

The logo consists of a dark green speech bubble shape pointing downwards. Inside the bubble, the word "TEMA" is written in a bold, white, sans-serif font.

TEMA

# App Inventor: from learners to makers

**Alberto Barbero**

I.I.S. "G.Vallauri"

[alberto.barbero@vallauri.edu](mailto:alberto.barbero@vallauri.edu)

Con il recente riordino delle scuole secondarie superiori, le nuove discipline in ambito informatico permettono di introdurre concetti di logica e di programmazione di base già nelle classi del primo biennio della scuola secondaria di secondo grado. Un'occasione da non perdere per valorizzare e dare dignità ad una disciplina quale l'Informatica senza travestirla o trasformarla in qualcosa d'altro. Quale curriculum prospettare a studenti adolescenti senza perdersi in complicate elucubrazioni mentali e, nello stesso tempo, fornendo strumenti che ne sappiano catturare l'interesse e l'attenzione? Un aiuto ci viene fornito dai linguaggi di programmazione a blocchi, come *Scratch* e *App Inventor*, entrambi gestiti dai ricercatori dell'M.I.T. MediaLab di Boston, che sanno unire la semplicità d'uso alla formalità dei linguaggi artificiali, per ottenere risultati motivanti in tempi ridotti lasciando spazio alla fantasia e alla creatività dei discenti.

Mentre *Scratch* è un linguaggio creato proprio con l'obiettivo di introdurre a studenti nella fascia dell'obbligo scolastico i concetti di base della programmazione e del *computational thinking* attraverso uno strumento che ad un primo approccio sa molto di ludico ma che in realtà esercita i

discenti alla logica e al ragionamento, App Inventor permette di introdurre gli studenti in modo abbastanza semplice ed amichevole alla programmazione di dispositivi *mobile* – *smartphone* e *tablet Android* – usando un approccio molto simile a quello utilizzato con Scratch. Infatti, anche App Inventor è un linguaggio di programmazione a blocchi per cui lo script lo si costruisce mettendo insieme blocchi di diversa forma e diverso colore in base al tipo di funzionalità e di comportamento, permettendo allo studente di decidere come la app apparirà e si comporterà.

## App Inventor: a robot in your mobile

App Inventor (<http://appinventor.mit.edu>) è un linguaggio di programmazione *free* che, come la nuova versione di Scratch 2.0, presenta una sua piattaforma di sviluppo *web based* in modo da lavorare direttamente sulla *cloud* fornita dai server del M.I.T. senza doverlo quindi installare sulla macchina in locale. È un linguaggio di programmazione a blocchi per cui il codice, chiamato script, lo si costruisce mettendo insieme blocchi di diversa forma e diverso colore (in base al tipo di funzionalità e di comportamento) costruendo delle pile di blocchi che si sviluppano non solo in verticale ma anche in orizzontale come in un puzzle, permettendo allo sviluppatore di programmare le tante funzionalità offerte oggi dai *mobile device*. Nello schema che segue (vedi Fig. 1) è possibile vedere la struttura della I.D.E. di App Inventor e quali sono nel dettaglio gli strumenti da utilizzare per sviluppare una app.

App Inventor può essere utilizzato per sviluppare app per Android programmandone le principali funzionalità offerte dai sensori supportati dai dispositivi mobili quali l'accelerometro, il giroscopio, il sensore GPS, il *touch*, il *drag*, le funzionalità di *texting*, la gestione degli *sprite* come in Scratch e altri ancora. Inoltre combina concetti di programmazione "object-based" utili per passare poi ad ambiti più "maturi" di programmazione. App Inventor permette inoltre di salvare in locale le applicazioni sviluppate mediante la creazione di file zippati, di creare l'eseguibile (con estensione .apk o il corrispondente QR) da caricare sul dispositivo Android per la sua installazione ed utilizzo.

