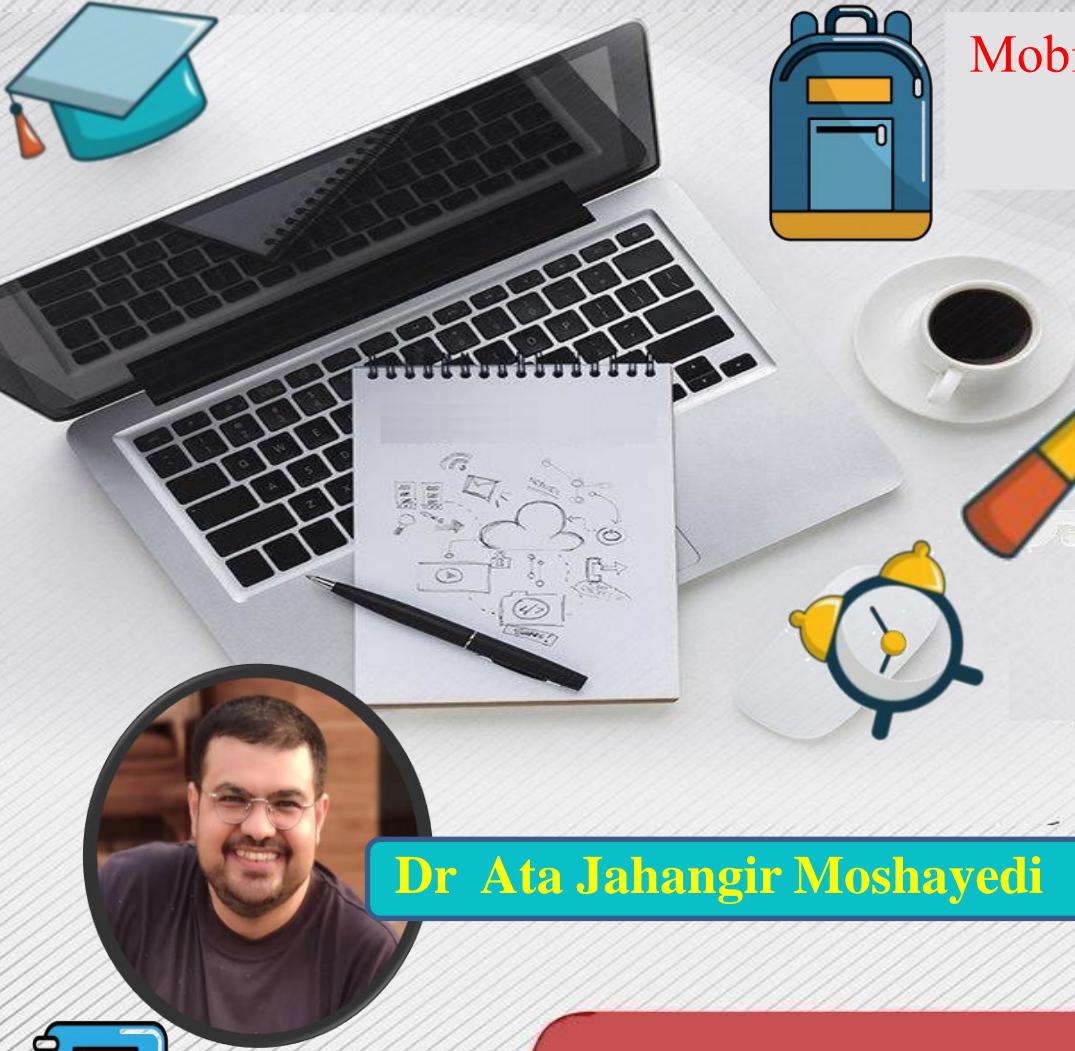




江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



Dr Ata Jahangir Moshayedi



EMAIL: ajm@jxust.edu.cn

Mobile application development

移动应用开发



Lecture 023:
Using the Activity Starter
Component and some examples

Prof Associate ,
School of information engineering Jiangxi
university of science and technology, China

Autumn _2021



江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

LECTURE 023:

APP Inventor _ enviroment

**Using the Activity Starter
Component and some examples**



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Agenda



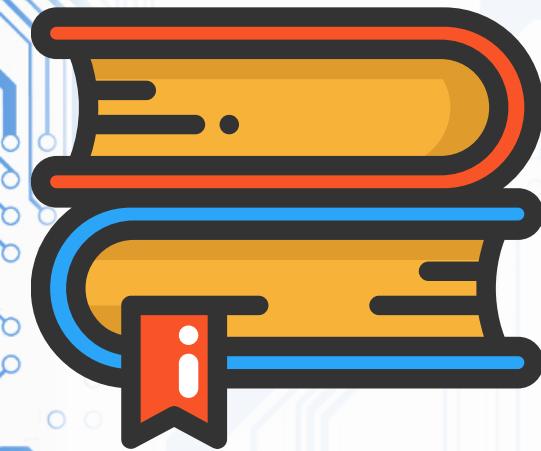
Using the Activity Starter Component

- Example01: Activity Starter Open the PDF file A&B
- Example02:Activity Starter Send an email
- Example03: Activity Starter Open the calculator/camera/map
- Example04:Send the SMS :APP inventor Social part (Self reading by student)
- Example05: Phone call. PhoneNumberPicker.



江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

Using the Activity Starter Component



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Using the Activity Starter Component



- The activity starter component lets you combine applications by having one application start up other applications.
- Activity Starter is mostly for advanced developers, but it's a key way to extend the App Inventor, because it means lets you take advantage of applications written by others, and created with other development frameworks.
- The apps you start can be App Inventor apps created by you or others; apps like Camera and Maps that are pre-installed on the device; or any app at all, created using App Inventor or not, as long as you have the necessary information to supply to the activity starter.



Using the Activity Starter Component



- For applications that are appropriately designed, you can pass values to the applications when starting it.
- You can also start an application and get back a result from it to use in further processing, provided that the application is designed to return a result.
- Presently, you can pass and receive text values only.
- To start another application, you must supply certain control information to the Android operating system.
- You do this by setting various properties of Activity Starter before calling the StartActivity method.

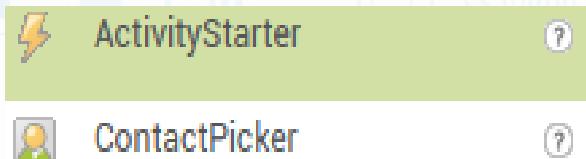


Activity Starter



- Activity Starter is a component that lets you start any Android app-- the browser, maps, etc. -- from your app.
- The component requires information to be entered with a low-level protocol.
- How to use this part :

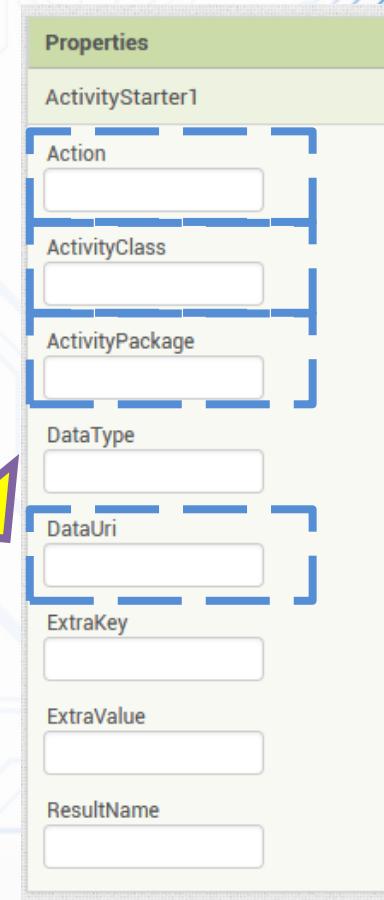
1. In the Designer, add an ActivityStarter component to your app.
2. It is found in the "other stuff" palette.
3. It comes with a button that triggers the activity.



2. Set the following properties of the component:

(these can be set in block section also)

- Action:
- ActivityPackage:
- ActivityClass:
- DataUri:





Activity Starter



What else we can do with that

- Open a deep link to an installed app
- The Activity Starter can only open another app if it's installed on a user's device
- Email
- Facebook
- Instagram
- Browser
- ,.....





Activity Starter



Starting Another App Inventor Application from your App Inventor App

- You can start another app that was created with App Inventor if you know its package name and class name.
- If you have the source code of the app, you can find these names as follows:
 - Download the source code to your computer.
 - Using a file explorer or unzip utility, find the file youngandroidproject/project.properties
 - The first line will start with " main=". Everything after that is the package and class name



Activity Starter



Starting Another App Inventor Application from your App Inventor App

For example, here is a first line you might see for an App named HelloPurr.:

main=appinventor.ai_HomerSimpson.HelloPurr

The HomerSimpson part of the name comes from the App Inventor user account. To start this app, you would use an activity starter component with these properties:

ActivityPackage: appinventor.ai_HomerSimpson.HelloPurr

ActivityClass: appinventor.ai_HomerSimpson.HelloPurr.Screen1

Invoking the activity starter's StartActivity method will start HelloPurr.

When HelloPurr finishes (if it does), the original app's ActivityStarter.AfterActivity method will be invoked.





Activity Starter

Starting Another App Inventor Application from your App Inventor App

Note: If you are going to start another App Inventor app, make sure you use the correct package name. For example, if someone posts the source code (zip file) for an app, and you repackage that app, you'll end up with a different package name than he had. In cases of doubt, or if you don't have the source code, start the activity by hand on the phone and run "adb logcat" to view the Android system log and see what was actually started, as explained below in "Figuring out how to set the properties".

Note: If you will be starting a second App Inventor app that you yourself are writing, it's generally easier and more convenient to design the whole thing as a single multiple screen app, rather than creating two separate apps. See the section below on "Activity Starter versus Multiple Screen Apps".



Activity Starter

Starting a Builtin Phone Application from your App Inventor App

- Apps that come with the phone can be invoked using package names and class names, as above. Android apps can also launch responses to an intent , which is the Android operating system's way of specifying something to do without necessarily designating a particular app.
- You can find information about intents (written for advanced developers) in the Android system documentation.
- Some apps are designed to take extra information when they are launched. For example, maps can take geographic information that specifies which map to display. Another example is launching a Web search with extra information that specifies the text to search for. You must consult the documentation for the particular app to learn what this extra information is and how to specify it. Generally, you specify the information by setting the activity starter's properties just before the other app is launched.



Start the Camera

To launch the Android Camera app, use an activity starter with these properties:

Action: android.intent.action.MAIN

ActivityPackage: com.google.android.camera

ActivityClass: com.android.camera.Camera

This is pretty much what the App Inventor Camera component does, although it is much more convenient to use the Camera component.



Launch a Web search

- To launch a Web search for a particular query, such as "Homer Simpson", use an activity starter with these properties:

Action:

android.intent.action.WEB_SEARCH

ExtraKey: query ExtraValue: Homer Simpson

• ActivityPackage:

com.google.android.providers.enhancedgooglesearch

• ActivityClass:

com.google.android.providers.enhancedgooglesearch.Launcher



Open the browser to a Web page

Use these activity starter properties to open a specific web page:

Action: android.intent.action.VIEW

DataUri: <http://news.google.com>



Start the mailer with pre-addressed message

To start the Android mailer application, use the action `android.intent.action.VIEW`.

You can use Activity starter's `DataUri` property to specify the recipient, the subject, and the body of the message. In each case, starting the activity will bring up the Android mailer, and you can complete the message and then press "Send".

