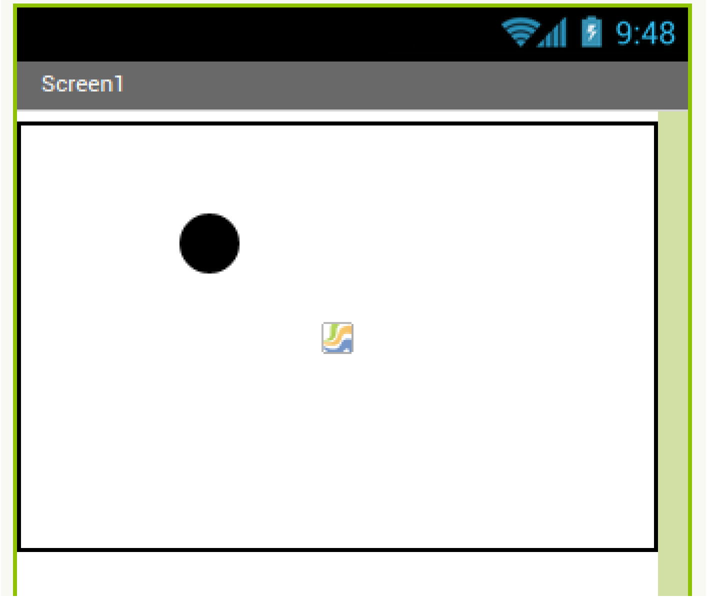
Bounce Sprite Off Canvas Edge

Make a ball bounce when it hits an edge of a Canvas.

Edge = 1

Edge =-3



Edge =-3

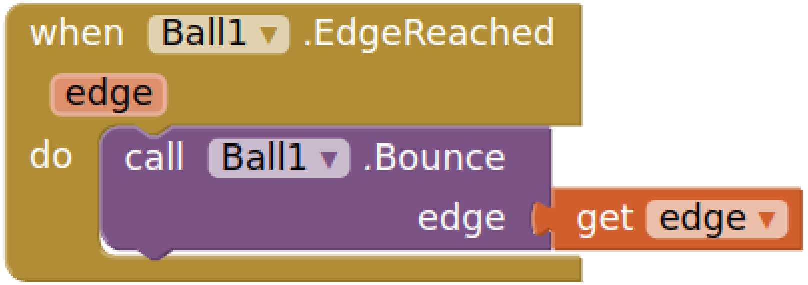
Edge = -1

Edges are assigned values (1,3,-1,-3
For top, right, bottom, left

Getting Ready

You will need the following components in your design screen:
* Canvas
* Sprite

Blocks Editor



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

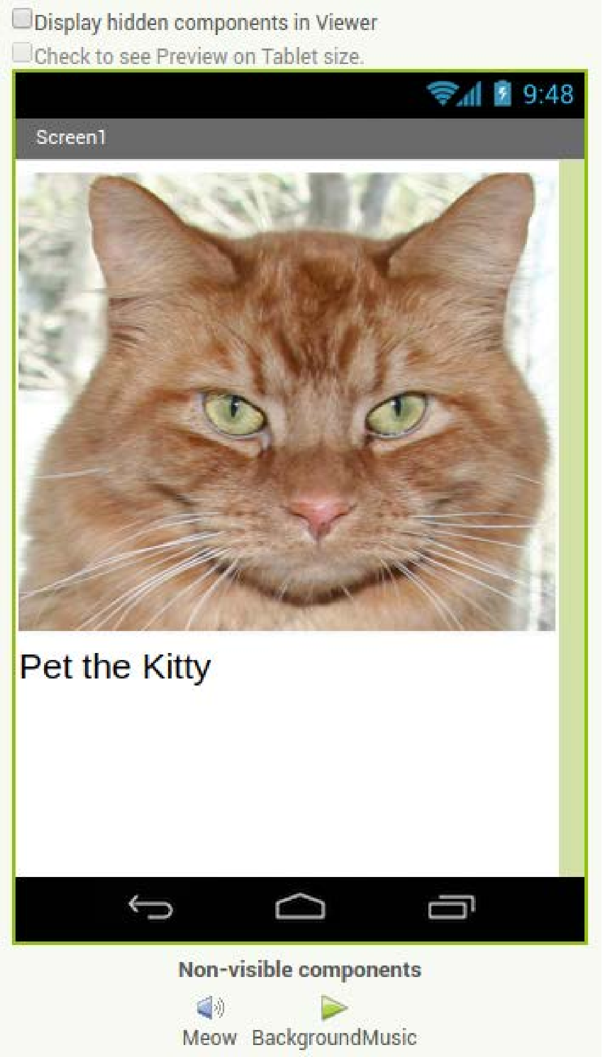
What does it mean?

The **Ball1.EdgeReached** event will detect when the Ball sprite hits the edge of the Canvas and provides an argument **edge**. Each **edge** of a Canvas holds a numeric value. So just feed back the same value **edge** into the **Bounce** call.

Ball1.Bounce causes the ball to bounce and move in the opposite direction off the wall (**Ball1.Heading** is changed by 180 degrees .

Adding Sound

Add sound by either adding short audio files
Like a crash sound when two objects collide
Or longer audio files like background music.



□ Display hidden components in Viewer

□ Check to see Preview on Tablet size.

Pet the Kitty

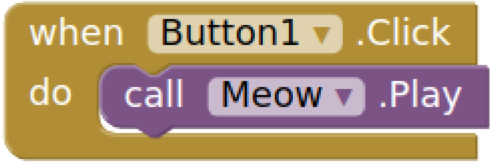
Non-visible components

Getting Ready

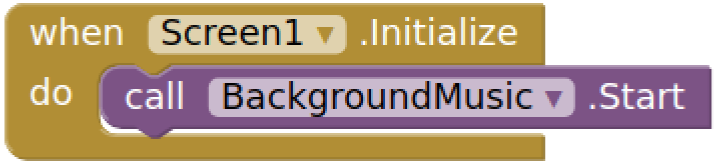
You will need the following components in your design screen:

* Button
* Sound
* Player

Blocks Editor



```
when Button1 .Click
do  call Meow .Play
```



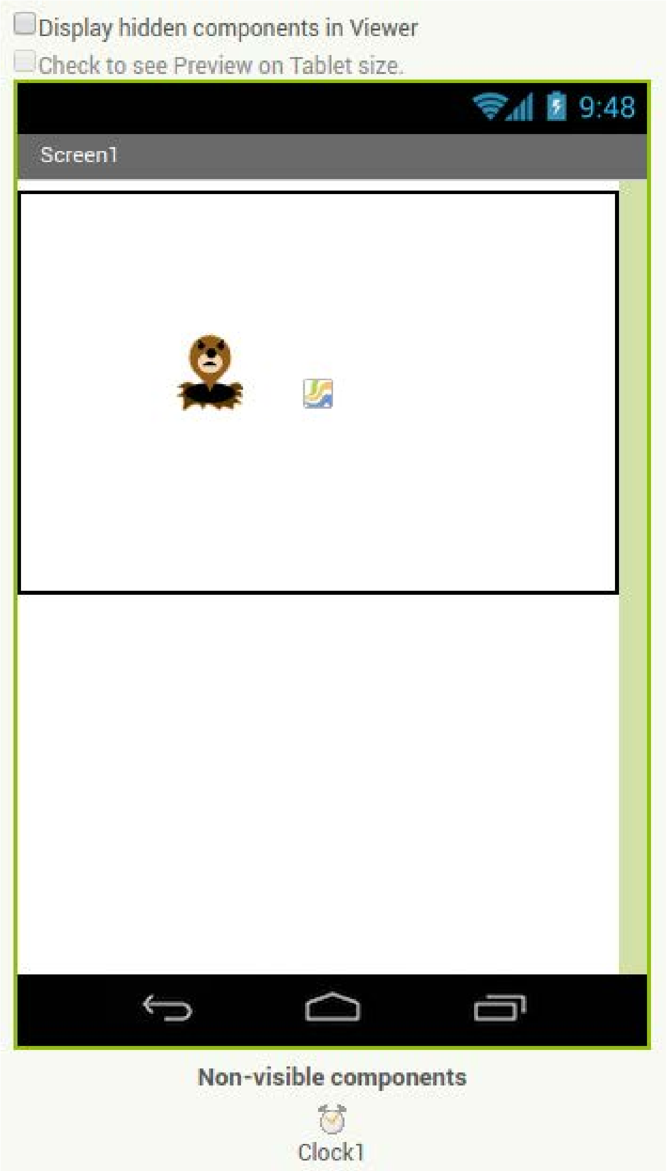
```
when Screen1 .Initialize
do  call BackgroundMusic .Start
```

What does it mean?

When **Button1** is clicked the **Meow**
Sound will play.
When **Screen1** comes up on your phone
The **BackgroundMusic** will start playing.

Movement on a Timer

Move an ImageSprite with the passing of time.



□ Display hidden components in Viewer

□ Check to see Preview on Tablet size.

Non-visible components

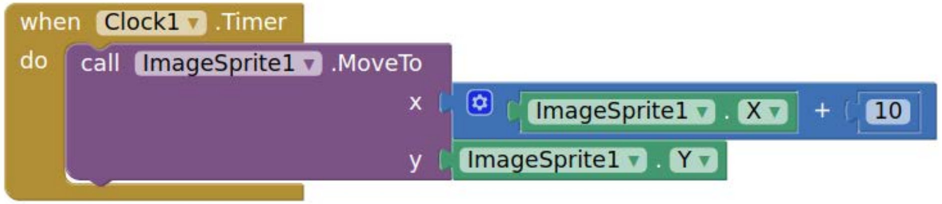
Clock1

Getting Ready

You will need the following components in your design screen:

* Canvas
* ImageSprite
* Clock

Blocks Editor



```
when Clock1 .Timer
do  call ImageSprite1 .MoveTo
        x  ImageSprite1 . X + 10
        y  ImageSprite1 . Y
```

What does it mean?

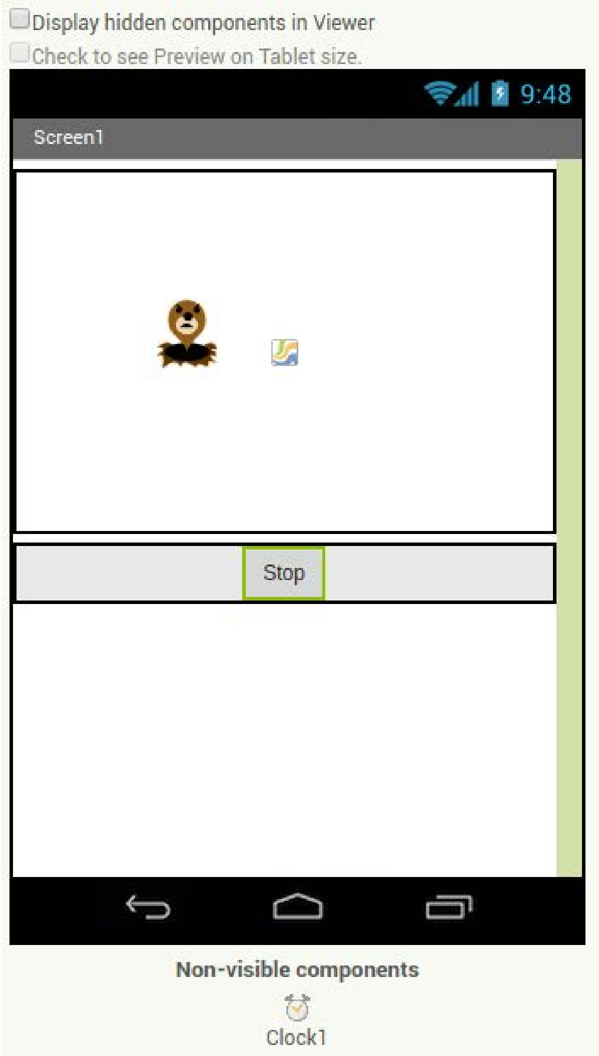
MoveTo moves an object to an absolute Location on the canvas, not a relative Amount.

Clock1 has an Interval that specifies how Often the Timer should go off and the Timer Event should be called.

Whenever the **Clock1.Timer** fires the Sprite will move to the right 10 Pixels since the Sprite's Interval is 1.

Start/Stop Timed Movement

This allows the end user to touch a button to start and stop an ImageSprite moving with the passing of time.



□ Display hidden components in Viewer

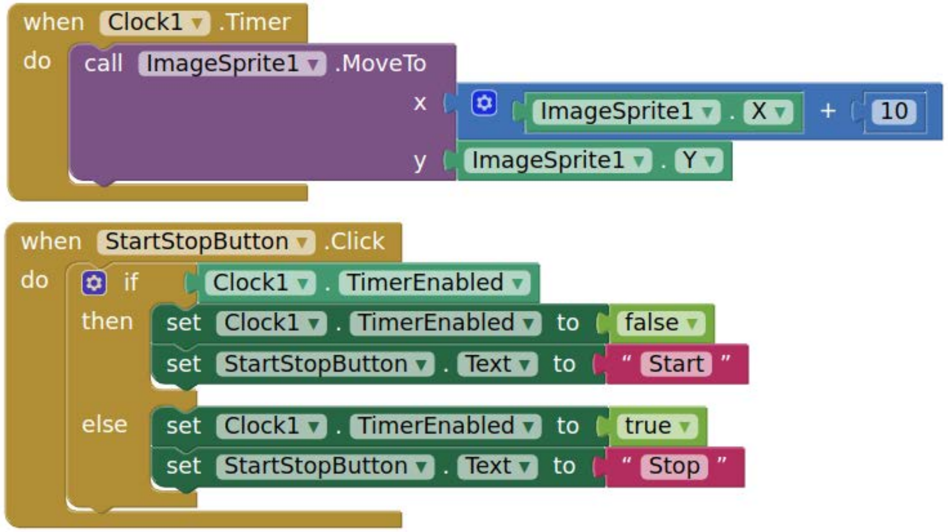
□ Check to see Preview on Tablet size.

