

APP Design (Day 3)

- Review
- Problem Solving
- Detecting motion
- Sending SMS
- Inter-app communications
- Map
- Mini-Project



MIT
APP INVENTOR

Review

- What do we use variables for?

The screenshot displays a programming environment with two main panes: 'Blocks' and 'Viewer'. The 'Blocks' pane on the left shows a category menu with 'Variables' selected. The 'Viewer' pane on the right shows a script area with several variable-related blocks: 'initialize global name to', 'get', 'set to', 'initialize local name to in', and another 'initialize local name to in'. A separate 'initialize global name to' block is also shown to the right of the script area.

Review

- The score is a value that can change
- In computer programming, we store such values in a **variable**



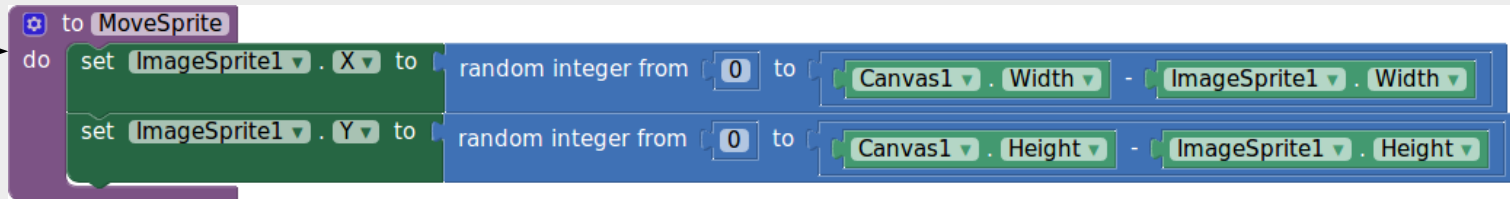
Variables

Nearly all programming languages have variables.

Variables can store numbers, text, boolean, and many more.

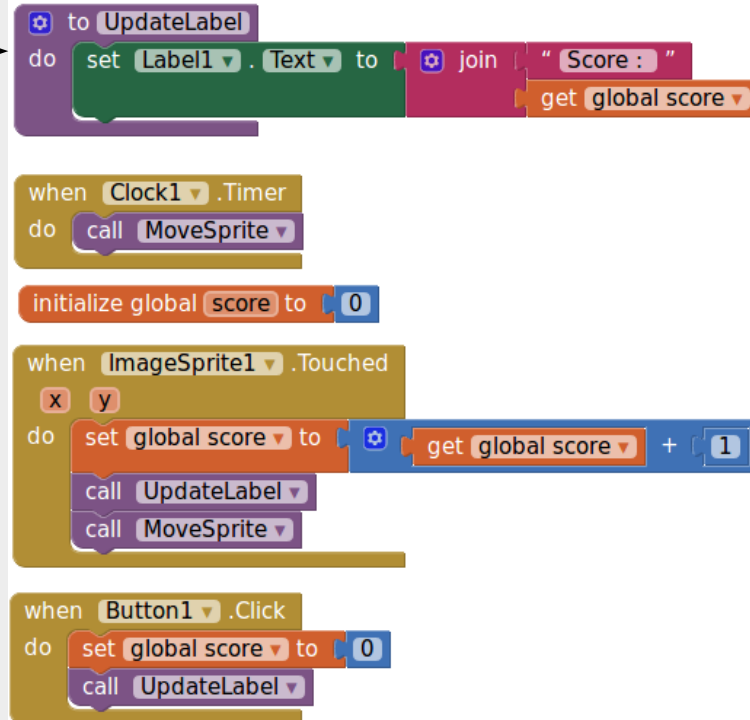
Review

What is this?



```
to MoveSprite
do
  set ImageSprite1 . X to random integer from 0 to Canvas1 . Width - ImageSprite1 . Width
  set ImageSprite1 . Y to random integer from 0 to Canvas1 . Height - ImageSprite1 . Height
```

Or this?



```
to UpdateLabel
do
  set Label1 . Text to join " Score : "
  get global score

when Clock1 . Timer
do
  call MoveSprite

initialize global score to 0

when ImageSprite1 . Touched
x y
do
  set global score to get global score + 1
  call UpdateLabel
  call MoveSprite

when Button1 . Click
do
  set global score to 0
  call UpdateLabel
```