For example, if you specify:

Action: `android.intent.action.VIEW` DataUri: `mailto:santa@northpole.com`

then starting the activity will bring up the mailer, with the addressee filed in.

If you specify:

`Action: android.intent.action.VIEW`

`DataUri: mailto:santa@northpole.com?subject=Please Santa&body=Bring me a pony`

then starting the activity will bring up the mailer, with the addressee, the subject, and the body filled in. You can use the mailer to edit these before sending the message, if you prefer.

The DataURI here must be a URL encoded message designed to work with the internet mailto service. Search the Web to find appropriate documentation and tools for creating these, for example Elements of a Mailto: URL



Show a map for a location

if you know the latitude and a longitude of a location, you can show a map using an activity starter with these properties to show a map of the area:

Action: android.intent.action.VIEW DataUri: geo:37.8,-122.23?z=23

The format of the DataURI is specific to the app. In this example, the URI specifies a z (zoom) of 23, which is the largest zoom value. Zoom value is optional and ranges from 1 (the entire Earth) to 23.

If you know the zip code, you can set the activity starter properties as follows:

Action: android.intent.action.VIEW DataUri: geo:0,0&q=94043

If you have a street address, you can use a DataUri that encodes the address with a scheme called URL encoding:

Action: android.intent.action.VIEW DataUri: geo:0,0&q=5000%20MacArthurBlvd%20Oakland%2CCA

Generally in URI encoding the only characters you have to replace are spaces (%20) and punctuation marks, such as comma (%2C) and period (%2E).



Play a YouTube video

- You'll need to know the URI of the YouTube video. Then set the Activity Starter properties as

Action: android.intent.action.VIEW

ActivityPackage: com.google.android.youtube

ActivityClass: com.google.android.youtube.PlayerActivity

- and set the Data URI to the URI of the video to be played, for example:
- DataUri: <http://www.youtube.com/watch?v=8ADwPLSFeY8>





Figuring out how to set the properties

- If you can't find documentation for the activities you want to start, one way to figure out how to set the properties is to start up the activity manually and look at what appears in the Android System Log. For example, if you use YouTube to play a video, you'll see in the log:
`I/ActivityManager(86): Starting activity: Intent { act=android.intent.action.VIEW
cat=[android.intent.category.BROWSABLE] dat=http://www.youtube.com/watch?v=8ADwPLSFeY8
flg=0x3800000 cmp=com.google.android.youtube/.PlayerActivity }`
- If you can find the "cmp=" string, then the ActivityPackage is the part before the slash, and the ActivityClass is the entire "cmp=" part, without the slash character.
- As in the YouTube example, there may also be "dat=" information that can specify with the DataUri property.



Starting arbitrary apps

- You can use the Activity Starter to start any activity at all if you know the package name and class name, or the appropriate intent. Some developers document these intents for the benefit of other Android developers.
- For hints on starting other apps using intents, see the Android API documentation or search the Android developer forums.
- If you have an app on your phone and you don't have the source code, you might still be able figure out the package name and class name (and sometimes the intent) by launching the app and inspecting the Android system log, as indicated above.





Receiving results from apps that you start

- Some apps are constructed to return a result.
- You can see that result as the result argument of the activity starter's AfterActivity event.
- Currently, App Inventor can receive text results only.



Returning results from App Inventor apps, and getting the result

- You can create App Inventor apps that return a (text) result to their callers, so they can be used as subroutines. To return a result, invoke the command close screen with plain text (located in the Control drawer). Your subroutine app will terminate, and the argument of close screen with plain text will become available to the app that called it.
- If the calling app was itself an App Inventor app that used an activity starter, it will see that text as the argument to the activity starter's AfterActivity event.
- **In more detail, suppose there are two App Inventor apps:**
- *A SubroutineApp designed to be started with an activity starter and return a value, and a CallerApp that uses an ActivityStarter to call SubroutineApp.*
- *To return the result, SubroutineApp executes close screen with plain text, giving it as argument the result (text only) that should be returned to the caller.*



Returning results from App Inventor apps, and getting the result

- On the caller side, CallerApp sets up its Activity Starter with the correct package name and class name for starting SubroutineApp. It must also set the ActivityStarter.ResultName property to the special text string APP_INVENTOR_RESULT. Then CallerApp starts the SubroutineApp activity. When the subroutine finishes, the Activity Starter's AfterActivity event triggers, and the result that was passed back from SubroutineApp will be available as the result argument to AfterActivity. That same information is available as the ActivityStarter's Result property.



Calling App Inventor Apps and passing values to them

- If you are writing an App Inventor CallerApp to call an App Inventor SubroutineApp, you can pass a text value to SubroutineApp. To do this, use an Activity Starter in CallerApp with the ExtraKey property to set to the special tag APP_INVENTOR_START and the ExtraValue property set to the text you want to pass. The SubroutineApp can retrieve that value by using the get start plain text block from the Control drawer.
- By passing and returning results, you can combine several App Inventor apps, both apps you write, as well as apps you share with others.



Returning results from arbitrary applications

- Getting values back from other applications is like getting values back from App Inventor apps. In general in Android, an activity will return a result that is designated by a name, where the result is designated by a name, where the name is specified with the Activity Starter's ResultName property.
- The name to use for App Inventor apps is APP_INVENTOR_RESULT, as described above. Other applications, not created with App Inventor, will use other names, and you'll need to know those names in order get values back from those applications.
- In general, you'll need to consult documentation about the app you want to use as a subroutine. Some developers provide this information, or sometimes you'll have access to the source code.



Returning results from arbitrary applications

- Not all Android apps use the Result and ResultName mechanism. For example, some apps return information via the properties ResultType and ResultUri. Again, you'll need to have information from the app developer to know which of these to use.
- For advanced developers: More specifically (with reference to the Android developer documentation) an app can be designed to return an intent. The Activity starter uses the specified ResultName to access `intent.getStringExtra(resultName)` to produce the result. The values for ResultType and ResultUri come from `intent.getType()` and `intent.getUri()`.



Example: Picking files from the SD Card

- Here's an example that illustrates using a third-party application:
- AndExplorer from Lysesoft is an application that lets you pick files for your SD card.
- You can call AndExplorer with an activity starter to add a file picking capability to your application.
- You'll need to have AndExplorer installed on your phone.
- You can get it from the Android Market.
- To start AndExplorer to pick a file, use an ActivityStarter with:

Action: android.intent.action.PICK

dataType: vnd.android.cursor.dir/lysesoft.andexplorer.file

dataURI: file:///sdcard

When you start the activity and pick a file, the resulting file name will be available as ResultUri.

Also, ResultType will give the type of the file, for example, image/jpeg or audio/mpeg.

Using this, you can write an app that lets you pick a file from the SD card and either displays the image or plays the music track as appropriate.

<https://appinventor.mit.edu/explore/ai2/activity-starter.html>



Open Facebook

- Open Facebook deep link in app if app is installed, open in website if not

```
initialize global fbWebUrl to " https://www.facebook.com/newmarketgoods "

when Facebook .Click
do
  set FacebookStarter . Action to " android.intent.action.VIEW "
  set FacebookStarter . Data Uri to join " fb://facewebmodal/f?href=" get global fbWebUrl
  call FacebookStarter .Start Activity

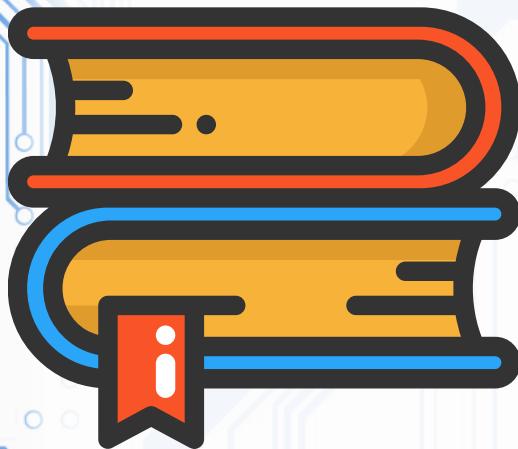
when FacebookStarter .ActivityError
message
do
  set FacebookStarter . Data Uri to get global fbWebUrl
  call FacebookStarter .Start Activity
```





