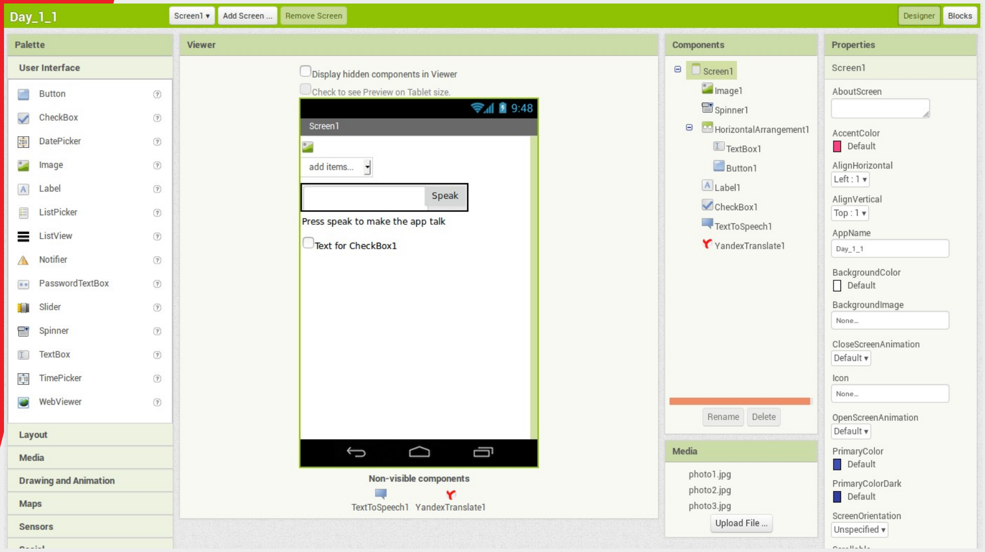
APP Design (Day 2)

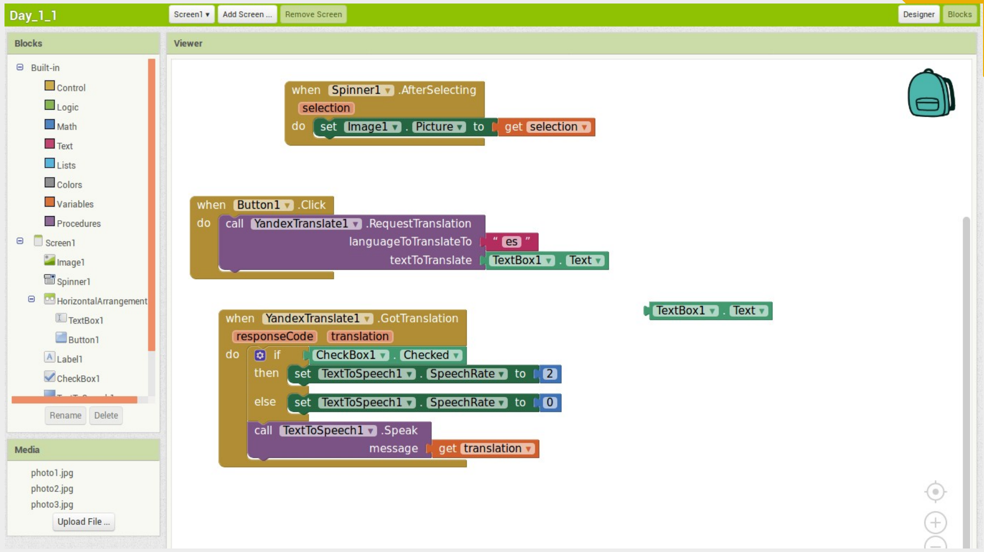
- Review
- Sprites and Graphics
- Variables
- Functions / Procedures
- Mini-Project

MIT
APP INVENTOR

Review



AboutScreen
AccentColor: Default
AlignHorizontal: Left : 1
AlignVertical: Top : 1
AppName: Day_1_1
BackgroundColor: Default
BackgroundImage: None...
CloseScreenAnimation: Default
Icon: None...
OpenScreenAnimation: Default
PrimaryColor: Default
PrimaryColorDark: Default
ScreenOrientation: Unspecified



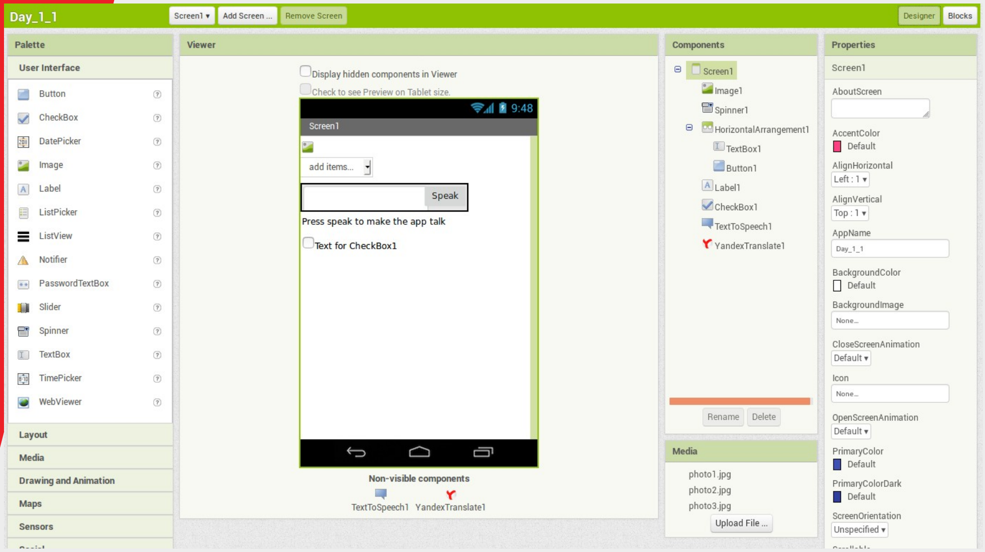
```
when Spinner1 .AfterSelecting
  selection
do  set Image1 . Picture to  get selection

when Button1 .Click
do  call YandexTranslate1 .RequestTranslation
              languageToTranslateTo  " es "
                    textToTranslate  TextBox1 . Text

when YandexTranslate1 .GotTranslation
  responseCode  translation
do  if    CheckBox1 . Checked
    then  set TextToSpeech1 . SpeechRate to 2
    else  set TextToSpeech1 . SpeechRate to 0
    call TextToSpeech1 .Speak
                 message  get translation

TextBox1 . Text
```

Designer

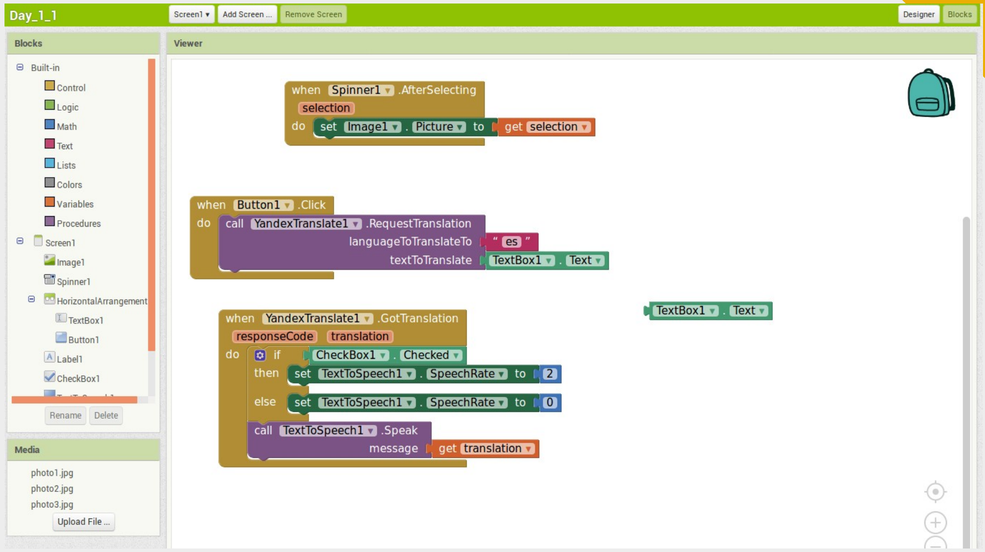
Blocks Editor

What are they for?

