

1. Camera and Image Picker

1.1 Camera

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. For example, the path can be used as the *Picture* property of an *Image* component.

1.2 ImagePicker

When the user taps an image picker, the device's image gallery appears, and the user can choose an image. After an image is picked, it is saved, and the *Selected* 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.

1.3 Image Library

Free game images can be obtained from <http://www.opengameart.org>

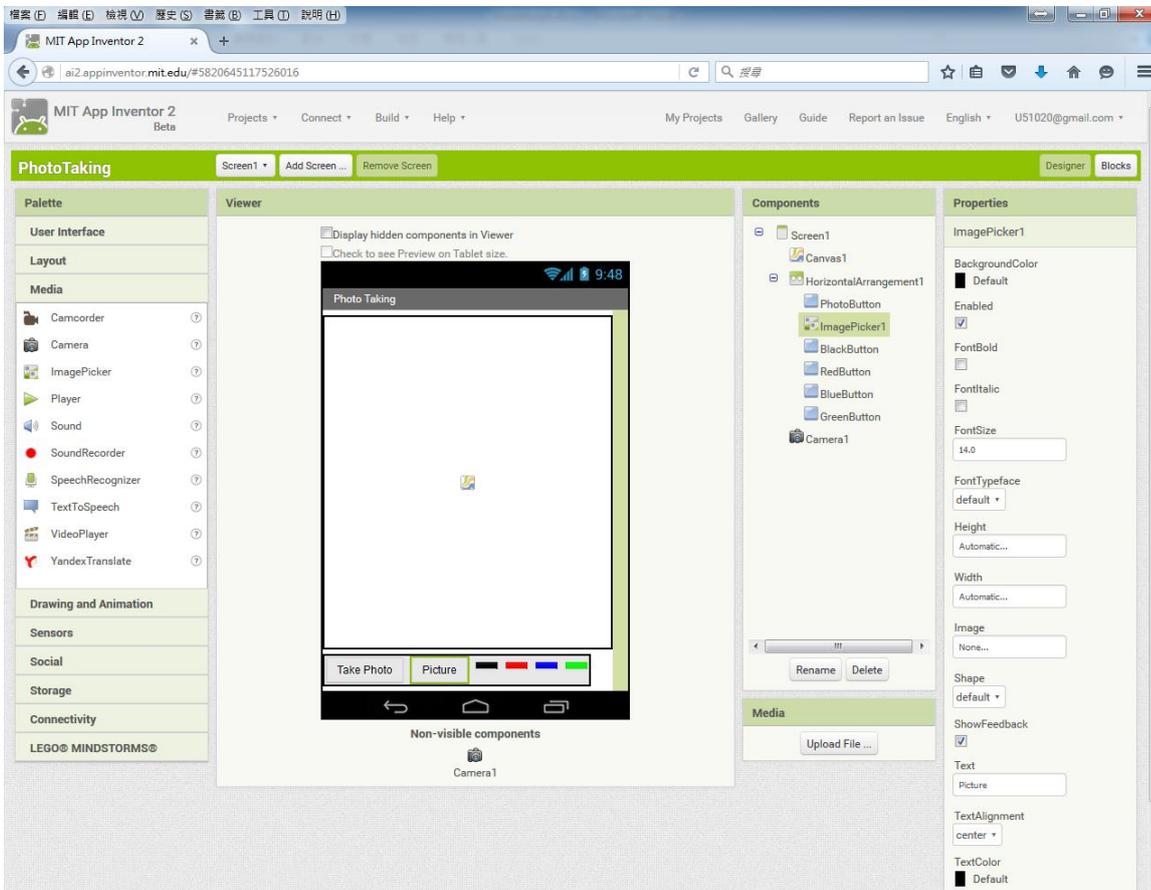
1.4 Supported Format

App Inventor supports the following image format:

- JPEG (*.jpg)
- Non-animated GIF (*.gif)
- Portable Network Graphics (*.png)
- Bitmap (*.bmp)
- WebP (.webp)

1.5 Exercise: Photo Taking

1.5.1 Designer View



1.5.2 Components

Component	Name	Properties	Remark
Screen	Screen1	Title = “Photo Taking”	
Canvas	Canvas1	Height = “Fill Parent” Weight = “Fill Parent”	
HorizontalArrangement	HorizontalArrangement1		
Button	PhotoButton	Text = “Take Photo”	Inside HorizontalArrangement1
ImagePicker	ImagePicker1	Text = “Picture”	Inside HorizontalArrangement1
Button	BlackButton	Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Button	RedButton	Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Button	BlueButton	Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Button	GreenButton	Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Camera	Camera1		

1.5.3 Block Configuration

```
when PhotoButton .Click
do call Camera1 .TakePicture

when ImagePicker1 .AfterPicking
do set Canvas1 . BackgroundImage to ImagePicker1 . Selection

when Camera1 .AfterPicture
image
do set Canvas1 . BackgroundImage to get image

when Canvas1 .Dragged
startX startY prevX prevY currentX currentY draggedAnySprite
do call Canvas1 .DrawLine
x1 get prevX
y1 get prevY
x2 get currentX
y2 get currentY

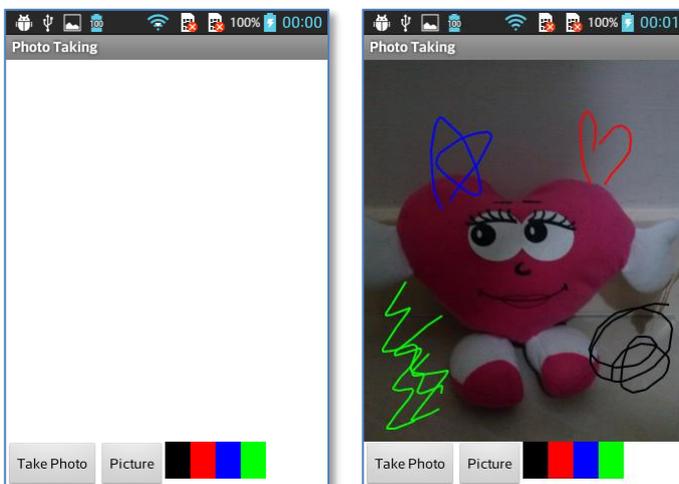
when BlackButton .Click
do set Canvas1 . PaintColor to [black]

when RedButton .Click
do set Canvas1 . PaintColor to [red]

when BlueButton .Click
do set Canvas1 . PaintColor to [blue]

when GreenButton .Click
do set Canvas1 . PaintColor to [green]
```

