江西理工大學 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

Mobile application development

移动应用开发

Lecture 03:
APP Inventor Environment_ User interface

Dr Ata Jahangir Moshayedi

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

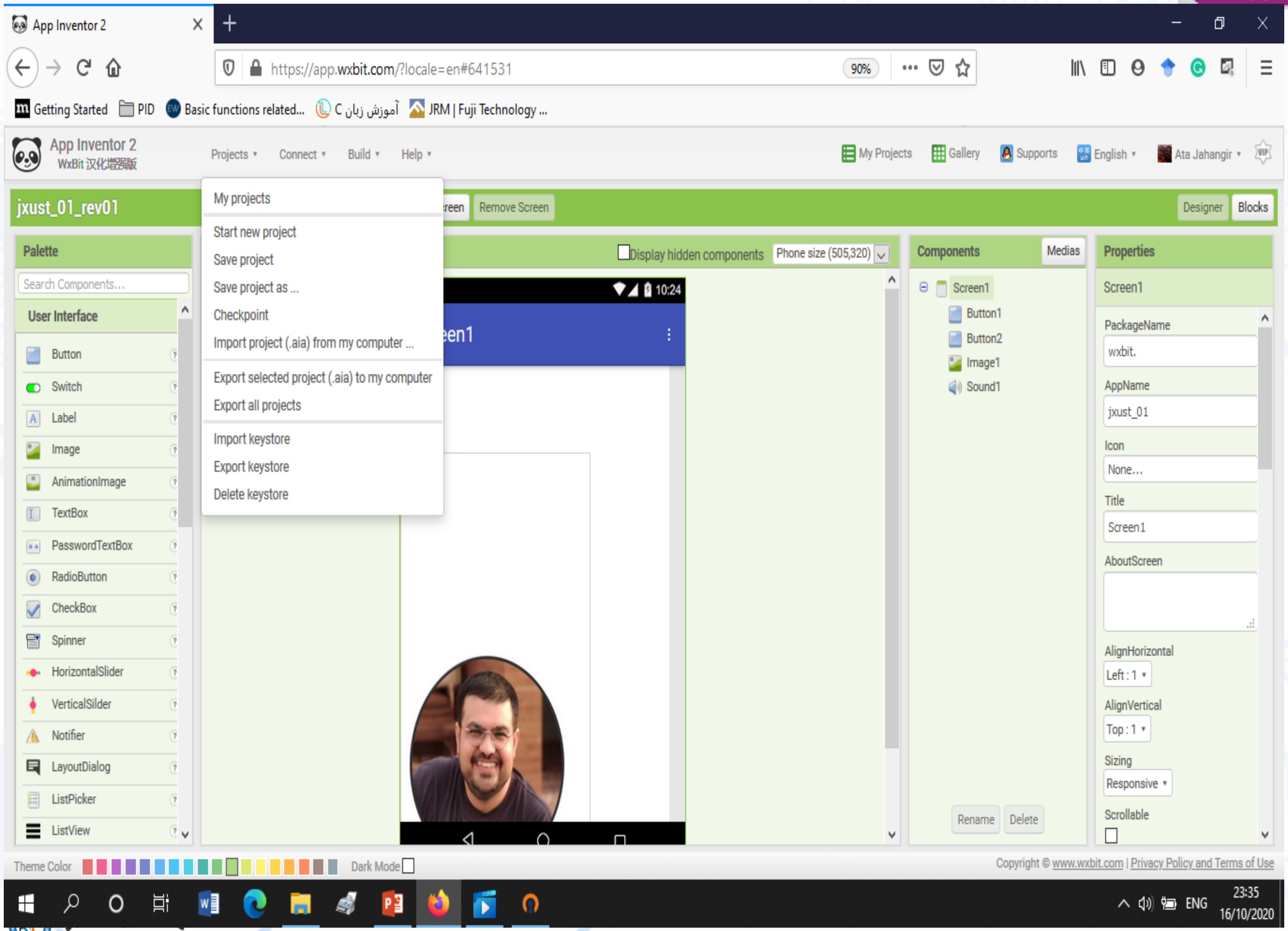
EMAIL: ajm@jxust.edu.cn

Autumn _2021

MOBILE APPLICATION DEVELOPMENT

LECTURE 03: APP Inventor Environment_ User interface

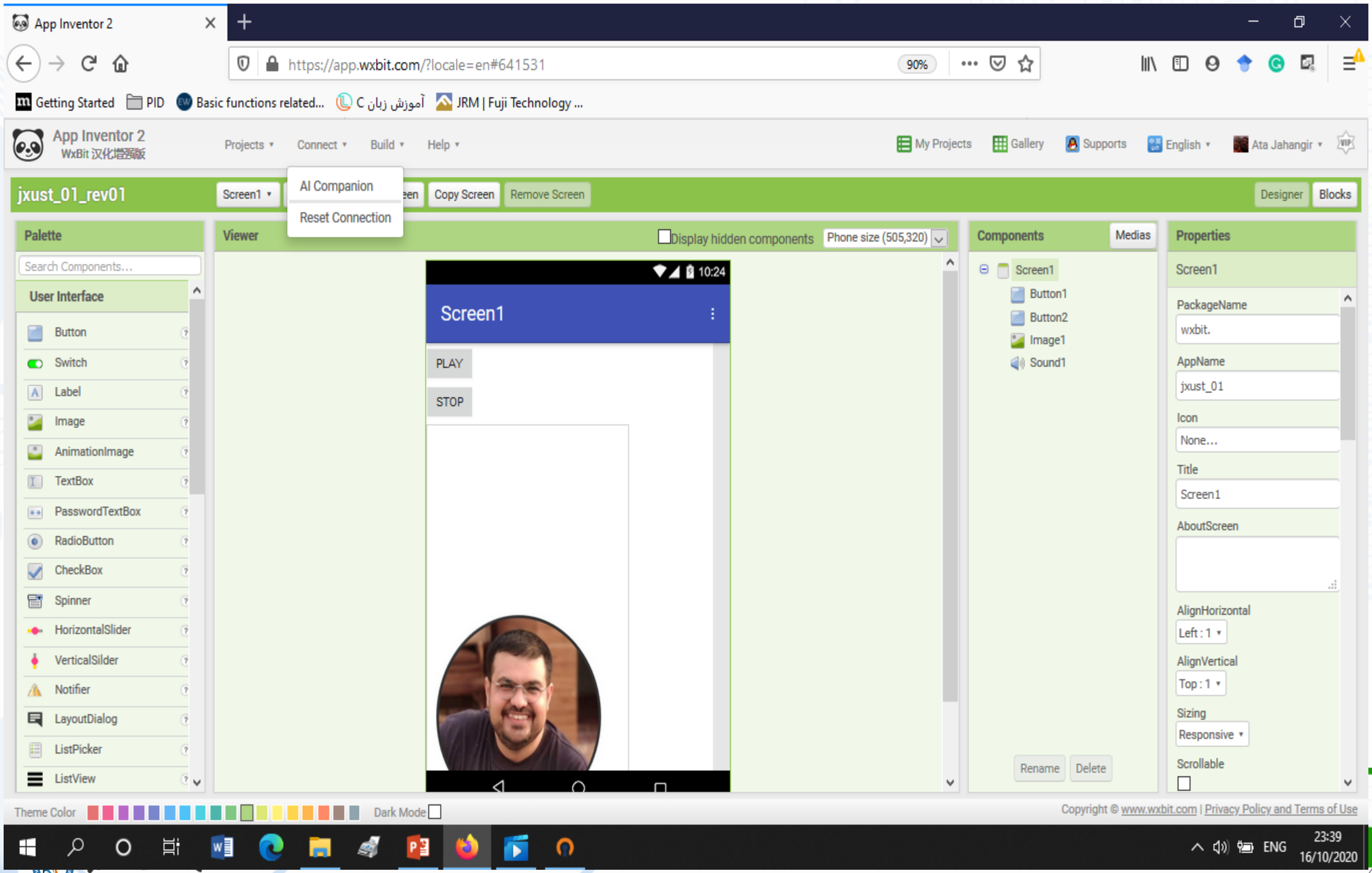
APP Invertor environment



Properties — Screen1

PackageName: wxbit.
AppName: jxust_01
Icon: None...
Title: Screen1
AboutScreen:
AlignHorizontal: Left : 1
AlignVertical: Top : 1
Sizing: Responsive
Scrollable: ☐

APP Invertor environment



App Inventor 2

https://app.wxbit.com/?locale=en#641531

Getting Started | PID | Basic functions related... | آموزش زبان C | JRM | Fuji Technology ...