Getting Ready

You will need the following components in your design screen:

* Canvas, ImageSprite, Clock, Button

Blocks Editor



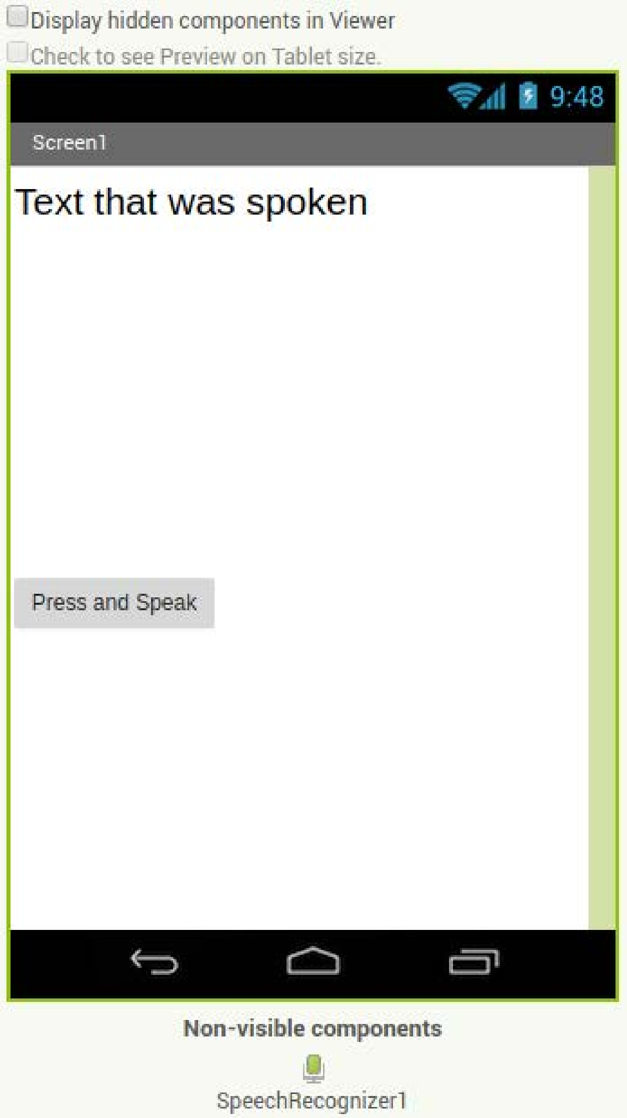
```
when Clock1 .Timer
do  call ImageSprite1 .MoveTo
                       x  ImageSprite1 . X + 10
                       y  ImageSprite1 . Y

when StartStopButton .Click
do  if    Clock1 . TimerEnabled
    then  set Clock1 . TimerEnabled to false
          set StartStopButton . Text to " Start "
    else  set Clock1 . TimerEnabled to true
          set StartStopButton . Text to " Stop "
```

What does it mean?

When the **StartStopButton.Click** is touched, if the Clock is enabled then stop the timer and display Start on the button. This will stop the Sprite's Movement. The opposite will happen when the clock is disabled.

Speech Recognition

Display the text of what is being said on the phone screen.



□ Display hidden components in Viewer

□ Check to see Preview on Tablet size.

9:48

Screen1

Text that was spoken

Press and Speak

Non-visible components

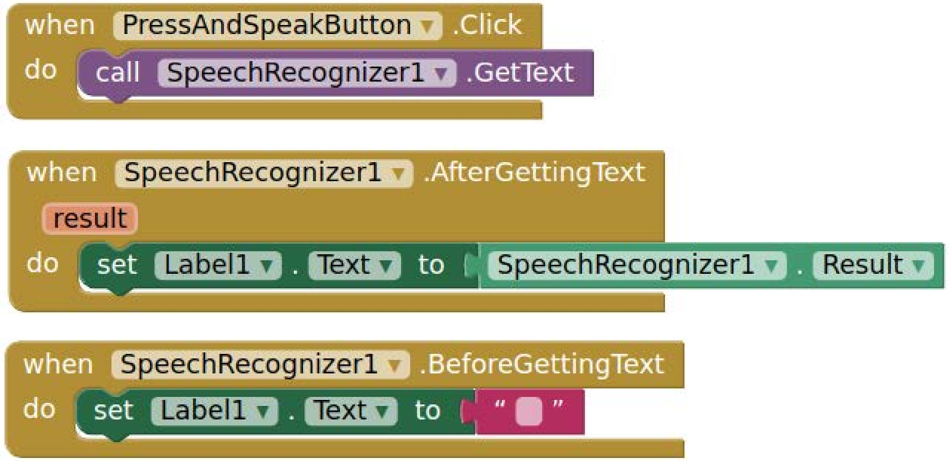
SpeechRecognizer1

Getting Ready

You will need the following components in your design screen:

* Label, Button, SpeechRecognizer

Blocks Editor



```
when PressAndSpeakButton .Click
do   call SpeechRecognizer1 .GetText

when SpeechRecognizer1 .AfterGettingText
  result
do   set Label1 . Text to SpeechRecognizer1 . Result

when SpeechRecognizer1 .BeforeGettingText
do   set Label1 . Text to " "
```

What does it mean?

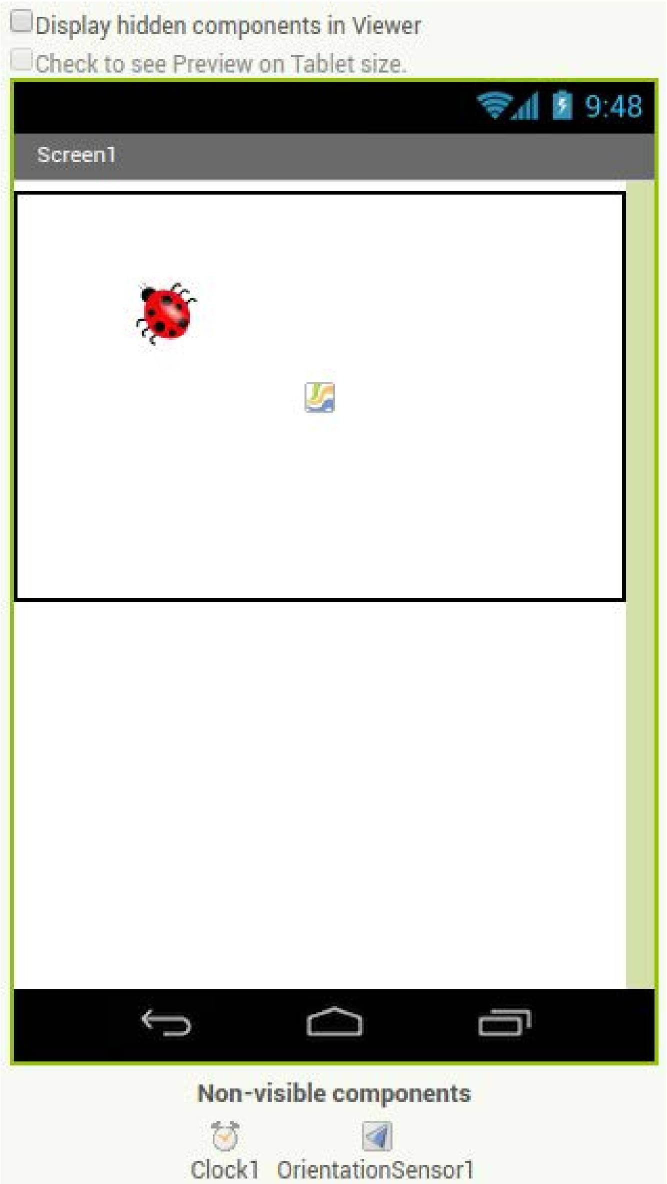
When the **PressAndSpeakButton** is clicked the **SpeechRecognizer** event is called and is ready for you to speak.