Review

This is a function...

```
to MoveSprite
do
  set ImageSprite1 . X to random integer from 0 to Canvas1 . Width - ImageSprite1 . Width
  set ImageSprite1 . Y to random integer from 0 to Canvas1 . Height - ImageSprite1 . Height
```

This one too...

```
to UpdateLabel
do
  set Label1 . Text to
```

Calling of a function

```
when Clock1 . Timer
do
  call MoveSprite

initialize global score to 0

when ImageSprite1 . Touched
  x y
do
  set global score to
  call UpdateLabel
  call MoveSprite

when Button1 . Click
do
  set global score to
  call UpdateLabel
```

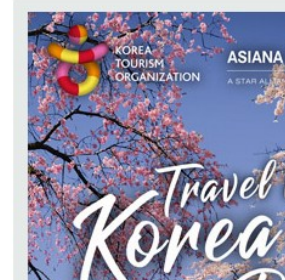
- Functions (aka. Procedures or Subroutines)
 - Separated from the rest of the code
 - Does something
 - Optionally accept inputs or provide outputs
 - Can be called by other pieces of code
- Minimize repetition
- Keeps code organized

Can these be prevented?

Man dies after suffering heart attack while jogging in Macritchie



A jogger in his 50s was found lying motionless along the MacRitchie Nature Trail near Singapore Island Country Club. PHOTO: S



Steve Cram's brother dies while jogging

by CLARE KITCHEN, Daily Mail

The brother of athletics hero Steve Cram collapsed and died while out jogging, it has emerged. He was 39

Kevin Cram, 39, a fitness fanatic, was running alone near his home in Cardiff when he suffered a suspected heart attack.

He was found slumped by the roadside on Tuesday evening by a passer-by and taken to hospital.

Solving the problem

- 1) Problem Selection
(I've done this for you...)
- 2) Understand the problem
(Read, Research, Talk to experts...)
- 3) Define the problem
- 4) Ideate (...think of a solution)
- 5) Prototype
- 6) Test!

Time to Work

- Understand the problem
 - Search online about similar cases
 - Learn about heart attacks and how to prevent or treat them
- Define the problem
 - You can't stop a heart attack, so it's not meaningful to say that "heart attacks" are the problem
 - What then is the problem?
- **You have 10 mins to work on this**

Understand the Problem

- Best time to treat a heart attack is within one to two hours of the first onset of symptoms
- 90% to 95% of heart attack victims who reach the hospital survive
- Victims often discovered late

Define the Problem

- Heart attack victims are not found until it is too late
- Especially for joggers in isolated areas
- Need a way to notify others that a heart attack (...or other emergencies) has occurred

Ideate

- Create an app that...
 - Detects when the user is not moving for sometime
 - Notify others of your location
 - Display location on a map

