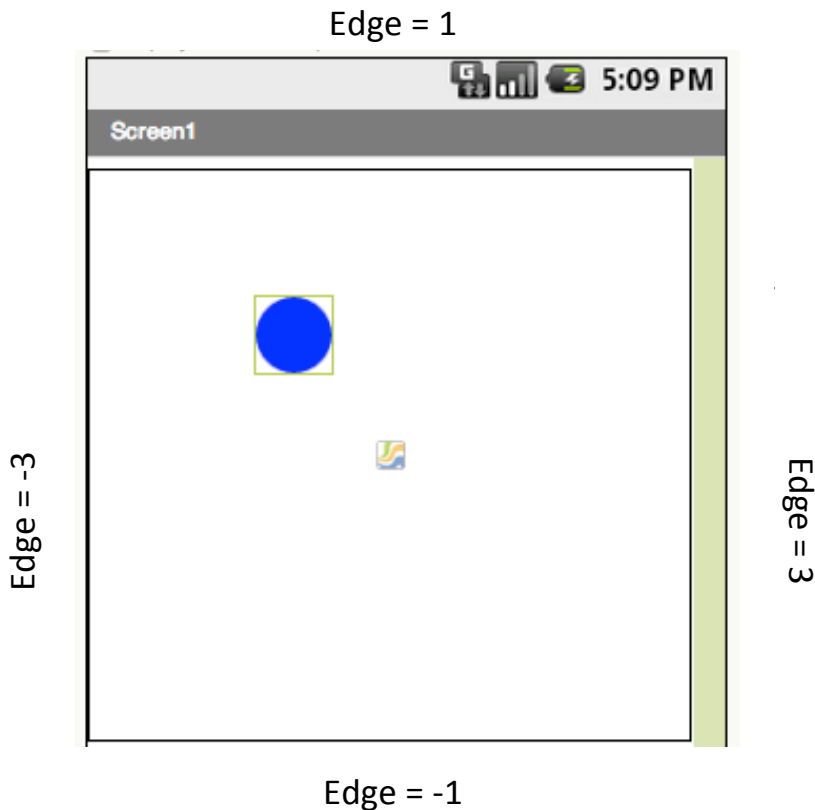




Bounce Sprite Off Canvas Edge

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



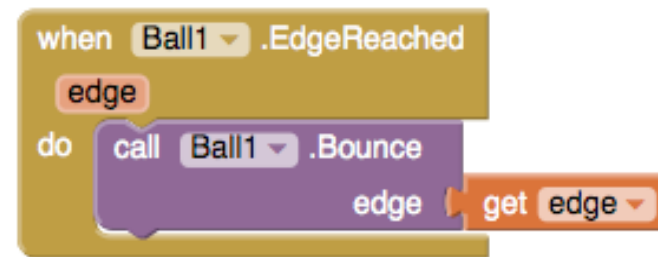
*Edges are assigned values (1, 3, -1, -3)
For top, right, bottom, left*

Getting Ready

You will need these components in your design screen:

- **Canvas**
- **Sprite**

Blocks Editor



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°).





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.

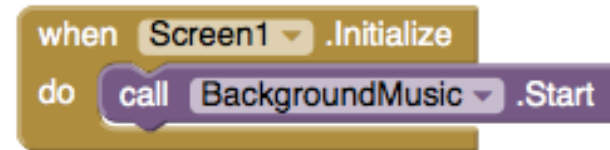


Get Ready

You will need these components in your design screen:

- **Button**
- **Sound**
- **Player**

Try These Blocks



What Does it Mean?

When **Button1** is clicked the **Meow** sound will play.