江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

Example01:
Activity Starter Open the PDF file



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor

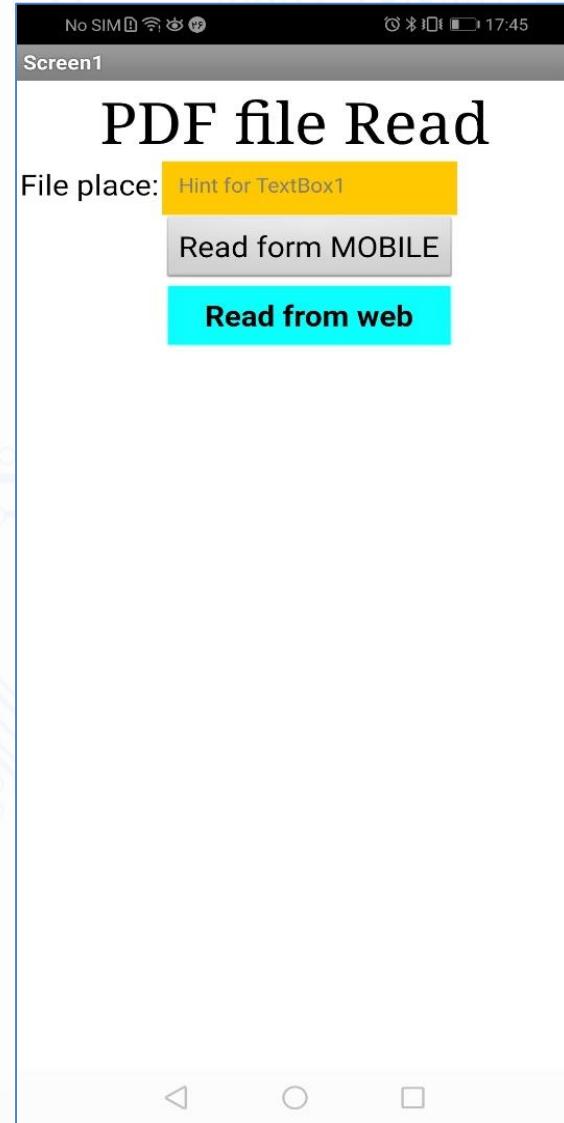


Activity Starter Open the PDF file



Example Aim

open the pdf file with two method
of activity starter and web viewer



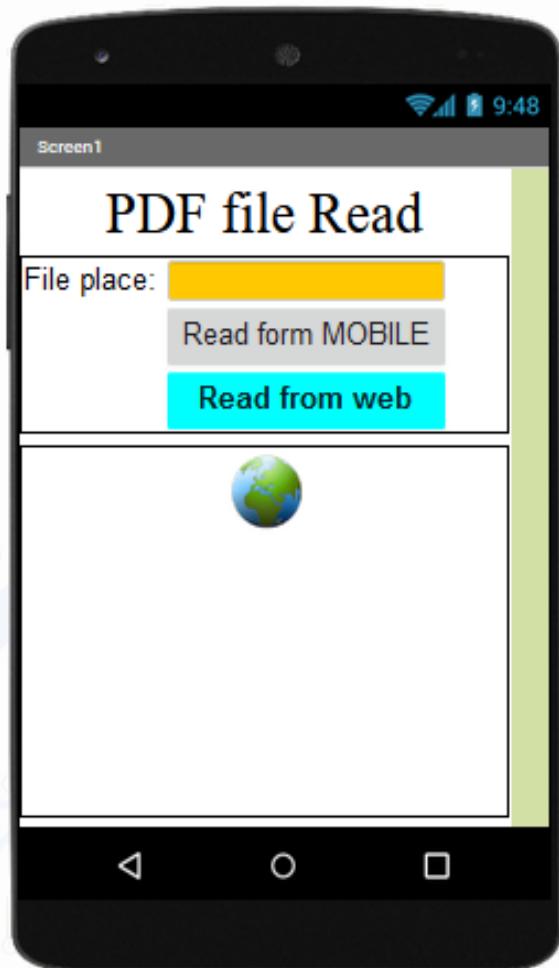
App Inventor



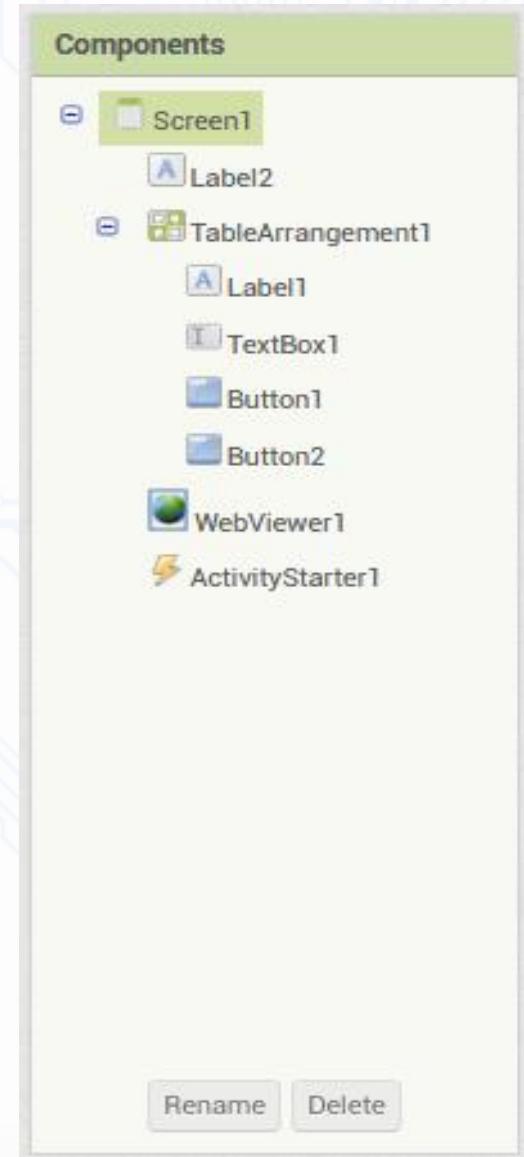
江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



Activity Starter Open the PDF file



江西理工大学信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Activity Starter Open the PDF file

application/pdf
android.intent.action.VIEW
file:///mnt/sdcard/ata/f1.pdf

webViewer
<https://docs.google.com/viewer?url=https://arxiv.org/ftp/arxiv/papers/2012/2012.05391.pdf>

```
when with_activestarter .Click
do set ActivityStarter1 . DataType to "application/pdf"
   set ActivityStarter1 . Action to "android.intent.action.VIEW"
   set ActivityStarter1 . DataUri to "file:///mnt/sdcard/ata/f1.pdf"
   call ActivityStarter1 . StartActivity
```

```
when with_web .Click
do call WebViewer1 . GoToUrl
   url join " https://docs.google.com/viewer?url=" " https://arxiv.org/ftp/arxiv/papers/2012/2012.053... "
```

Application/pdf

Android.intent.action.VIEW

file:///mnt/sdcard/ata/f1.pdf

<https://docs.google.com/viewer?url=https://arxiv.org/ftp/arxiv/papers/2012/2012.05391.pdf>

<https://arxiv.org/ftp/arxiv/papers/2012/2012.05391.pdf>





Activity Starter Open the PDF file

The screenshot shows the MIT App Inventor 2 environment. The top bar includes the menu (File, Edit, View, History, Bookmarks, Tools, Help), a browser header with the URL [www.BANDICAM.com](http://ai2.appinventor.mit.edu/#4708009713729536), and a toolbar with various icons. Below the header, the browser shows a search bar with the query "ai2.appinventor.mit.edu/#4708009713729536", a refresh button, and a download button. The main workspace is titled "PDF_read" and displays a smartphone screen with the title "Screen1". The "Components" panel on the right lists "Screen1" with options to "Rename" or "Delete". The "Properties" panel on the far right contains numerous settings for "Screen1", including "AccentColor", "AlignHorizontal", "AlignVertical", "AppName", "BackgroundColor", "BackgroundImage", "BlocksToolkit", "CloseScreenAnimation", "Icon", "OpenScreenAnimation", "PrimaryColor", "PrimaryColorDark", and "ScreenOrientation". The bottom of the screen shows the Windows taskbar with various application icons and the system tray.





Activity Starter Open the PDF file

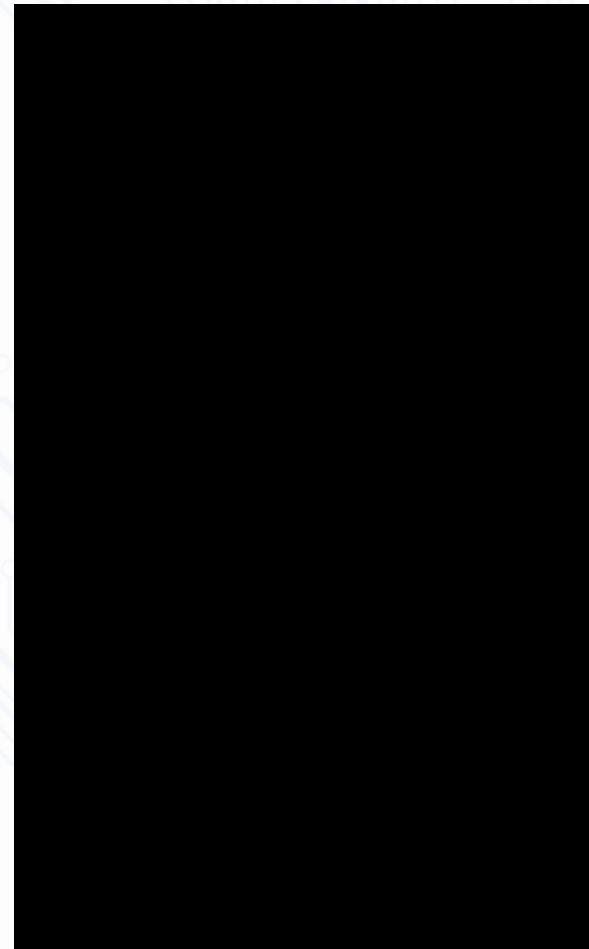
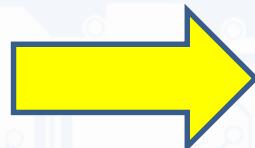
The screenshot shows the MIT App Inventor 2 web-based development environment. The top navigation bar includes File, Edit, View, History, Bookmarks, Tools, Help, and a URL bar pointing to [www.BANDICAM.com](http://ai2.appinventor.mit.edu/#4708009713729536). Below the URL bar are standard browser controls like back, forward, search, and refresh. The main workspace is titled 'PDF_read' and contains the following components:

- Blocks Editor:** On the left, a tree view shows categories like Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures, and components like Screen1, Label2, TableArrangement1, and ActivityStarter1. A central area displays two blocks:
 - A yellow 'when Button1.Click' event block with a nested 'do' loop containing:
 - Set ActivityStarter1.DataType to application/pdf
 - Set ActivityStarter1.Action to android.intent.action.VIEW
 - Set ActivityStarter1.DataUri to file:///mnt/sdcard/sta/f1.pdf
 - Call ActivityStarter1.StartActivity
 - A second 'when Button1.Click' event block with an empty 'do' loop.
- Viewer:** On the right, it shows a preview of the app's user interface with a green 'PDF Read' button and a blue 'PDF Write' button.
- Media:** A section at the bottom left for uploading files, currently showing 0 images and 0 videos.
- Warnings:** At the bottom center, there are three warning icons (triangle, exclamation, and circle) and a 'Show Warnings' button.
- System Bar:** The bottom of the screen features a taskbar with various icons (Windows, Search, Start, etc.) and a system tray with battery, signal, and volume indicators. The date and time are shown as 16/12/2020 17:17.



Activity Starter Open the PDF file

Demo APP



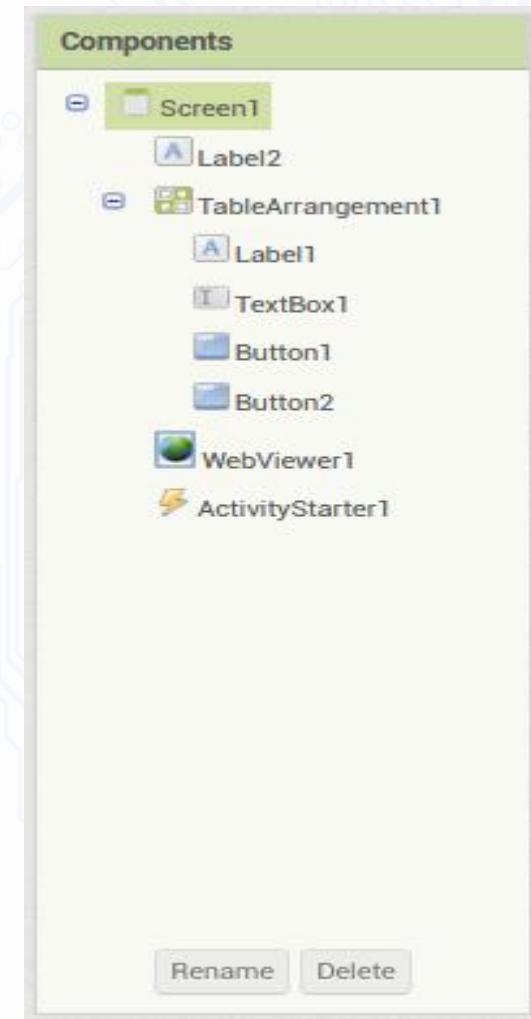
江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Activity Starter Open the PDF file



江西理工大学信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Activity Starter Open the PDF file

```
when Button1 .Click
do
  set ActivityStarter1 . DataType to " application/pdf "
  set ActivityStarter1 . Action to " android.intent.action.VIEW "
  set ActivityStarter1 . DataUri to join (" file:///mnt/sdcard/ata/ "
                                         TextBox1 . Text
                                         ".pdf ")
call ActivityStarter1 .StartActivity
```

application/pdf
android.intent.action.VIEW
file:///mnt/sdcard/ata.pdf

```
when Button2 .Click
do
  call WebViewer1 .GoToUrl
    url join (" https://docs.google.com/viewer?url=" )
              (" https://arxiv.org/ftp/arxiv/papers/2012/2012.053... " )
```





Extension: Select the file name

File Edit View History Bookmarks Tools Help

www.BANDICAM.COM

MIT App Inventor x +

MIT App Inventor

Getting Started PID Basic functions related... JRM | Fuji Technology ... YouTube New Tab

PDF_read_02 Screen1 Add Screen... Remove Screen Publish to Gallery Designer Blocks

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Dictionaries
 - Colors
 - Variables
 - Procedures
- Screen1
 - Label2
 - TableArrangement1
 - Label1
 - TextBox1
 - Button1
 - Button2
 - WebView1

Viewer

when Button1 .Click
do set ActivityStarter1 . . DataType to "application/pdf"
set ActivityStarter1 . . Action to "android.intent.action.VIEW"
set ActivityStarter1 . . DataUri to "file:///mnt/sdcard/ats/f1.pdf"
call ActivityStarter1 . StartActivity

when Button2 .Click
do call WebViewer1 . GoToUrl
url join https://docs.google.com/viewer?url= https://arxiv.org/ftp/arxiv/papers/12/12.053...

