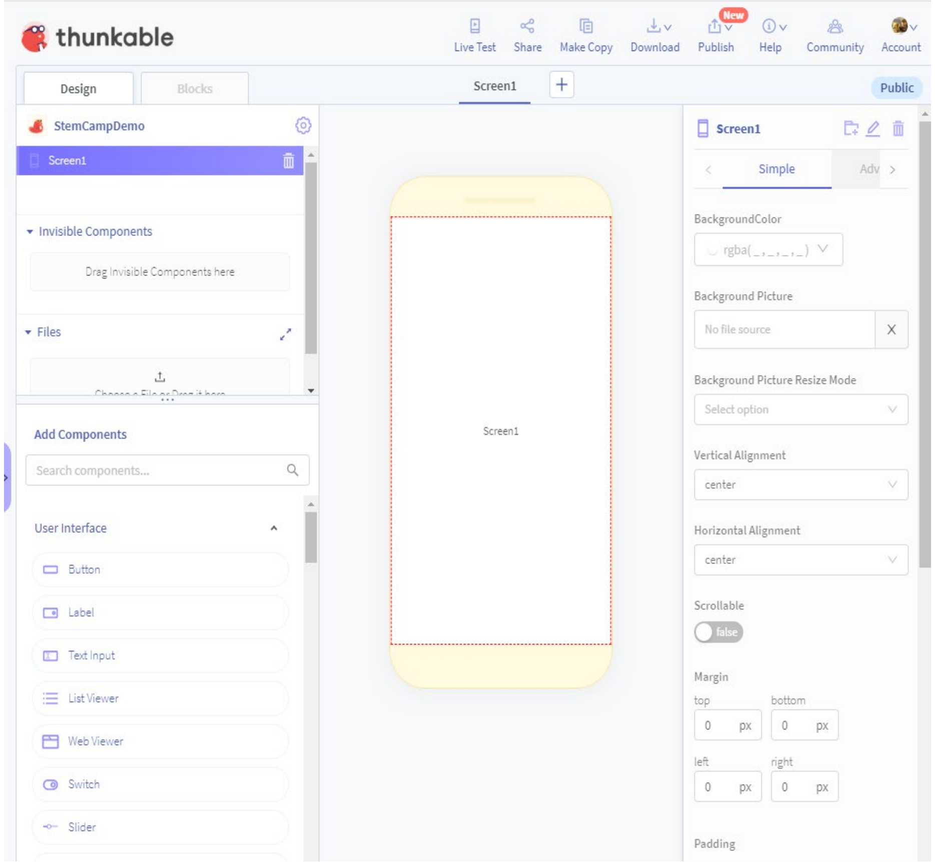
Category (Optional):

Please select category (at most 6)

Public Everyone can access this project here!

How-To for the Thunkable platform – Part 3

You should be greeted with this page:



BackgroundColor: rgba(_,_,_,_)

Background Picture: No file source X

Background Picture Resize Mode: Select option

Vertical Alignment: center

Horizontal Alignment: center

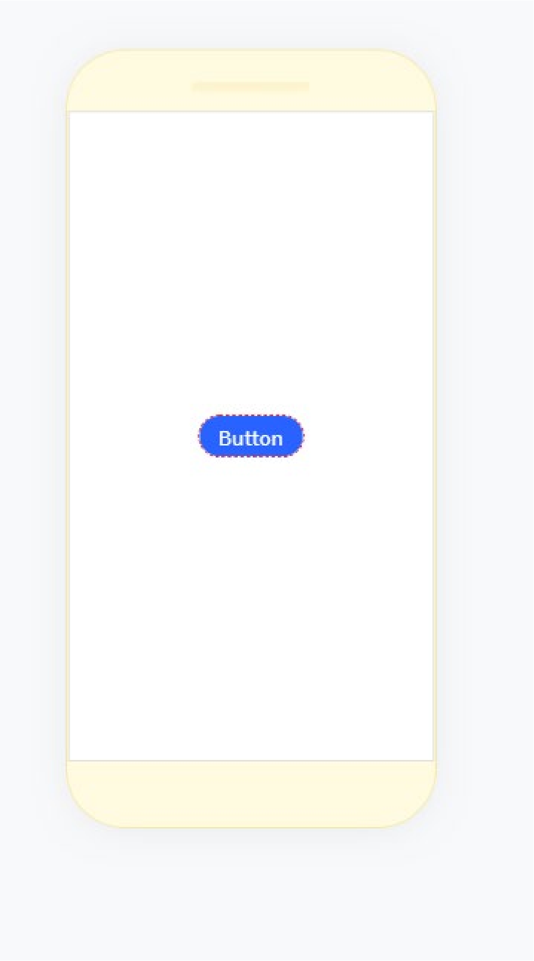
Scrollable: false

Margin
top 0 px
bottom 0 px
left 0 px
right 0 px

Padding

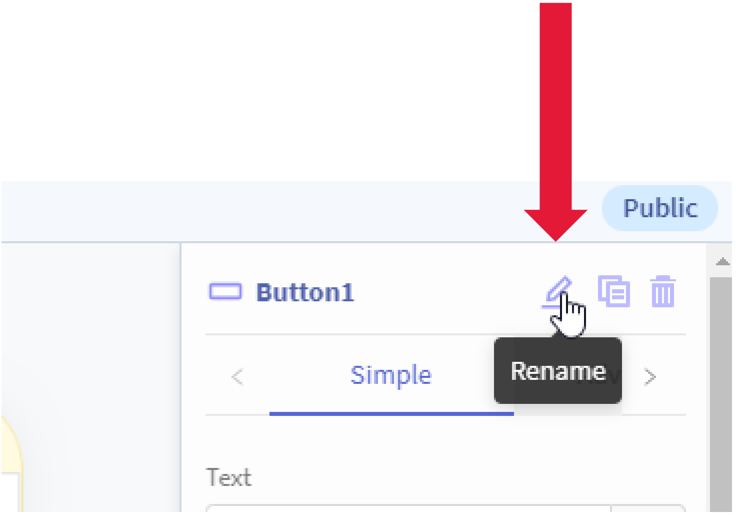
How-To for the Thunkable platform – Part 4

