

WRITING MOBILE APPS FOR INVESTIGATIVE STUDY OF PHYSICS

Session 1

Objectives

- This course aims to introduce how Cloud Computing can be used to facilitate e-Learning in schools. After finishing the course, participants should be able to:
 - use common free-of-charge development tools or platforms, such as the web-based Apps Inventor or other easily accessible development tools, to develop Mobile Apps for smartphones or tablet PCs so that these devices could be used as data loggers for Physics experiments;
 - acquire basic programming skills of writing simple Mobile Apps to invoke the embedded sensors, record and export data for Physics demonstrations / experiments by using free-of-charge Mobile Apps development tools or platforms mentioned in (a);

Objectives

- Understand the properties of the many sensors embedded in mobile devices such as GPS, compass, accelerometers, etc. and master the basic techniques to program these sensors, activate them for use in Physics experiments, and extract and export data for in-depth analysis;
- design some investigative experiments with worksheets and sample measurements for Physics by using the custom-made Mobile Apps developed by the teachers;
- exploit the potential of students and develop their creativity by guiding them to write Mobile Apps for investigative study of Physics or applications of Physics to improve living; and
- appreciate how knowledge and skills related to science (in particular Physics), technology and mathematics can be appropriately integrated in learning and in applications in the real world.

Rundown

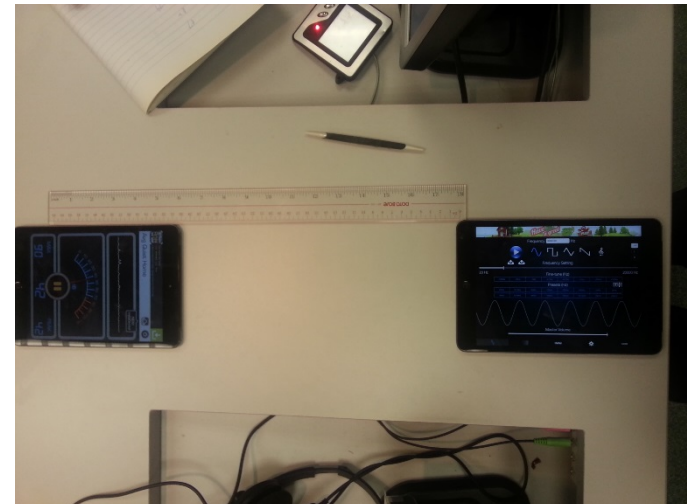
- Introduction
- Physics and Mobile Applications
- Introduction to App Programming
- Programming your First Mobile Application
- Programming with Sensors
- Doing experiments with Mobile Apps
- Homework

Resources

- Most of the programming projects are available for download from our course web page
- Online video tutorial are also available to guide teachers on the programming examples step by step
- Please refer to the course web page on
 - **Schoology** - <http://www.schoology.com/>
 - Access Code: **XTP3F-Z7J4T**
 - **Moodle** - <http://caite.fed.cuhk.edu.hk/m>
 - Login as Guest
 - Password: Caite!123

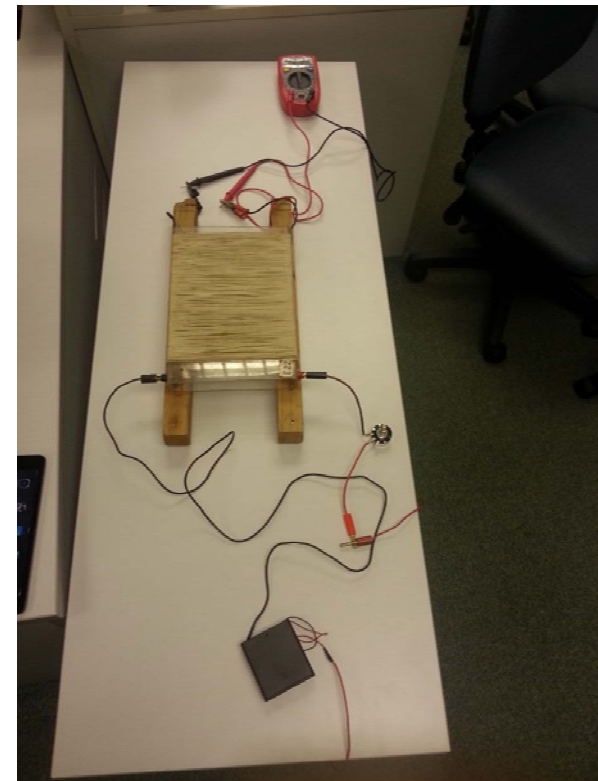
1.1 Brief Review of 2015 RTC Physics course

- Mobile free Apps to facilitate learning and teaching of Physics
- Smartphone as a sensor for Physics experiments/ a data logger with output in different formats – raw data, graphs, etc.
- A camera / video motion analysis
- A microphone/ pressure sensor
- A speaker / signal generator



1.1 Brief Review of 2015 RTC Physics course

- An accelerometer sensor
- Magnetic field sensor/ Hall probe
- An ambient light sensor
- Gyroscope sensor
- GPS logger



1.1 Advantages of using Mobile Apps

- Almost everyone gets a smartphone nowadays
- Experiments outside the laboratory
- Easy to use
- Different kinds of sensors are available
- Carry out experiments anytime anywhere during daily life

1.1 Limitations on free Apps

- Most data on iPhone/ iPad cannot be easily exported
- All data are recorded even not of interests of users (e.g. 3 direction accelerations are recorded)
- Further analysis of data on Excel before getting meaningful relationships
- Students get confused on handling accelerometer data: pseudo gravity, 3-axial acceleration, direction of angular velocity (polar vector)
- Not designed for specific tasks or experiments

1.1 Advantages of writing Mobile Apps

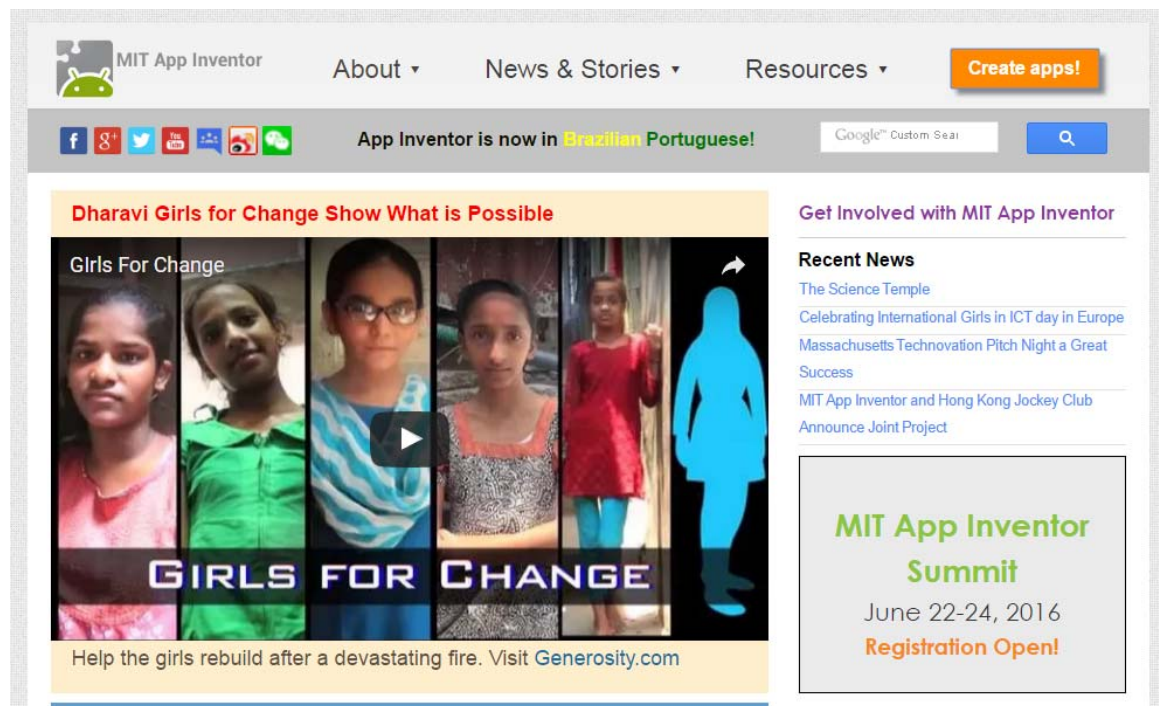
- Development tool of App Inventor
- Simple pre-coded blocks are relatively easy to use
- Tailor-made programs for specific tasks or experiments
- Necessary data can be obtained directly
- Data can be retrieved from mobile phone for processing
- Better clarity, less confusion
- Effective for investigative study
- An activity that showcases integration of application of knowledge and skills across different subjects discipline(a good means to promote STEM education in schools)
- Students can learn coding or computer programming through writing apps

1.2 Introduction to App Inventor

- App Inventor was introduced by Google's Mark Friedman and MIT Professor Hal Abelson in 2009
- App Inventor runs as a web service administered by MIT
- In 2015, the MIT App Inventor community consists of nearly 3 million users representing 195 countries.
- App Inventor provides a **simplified model** and **visual programming language** for users to build mobile apps for the Android Platform

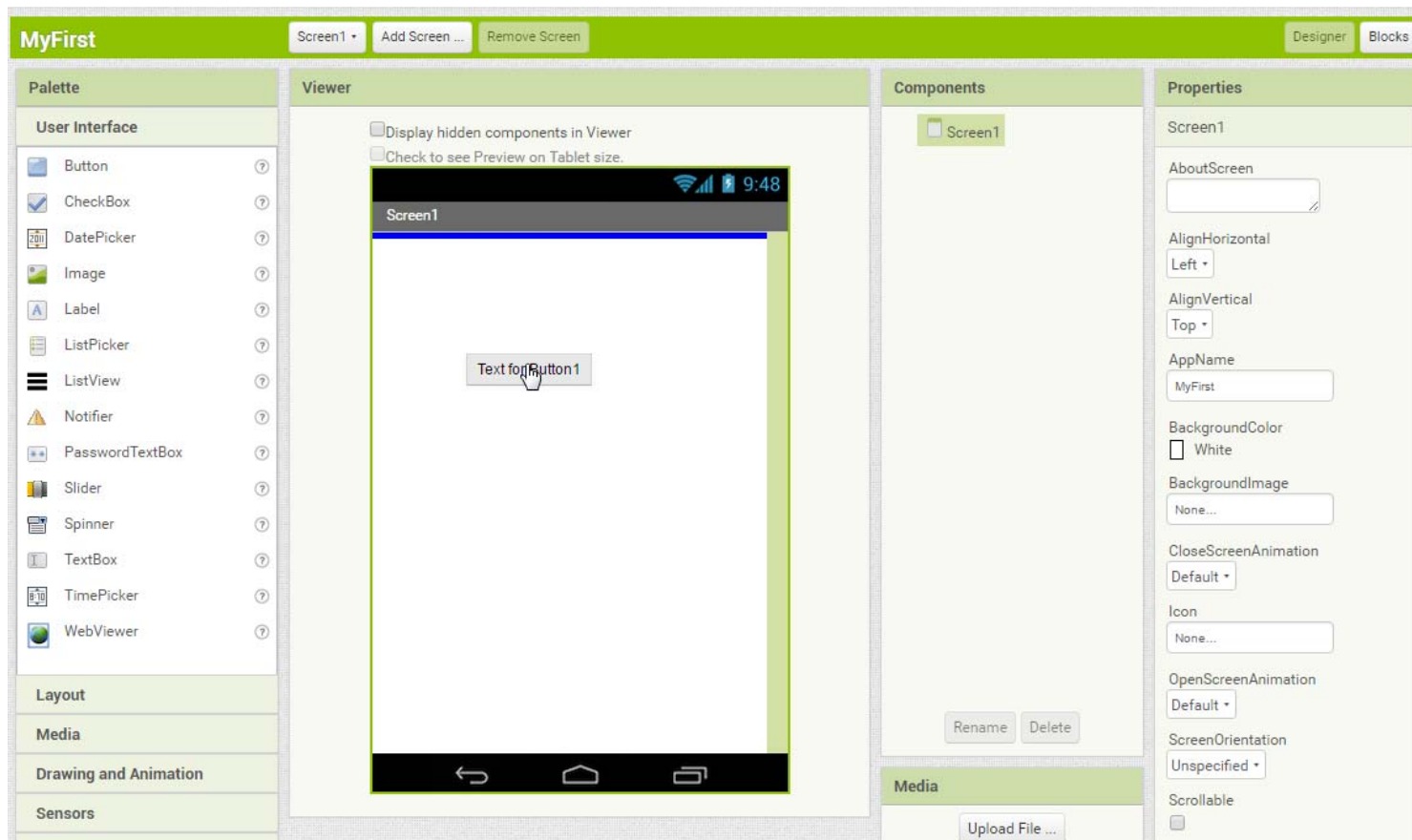
1.2 Introduction to App Inventor

- App Inventor is completely web-based, no software installation is required
- URL: <http://appinventor.mit.edu/explore/>
- Any users with a Google account can login to the online web system and starting building Apps for Android



