# Worksheet 1

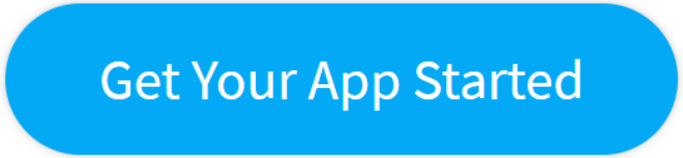
Tutorial - iOS and Android apps in Thunkable

# App 1 - TalkToMe

- This app involves a button you can press to get your phone to speak to you.

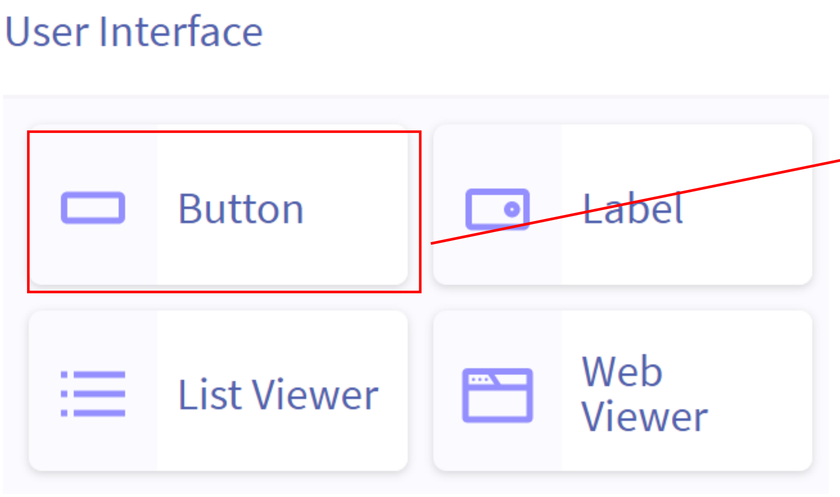
# Getting started

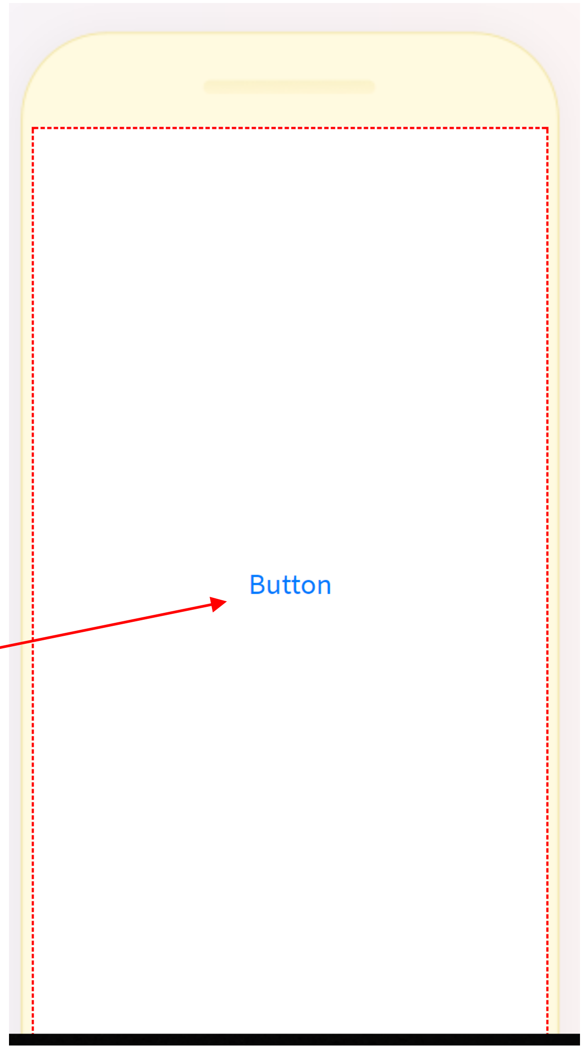
- Go to [https://thunkable.com](https://thunkable.com) or Google ‘Thunkable’.
- Click on the blue ‘Get Your App Started’ button.
- You should select iOS here, as you won’t be able to demo the Android version in the school.
- Click on ‘Create New App’ and call your app ‘TalkToMe’.