1.5.4 Sample Output



2. Video Player and Recording

2.1 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 `AfterRecording` event. The file name can be used, for example, to set the source property of a `VideoPlayer` component.

2.2 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. The 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 for Android 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. 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.

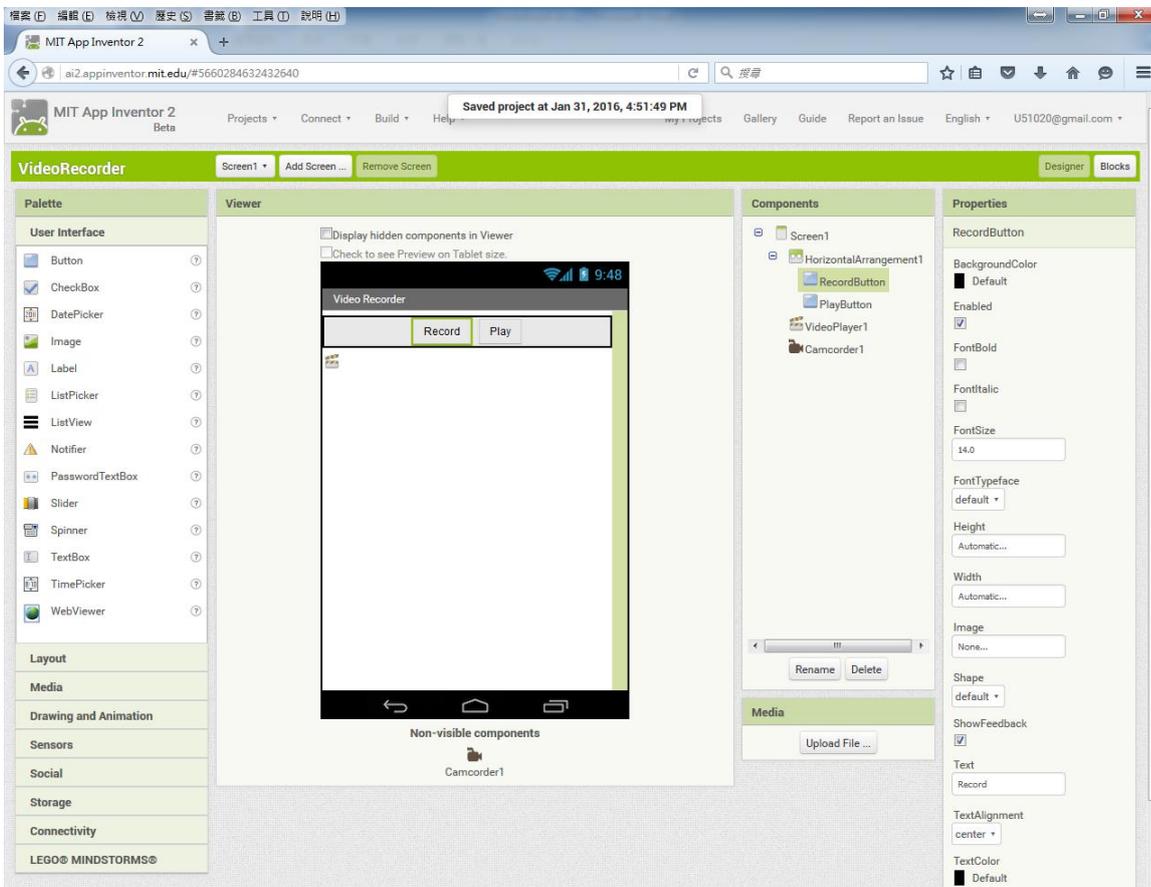
2.3 Supported Format

App Inventor supports the following video format:

- 3GPP (.3gp)
- MPEG-4 (.mp4)
- MPEG-TS (.ts, AAC audio only, not seekable, Android 3.0+)
- WebM (.webm)
- Matroska (.mkv, Android 4.0+)

2.4 Exercise: Video Recorder

2.4.1 Designer View



2.4.2 Components

Component	Name	Properties	Remark
Screen	Screen1	Title = “Video Recorder”	
HorizontalArrangement	HorizontalArrangement1	AlignHorizontal = “Center” Width = “Fill parent”	
Label	RecordButton	Text = “Record”	Inside HorizontalArrangement1
Label	PlayButton	Text = “Play”	Inside HorizontalArrangement1
Camcorder	Camcorder1		Recorded file are saved to “DCIM” by default.
VideoPlayer	VideoPlayer1	Width = “Fill parent” Height = “Fill parent”	

2.4.3 Block Configuration

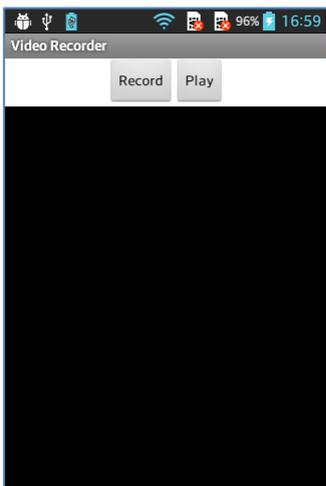
```
when RecordButton ▾ .Click
do call Camcorder1 ▾ .RecordVideo

when PlayButton ▾ .Click
do call VideoPlayer1 ▾ .Start

when Camcorder1 ▾ .AfterRecording
clip
do set VideoPlayer1 ▾ . Source ▾ to get clip ▾
```

2.4.4 Sample Output

In general, the recorded file can be found from `\\DCIM\`.