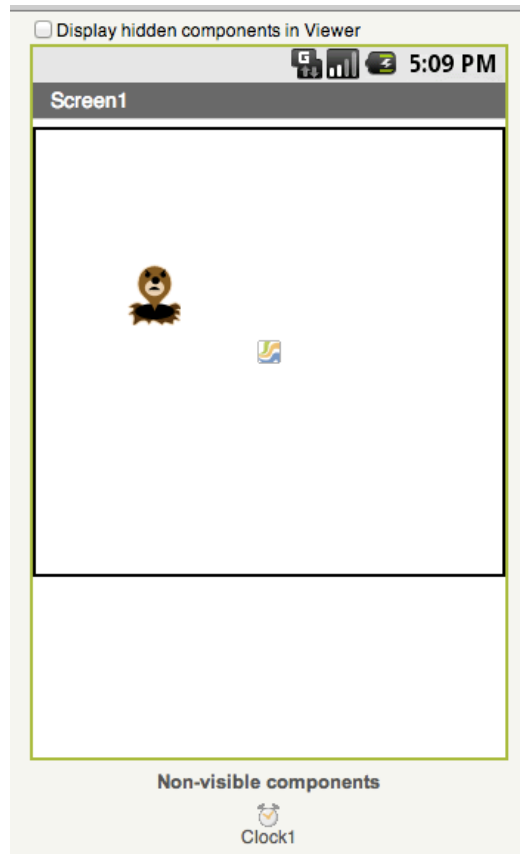
When **Screen1** comes up on your phone the **BackgroundMusic** will start playing.





Movement on a Timer

Move an ImageSprite with the passing of time.

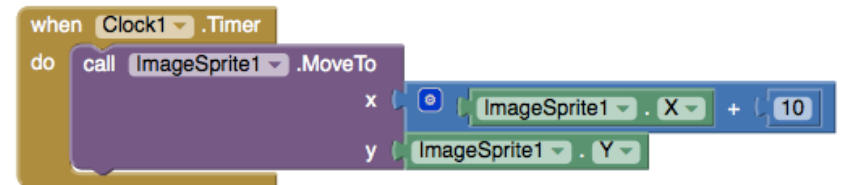


Get Ready

You will need these components in your design screen:

- **Canvas**
- **ImageSprite**
- **Clock**

Try These Blocks



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 **Clock 1.Timer** fires the Sprite will move to the right 10 pixels since the Sprite's Interval is





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.



Get Ready

You will need these components in the design screen:

- **Canvas**
- **ImageSprite**
- **Clock**
- **Button**

Try These Blocks

```
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.

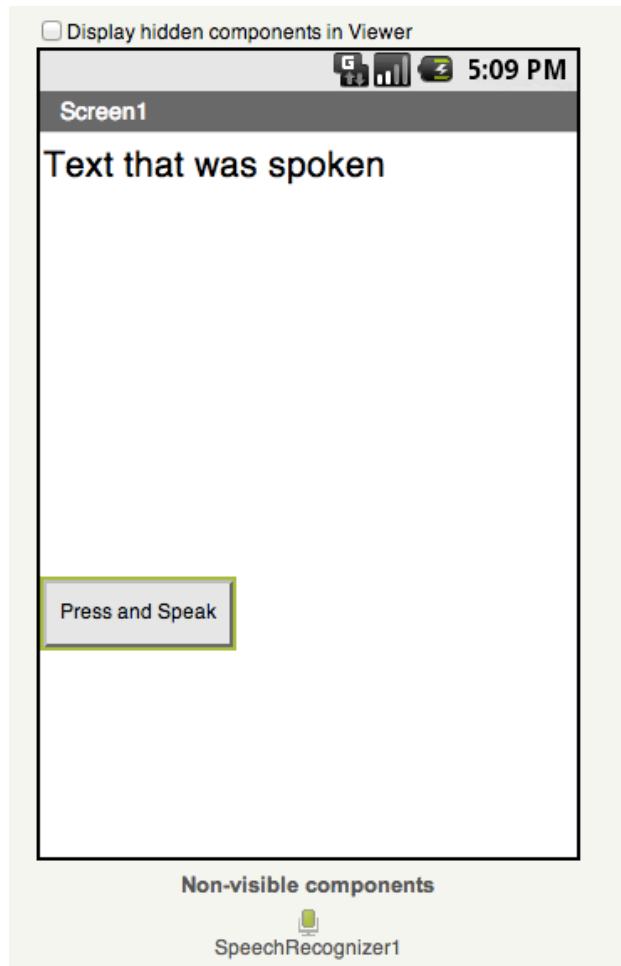
Now try making the object change also directions each time the button is clicked.





Speech Recognition

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



Get Ready

You will need these components in the Designer:

- **Label, Button, & SpeechRecognizer**

Try These Blocks

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

```
when SpeechRecognizer1 .AfterGettingText
result
do set TextLabel . Text to SpeechRecognizer1 . Result
```

```
when SpeechRecognizer1 .BeforeGettingText
do set TextLabel . 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.