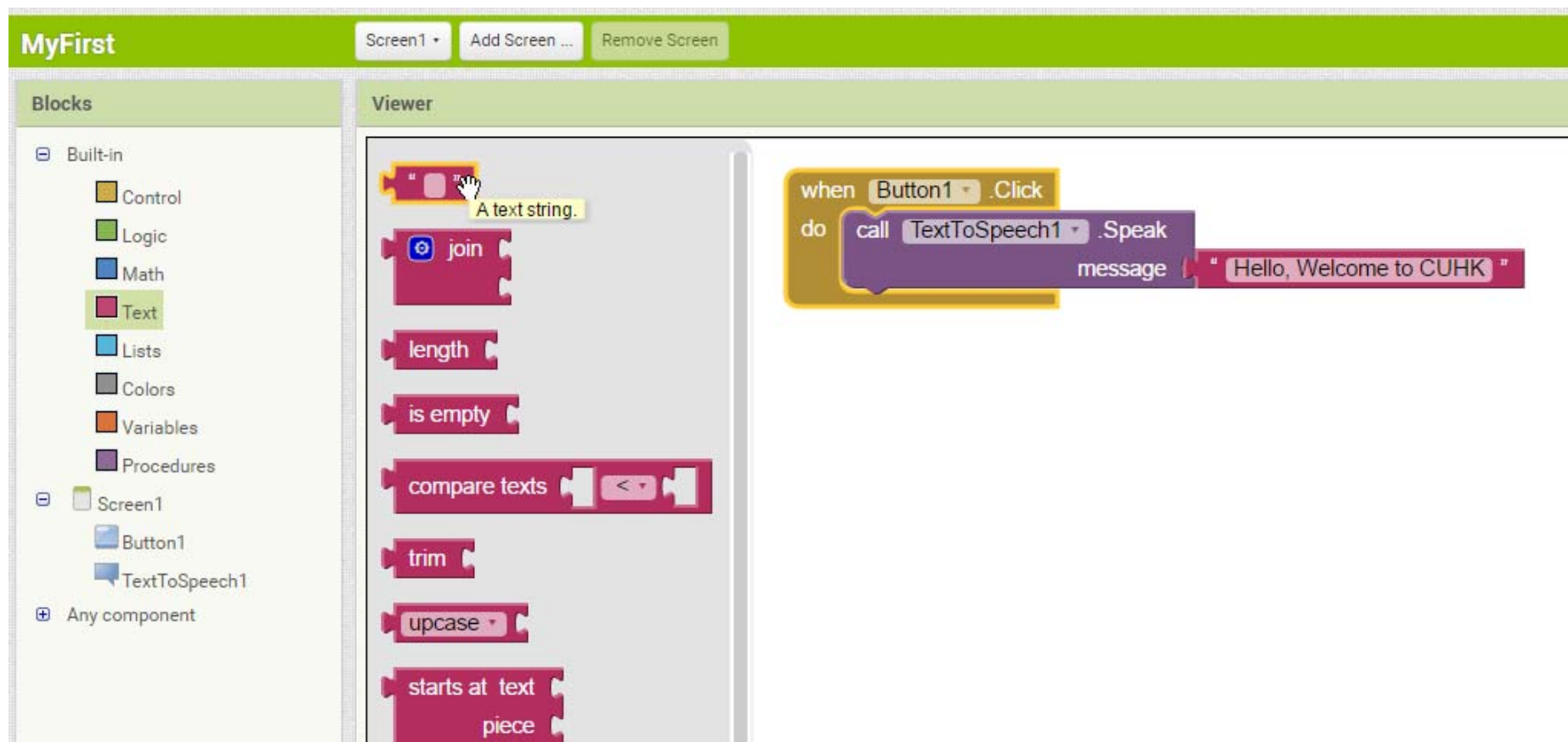
1.2 Introduction to App Inventor

- Drag and Drop Design Environment



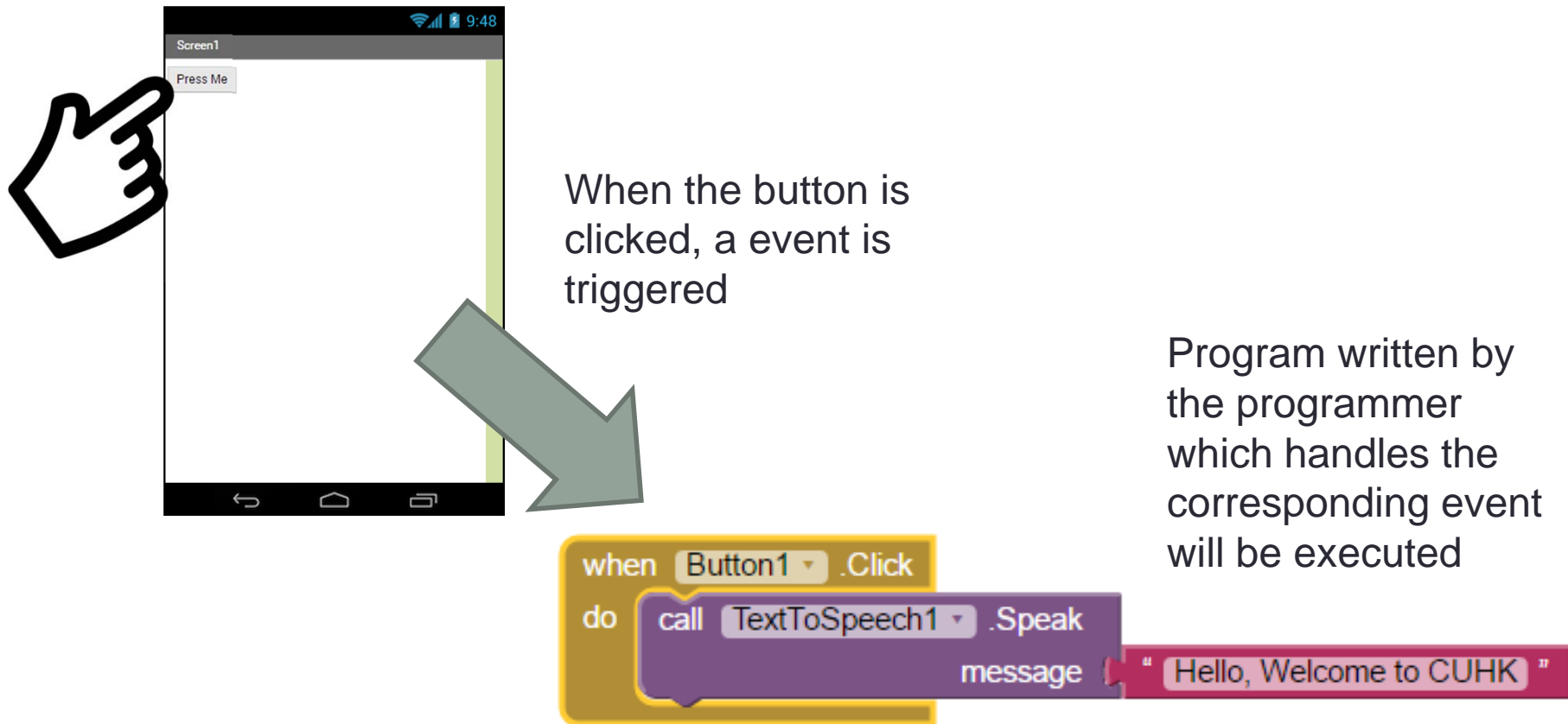
1.2 Introduction to App Inventor

- Visual Programming Language



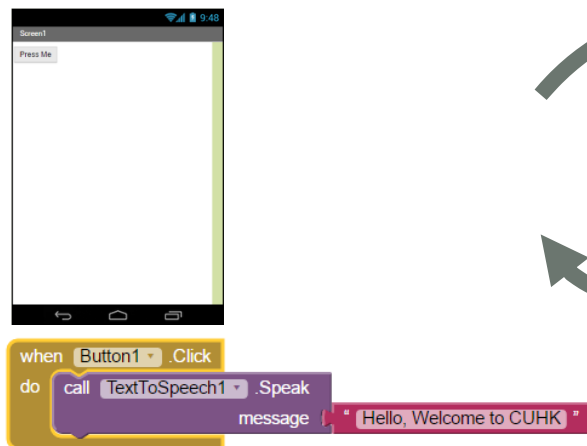
1.2 Introduction to App Inventor

- Event Based Programming Model

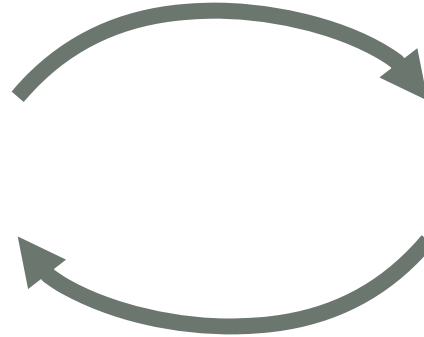


1.2 Introduction to App Inventor

- Steps to make an App using App Inventor



Step 1:
Design the User Interface
and Create the Events
Programs

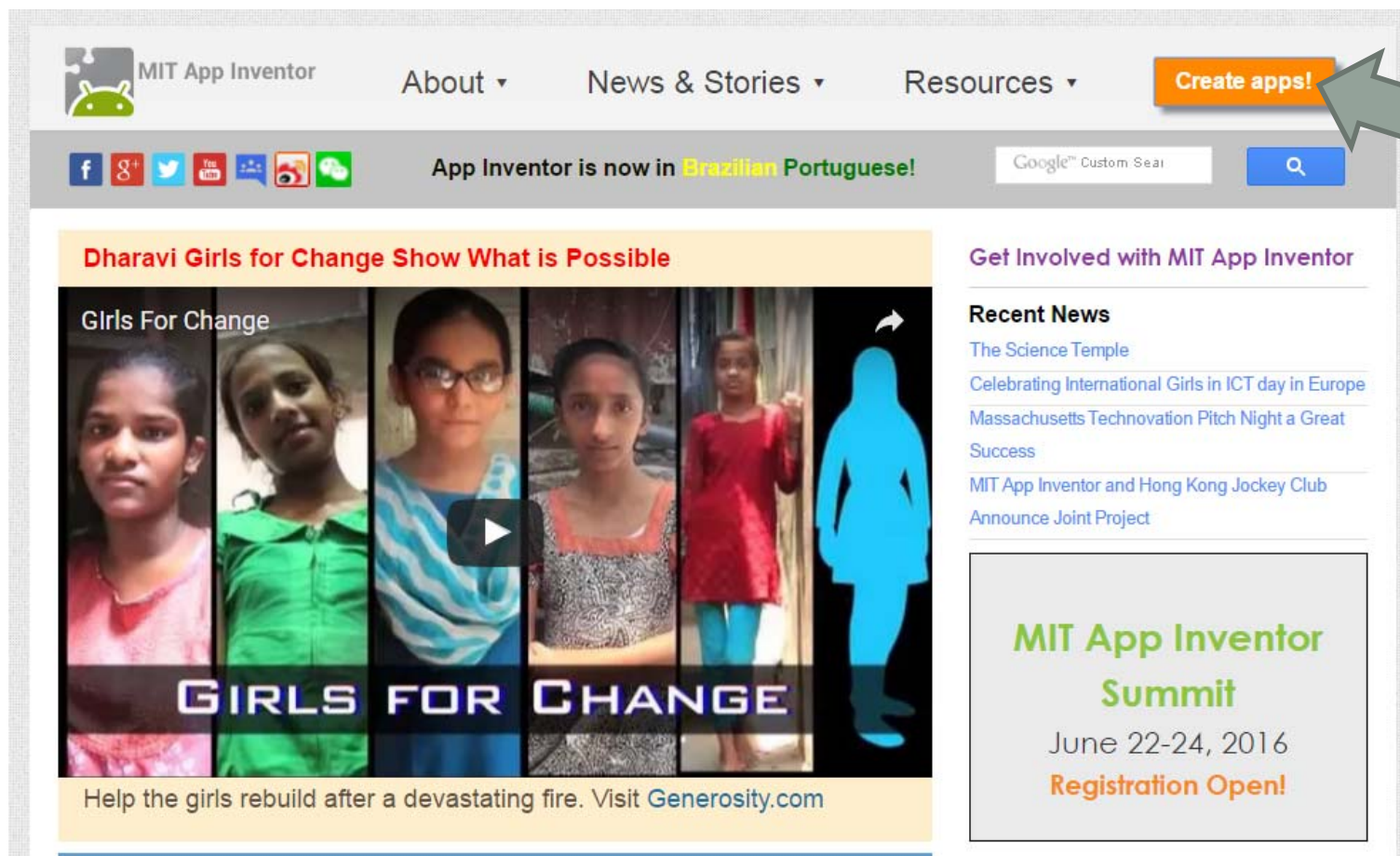


MIT
AICompanion
app

Step 2:
Test the program (and debug the
program) but sending the program to a
Android device through WiFi
(by using the free MIT AICompanion
App)

1.2 Introduction to App Inventor

- Login to your App Inventor Programming Environment



To use App Inventor for Android, you must accept the following terms of service.

Terms of Service

MIT App Inventor Privacy Policy and Terms of Use

MIT Center for Mobile Learning

Welcome to MIT's Center for Mobile Learning's App Inventor website (the "Site"). The Site runs on Google's App Engine service. You must read and agree to these Terms of Service and Privacy Policy (collectively, the "Terms") prior to using any portion of this Site. These Terms are an agreement between you and the Massachusetts Institute of Technology. If you do not understand or do not agree to be bound by these Terms, please immediately exit this Site.

MIT reserves the right to modify these Terms at any time and will publish notice of any such modifications online on this page for a reasonable period of time following such modifications, and by changing the effective date of these Terms. By continuing to access the Site after notice of such changes have been posted, you signify your agreement to be bound by them. Be sure to return to this page periodically to ensure familiarity with the most current version of these Terms.


Description of MIT App Inventor

From this Site you can access MIT App Inventor, which lets you develop applications for Android devices using a web browser and either a connected phone or emulator. You can also use the Site to store your work and keep track of your projects. App Inventor was originally developed by Google. The Site also includes documentation and educational content, and this is being licensed to you under the Creative Commons Attribution 4.0 International license ([CC BY 4.0](#)).

Account Required for Use of MIT App Inventor

In order to log in to MIT App Inventor, you need to use a Google account. Your use of that account is subject to Google's Terms of Service for


I accept the terms of service!

**MIT App Inventor 2**
Beta

Projects▼ Connect▼ Build▼ Help▼ My Projects Gallery Guide Report an Issue English▼ aijackyhui@gmail.com▼

Start new project Delete Project Publish to Gallery

My Projects

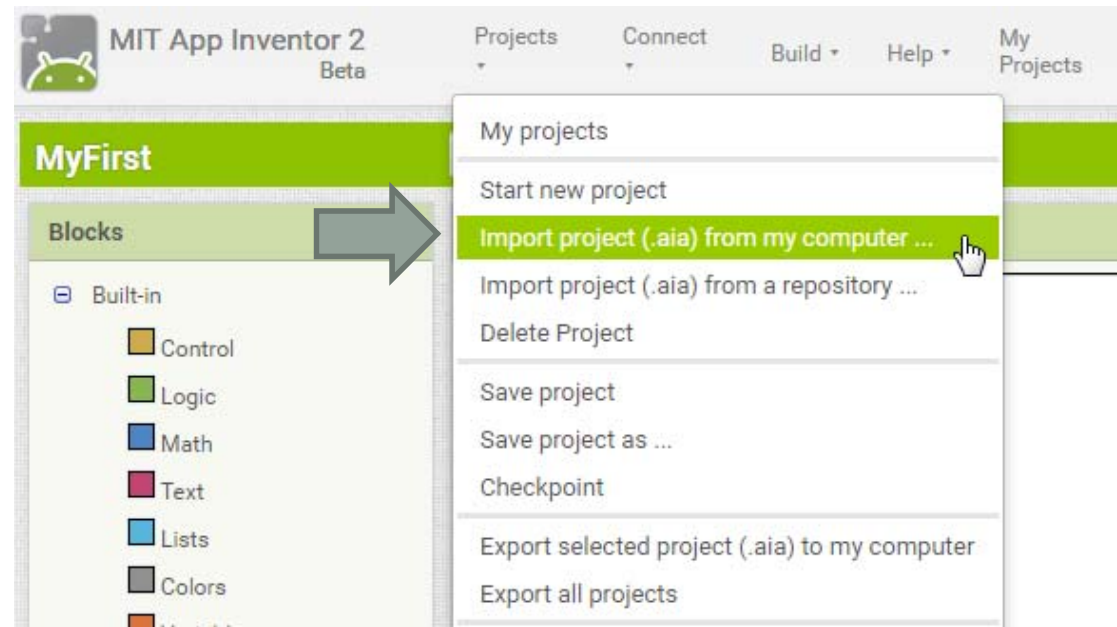
Name	Date Created	Date Modified ▼	Published
<div><div>Welcome to App Inventor 2!</div><div><p>You do not have any projects in App Inventor 2. To learn how to use App Inventor, click the "Guide" link at the top of the window; or to start your first project, click the "Start New Project" button at the upper left of the window.</p><p>Happy Inventing!</p></div></div>			

1.2 Introduction to App Inventor

- To allow you to get familiar with the programming environment, we have prepared three programs written in App Inventor
 - Unit Converter
 - Location Aware GPS app
 - Simple Measuring Tool – App to measure the frequency of a user tapping on a button)
- You can download the program files from the course web page
 - converter.aia
 - gps.aia
 - buttonfreq.aia

1.2 Introduction to App Inventor

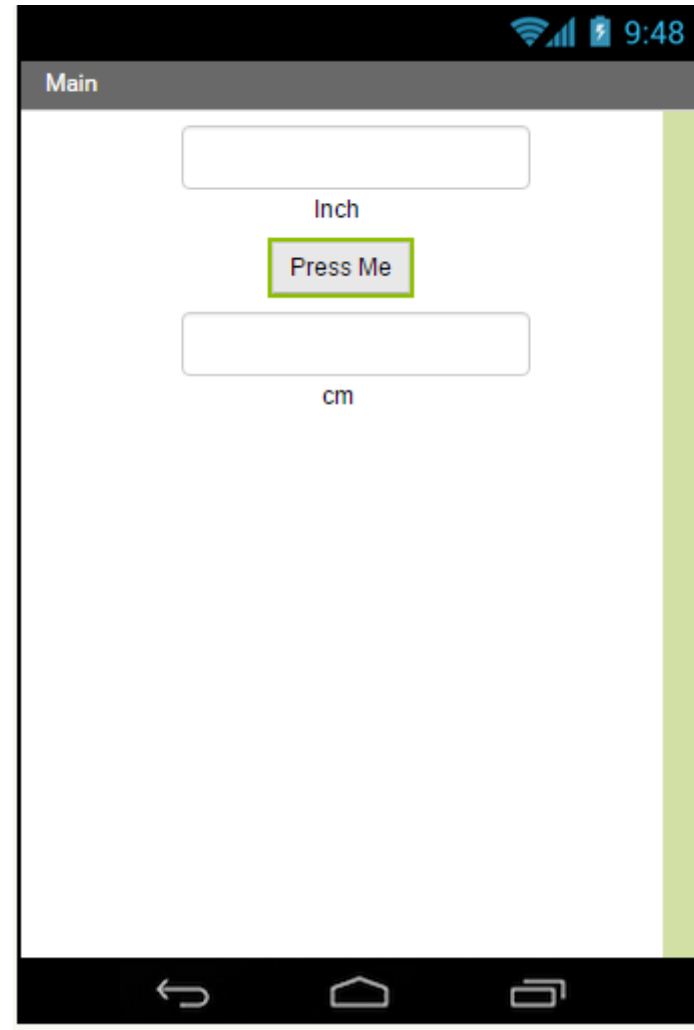
- You can import the program into your App Inventor account



- Let's import the program converter.aia first

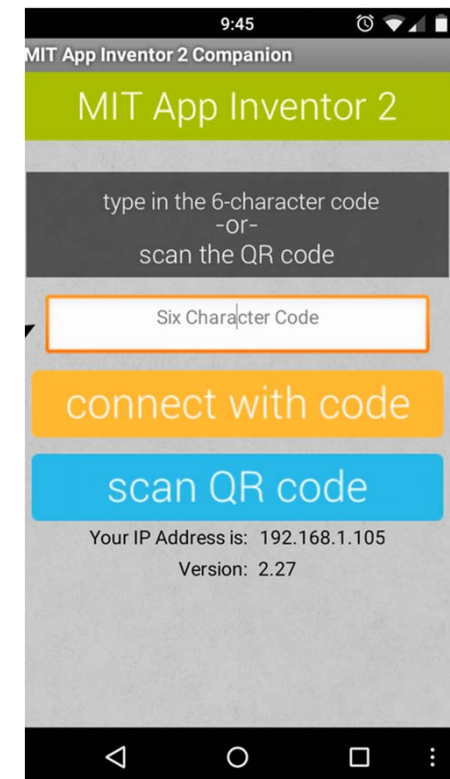
1.2 Introduction to App Inventor

- Once imported, you will see the interface of the App
- This is the interface of the App converter.aia



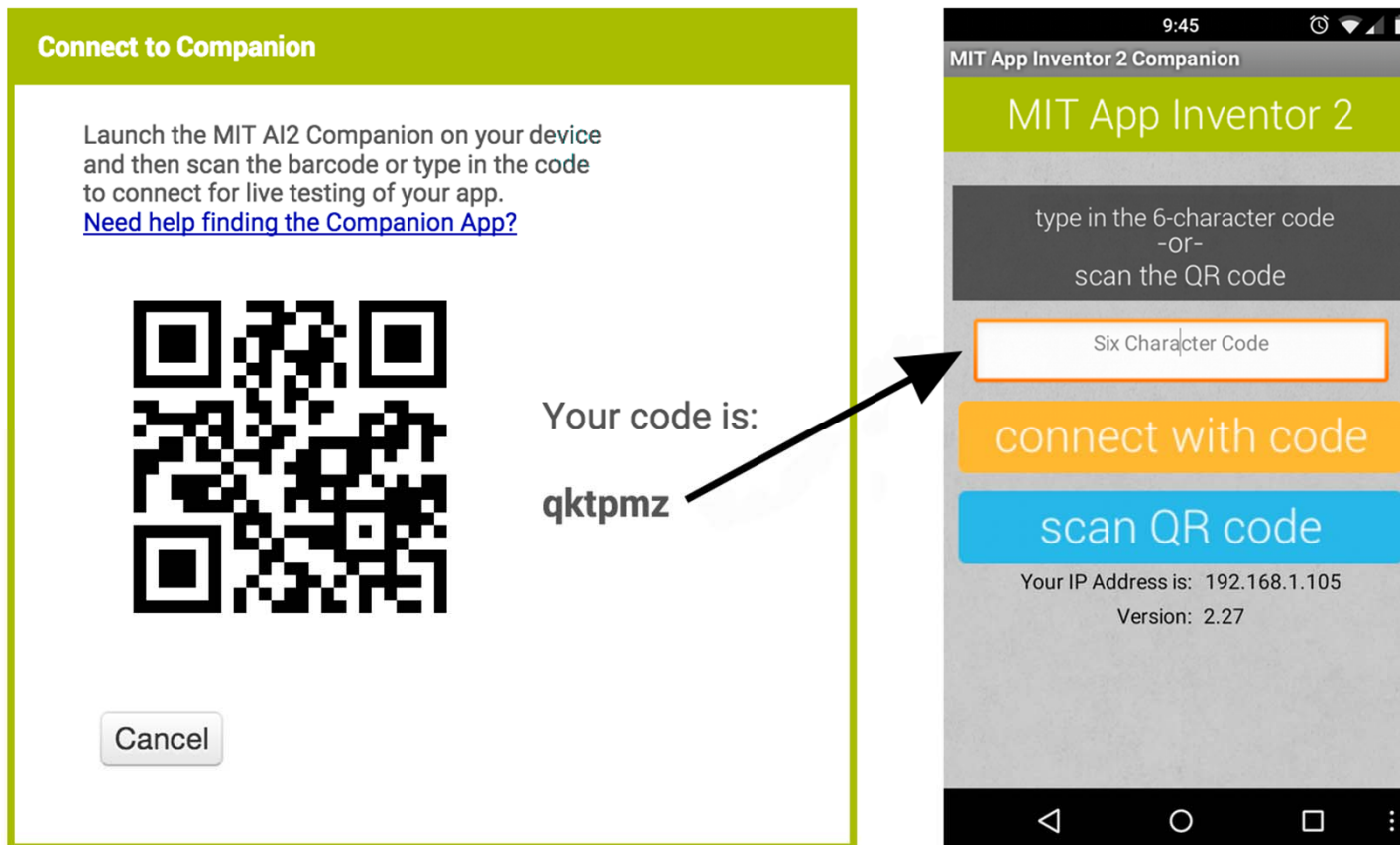
1.2 Introduction to App Inventor

- Now, it is time to test the program
- On your Android Tablet, Look for the app MIT AICompanion 2
- Once found, we can connect the Android Tablet with the program we have just written



1.2 Introduction to App Inventor

- You can type the 6 character code or scan the QR code and connect the MIT AICompanion app with the program

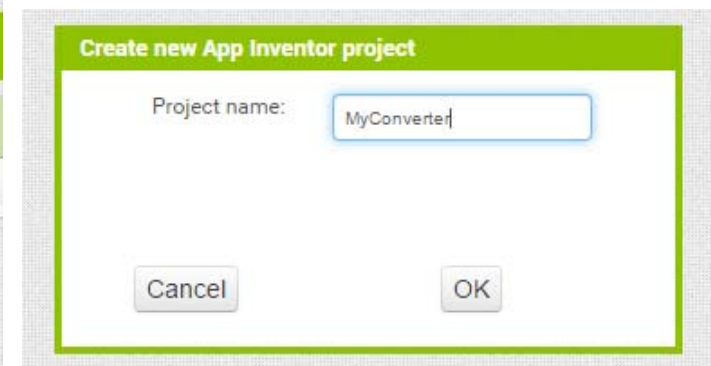
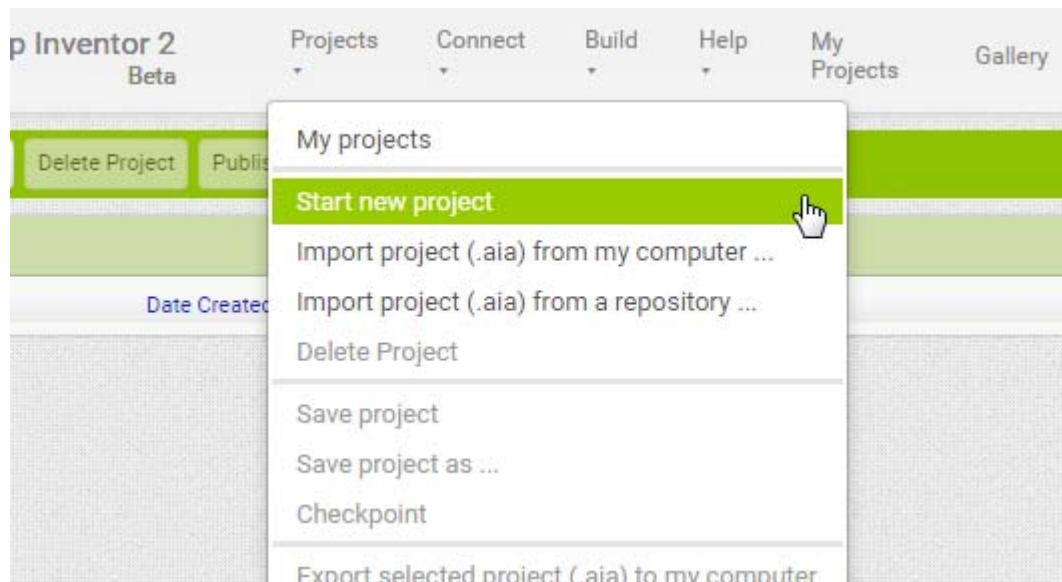


1.2 Introduction to App Inventor

- You can now try out your program
- You can also following similar steps to try out other programs now too.