Next go to the *Add Components* panel on the bottom left and select the Button option under user interface and drag it to the screen like so:



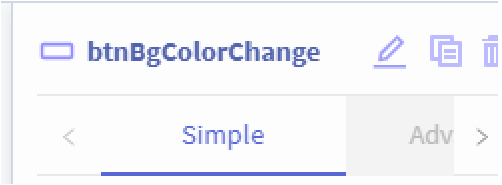
How-To for the Thunkable platform – Part 5

You don’t want to leave your button as a generic name, so rename it something descriptive by going to the top of the screen and clicking the pencil icon:



How-To for the Thunkable platform – Part 6

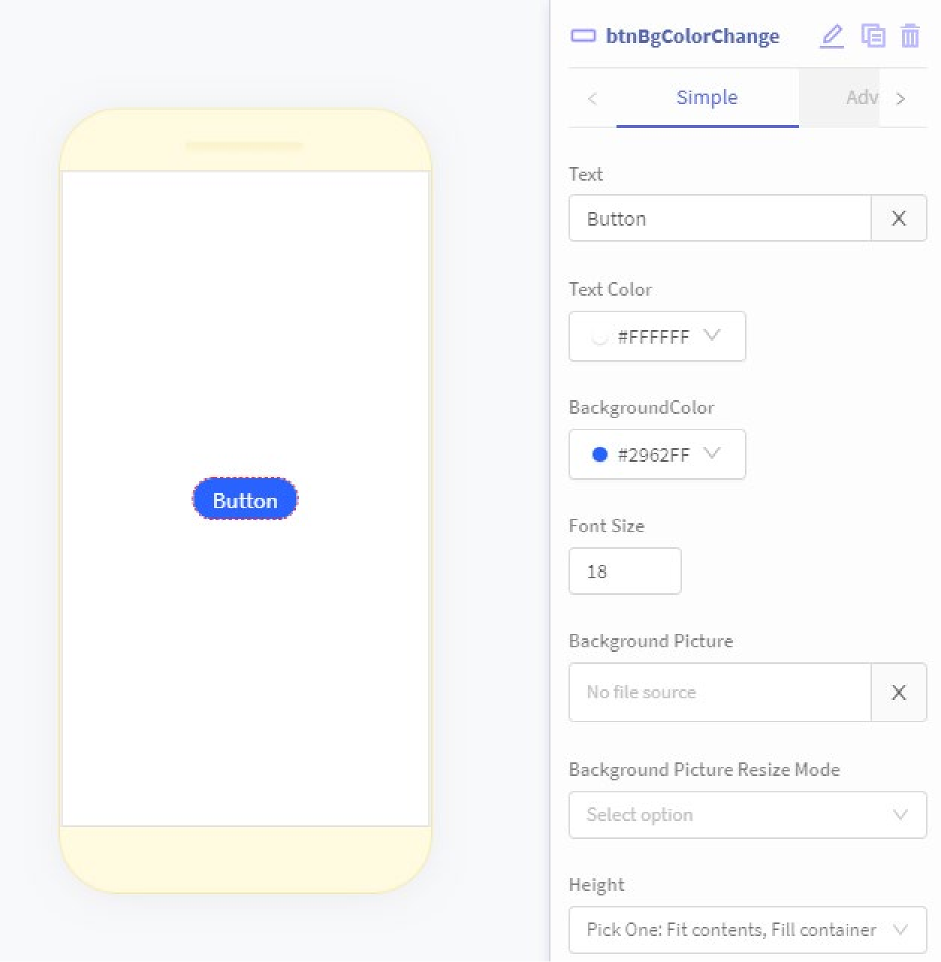
For this demo I’m going to name it “btnBgColorChange”, “btn” to denote that it’s a button, “BgColor” to denote that Background Color and Change to specify that this button will change the background color:



btnBgColorChange

How-To for the Thunkable platform – Part 7

We’re also going to change the button color and text from the defaults, I’m going to change the text to “Click Me!” and the color to **Black**.