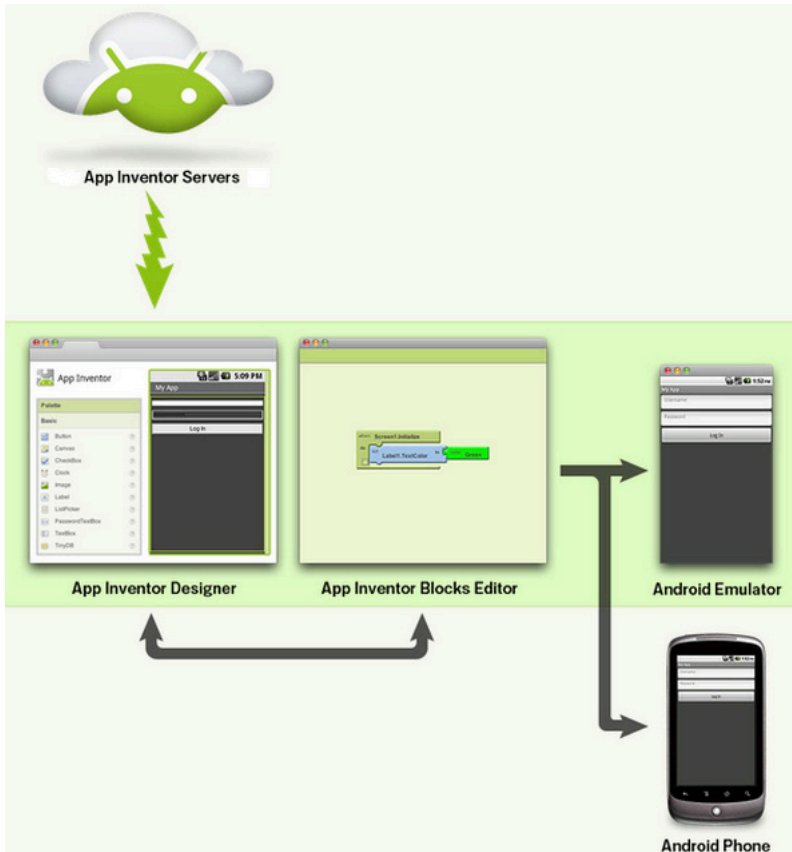


Figura 1 – Gli strumenti che costituiscono la I.D.E. di App Inventor.

Quando ci si collega al sito di App Inventor, dopo aver inserito un account *Gmail* per aver accesso al proprio spazio sui server dedicati, si apre il primo strumento della I.D.E. che è il *Designer* con il quale è possibile costruire l'interfaccia della app come comparirà sullo smartphone o sul tablet. App Inventor permette di usare le componenti più comuni dei linguaggi visuali (bottoni, caselle di testo, etichette, immagini, ecc.) più tutta una serie di componenti caratteristici dei dispositivi mobili quali l'*accelerometer sensor* e l'*orientation sensor* per la gestione del giroscopio nel posizionamento del device nelle tre dimensioni, il *location sensor* per gestire le funzionalità offerte dal GPS, il *texting* e il *phonecall* per la gestione degli sms e delle chiamate telefoniche, *camera* e *videoplayer*

per la gestione delle funzionalità di macchina fotografica e video, l'*activitystarter* per lanciare servizi quali, ad esempio, le *Google Maps* e molti altri ancora. Per non dimenticare il componente *imagesprite* che svolge la stessa funzione degli sprite di Scratch. Ad ognuno di questi componenti è possibile associare delle proprietà e gestirne gli eventi attraverso metodi predefiniti o costruiti ad hoc.

Il secondo strumento della I.D.E. di App Inventor è il *Blocks Editor* che presenta un'area in cui "trascinare" i blocchi che andranno a costituire lo script. I blocchi sono raccolti in diverse categorie e sono colorati in modo diverso a seconda della categoria e se si tratta di blocchi corrispondenti ad eventi, metodi, istruzioni di set o get.

Un ultimo strumento delle I.D.E. è costituito dall'*Android Emulator* che permette di lanciare l'esecuzione della app e vederne quindi il funzionamento durante lo sviluppo (è possibile svolgere la stessa funzione direttamente sul device Android collegato alla macchina via USB o *wifi* attraverso una apposita app).