Media

Upload File ...

Show Warnings

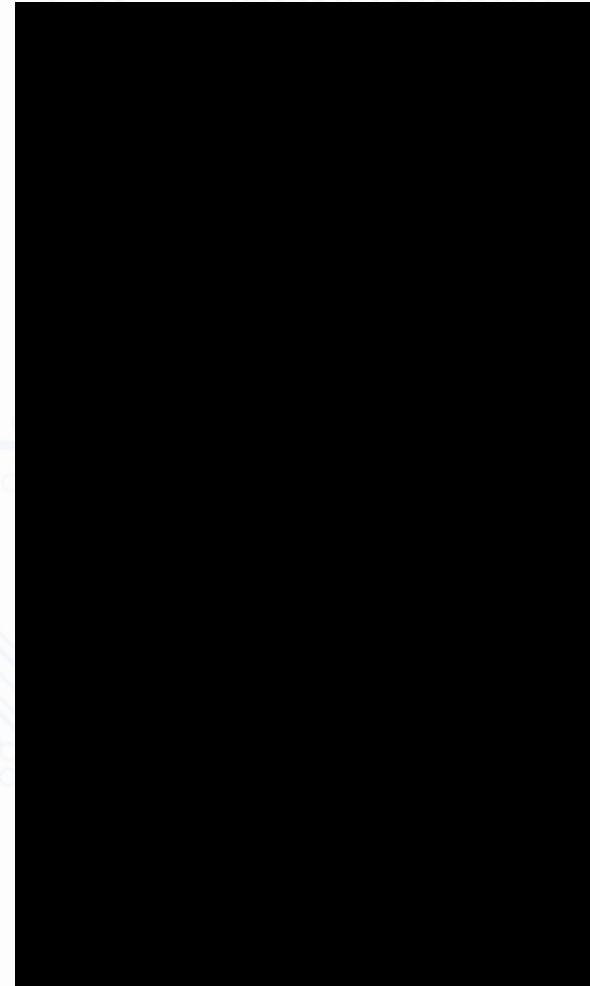
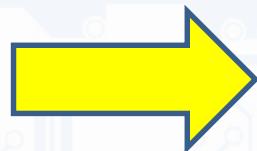
17:34 ENG 16/12/2020





Extension: Select the file name

Demo APP



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Extension: Select the file name



File Place: Hint for TextBox1

From Phone

Web

OTHERS
DEMO





江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

Example02:

Activity Starter Send an email.



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor

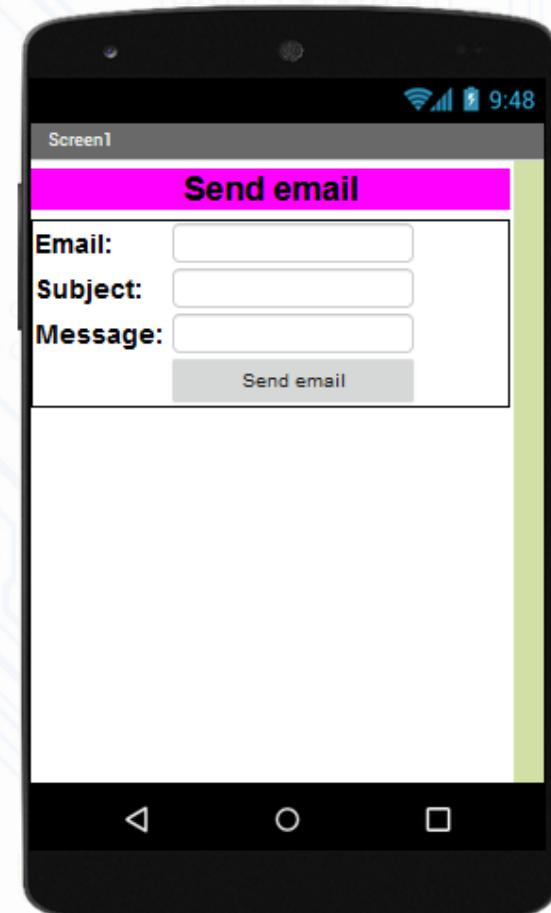


Activity Starter Send an email.



Example Aim

- We will send an email to an email address.
- Write an e-mail, a subject and a message, pressing the button **Send**, we will mail the gmail page, on that page press **Send**.
- You must have a **gmail account**, and that mail is sent to the sender that direction.



Non-visible components

ActivityStarter1



App Inventor



Activity Starter Send an email.



App Inventor



Non-visible components



江西理工大学信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

Components

- Screen1
- Label4
- TableArrangement1
 - Label1
 - Label2
 - Label3
 - TextBox1
 - TextBox2
 - TextBox3
 - Button1
- ActivityStarter1

Rename Delete



App Inventor



Activity Starter Send an email.

```
when Button1 .Click
do  set ActivityStarter1 . Action to " android.intent.action.VIEW "
    set ActivityStarter1 . DataUri to join " mailto: "
        TextBox1 . Text
        "?subject= "
        TextBox2 . Text
        "&body= "
        TextBox3 . Text
        "\n\n This is my message."
    call ActivityStarter1 . StartActivity
```

android.intent.action.View
mailto:
?subject=
&body=
\n\n This is my message.





Activity Starter Send an email.

The screenshot shows the MIT App Inventor Designer interface. The top bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The title bar says "www.BANDICAM.com". The browser tab is "MIT App Inventor". The main workspace is titled "send_email" and contains "Screen1". The Components panel shows "Screen1" selected. The Properties panel displays the following settings for "Screen1":

- AboutScreen: [empty]
- AccentColor: Default
- AlignHorizontal: Left : 1
- AlignVertical: Top : 1
- AppName: send_email
- BackgroundColor: Default
- BackgroundImage: None...
- BlocksToolkit: All
- CloseScreenAnimation: Default
- Icon: None...
- OpenScreenAnimation: Default
- PrimaryColor: Primary

The bottom taskbar shows various application icons, and the system tray indicates the date and time as 16/12/2020 16:28.



江西理工大学信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

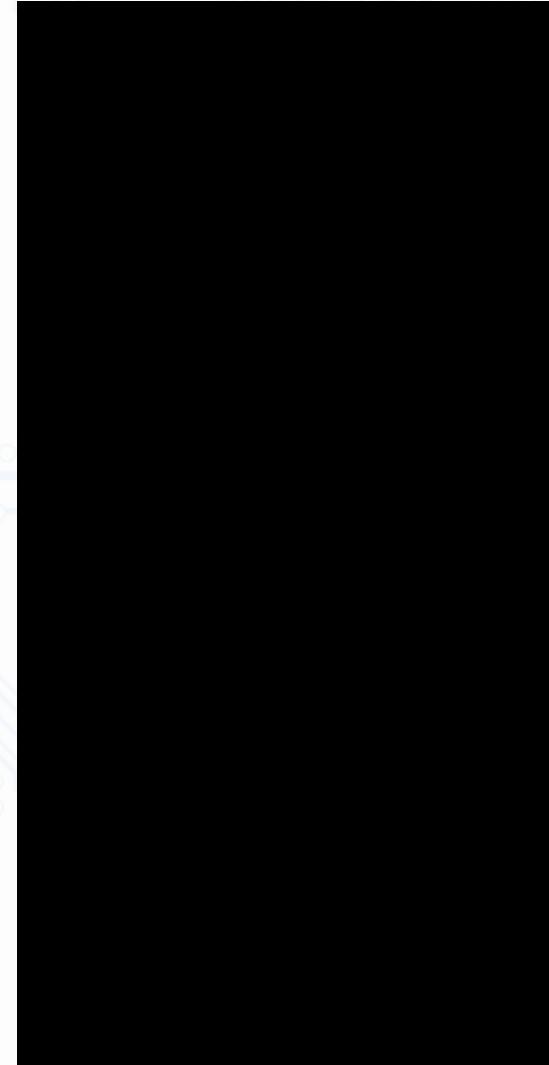
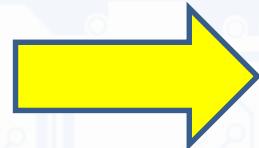


App Inventor



Activity Starter Send an email.

Demo APP



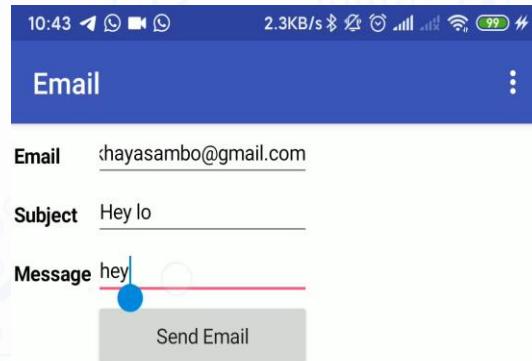
江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Example 2:Email





江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

Example03:

Activity Starter Open the
calculator/camera/map



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Example 03: Using the Activity Starter Component

- Example Aim.

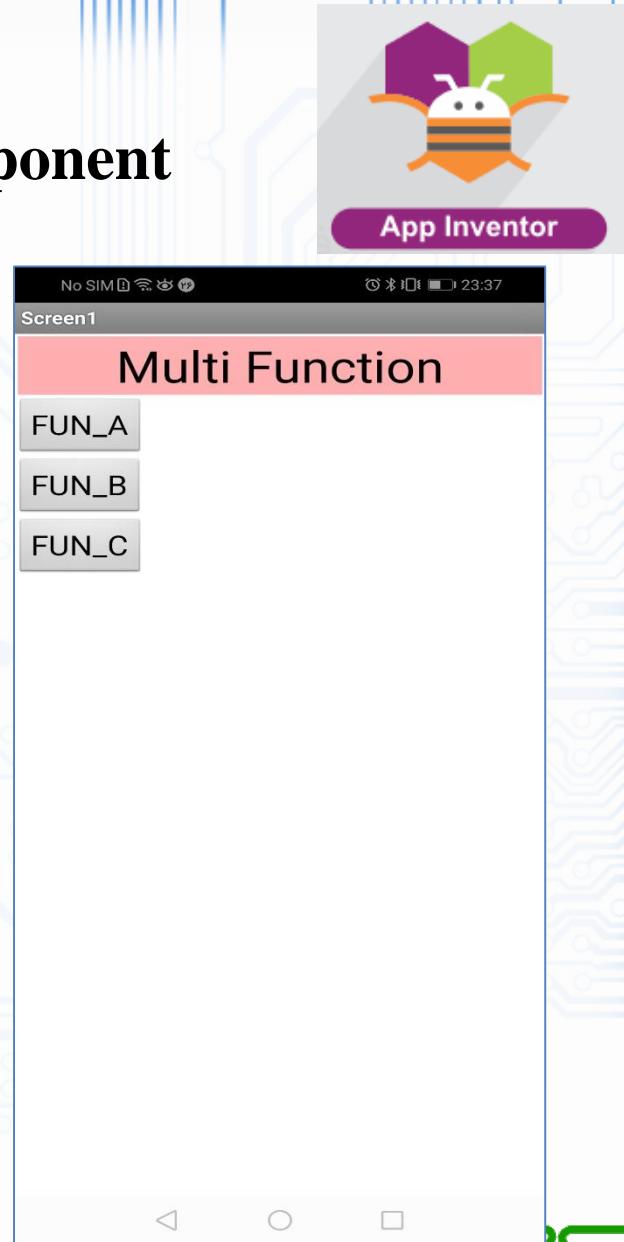
- ActivityStarter is used to start another activity, that is, to run another program, either those already on the phone or those who have done us.
- To start such programs is necessary to know some of your data, for example:

Action: **android.intent.action.MAIN**

ActivityPackage:

com.google.android.camera

ActivityClass: **com.android.camera.Camera**





Example 03: Using the Activity Starter Component

The image shows a screenshot of an Android application and its corresponding component palette.