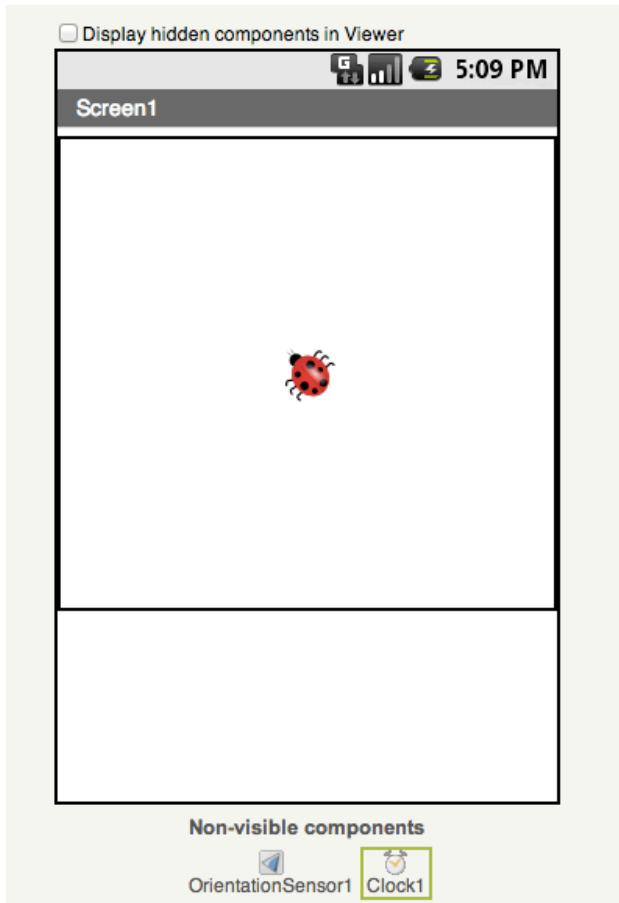
The **AfterGettingText** event will be triggered once speech has been received and recognized. Then the **Label** will display the text on the screen.





Movement with Sensors

Move an ImageSprite by tilting your phone

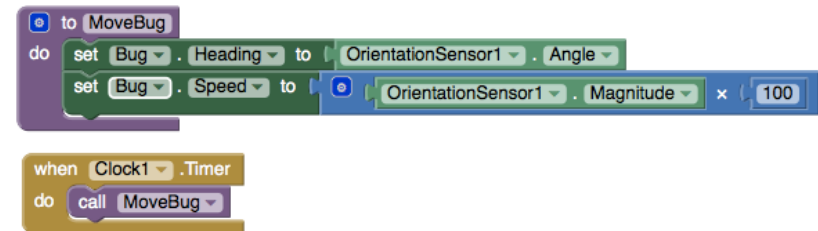


Get Ready

You will need these components in your design screen:

- **Canvas, ImageSprite, OrientationSensor, Clock**

Try These Blocks



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.

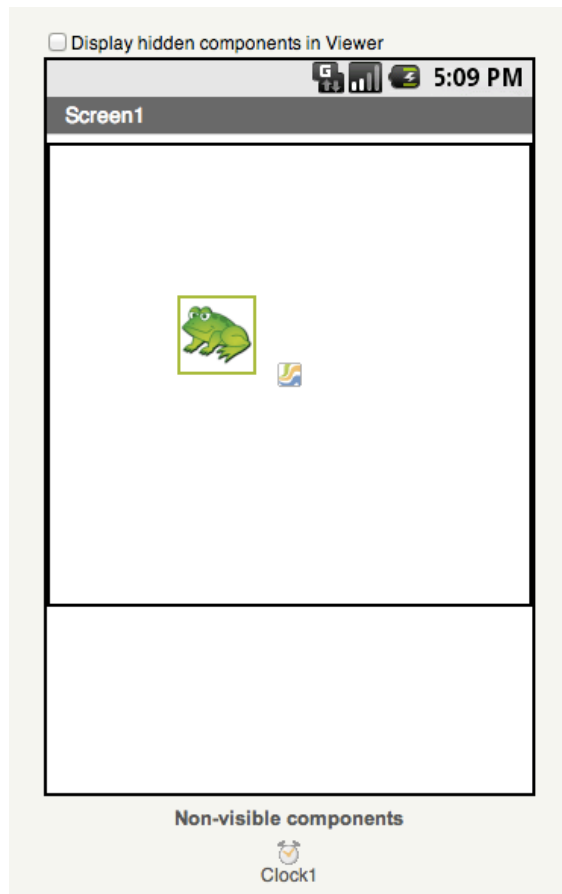
Whenever the **Clock1.Timer** fires, the event **MoveBug** will be called.