btnBgColorChange

Text: Button

Text Color: #FFFFFF

BackgroundColor: ● #2962FF

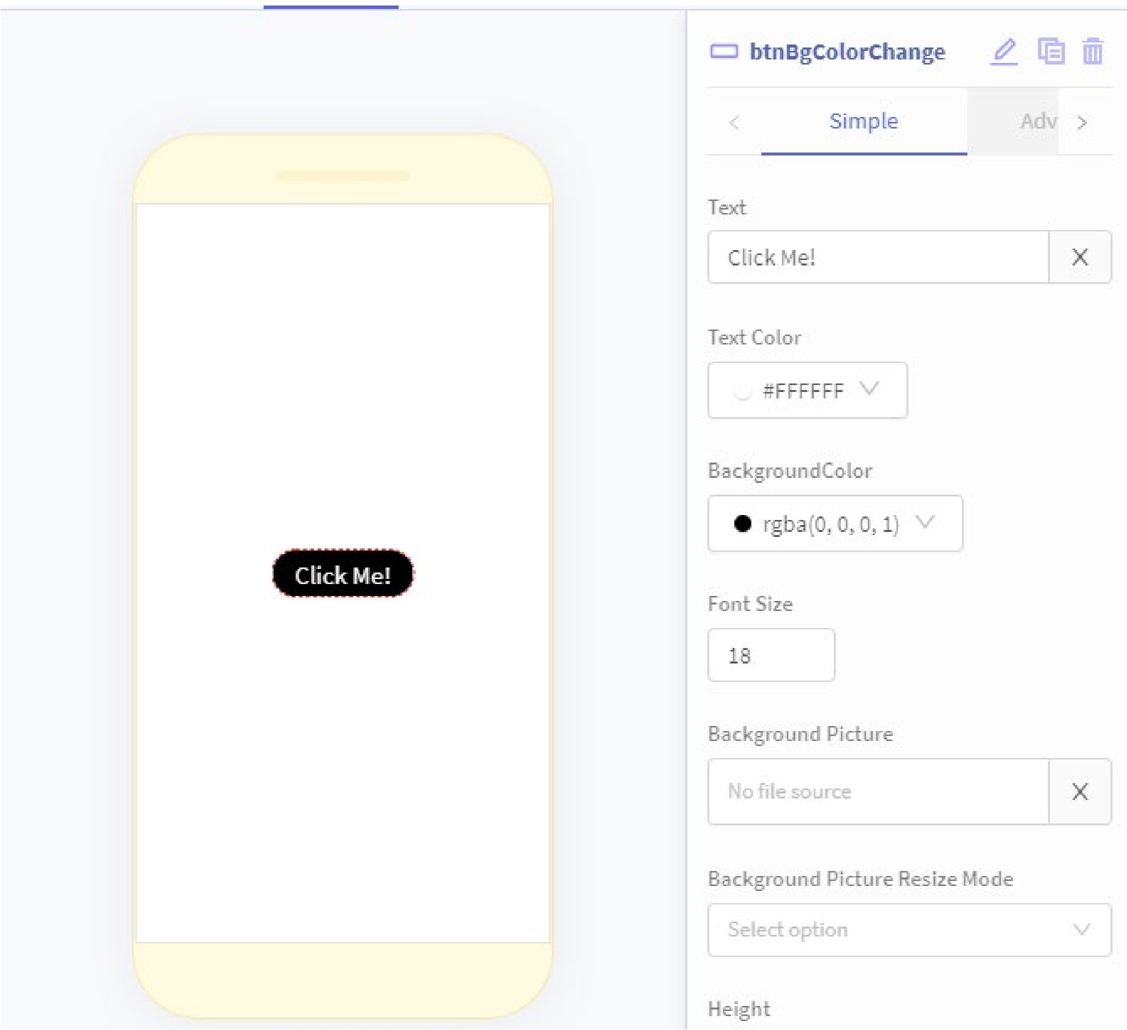
Font Size: 18

Background Picture: No file source

Background Picture Resize Mode: Select option

Height: Pick One: Fit contents, Fill container

How-To for the Thunkable platform – Part 7 (cont)



btnBgColorChange

Text: Click Me!

Text Color: #FFFFFF

BackgroundColor: ● rgba(0, 0, 0, 1)

