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

移动应用开发

Lecture 05:
APP Inventor_environment_B

Dr Ata Jahangir Moshayedi

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

EMAIL: ajm@jxust.edu.cn

Autumn _2021

Help me to have the better lecture:

1. some student report the problems to use the mooc I will update the ppt on this link but please check the mooc also: https://github.com/drajm/Mobile-development

   Please check the MOOC also it will check by university

2. I will answer your question but Pleas try to ask your relevant question in lecture time some student ask 1 am and expected answer.Even if you want we can confirm one day with the specific time for your question

3.Just send your task on MOOC the receive task in WeChat will be delete.

4.Consider the task assignment time and format after that time the system will not permit to send.

5.Please remember most of your final mark will be carried by task

For task

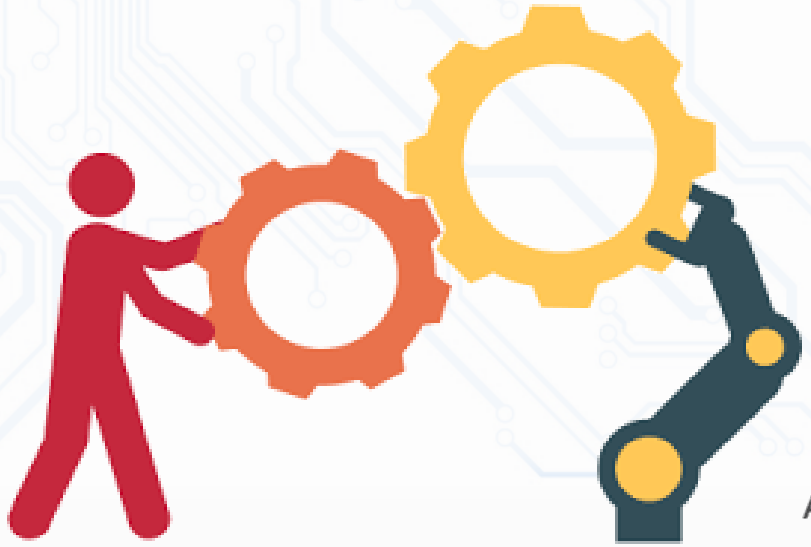
- **Just MOOC**
- **Just PPT with the format**
- **Just task**

JUST FOR STUDENTS WHO HAVE COURSES WITH ME

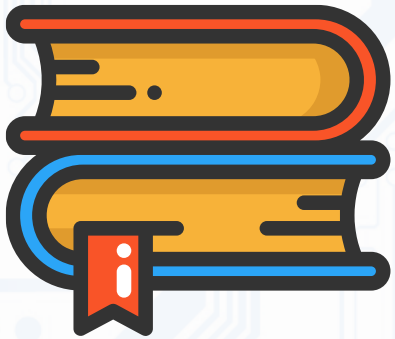
- Students who want to work on a project in image processing or robotic I can give them some research task and if they can finish we can have some project.

- For this project I can just select maximum 4 student and between this 4 , two student will be select