# Design/Properties View

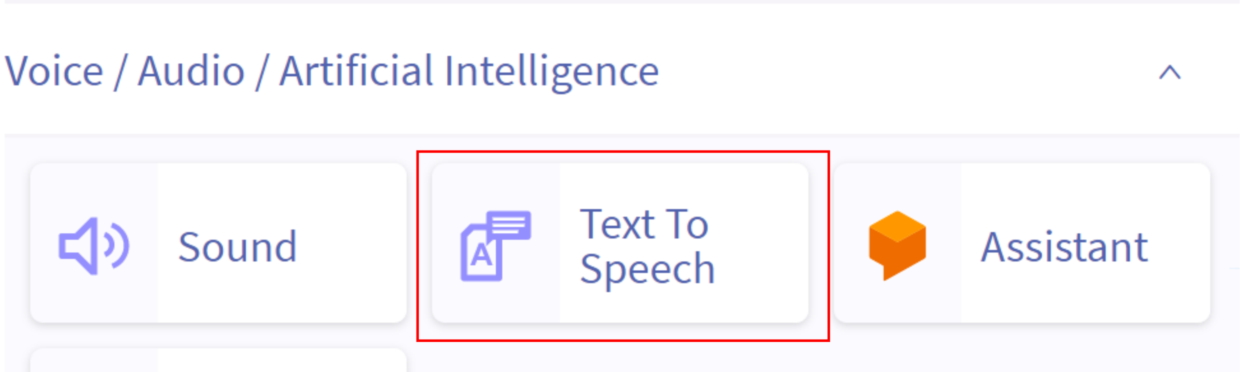
- Click on the new app that has appeared. In this view, you can control the look of the user interface.
- We want to add a button by clicking ‘Add component’ then finding ‘Button’ in the user interface section, dragging and dropping it into the editor.
- Set the text for this button to ‘Talk to Me!’





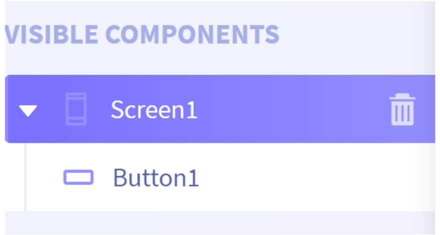
# TextToSpeech

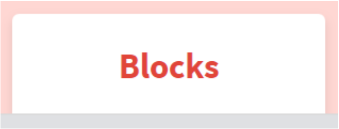
- We also want to add a ‘TextToSpeech’ component, which can be found under ‘Voice/Audio/Artificial Intelligence’.
- It will disappear when you drop it into the editor.



# Blocks view

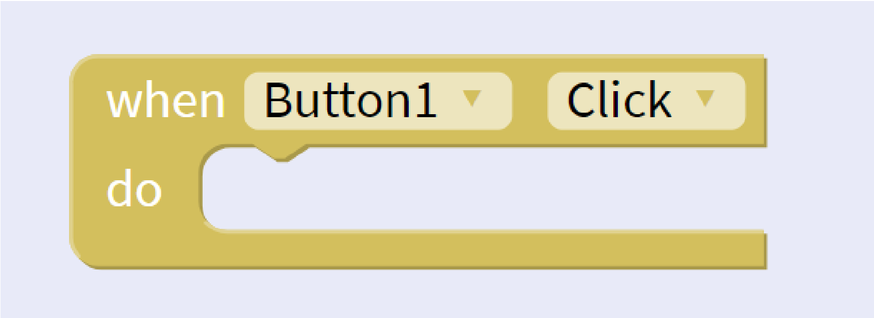
- Click on screen 1, then go to the ‘Blocks’ view by pressing the ‘Blocks’ tab at the top.
- From here, you can control what the app will do through a ‘block-style’ programming language, just like Scratch.