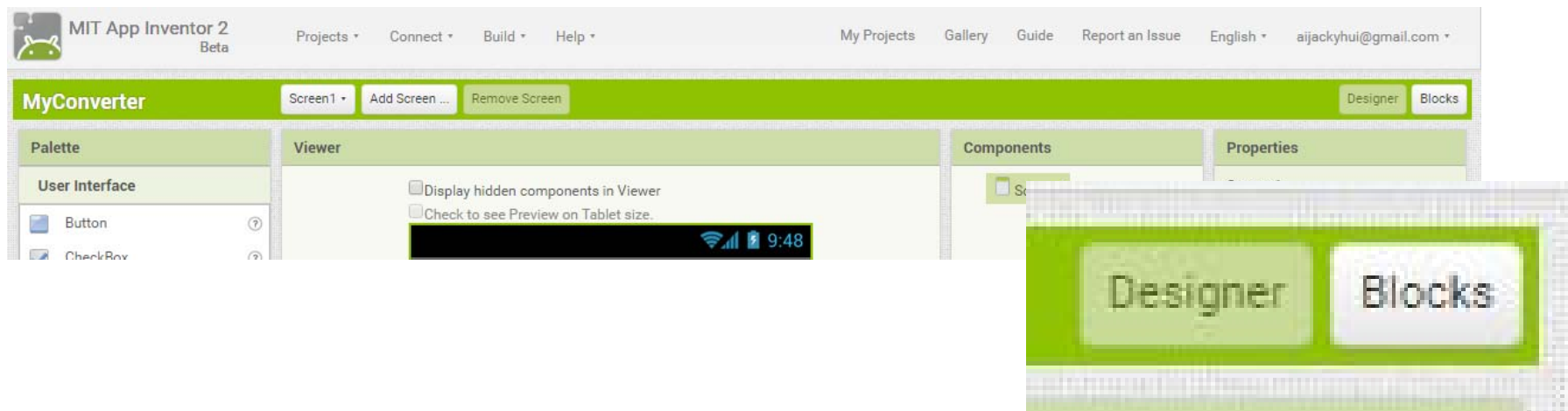
1.3 Programming in App Inventor

- App Inventor is complete cloud-based, no installation is required
- Projects created in App Inventor can be exported and saved to local hard disks. These files can be shared with others and imported back to App Inventor environment again
- Let's learn programming in App Inventor by creating the converter app from scratch
- First start a new project



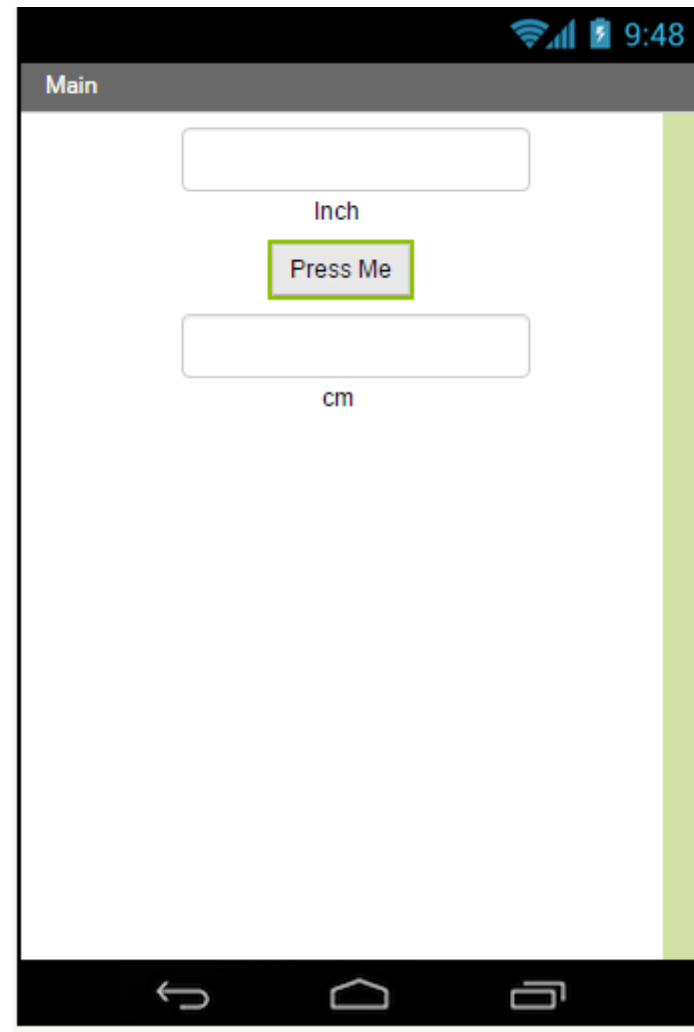
1.3 Programming in App Inventor

- The Programming environment is divided into two parts
 - The designer part – for designing the user interface (UI)
 - The blocks part – for writing programs



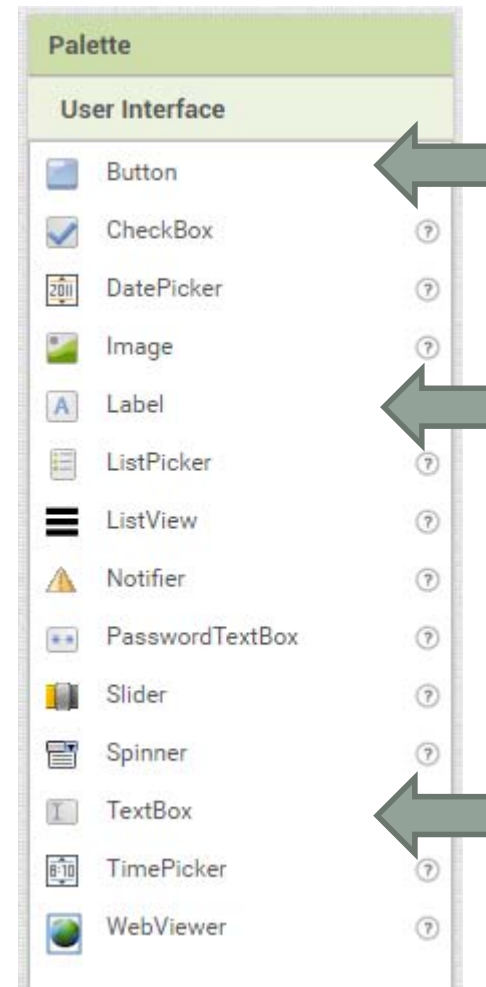
1.3 Programming in App Inventor

- Let's try to build the Unit Converter app
- First, we should design the User Interface



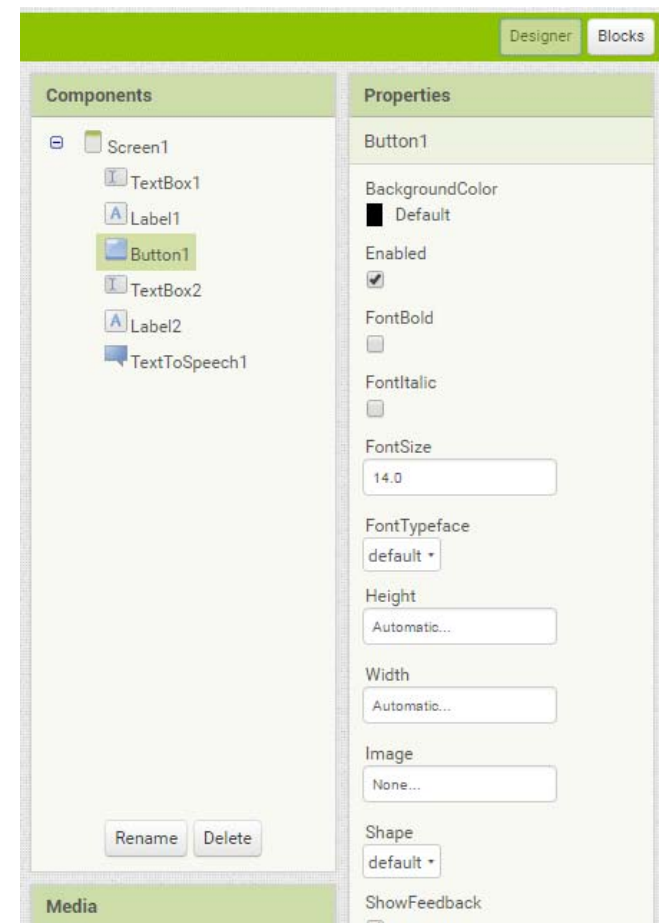
1.3 Programming in App Inventor

- Step 1:
 - Visit the App Inventor Website
<http://ai2.appinventor.mit.edu/>
- Step 2:
 - Login with your Google accounts and say yes to the user agreement
- Step 3:
 - Drag and Drop the components to the screen
 - 1 Button
 - 2 Labels
 - 2 Textboxes
- Step 4:
 - Rename the components for easier future reference



1.3 Programming in App Inventor

- By changing the properties, we can arrange the components as follows



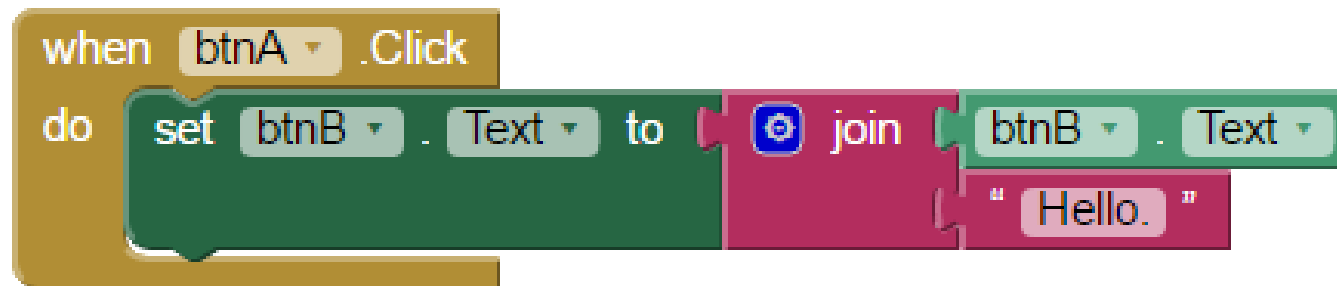
1.3 Programming in App Inventor

- Now it's time to create the program
- The program will perform calculation when the user click the button
- We will make use of the Events of the button to create an entry point to the program



1.3 Programming in App Inventor

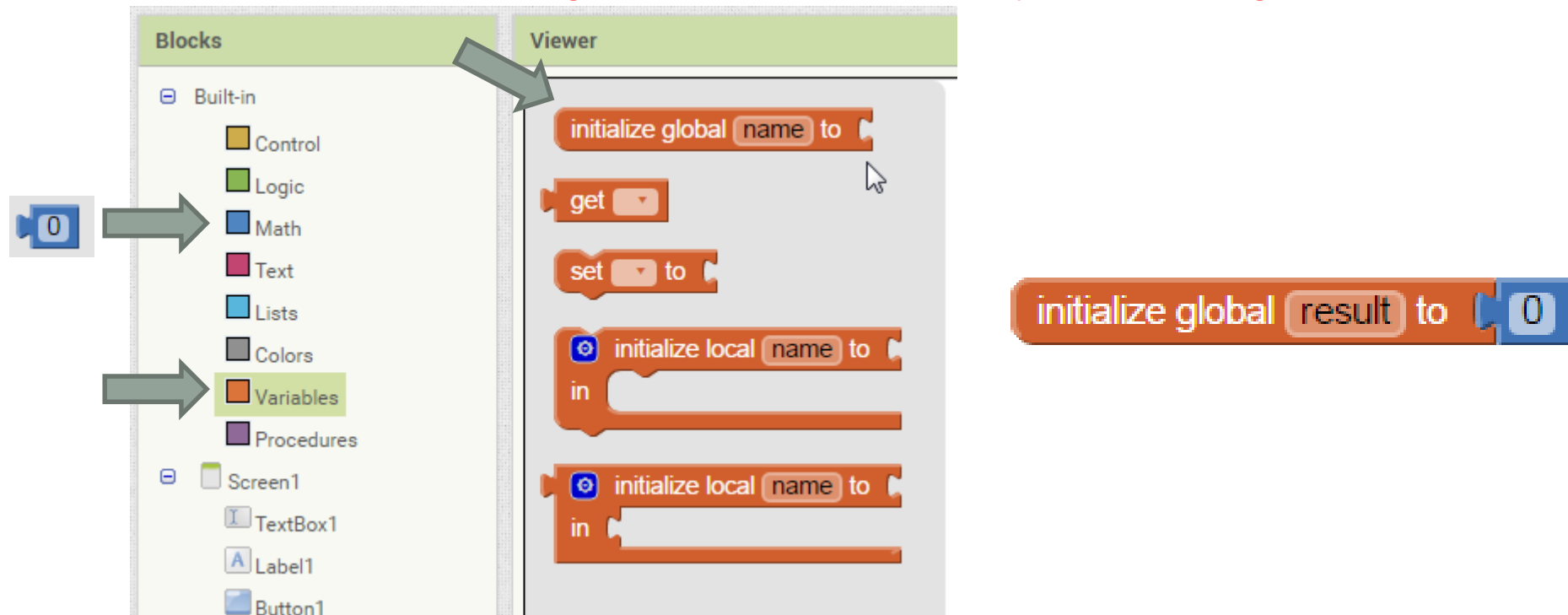
- Accessing and Modifying the properties of components programmatically
 - All properties can be accessed and modified using program statements
 - For example, the following statement change the Text of a button btnB and add the word Hello at the end of the original value



- You may try this sample by downloading the file Properties_Demo.aia from the course web page

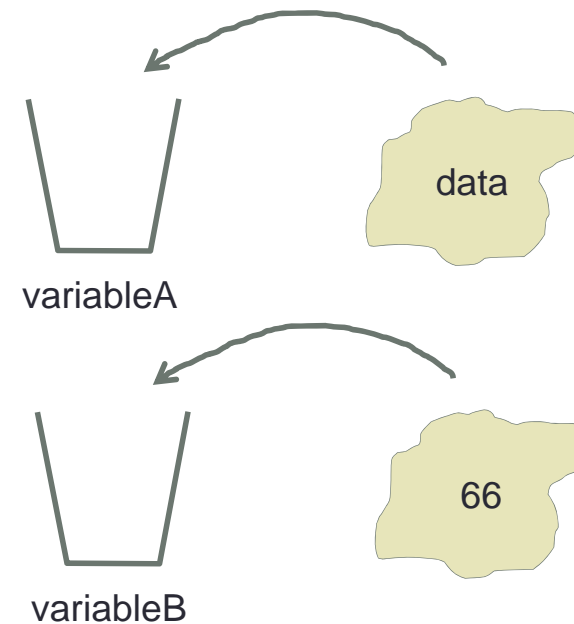
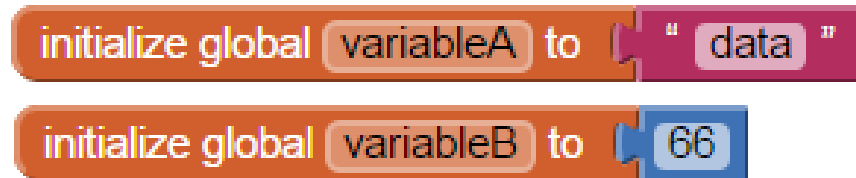
1.3 Programming in App Inventor

- Now, it's time to add the logic to this event
- To perform calculation, we have to perform mathematical operation and store the result in a **variable**
- A variable in a program is a named memory slot for storing values temporarily
- To use a variable in a program, we must set it up by the following blocks first



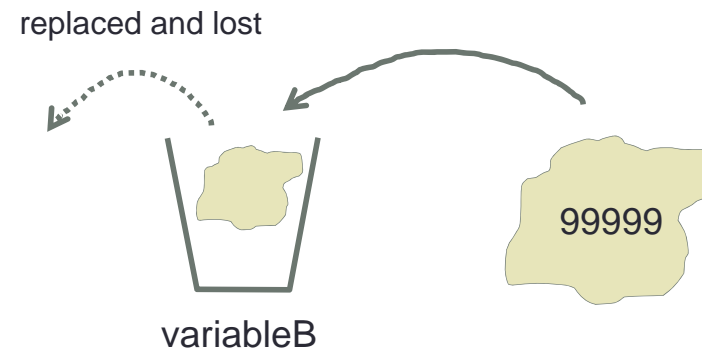
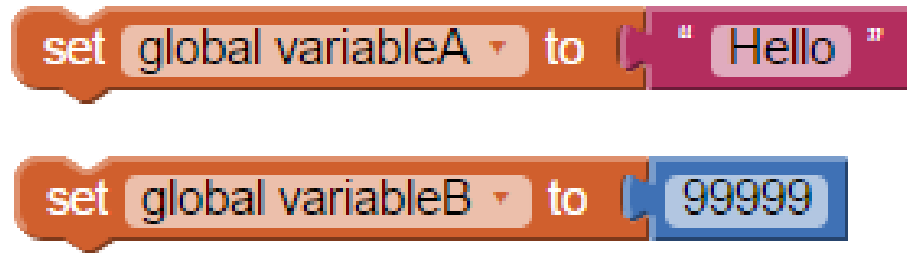
1.3 Programming in App Inventor