- Interested student inform me in private



Jiangxi University of Science and Technology



MOBILE APPLICATION DEVELOPMENT

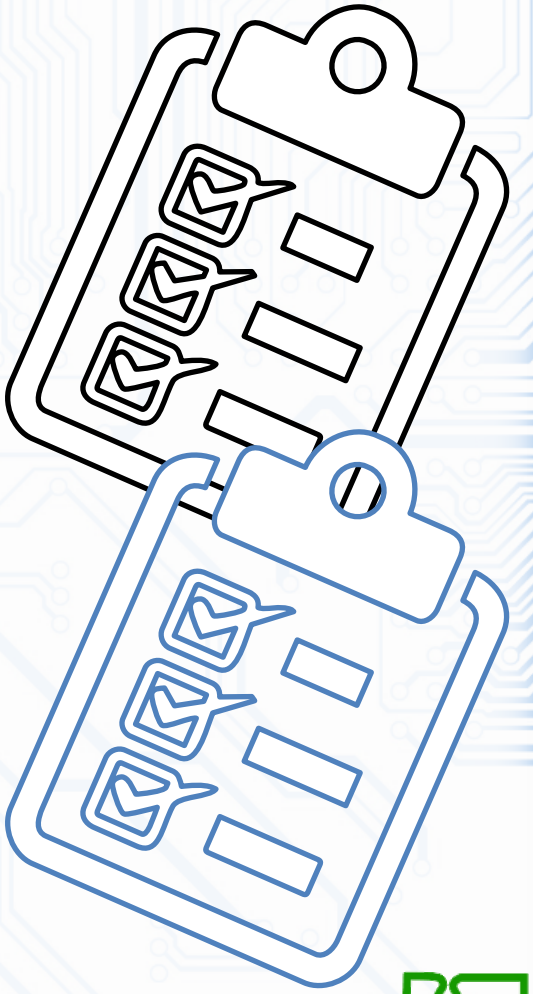
LECTURE 05: APP Inventor_environment_B

江西理工大學
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

App Inventor

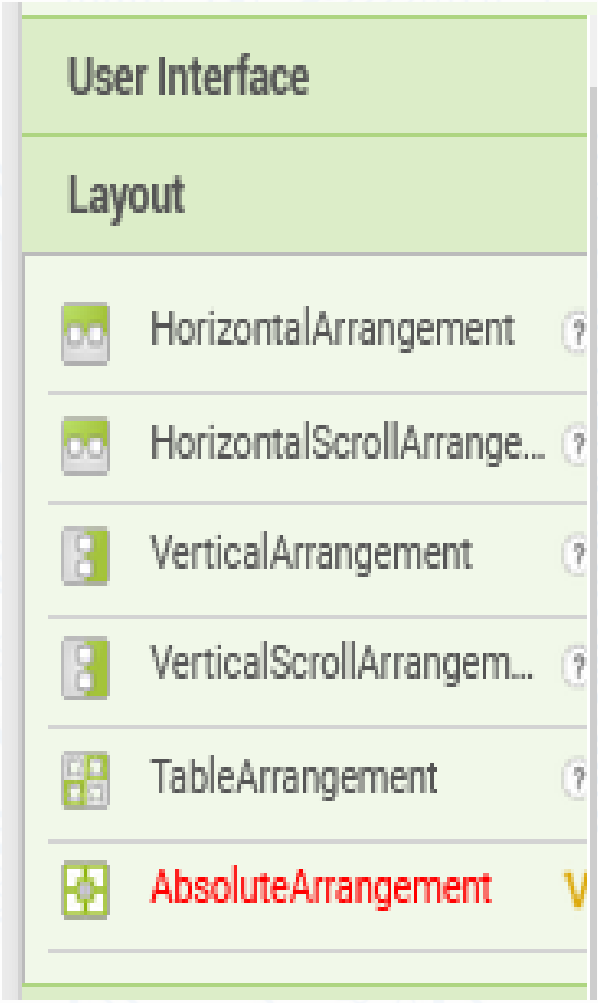
Agenda

- Layout
- Media
- Drawing and Animation



Layout

- Layout types
  - HorizontalArrangement
  - HorizontalScrollArrangement
  - VerticalArrangement
  - VerticalScrollArrangement
  - TableArrangement



HorizontalArrangement

- Use a horizontal arrangement component to display a group of components laid out from left to right.
- This component is a formatting element in which you place components that should be displayed from left to right. If you want to have components displayed one over another, use VerticalArrangement instead.
- In a HorizontalArrangement, components are arranged along the horizontal axis, vertically center-aligned.

HorizontalScrollArrangement

- A formatting element in which to place components that should be displayed from left to right.
- If you wish to have components displayed one over another, use VerticalScrollArrangement instead.
- This version is scrollable.

TableArrangement

- Use a table arrangement component to display a group of components in a tabular fashion.
- This component is a formatting element in which you place components that should be displayed in tabular form.

- In a TableArrangement, components are arranged in a grid of rows and columns, with not more than one component visible in each cell.
- If multiple components occupy the same cell, only the last one will be visible.