App Inventor 2 WxBit 汉化增强版

Projects ▾ Connect ▾ Build ▾ Help ▾ | My Projects | Gallery | Supports | English ▾ | Ata Jahangir ▾

jxust_01_rev01 | Screen1 ▾ | Copy Screen | Remove Screen | Designer | Blocks

AI Companion
Reset Connection

Palette: Search Components...
User Interface: Button, Switch, Label, Image, AnimationImage, TextBox, PasswordTextBox, RadioButton, CheckBox, Spinner, HorizontalSlider, VerticalSilder, Notifier, LayoutDialog, ListPicker, ListView

Viewer: Display hidden components | Phone size (505,320)

Screen1: PLAY, STOP

Components: Screen1, Button1, Button2, Image1, Sound1 | Medias | Rename | Delete

Properties: Screen1

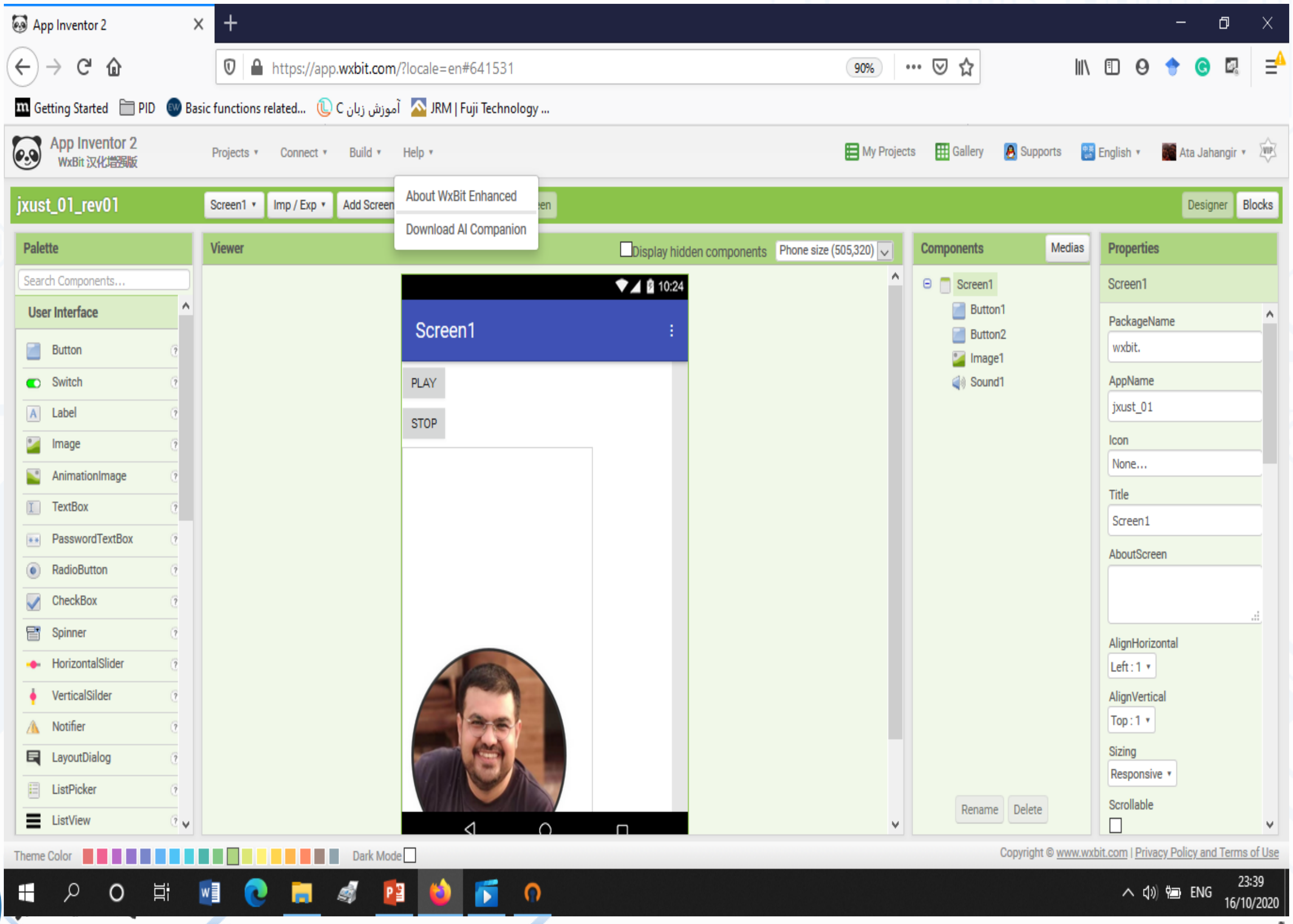
PackageName	wxbit.
AppName	jxust_01
Icon	None...
Title	Screen1
AboutScreen	
AlignHorizontal	Left : 1
AlignVertical	Top : 1
Sizing	Responsive
Scrollable	☐

Theme Color | Dark Mode ☐

Copyright © www.wxbit.com | Privacy Policy and Terms of Use

23:39 16/10/2020

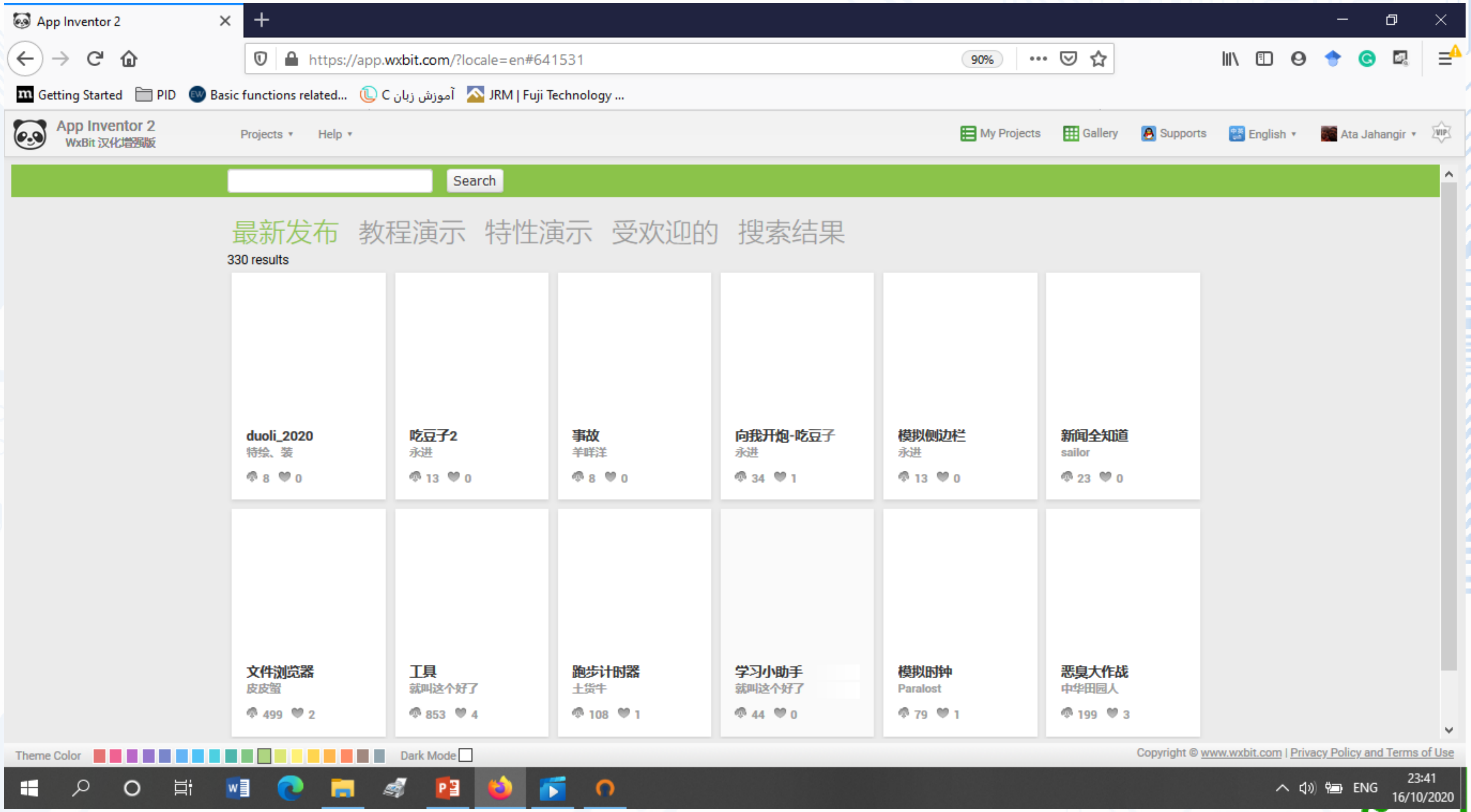
APP Invertor environment



JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

App Inventor

APP Invertor environment



App Inventor 2

https://app.wxbit.com/?locale=en#641531

App Inventor 2
WxBit 汉化增强版

Projects ▾ Help ▾ My Projects Gallery Supports English ▾ Ata Jahangir ▾

Search

最新发布 教程演示 特性演示 受欢迎的 搜索结果

330 results

duoli_2020 特绘、菱 8 0	吃豆子2 永进 13 0	事故 羊咩洋 8 0	向我开炮-吃豆子 永进 34 1	模拟侧边栏 永进 13 0	新闻全知道 sailor 23 0
文件浏览器 皮皮留 499 2	工具 就叫这个好了 853 4	跑步计时器 土货牛 108 1	学习小助手 就叫这个好了 44 0	模拟时钟 Paralost 79 1	恶臭大作战 中华田园人 199 3