- Variables Initialization



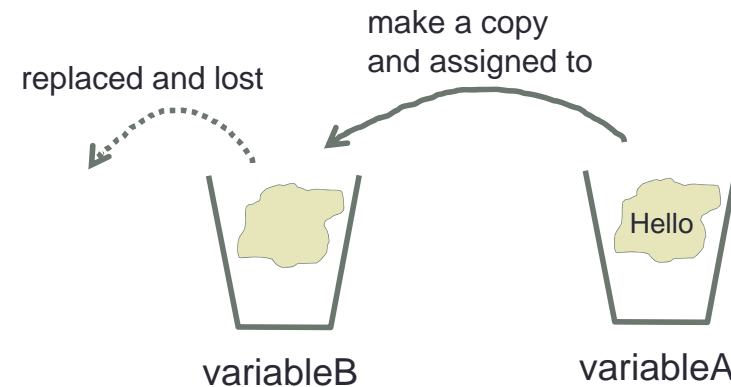
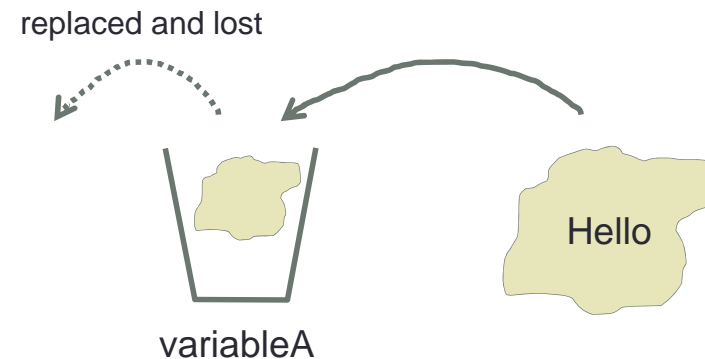
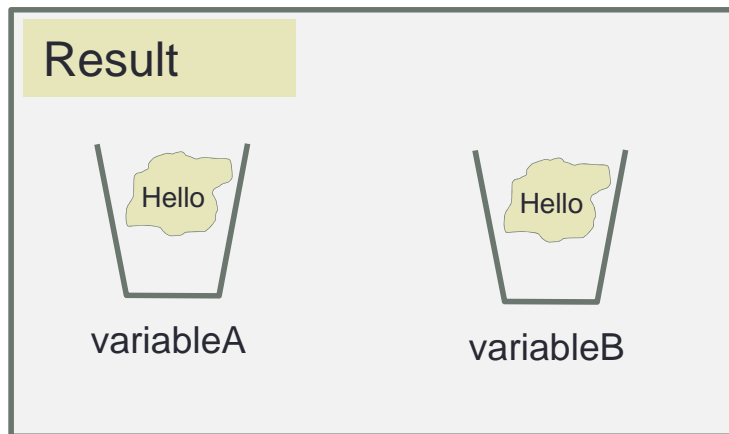
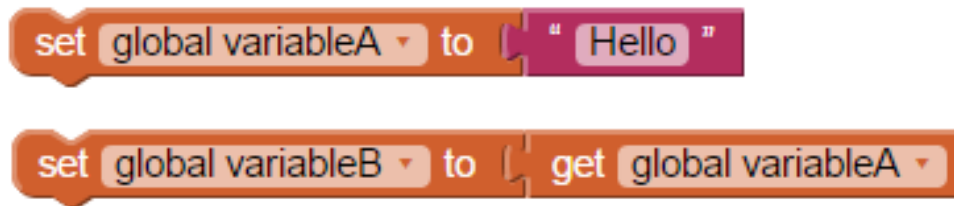
1.3 Programming in App Inventor

- Variables assignment



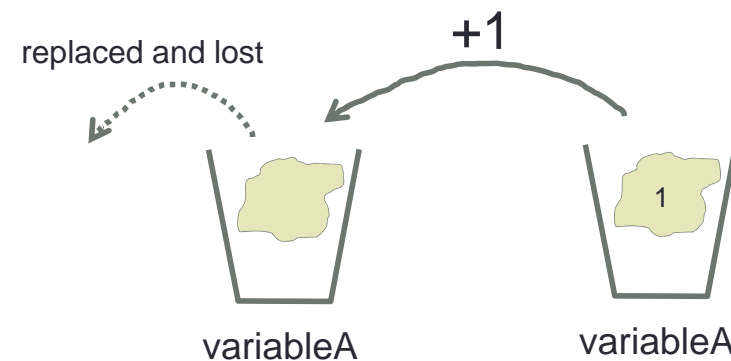
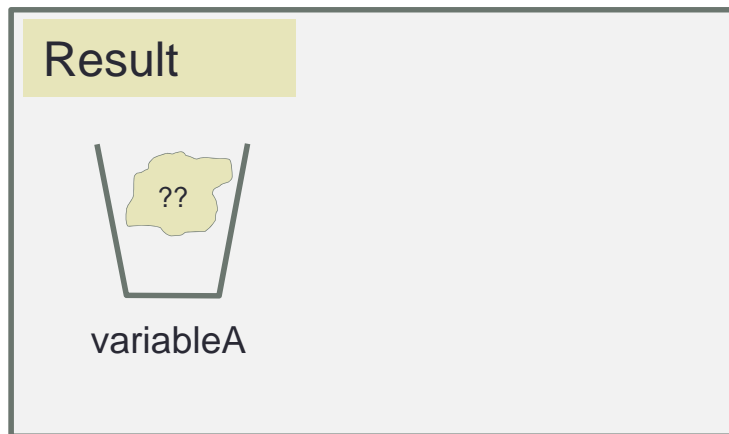
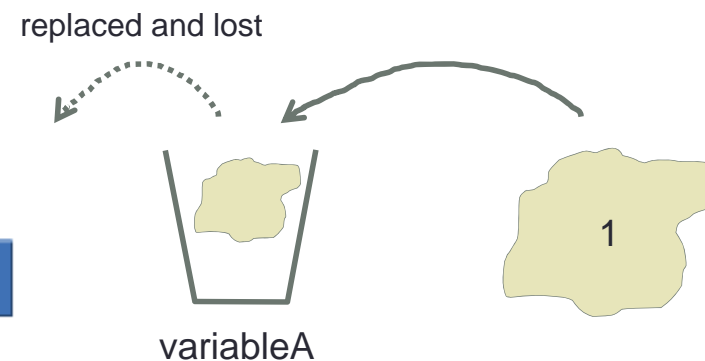
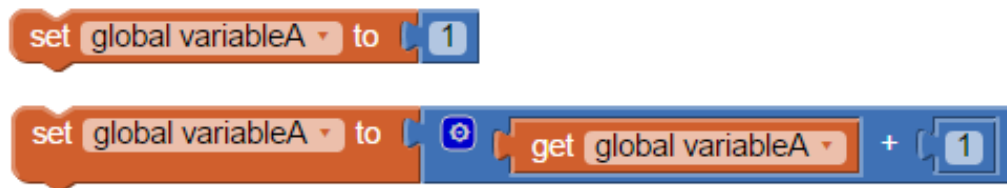
1.3 Programming in App Inventor

- Accessing variables



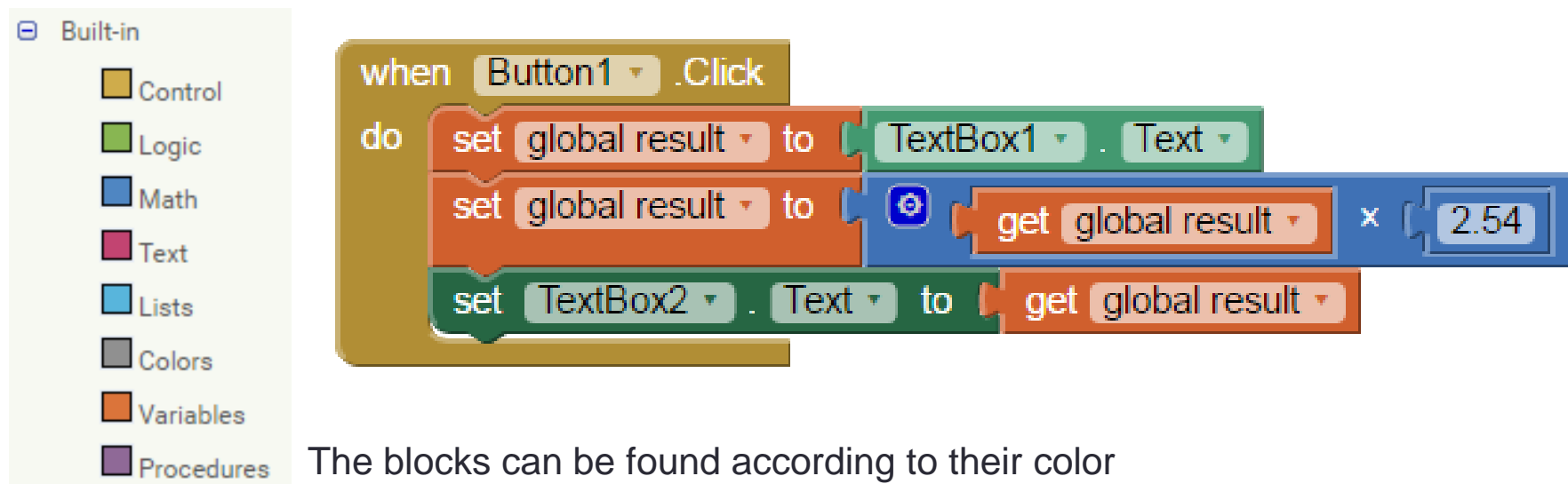
1.3 Programming in App Inventor

- Accumulation with variables



1.3 Programming in App Inventor

- The values stored in the UI (User Interface) components such as Labels and Textboxes can be retrieved and changed using programming statements
- To perform calculation, we can utilize the following logic
 - Retrieve the text stored in the first Textbox and store the value to the variable result
 - Perform calculation on the variable result and store the value back to the variable
 - Change the Text of the second Textbox to the value of the variable result



The image shows the App Inventor interface. On the left is a 'Built-in' block palette with categories: Control (yellow), Logic (green), Math (blue), Text (pink), Lists (light blue), Colors (grey), Variables (orange), and Procedures (purple). On the right, a sequence of code blocks is shown:

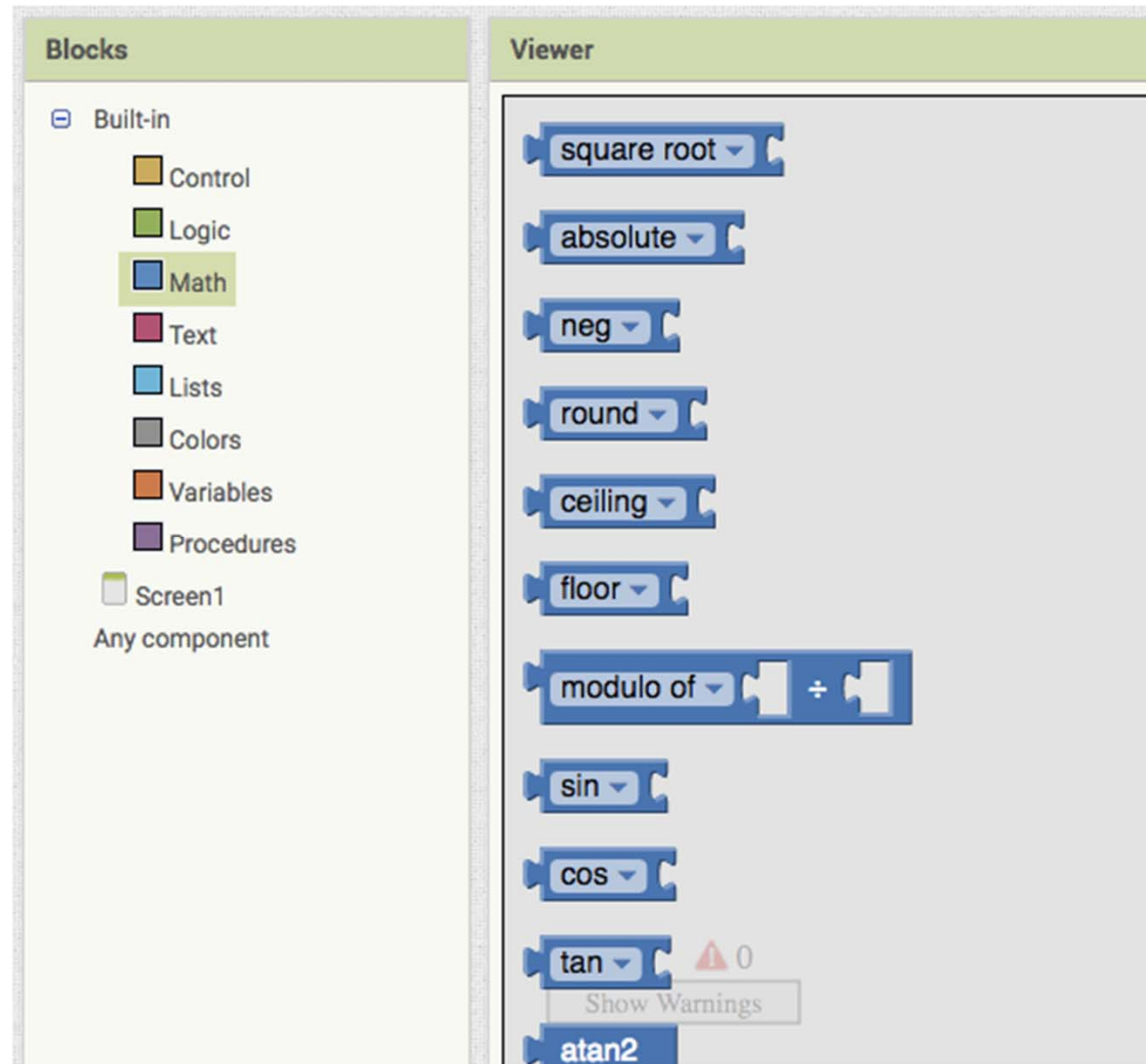
- A yellow 'when Button1 .Click' block.
- A green 'do' block containing three steps:
 - An orange 'set global result to' block connected to a green 'TextBox1 . Text' block.
 - An orange 'set global result to' block connected to a blue 'Math' block (multiplication) which is connected to an orange 'get global result' block and a blue numeric input block containing '2.54'.
 - A green 'set TextBox2 . Text to' block connected to an orange 'get global result' block.

The blocks are color-coded to match the categories in the palette: yellow for Control, green for Logic, orange for Variables, blue for Math, and green for Text.

The blocks can be found according to their color

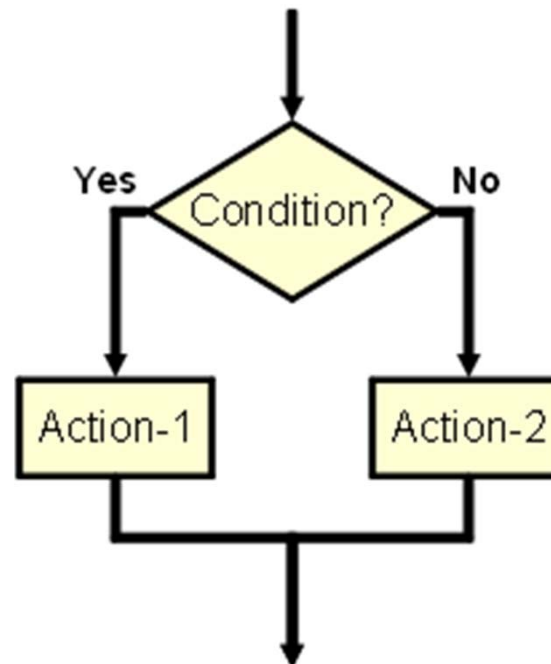
1.3 Programming in App Inventor

- Other mathematical operations
 - Many different types of mathematical operations is possible with AI2
 - Related blocks are placed inside the Math category
 - Some examples of using these mathematical operations are demonstrated on the online video tutorial session in the course web page

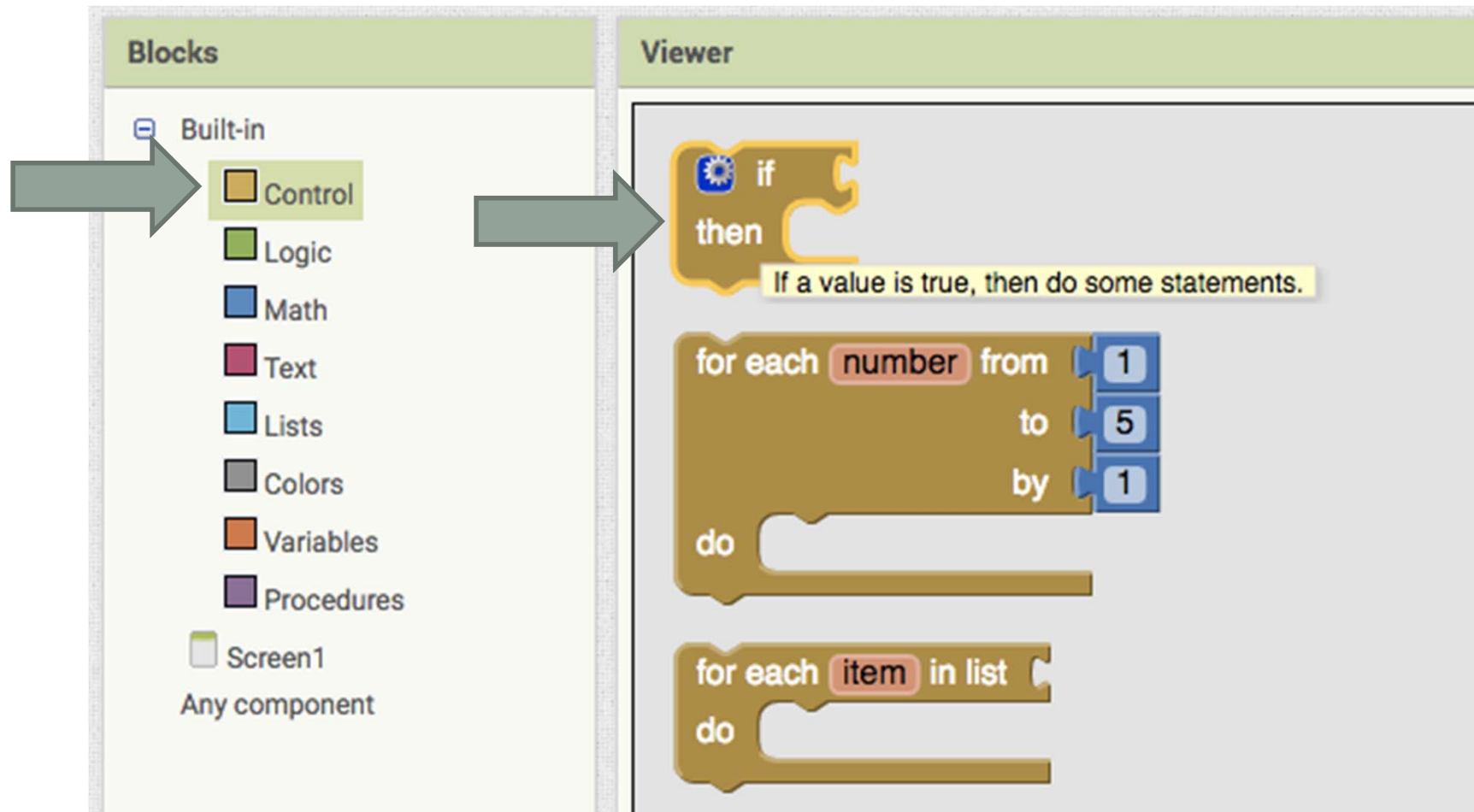


1.3 Programming in App Inventor

- For more complicated logic, for example, the conversion from inches to cm should only be performed when the value input by the user is non-negative, we can make use of simple **conditional statements**