Random Numbers

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

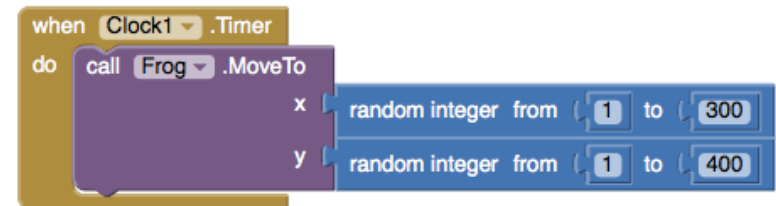


Get Ready

You will need these components in your design screen:

- **Canvas**
- **ImageSprite**
- **Clock**

Try These Blocks



What Does it Mean?

When the **Clock1.Timer** event is triggered, then **Frog.MoveTo** 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.

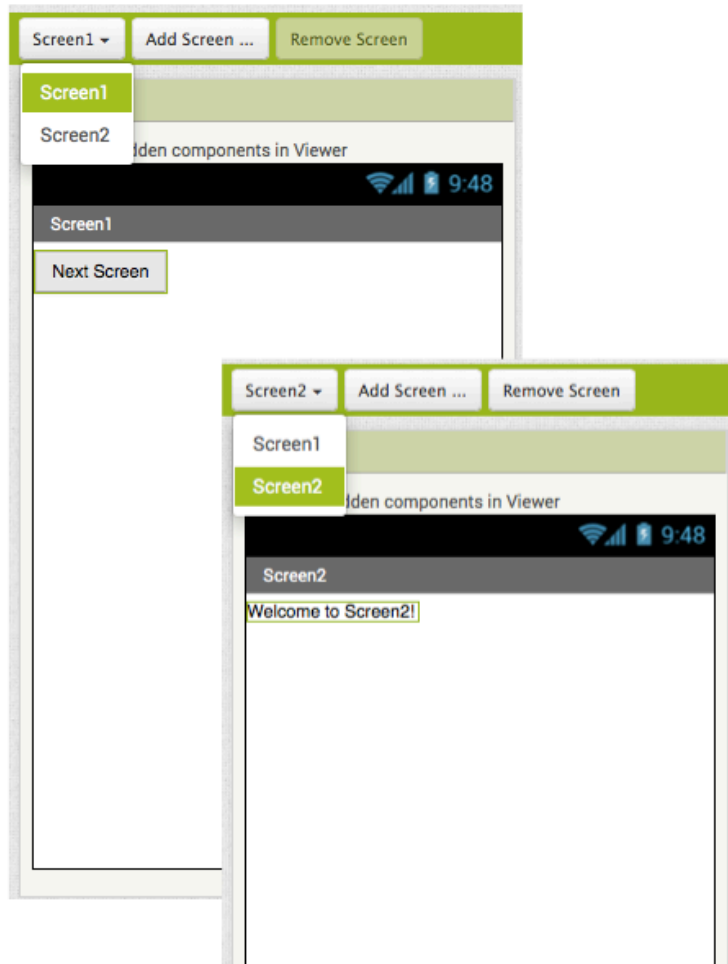
How can you use this for a game?





Multiple Screens

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



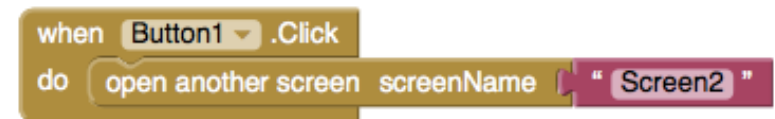
Getting Ready

You will need these components in your design screens:

- **Screen1 : Button**
- **Screen2: Label**

Blocks Editor

(For Screen1)



What Does it Mean?