Theme Color Dark Mode

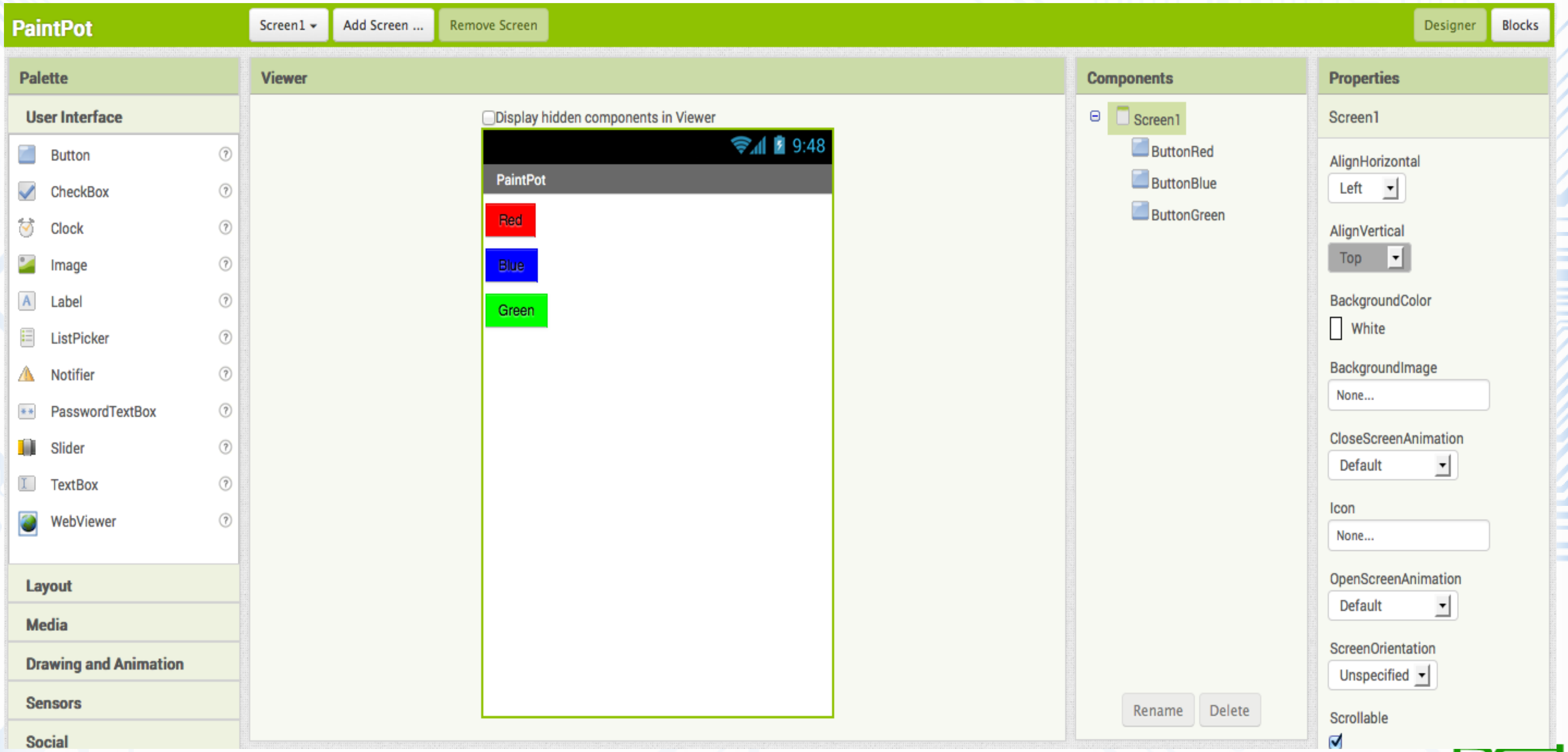
Copyright © www.wxbit.com | Privacy Policy and Terms of Use

23:41
16/10/2020

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

The installer environment

- Some parts are add in Chinese version



Screen1

AlignHorizontal: Left

AlignVertical: Top

BackgroundColor: White

BackgroundImage: None...

CloseScreenAnimation: Default

Icon: None...

OpenScreenAnimation: Default

ScreenOrientation: Unspecified

Scrollable: ✓

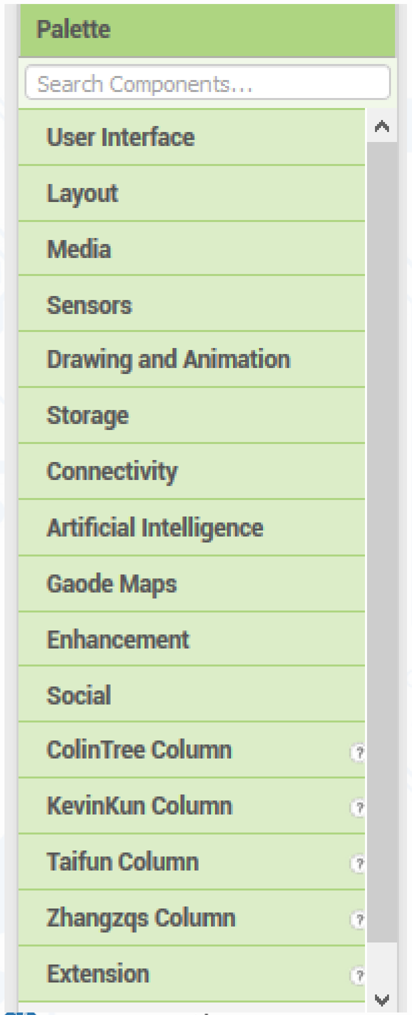
江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

App Inventor

APP Invertor environment

User Interface	Button	Drawing and Animation	Ball	Connectivity	ActivityStarter
	CheckBox		Canvas		BluetoothClient
	DatePicker		ImageSprite		BluetoothServer
	Image	Maps	Circle		Web
	Label		FeatureCollection	LEGO MINDSTORMS	NxtDirectCommands
	ListPicker		LineString		NxtColorSensor
	ListView		Map		NxtLightSensor
	Notifier		Marker		NxtSoundSensor
	PasswordTextBox		Polygon		NxtTouchSensor
	Screen		Rectangle		NxtUltrasonicSensor
	Slider	Sensor	AccelerometerSensor		NxtDrive
	Spinner		BarcodeScanner		Ev3Motors
	Switch		Clock		Ev3ColorSensor
	TextBox		GyroscopeSensor		Ev3GyroSensor
	TimePicker		LocationSensor		Ev3TouchSensor
Layout	HorizontalArrangement		NearField		Ev3UltrasonicSensor
	HorizontalScrollArrangement		OrientationSensor		Ev3Sound
	TableArrangement		Pedometer		Ev3UI
	VerticalArrangement		ProximitySensor		Ev3Commands
	VerticalScrollArrangement	Social	ContactPicker	Experimental	CloudDB
Media	Camcorder		EmailPicker		FirebaseDB
	Camera		PhoneCall		
	Image Picker		PhoneNumberPicker		
	Player		Sharing		
	Sound		Texting		
	Sound Recorder		Twitter		
	SpeechRecognizer	Storage	File		
	TextToSpeech		FusionTablesControl		
	VideoPlayer		TinyDB		
	YandexTranslate		TinyWebDB		

User Interface


Palette

Search Components...

User Interface

Layout

Media

Sensors

Drawing and Animation

Storage

Connectivity

Artificial Intelligence

Gaode Maps

Enhancement

Social

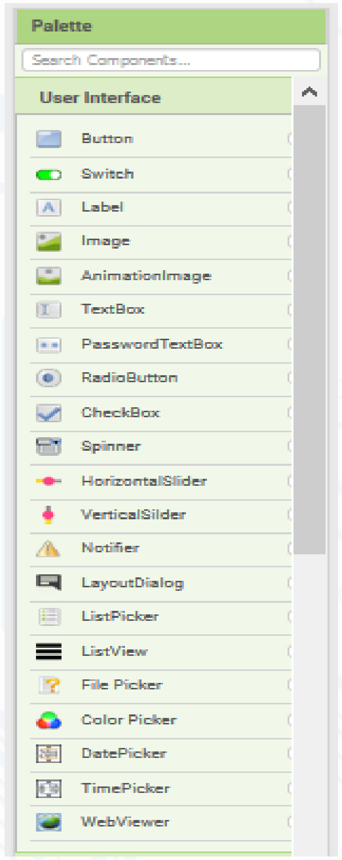
ColinTree Column

KevinKun Column

Taifun Column

Zhangzqs Column

Extension


Palette

Search Components...