Font Size: 18

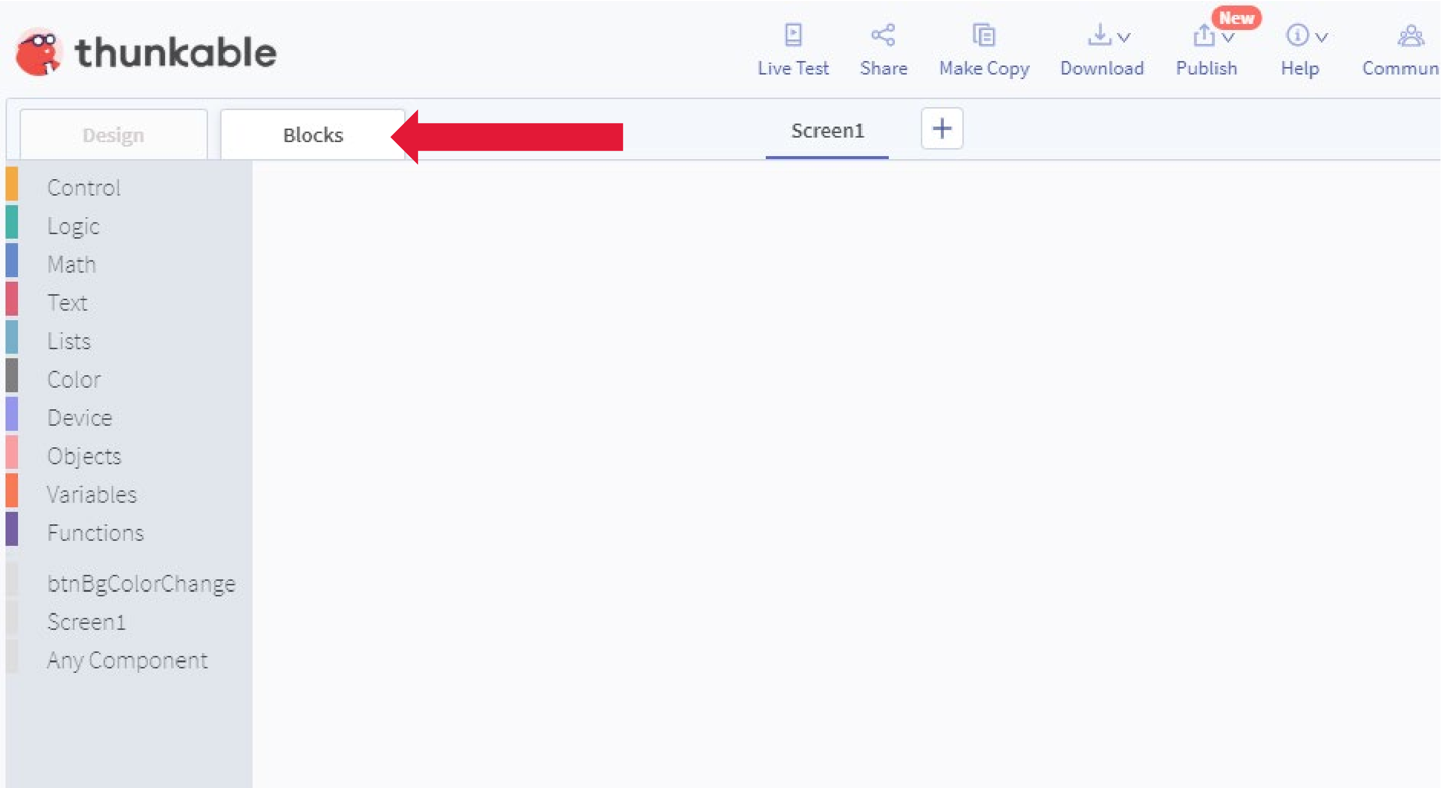
Background Picture: No file source

Background Picture Resize Mode: Select option

Height

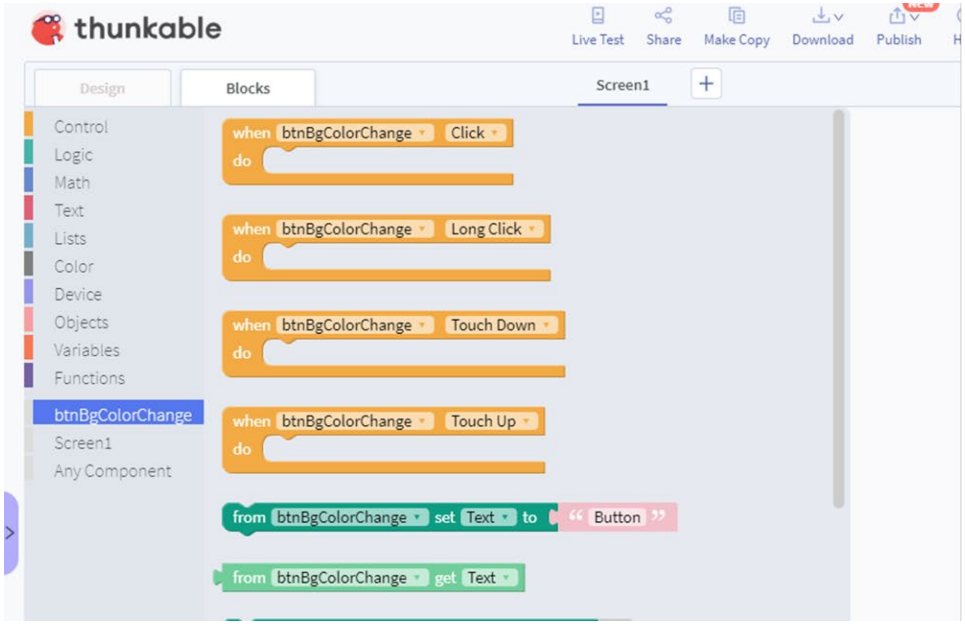
How-To for the Thunkable platform – Part 8

Next we're going to go to the top left of the screen under the Thunkable logo and click on *Blocks*

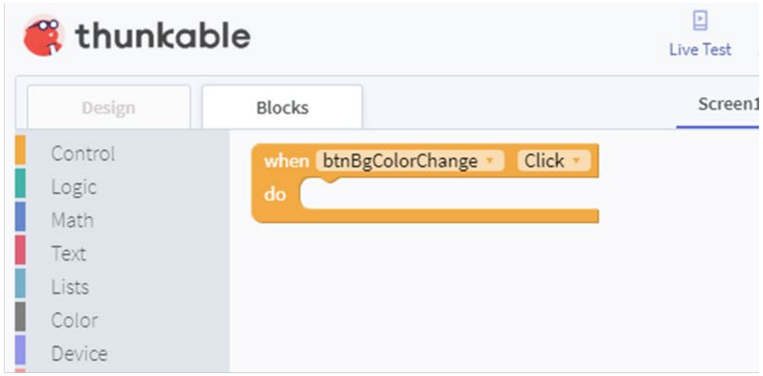


How-To for the Thunkable platform – Part 9

Now we’re going to select the button that was created and find the when “btnBgColorChange” Click (it should be the top option), select this control by clicking on it