VerticalArrangement

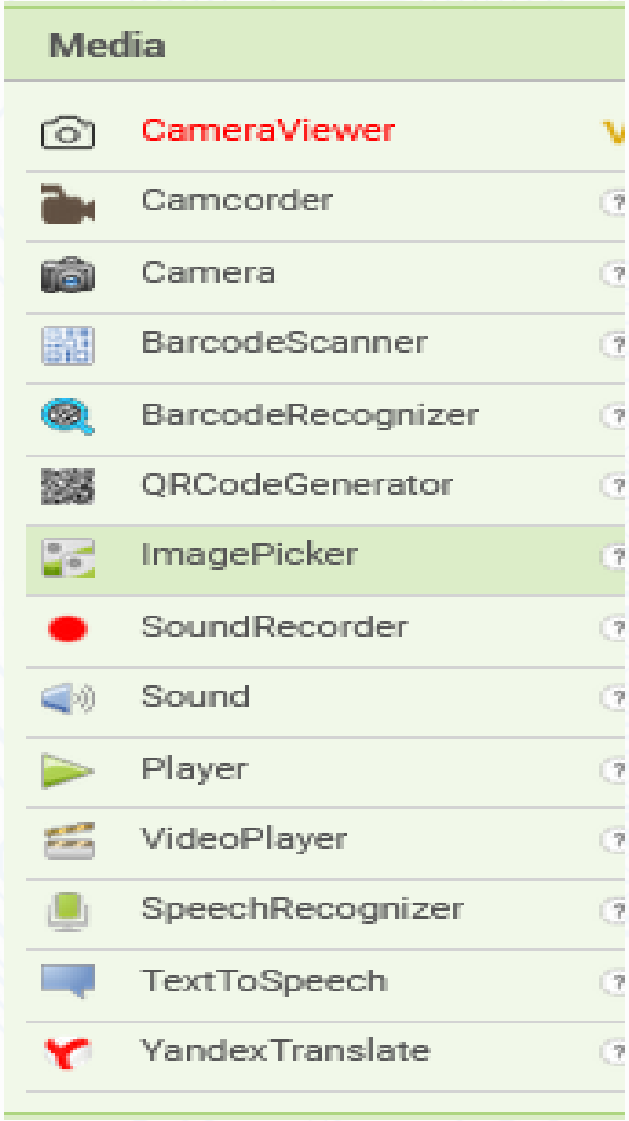
- Use a VerticalArrangement component to display a group of components laid out from top to bottom, left-aligned.
- This component is a formatting element in which you place components that should be displayed one below another.
- The first child component is stored on top, the second beneath it, and so on.
- If you want to have components displayed next to one another, use HorizontalArrangement instead.

VerticalScrollArrangement

- A formatting element in which to place components that should be displayed one below another. (The first child component is stored on top, the second beneath it, etc.) If you wish to have components displayed next to one another, use HorizontalScrollArrangement instead.
- This version is scrollable.

Media

- Camcorder
- Camera
- ImagePicker
- Player
- Sound
- SoundRecorder
- SpeechRecognizer
- TextToSpeech
- VideoPlayer
- YandexTranslate



Camcorder

- A component to record a video using the device's camcorder.
- After the video is recorded, the name of the file on the phone containing the clip is available as an argument to the After Recording event.
- The file name can be used, for example, to set the source property of a VideoPlayer component.

Events

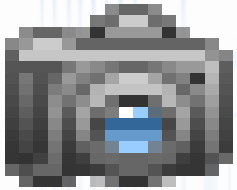
AfterRecording(*clip*)

Indicates that a video was recorded with the camera and provides the path to the stored video.

Methods

RecordVideo()

Records a video, then raises the AfterRecording event.



Camera

- Use a camera component to take a picture on the phone.
- Camera is a non-visible component that takes a picture using the device's camera.
- After the picture is taken, the path to the file on the phone containing the picture is available as an argument to the AfterPicture event.
- The path can be used, for example, as the Picture property of an Image component.

