



MIT APP INVENTOR

4

di FRANCESCO FICILI

Proseguiamo la trattazione di MIT App Inventor, descrivendo e sperimentando due nuove e interessanti famiglie di componenti dell'ambiente di sviluppo: Media e Social.

Nella puntata precedente ci siamo lasciati dopo avere iniziato a descrivere in dettaglio alcune famiglie di componenti di App Inventor ed in particolare ci siamo concentrati sulla descrizione delle famiglie **Storage** (la famiglia contiene componenti per la gestione di file e dei data-base) e **Sensor** (all'interno di questo gruppo si trovano componenti per la gestione dei sensori tipicamente integrati in uno smartphone, sia fisici come accelerometri, giroscopi, sensori di prossimità e GPS, sia virtuali, come ad esempio il pedometro). In questa puntata continuiamo l'analisi dei componenti di MIT App Inventor, passando in rassegna altre due famiglie di componenti: le famiglie di

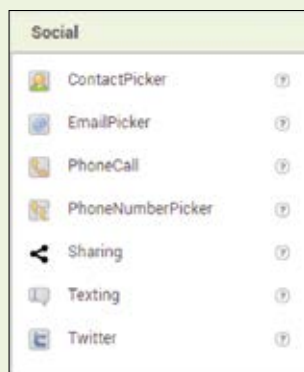


Fig. 1
Famiglia di componenti Social.

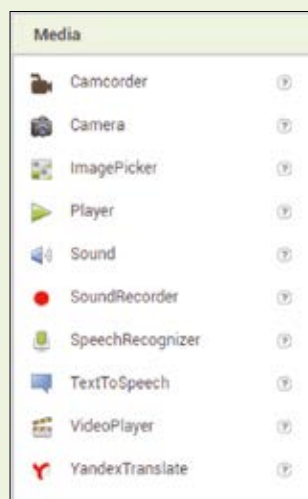


Fig. 2
Famiglia di componenti Media.

Tabella 1 - Descrizione dei componenti della famiglia Social.

COMPONENTE	DESCRIZIONE
ContactPicker	Picker che permette di scegliere un contatto tra la lista di contatti memorizzata sullo smartphone. Una volta selezionato il contatto possono venire fornite le seguenti informazioni relative: - ContactName, - EmailAddress, - EmailAddressList, - ContactUri, - PhoneNumber, - PhoneNumberList, - Picture
EmailPicker	Picker che si presenta come una textbox. Se l'utente inserisce il nome parziale di un contatto o di una email, il picker presenterà un dropdown menù di selezioni che completano la entry parziale.
PhoneCall	Componente non visibile che permette di effettuare telefonate.
PhoneNumberPicker	Componente che funziona in maniera simile al ContactPicker, ma restituisce un set inferiore di informazioni correlate: - ContactName, - PhoneNumber, - EmailAddress, - Picture
Sharing	Componente non visibile che permette di condividere files e/o messaggi tra l'app in esecuzione ed altre app installate sul dispositivo. Il componente visualizzerà un elenco di app che possono gestire le informazioni che si intende condividere e permetterà all'utente di selezionarne una (ad es. una app di email, un app di un social network, una app di messaggistica istantanea, etc..).
Texting	Componente non visibile che permette di inviare e ricevere messaggi SMS.
Twitter	Componente che permette di interfacciarsi con twitter. Il componente permette di eseguire le seguenti operazioni: - 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)

componenti **Media** e **Social**.

Prima di iniziare la trattazione di queste famiglie ricordiamo che in App Inventor si chiamano "componenti" quegli elementi che permettono di aggiungere funzionalità all'app che stiamo sviluppando; i componenti sono dunque lo strumento che ci permette sostanzialmente di creare l'interfaccia grafica della nostra app, di accedere alle varie funzionalità specifiche del nostro dispositivo mobile, sia esso uno smartphone o un tablet, nonché di utilizzare i vari sensori che si trovano al suo interno e che compongono la dotazione hardware.

I componenti Social

Iniziamo subito precisando che in MIT App Inventor vengono classificati come Social tutti quei componenti che permettono di interagire con i meccanismi

social integrati nello smartphone. Questi possono essere meccanismi classici, come ad esempio le chiamate telefoniche e gli SMS, oppure più recenti e strettamente legati alla Rete, come i social network (Twitter, Whatsapp e via di seguito).

Non ci soffermeremo molto sulla descrizione di questi componenti in quanto non li utilizzeremo nell'esempio pratico presentato in questa puntata; ci limiteremo, piuttosto, a fornire una tabella riassuntiva che ne illustra le caratteristiche generali.