Review



Designer



```
when Spinner1 .AfterSelecting
  selection
do  set Image1 . Picture to  get selection

when Button1 .Click
do  call YandexTranslate1 .RequestTranslation
            languageToTranslateTo  " es "
            textToTranslate  TextBox1 . Text

when YandexTranslate1 .GotTranslation
  responseCode  translation
do  if  CheckBox1 . Checked
    then  set TextToSpeech1 . SpeechRate to 2
    else  set TextToSpeech1 . SpeechRate to 0
    call TextToSpeech1 .Speak
            message  get translation

TextBox1 . Text
```

Blocks Editor

- Add components
- Layout of components
- Component settings (eg. font size)

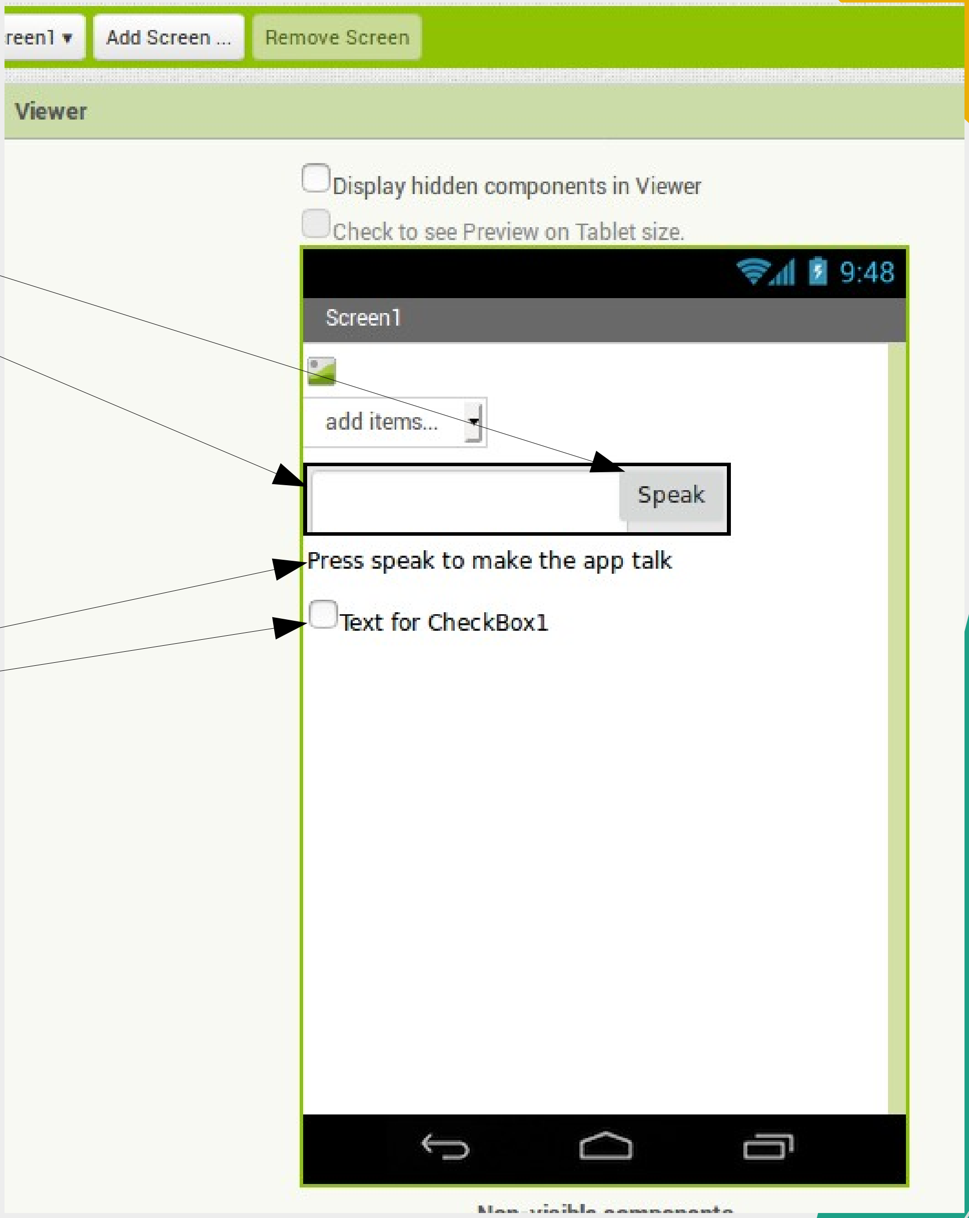
What are they for?

- Actions and behaviors (eg. what should the app do when the user click a button?)

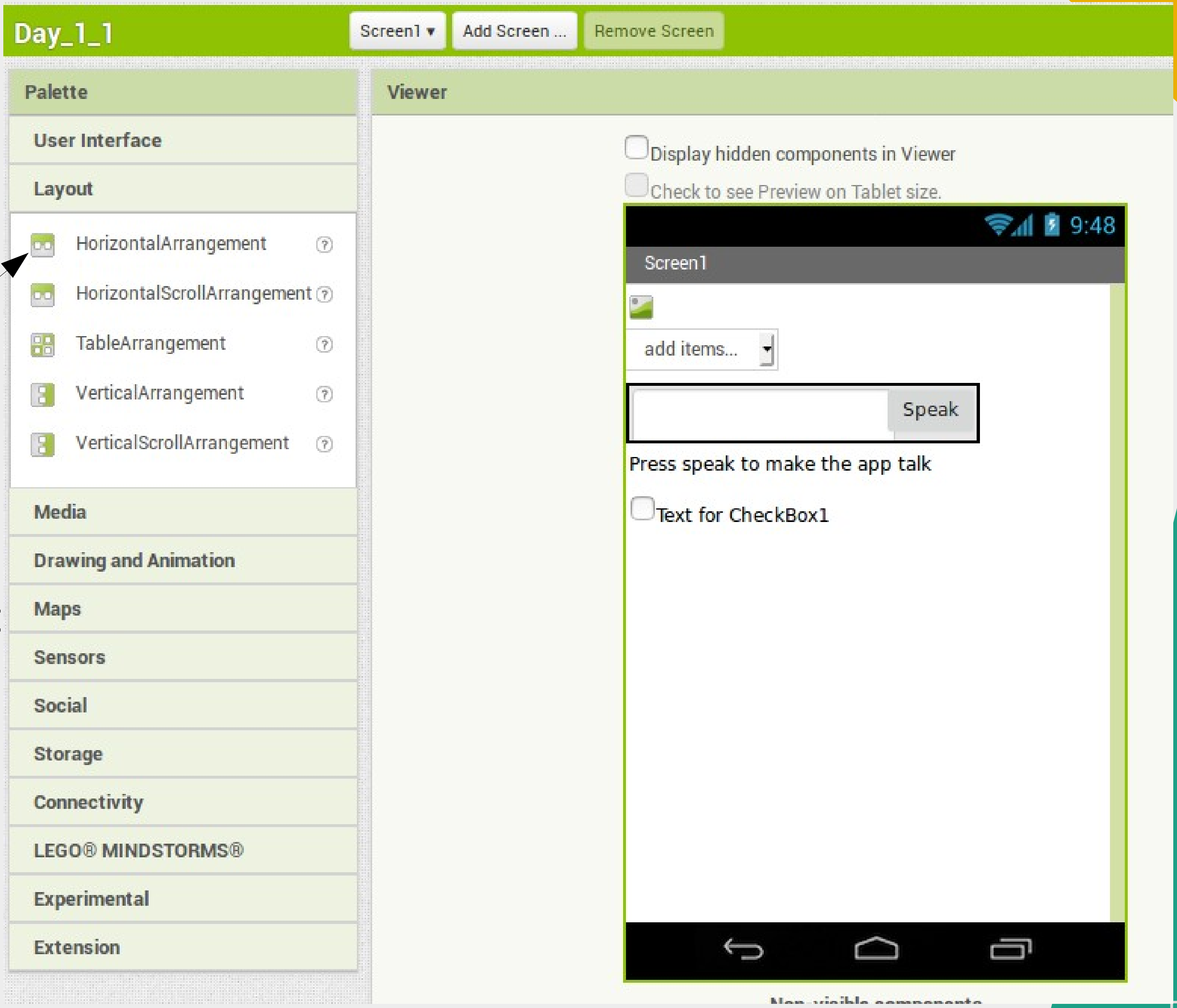
Review

How to place component horizontally...