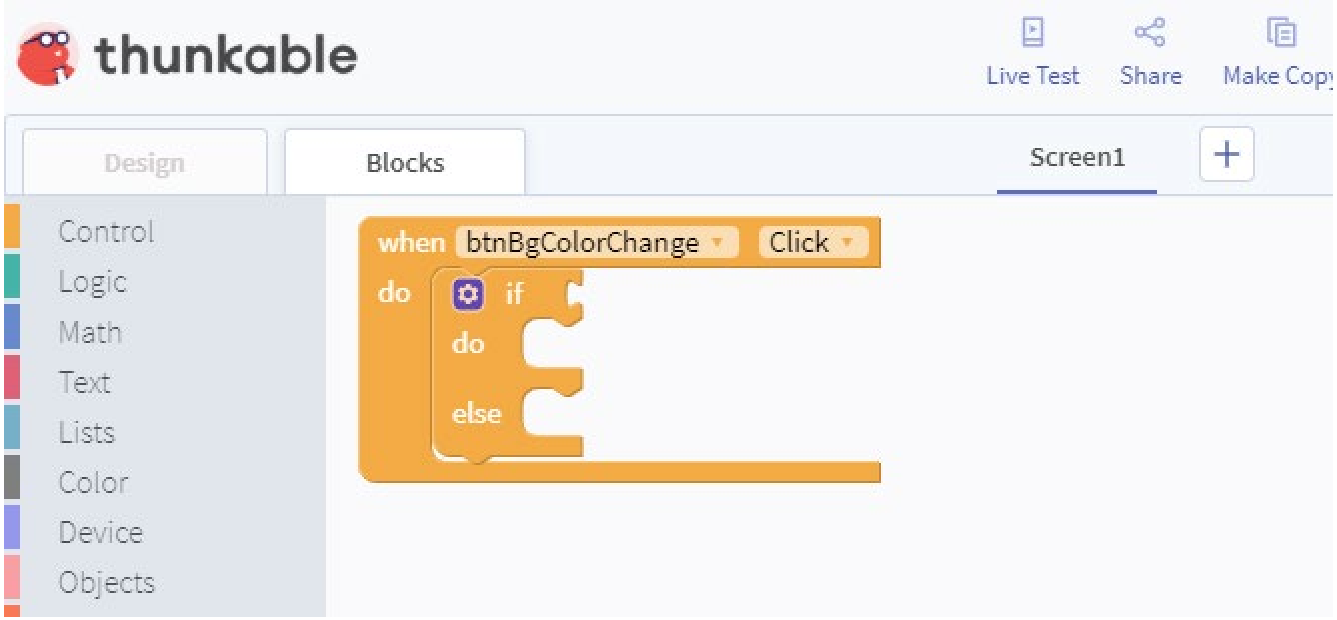
```
when btnBgColorChange Click
do

when btnBgColorChange Long Click
do

when btnBgColorChange Touch Down
do

when btnBgColorChange Touch Up
do

from btnBgColorChange set Text to " Button "

from btnBgColorChange get Text
```



```
when btnBgColorChange Click
do
```

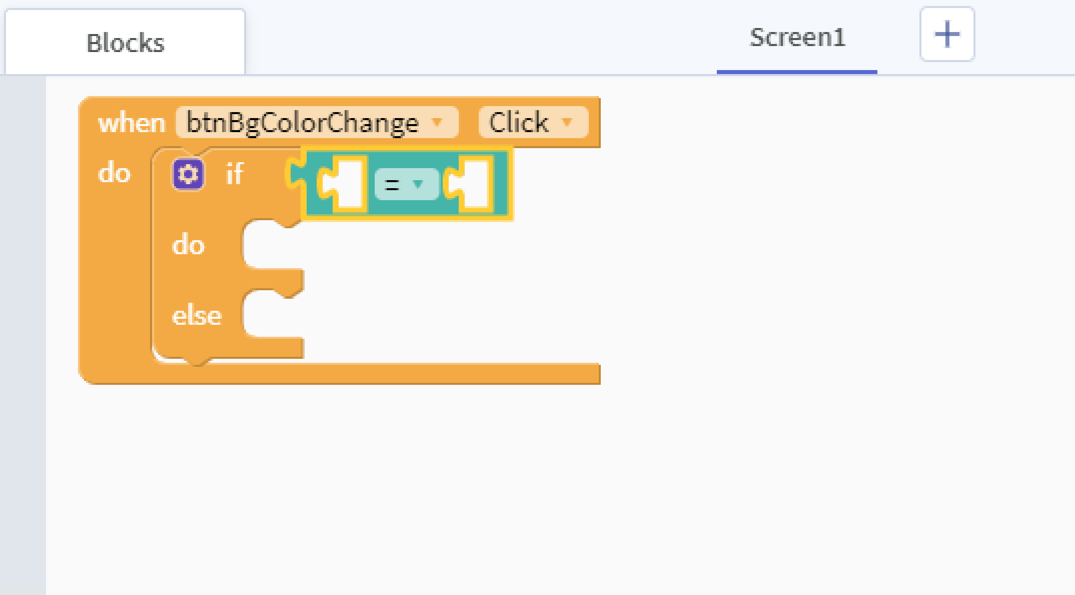
How-To for the Thunkable platform – Part 10

Now we’re going to go to the *Control Blocks* and find the if, do, else block and drag it into the When *Blocks* we just added to the application