Events

AfterPicture(*image*) : Called after the picture is taken. The text argument `image` is the path that can be used to locate the image on the phone.

Methods

TakePicture(): Takes a picture, then raises the `AfterPicture` event.

ImagePicker

- A special-purpose button.
- When the user taps an ImagePicker, the device's image gallery appears, and the user can choose an image.
- After an image is picked, it is saved, and the Selection property will be the name of the file where the image is stored.
- In order to not fill up storage, a maximum of 10 images will be stored. Picking more images will delete previous images, in order from oldest to newest.

Player

- Multimedia component that plays audio and controls phone vibration.
- The name of a multimedia file is specified in the Source property, which can be set in the Designer or in the Blocks Editor.
- The length of time for a vibration is specified in the Blocks Editor in milliseconds (thousandths of a second).
- For supported audio formats, see Android Supported Media Formats.
- This component is best for long sound files, such as songs, while the Sound component is more efficient for short files, such as sound effects.

Sound

- A multimedia component that plays sound files and optionally vibrates for the number of milliseconds (thousandths of a second) specified in the Blocks Editor.
- The name of the sound file to play can be specified either in the Designer or in the Blocks Editor.
- For supported sound file formats, see Android Supported Media Formats.
- This Sound component is best for short sound files, such as sound effects, while the Player component is more efficient for longer sounds, such as songs.

SoundRecorder

Multimedia component that records audio.

Properties

SavedRecording:

- Specifies the path to the file where the recording should be stored. If this property is the empty string, then starting a recording will create a file in an appropriate location.
- If the property is not the empty string, it should specify a complete path to a file in an existing directory, including a file name with the extension .3gp.

Events

AfterSoundRecorded(*sound*):Provides the location of the newly created sound.
StartedRecording(): Indicates that the recorder has started, and can be stopped.
StoppedRecording(): Indicates that the recorder has stopped, and can be started again.

Methods

Start(): Starts recording.
Stop(): Stops recording.

SpeechRecognizer

- Use a SpeechRecognizer component to listen to the user speaking and convert the spoken sound into text using the device's speech recognition feature.

TextToSpeech

- The TextToSpeech component speaks a given text aloud. You can set the pitch and the rate of speech.
- You can also set a language by supplying a language code. This changes the pronunciation of words, not the actual language spoken.
- For example, setting the Language to French and speaking English text will sound like someone speaking English (en) with a French accent.
- You can also specify a country by supplying a Country code. This can affect the pronunciation.
- For example, British English (GBR) will sound different from US English (USA). Not every country code will affect every language.
- The languages and countries available depend on the particular device, and can be listed with the AvailableLanguages and AvailableCountries properties.

VideoPlayer

- A multimedia component capable of playing videos.
- When the application is run, the VideoPlayer will be displayed as a rectangle on-screen.
- If the user touches the rectangle, controls will appear to play/pause, skip ahead, and skip backward within the video.
- application can also control behavior by calling the Start, Pause, and SeekTo methods.
- Video files should be in 3GPP (.3gp) or MPEG-4 (.mp4) formats.
- For more details about legal formats, see Android Supported Media Formats.
- App Inventor only permits video files under 1 MB and limits the total size of an application to 5 MB, not all of which is available for media (video, audio, and sound) files.