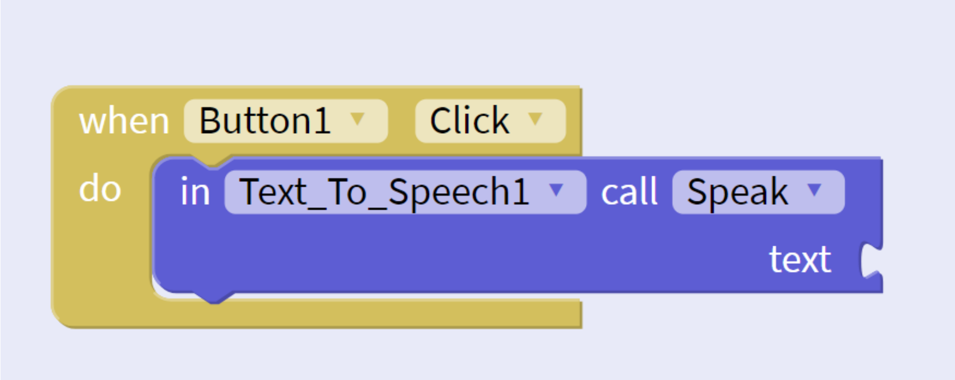
# Adding blocks…

- If you click on ‘Button1’ in the left panel you will see all the blocks for that button.
- Find the ‘when Button1.Click’ block and drag and drop it into the editor.



# Adding blocks…

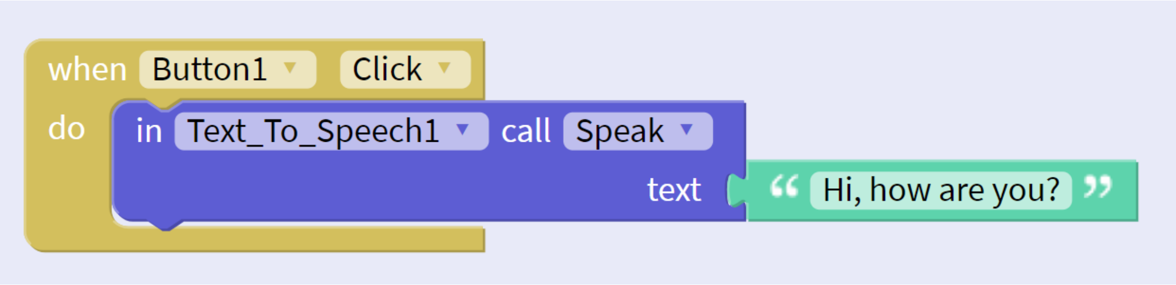
- Now we want a block for the ‘TextToSpeech’ component.
- Under ‘TextToSpeech1’, find the ‘in TextToSpeech1 call Speak’ block. Drop it inside the block you already have and it should click into place.



when Button1 Click
do in Text_To_Speech1 call Speak
text

# Adding blocks…

- The final block we need can be found under ‘Text’. It is an empty text block. Drag and drop it onto the end of the blocks in your editor.
- You can type anything you like in this block and your phone will speak it out loud when you press the button on the app.
- Once you have done this, the TalkToMe app is complete, well done!
- Can you think of any ways to make the app better?



```
when Button1 ▾ Click ▾
do  in Text_To_Speech1 ▾ call Speak ▾
                           text " Hi, how are you? "
```

# Demo your app

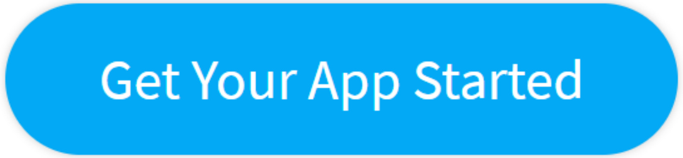
- Open the ‘Thunkable Live!’ app on an iPhone/iPad.
- Sign in to your Google account.
- On the Thunkable website on the PC, select ‘Live Test’ at the top of the screen.
- Your app should now run on the iPhone/iPad.
- To demo an Android app, which you will have to do at home, you should click on ‘Test’ and then use the Thunkable app on your phone to scan the QR code.  Unfortunately, this won’t work in the school.

