### **Tutorial Ai2 – 2 – User Interface 1**

Quindi, entrati nel nostro account, in alto a sinistra clicchiamo su "nuovo progetto" (New project), e andiamo a vedere nella finestra di sinistra, uno per uno tutti i componenti. Vediamo che nella finestra c'è scritto USER INTERFACE, con i vari componenti, partiamo dal primo:



**Button:** che dire, questo è sicuramente l'elemento piu utilizzato in assoluto, è il bottone, il bottone può assumere diverse forme, colori, dimensioni e "natura". Natura? se state sollevando un sopracciglio è un bene. Si, poichè al posto del bottone possiamo anche caricare un immagine e "addivenirla" a bottone, anzi, per spiegarmi meglio, per prima cosa, scelgo il componente button e lo posizione nella parte bianca dello schermo del mio progetto, secondo, una volta selezionato il bottone, sulla destra appare un altro menu, dove ci sono le specifiche del componente...dimensioni, tipologia, grandezza dei caratteri inscritti, e immagine. Carico un immagine e questa sarà il mio bottone, ovvero se ci clicco sopra, questa compirà una determinata azione che ho precedentemente impostato.

**Textbox:** è semplicemente il componente che utilizzeremo quando avremo necessità di far inserire all'utente un qualsivoglia dato scritto. In seguito vedremo come eliminare gli spazi, inserire o togliere virgole, confrontare il testo e tante altre cose.

**CheckBox:** come dice il nome stesso, si tratta di un componente preconfigurato, dove è presente una riga per inscrivere la frase da voi scelta e un quadrato da spuntare.. es: Acconsenti l'uso dei tuoi dati personali? [V] dove posizionerete la spunta. Tutto qui, molto semplice e utile...

Per facilitarvi la comprensione dei vari componenti, ho pensato che fosse meglio dividerli per difficoltà, in modo da tenere sempre il cervello ben acceso. Ecco quindi che dal "USER INTERFACE" passiamo alla sezione "SENSORS".

Palette	
User Interface	
Layout	
Media	
Drawing and Animation	
Sensors	
😚 Clock	(?)
RarcodeScanner	

**Clock:** se pensate che questo componente non possa esservi d'aiuto nella creazione delle vostre app, dovrete ricredervi...infatti, il clock è uno dei componenti piu utili in assoluto. Moltissimi i componenti con i quali interagisce...potremo infatti Iniziare un contatore, mettere in pausa una esecuzione, intercettare una funzione o terminare l'esecuzione di una procedura o di un processo....ma anche per questo componente troveremo modo di approfondire il discorso in seguito.

Ora invece, vediamo un componente che "DOVREBBE" essere semplice...ma in realtà.....

Palette	
User Interface	
Layout	
TableArrangement	7
VerticalArrangement	(?)
HorizontalArrangement	?
Media	
Drawing and Animation	
Sensors	
Social	
Storage	
Connectivity	
LEGO® MINDSTORMS®	

**Layout:** ahahaha e già, questo componente all'inizio mi ha davvero dato delle rogne, quando ancora non capivo il meccanismo, ma ad oggi non impegno nemmeno un secondo a scegliere il componente giusto per il mio scopo. Questo componente è in realtà semplicissimo ma difficile da capire quando si è alle prime armi... consideriamo la condizione in cui sono costretto a mettere tre tasti sulla stessa riga, poi altri tasti, sotto quello centrale, a formare una "T".

Quale layout dovremmo utilizzare per primo? il vertical Arrangment? o l'horizontal? beh, se dovete sistemare i tasti uno di fianco all'altro scegliete l'horizontal, ovvero il componente che dispone in orizzontale i componenti.... e gli altri? gli altri ovviamente sono da disporre all'interno del vertical. Da ricordare che ogni Layout (sia il Vertical che l'hoizontal), hanno un proprio allineamento distinto. L'horizontal è a destra, a sinistra e al centro, invece, il vertical è in alto, al centro e in basso... potrà sembrare che queste informazioni siano stupide, banali, ma vi garantisco che conoscerle vi eviterà parecchie frustrazioni future.