VideoPlayer

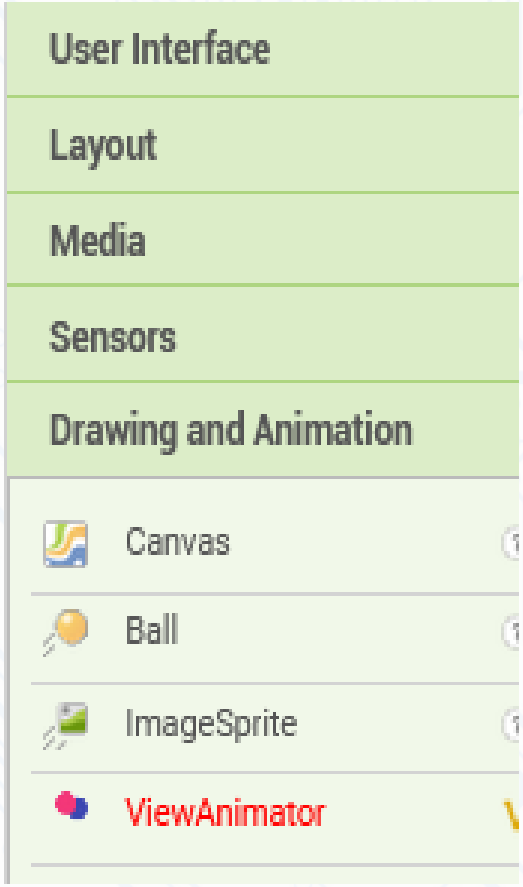
- If your media files are too large, you may get errors when packaging or installing your application, in which case you should reduce the number of media files or their sizes.
- Most video editing software, such as Windows Movie Maker and Apple iMovie, can help you decrease the size of videos by shortening them or re-encoding the video into a more compact format.
- You can also set the media source to a URL that points to a streaming video, but the URL must point to the video file itself, not to a program that plays the video.

YandexTranslate

- Use this component to translate words and sentences between different languages.
- This component needs Internet access, as it will request translations to the Yandex.Translate service.
- Specify the source and target language in the form source-target using two letter language codes. So “en-es” will translate from English to Spanish while “es-ru” will translate from Spanish to Russian. If you leave out the source language, the service will attempt to detect the source language.
- So providing just “es” will attempt to detect the source language and translate it to Spanish.
- **This component is powered by the Yandex translation service. See http://api.yandex.com/translate/ for more information, including the list of available languages and the meanings of the language codes and status codes**.
- Note: Translation happens asynchronously in the background. When the translation is complete, the GotTranslation event is triggered.

Drawing and Animation

- Ball
- Canvas
- ImageSprite



Ball

- A round ‘sprite’ that can be placed on a Canvas, where it can react to touches and drags, interact with other sprites (ImageSprites and other Balls) and the edge of the Canvas, and move according to its property values.
- For example, to have a Ball move 4 pixels toward the top of a Canvas every 500 milliseconds (half second), you would set the Speed property to 4 [pixels], the Interval property to 500 [milliseconds], the Heading property to 90 [degrees], and the Enabled property to true. These and its other properties can be changed at any time.
- **The difference between a Ball and an ImageSprite is that the latter can get its appearance from an image file, while a Ball’s appearance can only be changed by varying its PaintColor and Radius properties.**

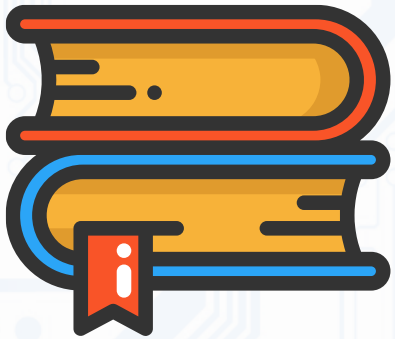
Canvas

- A two-dimensional touch-sensitive rectangular panel on which drawing can be done and sprites can be moved.
- The BackgroundColor, PaintColor, BackgroundImage, Width, and Height of the Canvas can be set in either the Designer or in the Blocks Editor.
- The Width and Height are measured in pixels and must be positive.
- Any location on the Canvas can be specified as a pair of (X, Y) values, where
  - X is the number of pixels away from the left edge of the Canvas
  - Y is the number of pixels away from the top edge of the Canvas
- There are events to tell when and where a Canvas has been touched or a Sprite (ImageSprite or Ball) has been dragged.
- There are also methods for drawing points, lines, circles, shapes, arcs, and text.