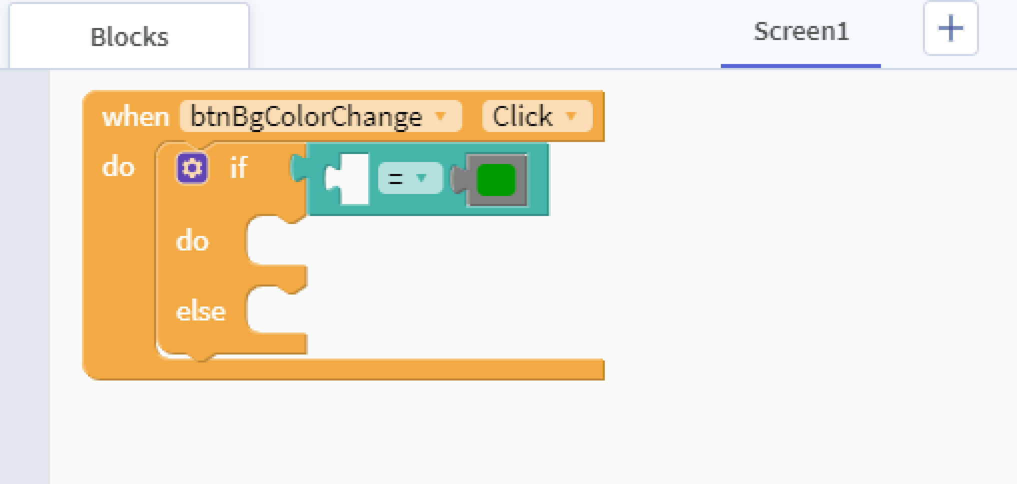
How-To for the Thunkable platform – Part 11

Once you have the if statement in place, we need to add an if condition, this is going to require an “=” logic block. So lets go to the *Logic Blocks* section and grab the equals block and drag it to our if statement



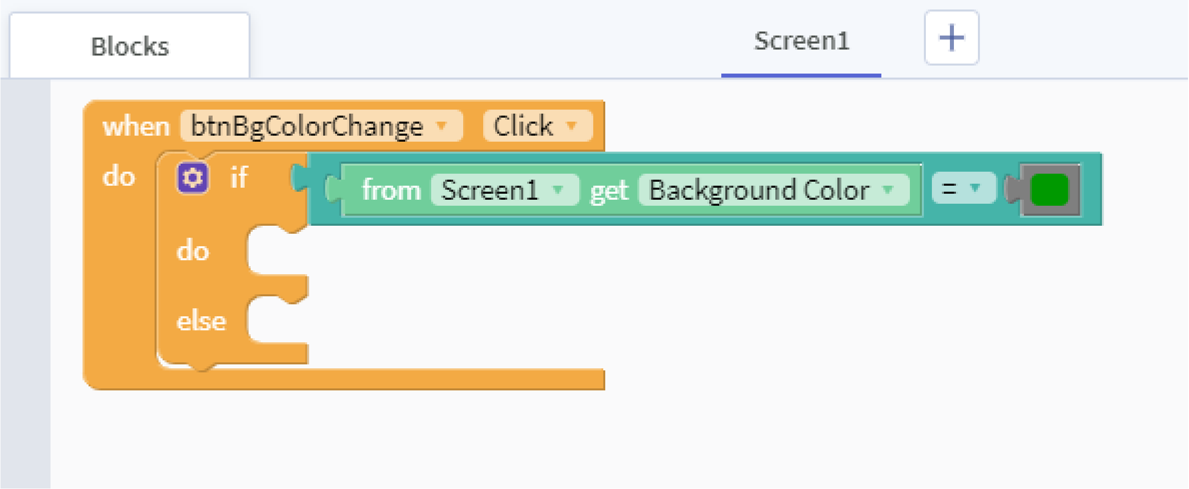
How-To for the Thunkable platform – Part 12

You’ve probably noticed that this equals statement doesn’t have actual conditions to check if they’re equal! Let’s fix that, first we’re going to grab a color from the Color block session and grab any of the preset *Color Blocks* and drag it to the right hand side of the equal logic and click on it to set it to a color of your choice.



How-To for the Thunkable platform – Part 13

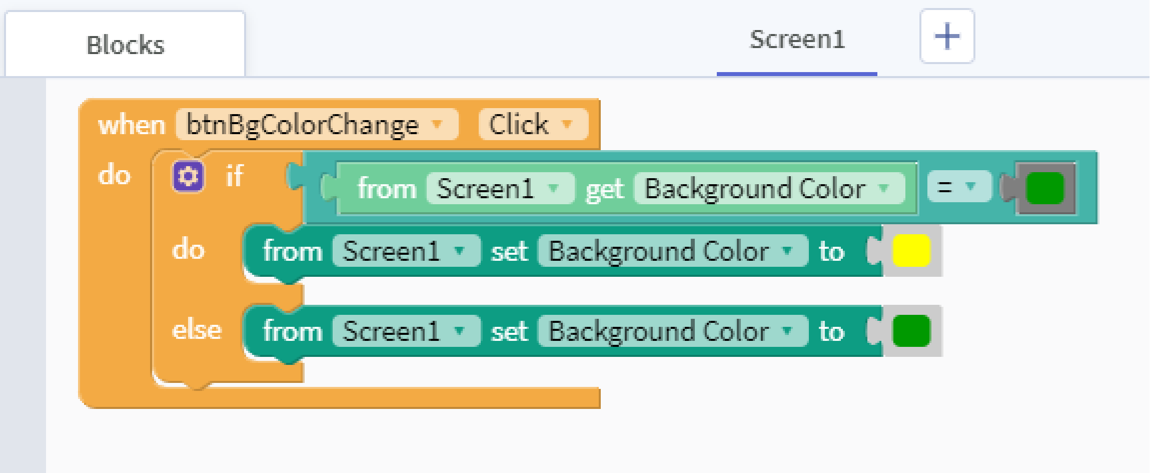
Now we have to finish filling out the if logic, let’s go to the screen that button is sitting on (In this example its Screen1) and pick the from Screen1 get *Background Color* block and add that to the left hand side of the “= ” logic block