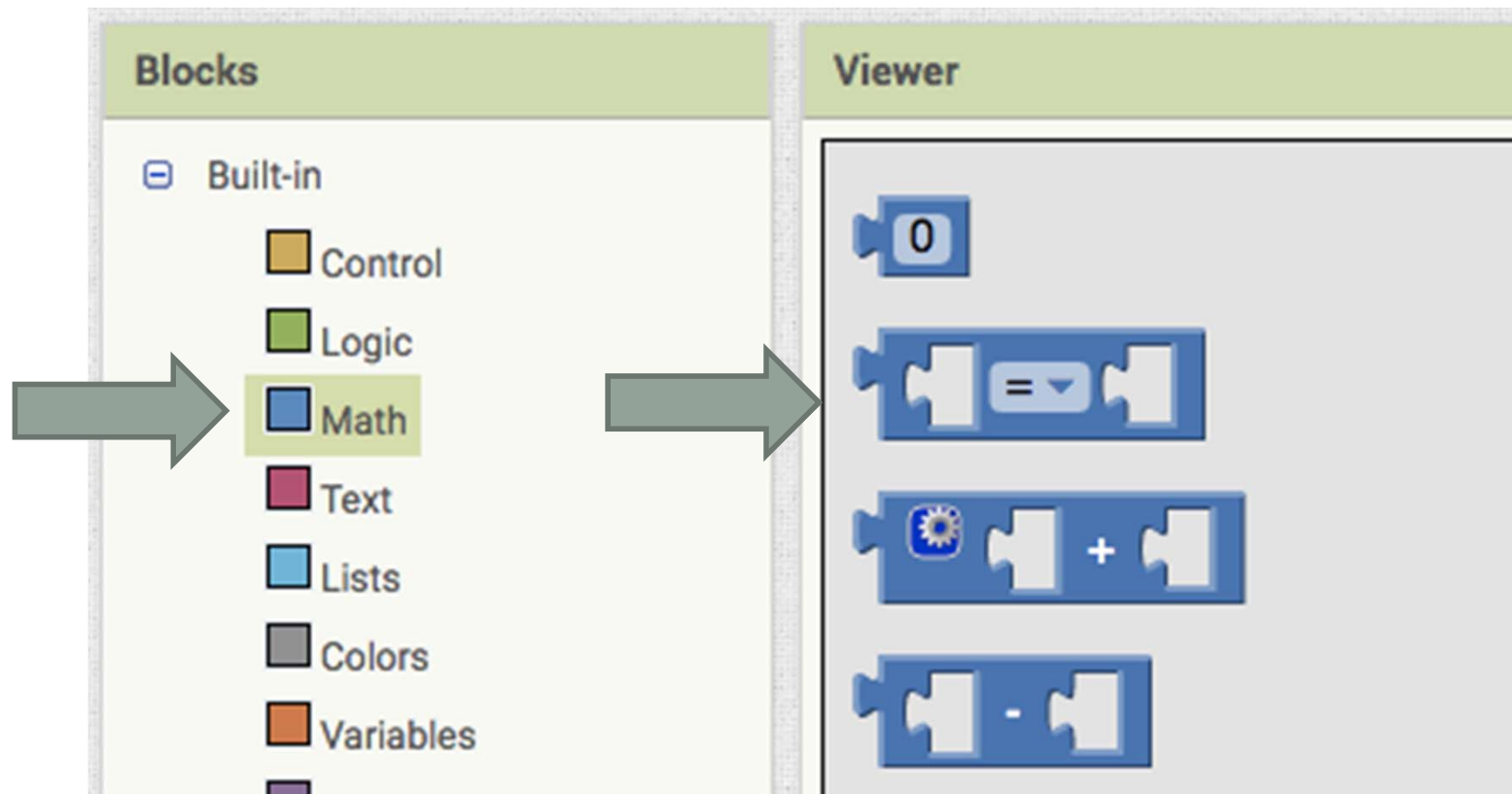


1.3 Programming in App Inventor

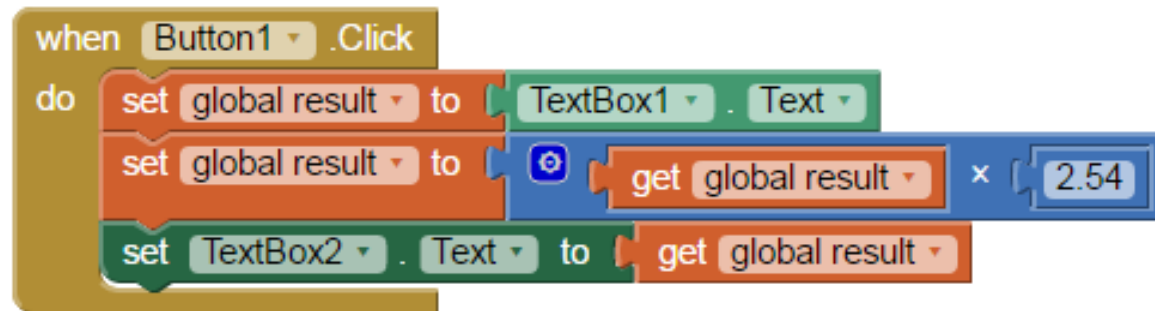


How To Use the If Else Block

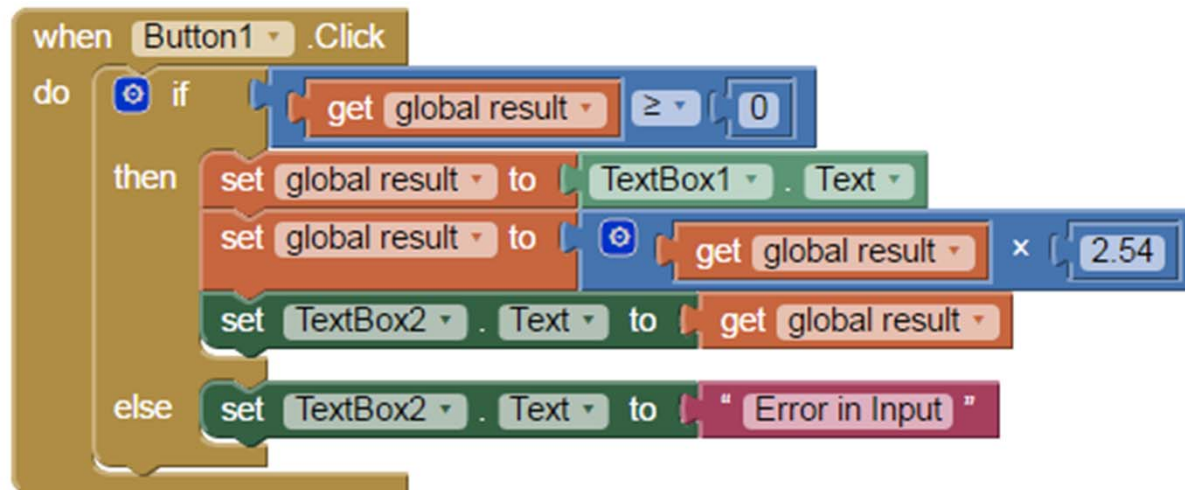
1.3 Programming in App Inventor



1.3 Programming in App Inventor

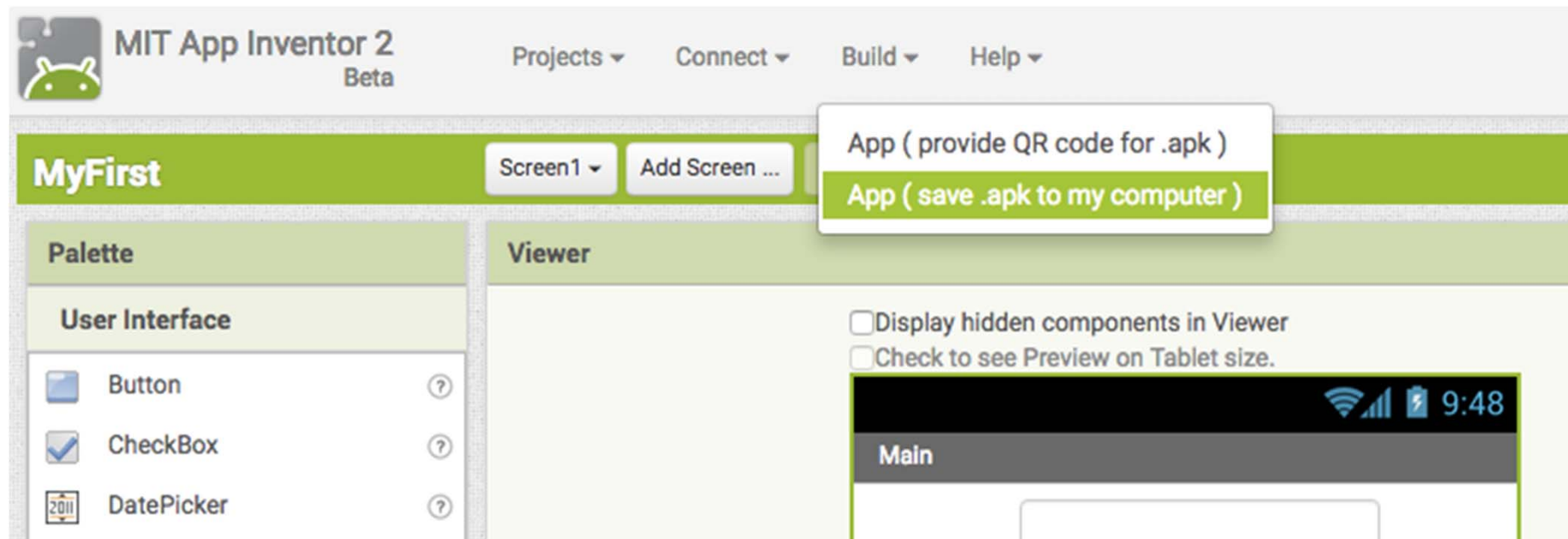


- The following is an updated program for the converter table



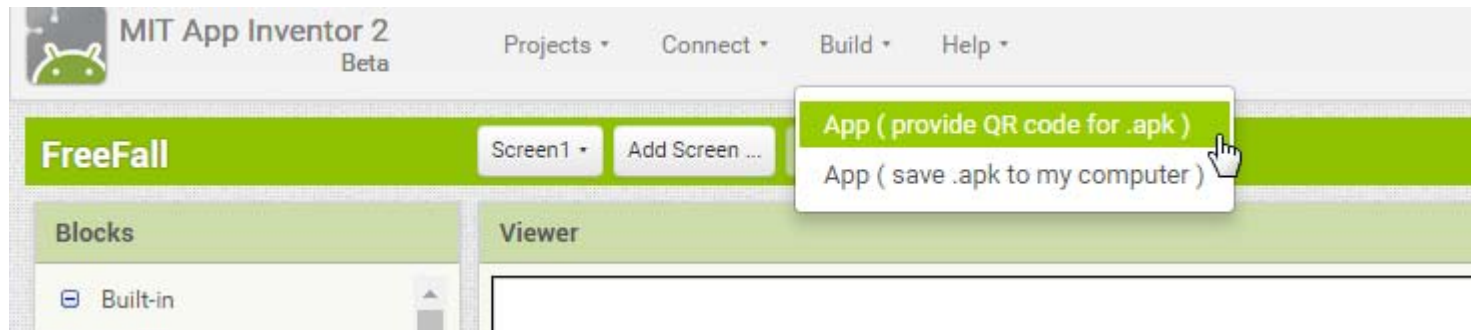
1.3 Programming in App Inventor

- Programs written in AI2 can be packaged as APK files
- APK files are program package that can be installed permanently on an Android Device



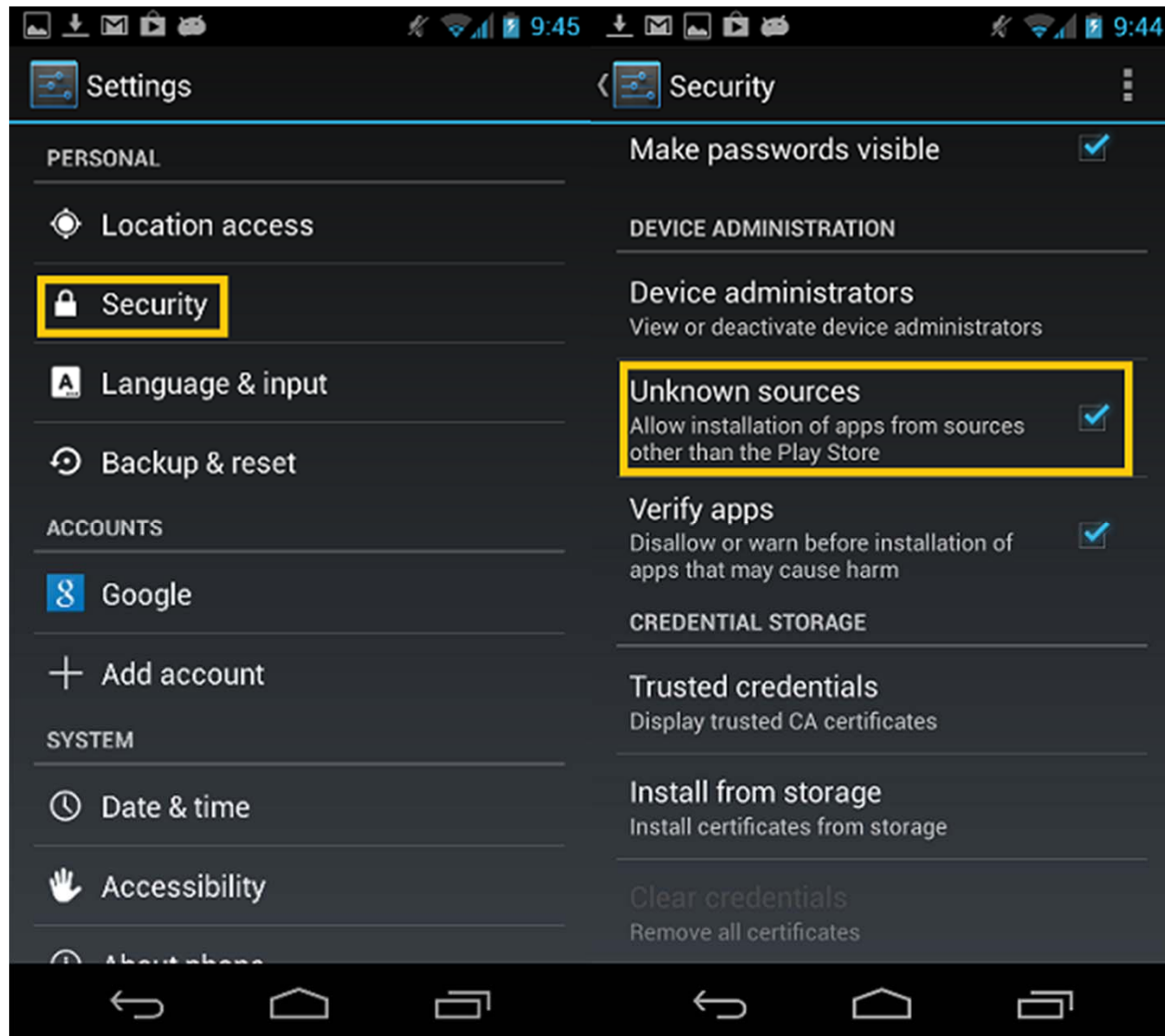
APK Installation

- APK installation can be achieved by directly download the generated APK file from App Inventor web site



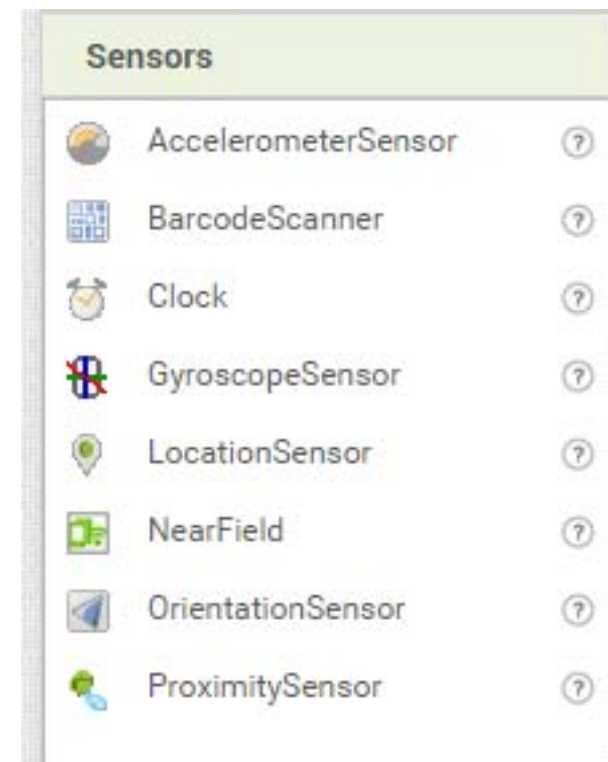
- Be aware of the restrictions on software installation on your mobile device (usually only software from the Google Play store is allowed)

APK Installation



1.4 Sensors programming

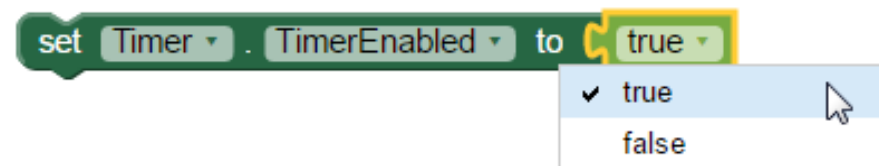
- There are many sensors available on mobile phones, some of them are accessible programmatically through App Inventor
- In this session, we will cover the use of AccelerometerSensor, GyroscopeSensor and the Clock
- Other sensors work similarly



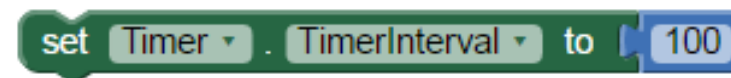
1.4 Sensors programming

- Timer

- Timer is a non-visible component which continuously generate an event every fixed interval
- There are 3 properties that we can set to control its operations
 - TimerEnabled – Controls whether the timer is running or not