3. Player and Sound Recording

3.1 Sound

A multimedia component that plays sound files and optionally vibrates for the number of milliseconds 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.

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.

3.2 Player

Player is a multimedia component that plays audio and controls phone vibration. The name of a multimedia field 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.

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.

3.3 SoundRecorder

SoundRecorder is a Multimedia component that records audio. By specify the SavedRecording property, you can define 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.

3.4 Audio Library

Free sound file can be obtained from <https://www.freesound.org>

3.5 Supported Format

App Inventor supports the following audio format:

- Wave (.wav)
- Midi (.mid)
- MP3 (.mp3)
- MPEG-4 (.mp4, .m4a)
- 3GPP (.3gp)
- ADTS raw AAC (.aac)

3.6 Exercise: Piano

3.6.1 Media File

3.6.1.1 Image Files



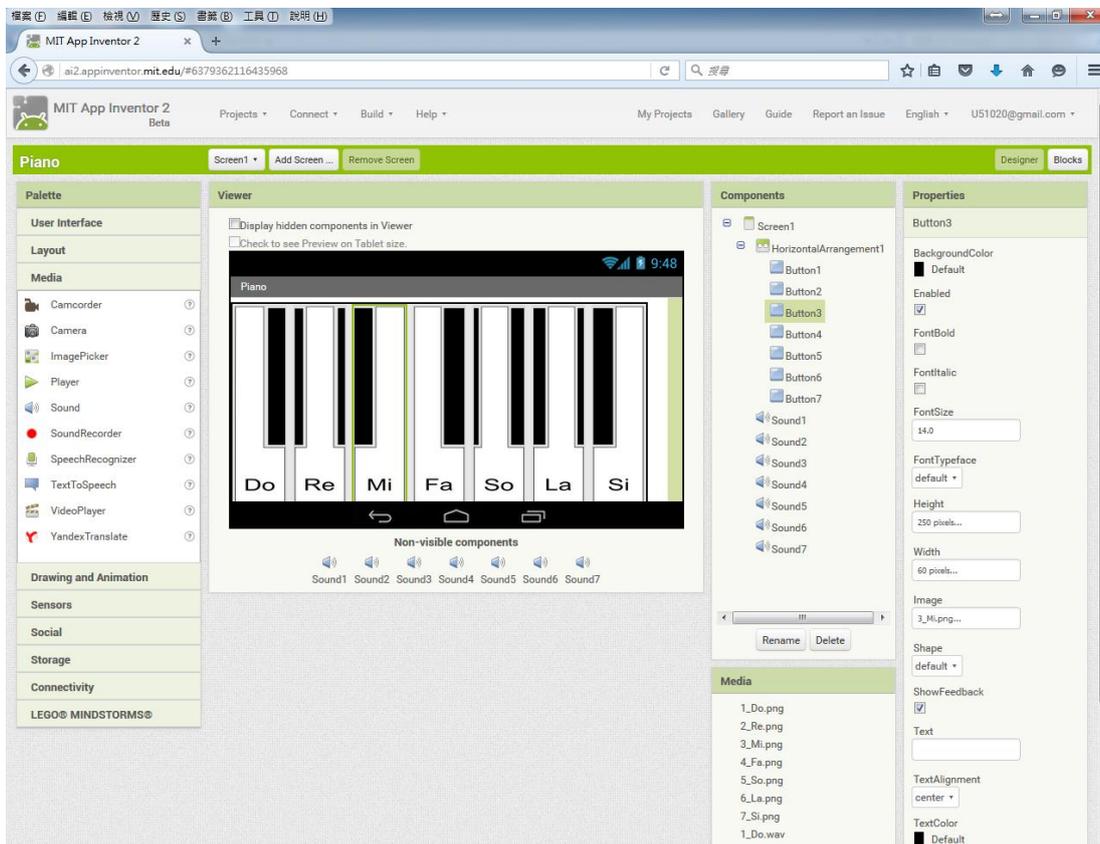
1_Do.png 2_Re.png 3_Mi.png 4_Fa.png 5_So.png 6_La.png 7_Si.png