I componenti Media

In MIT App Inventor sono identificati come componenti Media tutti quei componenti che sono in grado di gestire applicazioni multimediali, ossia quelle che consentono di effettuare la registrazione e la riproduzione di audio e video.

Tabella 2 - Descrizione dei componenti della famiglia Media.

COMPONENTE	DESCRIZIONE
Camcorder	Componente che permette di registrare un video utilizzando il camcorder del dispositivo.
Camera	Componente che permette di scattare una foto utilizzando la camera del dispositivo.
ImagePicker	Picker che permette di selezionare un'immagine della galleria di immagini del dispositivo.
Player	Componente multimediale che permette di riprodurre una traccia audio e controllare la vibrazione del dispositivo.
Sound	Componente multimediale che permette di riprodurre un suono e controllare la vibrazione del dispositivo.
SoundRecorder	Componente multimediale che permette di registrare audio usando il mic del dispositivo.
SpeechRecognizer	Componente che permette di convertire un suono acquisito tramite il microfono in un testo.
TextToSpeech	Componente che, dato un testo in ingresso, permette di sintetizzarlo e riprodurlo tramite lo speaker del dispositivo.
VideoPlayer	Componente multimediale che permette di riprodurre video sul display del dispositivo.
YandexTranslate	Componente che permette di effettuare traduzioni tra linguaggi differenti sfruttando il servizio online Yandex.Translate. Necessita di una connessione internet per funzionare.

Tabella 3 - Eventi, metodi e proprietà del componente SpeechRecognizer

PROPRIETÀ	
SINTASSI	DESCRIZIONE
Result	L'ultimo testo generato dalla conversione.
EVENTI	
SINTASSI	DESCRIZIONE
AfterGetting(Text result)	Evento triggerato una volta che il convertitore ha effettuato la conversione in testo. L'argomento result contiene il testo risultato della conversione.
BeforeGettingText()	Evento triggerato appena prima che il convertitore venga invocato.
METODI	
SINTASSI	DESCRIZIONE
GetText()	Questo metodo chiede all'utente di parlare e converte il parlato in testo. L'evento AfterGetting(Text) viene invocato subito dopo che la conversione viene ultimata.

Tali componenti interagiscono con le interfacce audio video del dispositivo mobile, quindi con il microfono e lo speaker (altoparlante) nel caso dell'audio e con la fotocamera nel caso del video. Inoltre alcuni componenti di questa famiglia sono in grado di interagire con la galleria di immagini memorizzate nella memoria di massa del nostro

smartphone o tablet.

In Fig. 2 è riportata un'immagine raffigurante la famiglia di componenti Media nell'IDE di App Inventor. In Tabella 2 è fornita una descrizione sommaria dei vari componenti della famiglia, mentre nei paragrafi successivi forniremo una descrizione più dettagliata, corredata della solita descrizione di

Tabella 4 - Eventi, metodi e proprietà del componente TextToSpeech.

PROPRIETÀ	
SINTASSI	DESCRIZIONE
AvailableCountries	Elenca tutti i country code disponibili sul dispositivo per essere utilizzati con il componente.
AvailableLanguages	Elenca tutti i language code disponibili sul dispositivo per essere utilizzati con il componente.
Country	Country code da utilizzare con il componente TextToSpeech. Questa proprietà incide sulla pronuncia, modificando l'accento. Ad esempio un accento con country code US (Stati Uniti) sarà diverso da un accento con country code UK (Gran Bretagna).
Language	Imposta il linguaggio da utilizzare quando viene sintetizzata la voce.
Pitch	Imposta il Pitch. Il range di valori è compreso tra 0 e 2.
SpeechRate	Imposta lo SpeechRate, ossia la velocità con cui viene pronunciata la parola o la frase. Il range di valori è compreso tra 0 e 2.
EVENTI	
SINTASSI	DESCRIZIONE
AfterSpeaking(boolean result)	Evento triggerato subito dopo la riproduzione del testo.
BeforeSpeaking()	Evento triggerato subito dopo l'invocazione del metodo Speak e subito prima della riproduzione del testo.
METODI	
SINTASSI	DESCRIZIONE
Speak(text message)	Metodo che avvia la sintesi del testo passato come parametro.

Tabella 5 - Eventi, metodi e proprietà del componente YandexTranslate.