Blocks
Screen1
+
when btnBgColorChange Click
do if from Screen1 get Background Color =
do
else

How-To for the Thunkable platform – Part 14

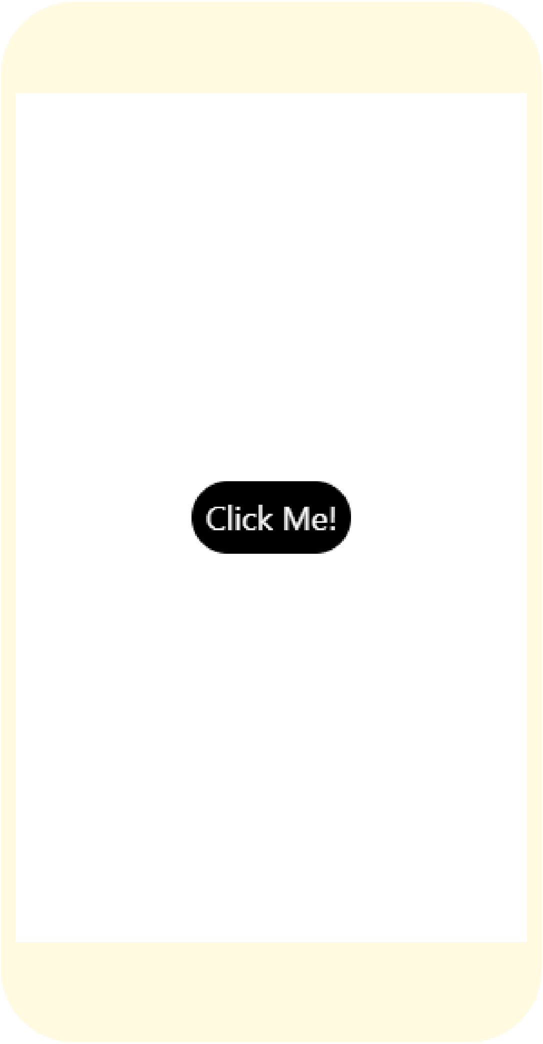
While we're working with the *Screen Blocks* lets go back and add a from Screen1 set *Background Color* block to both the do and the else sections of the if statement and set them to two different colors, with the color block in the else statement matching the if logic color



```
when btnBgColorChange ▾ Click ▾
do  if  from Screen1 ▾ get Background Color ▾  = ▾  🟩
    do    from Screen1 ▾ set Background Color ▾ to 🟨
    else  from Screen1 ▾ set Background Color ▾ to 🟩
```

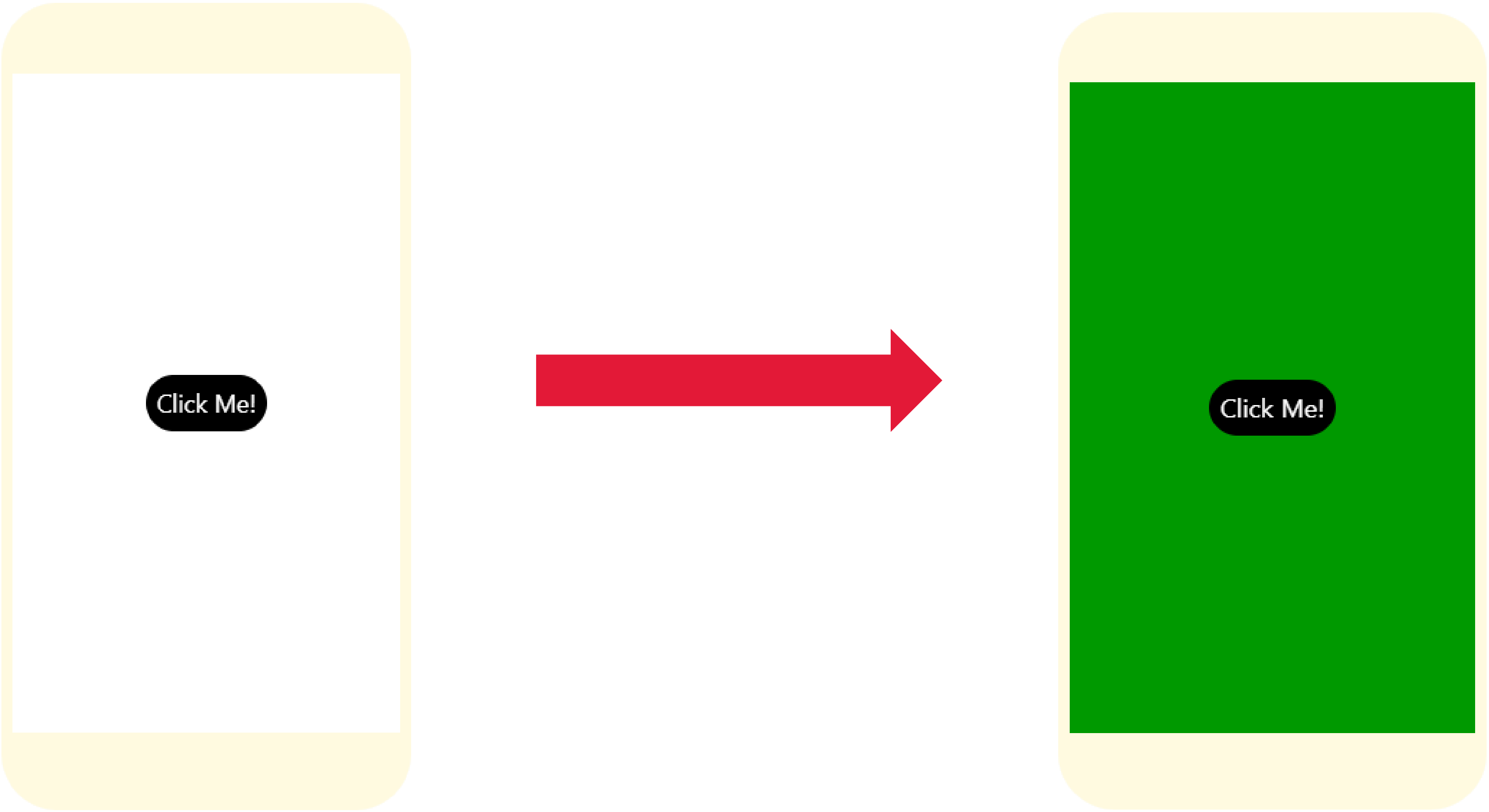
How-To for the Thunkable platform – Part 15