App Inventor permette, infine, di salvare in locale le applicazioni sviluppate mediante la creazione di file zippati e di creare l'eseguibile (con estensione .apk) da installare sul dispositivo Android via USB o attraverso il codice QR.

## Una app per giocare al Tris

Nella app "Tic-tac-toe", il classico gioco del tris, viene gestita la funzionalità del tocco del dispositivo Android mediante la gestione dell'evento *Touched* previsto da App Inventor. Il primo passo per lo sviluppo del progetto consiste nel disegnare l'interfaccia (vedi Fig. 2) utilizzando le componenti offerte nel Designer e andando a settarne le proprietà. In questo caso si è fatto uso di nove componenti *ImageSprite* inseriti all'interno di un componente *Canvas* per poterne gestire l'evento "Touched" e di un componente *Button* che reagirà all'evento "Click". I Media che faranno parte del progetto sono tre immagini rappresentanti i disegni della casella vuota, della casella con una croce all'interno e di un cerchio all'interno. Tra i media possono essere uploadati non solo immagini ma anche suoni e video.

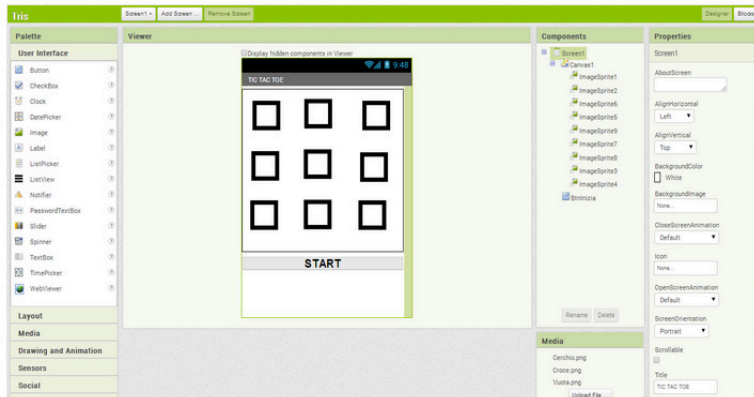


Figura 2 – Il Designer per la app “Tic-tac-toe”.

Il passo successivo consiste nell’andare a programmare gli eventi che riguardano le componenti con lo sviluppo dello script mediante l’utilizzo del Blocks Editor. L’immagine successiva (vedi Fig. 3) mostra il codice a blocchi associato alla gestione dell’evento “Touched” di uno dei nove componenti sprite utilizzati, mentre viene tralasciato il codice che gestisce l’evento “Click” del bottone Start. Il passo finale consiste nel lancio dell’emulatore Android con cui è possibile vedere l’esecuzione della app andando a caricare il programma sviluppato.

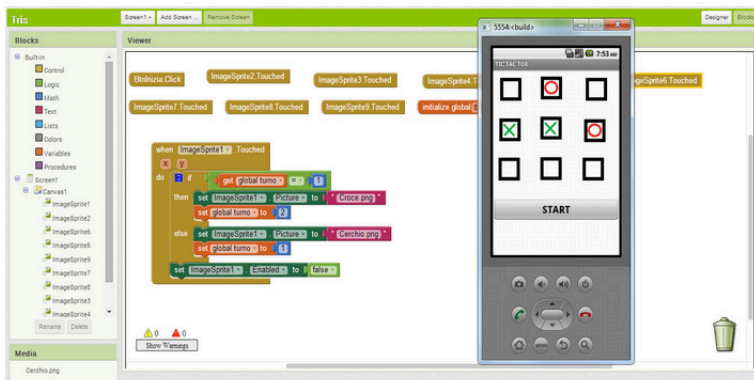


Figura 3 – Il Blocks Editor e l’emulatore per la app “Tic-tac-toe”.