PROPRIETÀ	
SINTASSI	
Nessuna	
EVENTI	
SINTASSI	DESCRIZIONE
GotTranslation(text responseCode, text translation)	Evento triggerato quando il servizio Yandex.Translate restituisce il testo tradotto. Insieme alla traduzione viene fornito un codice di risposta, se il codice è diverso da 200 qualcosa non ha funzionato correttamente e la traduzione non è stata effettuata.
METODI	
SINTASSI	DESCRIZIONE
RequestTranslation(text languageToTranslateTo, text textToTranslate)	Metodo che permette di effettuare una richiesta di traduzione al servizio online, fornendo il target language per la traduzione e il testo da tradurre. Il linguaggio target viene fornito con language code (ad es. it = italiano, en = inglese, es = spagnolo ...). Una volta completata la traduzione, il componente triggera un evento che fornisce il risultato. Yandex cerca di determinare il linguaggio di partenza direttamente dal testo, ma è anche possibile dichiararlo esplicitamente, aggiungendolo prima del linguaggio target e separandolo con uno spazio (ad es. it-en richiederà una traduzione da italiano ad inglese).

eventi, metodi e proprietà dei componenti SpeechRecognizer, TextToSpeech e YandexTranslate.

Speech Recognizer

Questo componente sfrutta l'input proveniente dal microfono integrato nello smartphone per acquisire il segnale audio e poi sfrutta le feature di speech recognizing integrate in Android per convertire il parlato in testo. In **Tabella 3** sono riportati eventi, metodi e proprietà.

Text To Speech

Il componente TextToSpeech permette di sintetizzare vocalmente un testo. È possibile anche impostare alcune proprietà, come ad esempio language e country per cambiare l'accento con cui viene sintetizzata la voce. In **Tabella 4** sono riportati, come di consueto eventi, metodi e proprietà.

Yandex translate

Questo componente permette di tradurre parole o frasi tra lingue differenti, utilizzando il servizio online di traduzione Yandex Translate. Il compo-

nente lavora in maniera del tutto asincrona, inviando una richiesta al servizio online tramite un metodo e fornendo il risultato tramite un apposito evento. Informazioni aggiuntive, come la lista dei linguaggi supportati, significato dei language code e altro, possono essere recuperate all'indirizzo web:

<http://api.yandex.com/translate/>

In **Tabella 5** sono riportati eventi, metodi e proprietà del componente.

Esempio pratico: traduttore vocale simultaneo

Sfruttando i tre componenti appena descritti nel dettaglio creiamo un semplice esempio pratico che implementa un traduttore vocale simultaneo, ossia una app in grado di acquisire un testo (parola singola o frase) da tradurre, effettuare la traduzione e riprodurla. In questa semplice implementazione faremo sì che il linguaggio di partenza sia sempre l'italiano, mentre il target language sarà selezionabile tra una serie di linguaggi a scelta da un elenco

Fig. 3 - Sistemazione degli elementi del layout dell'app di esempio.

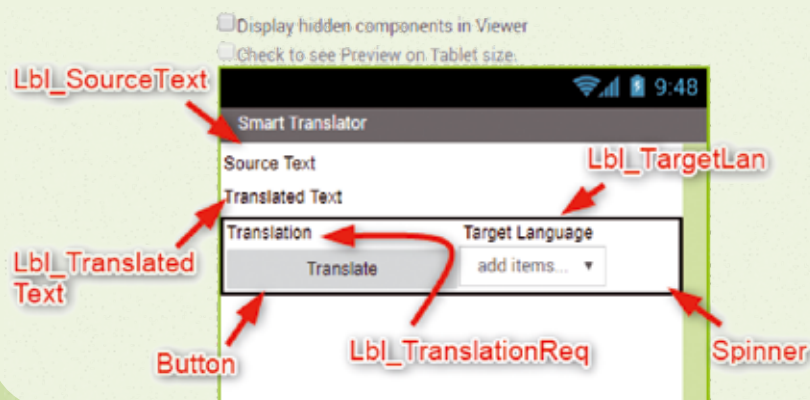
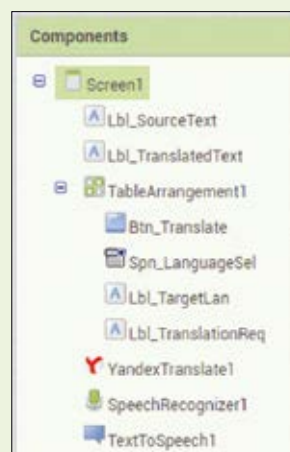


Fig. 4 - Alberatura dei componenti del layout.



apposito, ma l'app può essere facilmente estesa ad una casistica più generale.