The **BeforeGettingText** event will be triggered before speech has been received and recognized. Then the Label will display no text on the screen.

The **AfterGettingText** event will be triggered once speech has been received and recognized. Then the **Label** will display the text on the screen.

Movement with Sensors

Move an ImageSprite by tilting your phone



□ Display hidden components in Viewer

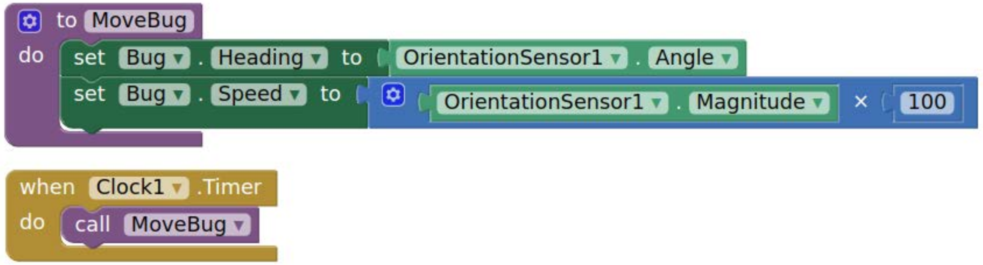
□ Check to see Preview on Tablet size.

Getting Ready

You will need the following components in your design screen:

* Canvas, ImageSprite, OrientationSensor, Clock

Blocks Editor



```
to MoveBug
do  set Bug . Heading to  OrientationSensor1 . Angle
    set Bug . Speed to  OrientationSensor1 . Magnitude × 100

when Clock1 .Timer
do  call MoveBug
```

What does it mean?

A procedure called **MoveBug** was created that moves the bug in the direction that the phone is tilted.

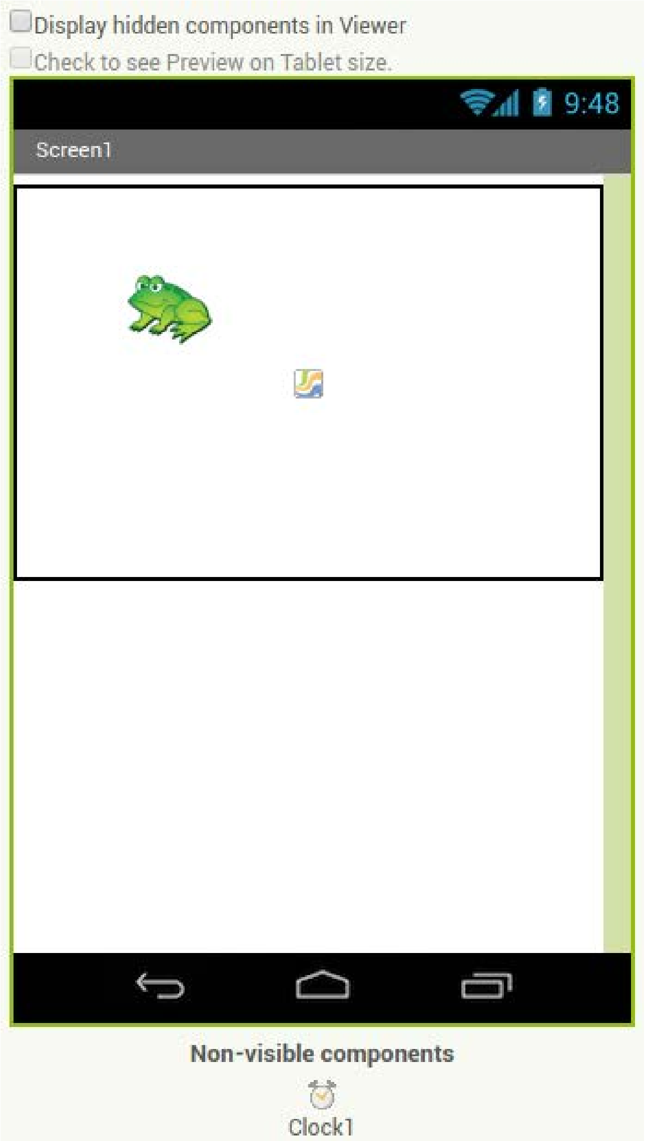
The **OrientationSensor.Angle** is used to tell the bug which direction to move based on what angle your phone is tilted.

The **OrientationSensor.Magnitude** is used to tell the bug what speed to move based on how much tilt you are putting on your phone.

Whenever the **Clock1.Timer** fires, the event **MoveBug** will be called.

Random Numbers

Generate random numbers to make ImageSprites appear in random (x.y)



□ Display hidden components in Viewer

□ Check to see Preview on Tablet size.

Non-visible components

Clock1

Getting Ready

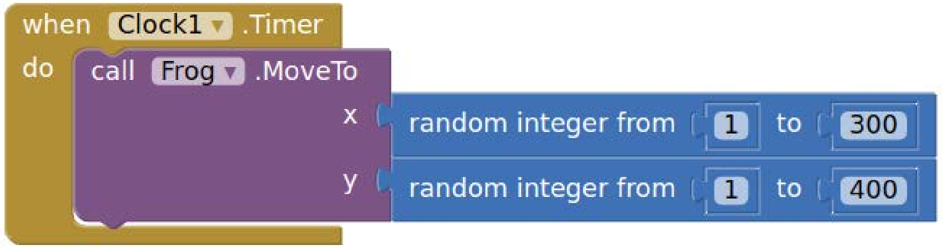
You will need the following components in your design screen:

* Canvas

* ImageSprite

* Clock

Blocks Editor



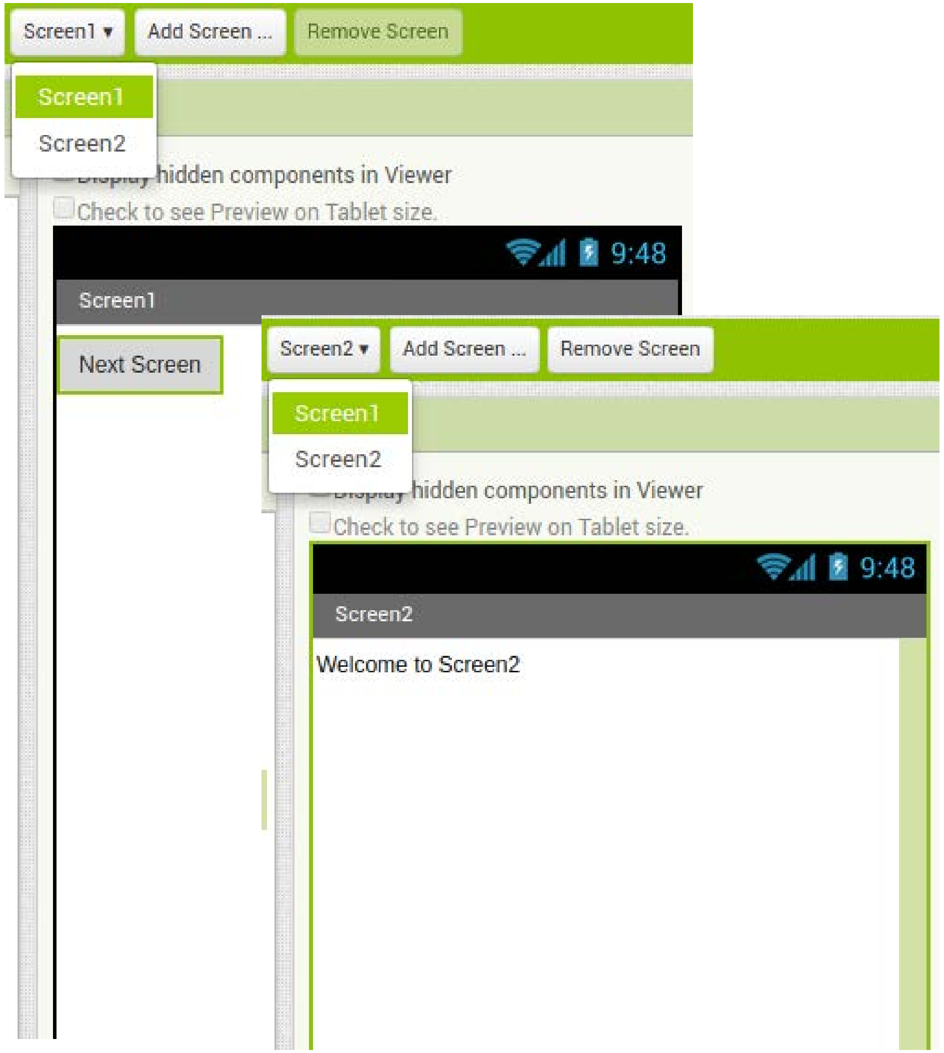
```
when Clock1 .Timer
do  call Frog .MoveTo
                  x  random integer from 1 to 300
                  y  random integer from 1 to 400
```