Ora poniamo il caso in cui, questi bottoni debba metterli non piu a forma di "T" ma a forma di "U", in questo caso come fareste per disporre i bottoni solo sui lati e in basso? Quanti horizzontal/vertical dovrò utilizzare per raggiungere la soluzione? La risposta è: quanti ne servono. Considerando che ci sono milioni di modi per arrivare ad un unica soluzione, sentitevi liberi di provare e verificare se è meglio utilizzare gli uni o gli altri. Ricordate però la regola generale, meno affatichiamo il processore, maggiore sarà la velocità di esecuzione dell'app, migliore il risultato finale.



### ene, dove eravamo rimasti?

Pale	ette	
Us	er Interface	
	Button	۲
00	TextBox	۲
≡	ListView	۲
201	DatePicker	۲
8.0	TimePicker	٢
~	CheckBox	۲
A	Label	٢
	ListPicker	۲
-	Slider	۲
	PasswordTextBox	۲
	Notifier	۲
۳	Image	۲
۲	WebViewer	۲
	Spinner	۲

La Label: la Label è il componente che utilizzeremo ogni volta che vi sarà necessità di far apparire un testo, comunicare con l'utente, informare, presentare una lista, numeri, indirizzi, qualsiasi cosa sia scritta.

ListView e ListPicker: come dice il loro nome, il Listview serve per presentare una lista, per poterla visualizzare, mentre il ListPicker si presenta come un Button ma ha la particolarità di aprire una lista non appena selezionato. Utilizzare l'uno o l'altro cambia e deve essere adattato alle nostre esigenze. Per predisporre la lista di entrambi i componenti abbiamo due possibilità:

1) possiamo scrivere la lista dal blocks editor, scegliendo il componente "Elements from strings" e dovremo scrivere la lista in questo modo: patate, carote, ananas, pere, pesche, ecc, ecc,



2) nella stessa maniera, ma direttamente dalla parte del design editor selezionando il componente e immettendo in alto a destra le nostre voci della lista SENZA NECESSITA' DI DOVERLA SCRIVERE CON I BLOCCHI. Il perchè della scelta per la prima o la seconda possibilità è presto detto:

(a mio avviso OVVIAMENTE) Utilizzeremo la prima scelta quando la lista è di dimensione variabile (es: nel caso di una rubrica nella quale posso aggiungere nomi all'infinito), nella seconda, quando la lista è fissa (es: se eseguo la scelta di una città da selezionare).

# RICORDA: IL LISTVIEW NON FUNZIONERA' O FUNZIONERA' MALE SE LO SCHERMO (SCREEN) E' SETTATO SU SCROLLABILE (scrollable).

Date Picker: semplice ma (un sentito ringraziamento agli sviluppatori del MIT) utilissimo componente che ci permetterà di selezionare una data per poterla visualizzare, utilizzare e gestire.

Sto saltando da un componente all'altro? Si esatto, è un ottimo esercizio per mantenere il cervello attivo.



TimePicker: stessa cosa dell'ultimo componente ma al posto della data, sarà possibile visualizzare, utilizzare e gestire l'orario.

**Spinner:** stessa funzione del ListView ma....ma porca paletta quanto mi ha fatto indiavolare questo componente quando ero alle prime armi! Ho perso più di un mese cercando di utilizzarlo in una app prima di accorgermi che non era il componente adatto alla funzione che doveva assolvere! Infatti, questo componente è stato appositamente creato con la "spunta" sul primo Elemento (Element) della lista...di conseguenza non genera un evento After Picking... beh...e quindi.... il suo utilizzo è dedicato alla presentazione di una lista laddove il primo elemento risulta essere la scelta migliore per tutti gli utenti. Esempio: finisci la benzina e pian piano ti fermi e accosti sul ciglio della strada, metti le quattro frecce, scendi dall'auto utilizzando l'apposito gilet ad alta visibilità, e noti che proprio li vicino c'è il cancello (aperto) di un terreno...cosa fai?