...instead of vertically?


reen1 ▾
Add Screen ...
Remove Screen
Viewer
Display hidden components in Viewer
Check to see Preview on Tablet size.
9:48
Screen1
add items...
Speak
Press speak to make the app talk
Text for CheckBox1

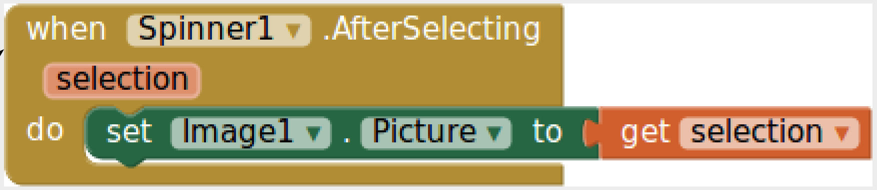
Review



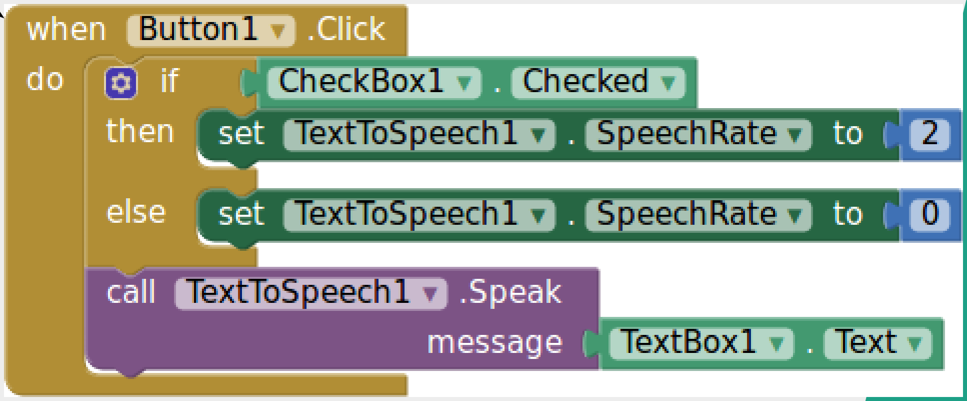
- Place a "Horizontal Arrangement" component
- Place other components inside the horizontal arrangement component

Review

What are these blocks that starts with "when"?



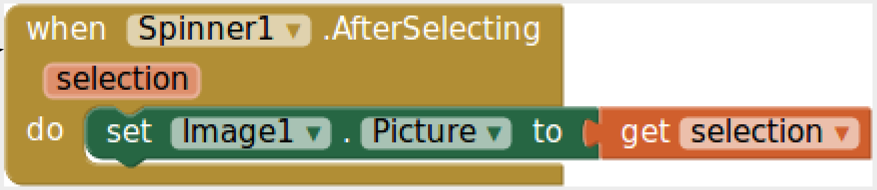
```
when Spinner1 .AfterSelecting
  selection
do  set Image1 . Picture to  get selection
```



```
when Button1 .Click
do  if  CheckBox1 . Checked
    then  set TextToSpeech1 . SpeechRate to 2
    else  set TextToSpeech1 . SpeechRate to 0
    call TextToSpeech1 .Speak
                 message  TextBox1 . Text
```

Review

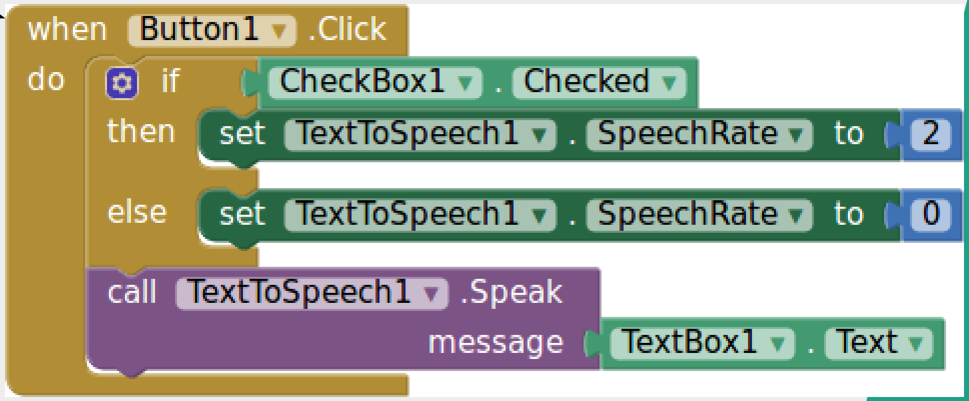
- App Inventor uses an “event driven” programming language
- “when” blocks are “events”
- Runs the code inside the block when the event happens



```
when Spinner1 .AfterSelecting
  selection
do  set Image1 . Picture to  get selection
```

“When” button1 is clicked...

...run this set of code

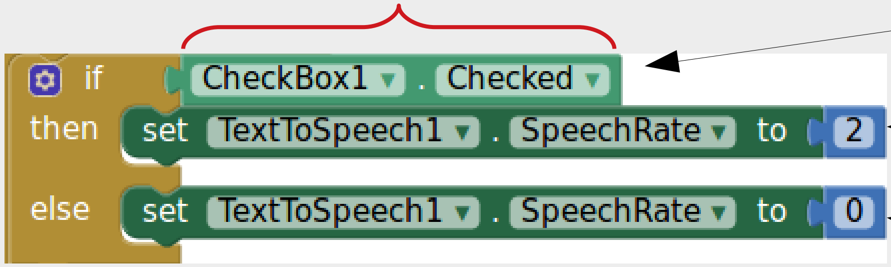


```
when Button1 .Click
do  if    CheckBox1 . Checked
    then  set TextToSpeech1 . SpeechRate to 2
    else  set TextToSpeech1 . SpeechRate to 0
    call TextToSpeech1 .Speak
                       message  TextBox1 . Text
```

Review

“if” statements

Condition

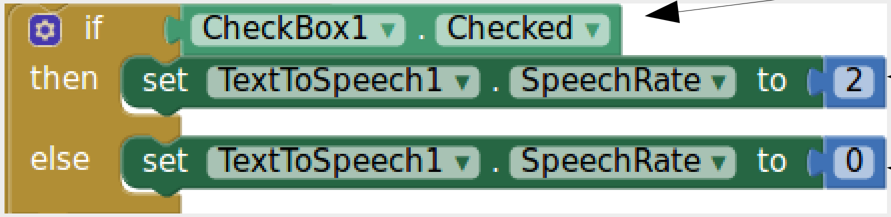


```
if   CheckBox1 . Checked
then set TextToSpeech1 . SpeechRate to 2
else set TextToSpeech1 . SpeechRate to 0
```

if condition is true...

...do this...