Partiamo, come sempre, dal layout, punto di partenza dello sviluppo di qualsiasi app con AI. Ci servono i seguenti componenti:

- 4 Labels, chiamate rispettivamente Lbl_SourceText, Lbl_TranslatedText, Lbl_TranslationReq, Lbl_TargetLang
- Un pulsante Btn_Translate
- Uno spinner Spn_LanguageSel
- Un TableArrangement
- Un componente TextToSpeech
- Un componente SpeechRecognition
- Un componente YandexTranslate

A questo punto sistemiamo i vari componenti nel nostro layout, come riportato nella Fig. 3.

Nella Fig. 4 trovate riportata l'alberatura (la struttura ad albero, per intenderci...) dei componenti del nostro progetto App Inventor.

A questo punto dobbiamo impostare alcune proprietà dello **spinner**: lo spinner è un elemento che permette di far comparire una finestra di pop-up con una lista di elementi tra cui scegliere. Tali elementi possono essere impostati tramite la proprietà **ElementFromString**, settabile sia dal block che dal designer editor.

Noi, nel nostro esempio pratico utilizzeremo lo spinner per creare l'elenco dei "target language", ossia delle lingue nelle quali vogliamo che venga tradotto l'input vocale fornito in italiano, ossia il parlato che il microfono dello smartphone o tablet andrà a captare.

Nell'implementare il nostro traduttore desideriamo poter fornire la scelta del target language tra le seguenti lingue:

- En – Inglese,
- Fr – Francese,
- Ru – Russo,



Fig. 5
Modifica della proprietà ElementFromString dello spinner.

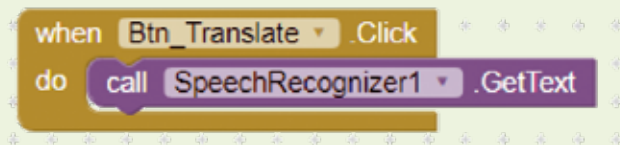


Fig. 6 - *Call del metodo GetText al verificarsi dell'evento Click.*

- Es – Spagnolo,
- De – Tedesco

Quindi modifichiamo la proprietà **ElementFromString** dello spinner come illustrato in Fig. 5, vale a dire inserendo la concatenazione di stringhe: en, de, ru, es, fr (le lingue che il traduttore dovrà gestire).

Inoltre inseriamo anche la stringa che vogliamo venga introdotta in testa al pop-up: lo facciamo inserendola nella textbox della proprietà **Prompt** ("Language Selection").

A questo punto possiamo passare al block editor



Fig. 7 - *Codice grafico che implementa la richiesta di traduzione al verificarsi dell'evento AfterGettingText.*

ed iniziare a lavorare sui behaviours; in particolare, desideriamo che alla pressione del pulsante **Btn_Translate** venga triggerata l'acquisizione del testo tramite il componente **SpeechRecognizer**, subito dopo venga effettuata la richiesta di traduzione al servizio di **Yandex** ed, una volta intercettato, l'evento **GotTranslation**, sintetizzata la traduzione. Contestualmente stamperemo sulle label **Lbl_SourceText** ed **Lbl_TranslatedText** sia il testo acquisito, sia la sua traduzione.

Per fare quanto descritto sopra, per prima cosa intercettiamo l'evento **Click** del pulsante **Btn_Translate** e chiamiamo il metodo **GetText** del componente **SpeechRecognizer** per attivare la conversione del

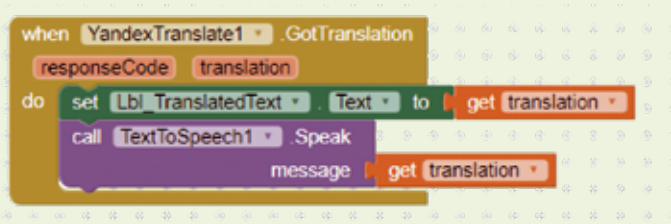


Fig. 8 - *Implementazione della parte finale del codice grafico.*

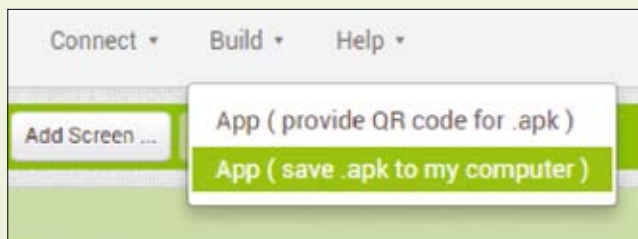


Fig. 9 - Creazione del file apk.

parlato in testo, come illustrato in Fig. 6.