A) Essendo in Sardegna, non mi avvicino, troverò sicuramente qualcuno lungo la strada per aiutarmi. (Bravi, qui se non si è invitati ad entrare in un terreno, o in una casa, PRIMA si spara e POI si chiede chi è).

B) Cautamente percorro la recinzione del terreno e cerco di scorgere il pastore per chiedergli aiuto, essendo in Sardegna e conoscendo l'immensa ospitalità dei Sardi, mi aiuterà. (Bravi, anche questa scelta sarebbe un'ottima risposta).

C) Entro nel terreno e prego che il pastore fonnese che mi corre incontro con gli occhi rosso ardente sia semplicemente stupito di vedermi e quindi voglia farmi le feste.

(Tanti Auguri...in quanto tempo fate i 100 metri piani? Penso sia arrivato il momento di scoprirlo 🧡

### tutorial AI2 – 2 – User Interface 3



**Notifier:** avete presente quella finestrella che esce quando per errore cliccate EXIT al posto di SALVA che vi chiede "Sei sicuro di voler uscire senza salvare?" ecco, quello è una delle tante forme che può assumere il notifier. Infatti, la sua funzione è proprio quella di Notificare all'utente delle informazioni che lo aiutino nell'uso dell'app, mediante la proposta di una scelta, informandolo di un evento (es: login-logout, Caricamento, ottenimento delle coordinate, ecc) oppure per informare l'utente DOPO aver inserito del testo in una textbox. Insomma vengono distinti in due categorie i "dialog" e gli "alert".

E' importante sapere che quando utilizziamo il Notifier del tipo "dialog" possiamo modificarne la formattazione mediante l'uso dei tag HTML (piccole stringe di codice che informano Java di modificare il testo posto all'interno di specifici marcatori (come <html>). Nello specifico, quelli che potremmo utilizzare sono: HTML tags:

<b>, <big>, <blockquote>, <br>, <cite>, <dfn>, <div>, <em>, <small>, <strong>, <sub>, <sup>, <tt>. <u> .

Sperimentate! ah, dimenticavo, anche il colore può essere modificato mediante questo tag: <font color="green"> (ovviamente al posto di green potete usare altri colori).

Una piccola lista di altri colori utilizzabili è questa: aqua, black, blue, fuchsia, grey, lime, maroon, navy, olive, purple, red, silver, teal, white, e yellow.

**Password Textbox:** è esattamente uguale alla textbox, con l'unica differenza che, quando viene immesso del testo, questo viene occultato, in modo da non poter essere letto quando viene digitato.

Pale	ette	
Us	er Interface	
	Button	۲
80	TextBox	۲
=	ListView	۲
201	DatePicker	۲
8.0	TimePicker	۲
~	CheckBox	۲
A	Label	۲
	ListPicker	۲
-	Slider	۲
	PasswordTextBox	۲
	Notifier	۲
2	Image	۲
۲	WebViewer	۲
	Spinner	۲
12		115

**Slider:** lo slider è un componente alquanto particolare, si tratta infatti di una "Progress bar" utilizzata principalmente come strumento per aumentare o diminuire un certo valore (per esempio il Volume Audio). Si ho detto principalmente, poichè può essere utilizzato per tantissimi altri motivi, es: lo zoom su una mappa oppure per notificare all'utente lo stato di caricamento del segnale gps, o ancora, la progressione di un file audio o video...

Inoltre si può decidere quale colori utilizzare per la parte a sinistra del cursore e quale per la parte a destra, il valore minimo e massimo e di quanto deve essere la progressione tra questi due valori.

Webviewer: questo componente è in breve una gran figata.

Ovvero, è il componente che ci permette di collegarci e visualizzare le pagine Web, ma…non solo!, infatti è utilizzato per comunicare tra l'App e il Javascript contenuto nelle pagine WEB.

ATTENZIONE: a prima vista potrebbe sembrare un Browser, ma non lo è, infatti, qualora si dovesse cliccare su (BACK) indietro, non tornerete all'app ma uscirete completamente, chiudendo l'app. Sempre che non abbiate messo la funzione FollowLinks e non abbiate programmato il tasto indietro in maniera da svolgere altre funzioni.