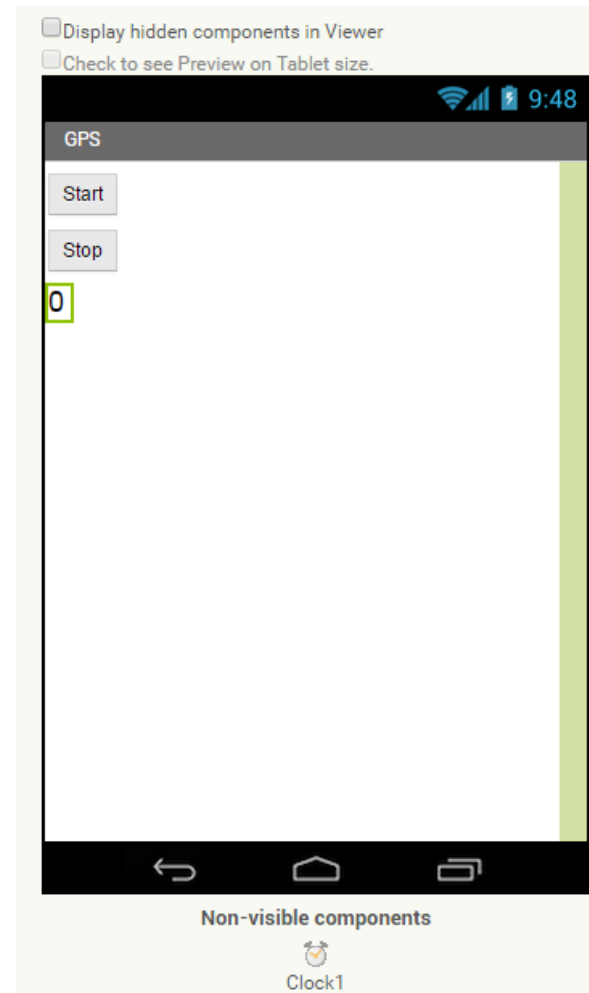


- TimerInterval – Controls how frequent the timer will trigger a Timer event

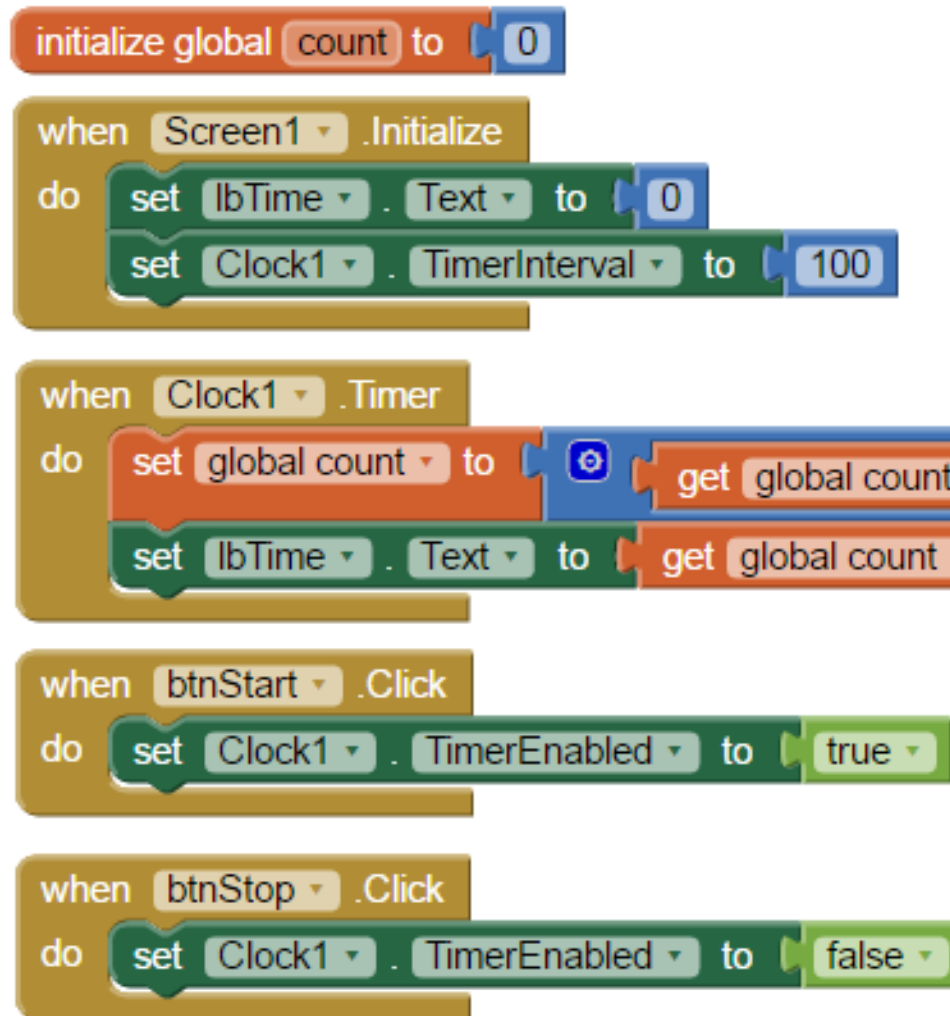


1.4 Sensors programming

- Let's try out a project with a timer
- Download the sample project Timer_Sample.aia from the course website



1.4 Sensors programming



The first event when the apps starts

Setup the properties of the Clock component

The Timer event of the Clock component
This event will take place every 100 ms

The buttons events to enable / disable the Timer

1.4 Accelerometer

- Non-visible component that can detect shaking and measure acceleration approximately in three dimensions using SI units (m/s^2). The components are:
 - xAccel: 0 when the phone is at rest on a flat surface, positive when the phone is tilted to the right (i.e., its left side is raised), and negative when the phone is tilted to the left (i.e., its right side is raised).
 - yAccel: 0 when the phone is at rest on a flat surface, positive when its bottom is raised, and negative when its top is raised.
 - zAccel: Equal to -9.8 (earth's gravity in meters per second per second) when the device is at rest parallel to the ground with the display facing up, 0 when perpendicular to the ground, and $+9.8$ when facing down. The value can also be affected by accelerating it with or against gravity.

1.4 Accelerometer

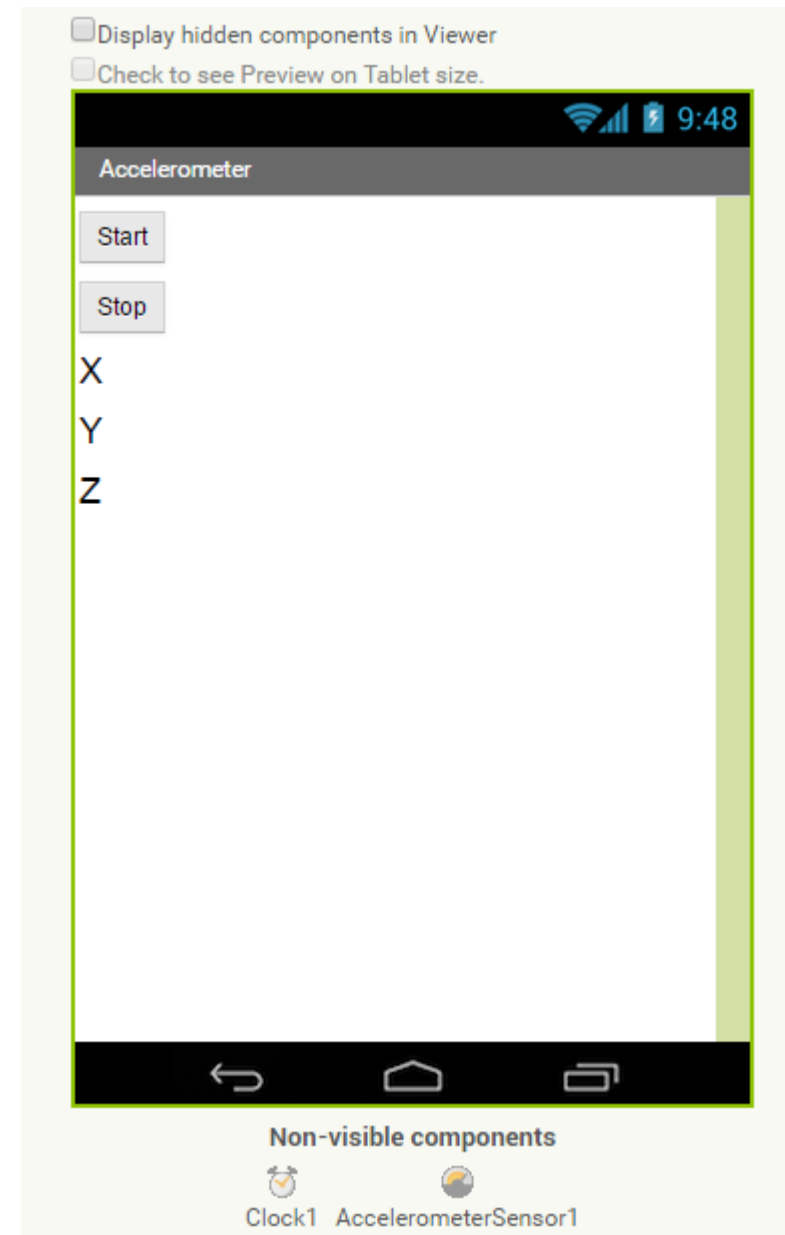
- Properties

- Sensitivity - A number that encodes how sensitive the accelerometer is. The choices are: 1 = weak, 2 = moderate, 3 = strong.
- Available – Whether Accelerometer is available on the device or not
- Enabled – Whether the accelerometer is enabled or not

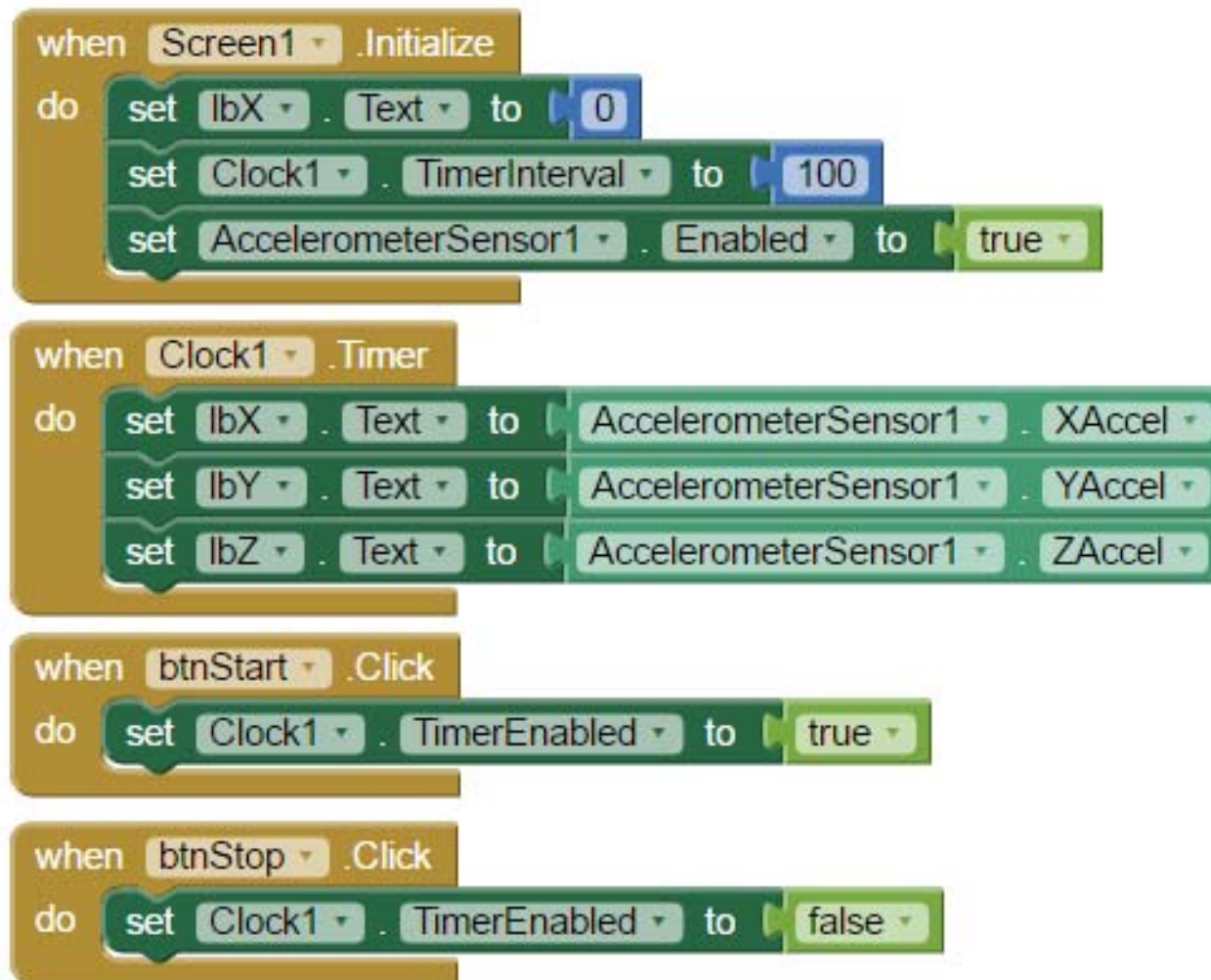


1.4 Accelerometer

- Let's try out a project with a Accelerometer
- Download the sample project Accelerometer_Sample.aia from the course website
- This project make use of the Clock component and the Accelerometer component to monitor the X, Y, Z acceleration of the mobile phone in 100 ms intervals



1.4 Accelerometer

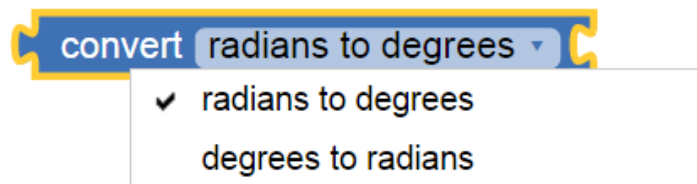


1.4 Gyroscope

- Non-visible component that can measure angular velocity in three dimensions in units of **degrees per second**.
- In order to function, the component must have its Enabled property set to True, and the device must have a gyroscope sensor.
- Properties
 - Available
 - Indicates whether a gyroscope sensor is available.
 - Enabled
 - If enabled, then sensor events will be generated and XAngularVelocity, YAngularVelocity, and ZAngularVelocity properties will have meaningful values.
 - XAngularVelocity
 - The angular velocity around the X axis, in degrees per second.
 - YAngularVelocity
 - The angular velocity around the Y axis, in degrees per second.
 - ZAngularVelocity
 - The angular velocity around the Z axis, in degrees per second.

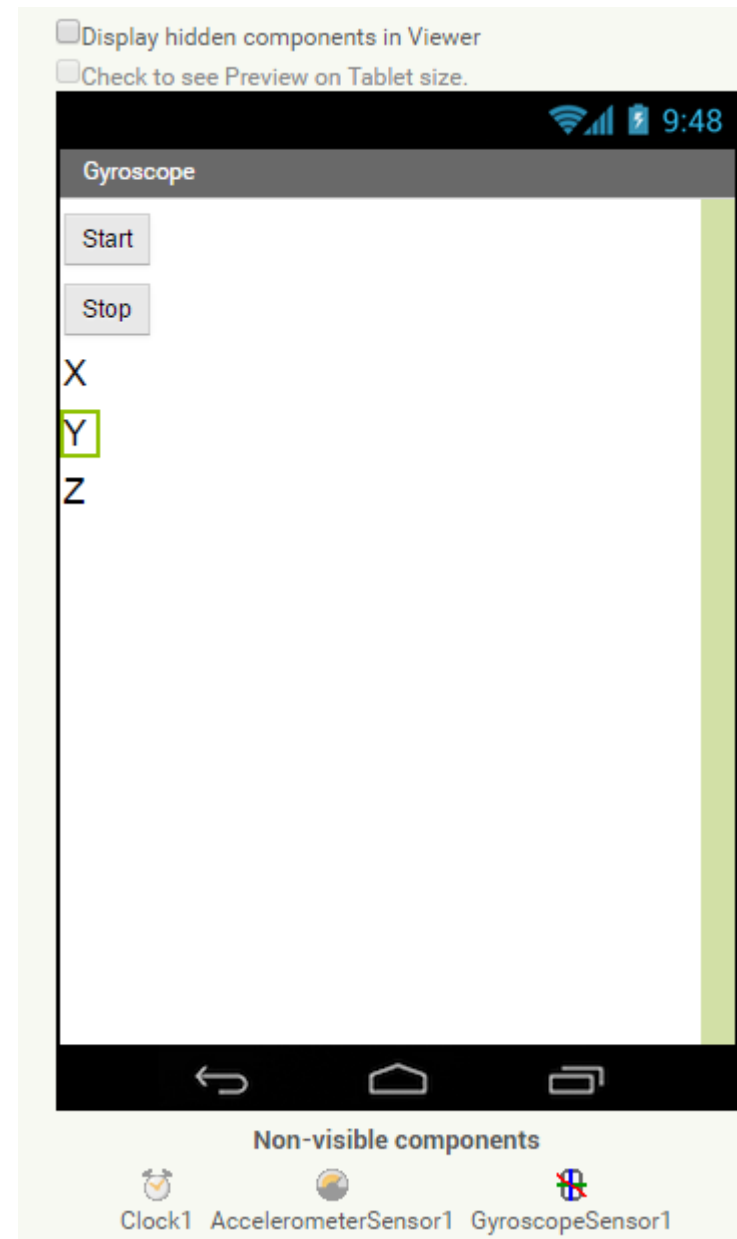
1.4 Gyroscope

- The unit returned from the Gyroscope is degree per second
- The value can be easily turned into radian per second by the following Math block



1.4 Gyroscope

- Let's try out a project with a Gyroscope
- Download the sample project Gyroscope_Sample.aia from the course website
- This project make use of the Clock component and the Gyroscope component to monitor the X, Y, Z angular velocity of the mobile phone in 100 ms intervals



1.4 Gyroscope

```
when Screen1.Initialize  
do set Clock1.TimerInterval to 100
```

```
when Clock1.Timer  
do  
  set lbX.Text to convert degrees to radians GyroscopeSensor1.XAngularVelocity  
  set lbY.Text to convert degrees to radians GyroscopeSensor1.YAngularVelocity  
  set lbZ.Text to convert degrees to radians GyroscopeSensor1.ZAngularVelocity
```

```
when btnStart.Click  
do set Clock1.TimerEnabled to true
```

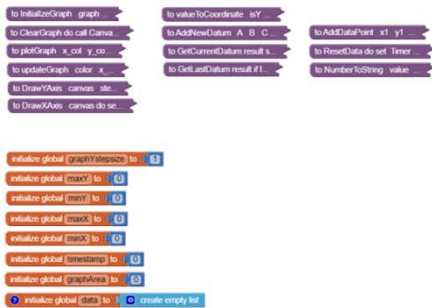
```
when btnStop.Click  
do set Clock1.TimerEnabled to false
```