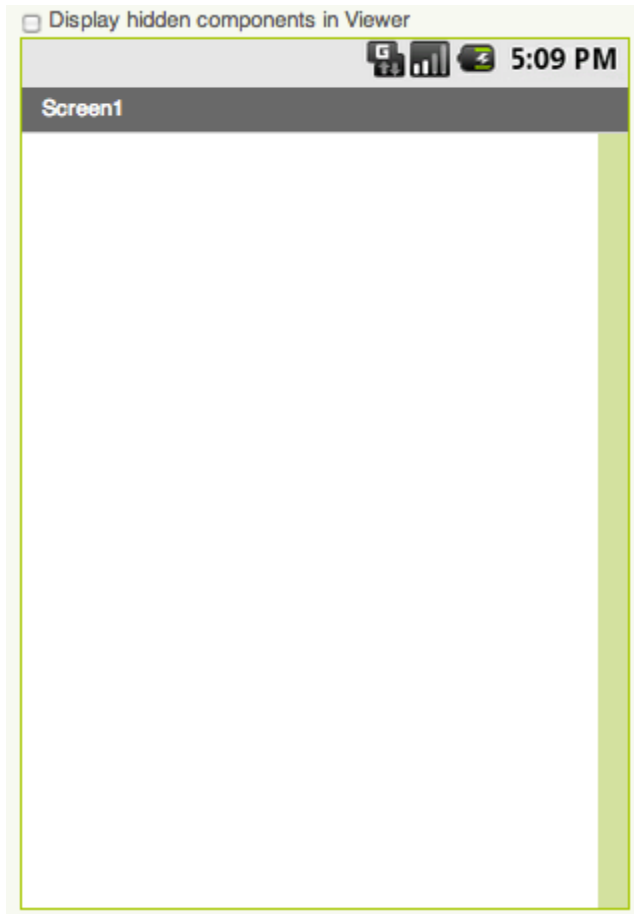
Open another screen takes in a text block. The text inside this block is the name of another screen. When the button is clicked, Screen2 will be opened.





Creating Your Own Color

Create your own colors using the make a color block.



Getting Ready

No components are necessary to use make color.

Blocks Editor



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.

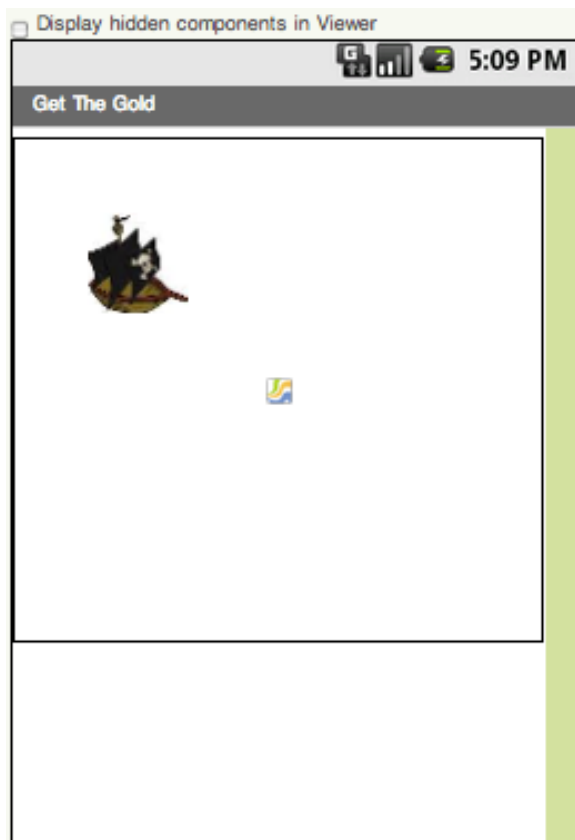
Can you make a Turquoise color?





Fling Movement

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

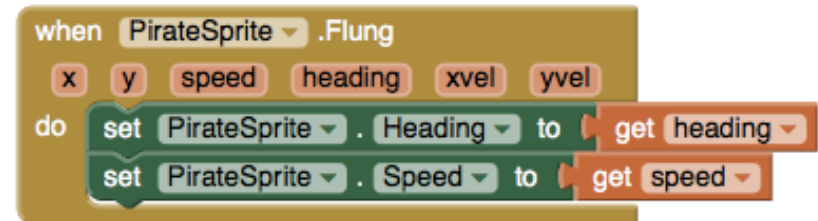


Getting Ready

You will need these components in the Designer:

- **Canvas**
- **ImageSprite**

Blocks Editor



What Does it Mean?