ImageSprite

- A ‘sprite’ that can be placed on a Canvas, where it can react to touches and drags, interact with other sprites (Balls and other ImageSprites) and the edge of the Canvas, and move according to its property values.
- Its appearance is that of the image specified in its Picture property (unless its Visible property is false.
- To have an ImageSprite move 10 pixels to the left every 1000 milliseconds (one second), for example, you would set the Speed property to 10 [pixels], the Interval property to 1000 [milliseconds], the Heading property to 180 [degrees], and the Enabled property to true.
- A sprite whose Rotates property is true will rotate its image as the sprite’s heading changes.
- Checking for collisions with a rotated sprite currently checks the sprite’s unrotated position so that collision checking will be inaccurate for tall narrow or short wide sprites that are rotated.
- Any of the sprite properties can be changed at any time under program control.

Jiangxi University of Science and Technology



MOBILE APPLICATION DEVELOPMENT

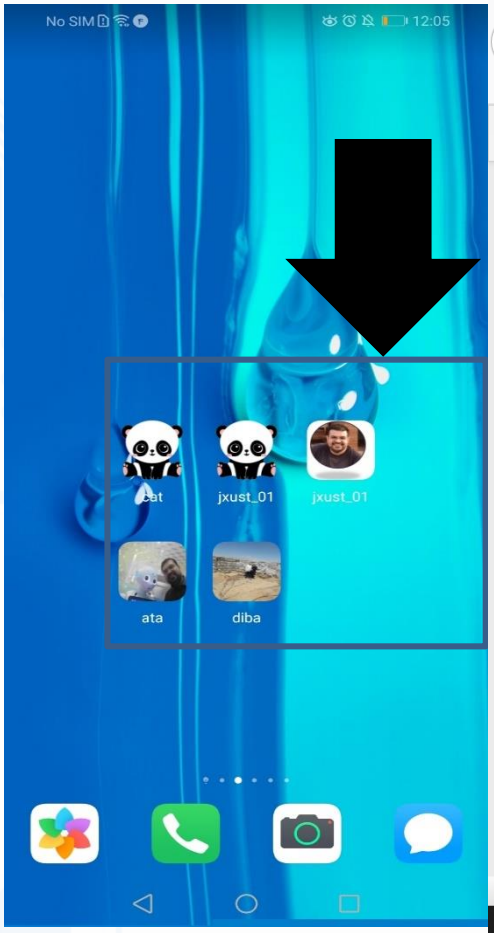
LECTURE 08: APP Inventor Practical point