User Interface

Button

Switch

Label

Image

AnimationImage

TextBox

PasswordTextBox

RadioButton

CheckBox

Spinner

HorizontalSlider

VerticalSilder

Notifier

LayoutDialog

ListPicker

ListView

File Picker

Color Picker

DatePicker

TimePicker

WebViewer

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

江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

User Interface

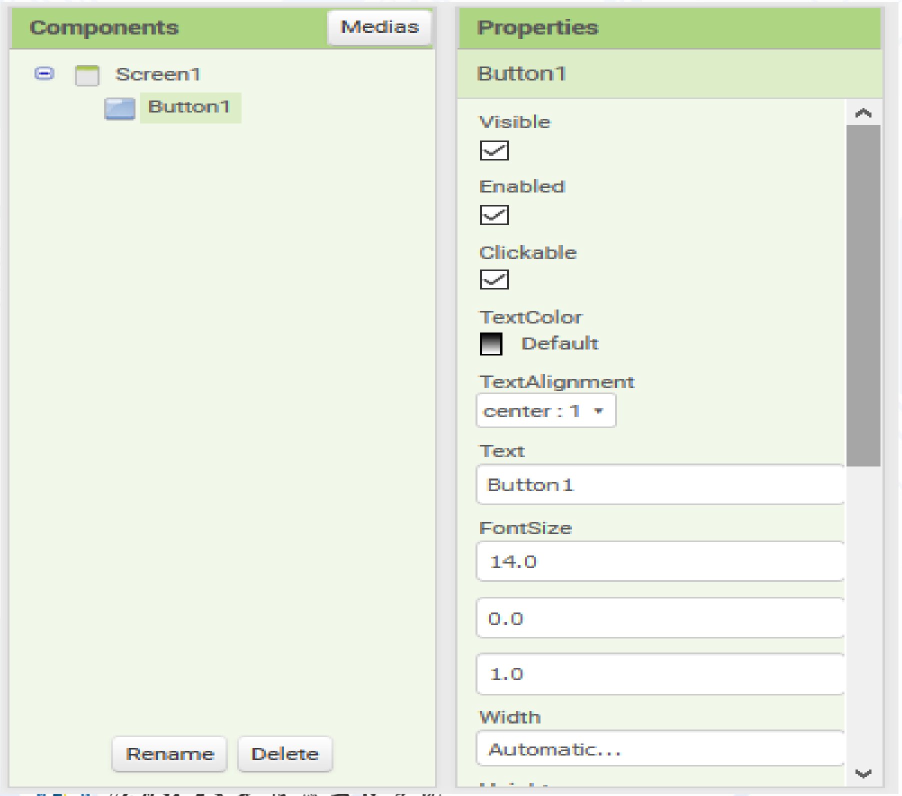
- Each part have bellow sections
- **Properties**
- **Events**
- **Methods**

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

App Inventor

Button

- Button with the ability to detect clicks.
- Many aspects of its appearance can be changed, as well as whether it is clickable (`Enabled`).
- Its properties can be changed in the Designer or in the Blocks Editor.



Components | Medias

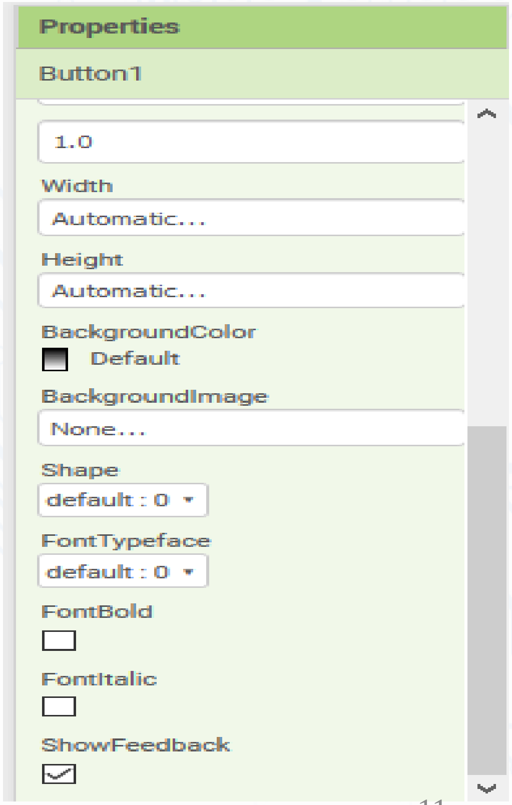
Screen1
Button1

Rename Delete

Properties

Button1

Visible ✓
Enabled ✓
Clickable ✓
TextColor: Default
TextAlignment: center : 1
Text: Button1
FontSize: 14.0
0.0
1.0
Width: Automatic...



Properties

Button1

1.0
Width: Automatic...
Height: Automatic...
BackgroundColor: Default
BackgroundImage: None...
Shape: default : 0
FontTypeface: default : 0
FontBold ☐
FontItalic ☐
ShowFeedback ✓

Button _Events

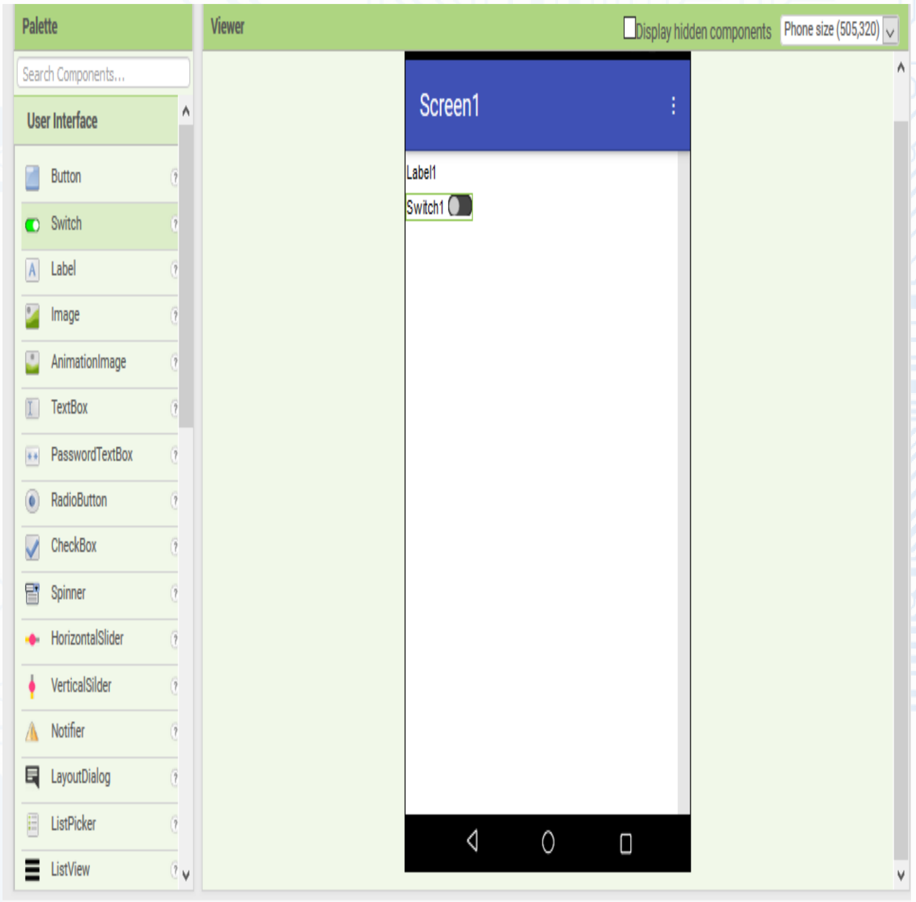
Click()	Indicates that the user tapped and released the Button.
GotFocus()	Indicates the cursor moved over the Button so it is now possible to click it.
LongClick()	Indicates that the user held the Button down.
LostFocus()	Indicates the cursor moved away from the Button so it is now no longer possible to click it.
TouchDown()	Indicates that the Button was pressed down.
TouchUp()	Indicates that the Button has been released.

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

App Inventor

Switch

- Switch components can detect user taps and can change their boolean state in response.
- They are identical to CheckBoxes except in appearance.
- Switches have an on (true) state and an off (false) state.
- A Switch component raises an event when the user taps it to toggle between states.



Label

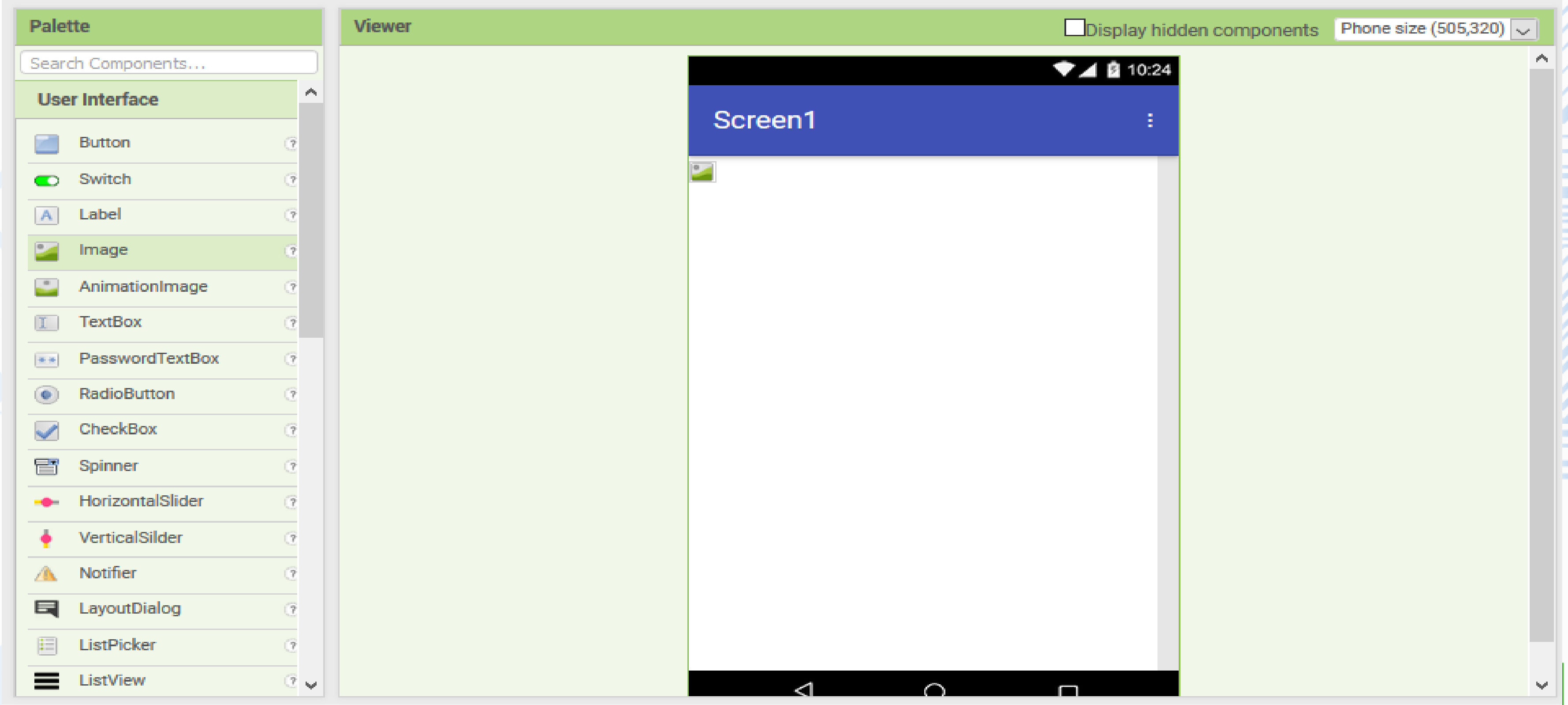
- Labels are components used to show text.
- A label displays text which is specified by the Text property. Other properties, all of which can be set in the Designer or Blocks Editor, control the appearance and placement of the text.