# App 2 - BallBounce

- The BallBounce app allows you to through a ball around the screen and bounce it off the sides.
- This will be an **Android** app.  Unfortunately you won’t be able to demo it in the school, however, you can demo it at home on your own computer.
- This app and the next one will help you to explore the features of Thunkable more deeply.

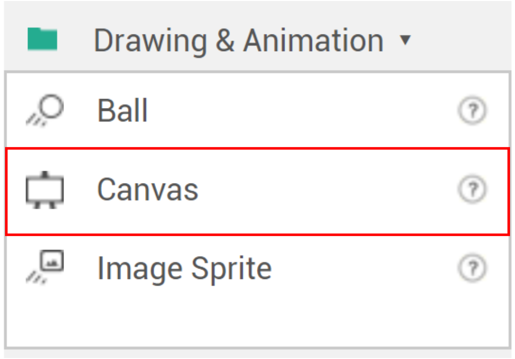
# Getting started

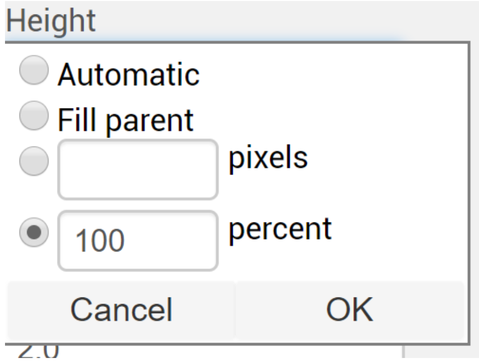
- Return to the Thunkable homepage: [https://thunkable.com](https://thunkable.com)
- This time we will be making an Android app, so click on ‘Get your app started’ then choose Android. You will notice that the editor is slightly different from the iOS version.



# User Interface

- This time we need two components from the ‘Drawing and Animation’ section.
- Find the ‘Canvas’ component and drop it in.  In the properties menu for the canvas, set its height and width to 100%.




Height

○ Automatic

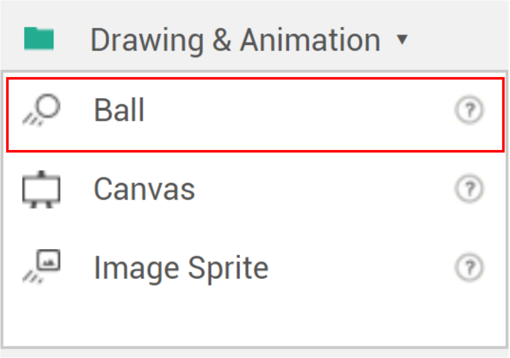
○ Fill parent

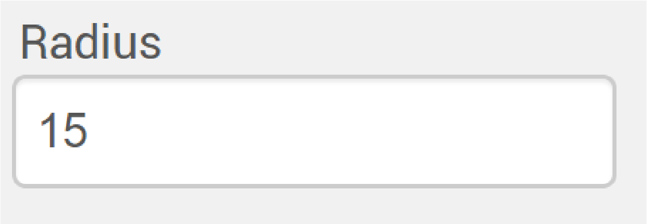
○ pixels

● 100 percent

# User interface

- Drag and drop the ‘Ball’ component into the canvas.
- To make it a bit bigger, set its radius to 15, in the properties menu.