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

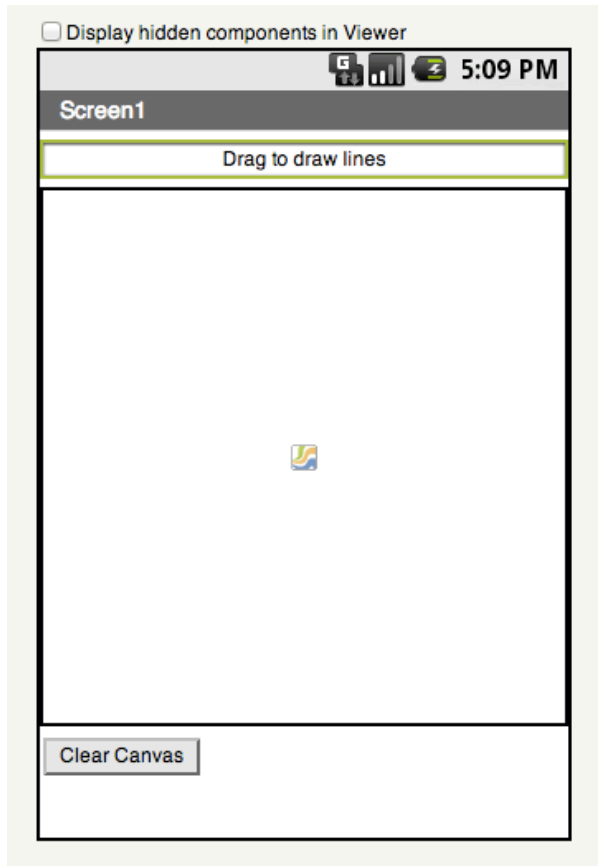
The user sets the heading and speed from the fling to the PirateSprite's **Heading** and **Speed**.





Drawing on a Canvas

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

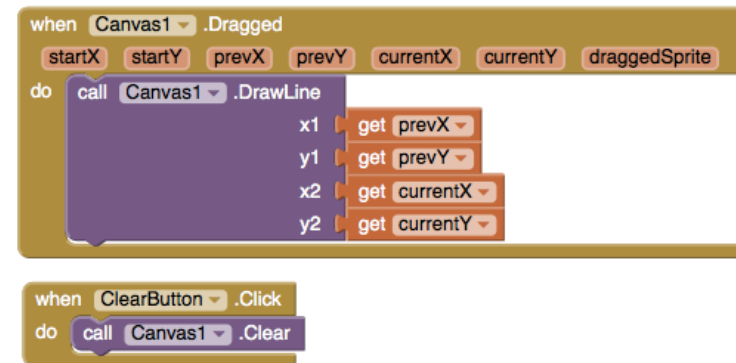


Get Ready

You will need these components in your design screen:

- **Canvas**
- **Button**
- **Label**

Try These Blocks



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.

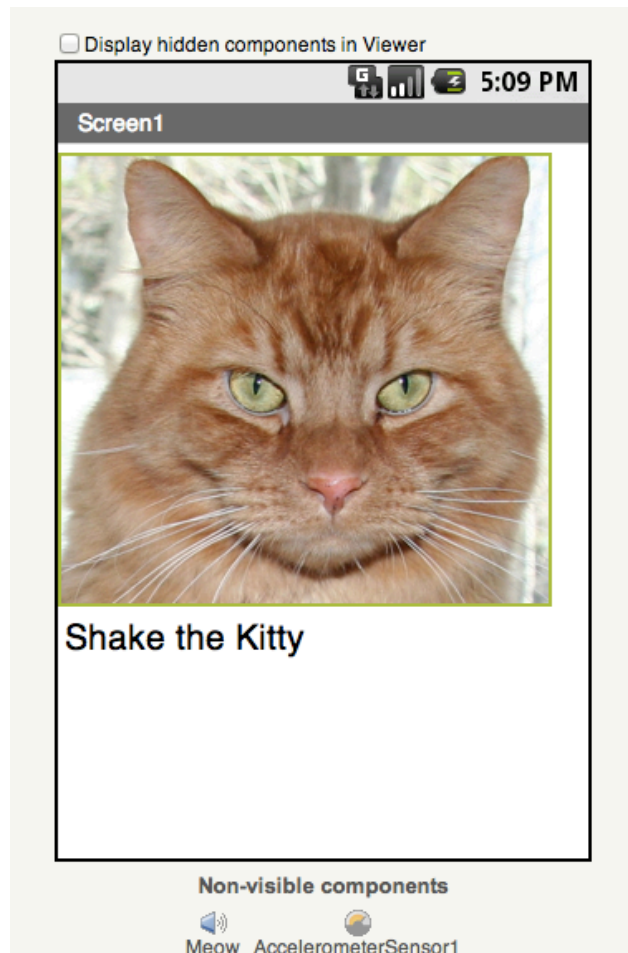
When the **Clear.Click** button is touched the canvas will be cleared.