3.6.1.2 Sound Files



1_Do.wav 2_Re.wav 3_Mi.wav 4_Fa.wav 5_So.wav 6_La.wav 7_Si.wav

3.6.2 Designer View



3.6.3 Components

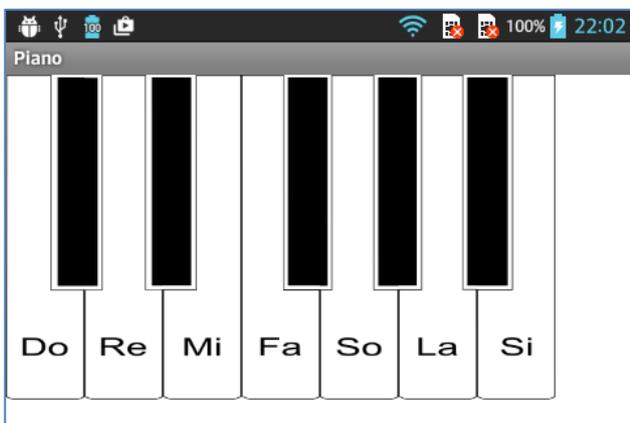
Component	Name	Properties	Remark
Screen	Screen1	Title = “Piano” ScreenOrientation = “Landscape”	
HorizontalArrangement	HorizontalArrangement1		
Button	Button1	Height = “250 pixels” Width = “60 pixels” Image = “1_Do.png” Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Button	Button2	Height = “250 pixels” Width = “60 pixels” Image = “2_Re.png” Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Button	Button3	Height = “250 pixels” Width = “60 pixels” Image = “3_Mi.png” Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Button	Button4	Height = “250 pixels” Width = “60 pixels” Image = “4_Fa.png” Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Button	Button5	Height = “250 pixels” Width = “60 pixels” Image = “5_So.png” Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Button	Button6	Height = “250 pixels” Width = “60 pixels” Image = “6_La.png” Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Button	Button7	Height = “250 pixels” Width = “60 pixels” Image = “7_Si.png” Text = <i>[Blank]</i>	Inside HorizontalArrangement1
Sound	Sound1	Source = “1_Do.wav”	
Sound	Sound2	Source = “2_Re.wav”	
Sound	Sound3	Source = “3_Mi.wav”	
Sound	Sound4	Source = “4_Fa.wav”	
Sound	Sound5	Source = “5_So.wav”	

Sound	Sound6	Source = "6_La.wav"	
Sound	Sound7	Source = "7_Si.wav"	