Now that we have our basic logic blocks in place let's click the *Live Test*! For the purpose of this how-to we'll use the browser demo and when we click the button we can see that it switches the background colors based off of the logic we set up!



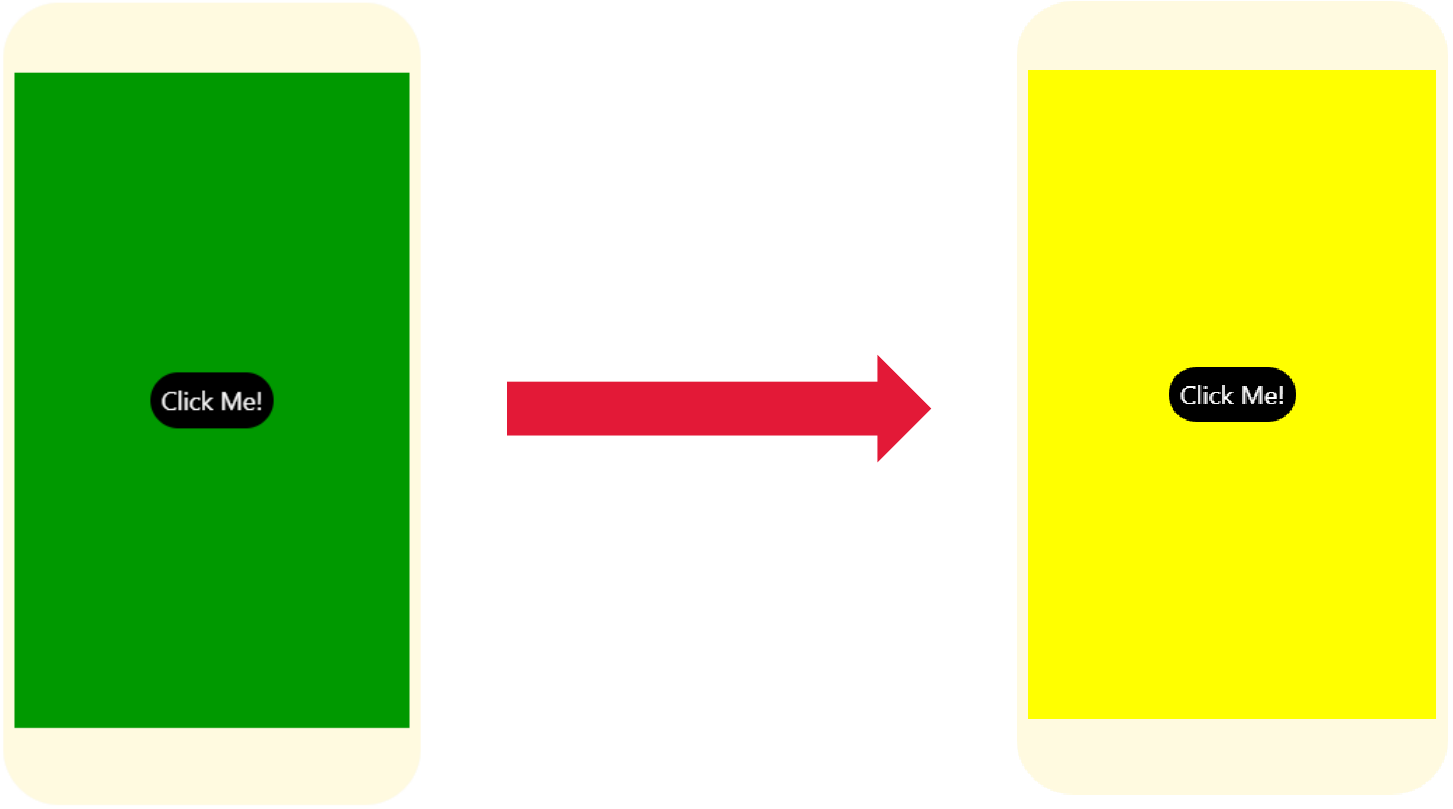
How-To for the Thunkable platform – Part 15 (cont.)

When it first opens the background color is White, which means when we click on the button it should turn green based off of the logic we created.



How-To for the Thunkable platform – Part 15 (cont.)

And here we can see that it did! When we click it again it should turn Yellow. And it did! And that covers the basics of setting up a simple application in Thunkable.



Our commitment

We are passionate about helping students in our communities become the next-generation of information technology professionals.