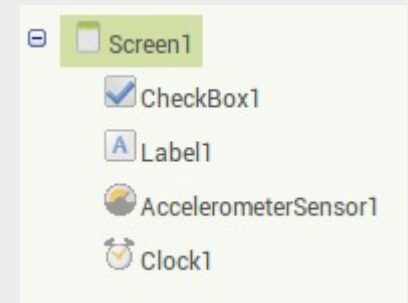
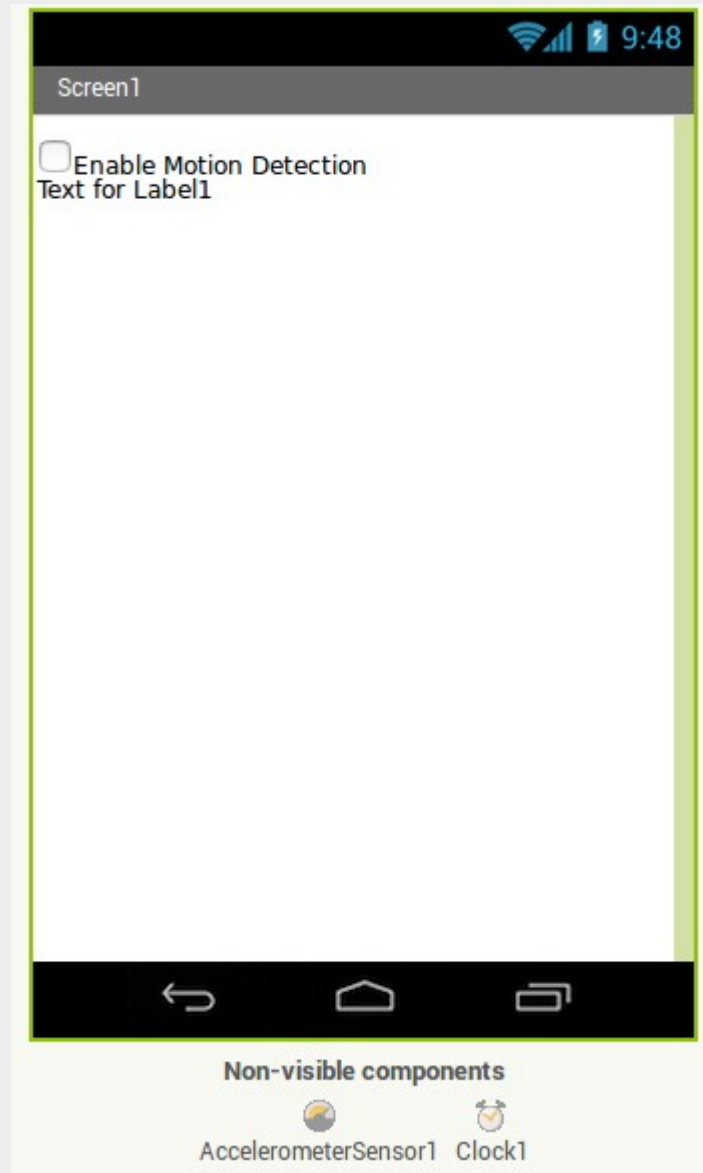
Prototype

- When prototyping, don't try to do everything at once
- Add features in gradually

Detect Motion

Add in...

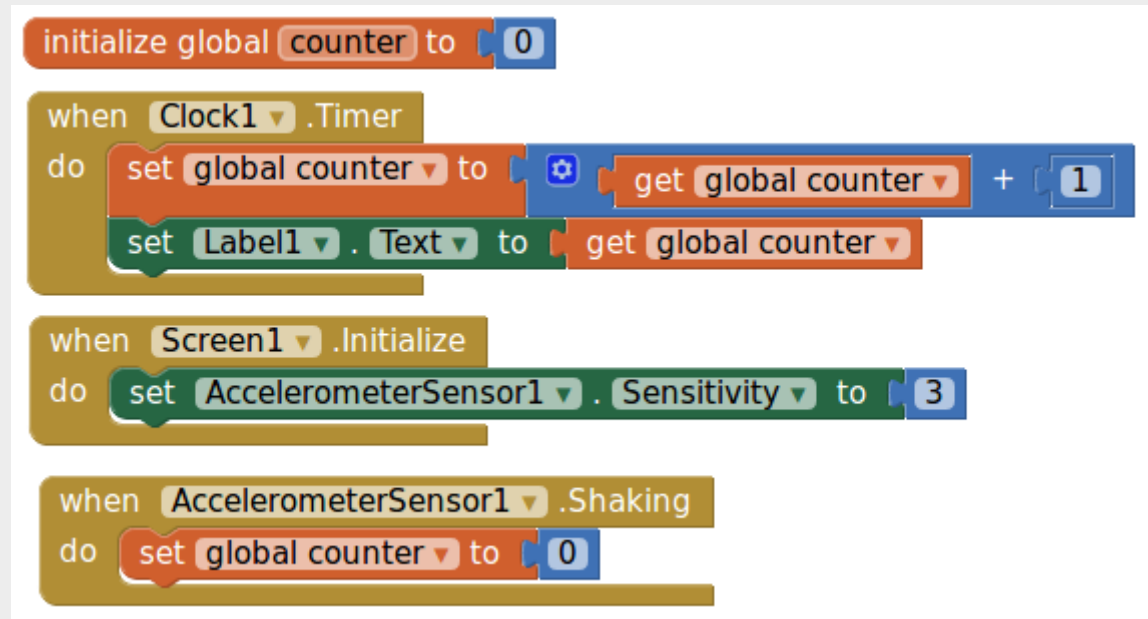
- Checkbox (enable or disable motion detection)
- Label (for testing)
- Accelerometer (Detects motion)
- Clock (Measure time)



Detect Motion

In the blocks editor...