3.6.4 Block Configuration

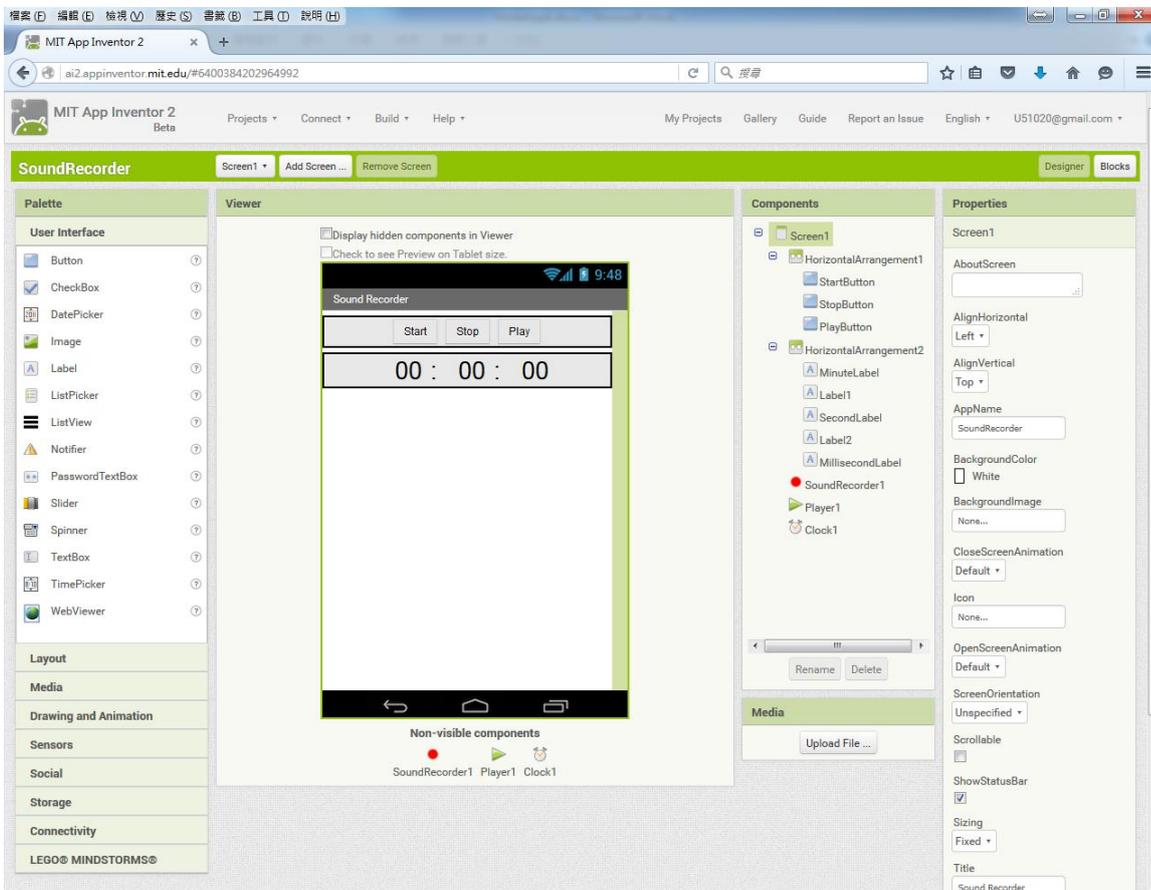


3.6.5 Sample Output



3.7 Exercise: Sound Recorder

3.7.1 Designer View

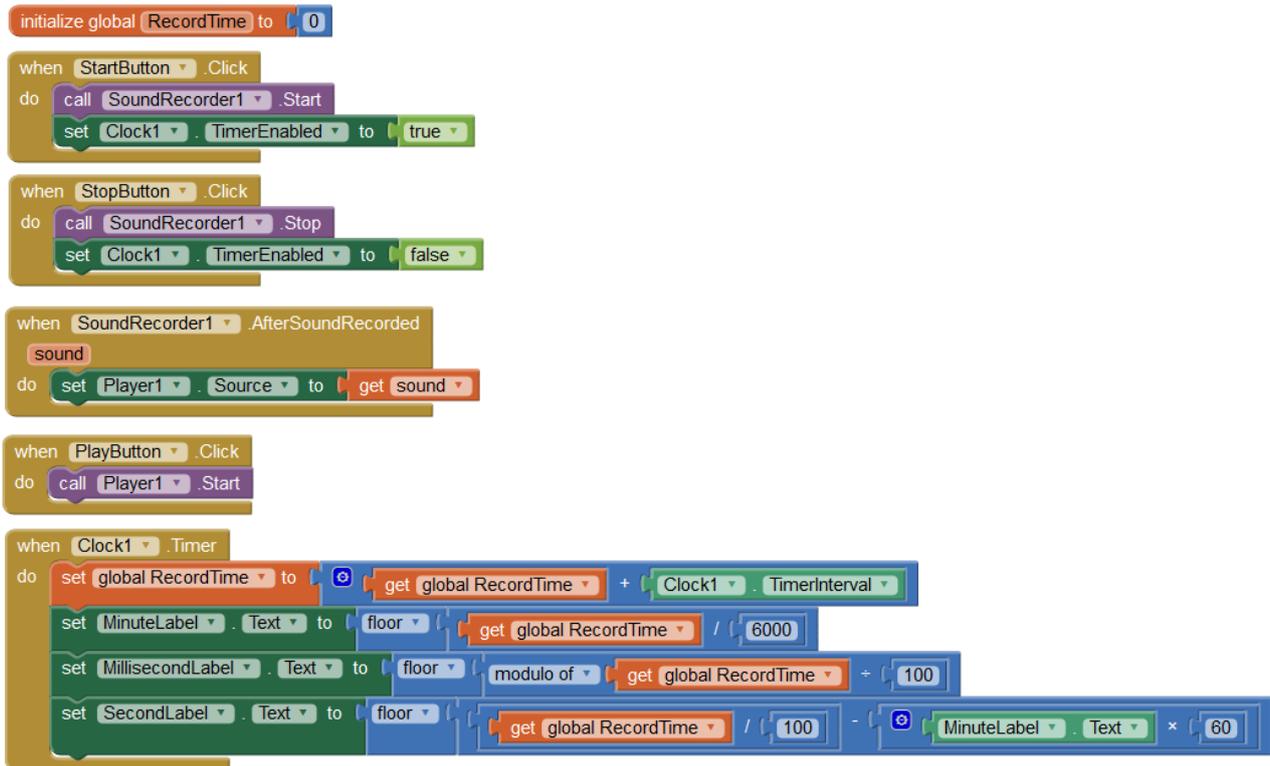


3.7.2 Components

Component	Name	Properties	Remark
Screen	Screen1	Title = “Sound Recorder”	
HorizontalArrangement	HorizontalArrangement1	AlignHorizontal = “Center” Width = “Fill parent”	
Label	StartButton	Text = “Start”	Inside HorizontalArrangement1
Label	StopButton	Text = “Stop”	Inside HorizontalArrangement1
Label	PlayButton	Text = “Play”	Inside HorizontalArrangement1
HorizontalArrangement	HorizontalArrangement2	AlignHorizontal = “Center” Width = “Fill parent”	

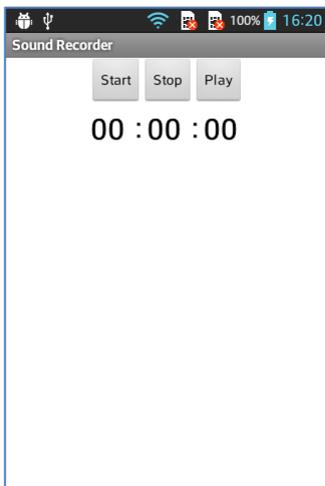
Label	MinuteLabel	FontSize = “ 30 ” Width = “ 40 pixels ” Text = “ 00 ” TextAlighment = “ right ”	Inside HorizontalArrangement2
Label	Label1	FontSize = “ 30 ” Width = “ 40 pixels ” Text = “ : ”	Inside HorizontalArrangement2
Label	SecondLabel	FontSize = “ 30 ” Width = “ 40 pixels ” Text = “ 00 ” TextAlighment = “ right ”	Inside HorizontalArrangement2
Label	Label2	FontSize = “ 30 ” Width = “ 40 pixels ” Text = “ : ”	Inside HorizontalArrangement2
Label	MillisecondLabel	FontSize = “ 30 ” Width = “ 40 pixels ” Text = “ 00 ” TextAlighment = “ right ”	Inside HorizontalArrangement2
SoundRecorder	SoundRecorder1		Recorded file are saved to “My Documents\ Recordings” by default.
Player	Player1		
Clock	Clock1	TimerAlwaysFires = <i>[Blank]</i> TimeEnabled = <i>[Blank]</i> TimeInterval = “ 10 ”	

3.7.3 Block Configuration



3.7.4 Sample Output

In general, the recorded file can be found from `\\My Documents\Recordings\`.



4. Speech Recognition and Text-to-Speech

4.1 Speech Recognition

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

4.2 Yandex Translator

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.