1.4 Putting things together

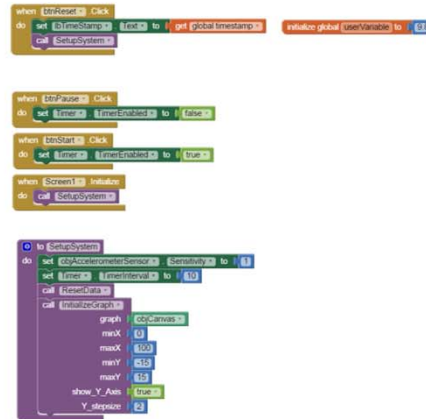
- Sensors and the Clock component can be put together and create a unique app for a specific experiment
- To help teachers make good use of the programming environment, a template app is written for you
 - CAITE_Physics_Template1.aia
- This template provides a easier to use interface and graphing capabilities



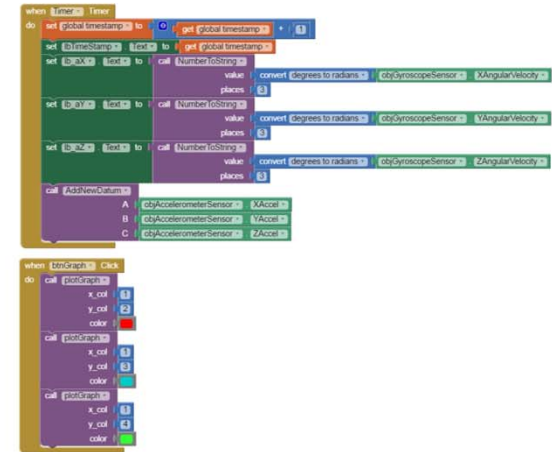
1.4 Putting things together



System pre-written
procedures
DON'T TOUCH
but welcome to study if
you want

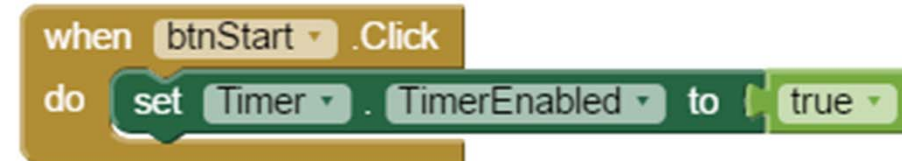
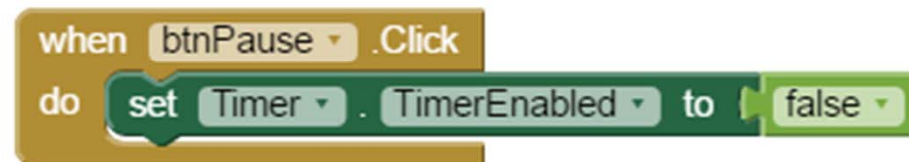
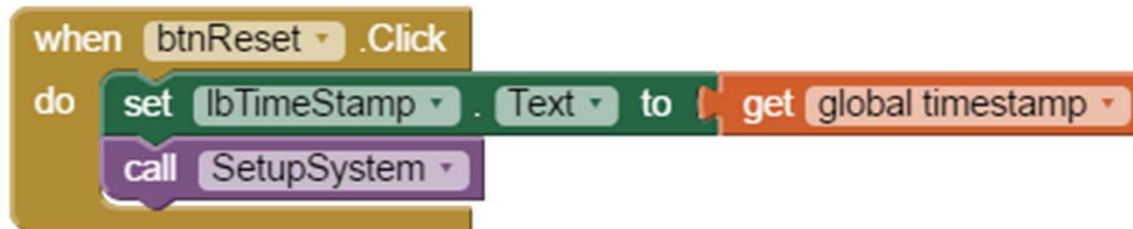


Programs to be manipulated by teachers



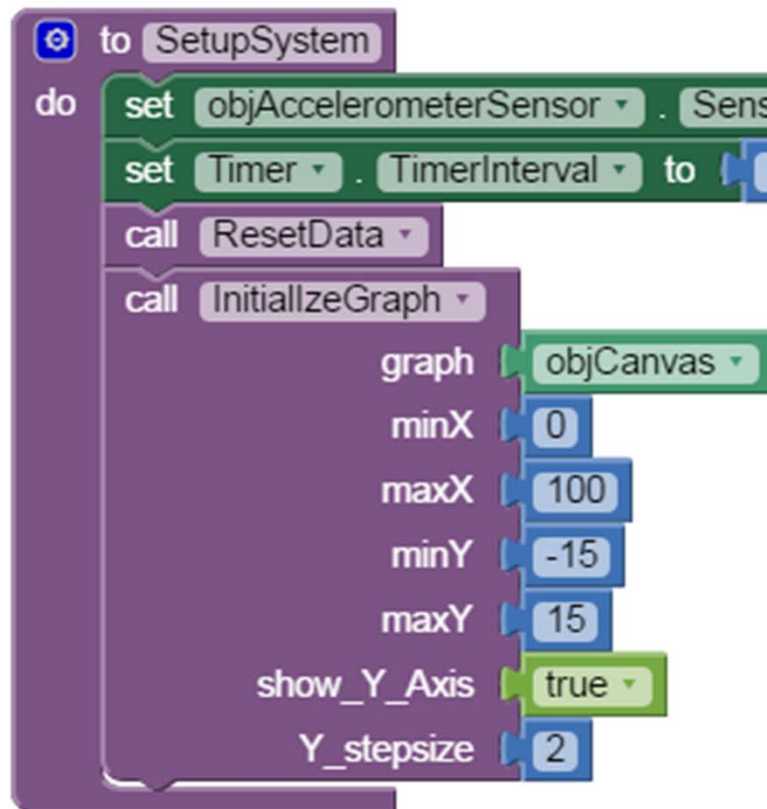
The program comes with many pre-written procedures that you may use directly for manipulating graphs output and later export to CSV or sharing data with email

1.4 Putting things together



Event handler for the
button `btnReset`,
`btnPause`, `btnStart`
and
`Screen1.Initialize`

1.4 Putting things together



The Procedure **SetupSystem** is used to setup the parameters for graphing and the properties for the sensors

Setup the properties of the Accelerometer and the Timer

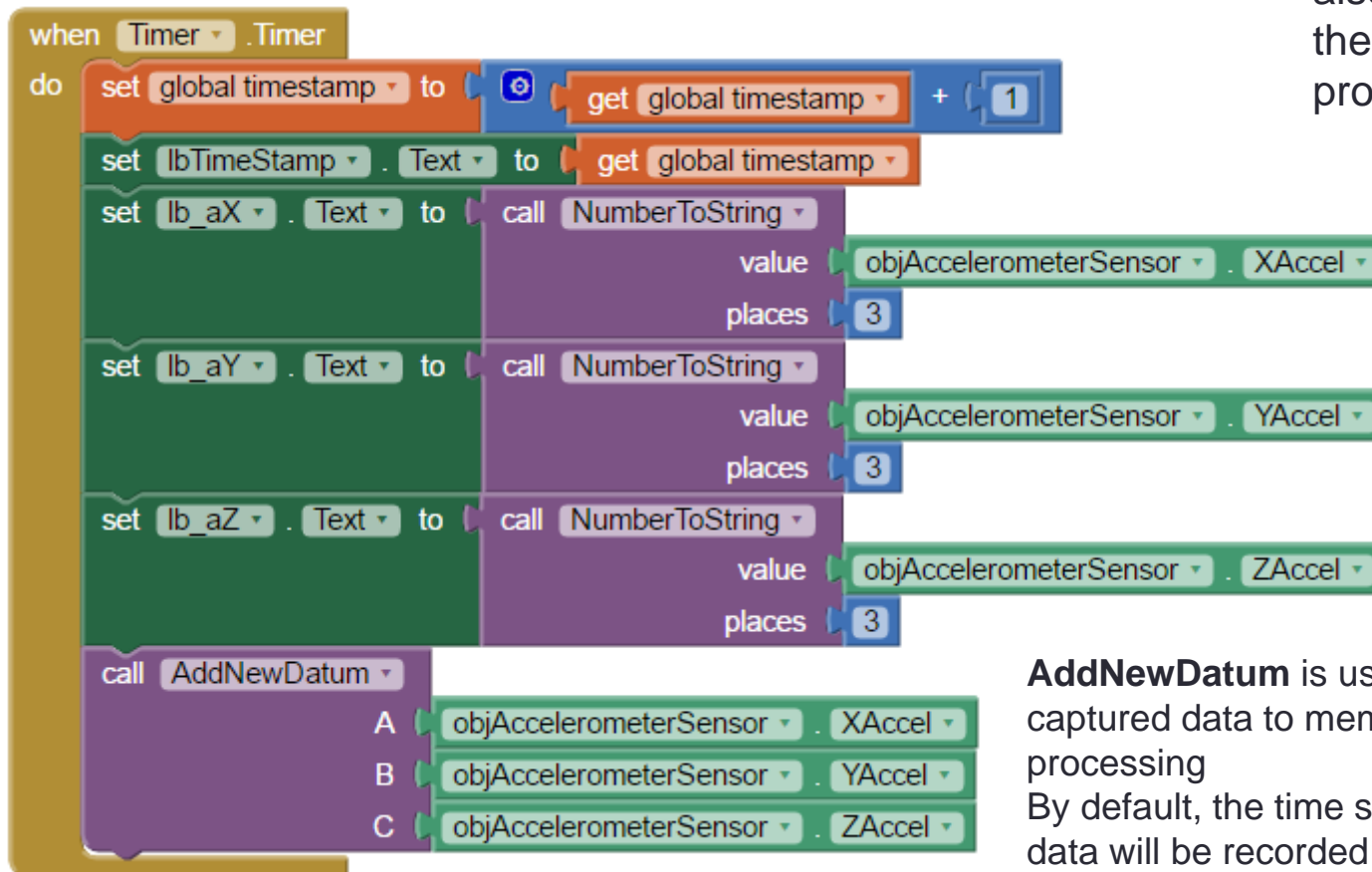
ResetData will clear all the recorded data

InitializeGraph setups the parameters for graphing

- graph – which Canvas component will be used for graphing
- minX – the minimum value on the X-axis
- maxX – the maximum value on the X-axis
- minY – the minimum value of the Y-axis
- maxY – the maximum value of the Y-axis
- show_Y_Axis – whether to draw the Y-Axis
- Y_stepsize – the step size between marking on the Y-Axis

1.4 Putting things together

The Timer Event is the main event for collecting data, it will display the data on the Labels and also add the data items to the memory for future processing

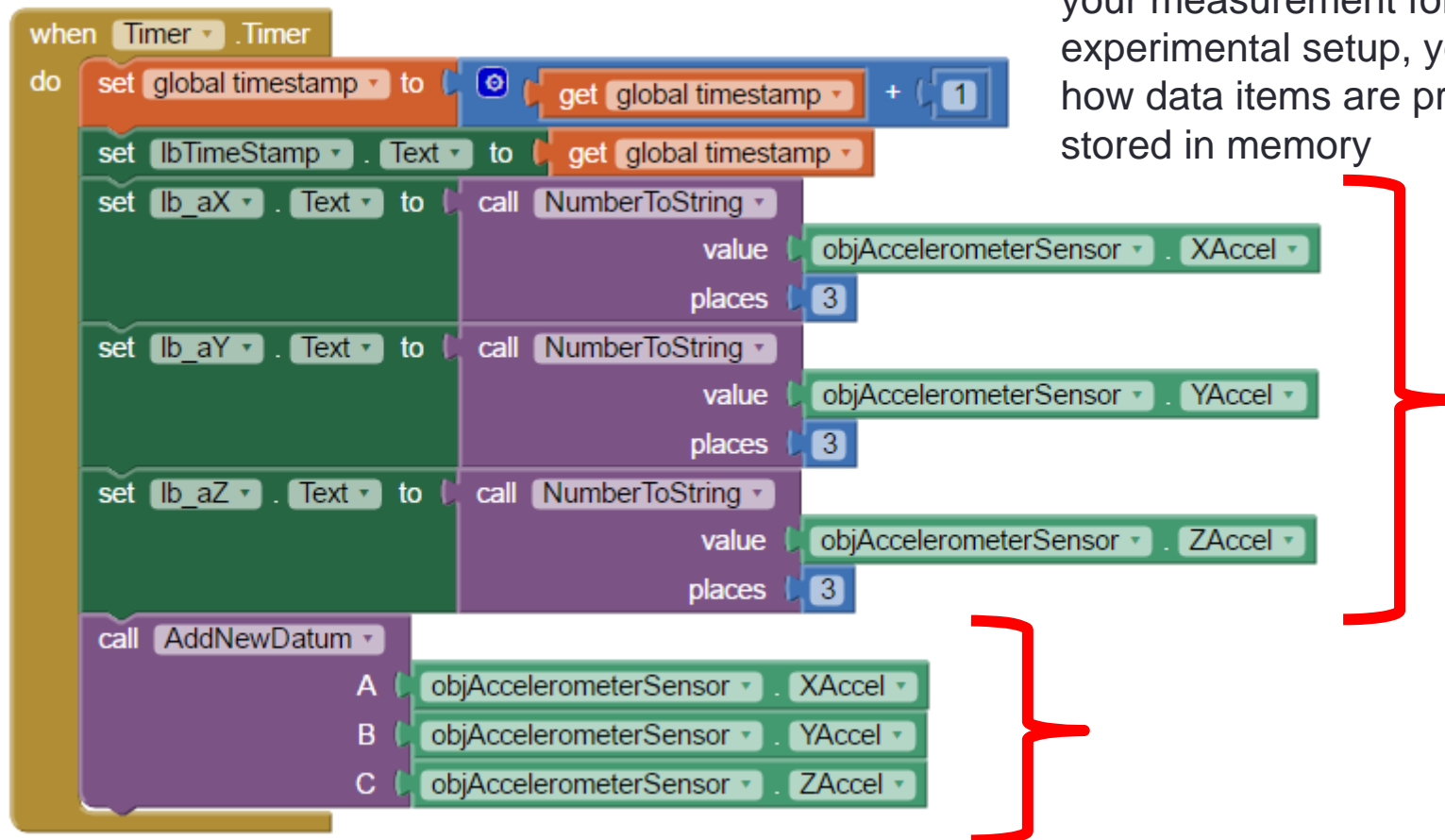


AddNewDatum is used to store the captured data to memory for future processing. By default, the time stamp and three set of data will be recorded (A, B and C)

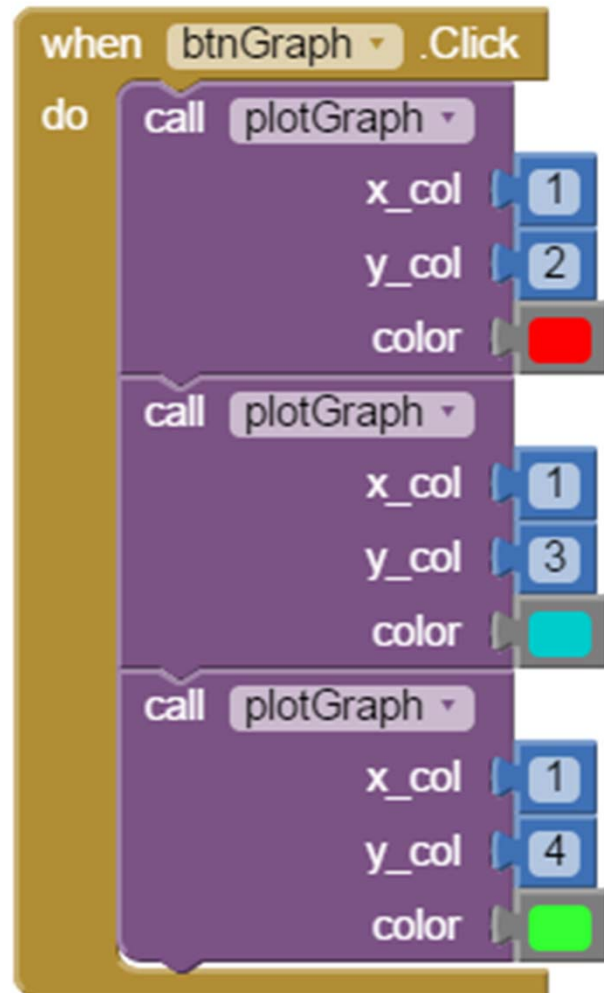
1.4 Putting things together

Modifying the program

For example, you may want to deduct the static gravitational g from your measurement for a specific experimental setup, you may modify how data items are presented and stored in memory



1.4 Putting things together



The btnGraph.Click Event is used to plot the graph from the stored data in memory. (Live plot is only possible if the sampling rate is low e.g. 10Hz, a big limit for App Inventor)

plotGraph is used to plot the graph of two data series from the data stored in memory. You may specify the column and color used for plotting.