- Use a **variable** to track how many seconds has passed and display on screen
- Reset to zero when the phone is shaken
- Set sensitivity to 3 (high)



```
initialize global counter to 0

when Clock1.Timer
do
  set global counter to get global counter + 1
  set Label1.Text to get global counter

when Screen1.Initialize
do
  set AccelerometerSensor1.Sensitivity to 3

when AccelerometerSensor1.Shaking
do
  set global counter to 0
```

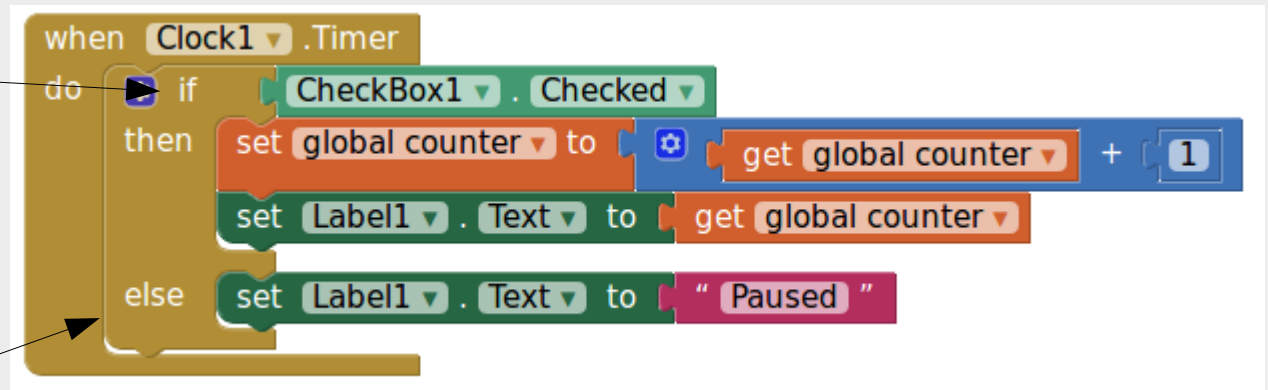
The image shows a sequence of code blocks in a Scratch-like editor. The first block is 'initialize global counter to 0'. The second block is a 'when Clock1.Timer' event block with a 'do' block containing two actions: 'set global counter to get global counter + 1' and 'set Label1.Text to get global counter'. The third block is a 'when Screen1.Initialize' event block with a 'do' block containing 'set AccelerometerSensor1.Sensitivity to 3'. The fourth block is a 'when AccelerometerSensor1.Shaking' event block with a 'do' block containing 'set global counter to 0'.

Challenge

- We haven't used the checkbox yet
- Change the program, so that the counter only increases if the checkbox is checked
- Hint: Use an "if" block

Solution

Only increase the counter if checkbox is checked...

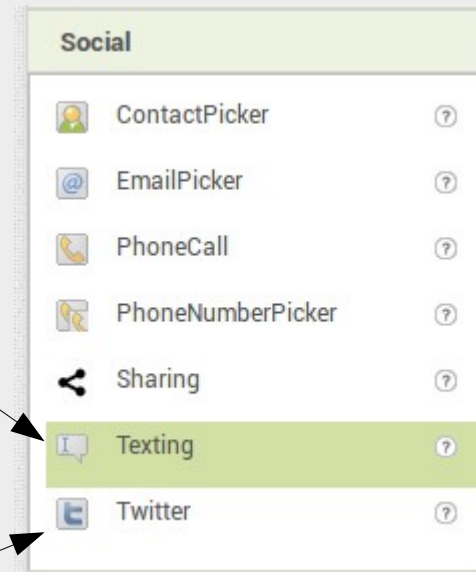


```
when Clock1.Timer
do
  if CheckBox1.Checked
  then
    set global counter to get global counter + 1
    set Label1.Text to get global counter
  else
    set Label1.Text to "Paused"
```

...else we'll set the text to "Paused".
(optional)

Calling for Help

Add in a “Texting” component. This is used for sending SMS.