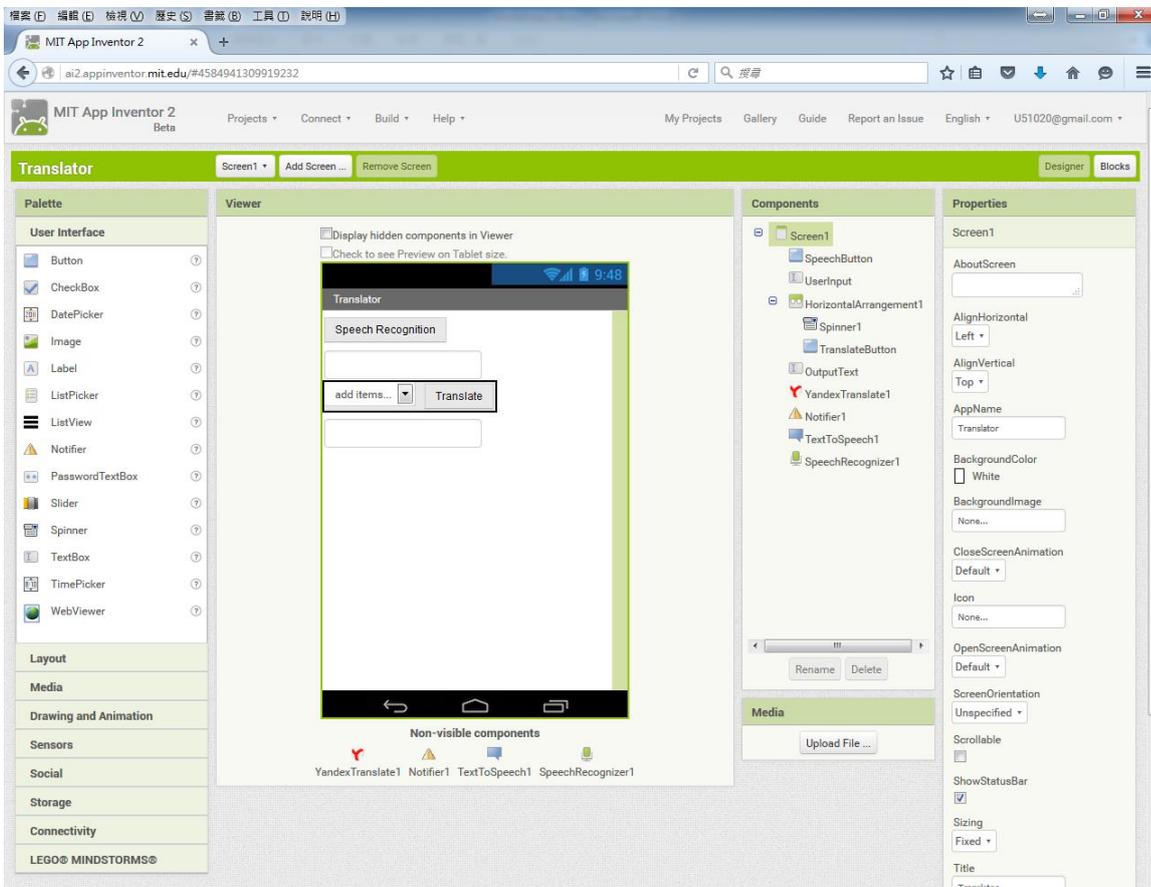
4.3 Text to Speech

Use a text-to-speech component to have the device speak text audibly. In order for this component to work, the device must have the TTS Extended Service app by Eyes-Free Project installed. You can download this from <http://code.google.com/p/eyes-free/downloads/list>.

The text-to-speech component has properties you can set to guide the pronunciation of the text to be spoken. 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.

4.4 Exercise: Translator

4.4.1 Designer View



4.4.2 Components

Component	Name	Properties	Remark
Screen	Screen1	Title = “ Translator ”	
Button	SpeechButton	Text = “ Speech Recognition ”	
TextBox	UserInput	Hint = “ Your Text ”	
HorizontalArrangement	HorizontalArrangement1		
Spinner	Spinner1		Inside HorizontalArrangement1
Button	TranslateButton	Text = “ Translate ”	Inside HorizontalArrangement1
TextBox	OutputText	Hint = “”	
YandexTranslate	YandexTranslate1		
Notifier	Notifier1		
TextToSpeech	TextToSpeech1		
SpeechRecognizer1	SpeechRecognizer1		

4.4.3 Block Configuration

```
initialize global LanguageList to make a list ["en", "es", "fr", "zh", "ja"]

when Screen1.Initialize
do set Spinner1.Elements to get global LanguageList

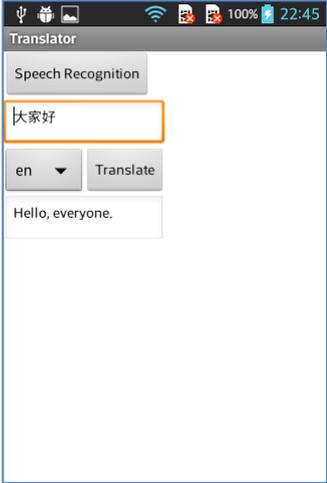
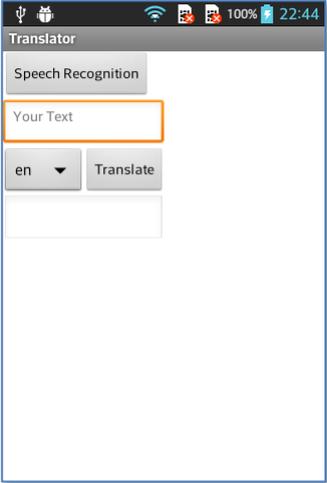
when SpeechButton.Click
do call SpeechRecognizer1.GetText

when SpeechRecognizer1.AfterGettingText
result
do set UserInput.Text to get result

when TranslateButton.Click
do call YandexTranslate1.RequestTranslation
   languageToTranslateTo Spinner1.Selection
   textToTranslate UserInput.Text
   call Notifier1.ShowProgressDialog
     message "Translating ..."
     title "Please Wait"

when YandexTranslate1.GotTranslation
responseCode translation
do call Notifier1.DismissProgressDialog
   set OutputText.Text to get translation
   set TextToSpeech1.Language to Spinner1.Selection
   call TextToSpeech1.Speak
     message OutputText.Text
```

4.4.4 Sample Output



5. Accessing Images and Sounds

Applications built with App Inventor can access sound, image, and video sources from three different kinds of locations:

5.1 Application Assets

The sources labeled Media shown in the designer — part of the application's assets — are packaged with the application. Anyone who installs your application will have them, as part of the application. You also specify them in the designer, which is convenient. You can also specify these in programs by their file name: just use the file name without any special prefix. For example, if you have an image asset named `kitty.png`, you can use it as an image: just set the Picture property of an image component to the text `kitty.png`. You can similarly use files names for sound (Sound or Player) or video (VideoPlayer).