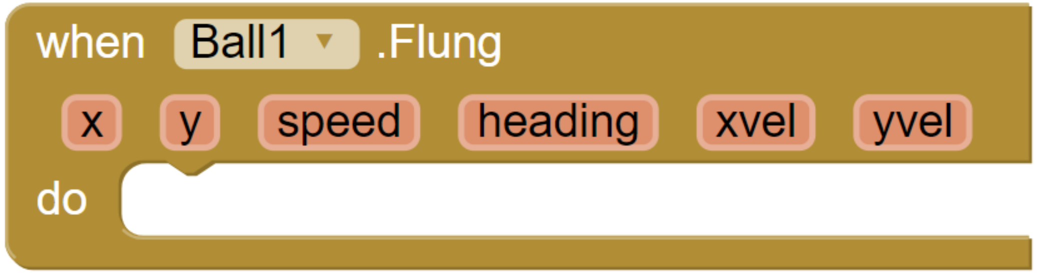


Radius

15

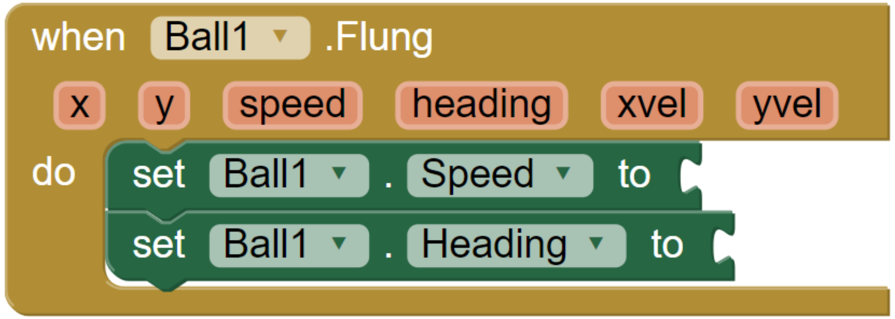
# Blocks editor

- Return to the blocks editor for this app.
- Find the ‘Ball1’ component at the left-hand-side, and click on it to find the ‘when Ball1.Flung’ block. Drag and drop this into the editor.



# Blocks editor

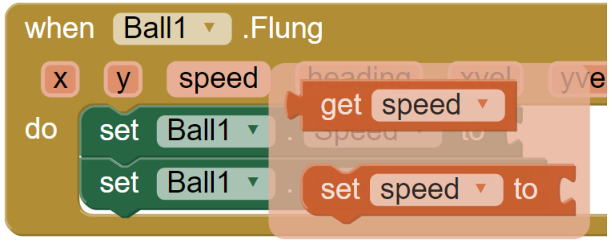
- Inside the ‘Ball1’ component you will find blocks to set the balls speed and heading.  Drag and drop them in.



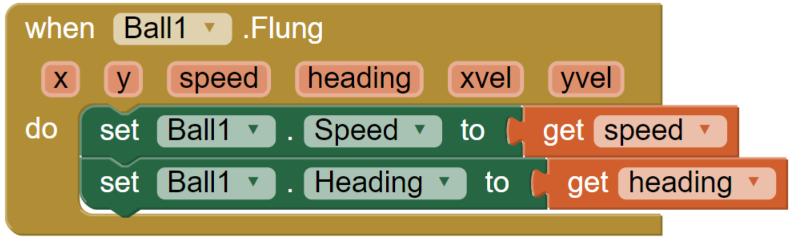
```
when Ball1 ▾ .Flung
 x  y  speed  heading  xvel  yvel
do  set Ball1 ▾ . Speed ▾ to
    set Ball1 ▾ . Heading ▾ to
```

# Blocks editor

- Click on the orange word ‘speed’. This will give you a new block called ‘get speed’. Drop that new block onto the end of the ‘setBall1.Speed’ block, and do the same for heading.
- These values are called **parameters**.