Bisogna inoltre perdere un minuto per spiegare brevemente alcune voci che non sono molto comprensibili: Ignore SSL errors.

Spuntate questa funzione se desiderate che l'app salti l'errore che si verificherebbe entrando in un sito che si autodefinisce come "SICURO".

**Prompt for Permission**: è la funzione che se è settata su "true" vi richiederà il permesso di avere l'accesso al servizio di geolocalizzazione, mentre se è settata su false viene considerata come "permesso accordato" in automatico.

### **Tutorial AI2 – 3 – Media Component**

Pale	ette	
Us	er Interface	
La	yout	
Me	edia	
	Player	1
(۵	Sound	(?)
	TextToSpeech	0
۲	YandexTranslate	(?)
-	VideoPlayer	0
-	Camcorder	0
Ô	Camera	(?)
	SpeechRecognizer	(?)
	ImagePicker	0
•	SoundRecorder	?
Dra	awing and Animation	
Se	nsors	
So	cial	
Sto	orage	
Co	nnectivity	
LE	GO® MINDSTORMS®	

**Camcorder:** grazie a questo componente sarà possibile utilizzare il nostro dispositivo per registrare i video, al termine della registrazione è possibile impostare una determinata azione (es: dare un nome al video, decidere se salvarlo sulla sd o sul dispositivo, ecc). Davvero molto semplice.

**Camera:** come è facile capire, questo componente è utilizzato per gestire la fotocamera del dispositivo, è possibile sfruttare (se presente) anche la fotocamera frontale, e esattamente come il componente Camcorder è possibile, dopo lo scatto, impostare una determinata azione.

**Image Picker:** questo componente si presenta come un tasto che, se cliccato, apre la galleria immagini, una volta selezionata l'immagine, questa viene salvata in una porzione di memoria apposita. Per non riempire la memoria si è scelto di memorizzare fino ad un massimo di 10 immagini, oltre le quali, la successiva sovrascriverà la piu vecchia.

**Player:** se dovete far riprodurre una canzone, una colonna sonora, in generale "musica" o la vostra app necessita di una "vibrazione", il componente player non dovrà mancare. Ricordo che la vibrazione viene espressa (così come il clock) in millisecondi, quindi, se volessimo far vibrare il telefono per 2 secondi, impostiamo il blocco "Call Player1 vibrate millisecond" e aggiungiamo "2000".

ATTENZIONE: il componente Player è stato studiato appositamente per riprodurre file audio di dimensioni medio – grandi, per riprodurre un suono o un effetto è consigliato utilizzare un altro componente di nome "Sound".

**Sound:** questo componente si utilizza esattamente come il Player ma è prerogativa per la riproduzione dei file di piccole – piccolissime dimensioni.

**VideoPlayer:** il nome dice tutto, grazie a questo componente è possibile riprodurre video in .3gp e MPEG-4, purtroppo, con App Inventor 2, non è possibile costruire applicazioni maggiori di 5Mb (o meglio, è possibile ma non utilizzando la maniera "convenzionale"), pochissimo se consideriamo che la qualità e la definizione delle videocamere, come anche degli smartphone sta sempre andando in crescendo…ma, e qui viene la cosa bella, è possibile dare in pasto al componente (intendo proprio come file sorgente) un <u>URL</u> che riporta ad un video in streaming… da qui possiamo capire che le possibilità crescono notevolmente.

SoundRecorder: è un componente semplicissimo, registra l'audio esterno.

**SpeechRecognizer:** vi è sicuramente capitato di provare l'app Google Now, ecco, l'interfaccia di funzionamento è molto simile. Mediante questo componente, infatti, è possibile parlare al telefono e riprodurre in testo le parole dette.

**TextToSpeech:** avete presente il componente SpeechRecognizer? Questo componente è il suo esatto contrario, ovvero è utilizzato per far dettare al dispositivo, un determinato testo. Piccola particolarità: è possibile impostare le lingua e la città di provenienza, questo fa sì che possiamo per assurdo sentire parlare un Inglese con accento Tedesco o un Italiano con accento Francese ecc, ecc.