Assets are the most convenient to use, but the space for them is limited to a few megabytes, because they must be packaged with the application. They are good for small images and short audio selections. But you would probably not use them for complete songs or videos.

5.2 The Phone SD card

You can access files on your phone's SD (secure digital) card using file names that begin with `/sdcard`. You could play a song on your SDCard by setting the source of a Player component to `/sdcard/Music/Blondie/The Best of Blondie/Heart of Glass.mp3` and starting the Player (assuming of course, that the song file is on the SDCard). Make sure to specify the complete file name, including the "mp3".

The Android system also includes an alternative way to designate SDCard files as URLs. Here you prefix the file name with `file:///sdcard` and use "URL encoding" for special characters. For example, a space is "%20". So you could designate the same file by setting the player source to `file:///sdcard/Music/Blondie/The%20Best%20of%20Blondie/Heart%20of%20Glass.mp3`

Note that you'll want to use a Player component for this, not Sound. A complete song like this is too large for Sound to handle.

5.2.1 Images and videos can be designated similarly.

App Inventor doesn't (yet) include any way to store files on the SD card. It also doesn't (yet) include a way to list the files on the SDCard. You'll have to use other applications or the Android phone file manager for that.

Using the SD Card provides a lot more space for media than trying to package things as assets. The drawback is that users won't automatically get them by installing your application.

5.3 URLs and the Web

You can access files on Web using URLs, starting with `http://` , for example, setting the picture property of an image to http://www.google.com/images/srpr/nav_logo14.png and similarly for music and videos. Make sure you use the link that points to the actual file, not to players for the files, which is much more common on the Web, especially for music and videos.

5.3.1 Other content URLs

The Android system also uses URLs to access various places that media is stored on the phone. For example, the images in the photo gallery can be accessed with file names beginning `content://media/external/images/media` , as you can see by using the ImagePicker and examining the resulting image path.