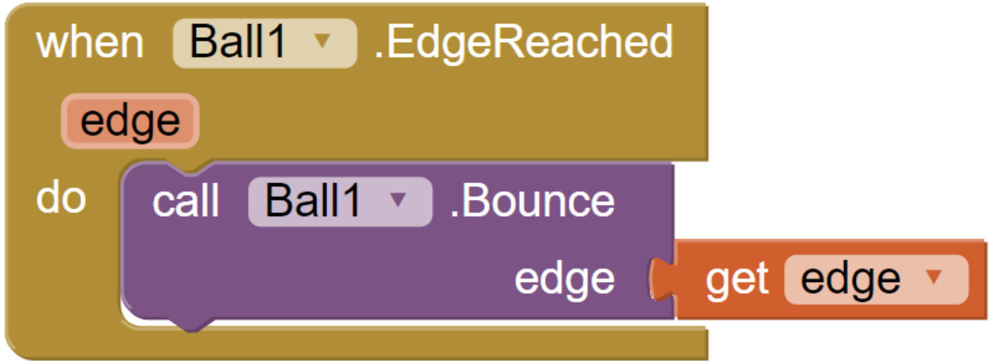
```
when Ball1 .Flung
  x  y  speed  heading  xvel  yve
do  set Ball1 . Speed to   get speed
    set Ball1 .            set speed to
```



```
when Ball1 .Flung
  x  y  speed  heading  xvel  yvel
do  set Ball1 . Speed to get speed
    set Ball1 . Heading to get heading
```

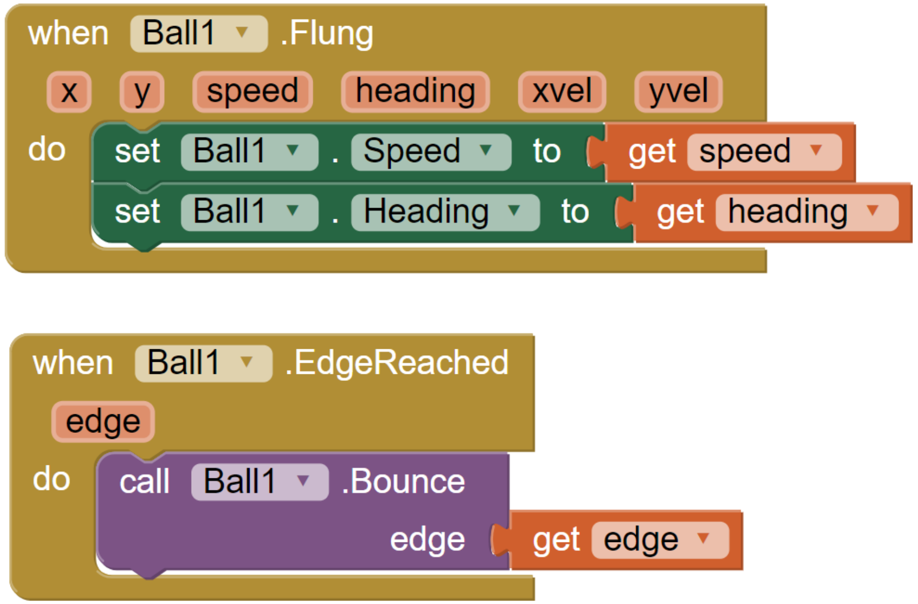
# Blocks editor

- The last thing we want to do is make the ball bounce when it hits the edge of the editor.  See if you can create the following set of blocks:



# Final BallBounce Program

- Your program should now look like this:



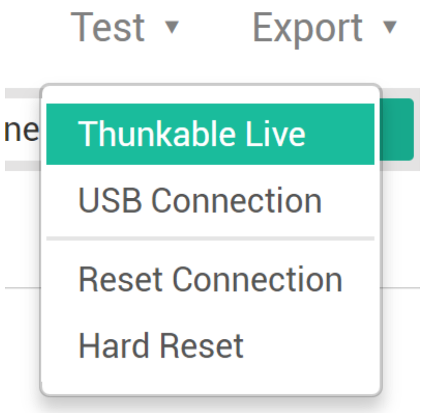
```
when Ball1 .Flung
 x  y  speed  heading  xvel  yvel
do  set Ball1 . Speed to  get speed
    set Ball1 . Heading to  get heading

when Ball1 .EdgeReached
 edge
do  call Ball1 .Bounce
                  edge  get edge
```

# Demo your app

- Unfortunately, you won’t be able to demo this app in the school. However, you can demo it at home.
- To do this, return to Thunkable, find your app, and select ‘Test’

  ->’Thunkable Live’.
- Then scan the barcode using the Thunkable app on your Android phone.