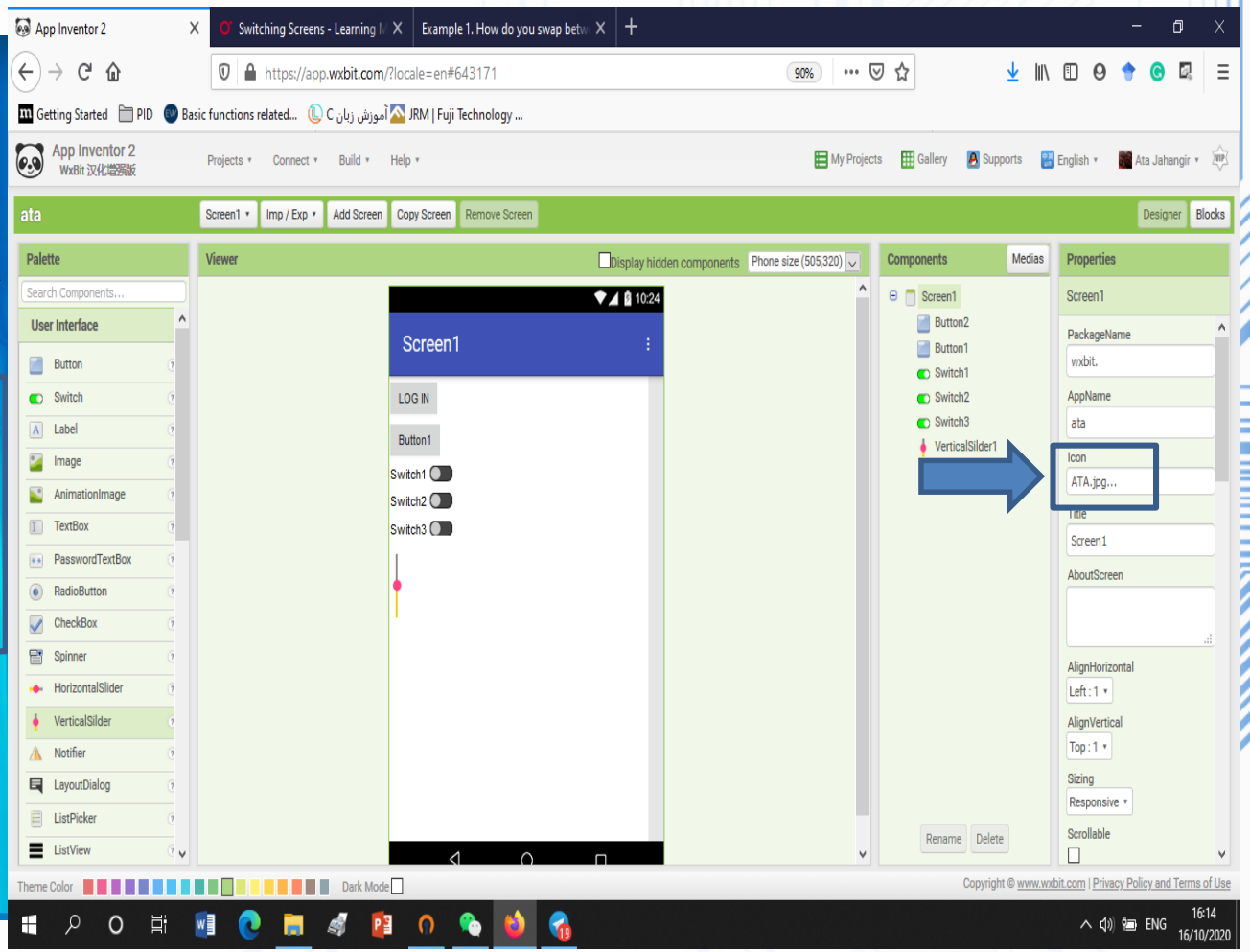
江西理工大學
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

App Inventor

Point 1:

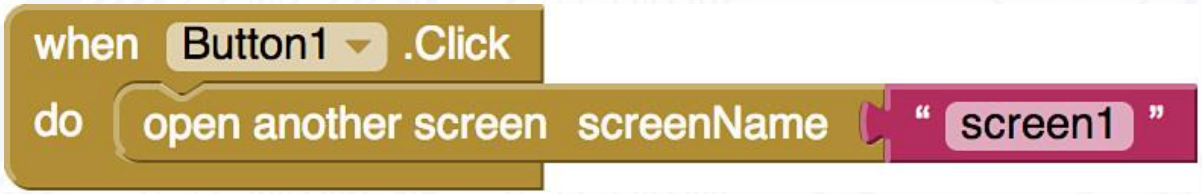
How to change the app icon





Point 2:
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.