Label

BackgroundColor	Specifies the label’s background color as an alpha-red-green-blue integer.
FontBold	Specifies whether the label’s text should be bold. Some fonts do not support bold.
FontItalic	Specifies whether the label’s text should be italic. Some fonts do not support italic.
FontSize	Specifies the label’s text’s font size, measured in sp(scale-independent pixels).
FontTypeface	Specifies the label’s text’s font face as default, serif, sans serif, or monospace.
HTMLContent	Returns the content of the Label as HTML. This is only useful if the HTMLFormat property is true.
HTMLFormat	Specifies the label’s text’s format
HasMargins	Specifies whether the label should have margins. This margin value is not well coordinated with the designer, where the margins are defined for the arrangement, not just for individual labels.
Height	Specifies the Label’s vertical height, measured in pixels.
HeightPercent	Specifies the Label’s vertical height as a percentage of the Screen’s Height.
Text	Specifies the text displayed by the label.
TextAlignment	Specifies the alignment of the label’s text: center, normal (e.g., left-justified if text is written left to right), or opposite (e.g., right-justified if text is written left to right).
TextColor	Specifies the label’s text color as an alpha-red-green-blue integer.
Visible	Specifies whether the Label should be visible on the screen. Value is true if the Label is showing and false if hidden.
Width	Specifies the horizontal width of the Label, measured in pixels.
WidthPercent	Specifies the horizontal width of the Label as a percentage of the Screen’s Width.

Image

- Component for displaying images and basic animations.
- The picture to display, and other aspects of the Image's appearance, can be specified in the Designer or in the Blocks Editor.

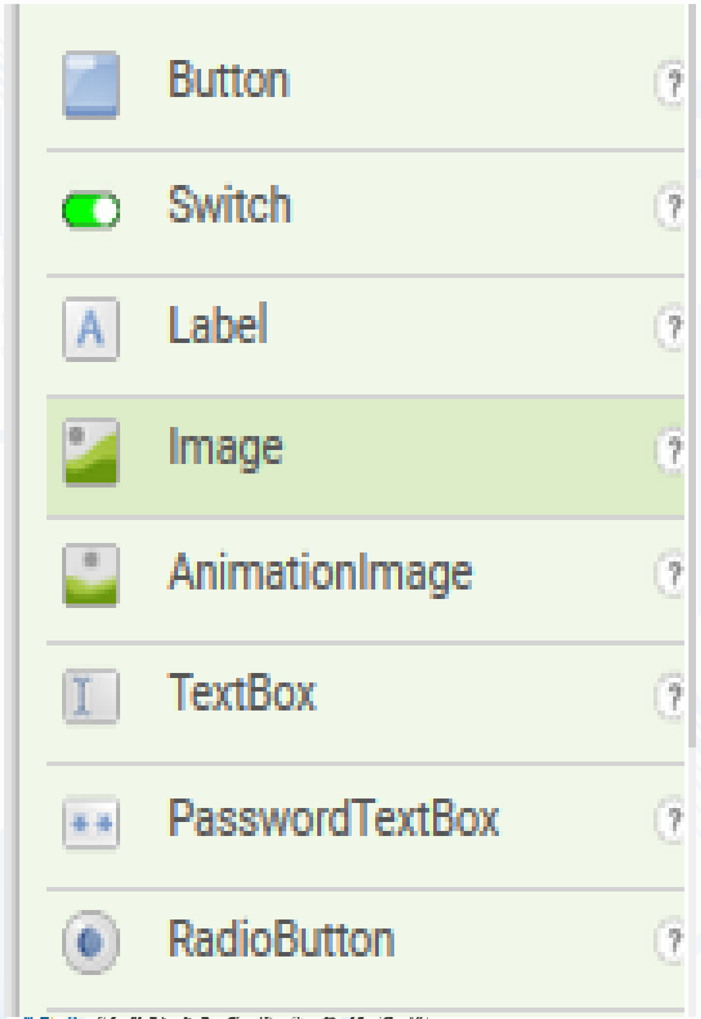


Image

Animation	This is a limited form of animation that can attach a small number of motion types to images. The allowable motions are ScrollRightSlow, ScrollRight, ScrollRightFast, ScrollLeftSlow, ScrollLeft, ScrollLeftFast, and Stop.
Clickable	Specifies whether the image should be clickable or not.
Height	Specifies the Image’s vertical height, measured in pixels.
HeightPercent	Specifies the Image’s vertical height as a percentage of the Screen’s Height.
Picture	Specifies the path of the Image’s Picture.
RotationAngle	The angle at which the image picture appears rotated. This rotation does not appear on the designer screen, only on the device.
ScalePictureToFit	Specifies whether the image should be resized to match the size of the ImageView.
Scaling	This property determines how the picture scales according to the Height or Width of the Image. Scale proportionally (0) preserves the picture aspect ratio. Scale to fit (1) matches the Image area, even if the aspect ratio changes.
Visible	Specifies whether the Image should be visible on the screen. Value is true if the Image is showing and false if hidden.
Width	Specifies the horizontal width of the Image, measured in pixels.
WidthPercent	Specifies the horizontal width of the Image as a percentage of the Screen’s Width.

Events :
Click() An event that occurs when an image is clicked.

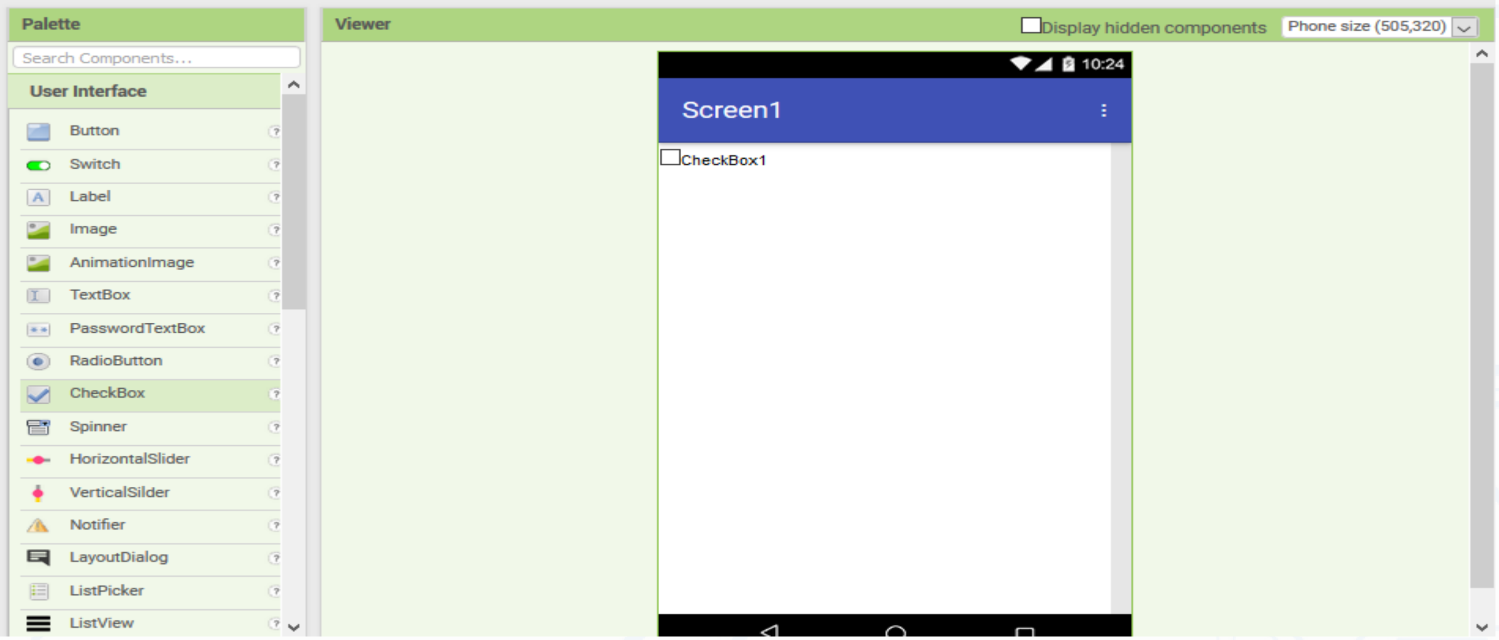
User interface extra….