# App 3 - DigitalDoodle

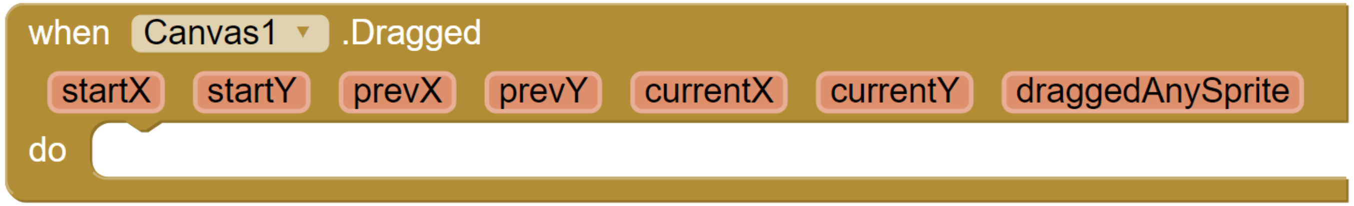
- DigitalDoodle is an app that lets you draw on the screen, as if you were drawing with a pencil.
- This is also an **Android** app, which you will only be able to demo at home.

# Getting Started

- Start off as you did before.  Make sure you select ‘Android’.
- Name your app ‘DigitalDoodle’.

# User Interface & First Block

- Once again, we want to use a canvas component.  Just like you did for BallBounce, drag and drop the canvas in and make its height and width 100%.
- Now switch to the blocks editor, and find the following block:



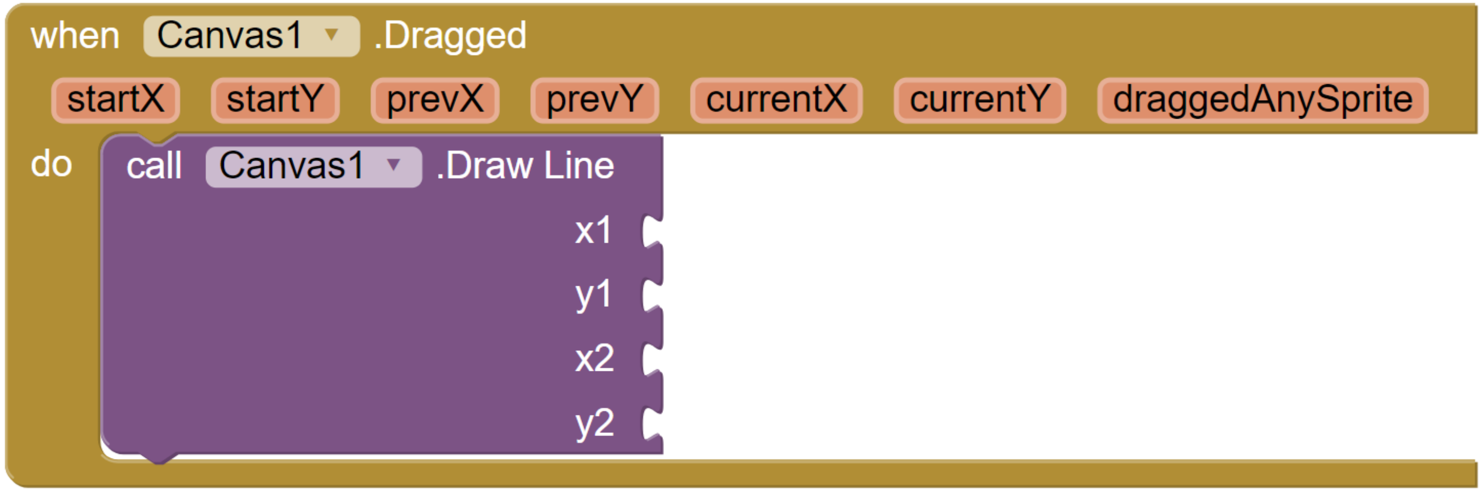
when Canvas1 ▾ .Dragged

startX startY prevX prevY currentX currentY draggedAnySprite

do

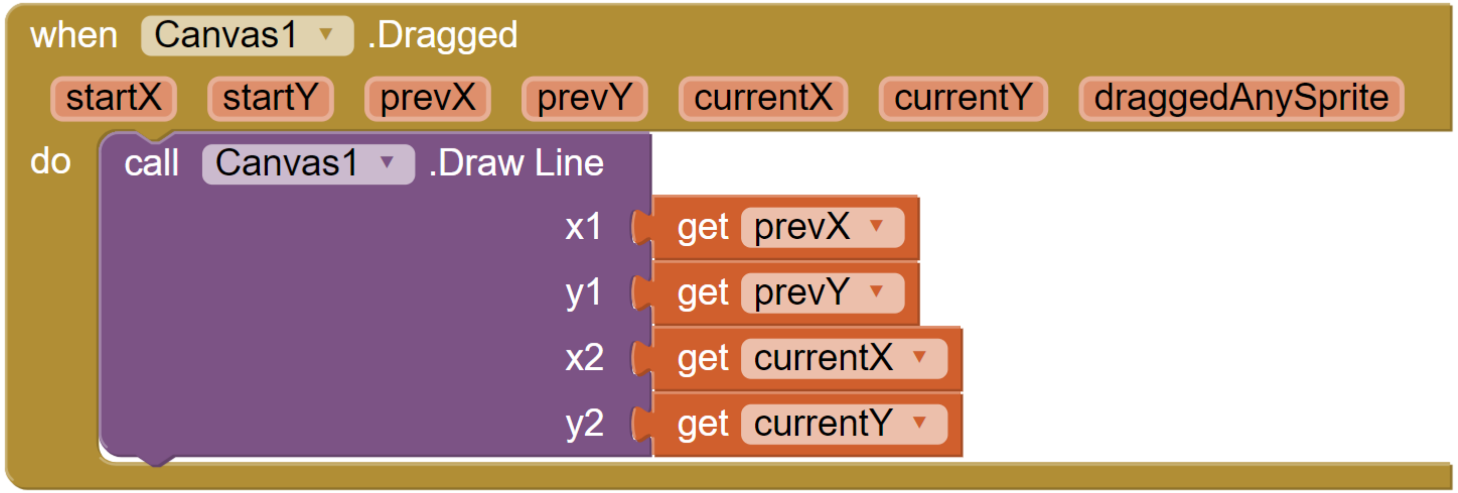
# Blocks editor

- When a user drags their finger across the canvas, we want it to draw a line, so we need the ‘Canvas1.DrawLine’ block.



# Final DigitalDoodle Program

- Use the **parameters**, to get the following block.  Remember, you used parameters already for the BallBounce app.



```
when Canvas1 .Dragged
  startX  startY  prevX  prevY  currentX  currentY  draggedAnySprite
do  call Canvas1 .Draw Line
                     x1  get prevX
                     y1  get prevY
                     x2  get currentX
                     y2  get currentY
```

# Well Done!

- You have now completed the first worksheet.  You can now proceed onto worksheet 2, which you should work on in **pairs**.
- Remember, you can easily demo any iOS apps you have made in the school on the iPads or on your iPhone if you have one.
- You can’t demo your Android apps in the school, but you can demo them at home, so it is up to you what kind of apps you decide to build later on.