Screenshot: The application's main screen is titled "Screen1". It features a red header bar with the text "Multi Function". Below the header are three grey rectangular buttons labeled "FUN_A", "FUN_B", and "FUN_C". The bottom of the screen has a black navigation bar with three white icons: a triangle pointing left, a circle, and a square.

Component Palette: A window titled "Components" is displayed on the right side. It lists the following components:

- Screen1
- Label1
- FUN_A
- FUN_B
- FUN_C
- ActivityStarter1

At the bottom of the palette are two buttons: "Rename" and "Delete".





Example 03:

Using the Activity Starter Component



- Button 1

android.intent.action.MAIN

com.android.calculator2.Calculator

com.android.calculator2

- Button 2

android.media.action.STILL_IMAGE_CAMERA

- Button3

android.intent.action.VIEW

google.streetview:cbll=36.52856,-6.1922312&cbp

- Google maps can also be put:

Action: android.intent.action.VIEW

Class: com.google.android.maps.MapActivity

Package: com.google.android.apps.maps

DataUri: geo: geo:latitud,longitud?z=20





Example 03:

Using the Activity Starter Component



App Inventor

```
when FUN_A .Click
do set ActivityStarter1 . Action to " android.intent.action.MAIN "
set ActivityStarter1 . ActivityClass to " com.android.calculator2.Calculator "
set ActivityStarter1 . ActivityPackage to " com.android.calculator2 "
call ActivityStarter1 . StartActivity

when FUN_B .Click
do set ActivityStarter1 . Action to " android.intent.action.MAIN "
set ActivityStarter1 . ActivityClass to " com.android.camera.Camera "
set ActivityStarter1 . ActivityPackage to " com.google.android.camera "
call ActivityStarter1 . StartActivity

when FUN_C .Click
do set ActivityStarter1 . Action to " android.intent.action.VIEW "
set ActivityStarter1 . ActivityClass to " com.google.android.maps.MapsActivity "
set ActivityStarter1 . ActivityPackage to " com.google.android.apps.maps "
set ActivityStarter1 . DataUri to " geo: geo:latitud,longitud?z=20 "
call ActivityStarter1 . StartActivity
```



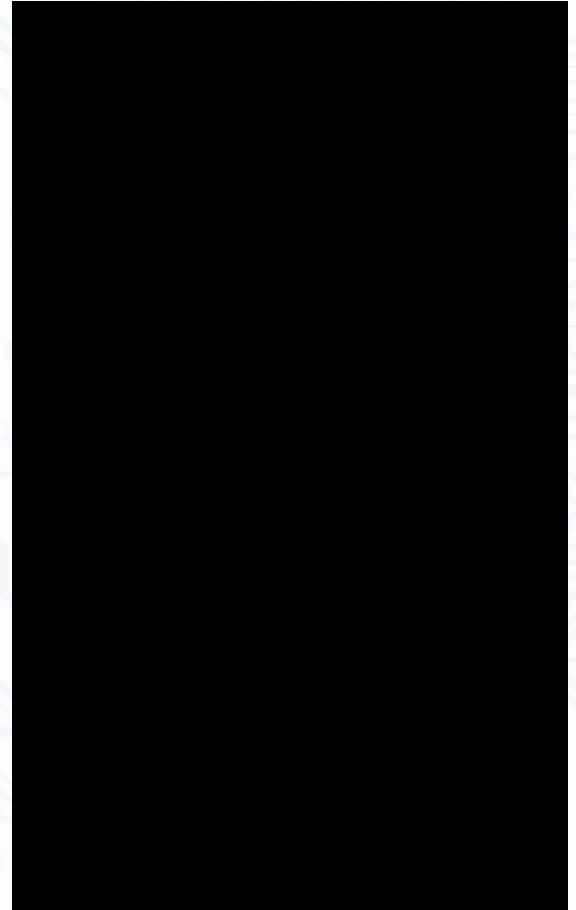
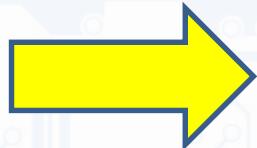


Example 03:

Using the Activity Starter Component



Demo APP



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

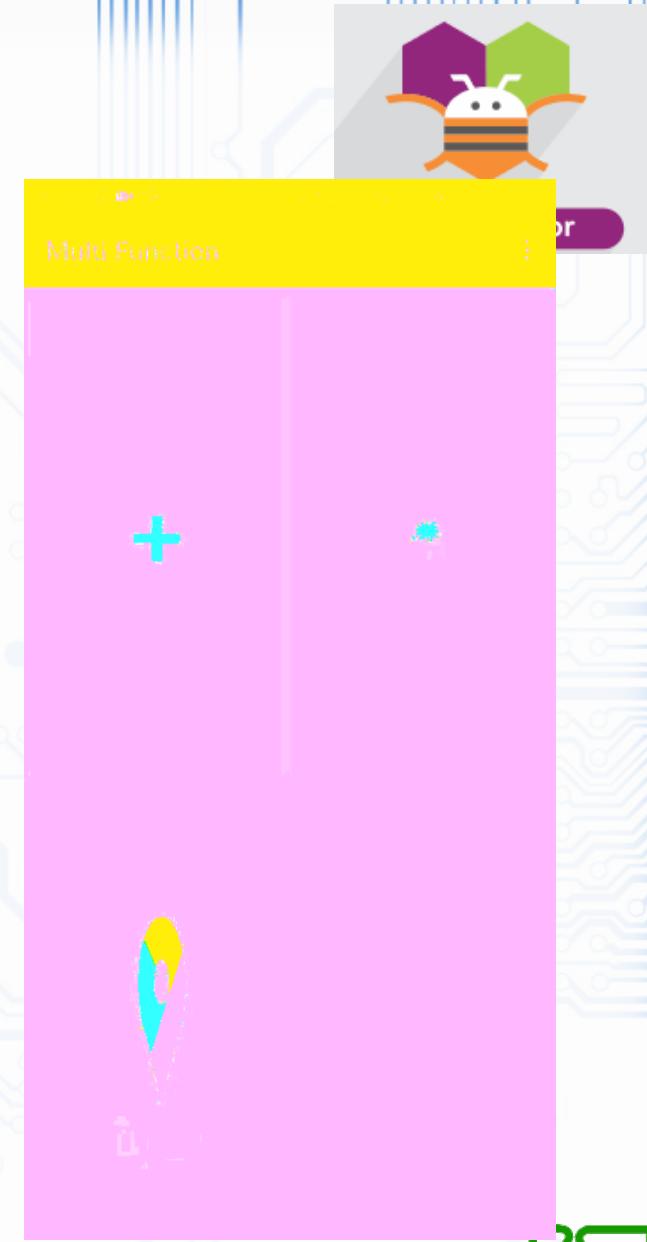


App Inventor



Example3

OTHERS
DEMO



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Example 03:

Using the Activity Starter Component

Google maps can also be put:

- Action:** android.intent.action.VIEW
- Class:** com.google.android.maps.MapActivity
- Package:** com.google.android.apps.maps
- DataUri:** geo: geo:latitud,longitud?z=20

Find the
problem
???????????





江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

APP inventor Social
(Self reading by student)



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Social



- Contact Picker
- Email Picker
- Phone Call
- PhoneNumberPicker
- Sharing
- Texting
- Twitter





ContactPicker

- A button that, when clicked on, displays a list of the contacts to choose among.
- After the user has made a selection, the following properties will be set to information about the chosen contact:
 - **ContactName:** the contact's name
 - **EmailAddress:** the contact's primary email address
 - **EmailAddressList:** a list of the contact's email addresses
 - **ContactUri:** the contact's URI on the device
 - **PhoneNumber:** the contact's primary phone number (on Later Android Versions)
 - **PhoneNumberList:** a list of the contact's phone numbers (on Later Android Versions)
 - **Picture:** the name of the file containing the contact's image, which can be used as a Picture property value for the Image or ImageSprite component.
- Other properties affect the appearance of the button (TextAlignment, BackgroundColor, etc.) and whether it can be clicked on (Enabled).
- The ContactPicker component might not work on all phones. For example, on Android systems before system 3.0, it cannot pick phone numbers, and the list of email addresses will contain only one email.



EmailPicker

- An EmailPicker is a kind of text box. If the user begins entering the name or email address of a contact, the phone will show a dropdown menu of choices that complete the entry.
- If there are many contacts, the dropdown can take several seconds to appear, and can show intermediate results while the matches are being computed.
 - The initial contents of the text box and the contents< after user entry is in the Text property. If the Text property is initially empty, the contents of the Hint property will be faintly shown in the text box as a hint to the user.
 - Other properties affect the appearance of the text box (TextAlignment, BackgroundColor, etc.) and whether it can be used (Enabled).
 - Text boxes like this are usually used with Button components, with the user clicking on the button when text entry is complete.



PhoneCall

- A non-visible component that makes a phone call to the number specified in the PhoneNumber property, which can be set either in the Designer or Blocks Editor. The component has a MakePhoneCall method, enabling the program to launch a phone call.
- You may also use MakePhoneCallDirect to directly initiate a phone call without user interaction. However, apps using this block may require further review by Google if submitted to the Play Store so it is advised to use MakePhoneCall instead.
- Often, this component is used with the ContactPicker component, which lets the user select a contact from the ones stored on the phone and sets the PhoneNumber property to ContactPicker's PhoneNumber property.
- To directly specify the phone number (e.g., 650-555-1212), set the PhoneNumber property to a Text with the specified digits (e.g., “6505551212”).
- Dashes, dots, and parentheses may be included (e.g., “(650)-555-1212”) but will be ignored; spaces may not be included.



PhoneNumberPicker

- A button that, when clicked on, displays a list of the contacts' phone numbers to choose among. After the user has made a selection, the following properties will be set to information about the chosen contact:
 - ContactName: the contact's name
 - PhoneNumber: the contact's phone number
 - EmailAddress: the contact's email address
 - Picture: the name of the file containing the contact's image, which can be used as a Picture property value for the Image or ImageSprite component.
- Other properties affect the appearance of the button (TextAlignment, BackgroundColor, etc.) and whether it can be clicked on (Enabled).
- The PhoneNumberPicker component may not work on all Android devices. For example, on Android systems before system 3.0, the returned lists of phone numbers and email addresses will be empty.