...else do this instead!



```
if   CheckBox1 . Checked
then set TextToSpeech1 . SpeechRate to 2
else set TextToSpeech1 . SpeechRate to 0
```

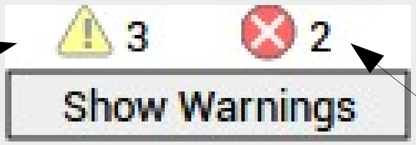
if checkbox 1 is “checked” (ticked)

...set SpeechRate to 2 (fast)

...else set SpeechRate to 0 (slow)

Review

Errors and Warnings

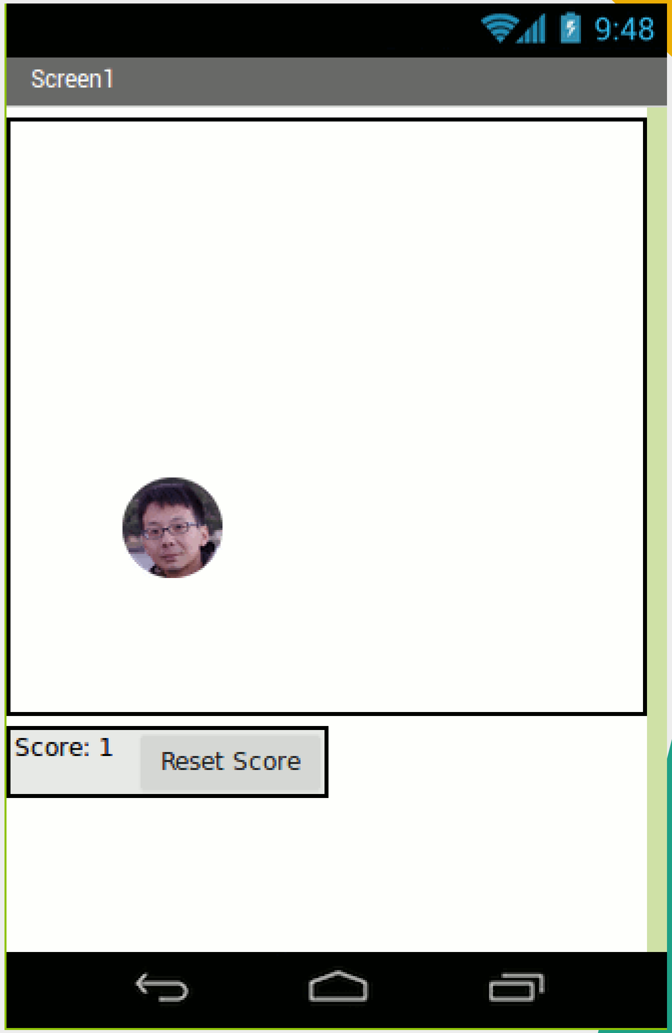


Warnings
- Something is **probably** wrong.
- Click "Show Warnings" to display a warning sign next to the possible problem
- You **should** fix this

Errors
- Something is **definitely** wrong.
- You **must** fix this

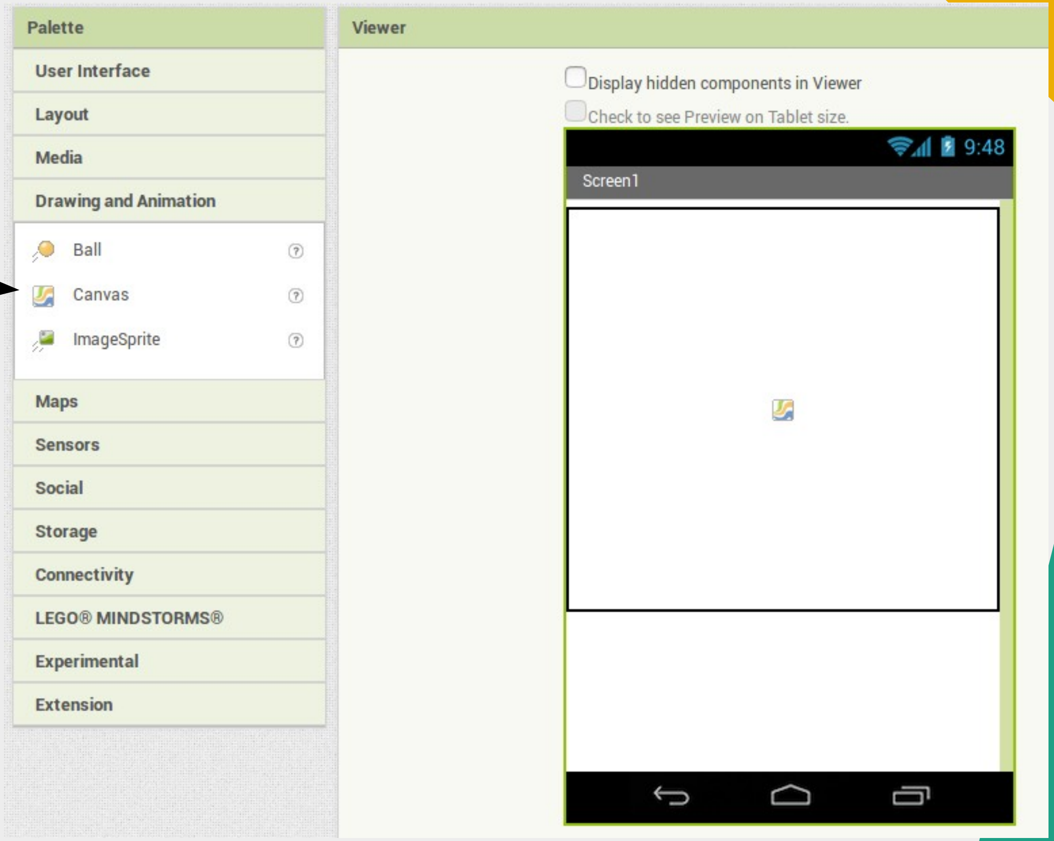
Whack the Mole

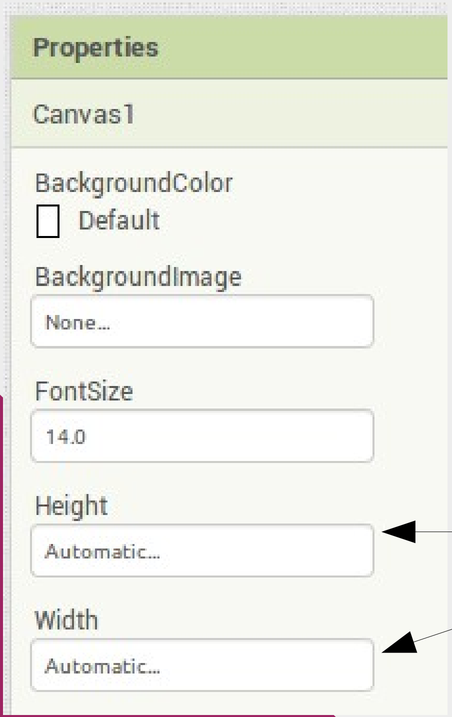
- Simple game, tap on the "mole" to score points
- Learn about graphics, sprites
- Learn about variables
- Learn about functions



Sprites and Graphics

To draw graphics in your app, you first need a **Canvas**...





Canvas1

BackgroundColor
Default

BackgroundImage
None...

FontSize
14.0

Height
Automatic...

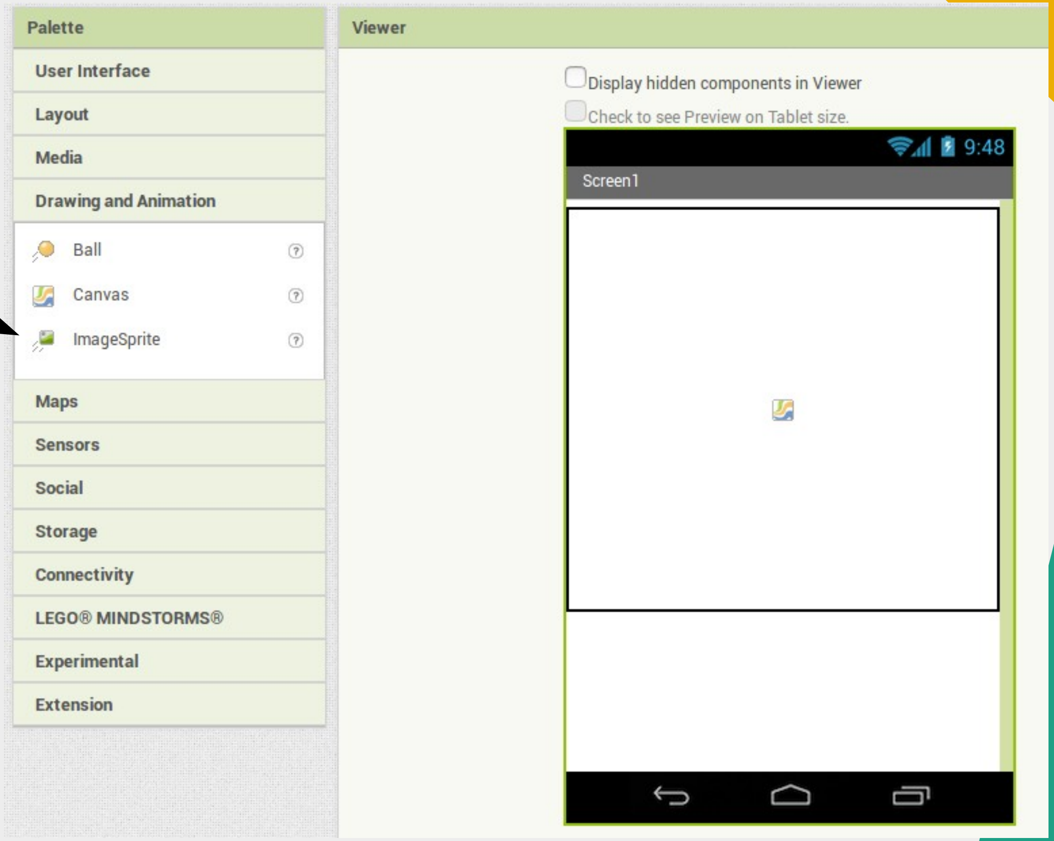
Width
Automatic...