Shaking Phone

Make something to happen when you shake your phone.

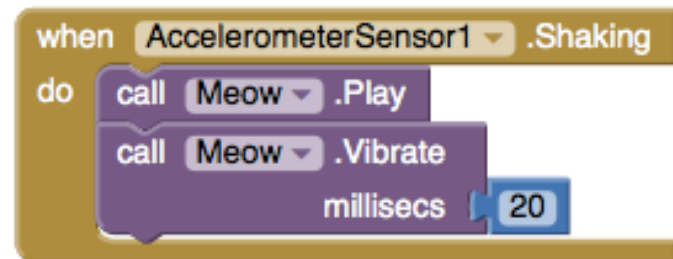


Get Ready

You will need these components in your design screen:

- **Image**
- **Sound**
- **AccelerometerSensor**
- **Label**

Try These Blocks



What Does it Mean?

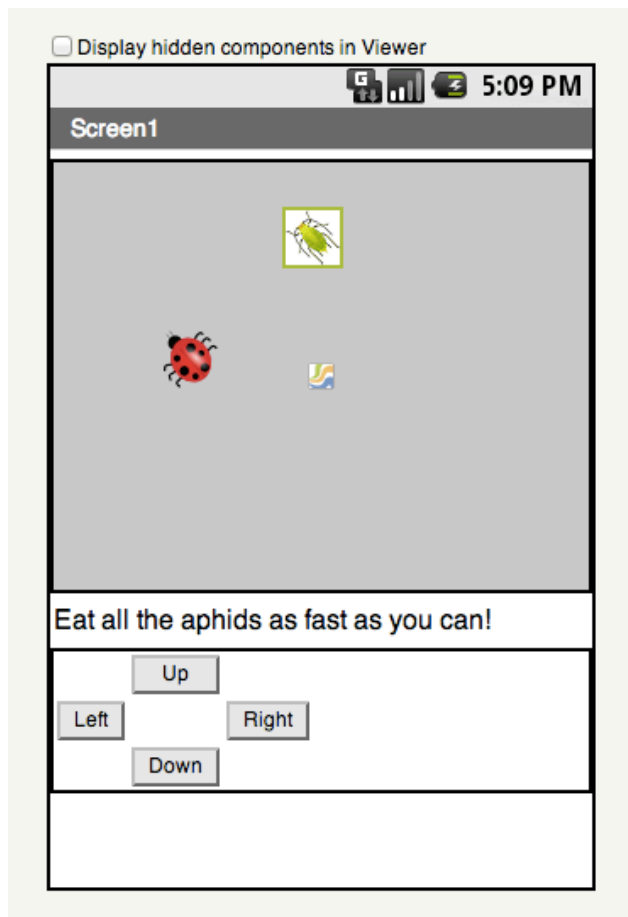
The **AccelerometerSensor.Shaking** event will detect when the phone is shaking and then the Meow sound will play and the phone will vibrate for 20 milliseconds.





Collision Detection

Make something happen when one Sprite collides with another.



Get Ready

You will need these components in your design screen:

- **Canvas**
- **Sprite**
- **Button**

Try These Blocks

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



What Does it Mean?