Button
Switch
Label
Image
AnimationImage
TextBox
PasswordTextBox
RadioButton

SOME PARTS ARE EXTRA

CheckBox

- CheckBox components can detect user taps and can change their boolean state in response.
- A CheckBox component raises an event when the user taps it. There are many properties affecting its appearance that can be set in the Designer or Blocks Editor.



CheckBox_events

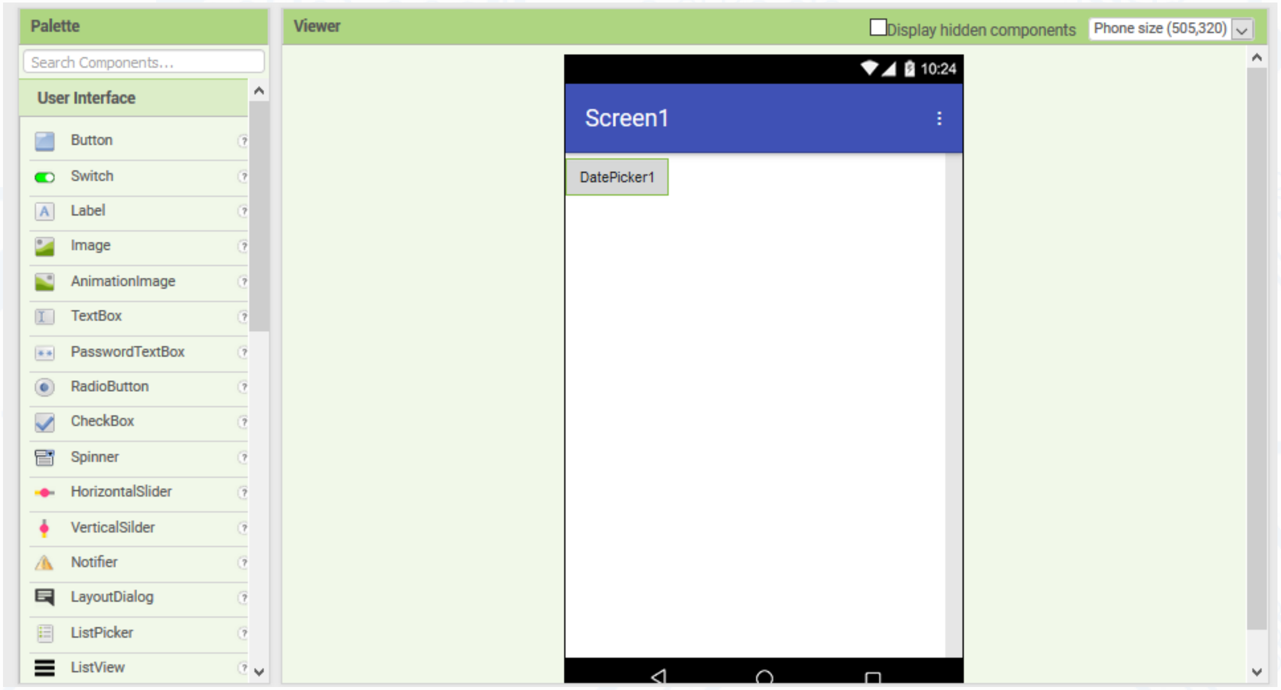
Changed()	User tapped and released the CheckBox.
GotFocus()	CheckBox became the focused component.
LostFocus()	CheckBox stopped being the focused component.

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

App Inventor

DatePicker

- A button that, when clicked on, launches a popup dialog to allow the user to select a date on the Gregorian Calendar.
- Note: Date and time are manipulated using methods in the Clock component.



App Inventor

DatePicker Events/Methods

AfterDateSet()	Event that runs after the user chooses a Date in the dialog.
GotFocus()	Indicates the cursor moved over the DatePicker so it is now possible to click it.
LostFocus()	Indicates the cursor moved away from the DatePicker so it is now no longer possible to click it.
TouchDown()	Indicates that the DatePicker was pressed down.
TouchUp()	Indicates that the DatePicker has been released.

Methods

LaunchPicker()	Launches the DatePicker dialog. The AfterDateSet event will be run after the user confirms their selection.
SetDateToDisplay(year,month,day)	Allows the user to set the date to be displayed when the date picker opens. Valid values for the month field are 1-12 and 1-31 for the day field.
SetDateToDisplayFromInstant(instant)	Allows the user to set the date from the instant to be displayed when the date picker opens.

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

App Inventor

ListPicker

- A button that, when clicked on, displays a list of texts for the user to choose among.
- The texts can be specified through the Designer or Blocks Editor by setting the ElementsFromString property to their string-separated concatenation (for example, choice 1, choice 2, choice 3) or by setting the Elements property to a List in the Blocks editor.
- Setting property ShowFilterBar to true, will make the list searchable. Other properties affect the appearance of the button (TextAlignment, BackgroundColor, etc.) and whether it can be clicked on (Enabled).

ListPicker

BackgroundColor	Specifies the ListPicker’s background color as an alpha-red-green-blue integer. If an Image has been set, the color change will not be visible until the Image is removed.
Elements	Specifies the list of choices to display.
ElementsFromString	Set the list of choices from a string of comma-separated values.
Enabled	Specifies whether the ListPicker should be active and clickable.
FontBold	Specifies whether the text of the ListPicker should be bold. Some fonts do not support bold.
FontItalic	Specifies whether the text of the ListPicker should be italic. Some fonts do not support italic.
FontSize	Specifies the text font size of the ListPicker, measured in sp(scale-independent pixels).
FontTypeface	Specifies the text font face of the ListPicker as default, serif, sans serif, or monospace.
Height	Specifies the ListPicker’s vertical height, measured in pixels.
HeightPercent	Specifies the ListPicker’s vertical height as a percentage of the Screen’s Height.
Image	Specifies the path of the ListPicker’s image. If there is both an Image and a BackgroundColor specified, only the Image will be visible.
ItemBackgroundColor	The background color of the ListPicker items.
ItemTextColor	The text color of the ListPicker items.
Selection	The selected item. When directly changed by the programmer, the SelectionIndex property is also changed to the first item in the ListPicker with the given value. If the value is not in Elements, SelectionIndex will be set to 0.
SelectionIndex	Selection index property setter method.
Shape	Specifies the shape of the ListPicker. The valid values for this property are 0 (default), 1 (rounded), 2 (rectangle), and 3 (oval). The Shape will not be visible if an Image is used.
ShowFeedback	Specifies if a visual feedback should be shown when a ListPicker with an assigned Image is pressed.
ShowFilterBar	If true, the ListPicker will show a search filter bar.
Text	Specifies the text displayed by the ListPicker.
TextAlignment	Specifies the alignment of the ListPicker’s text. Valid values are: 0 (normal; e.g., left-justified if text is written left to right), 1 (center), or 2 (opposite; e.g., right-justified if text is written left to right).
TextColor	Specifies the text color of the ListPicker as an alpha-red-green-blue integer.
Title	Optional title displayed at the top of the list of choices.
Visible	Specifies whether the ListPicker should be visible on the screen. Value is true if the ListPicker is showing and false if hidden.
Width	Specifies the horizontal width of the ListPicker, measured in pixels.
WidthPercent	Specifies the horizontal width of the ListPicker as a percentage of the Screen’s Width.