## Un puzzle in una app

Nella app che segue vengono programmati gli eventi "Dragged" e "CollideWith" che permettono, rispettivamente, di gestire lo spostamento di un oggetto sullo schermo del dispositivo e la sua collisione contro un altro oggetto. In questo esempio, occorre andare a posizionare la tessera nella posizione corrispondente completando la figura. Nel Designer (vedi Fig.4) si costruisce l'interfaccia desiderata selezionando le componenti e settandone le proprietà interessate.

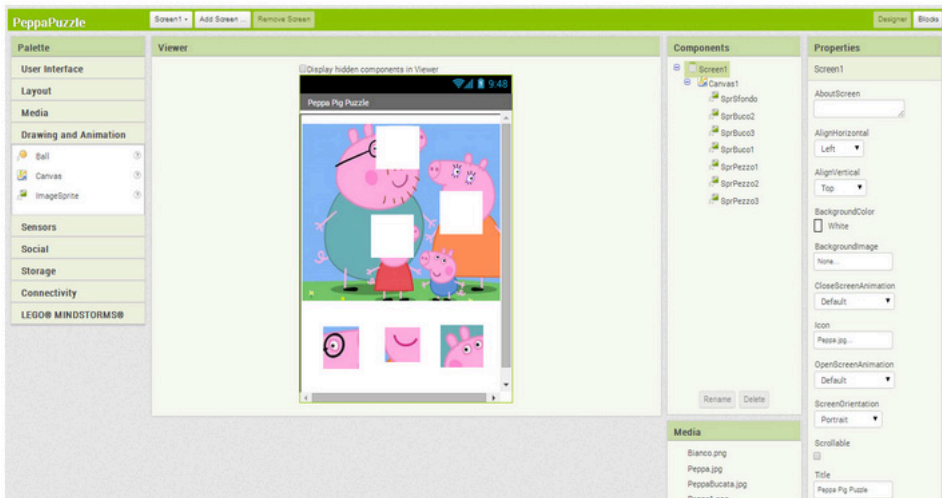


Figura 4 – Il Designer per la app "PeppaPuzzle"

Quindi, nel Blocks Editor, si passa alla programmazione degli script per la gestione degli eventi interessati. Nell'immagine che segue (vedi Fig. 5) sono visualizzati i blocchi per la gestione degli eventi "Dragged" e "CollideWith" per la prima tessera; come si può osservare, i blocchi che costituiscono gli script sono impilati in verticale ma si combinano tra di loro anche in orizzontale come in un puzzle. In questo modo il controllo sulla sintassi dei comandi avviene in modo automatico sgravando quindi l'utente da eventuali problemi.

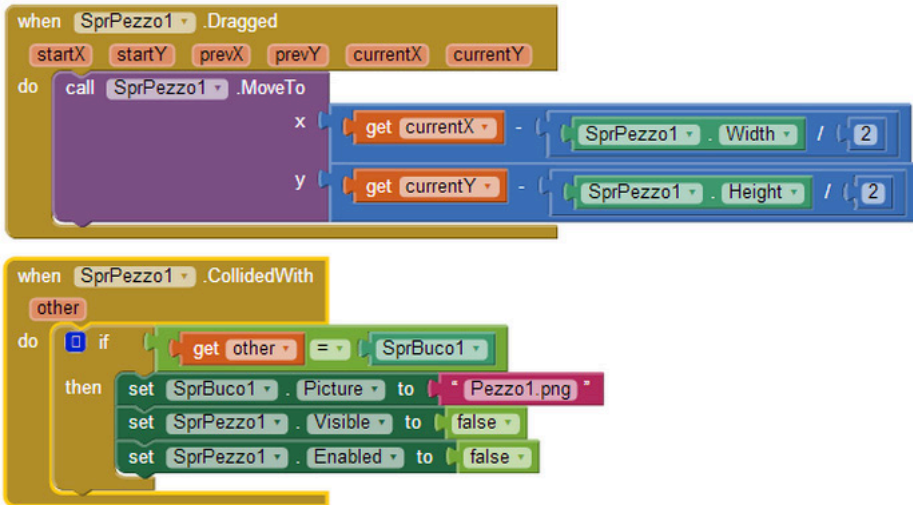


Figura 5 – Gli script per la app “PeppaPuzzle”.