...the default canvas size is quite underwhelming, set a suitable Height and Width for the canvas in the **Properties** panel.

Feel free to mess around with the other settings if it strikes your fancy.

Sprites and Graphics

Next, add in a **sprite**

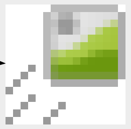


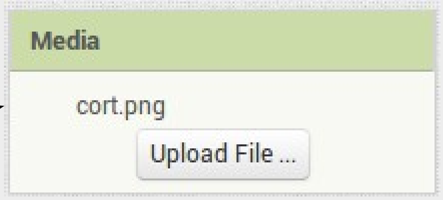
Sprites
"Sprite" is a computing term for a piece of 2D graphics that is integrated into a larger scene.

In this case, the sprite is integrated on top of the canvas.

Sprites and Graphics

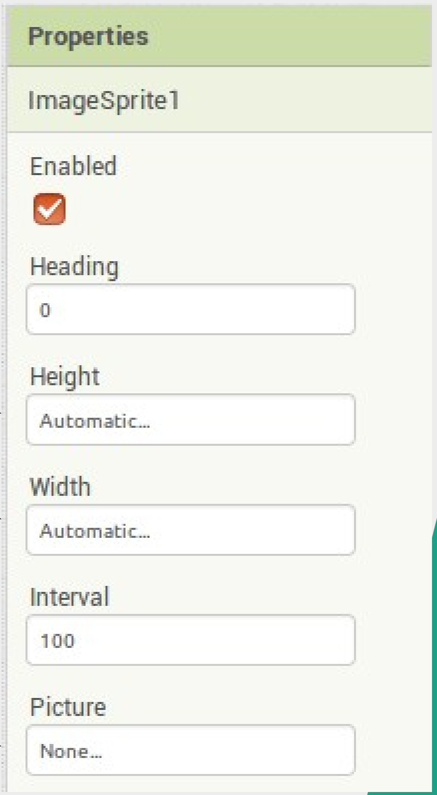
Your default sprite looks
like this... boring...
(this is just a placeholder)





cort.png

Upload a cool image...
(...take your pick, my photo is available upon request)



Properties

ImageSprite1

Enabled ✓

Heading 0

Height Automatic...

Width Automatic...

Interval 100

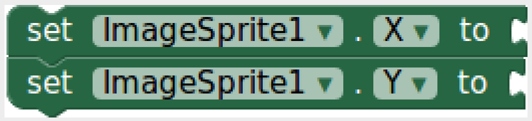
Picture None...

...set the size of the sprite...

...then set the sprite to use
your uploaded image.

Moving the Sprite

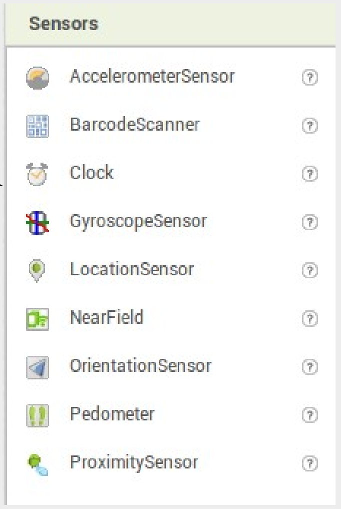
- We want the sprite to jump to a random position every few seconds
- Easy enough to move the sprite, just need to change its X and Y coordinates



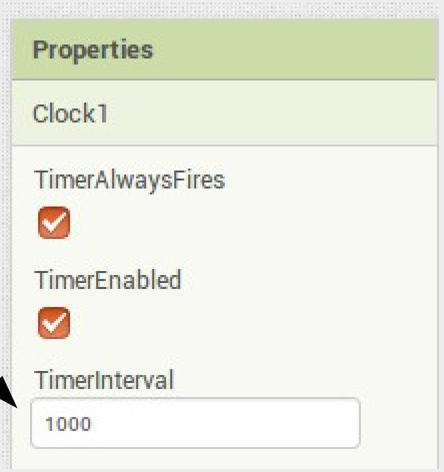
- ...but remember that App Inventor is “Event Driven”, code only runs when there is an event to trigger it
- What event can we use?

Moving the Sprite

Use the “Clock” component…
(under “Sensors”)



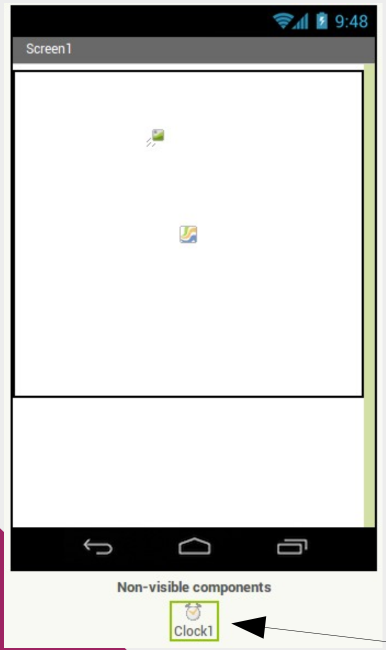
Set the interval between each firing of the “Timer” event
(units is in milliseconds)



TimerAlwaysFires ✓

TimerEnabled ✓

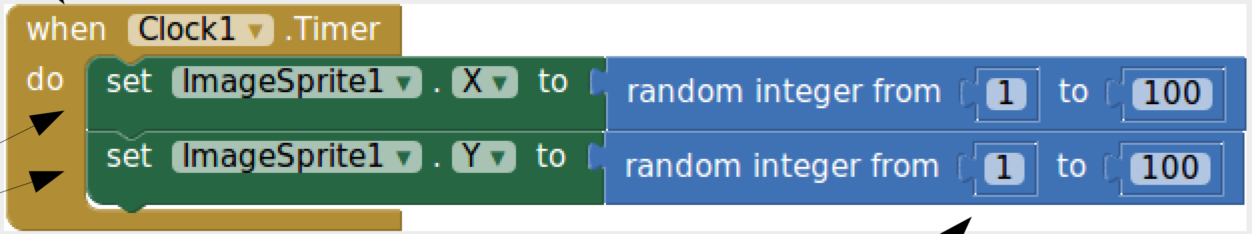
TimerInterval 1000



Non-visible component
(listed below the screen)

Writing the Code

When the “Timer” event is triggered...
(...based on your earlier setting)



```
when Clock1 .Timer
do  set ImageSprite1 . X to  random integer from 1 to 100
    set ImageSprite1 . Y to  random integer from 1 to 100
```

Change the X and Y coordinates of the sprite

We’ll change the X and Y coordinates to a random integer

Now YOU Try!