What does it mean?

When the **Clock1.Timer** event is triggered, then **Frog.Move**To moves the frog to a random coordinates between the values of 1 and 300 for the x coordinate and 1 and 400 for the y coordinate.

Multiple Screens

Use multiple screens in your app.
Get the next screen by clicking a button.



Getting Ready

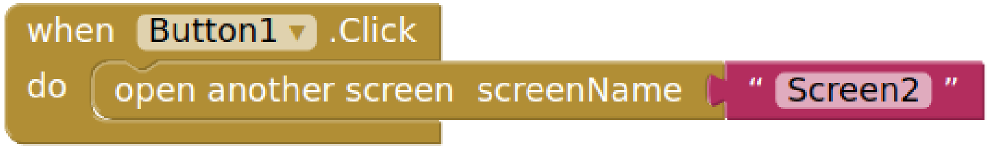
You will need the following components in your design screen:

* Screen1: Button

* Screen2: Label

Blocks Editor

(For Screen1



```
when Button1 .Click
do  open another screen  screenName " Screen2 "
```

What does it mean?

Open another screen takes in a text block. The text inside this block is the name of another screen. When the button is clicked, Screen2 will be opened.

Creating Your Own Color

Create your own colors using the make a color block.

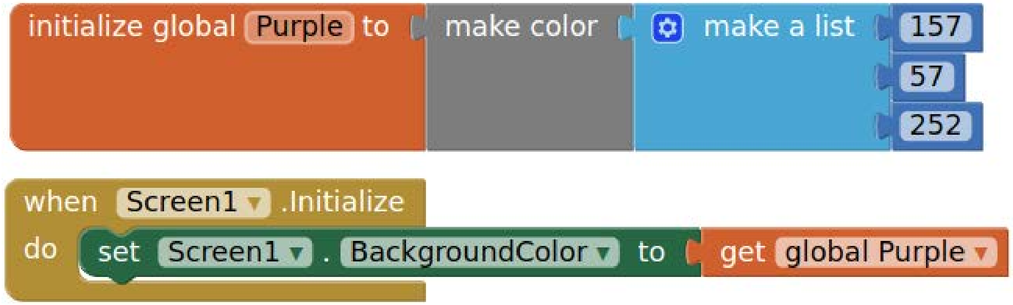
□ Display hidden components in Viewer

□ Check to see Preview on Tablet size.

Getting Ready

No components are necessary to use Make color.

Blocks Editor



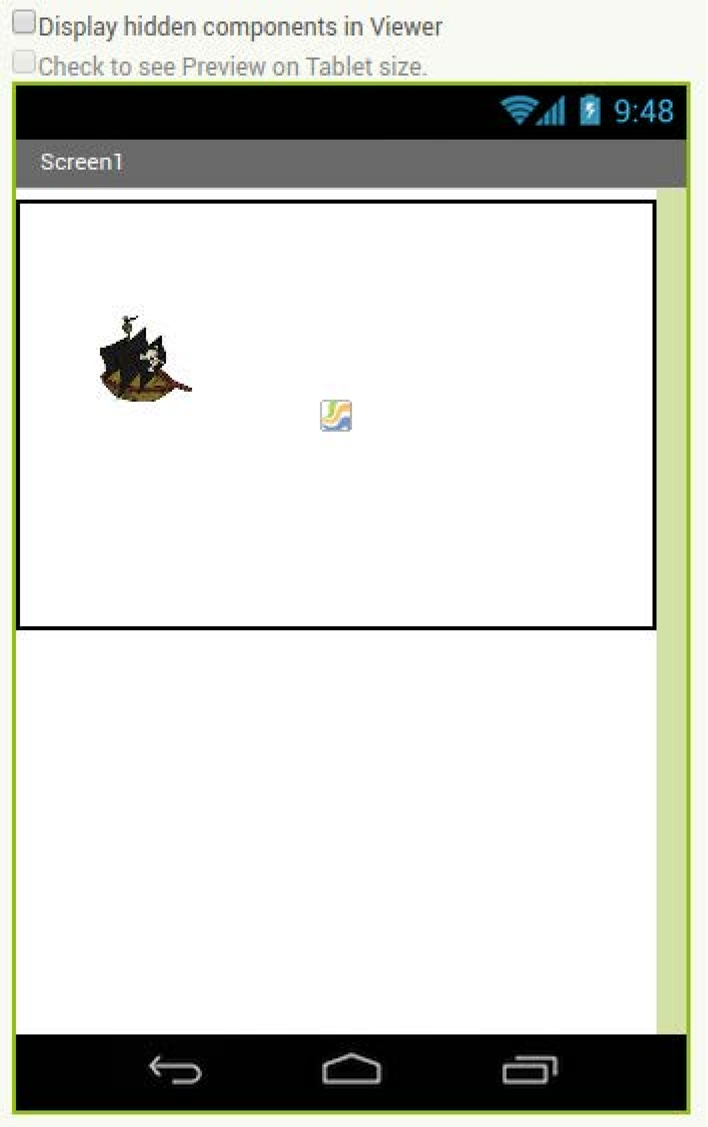
```
initialize global Purple to  make color  make a list  157
                                                       57
                                                       252

when Screen1 .Initialize
do  set Screen1 . BackgroundColor to  get global Purple
```

What does it mean?

Make color takes in a list of 4 numbers. The first three numbers represent the RGB values. The last is the alpha or how strong the color is. **Purple** is made from using 157 as R, 57 as G, 252 as B and 100 as alpha. **When Screen1** is initialized, the background color is set the color we created in the variable Purple.

Fling Movement

Change the heading and speed of a Sprite by flinging your fingers.



□ Display hidden components in Viewer

□ Check to see Preview on Tablet size.