A questo punto intercettiamo l'evento AfterGettingText del componente SpeechRecognizer per eseguire le seguenti operazioni:

- Impostare la proprietà language del componente TextToSpeech pari al valore selection dello spinner (questo permette di utilizzare il corretto linguaggio nella riproduzione),
- Invocare il metodo RequestTranslation del componente YandexTranslate passando il testo da tradurre (argomento result dell'evento AfterGettingText) ed i language code da utilizzare (in questo caso facciamo una join tra la stringa fissa "it-" e la selection corrente dello spinner).
- Settare la proprietà Text della label Lbl_SourceText pari al testo da tradurre (sempre argomento result dell'evento AfterGettingText), in modo da stampare sulla label il testo da tradurre, appena acquisito tramite il componente SpeechRecognizer.

Il codice grafico che implementa quanto descritto sinora è riportato nella Fig. 7.

A questo punto non rimane altro da fare che intercettare l'evento GotTranslation del componente YandexTranslate ed eseguire le seguenti due operazioni:

- settare la label Lbl_TranslatedText pari all'argomento translation dell'evento GotTranslation, in modo da visualizzare il testo tradotto sulla

rispettiva label;

- invocare il metodo Speak del componente TextToSpeech passando anche in questo caso l'argomento translation dell'evento GotTranslation in modo da riprodurlo vocalmente.

Quando descritto viene implementato tramite lo snippet grafico riportato in Fig. 8.

Per testare questo specifico esempio pratico non potete utilizzare l'emulatore; vi suggeriamo quindi di eseguire le vostre prove con uno smartphone reale, utilizzando uno dei metodi di emulazione tramite aiCompanion descritti nella prima puntata (il metodo che impiega il canale WiFi è di gran lunga il più efficiente e comodo da utilizzare).

In alternativa potete anche generare il file apk e scaricarlo sul vostro Smartphone, per poi installarlo ed eseguirlo su di esso. Per fare ciò, dovete selezionare l'opzione "APP (Save .apk on my computer)" dalla voce Build che trovate nella top bar, come illustrato in Fig. 9.

Nella Fig. 10 è riportata la fase di generazione dell'apk che consegue a tale operazione.

Conclusioni

In questa quarta puntata del nostro corso abbiamo passato in rassegna altri due interessanti componenti di MIT App Inventor, ossia i componenti Media e Social; per farvi impraticare con essi, come di consuetudine vi abbiamo presentato un esempio pratico dove vengono utilizzati tre componenti Media: stavolta si è trattato di implementare un semplice traduttore vocale simultaneo, ossia una app in grado di acquisire un testo (parola singola o frase) da tradurre, effettuare la traduzione e riprodurre il parlato corrispondente tramite l'altoparlante del device mobile.

Nella prossima puntata continueremo la nostra analisi occupandoci della descrizione di nuovi componenti di App Inventor.

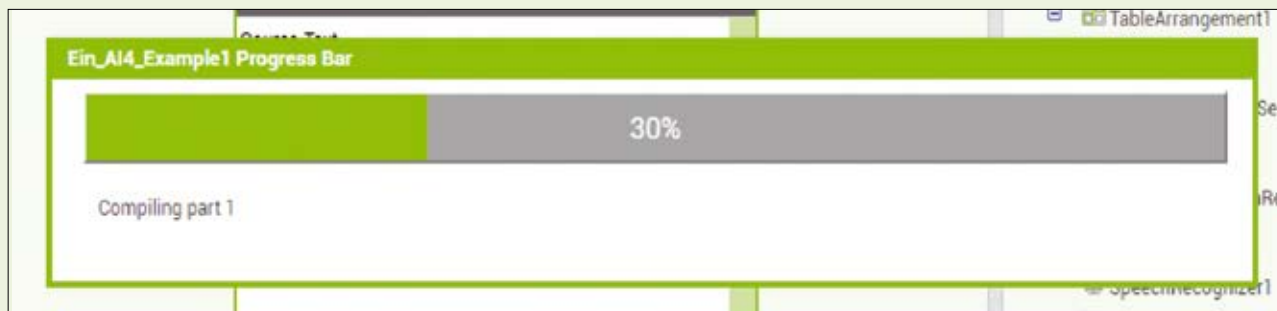


Fig. 10 - Fase di generazione del file apk.