PhoneNumberPicker

- A button that, when clicked on, displays a list of the contacts' phone numbers to choose among. After the user has made a selection, the following properties will be set to information about the chosen contact:
 - ContactName: the contact's name
 - PhoneNumber: the contact's phone number
 - EmailAddress: the contact's email address
 - Picture: the name of the file containing the contact's image, which can be used as a Picture property value for the Image or ImageSprite component.
- Other properties affect the appearance of the button (TextAlignment, BackgroundColor, etc.) and whether it can be clicked on (Enabled).
- The PhoneNumberPicker component may not work on all Android devices. For example, on Android systems before system 3.0, the returned lists of phone numbers and email addresses will be empty.



Sharing

- Sharing is a non-visible component that enables sharing files and/or messages between your app and other apps installed on a device. The component will display a list of the installed apps that can handle the information provided, and will allow the user to choose one to share the content with, for instance a mail app, a social network app, a texting app, and so on.
- The file path can be taken directly from other components such as the Camera or the ImagePicker, but can also be specified directly to read from storage.
- Be aware that different devices treat storage differently, so a few things to try if, for instance, you have a file called arrow.gif in the folder Appinventor/assets, would be:
- "file:///sdcard/Appinventor/assets/arrow.gif"; or
- "/storage/Appinventor/assets/arrow.gif"



Twitter

- A non-visible component that enables communication with Twitter. Once a user has logged into their Twitter account (and the authorization has been confirmed successful by the IsAuthorized event), many more operations are available:
- Searching Twitter for tweets or labels (SearchTwitter)

Sending a Tweet (Tweet)

- Sending a Tweet with an Image (TweetWithImage)
- Directing a message to a specific user (DirectMessage)
- Receiving the most recent messages directed to the logged-in user (RequestDirectMessages)
- Following a specific user (Follow)
- Ceasing to follow a specific user (StopFollowing)
- Getting a list of users following the logged-in user (RequestFollowers)
- Getting the most recent messages of users followed by the logged-in user (RequestFriendTimeline)
- Getting the most recent mentions of the logged-in user (RequestMentions)
- You must obtain a Consumer Key and Consumer Secret for Twitter authorization specific to your app from http://twitter.com/oauth_clients/new



江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

Example04: Send the SMS



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



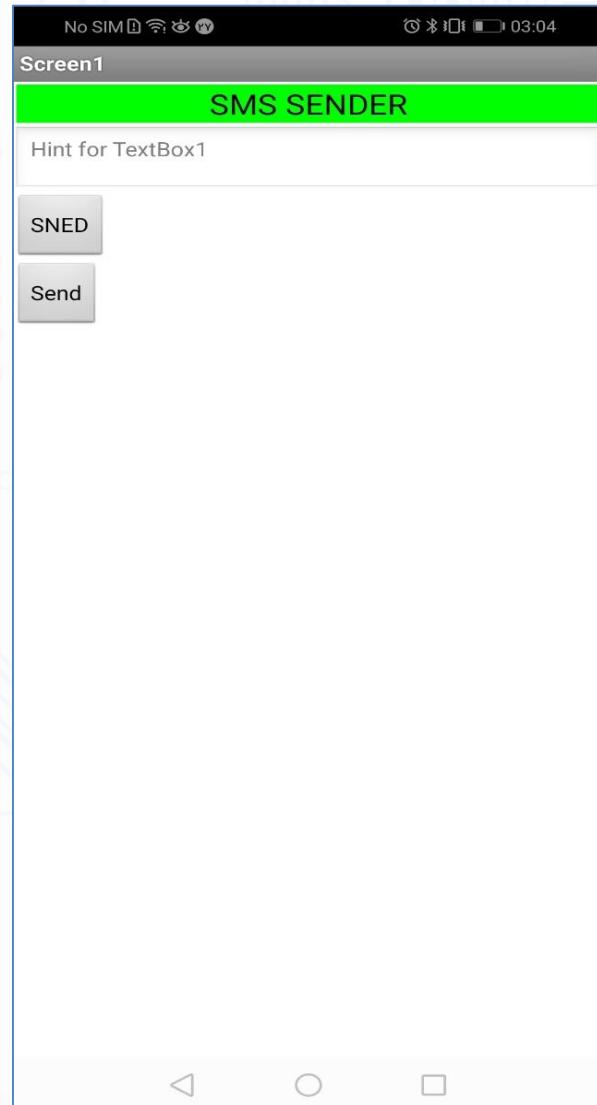
App Inventor



Example04: Send the SMS

Example Aim.

- Send SMS
- Pick number and send the sms



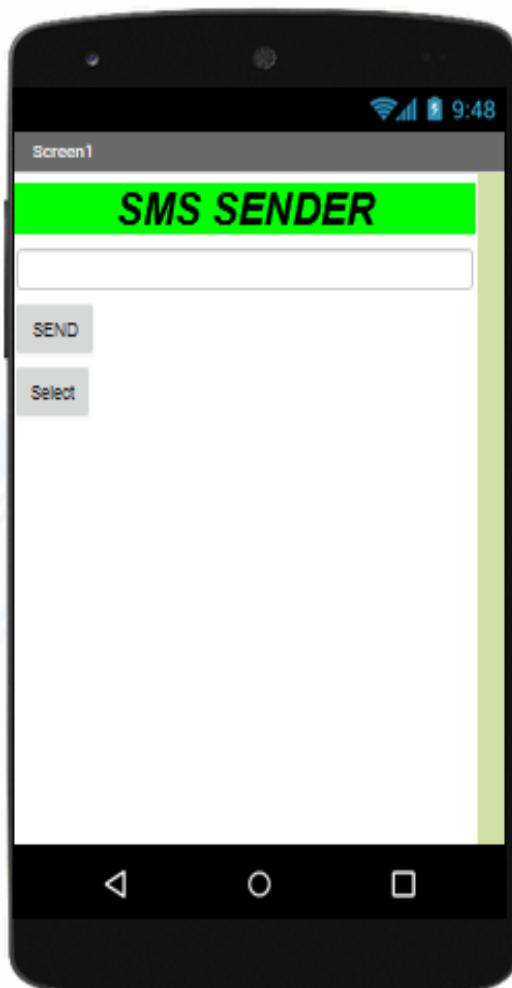
App Inventor



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



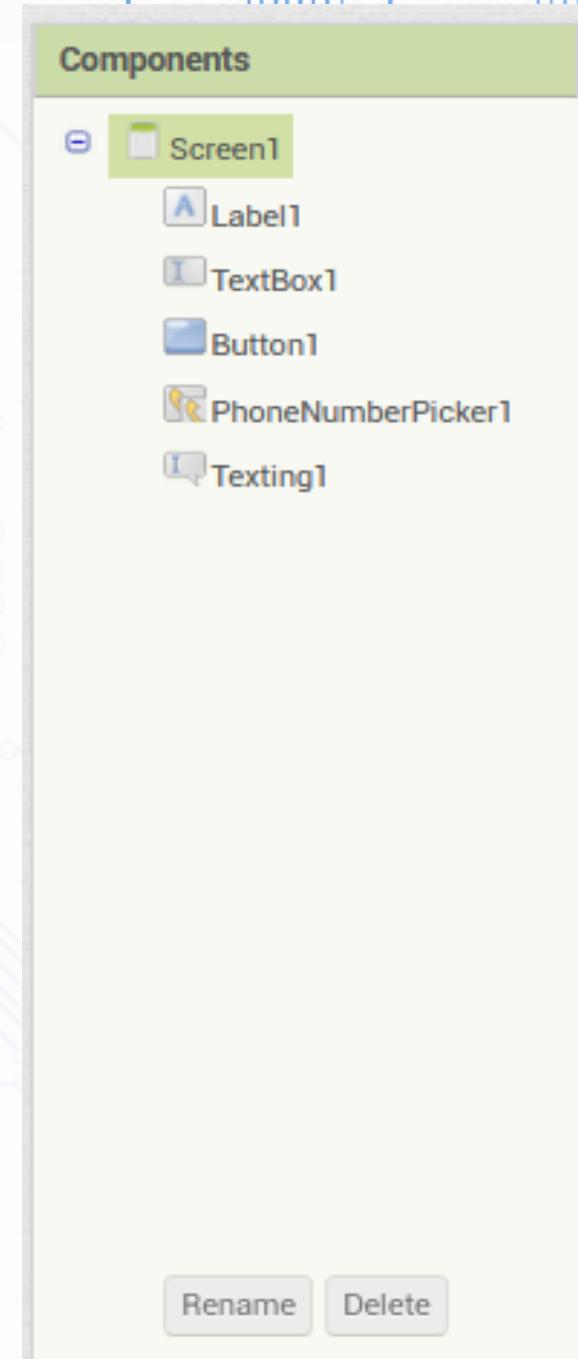
Send SMS



Non-visible components



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Send SMS

```
when PhoneNumberPicker1 .AfterPicking
do set Texting1 . PhoneNumber to PhoneNumberPicker1 . PhoneNumberList
```

```
when Button1 .Click
do set Texting1 . Message to TextBox1 . Text
call Texting1 . SendMessage
```





Send SMS

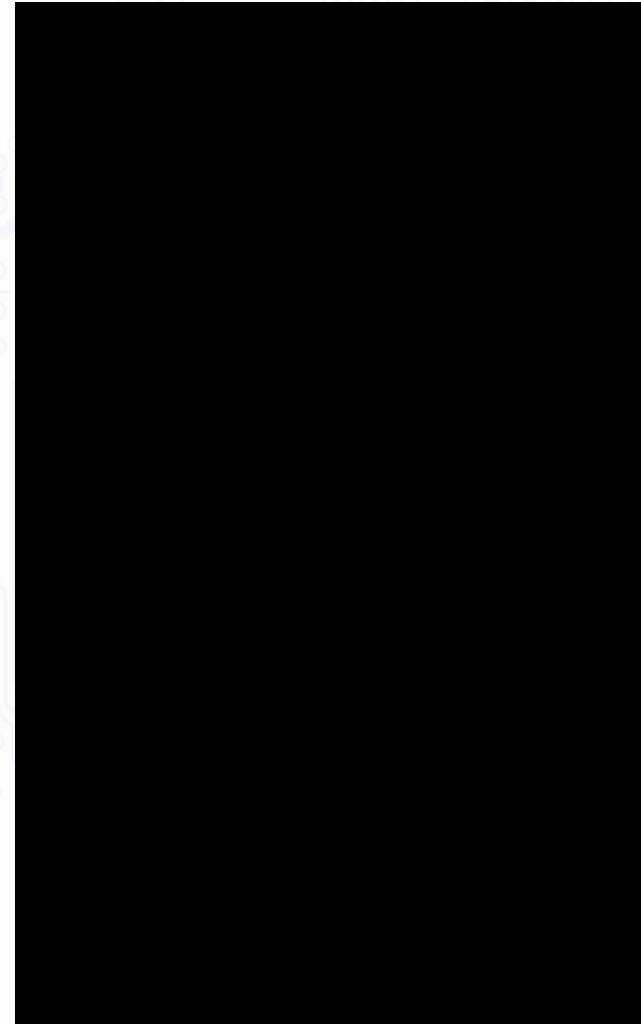
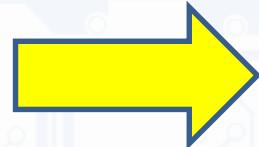
The screenshot shows the MIT App Inventor 2 Designer window. At the top, the menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The title bar displays the URL www.BANDICAM.com. Below the menu is a toolbar with various icons for navigation and file operations. The main workspace is titled "Sendsms_rev02". It features a "Palette" on the left containing categories like User Interface (with items like Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, and WebViewer), Layout, Media, Drawing and Animation, Maps, and Sensors. A "Components" panel on the right lists "Screen1" with properties such as AccentColor (Default), AlignHorizontal (Left: 1), AlignVertical (Top: 1), AppName (Sendsms_rev02), BackgroundColor (Default), BackgroundImage (None...), BlocksToolkit (All), CloseScreenAnimation (Default), Icon (None...), OpenScreenAnimation (Default), PrimaryColor (Default), PrimaryColorDark (Default), and ScreenOrientation (Unspecified). The central area is the "Viewer" which shows a smartphone icon representing the app's user interface. The phone screen is labeled "Screen1" and has a status bar showing signal strength, battery level, and time (9:48). Above the phone icon are two dropdown menus: "Display hidden components in Viewer" and "Phone size (505,320)".