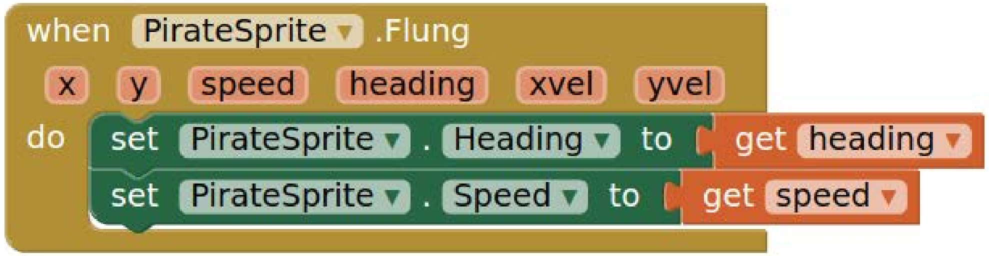
Getting Ready

You will need the following components in your design screen:

* Canvas

* ImageSprite

Blocks Editor



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

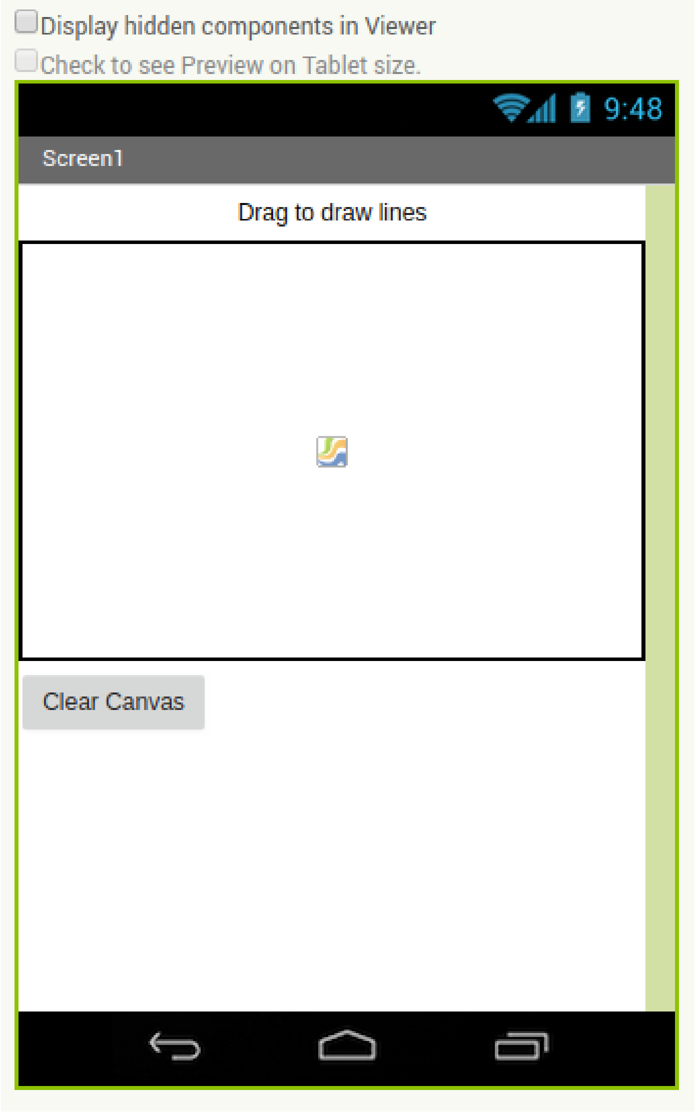
What does it mean?

Flung detects when the user makes a fling motion with the sprite across the screen.

The user sets the heading and speed from the fling to the PirateSprite's **Heading** and **Speed**.

Drawing on a Canvas

Drag your finger across the screen to
Draw a curved line along the path of your finger.



☐ Display hidden components in Viewer

☐ Check to see Preview on Tablet size.

Getting Ready

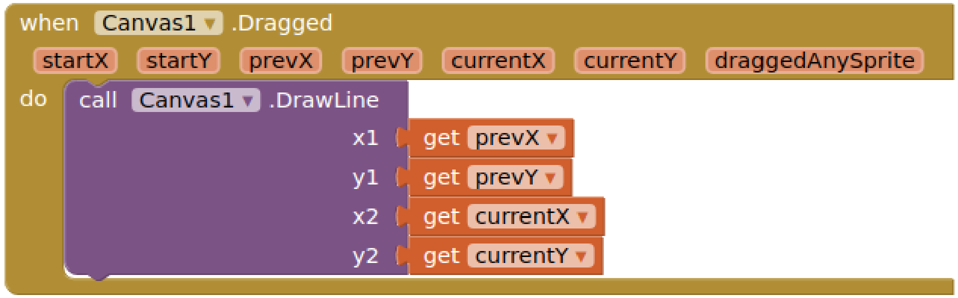
You will need the following components in your design screen:

* Canvas

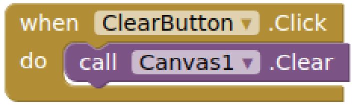
* Button

* Label

Blocks Editor



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



```
when ClearButton .Click
do  call Canvas1 .Clear
```

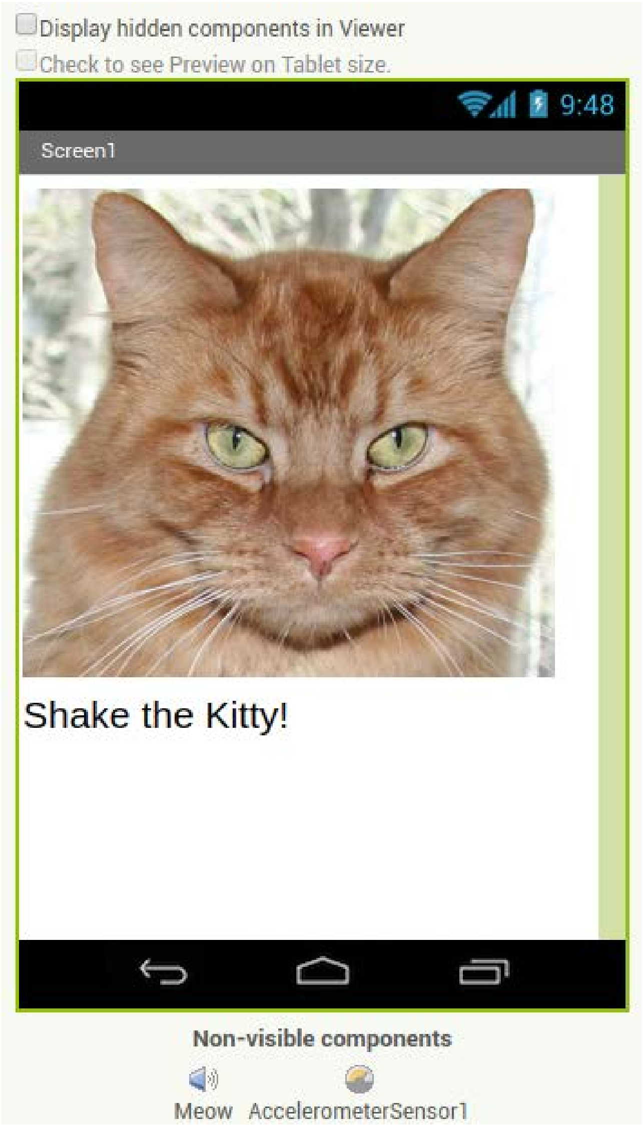
What does it mean?