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING
App Inventor

ListPicker **Events/Methods**

ListPicker **Events**

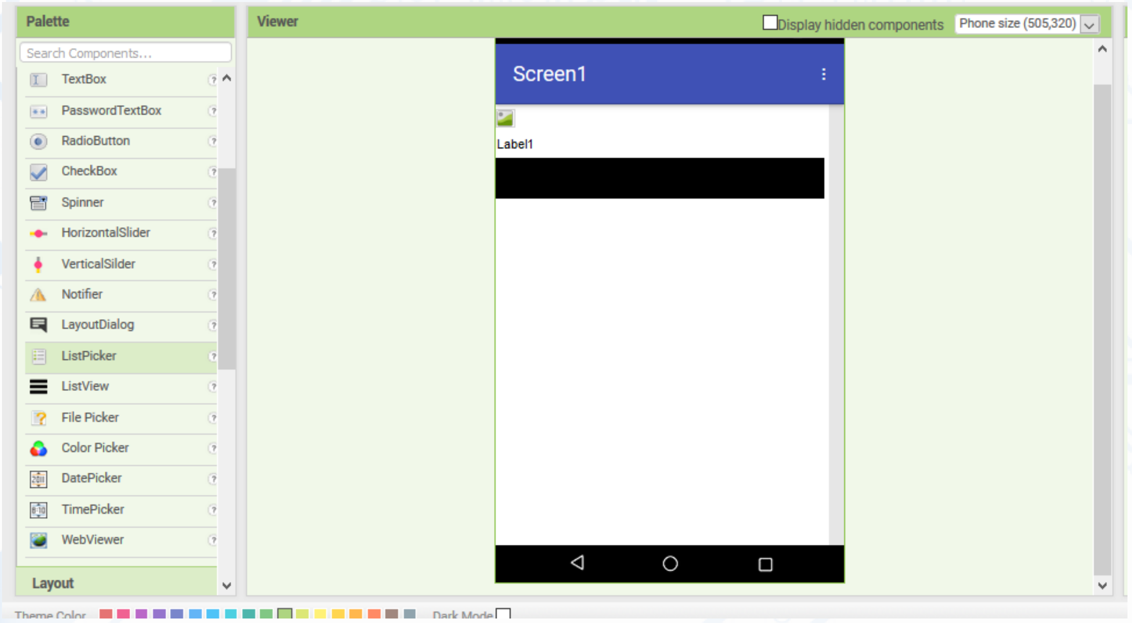
AfterPicking()	Event to be raised after the ListPicker activity returns its result and the properties have been filled in.
BeforePicking()	Event to raise when the ListPicker is clicked or the picker is shown using the Open method. This event occurs before the picker is displayed, and can be used to prepare the picker before it is shown.
GotFocus()	Indicates the cursor moved over the ListPicker so it is now possible to click it.
LostFocus()	Indicates the cursor moved away from the ListPicker so it is now no longer possible to click it.
TouchDown()	Indicates that the ListPicker was pressed down.
TouchUp()	Indicates that the ListPicker has been released.

Methods

Open()	Opens the ListPicker, as though the user clicked on it.

ListView

- This is a visible component that allows to place a list of text elements in your Screen to display.
- The list can be set using the ElementsFromString property or using the Elements block in the blocks editor.
- **Warning: This component will not work correctly on Screens that are scrollable if its Height is set to Fill Parent.**



Events

AfterPicking()	Simple event to be raised after the an element has been chosen in the list. The selected element is available in the Selection property.

Notifier

- The Notifier component displays alert messages and creates Android log entries through an assortment of methods.

BackgroundColor	Specifies the background color for alerts (not dialogs).
NotifierLength	Specifies the length of time that the alert is shown – either “short” or “long”.
TextColor	Specifies the text color for alerts (not dialogs).

AfterChoosing(choice)	Event after the user has made a selection for ShowChooseDialog.
AfterTextInput(response)	Event raised after the user has responded to ShowTextDialog
ChoosingCanceled()	Event raised when the user cancels choosing an option. ShowChooseDialog.
TextInputCanceled()	Event raised when the user cancels ShowChooseDialog, ShowPasswordDialog, or ShowTextDialog.

DismissProgressDialog()	Dismisses the alert created by the ShowProgressDialog block
LogError(message)	Writes an error message to the Android system log. See the Google Android documentation for how to access the log.
LogInfo(message)	Writes an information message to the Android log.
LogWarning(message)	Writes a warning message to the Android log. See the Google Android documentation for how to access the log.
ShowAlert(notice)	Display a temporary notification.
ShowChooseDialog(message,title,button1Text,button2Text,cancelable)	Shows a dialog box with two buttons, from which the user can choose. If cancelable is true there will be an additional CANCEL button. Pressing a button will raise the AfterChoosing event. The "choice" parameter to AfterChoosing will be the text on the button that was pressed, or "Cancel" if the CANCEL button was pressed. If canceled, the TextInputCanceled event will also run.
ShowMessageDialog(message,title,buttonText)	Display an alert dialog with a single button that dismisses the alert.
ShowPasswordDialog(message,title,cancelable)	Shows a dialog box where the user can enter password (input is masked), after which the AfterTextInput event will be raised. If cancelable is true there will be an additional CANCEL button. The AfterTextInput and TextInputCanceled events behave the same way as described in ShowTextDialog.
ShowProgressDialog(message,title)	Shows a dialog box with an optional title and message (use empty strings if they are not wanted). This dialog box contains a spinning artifact to indicate that the program is working. It cannot be canceled by the user but must be dismissed by the App Inventor Program by using the DismissProgressDialog method.
ShowTextDialog(message,title,cancelable)	Shows a dialog box where the user can enter text, after which the AfterTextInput event will be raised. If cancelable is true there will be an additional CANCEL button. Entering text will raise the AfterTextInput event. The "response" parameter to AfterTextInput will be the text that was entered, or "Cancel" if the CANCEL button was pressed. If canceled, the TextInputCanceled event will also run.

PasswordTextBox

- Users enter passwords in a password text box component, which hides the text that has been typed in it.
  - A password text box is the same as the ordinary TextBox component, except that it does not display the characters typed by the user.
  - You can get or set the value of the text in the box with the Text property.
  - If Text is blank, you can use the Hint property to provide the user with a suggestion of what to type. The Hint appears as faint text in the box.
  - Password text box components are usually used with a Button component. The user taps the Button after entering text.

PasswordTextBox

Events

GotFocus()	Event raised when the PasswordTextBox is selected for input, such as by the user touching it.
LostFocus()	Event raised when the PasswordTextBox is no longer selected for input, such as if the user touches a different text box.

Methods

RequestFocus()	Request focus to current PasswordTextBox.

App Inventor

Slider

- This class is used to display a Slider.
- A Slider is a progress bar that adds a draggable thumb. You can touch the thumb and drag left or right to set the slider thumb position.
- As the Slider thumb is dragged, it will trigger the PositionChanged event, reporting the position of the Slider thumb.

- The reported position of the thumb can be used to dynamically update another component attribute, such as the TextBox's FontSize of a TextBox or the Radius of a Ball.

- The Slider uses the following default values. However these values can be changed through the Designer or Blocks editor:

  MinValue = 10

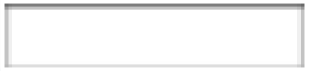
  MaxValue = 50

  ThumbPosition = 30

Spinner

- A Spinner component that displays a dialog with a list of elements.
- These elements can be set in the Designer or Blocks Editor by setting the ElementsFromString property to a comma-separated list of values (for example, choice 1, choice 2, choice 3) or by setting the Elements property to a List in the Blocks editor.
- Spinners are created with the first item already selected, so selecting it does not generate an AfterSelecting event.
- Consequently it’s useful to make the first Spinner item be a non-choice like “Select from below…”.

TextBox



- Users enter text in a text box component.
- The initial or user-entered text value in a text box component is in the Text property. If Text is blank, you can use the Hint property to provide the user with a suggestion of what to type.
- The Hint appears as faint text in the box.
- The MultiLine property determines if the text can have more than one line. For a single line text box, the keyboard will close automatically when the user presses the Done key. To close the keyboard for multiline text boxes, the app should use the HideKeyboard method or rely on the user to press the Back key.

TextBox

- The NumbersOnly property restricts the keyboard to accept numeric input only.
- Other properties affect the appearance of the text box (TextAlignment, BackgroundColor, etc.) and whether it can be used (Enabled).
- Text boxes are usually used with the Button component, with the user clicking on the Button when text entry is complete.
  - If the text entered by the user should not be displayed, use PasswordTextBox instead.

TimePicker

- A button that, when clicked on, opens a dialog to allow the user to select a time.
- Note: Date and time are manipulated using methods in the Clock component.

WebViewer

- Component for viewing Web pages.
- The HomeUrl can be specified in the Designer or in the Blocks Editor. The view can be set to follow links when they are tapped, and users can fill in Web forms.
- **Warning: This is not a full browser. For example, pressing the phone's hardware Back key will exit the app, rather than move back in the browser history.**
- You can use the WebViewString property to communicate between your app and Javascript code running in the WebViewer page.
- In the app, you get and set WebViewString.
- In the WebViewer, you include Javascript that references the window.
- AppInventor object, using the methods getWebViewString() and setWebViewString(text).

WebViewer

For example

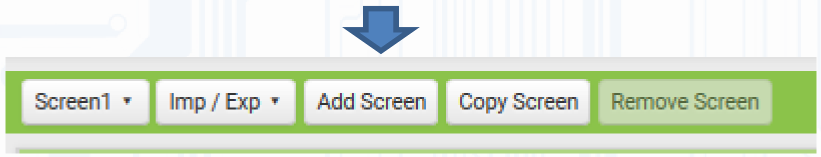
- if the Web Viewer opens to a page that contains the JavaScript command document. Write("The answer is" + window.AppInventor.getWebViewString()); and if you set WebViewString to “hello”, then the web page will show The answer is hello.
- And if the Web page contains JavaScript that executes the command windowAppInventor.setWebViewString("hello from JavaScript"),then the value of the Web View String property will be hello from JavaScript.