**YandexTranslate:** eccovi il vostro traduttore, per gentile concessione della casa <u>"Yandex"</u>. Ricordate però che per poter funzionare, questo componente necessita della connessione dati. Inoltre, dovete sapere che questa app opera in background e in maniera asincrona, per cui, quando il servizio della yandex risponderà alla nostra richiesta di traduzione, l'evento "Got Translation" verrà <u>"triggerato"</u>.

### tutorial Ai2 – 4 – Drawing and Animation



**Canvas:** questo componente è un ottimo punto di partenza se si desidera creare dei giochini poco impegnativi, infatti, questo componente è sensibile al tocco e permette lo scorrimento ed il movimento di un altro componente che spiegheremo immediatamente dopo, lo sprite. Tanto per capirci, il Canvas non è altro che la base, lo sfondo per le nostre animazioni. E bene sapere che per identificare un punto nel canvas bisogna individuarne le coordinate, niente paura, è piu difficile dirlo che farlo, presto capirete il perchè.



**ImageSprite:** questo componente trasformerà una qualsivoglia immagine (jpg o jpeg) in uno sprite, ovvero un componente capace di interagire con altri sprite o con lo stesso canvas, infatti, per esempio, è possibile determinare una azione quando lo Sprite collide (va a sbattere) con il margine del Canvas.

	Components	Properties
Display hidden components in Viewer	B Screen1	ImageSprite1
Check to see Preview on Tablet size.	Canvas1	Enabled
Screen	Clock1	Heading 0 Height 150 pixels
		Width 150 pixels Interval
		Picture IMG_20141207_155254.jpg Rotates
	Rename Delete	Speed 0.0 Visible
	Media	181
Non-visible components Motifier1 Clock1	File0001.jpg File0032.jpg IMG_201455254.jpg Upload File	Y -18 Z

Ball: è una palla.

Si, con questo non voglio dire che l'argomento è una palla, intendo dire che questo componente è realmente una palla, della quale possiamo modificare la dimensione, il colore, impostare l'inclinazione e molto altro ancora.



E' utile sapere che ognuno di questi componenti supportano il trascinamento (Flung). Inoltre, tenete bene a mente che quando utilizzate questo componente, nella parte del Design editor, appariranno le coordinate X e Y che vi aiuteranno per lo sviluppo dei vostri giochini.



Piccola anticipazione sulle novità di App Inventor: con l'arrivo delle extensions, arrivano molte nuove, ottime funzionalità, tra queste, il Pitch to Zoom... ovvero la possibilità di selezionare un punto del Canvas (dove inseriremo per esempio una foto), e tramite un particolare studio di astrazione (grazie al <u>Prof. Hal Habelson</u>, (il principale autore di App Inventor!) è possibile utilizzare la ball per ottenere l'effetto Zoom, se non avete capito niente di quello che ho appena scritto, tranquilli è normale! ma vi basterà seguire questo link del mitico Seblog perchè tutto vi venga svelato: <u>mit-app-inventor-multitouch-e-pinch-to.html</u>.

# Ai2 Tutorial- 5 – Componenti Sensors

**Il Barcode Scanner**: questo componente è il classico lettore Qr-Code (se volete scaricarne uno, vi consiglio questo: <u>QR-CODE scanner Light&Fast</u> l'app è fatta proprio con questo componente.



Ricordate: quando preparerete un'app lettore Qr-Code, dovrete togliere la spunta dall'opzione "UseExternalScanner" perchè questo significa che la vostra app, cercherà di aprire il vostro programma per la lettura qr code già installato nel vostro device...e se non ne avete neanche uno, l'app andrà in errore, presentandovi l'apposito Notifier...

**Location Sensors**: Niente popò di meno che il classico GPS Global Position System. Già, però se lo uso in una app, questa consumerà un sacco di batteria... Sbagliato! Questa è l'informazione che si è tramandata da quando il gps era così enorme da richiedere due batterie di camion per funzionare... Aggiornatevi! Gli attuali sistemi GPS sono basati sul "<u>A-GPS</u>" e sul sistema "<u>GLONASS</u>", se volete saperne di più vi inserisco il link per togliervi ogni curiosità.