When the **Canvas1.Dragged** event is triggered, a curved line will be drawn from where you finger started on the screen to where it was dragged.

When the **Clear.Click** button is touched the canvas will be cleared.

Shaking Phone

Make something to happen when you Shake your phone.



□ Display hidden components in Viewer

□ Check to see Preview on Tablet size.

9:48

Screen1

Shake the Kitty!

Non-visible components

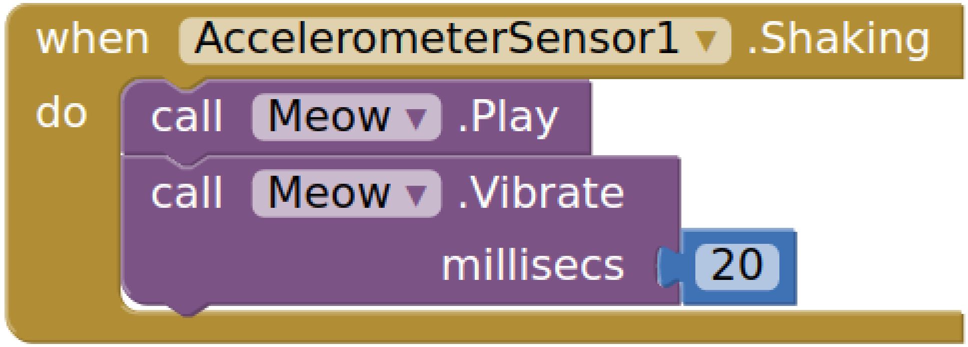
Meow AccelerometerSensor1

Getting Ready

You will need the following components in your design screen:

* Image, Sound, AccelerometerSensor, Label

Blocks Editor



```
when AccelerometerSensor1 ▾ .Shaking
do  call Meow ▾ .Play
    call Meow ▾ .Vibrate
                millisecs  20
```

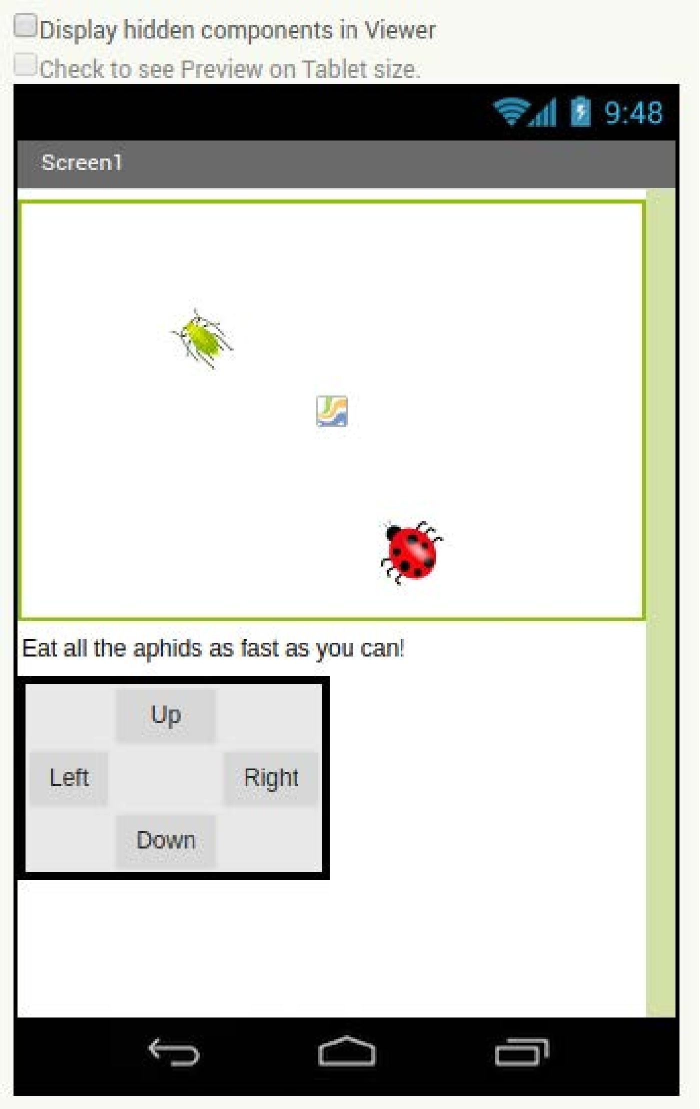
What does it mean?

The **AccelerometerSensor.Shaking** event will detect when the phone is shaking and then the Meow sound will play and the phone will vibrate for 20 milliseconds.

MIT APP INVENTOR

Collision Detection

Make something happen when one Sprite Collides with another.



□ Display hidden components in Viewer

□ Check to see Preview on Tablet size.

Eat all the aphids as fast as you can!

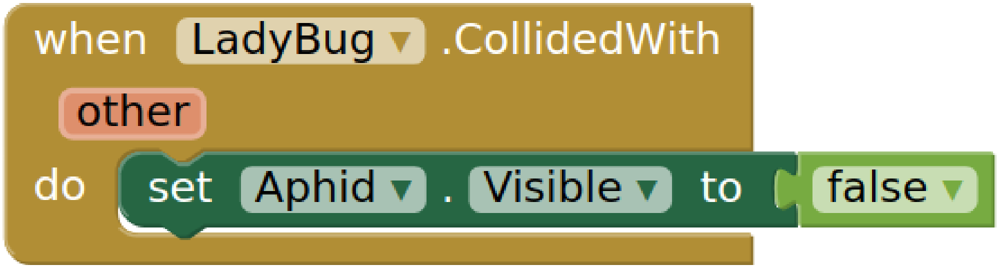
Getting Ready

You will need the following components in your design screen:

* Canvas
* Sprite
* Button

Blocks Editor

HINT: To make your ladybug move by clicking Buttons, check out the Movement cards.



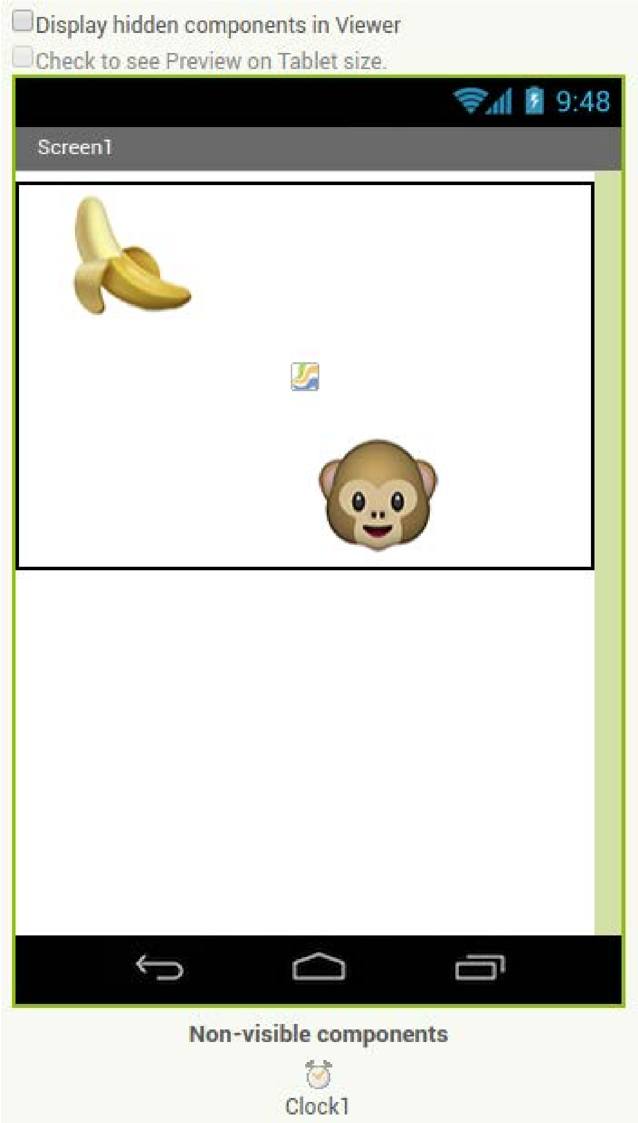
```
when LadyBug .CollidedWith
  other
do  set Aphid . Visible to  false
```

What does it mean?