...calling for help via Twitter... probably won't be as effective. But it's up to you.

Calling for Help

When the counter reaches 10...

...set the phone number and message, and send the SMS directly

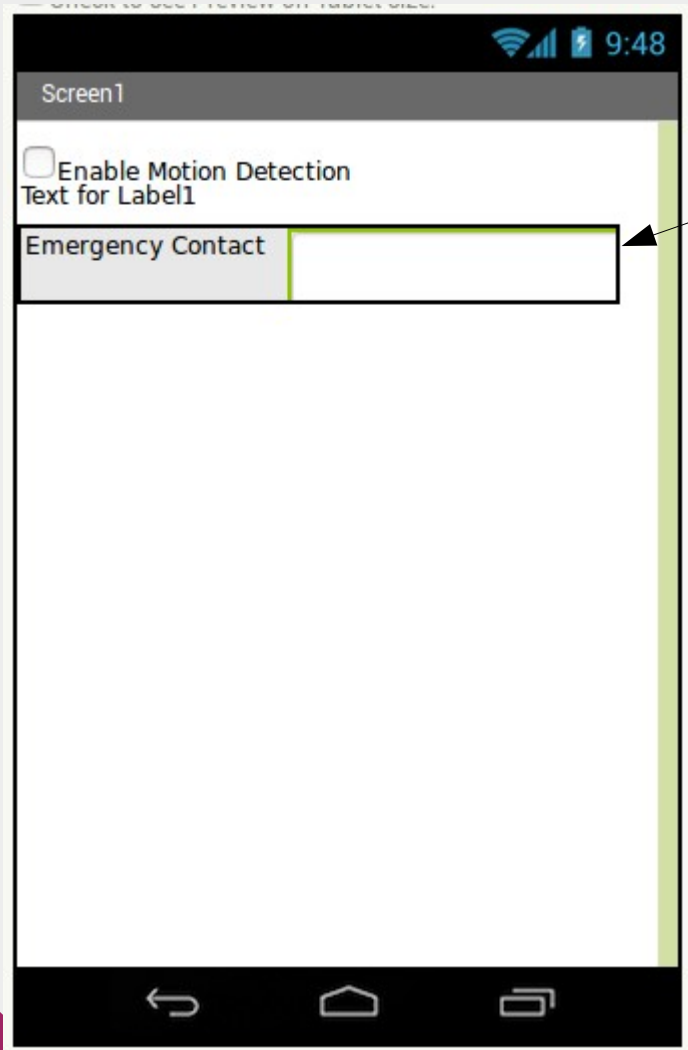
Note:
You may need to enable the SMS permission for App Inventor

```
when Clock1.Timer
do
  if CheckBox1.Checked
  then
    set global counter to get global counter + 1
    set Label1.Text to get global counter
    if get global counter = 10
    then
      set Texting1.PhoneNumber to [redacted]
      set Texting1.Message to "I need help!"
      call Texting1.SendMessageDirect
    else
      set Label1.Text to "Paused"
```

Challenge

- The phone number is fixed in the program
- Change the program to allow the user to set the phone number
- Optionally, allow the user to set their own message as well
- Hint: You'll need to add a textbox

Solution



Add a textbox...

...and use the textbox value to set the phone number

```
when Clock1.Timer
do
  if CheckBox1.Checked
  then
    set global counter to get global counter + 1
    set Label1.Text to get global counter
    if get global counter = 10
    then
      set Texting1.PhoneNumber to TextBox1.Text
      set Texting1.Message to "I need help!"
      call Texting1.SendMessageDirect
    else
      set Label1.Text to "Paused"
```