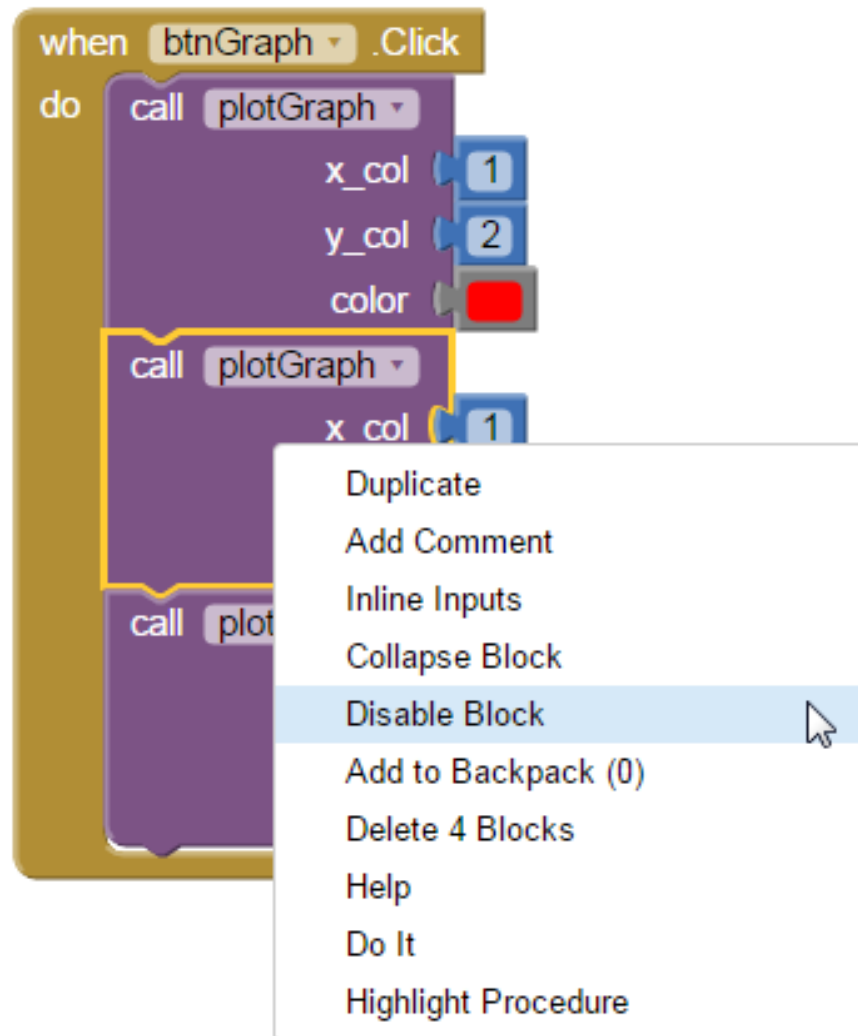
Column 1 – time stamp

Column 2 – Data series A

Column 3 – Data series B

Column 4 – Data series C

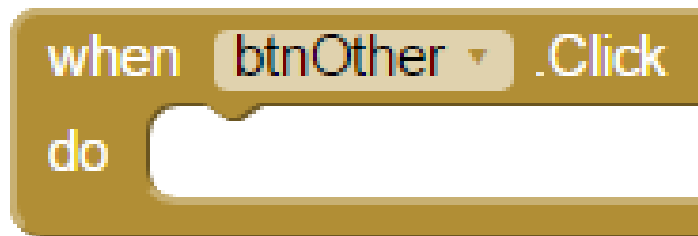
1.4 Putting things together



Modifying the program

If you just want to plot the XAccel of the records, you can simply disable or even remove the block **plotGraph**

1.4 Putting things together



Modifying the program

Additional button btnOther for other purposes

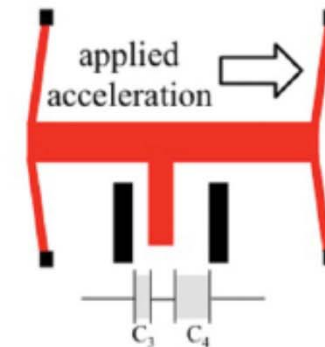
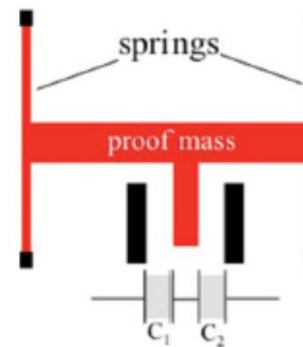
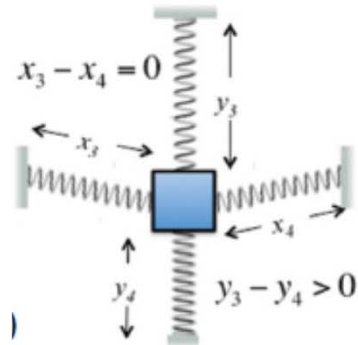
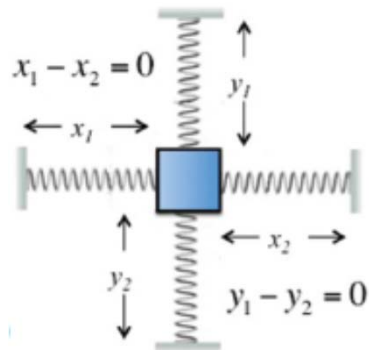
1.5.1 Pedagogical Design, Rationale and Implementation

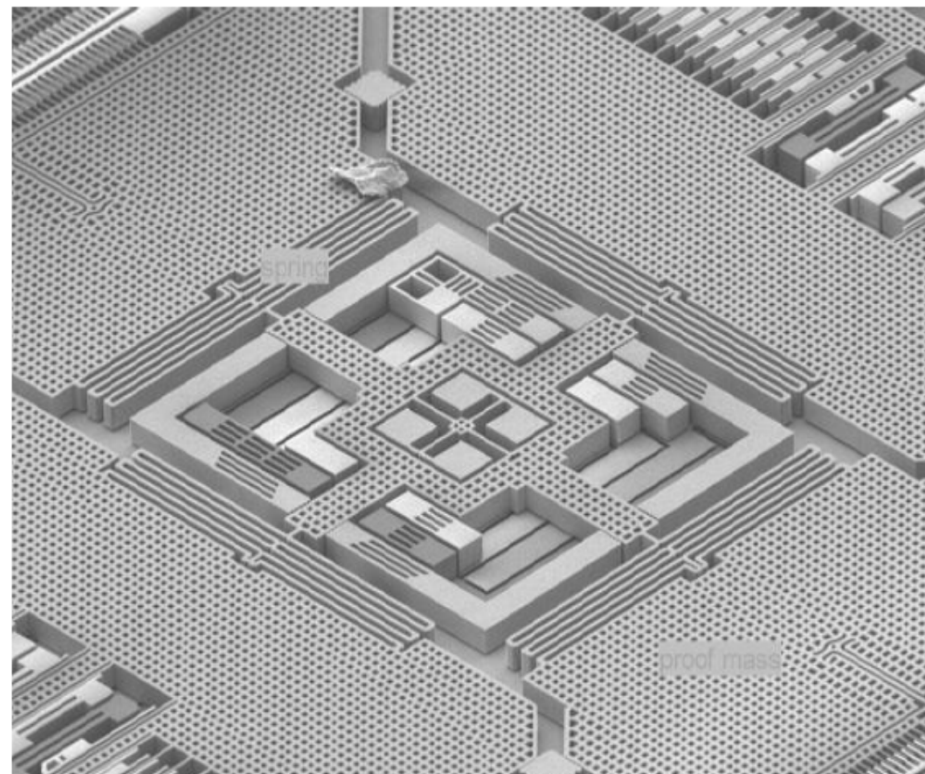
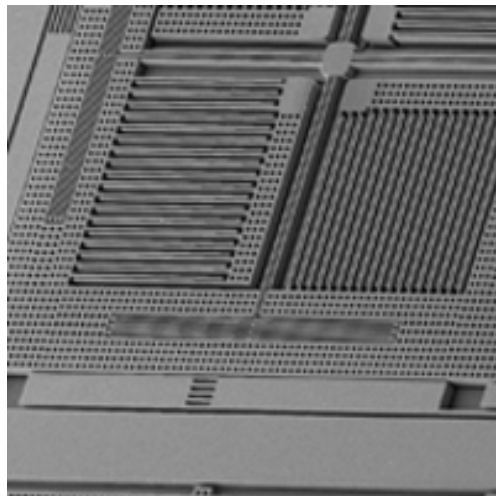
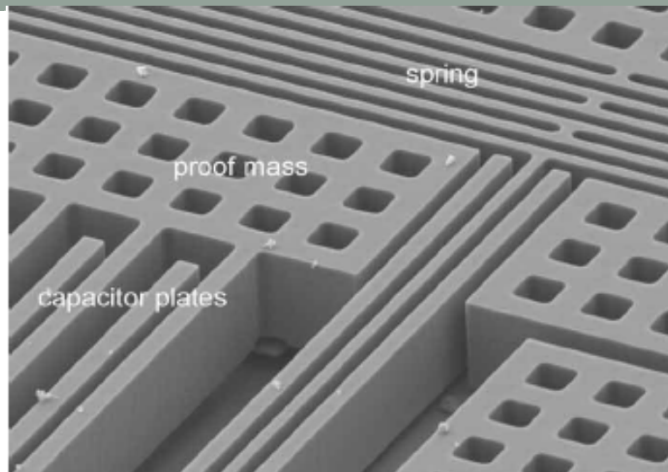
- Two sensors on App Inventor: accelerometer and gyroscope
- Facilitate students to learn Physics
- Help students to verify the physics laws
- Enhance experiment practical skills and data-handling skills
- Promote investigative study and Inquiry-based Learning

1.5.2 ACCELEROMETER

2.5 acceleration sensor

- Acceleration sensor
- $a_x = a_z = 0, a_y = -g$



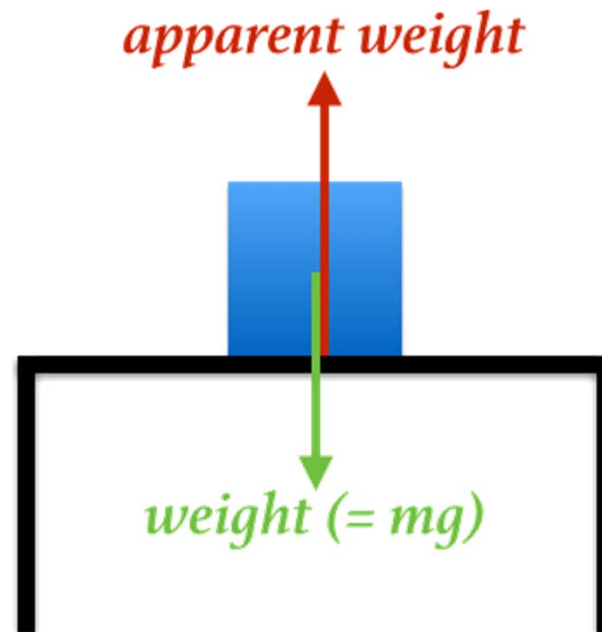


**Chip image:
STMicroelectronics
LIS331DLH**

[Video of Accelerometer https://www.youtube.com/watch?v=KZVgKu6v808](https://www.youtube.com/watch?v=KZVgKu6v808)

Experiment 1: Measurement of apparent weight

- Program can be simplified to obtain data only in one direction of acceleration
- Program to eliminate pseudo g



Experiment 1: Measurement of apparent weight

- Pseudo gravity
- Give the value g even at rest
- Eliminate the pseudo g on the program
- Reduce confusion
- Further verification from the data : $g = 2s/t^2$



Experiment 1: Measurement of apparent weight

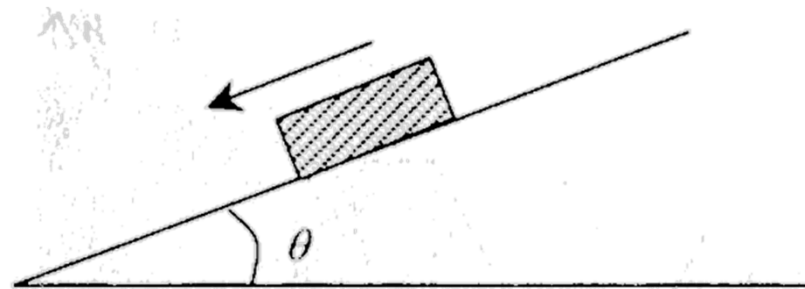
- Amusement park measurement
- Lift, Roller Coaster
- Relate to daily life experience



Analyzing elevator oscillation with the smartphone acceleration sensors
Phys. Teach. **52**, 55 (2014); 10.1119/1.4849161

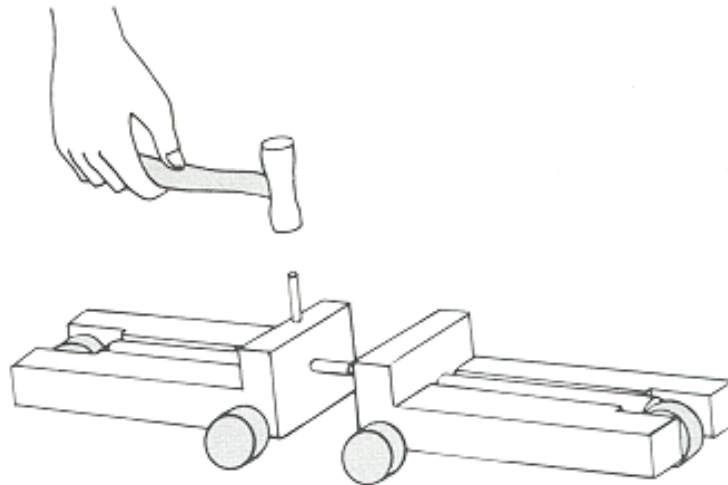
Experiment 2: Measurement of “diluted gravity” ($g \sin \theta$)

- Measurement of acceleration along a slope
- Relationship between acceleration with different angle of inclination is verified
- Friction



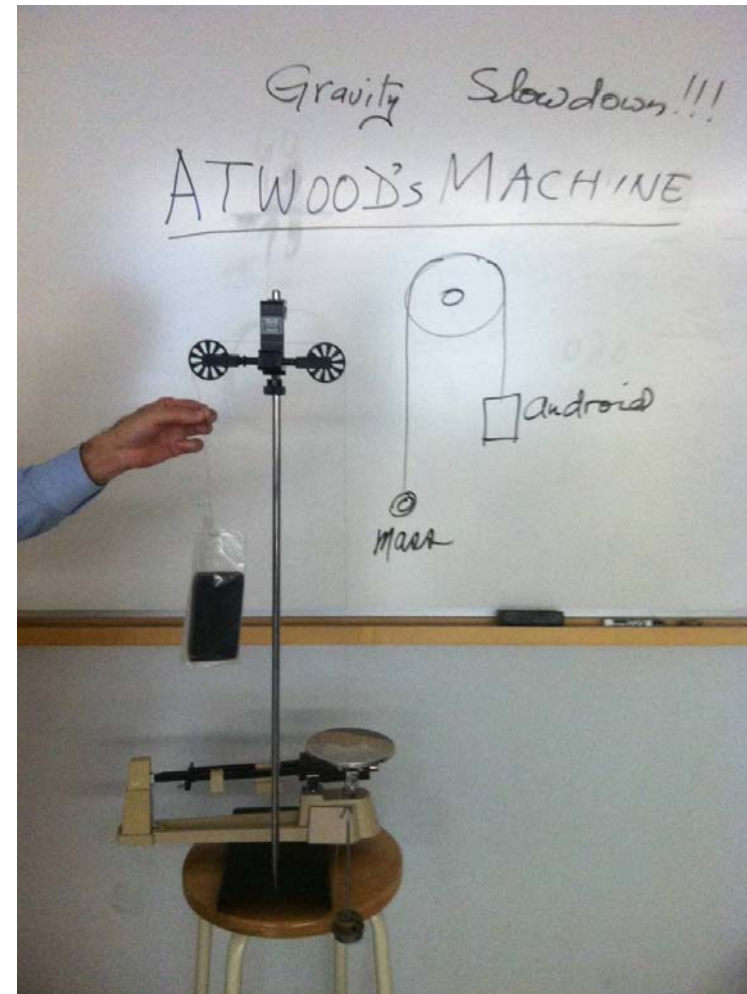
Experiment 3: Newton's 3rd law

- Use accelerometer data to verify Newton's 3rd law
- Net forces on different masses by measuring their accelerations



Experiment 4: Atwood machine

- Verify the relationship of accelerations of different masses
- Acceleration along one axis



Experiment 5: acceleration of pendulum

- To determine the period of oscillation from the acceleration-time graph
- Verify the relationship between the period and the length of the pendulum

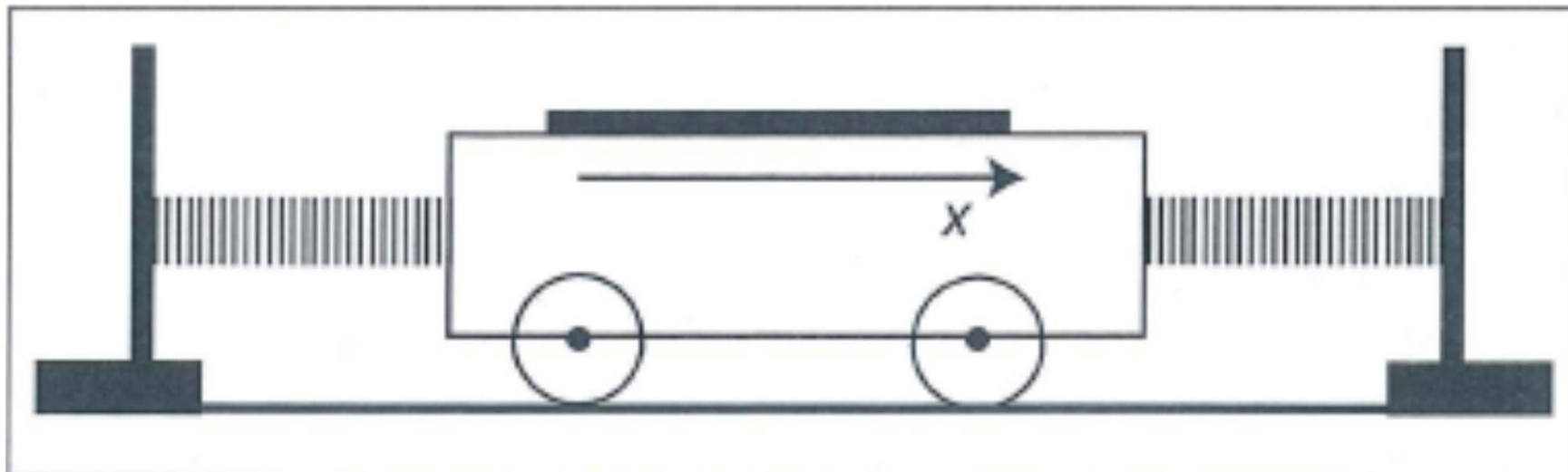


A physical pendulum

Analyzing simple pendulum phenomena with a smartphone acceleration sensor
Phys. Teach. **50**, 439 (2012); 10.1119/1.4752056

Experiment 6: acceleration of oscillations

- Damped SHM
- Measure acceleration along one axis



Using a mobile phone acceleration sensor in physics experiments on free and damped harmonic oscillations
Am. J. Phys. **81**, 472 (2013); 10.1119/1.4793438

1.5.2 GYROSCOPE

Gyroscope

- Angular velocity ω with rotational axis along z direction
- Centripetal acceleration a on xy -plane
- Verify the relationship between a and ω^2

Gyroscope

- Program is set to record angular speed along z axis and acceleration on xy -plane

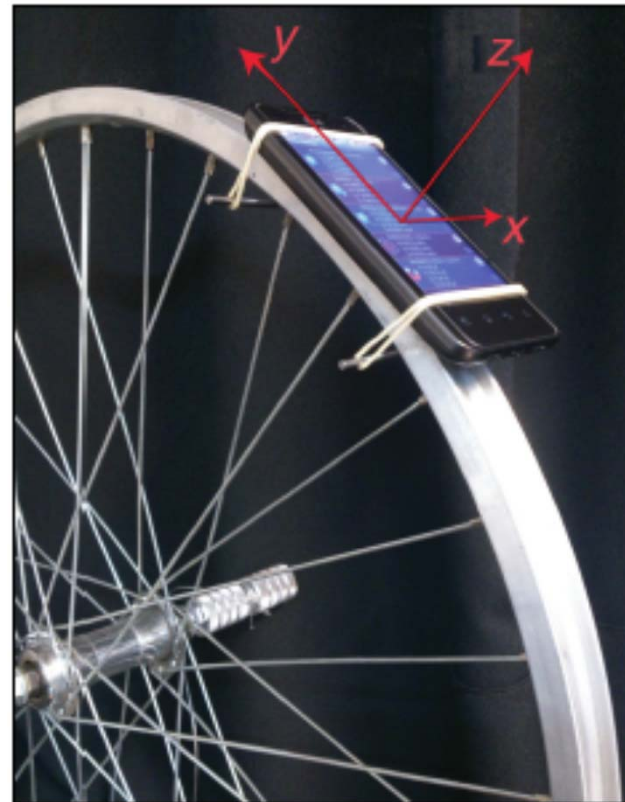


Analyzing radial acceleration with a smartphone acceleration sensor
Phys. Teach. **51**, 182 (2013); 10.1119/1.4792021

Non-uniform circular motion

- Further investigation
- Verify the relationship

$\omega^2_{\text{bottom}} - \omega^2_{\text{top}}$ is a constant



Homework

- Complete the assignment before attending Session 2
- A brief lesson plan
- Describe how to use mobile apps for investigative study of Physics or Science subjects

Homework

- Some participants might be invited to share their plan in Session 2
- A proposal template (worksheet) will be given
- Email: phyteacher2015@gmail.com
- Title: C2_AA_Physics
- File: C2_AA_(Your Name).docx
- Submit through Schoology Assignment Collection Box or email