The **LadyBugCollidedWith** event is Triggered when the Ladybug touches the Aphid. Then this will make the Aphid disappear.

Drag A Sprite

Move a sprite side to side by dragging your finger



□ Display hidden components in Viewer

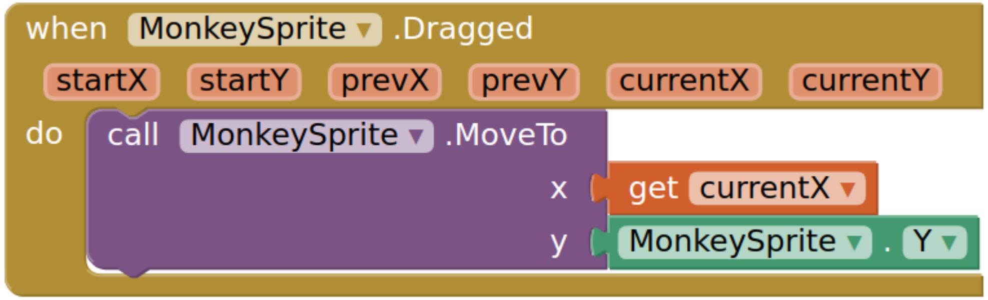
□ Check to see Preview on Tablet size.

Getting Ready

You will need the following components in your design screen:

* Canvas, ImageSprite, Clock

Blocks Editor



```
when MonkeySprite .Dragged
  startX  startY  prevX  prevY  currentX  currentY
do  call MonkeySprite .MoveTo
                          x  get currentX
                          y  MonkeySprite . Y
```

What does it mean?

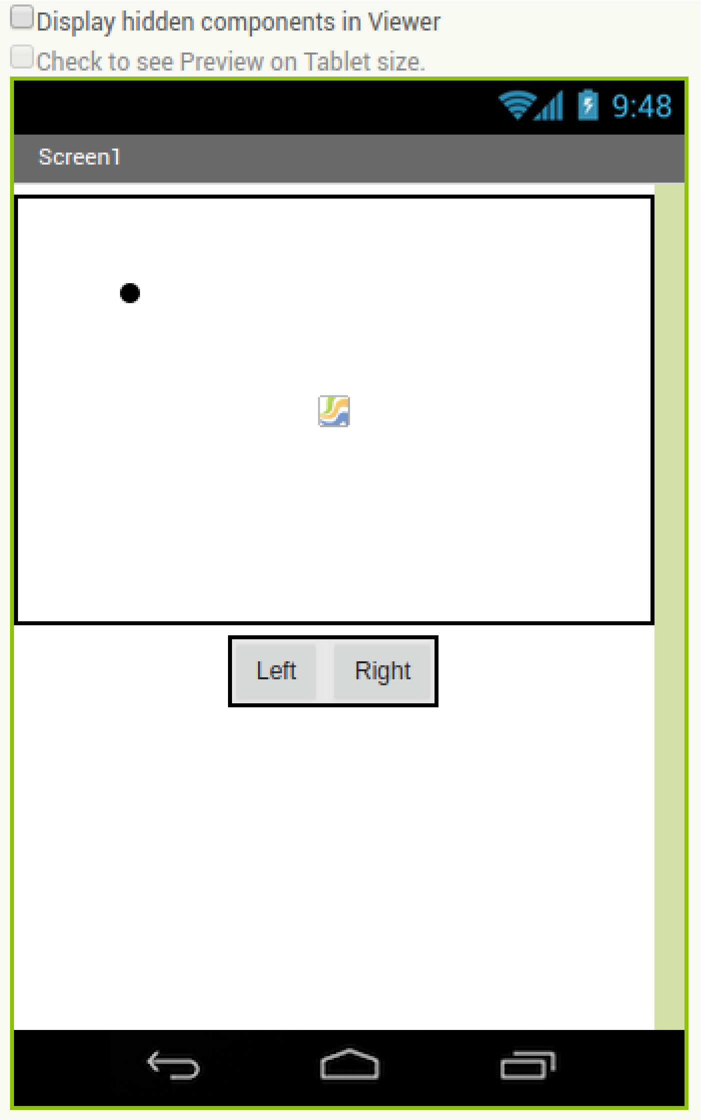
While the user is dragging the sprite, **MonkeySprite.Dragged** gets called multiple times. Each call has 6 arguments:

- **startX** and **startY**, where the user initially touched the screen.
- **currentX** and **currentY**, where the user is currently touching
- **prevX** and **prevY** hold whatever values were in **currentX** and **currentY** on the previous call to the event. (On the first call of this event, prevX and prevY are the same as startX and startY.)

When the user drags the MonkeySprite, it will be moved to the new X location, **currentX**, of the drag. The Y stays the same so the monkey can only move in the X-direction.

Movement with Buttons

Move a sprite by touching a button.



□ Display hidden components in Viewer

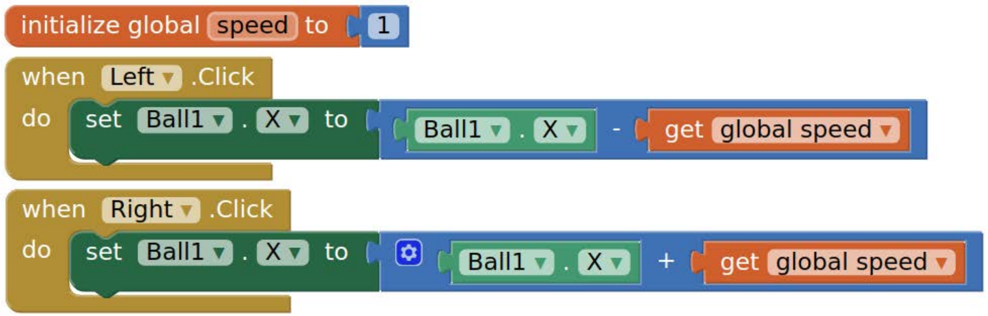
□ Check to see Preview on Tablet size.

Getting Ready

You will need the following components in your design screen:

* Canvas, Sprite, Button

Blocks Editor



```
initialize global speed to 1

when Left .Click
do  set Ball1 . X to Ball1 . X - get global speed

when Right .Click
do  set Ball1 . X to Ball1 . X + get global speed
```

What does it mean?

Define a **speed** variable to 1 to set the how far the sprite will move each time the button is clicked.

The **Left.Click** event moves the ball to the left every time the button is touched.
The **Right.Click** event moves the ball to the right every time the button is touched.