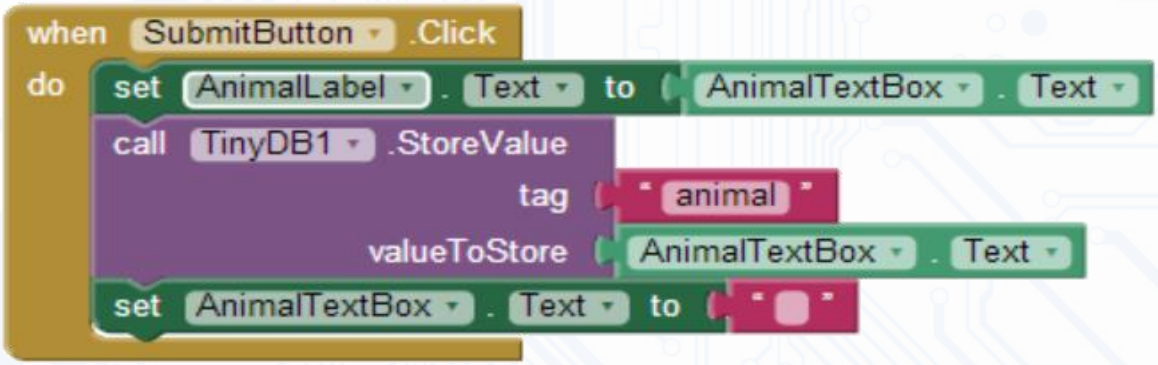


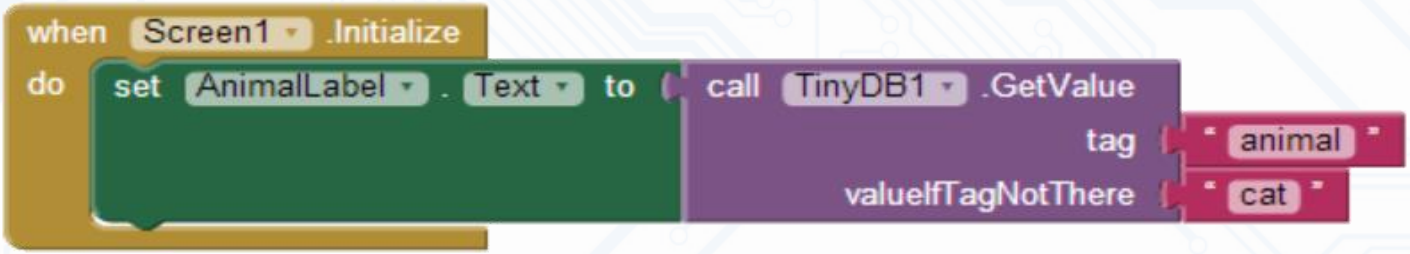
- Figure Programming an app to switch screens.
- Screen behavior is like a small deck of cards on a table.
- You start with just one card, which is Screen1, the first screen you see. When you open another screen, that screen opens on top of the previous one, like another card being put on the deck.

Point 3:

How do you share information between screens?

- The components and variables in one screen are not available to another.
- To share information between screens, use a TinyDB component.
- Though you'll have a TinyDB defined on each screen, both really point to the one and only TinyDB database.
- One or both screens might change the shared data. But in each screen, you'll need a Screen.Initialize as in the blocks shown, to read the current values in the database.





Point 4:

How to Create EXIT Button with a Notification

Point 5:

how to Use the camera function

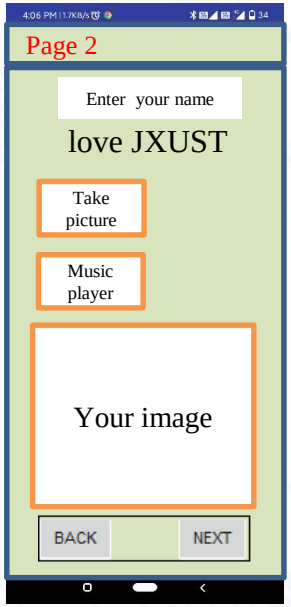
Take the picture and show it on APP

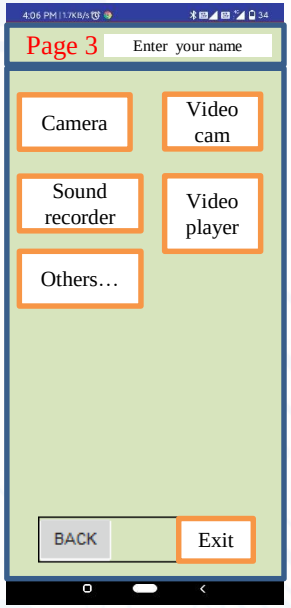
Point 6:

How to Load Screen 1 to Screen 2 on Button Click

Student Task_5

1. Make the below GUI





Use the new task template task

- First user should enter his/her name and then name should pint in all other pagers as per as drawn screen
- The music player section is like the last example
- For the section I call others please try to counting more bottom and use the media section in app inventor

Check the op of camera and video place as well as the format of file and number of clips and image that you can take

- JUST in MOOC
- **Your file should have this format of name**

MD_<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."