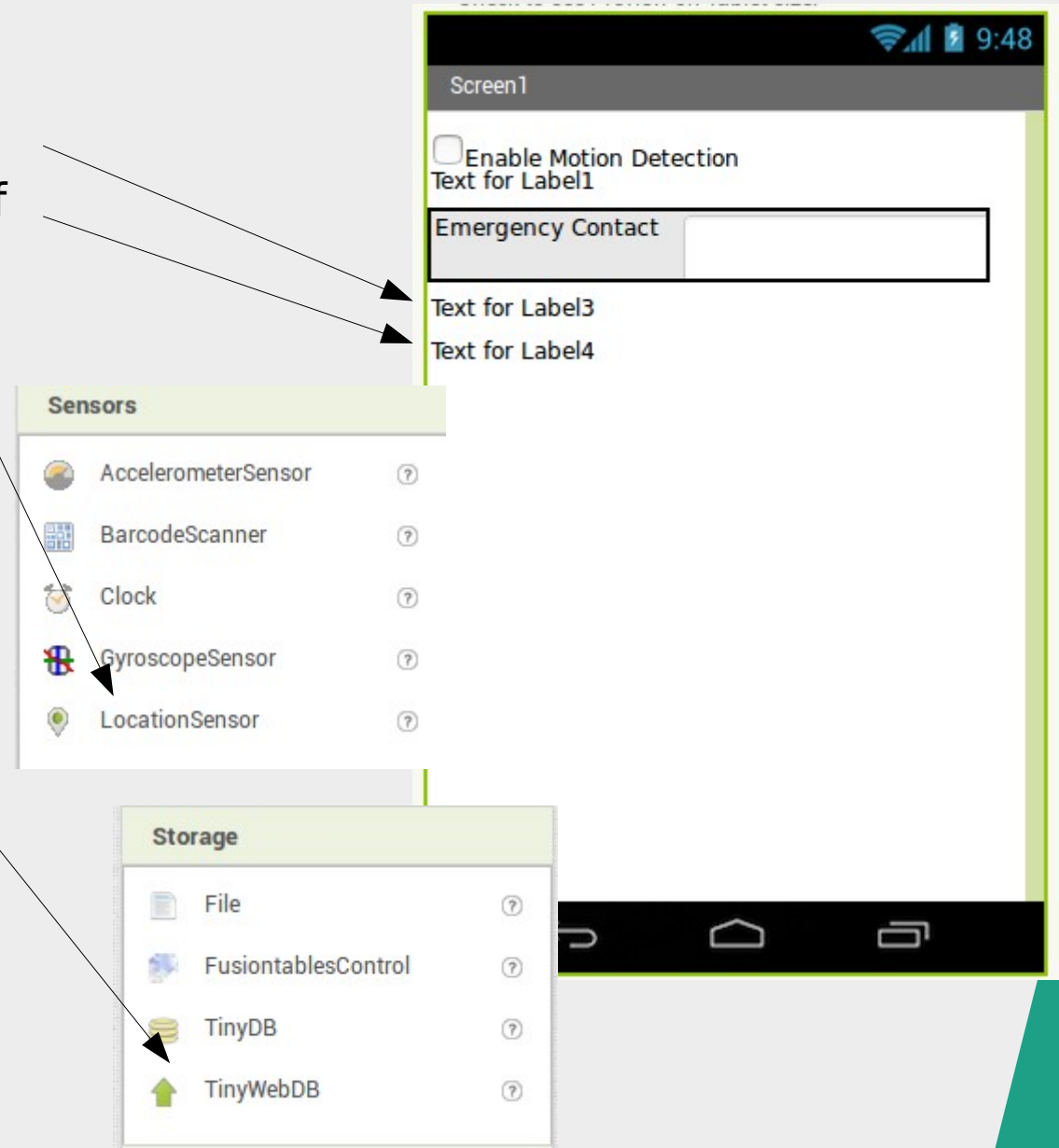
Upload Location

Add in two more labels.
We'll use it to display
latitude and longitude of
the victim.

Add in a "LocationSensor"
component (under sensors)

Add in a TinyWebDB
component (non-visible).

This is a shared storage.
Anything that you store in
this database is visible to
everyone. You can use this
to communicate between
different phones.