6. Interactive Multimedia Game

6.1 Exercise: Mole Mash

6.1.1 Media File

6.1.1.1 Image Files



mole.png



hole.png

6.1.1.2 Sound Files

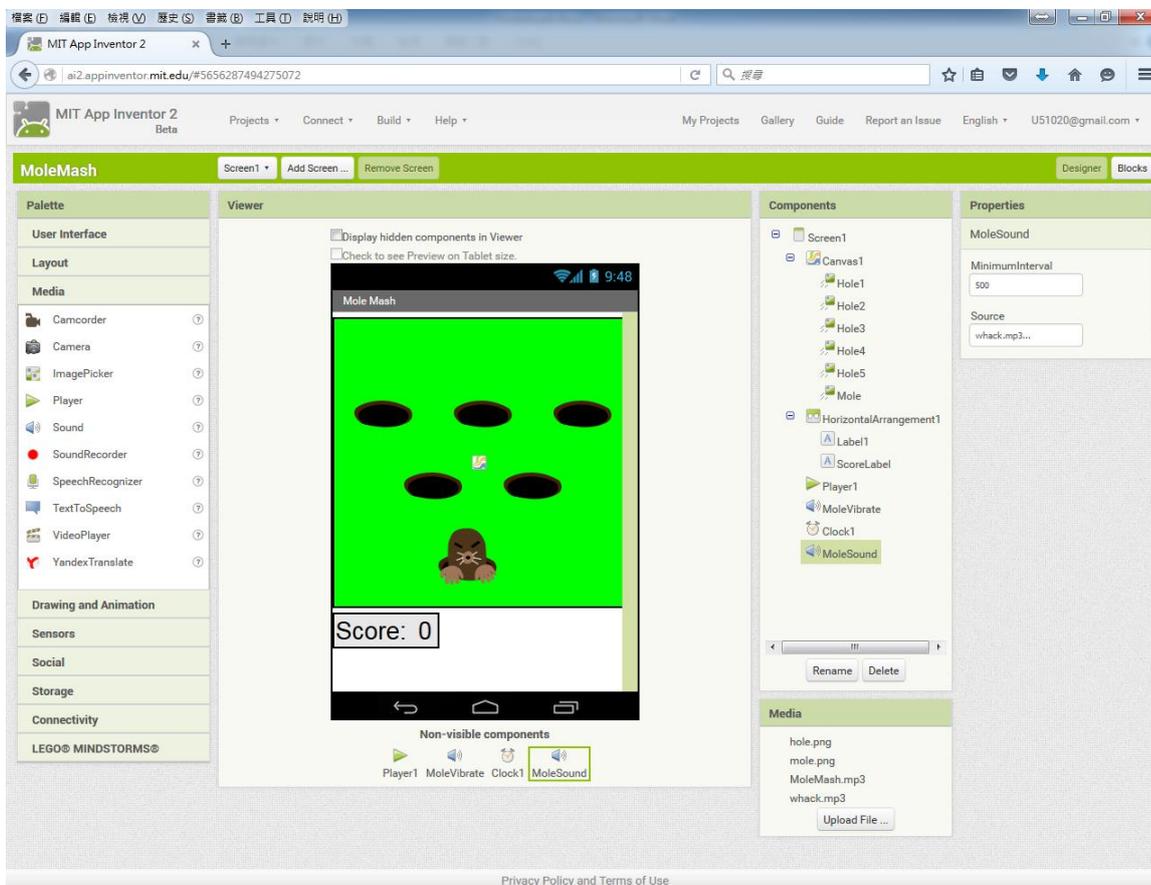


MoleMash.mp3



whack.mp3

6.1.2 Designer View

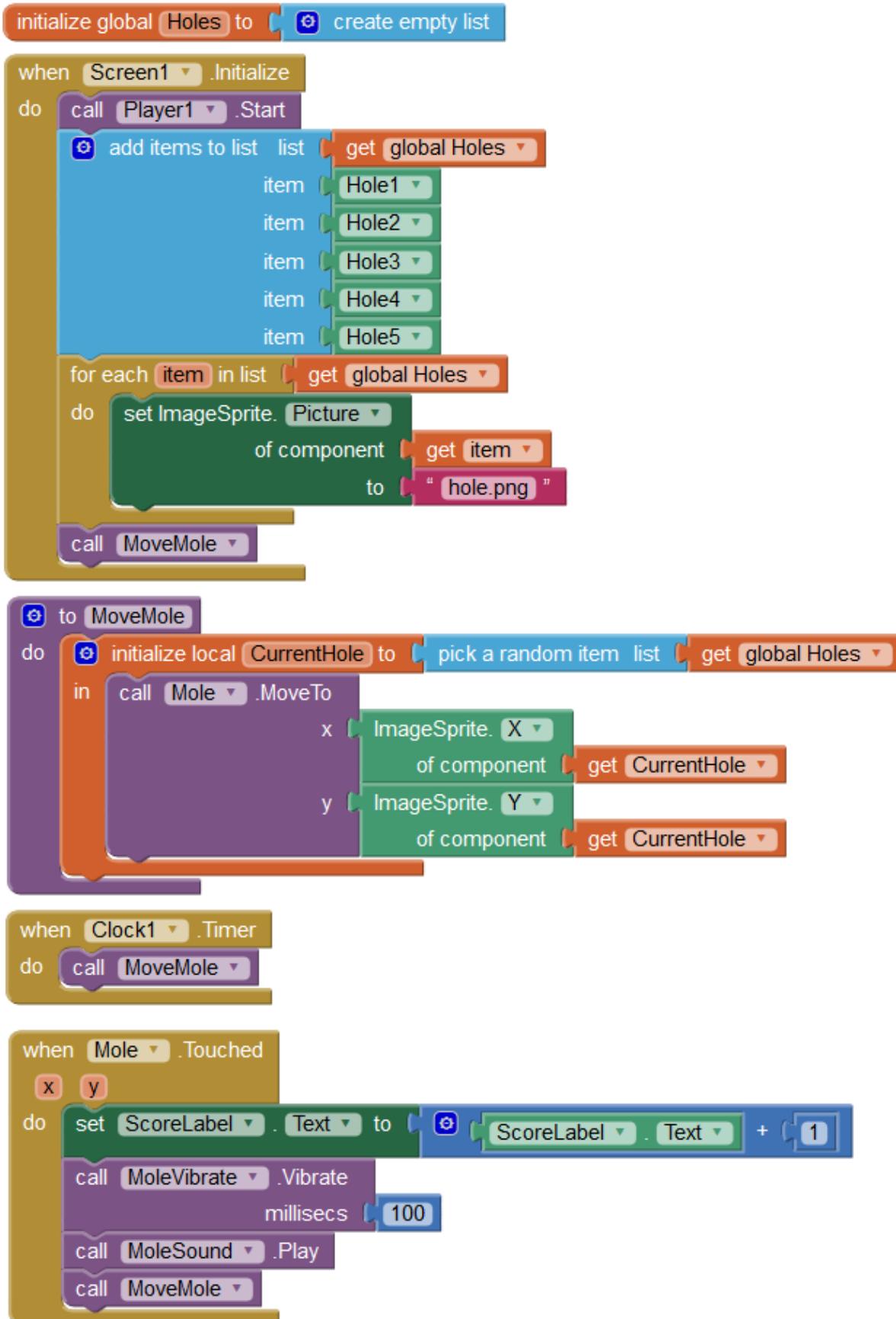


The screenshot shows the MIT App Inventor 2 web interface in Designer View. The main workspace displays a mobile app preview for 'Mole Mash' with a green background, five black holes, a mole character, and a 'Score: 0' label. The left sidebar contains a 'Media' section with various assets. The right sidebar shows the 'Components' list and the 'Properties' panel for the selected 'MoleSound' component, which has a 'Source' property set to 'whack.mp3'.

6.1.3 Components

Component	Name	Properties	Remark
Screen	Screen1	Title = “Mole Mash”	
Canvas	Canvas1	BackgroundColor = “Green” Height = “320 pixels” Width = “320 pixels”	
ImageSprite	Hole1	Picture = “hole.png” X = “20” Y = “60”	
ImageSprite	Hole2	Picture = “hole.png” X = “130” Y = “60”	
ImageSprite	Hole3	Picture = “hole.png” X = “240” Y = “60”	
ImageSprite	Hole4	Picture = “hole.png” X = “75” Y = “140”	
ImageSprite	Hole5	Picture = “hole.png” X = “185” Y = “140”	
ImageSprite	Mole	Picture = “mole.png”	
HorizontalArrangement	HorizontalArrangement1		
Label	Label1	FontSize = “30” Text = “ Score:”	Inside HorizontalArrangement1
Label	ScoreLabel	FontSize = “30” Text = “0”	Inside HorizontalArrangement1
Player	Player1	Loop = “X” PlayOnlyInForeground = “X” Source = “MoleMash.mp3”	
Sound	MoleVibrate		
Sound	MoleSound	Source = “whack.mp3”	
Clock	Clock1		

6.1.4 Block Configuration



6.1.5 Sample Output