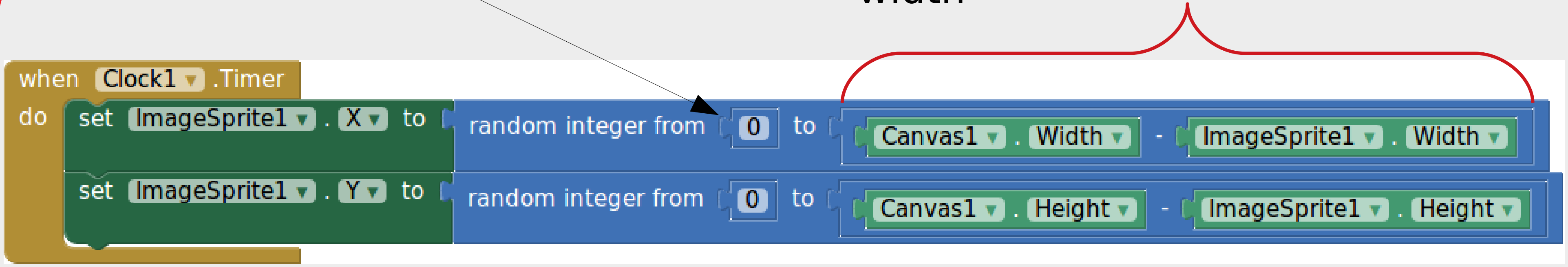
- Challenges
  - Random integer 1 to 100 won’t make full use of the canvas, how can we utilize the full width and height of the canvas?
  - Hint: there are blocks providing the height and width of the canvas and sprite
- If you need to refer, these slides are available at...

  http://www.aposteriori.com.sg/projects
- ...don’t cheat and look at the solution on the next slide. You gotta figure it out yourself!

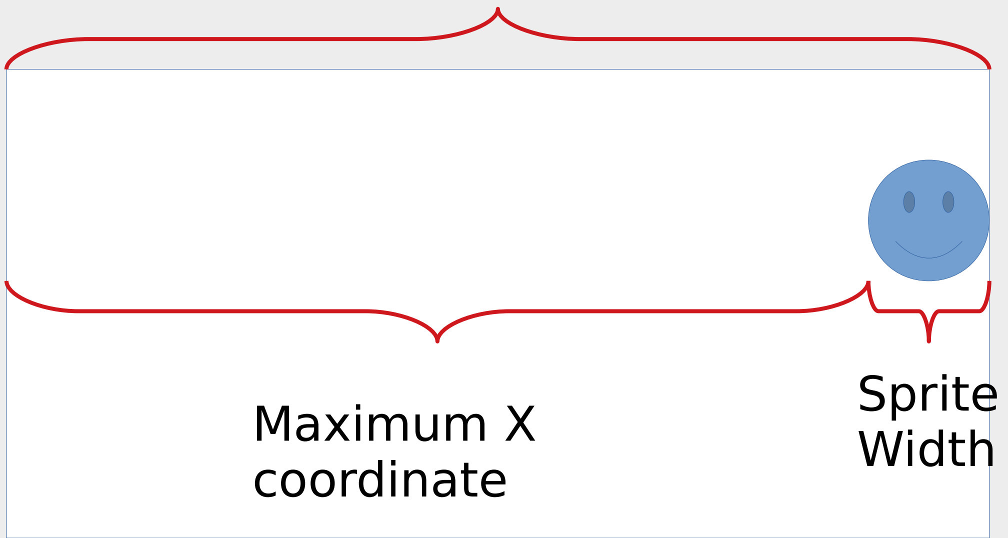
Challenge Solution

Minimum X coordinate is 0...

Maximum X coordinate is equal to the canvas width minus the sprite width

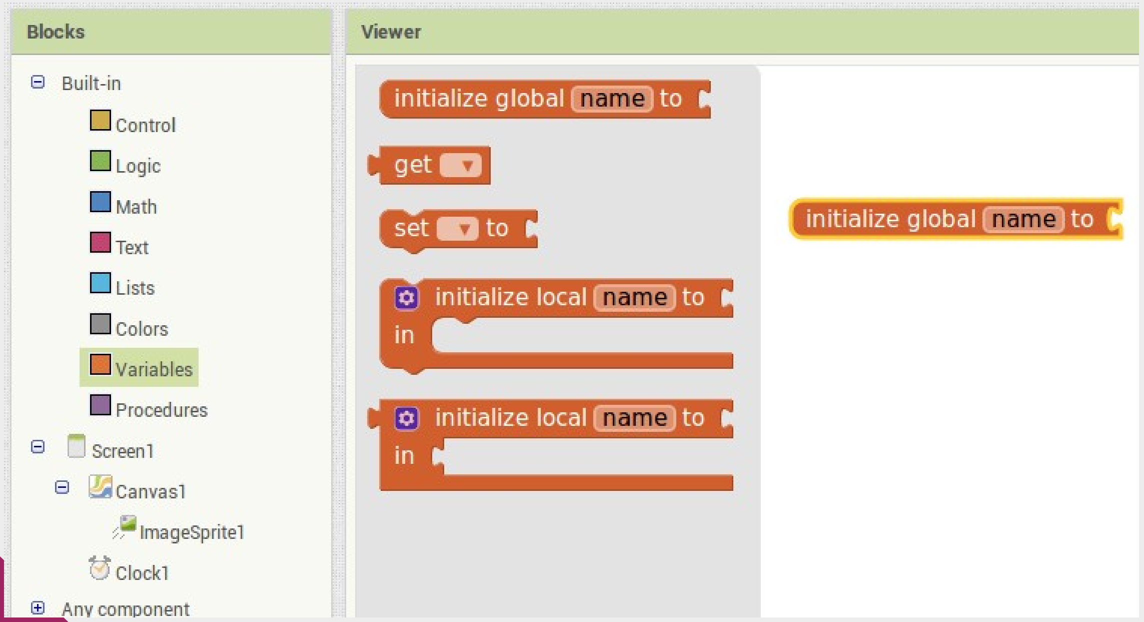


```
when Clock1 .Timer
do  set ImageSprite1 . X to  random integer from 0 to  Canvas1 . Width - ImageSprite1 . Width
    set ImageSprite1 . Y to  random integer from 0 to  Canvas1 . Height - ImageSprite1 . Height
```



Keeping Scores

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

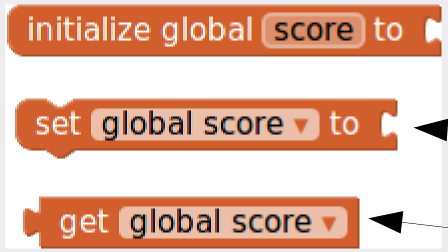


<u>Variables</u>
Nearly all programming languages have variables.

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

Keeping Scores

3 basic operations for every variable...



Initialize: Set value at start of program

Set: Set a value

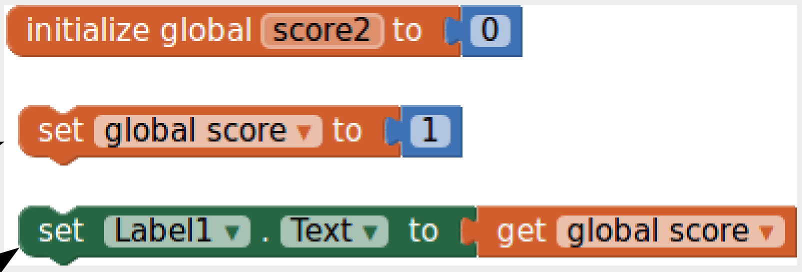
Get: Get the value

<u>Example usage</u>

Initialize score to 0 (Initialize is special, it doesn't need to be inside an event!)

Set score to 1. This needs to be in an event.

Set the text in Label1 to whatever value is is score. Needs to be in an event.


initialize global score2 to 0
set global score to 1
set Label1 . Text to get global score

<u>Global</u>
Global variables are available for read and write anywhere in the program.