Upload Location

```
when Clock1.Timer
do
  if CheckBox1.Checked
  then
    set global counter to get global counter + 1
    set Label1.Text to get global counter
    if get global counter = 10
    then
      set Texting1.PhoneNumber to TextBox1.Text
      set Texting1.Message to "I need help"
      call Texting1.SendMessageDirect
      call TinyWebDB1.StoreValue
        tag "CortsLongitude"
        valueToStore LocationSensor1.Longitude
      call TinyWebDB1.StoreValue
        tag "CortsLatitude"
        valueToStore LocationSensor1.Latitude
    else
      set Label1.Text to "Paused"
```

Upload location to server

Make sure you have a unique tag!

Retrieve Location

```
when Clock2.Timer
do
  call TinyWebDB1.GetValue tag "CortsLongitude"
  call TinyWebDB1.GetValue tag "CortsLatitude"
```

Create a new timer and set it to 5 seconds.

Every 5 seconds, we'll request for the location using the same tag.

This is normally done using another phone, but we'll use the same phone for testing purposes.

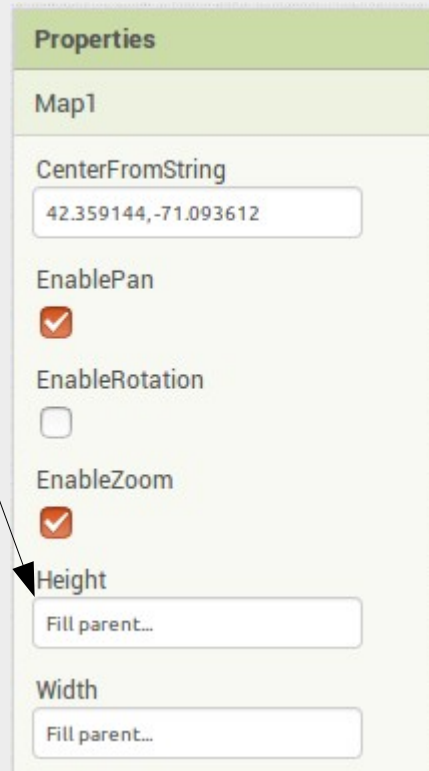
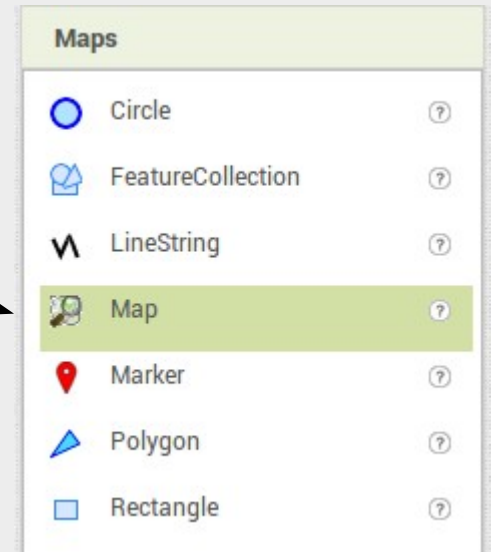
```
when TinyWebDB1.GotValue
  tagFromWebDB valueFromWebDB
do
  if get tagFromWebDB = "CortsLongitude"
  then set Label3.Text to get valueFromWebDB
  if get tagFromWebDB = "CortsLatitude"
  then set Label4.Text to get valueFromWebDB
```

When we got the data, we'll set it into the label

Display on Map

Add in a Map

Adjust the properties to make it as large or small as you like



Display on Map

Create two new variables

Save the longitude and latitude to these variables

Pan the map to the location (Play around with the zoom to see what it means)

```
initialize global longitude to 0
initialize global latitude to 0
when TinyWebDB1 GotValue
  tagFromWebDB valueFromWebDB
do
  if get tagFromWebDB = "CortsLongitude"
  then
    set global longitude to get valueFromWebDB
    set Label3 . Text to get valueFromWebDB
  if get tagFromWebDB = "CortsLatitude"
  then
    set global latitude to get valueFromWebDB
    set Label4 . Text to get valueFromWebDB
  call Map1 .PanTo
    latitude get global latitude
    longitude get global longitude
    zoom 16
```

Mini Project

Polishing up an App

- The app works, but is really rough and simple. Can you add in...
 - A manual “Call for help” button?
 - Only show the map if there’s a request for help?
 - Add a marker to the emergency location?
- Test the app thoroughly. There are many more bugs and shortcomings. See if you can find and fix them.