Send SMS

Demo APP



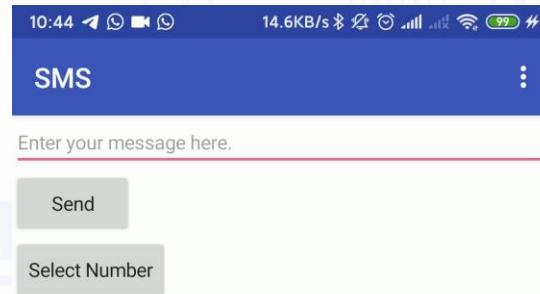
江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



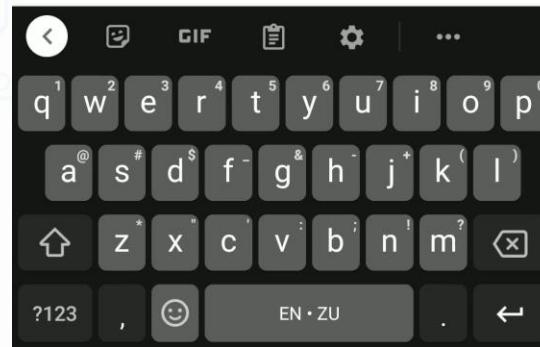
App Inventor



Example 4



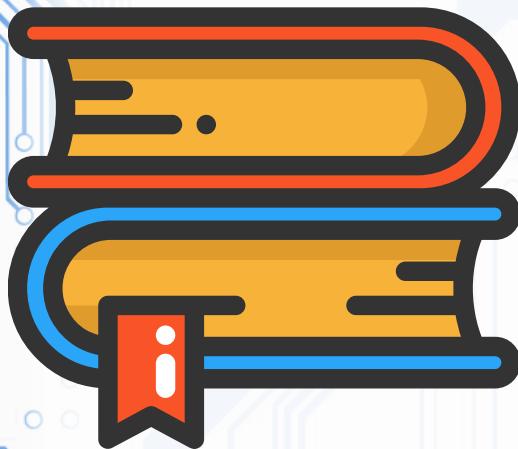
OTHERS
DEMO





江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

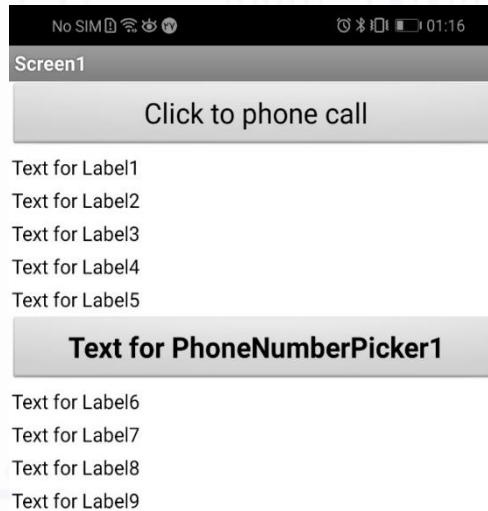
Example05:
**Phone call.
PhoneNumberPicker.**



Phone call. PhoneNumberPicker.

- Presentation.

- We will now make receive phone calls. It would be desirable to have two phones to see the operation of this application.
- We put a real phone number which we call, in this example I have set: 4444444444.
- Incoming call (incoming call), is the call that we receive.
- Outgoing call (outcoming call), it is the call we make.
- We'll call from our application to another phone and other mobile phone. The labels will get a message with the status of the call.





Phone call. PhoneNumberPicker.

Social

- ContactPicker
- EmailPicker
- PhoneCall
- PhoneNumberPicker
- Sharing
- Texting
- Twitter

Display hidden components in Viewer
Check to see Preview on Tablet size.

Phone call - (@ Juan A. Villalpando) 9:48

Click to phone call.

Text to Label1
Text to Label2
Text to Label3
Text to Label4
Text to Label5

Text for PhoneNumberPicker1

Text to Label6
Text to Label7
Text to Label8
Text to Label9

Non-visible components

PhoneCall1

Components

- Screen1
 - Button1
 - Label1
 - Label2
 - Label3
 - Label4
 - Label5
 - PhoneNumberPicker1**
 - Label6
 - Label7
 - Label8
 - Label9
 - PhoneCall1

KIO4.COM

Rename Delete

Media

Upload File ...

Inventor



Phone call.

PhoneNumberPicker.

```
initialize global [message] to " " 

when [Button1 Click]
do   set [PhoneCall1.PhoneNumber] to "4444444444"
    call [PhoneCall1 MakePhoneCall]

when [PhoneCall1.PhoneCallStarted]
status [phoneNumber]
do   if [get status] = 1
then   set global message to "Incoming call"
else if [get status] = 2
then   set global message to "Outcoming call"
set [Label1.Text] to [join ["Status", [get status], [get global message]]]
set [Label2.Text] to [join ["Number", [get phoneNumber]]]
```





Phone call.

PhoneNumberPicker.

```
when PhoneCall1 .PhoneCallEnded
  status phoneNumber
  do
    if get status = 1
      then set global message to "Incoming call is missed or rejected"
    else if get status = 2
      then set global message to "inoming call is answered before hanging up"
    else if get status = 3
      then set global message to "Outcoming call is hanging up"
    set Label3 .Text to join ("Status", get status, get global message)
    set Label4 .Text to join ("Number", get phoneNumber)

when PhoneCall1 .IncomingCallAnswered
  phoneNumber
  do
    set Label5 .Text to join ("Phone Number incoming", get phoneNumber)
```





Phone call. PhoneNumberPicker.

```
initialize global phones to [ create empty list ]  
  
when [PhoneNumberPicker1] .AfterPicking  
do  
    set [Label6] .Text to [PhoneNumberPicker1] .ContactName  
    set [Label7] .Text to [PhoneNumberPicker1] .PhoneNumber  
    set [Label8] .Text to [PhoneNumberPicker1] .EmailAddress  
    set global phones to [ create empty list ]  
    [ add items to list ] list [ get global phones ]  
        item [PhoneNumberPicker1] .PhoneNumberList  
    set [Label9] .Text to [ get global phones ]  
    set [PhoneCall1] .PhoneNumber to [PhoneNumberPicker1] .PhoneNumber  
    call [PhoneCall1] .MakePhoneCall
```





Phone call. PhoneNumberPicker.

File Edit View History Bookmarks Tools Help

www.BANDICAM.com

MIT App Inventor

Getting Started PID Basic functions related... C اپلیکیشن آموزش زبان JRM | Fuji Technology ... (36) YouTube New Tab

Palette

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Layout

Media

Drawing and Animation

Maps

Viewer

Display hidden components in Viewer
Phone size (505,320)

Screen1

Components

Properties

Screen1

AboutScreen

AccentColor Default

AlignHorizontal Left : 1

AlignVertical Top : 1

AppName Phone_call_picker

BackgroundColor Default

BackgroundImage None...

BlocksToolkit All

CloseScreenAnimation Default

Icon None...

OpenScreenAnimation Default

PrimaryColor Default

PrimaryColorDark Default

ScreenOrientation Unspecified

Rename Delete

Media Upload File ...

00:35 17/12/2020



江西理工大学信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Phone call. PhoneNumberPicker.

The screenshot shows the MIT App Inventor development environment. The top bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The URL in the browser is [www.BANDICAM.com](http://ai2.appinventor.mit.edu/#6390566104727552). The main workspace displays a smartphone screen titled "Screen1" with the text "Click to phone call". Below this, there is a list of labels: "Text for Label1", "Text for Label2", "Text for Label3", "Text for Label4", "Text for Label5", "Text for PhoneNumberPicker1", "Text for Label6", "Text for Label7", "Text for Label8", and "Text for Label9". The "Components" panel on the right lists the following components and their properties:

Component	Properties
Screen1	Label9 BackgroundColor: None FontBold: False FontItalic: False FontSize: 14.0 FontTypeface: default HTMLFormat: False HasMargins: True Height: Automatic... Width: Automatic... Text: Text for Label9 TextAlignment: left: 0 TextColor: #000000
Button1	
Label1	
Label2	
Label3	
Label4	
Label5	
PhoneNumberPicker1	
Label6	
Label7	
Label8	
Label9	
PhoneCall1	





Phone call.

PhoneNumberPicker.

File Edit View History Bookmarks Tools Help

www.BANDICAM.com

MIT App Inventor

ai2.appinventor.mit.edu/#6390566104727552

Getting Started PID Basic functions related... اموزش زبان C JRM | Fuji Technology ... (36) YouTube New Tab

MIT APP INVENTOR

Projects Connect Build Settings Help My Projects View Trash Guide Report an Issue English moshaydi@gmail.com

Phone_call_pick Screen1 Add Screen ... Remove Screen Publish to Gallery Designer Blocks

Blocks

Colors Variables Procedures

Screen1 Button1 Label1 Label2 Label3 Label4 Label5 PhoneNumberPicker Label6 Label7 Label8 Label9 PhoneCall1 Any component

Rename Delete Show Warnings

Viewer

```
blocks = initialize global message to " "
when Button1 Click
do set PhoneCall1 .PhoneNumber to "44444444"
call PhoneCall1 .MakePhoneCall

when PhoneCall1 .PhoneCallStarted
status phoneNumber
do if get status = 1
then set global message to "Incoming call"
else if get status = 2
then set global message to "Outcoming call"
set Label1 .Text to [join [Status] [get status] [get global message]]
set Label1 .Text to [join [Number] [get phoneNumber]]
```

```
when PhoneCall1 .PhoneCallEnded
status phoneNumber
do if get status = 1
then set global message to "Incoming call is missed or rejected"
else if get status = 2
then set global message to "Outcoming call is answered before hanging up"
else if get status = 3
then set global message to "Outcoming call is hanging up"
set Label3 .Text to [join [Status] [get status] [get global message]]
set Label4 .Text to [join [Number] [get phoneNumber]]
```

Designer Blocks

Upload File ...

Windows taskbar: File Explorer, Microsoft Edge, Microsoft Word, Microsoft Powerpoint, Mozilla Firefox, Spotify, OneDrive, Task View, Taskbar icons, System tray: ENG 00:47 17/12/2020



Phone call.

PhoneNumberPicker.