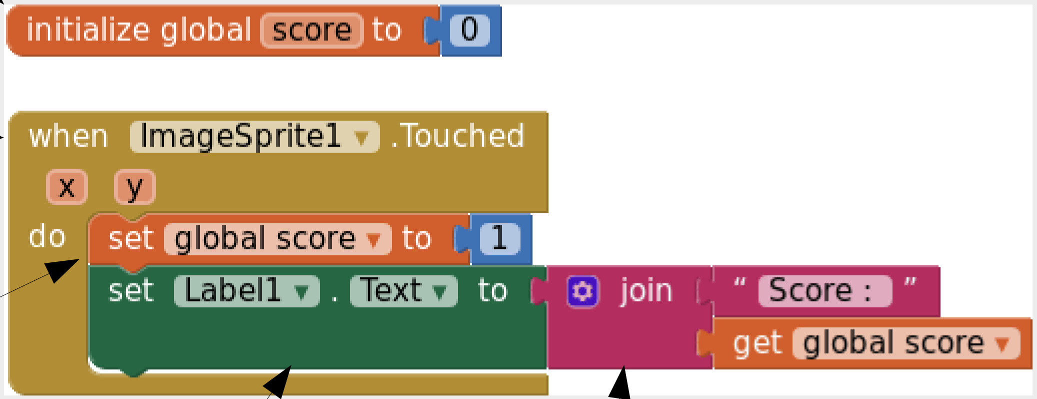
<u>Local</u>
Local variables are only available within a limited region of the program.

Keeping Scores

Initialize score to be 0 at the start of the program

Whenever the sprite is "Touched"...

...set the score to 1...



```
initialize global score to 0

when ImageSprite1 .Touched
  x  y
do  set global score to 1
    set Label1 . Text to  join  " Score : "
                                get global score
```

...then display it in the label (Don't forget to add a label component first!)

This joins the string "Score :" to the actual value of score. The result will look like...

Score : 1

Now YOU Try!

- Challenges
  - The score is set to 1 on each touch. How can we make it increment on every touch? (ie. 1, 2, 3, 4)
  - Add in a “Reset” button, and make it reset the score to zero when pressed
  - Immediately move sprite to random position on every touch
- If you need to refer, these slides are available at...

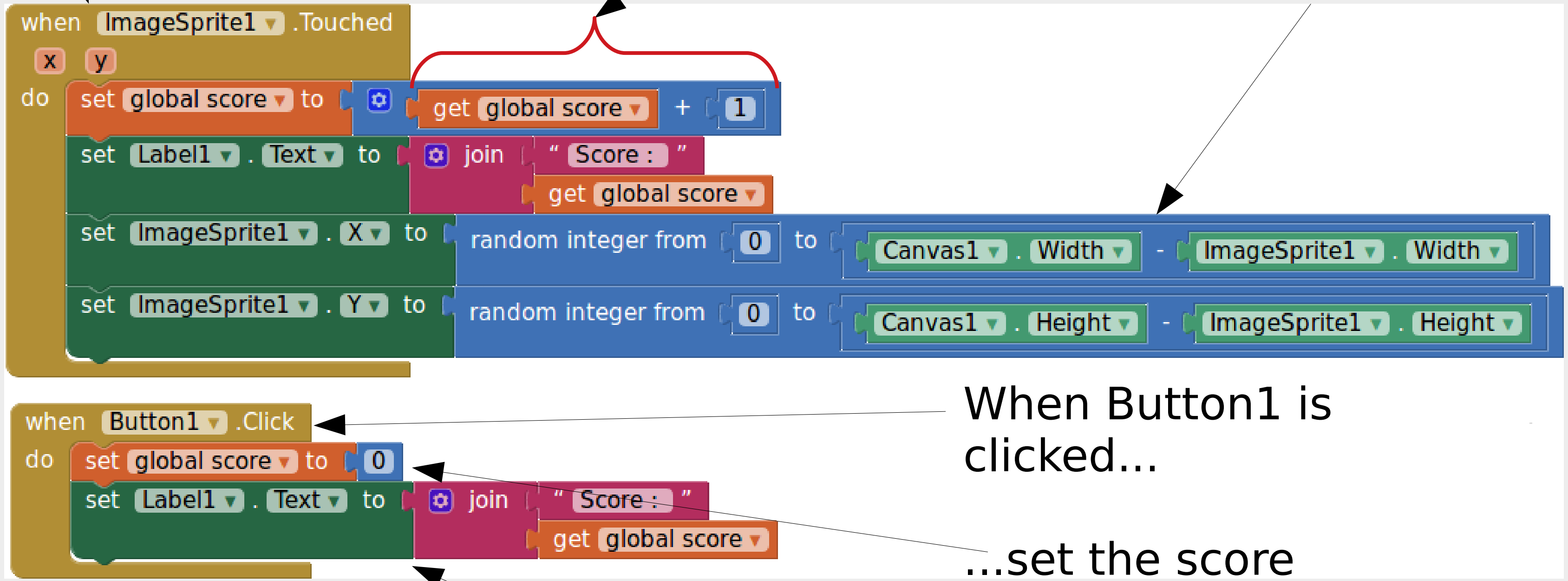
  http://www.aposteriori.com.sg/projects
- ...don’t cheat and look at the solution on the next slide. You gotta figure it out yourself!

Challenge Solution

Whenever the sprite is "Touched"...

Set the score to "score + 1"

Move the sprite to a random position



```
when ImageSprite1 .Touched
  x  y
do  set global score to ( get global score + 1 )
    set Label1 . Text to join ( " Score : " , get global score )
    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 )

when Button1 .Click
do  set global score to 0
    set Label1 . Text to join ( " Score : " , get global score )
```

When Button1 is clicked...

...set the score to zero...

...and display it in the label

Functions / Procedures

- Functions are pieces of code that…
  - Separated from the rest of the code
  - Does something
  - Optionally accept inputs or provide outputs
  - Can be called by other pieces of code
- May also be called
  - Procedures (App Inventor uses this term)
  - Subroutines (rarely used these days)

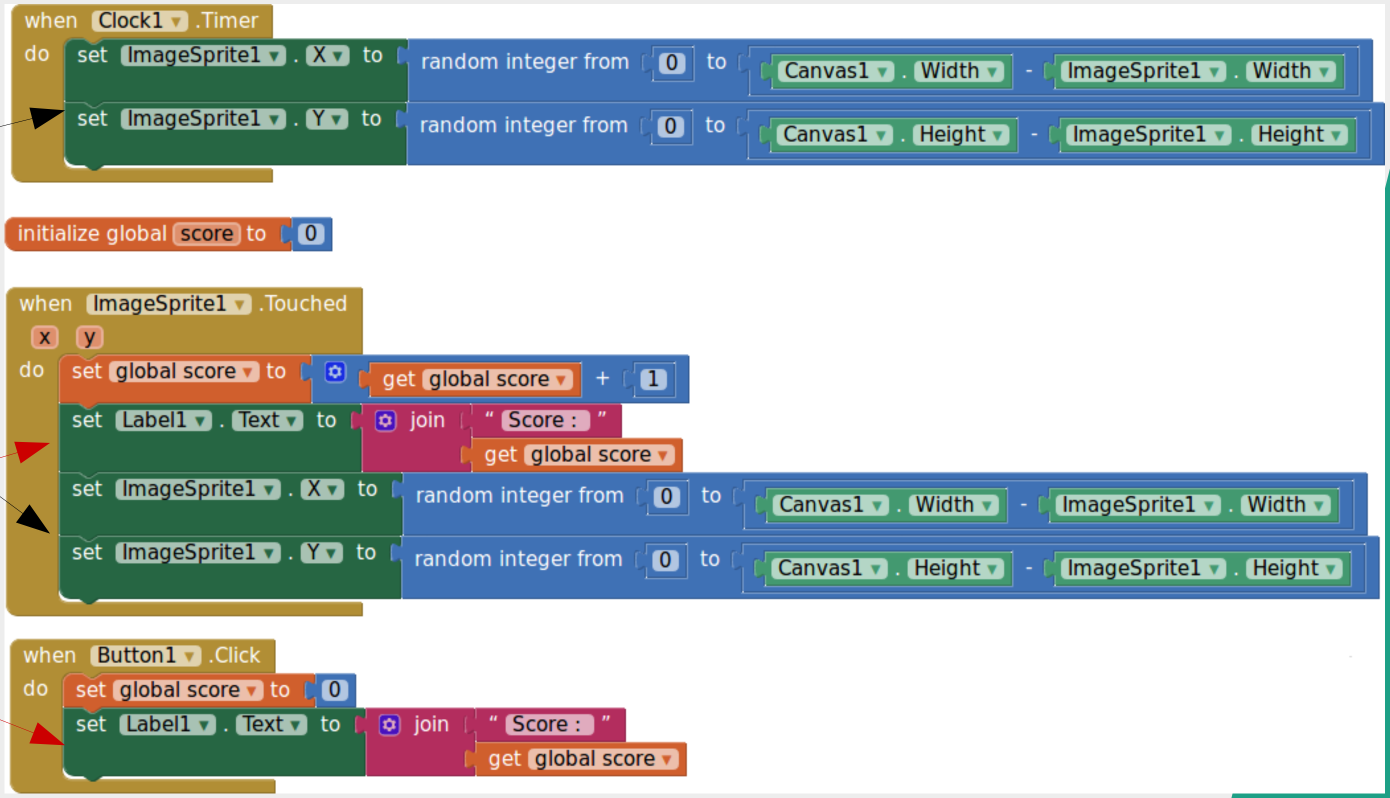
Functions / Procedures

- Why use functions?
  - Minimize repetition!