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

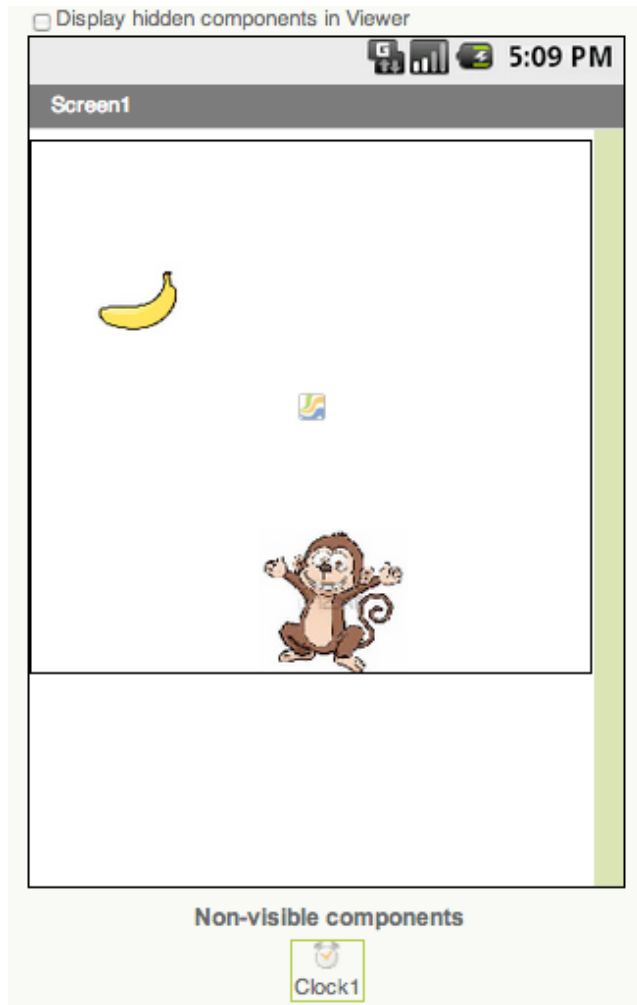
Can you add multiple aphids? How about adding a sound every time the ladybug eats an aphid?





Drag A Sprite

Move a sprite side to side by dragging your finger



Getting Ready

You will need these 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.)

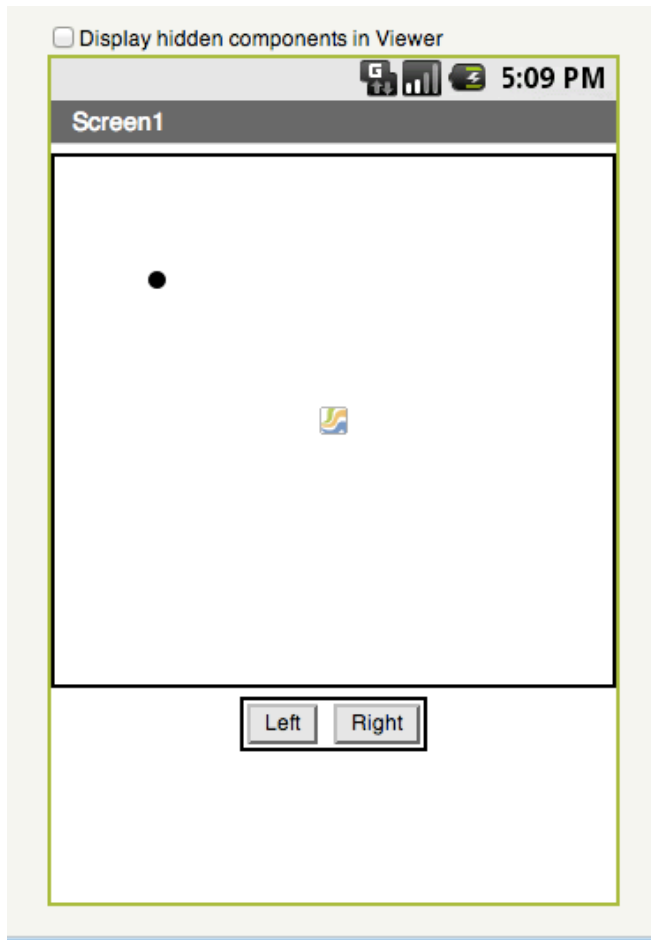
When the user drags the MonkeySprite, it will be moved to the new X location, **currentX**, of the drag. The Y stays the same so the monkey can only move in the X-direction.





Movement with Buttons

Move a sprite by touching a button.



Get Ready

You will need these components in the design screen:

- **Canvas**
- **Sprite**
- **Button**

Try These Blocks

```
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.

Can you add buttons to make the ball move up and down?