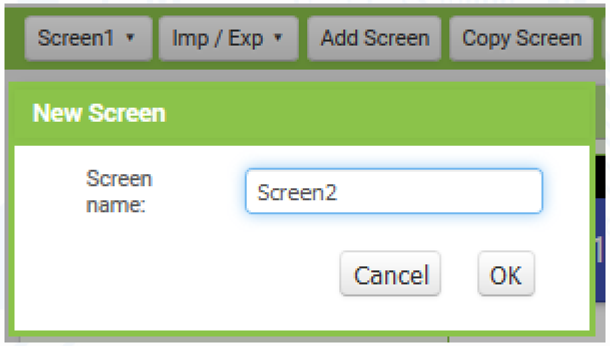
Calling set WebViewString from JavaScript will also run the Web View String Change event so that the blocks can handle when the WebViewString property changes.

Beginning with release nb184a, you can specify a Home Url beginning with http://localhost/ to reference assets both in the Companion and in compiled apps. Previously, apps needed to use file:///android_asset/ in compiled apps and /sdcard/AppInventor/assets/ in the Companion. Both of these options will continue to work but the http://localhost/ approach will work in both scenarios. You may also use “file:///appinventor_asset/” which provides more security by preventing the use of asynchronous requests from JavaScript in your assets from going out to the web.

How do you setup an app with multiple screens?

- You can add a new screen in the Designer by clicking on “Add Screen…” Suppose you named the screen "Home Screen".
- Then you can add a button on it called "Home Button".
- When you click Home Button, use the 'open another screen' block to swap to Screen1.
- You can provide a similar mechanism on Screen1 to swap to Home Screen.

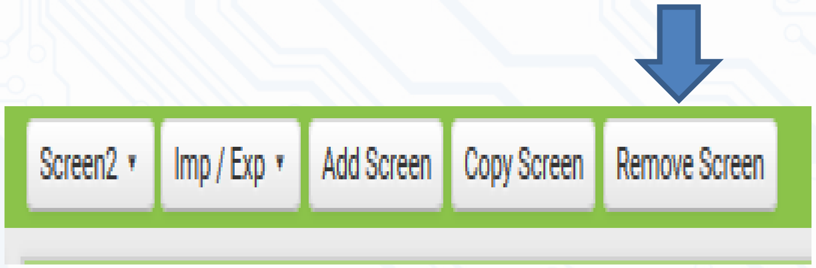




New Screen

Screen name: Screen2

Cancel OK



Switching Screens

To open another screen, you use the block under the Control palette called **open another screen**. This block requires one input, which must be the name of the screen you want to open, in a text block.
See figure for the programming that opens Screen2 when a button is pressed.

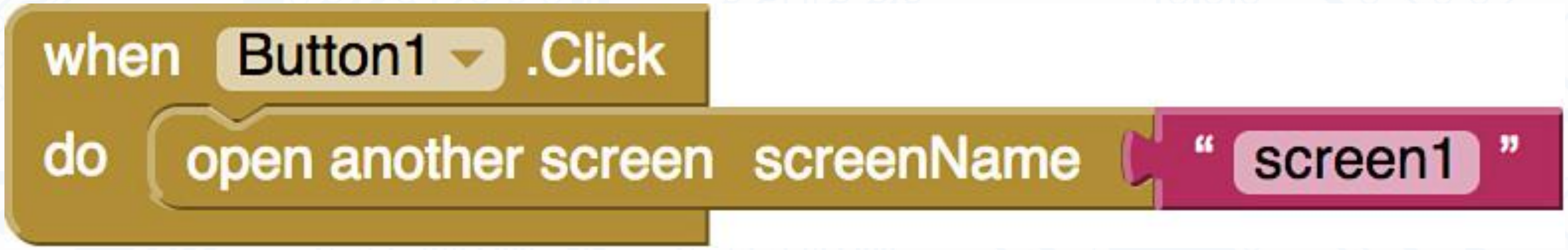
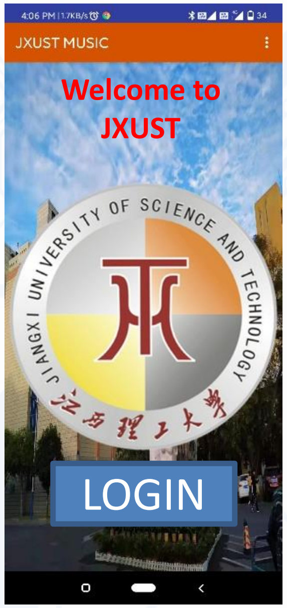
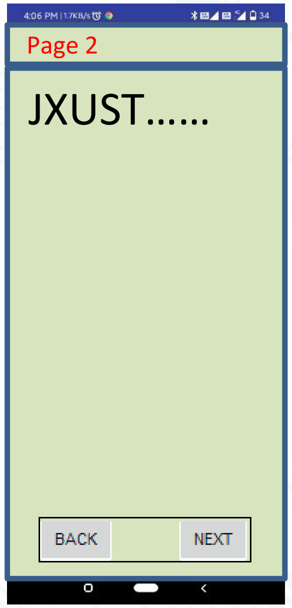


Figure Programming an app to switch screens.
Screen behavior is like a small deck of cards on a table. You start with just one card, which is Screen1, the first screen you see. When you open another screen, that screen opens on top of the previous one, like another card being put on the deck

Student Task_3

1. Make the below GUI
2. When press the Login enter to the next page which have your image





Send ppt and doc file +the process of design app clip should be inside ppt

- 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

Reference

- **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/

江西理工大学

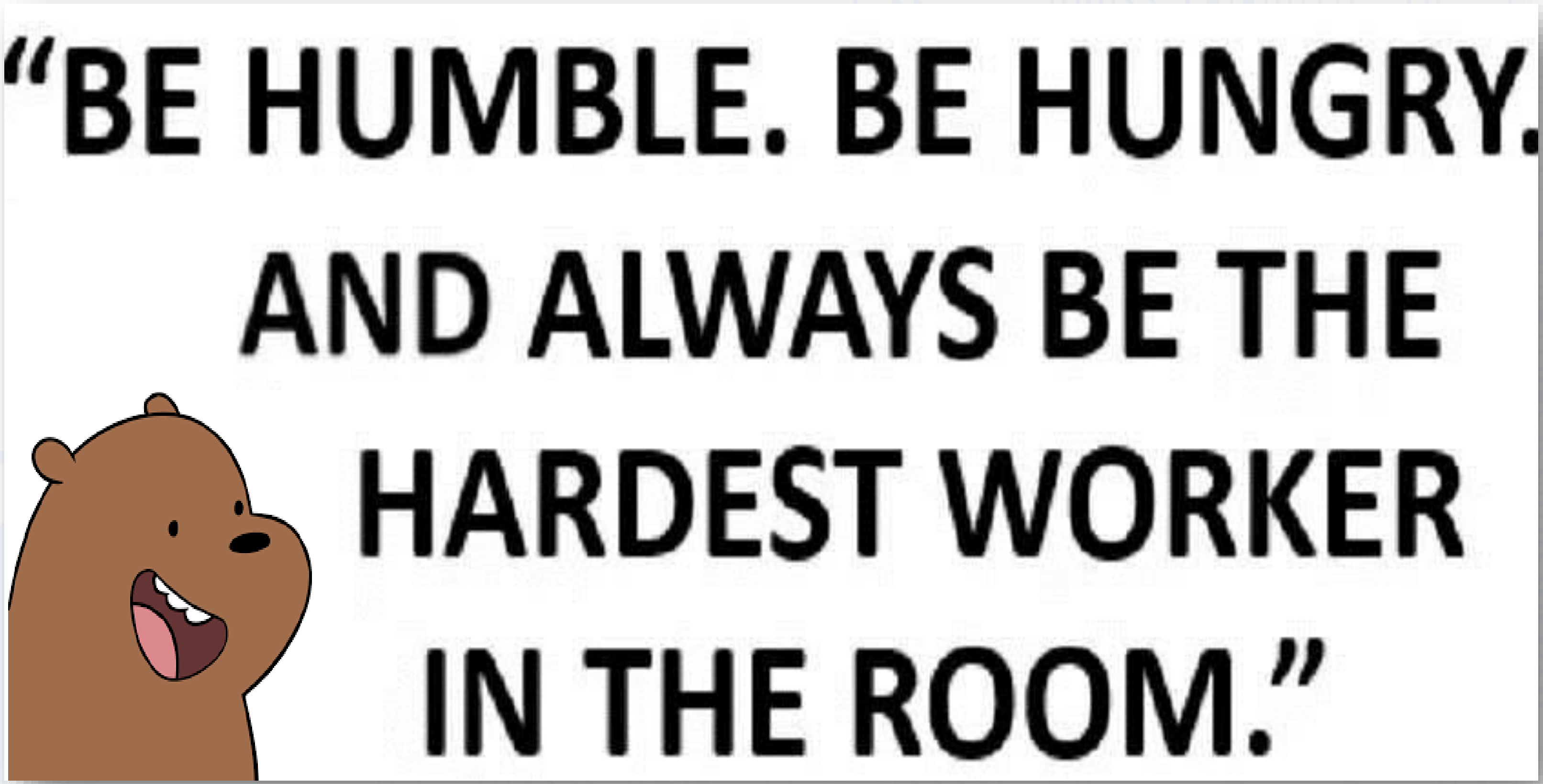
Jiangxi University of Science and Technology

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."

"The beauty of research is that you never know where it's going to lead."

RICHARD ROBERTS
Nobel Prize in Physiology or Medicine 1993