Our current program...

Repeated

Repeated



```
when Clock1 .Timer
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

initialize global score to 0

when ImageSprite1 .Touched
  x  y
do  set global score to  get global score + 1
    set Label1 . Text to  join  " Score : "
                                get global score
    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

when Button1 .Click
do  set global score to 0
    set Label1 . Text to  join  " Score : "
                                get global score
```

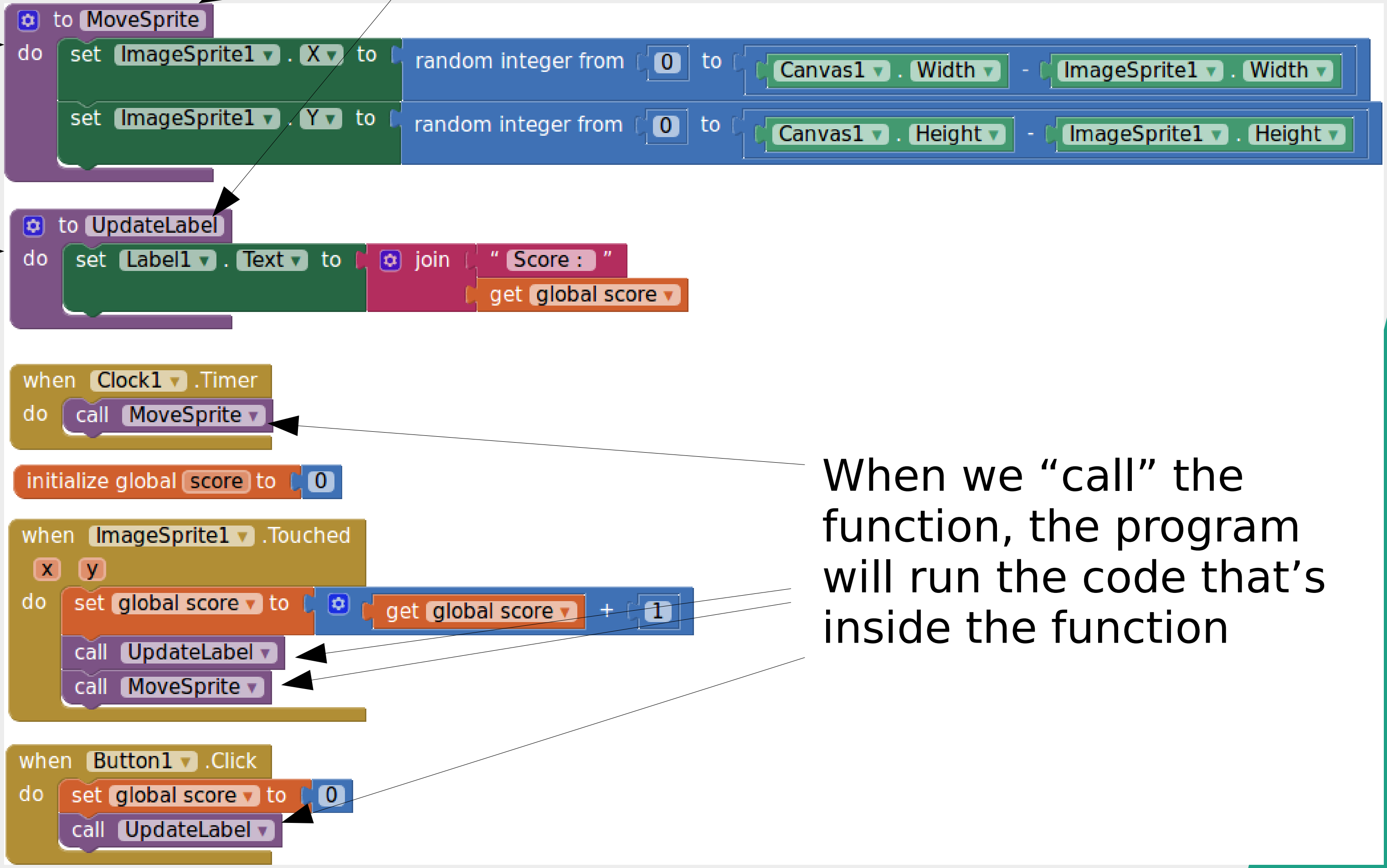
Functions / Procedures

This is a function…

This one too…

You can name your functions whatever you want

When we “call” the function, the program will run the code that’s inside the 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

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
```

Functions / Procedures

Functions can also accept inputs and return an output

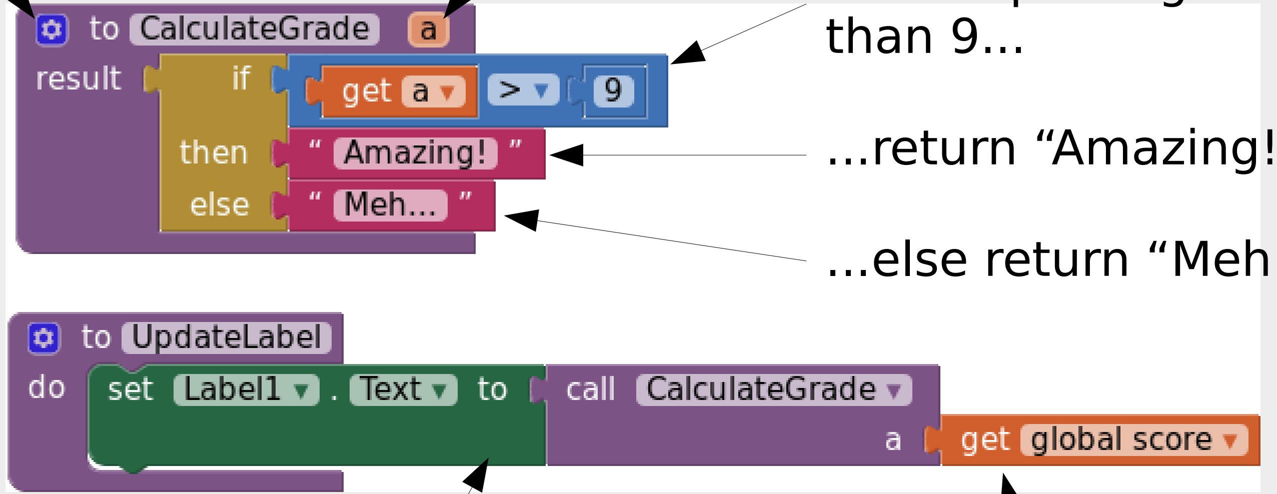
Use the “gear” to add or remove inputs

We named the input “a”. (Note: “a” is a **Local Variable**. It is only available inside this function.

If the input is greater than 9…

…return “Amazing!”…

…else return “Meh…”



```
to CalculateGrade a
result  if   get a > 9
        then “ Amazing! ”
        else “ Meh... ”

to UpdateLabel
do  set Label1 . Text to  call CalculateGrade
                               a  get global score
```

The function output is used to set the label text

We provide “score” as the input to the function

Challenges

- Track both hits and misses
- Increase the speed a little every time you gain a point; the higher the score, the faster the speed
- Add in sound, vibration, graphical effects

Mini Project

School Map

- Create an app that shows visitors and new students the location of classrooms, sports hall, etc on a map
- Hint: Use a school map image as the background of a canvas
- Hint 2: Let the user select locations using a Spinner