The screenshot shows the MIT App Inventor 2 interface with the following details:

- File Edit View History Bookmarks Tools Help** menu bar.
- www.BANDICAM.com** watermark at the top center.
- MIT App Inventor** tab.
- ai2.appinventor.mit.edu/#6390566104727552** URL in the address bar.
- 80%** zoom level.
- Getting Started PID Basic functions related... آموزش زبان C JRM | Fuji Technology ... (36) YouTube New Tab** tabs.
- APP INVENTOR** status bar.
- Phone_call_pick** project name.
- Screen1** screen selected.
- Blocks** tab.
- Blocks List:**
 - Colors
 - Variables
 - Procedures
 - Screen1
 - Button1
 - Label1
 - Label2
 - Label3
 - Label4
 - Label5
 - PhoneNumberPicker1
 - Label6
 - Label7
 - Label8
 - Label9
 - PhoneCall1
 - Any component
- Blocks View:** The main workspace displays several blocks of code:
 - An **initialize global [message]** block.
 - A **when Button1 .Click** event block with a **do** loop containing a **set PhoneCall1 .PhoneNumber** block to "44444444" and a **call PhoneCall1 .MakePhoneCall** block.
 - A **when PhoneCall1 .PhoneCallStarted** event block with a **do** loop containing an **if** condition block for **get status = 1**, setting **global message** to "Incoming call". It also handles cases for **status = 2** ("Outgoing call") and **status = 3** ("Hanging up").
 - A **when PhoneCall1 .PhoneCallEnded** event block with a **do** loop containing an **if** condition block for **get status = 1**, setting **global message** to "Incoming call is missed or rejected". It also handles cases for **status = 2** ("Answered before hung up") and **status = 3** ("Hanging up").
 - A **when PhoneCall1 .IncomingCallAnswered** event block with a **do** loop setting **Label5 .Text** to a joined string of **Status**, **get status**, **get global message**, **Number**, and **get phoneNumber**.
- Media:** Buttons for **Rename**, **Delete**, **Show Warnings**, and **Upload File ...**.
- Bottom Bar:** Includes links for **Privacy Policy and Terms of Use**, a date/time stamp (**05:54 17/12/2020**), and language/region settings (**ENG**).



Phone call.

PhoneNumberPicker.

File Edit View History Bookmarks Tools Help

www.BANDICAM.com

MIT App Inventor

ai2.appinventor.mit.edu/#6390566104727552

Getting Started PID Basic functions related... اموزش زبان C JRM | Fuji Technology ... (36) YouTube New Tab

My Projects View Trash Guide Report an Issue English moshaydi@gmail.com

Phone_call_pick Screen1 Add Screen ... Remove Screen Publish to Gallery Designer Blocks

Blocks

Viewer

```
when PhoneCall1 .PhoneCallStarted
  status phoneNumber
  do
    if get (status) = 1
      then set global message to "Incoming call"
    else if get (status) = 2
      then set global message to "Outcoming call"
    set Label1 .Text to (join ("Status", get status))
    set Label1 .Text to (join ("Number", get phoneNumber))
  initialize global phones to (create empty list)

when PhoneNumberPicker1 .AfterPicking
  do
    set Label6 .Text to PhoneNumberPicker1 .ContactName
    set Label7 .Text to PhoneNumberPicker1 .PhoneNumber
    set Label8 .Text to PhoneNumberPicker1 .EmailAddress
    set global phones to (create empty list)
    add items to list phones get global phones

else if get (status) = 2
then set global message to "Incoming call is answered before hand"
else if get (status) = 3
then set global message to "Outcoming call is hanging up"
set Label3 .Text to (join ("Status", get status))
set Label4 .Text to (join ("Number", get phoneNumber))

when PhoneCall1 .IncomingCallAnswered
  phoneNumber
  do
    set Label5 .Text to (join ("Phone Number incoming", get phoneNumber))
```

Screen1

Button1

Label1

Label2

Label3

Label4

Label5

PhoneNumberPicker1

Label6

Label7

Label8

Label9

PhoneCall

Any component

Rename Delete

Show Warnings

Upload File ...

Windows taskbar icons

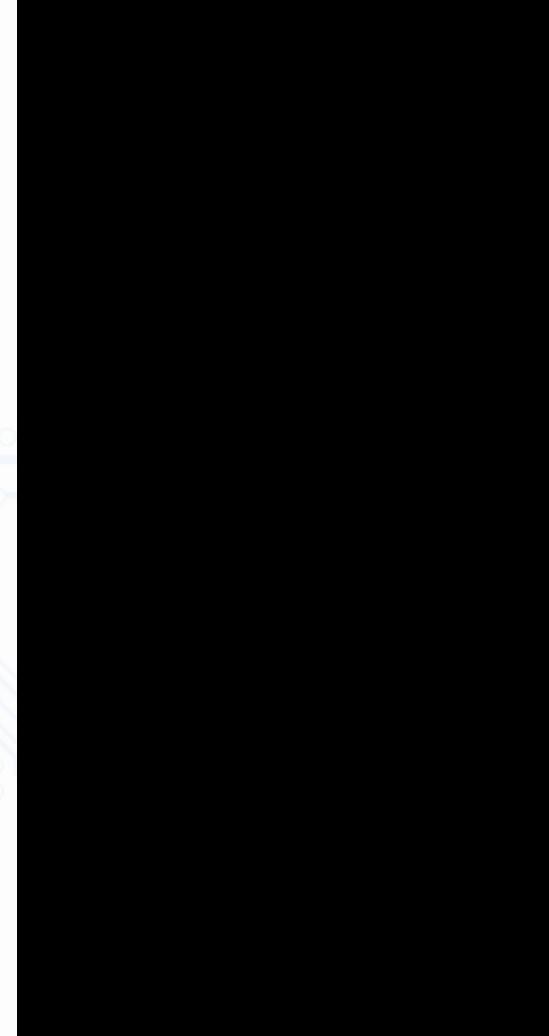
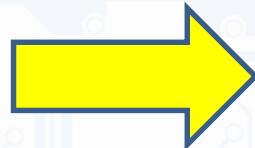
01:06 17/12/2020 ENG





Phone call. PhoneNumberPicker.

Demo APP



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



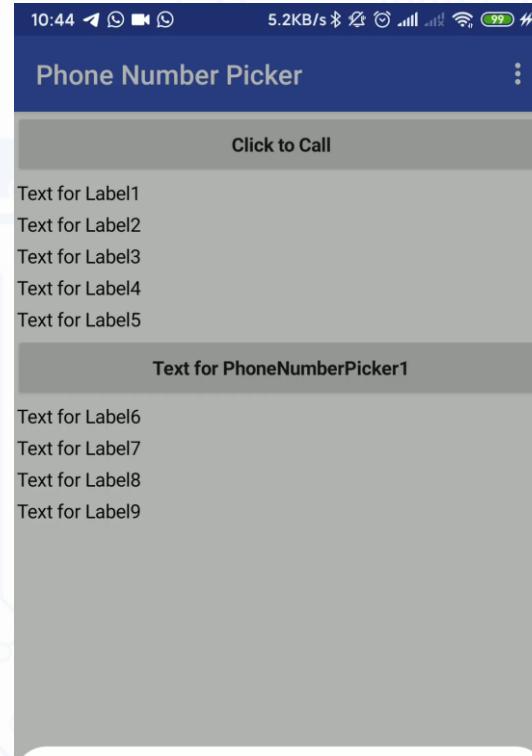
App Inventor



Example5_PhonePicker



OTHERS
DEMO



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING





© MIT App Inventor 2 • 課 MIT App Inventor 2

MIT App Inventor 2
90

Prepared exclusively for [REDACTED]

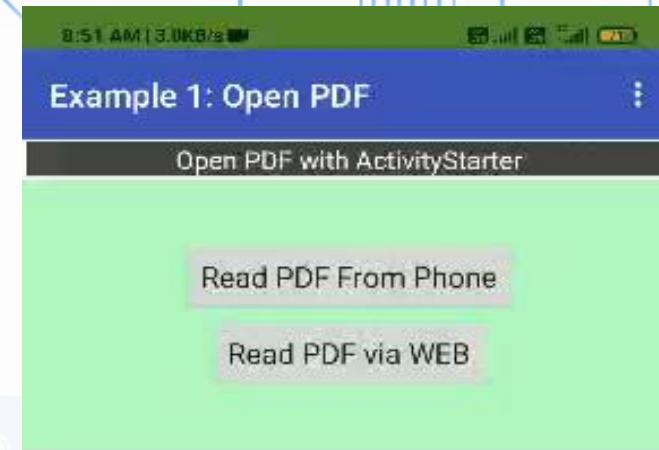
始作者：胡成、劉曉輝、周曉輝、周曉輝

The screenshot shows the Sencha Touch IDE interface. The top navigation bar includes tabs for 'Screen', 'Add Screen...', 'Delete screen...', 'Server', and 'Logout'. The left sidebar is titled 'Palette' and contains sections for 'User Interface' (Buttons, Checkbox, Interface, Image, Link, List, Upload), 'Layout' (Accordion, Box, Card, Content, Form, Grid, List, Panel, Tab, Tree, Viewport), 'Media' (Image, Video), 'Drawing and Animations' (Shapes, Social), 'Storage' (LocalStorage, Storage), and 'Connectivity' (Network, LocalStorage). The main workspace displays a mobile application screen with a title bar 'Sencha' and a content area. A tooltip 'Drag user interface components to View' is visible above the content area. The right sidebar is titled 'Properties' and shows settings for a 'Label' component, including 'Text' (set to 'Hello'), 'Background color' (set to 'white'), 'Outline color' (set to 'black'), 'Font' (set to 'Tita'), and 'FontSize' (set to '16'). Below the properties are sections for 'Media' (with 'UploadFile' selected), 'Localization' (with 'English' selected), and 'VersionCode' (set to '1').





Demo
of all





Student Task_16



- Make the report and repeat all the task in this ppt
- Solve the problem of example 3 with the comment and text
- Show the demo of all example
- Send the process clip on example 03
- Please send all file in all format with your name its better to send on zip file

- You have time to send your task before 9 am of lecture
- Send the file in PPT(power point format) to this email :

drajm@yahoo.com

- Your file should have this format of name
<Task number><student name><Student ID>.ppt

Next lecture



江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Reference

- <https://imagnity.com/tutorials/app-inventor/website-viewer-on-app-inventor/>
- <http://ai2.appinventor.mit.edu/reference/other/activitystarter.html>
- <https://appinventor.mit.edu/explore/ai2/concepts>
- <http://kio4.com/appinventori/23datasbetweenscreen.htm>
- <http://kio4.com/appinventori/7canvas.htm>
- <http://ai2.appinventor.mit.edu/reference/blocks/lists.html#selectlistitem>
- **[https://appinventor.mit.edu/explore/content/alertme.html](#)**
- **Teaching with AppInventor** <http://appinventor.mit.edu/explore/teach.html>
- **AppInventor Tutorials:**
<http://appinventor.mit.edu/explore/ai2/tutorials.html>
- **Sounds** <http://www.soundbible.com>
- **App Inventor:** <http://appinventor.googlelabs.com/>
- **Appinventor.org:** <http://www.appinventor.org/>
- **Wolber, Abelson et al. text:** <http://www.appinventor.org/text2011>
- **Group:** <http://groups.google.com/group/app-inventor-instructors>
- **Wolber course:** <http://appinventor.org/course-in-a-box>
- **Morelli course:** <http://turing.cs.trincoll.edu/~ram/cpsc110/>

“We are one
society. We are
one globe.”

STEVEN CHU
Nobel Prize in Physics 1997



江西理工大学

Jiangxi University of Science and Technology

信息工程学院

School of information engineering

Digital Image Processing

THANK YOU





**“BE HUMBLE. BE HUNGRY.
AND ALWAYS BE THE
HARDEST WORKER
IN THE ROOM.”**