Una innovazione di App Inventor consiste non solo nel permettere all’utente di selezionare i blocchi trascinandoli sul “Blocks Editor” ma anche di scriverne direttamente l’identificativo, creando così automaticamente il blocco desiderato. Con questa modalità si abitua l’utente a velocizzare l’editing dello script passando dalla selezione “visuale” del blocco desiderato alla sua codifica “testuale” come per i linguaggi di programmazione standard.

## Una app che risponde agli sms

In questa app il nostro smartphone risponde in automatico ai messaggi di testo che arrivano rinviando al mittente un messaggio predefinito e leggendo al destinatario il contenuto del messaggio ricevuto attraverso la sintesi vocale impostata. Nell’immagine che segue (vedi Fig. 6) è possibile vedere l’utilizzo dei componenti *Texting* per gestire la ricezione e l’invio di sms, *TextToSpeech* per la gestione della sintesi vocale e *TinyDB* per l’implementazione della persistenza dei dati nel device.



Figura 6 – Il Designer per la app “No text while driving”.

Quindi, nel Blocks Editor, si passa alla programmazione degli script per la gestione degli eventi interessati. Nell’immagine che segue (vedi Fig. 7) sono visualizzati i blocchi per la gestione degli eventi “MessageReceived” e “Initialize”. Il primo viene attivato alla ricezione di un sms e rimanda automaticamente al mittente un altro sms di risposta e lo fa leggere al sintetizzatore vocale; il secondo evento viene attivato al lancio della app e permette di leggere dalla memoria del device l’eventuale nuovo messaggio di risposta impostato.



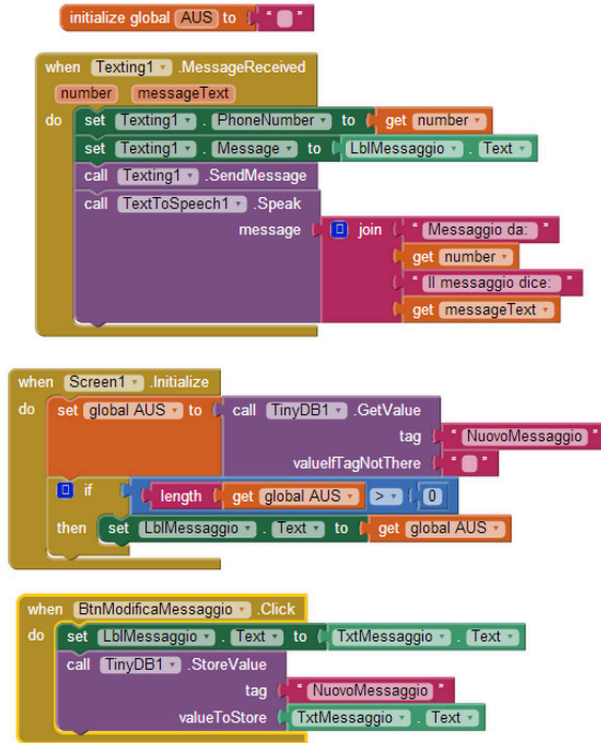


Figura 7 – Gli script della app “No text while driving”.

## Conclusioni

Dopo aver studiato e sperimentato il linguaggio Scratch, il passaggio allo studio di App Inventor è risultato sicuramente vincente per il fatto che la sua semplicità ed immediatezza, uniti alla sua vocazione per la multimedialità e per la gestione delle “mobile features”, ne fanno uno strumento particolarmente adatto a motivare gli studenti all’apprendimento delle basi della programmazione di applicazioni per tablet e smartphone Android.

Motivazione che rende gli studenti consapevoli di aver acquisito capacità operative già spendibili per progettare e realizzare strumenti che non sono solo mere esercitazioni scolastiche con lo scopo precipuo di “imparare un qualcosa” ma per “fare un qualcosa”. Un cambiamento nel modo di fare didattica che rende gli studenti reali protagonisti del loro apprendimento.