**Orientation Sensors**: avete presente l'app bussola o la livella? bene, è grazie a questo componente che potrete sviluppare queste app. In seguito approfondiremo l'argomento, ma sappiate che è talmente automatizzato che risulta essere molto semplice da utilizzare.

**Near Field**: Questo componente è adibito alla comunicazione tra dispositivi vicini, che supportino il sistema Near Field Communications, c'è da dire che attualmente, questa versione supporta la lettura e la scrittura di soli Text Tags (piccole stringhe di testo), mentre immagini e audio non sono supportati.

Accellerometer Sensor: è un sensore in grado di valutare l'accelerazione imposta al dispositivo, nelle tre dimensioni (graficamente sono riconosciute come X,Y,Z). Con questo componente, se riuscirete ad ottenerne la piena padronanza, vi garantisco che vi divertirete parecchio, tanto per citare un esempio, immaginate di impostare il suono della spada laser di Star Wars nelle varie dimensioni (rotazione a destra-sinistra, avanti e indietro, su e giu) ...potete immaginare quanto si divertirebbero i bambini... e anche qualche adulto...ed ho fatto solo un esempio.

**Proximity Sensor:** alcuni modelli di smartphone, hanno in alto a destra dello schermo un piccolo sensore che ha lo scopo di togliere la corrente allo schermo, quando questo viene avvicinato all'orecchio, o poggiato sul tavolo, affinchè si eviti di digitare qualcosa per errore e con l'intento di risparmiare energia. Questo sensore è decisamente divertente da utilizzare con AI2, perchè possiamo utilizzarlo "dall'esterno" del nostro dispositivo, semplicemente passando la mano vicino allo schermo,

# Ai2 Tutorial – 6 – Componenti Social



Ciao a tutti, vi ringrazio ancora una volta per il continuo interesse nei miei articoli, vorrei iniziare quest'ultimo facendo una puntualizzazione: errare è umano, se doveste trovare degli errori fatemelo notare! Come? semplicemente scrivendomi una mail all'indirizzo gabrip765@gmail.com.

**Email Picker**: questo componente è simile ad una text box, cambia pero il comportamento, ovvero, quando si immette del testo al suo interno, viene presentata una lista che riporta i risultati corrispondenti al testo inserito, c'è da dire che se le mail o i contatti sono molti, questo componente ha una certa latenza. Inoltre, elabora i risultati man mano che si immette il testo. Tenetene conto se desiderate creare una app snella e veloce.

**Texting:** questo è un bel componente, infatti permette la lettura e la scrittura di messaggi sms, permette di scriverli o leggerli tramite Google Voice e, a seconda dell'impostazione "Receiving Enabled Property" (proprietà ricezione abilitata) funzionerà con l'app in uso, in background o addirittura con l'app chiusa!

**PhoneNumberPicker**: appare come un tasto, se cliccato, presenterà una lista di numeri di telefono (la rubrica) dalla quale scegliere il contatto desiderato, di questo ci restituirà l'identità, numero di telefono, email e immagine del profilo. Non funziona su tutti i device purtroppo.

		🦻 🗐 🥵 🗐 🦻
Screen1		
	Text for PhoneNumberP	icker1
	Text for ContactPicke	r1
		_
	ý L	
	Non-visible compo	nents
-	Eexting1_PhoneCall1_Shari	na1 Twitter1

PhoneCall: usato spesso con il ContactPicker che spiegheremo qualche riga sotto, questo componente è utile per poter avviare una chiamata.

**Sharing:** utilissimo componente che permette lo scambio di file e/o messaggi, tra l'applicazione e le altre applicazioni in grado di utilizzarli. Un classico esempio di utilizzo è dato dal messaggio preimpostato, contenente il link per il download dell'app.

**Twitter**: devo dire che se si è appassionati di social network, nello specifico di twitter, questo componente è una vera e propria "manna dal cielo", infatti sarà possibile, previa creazione di un account su <u>http://twitter.com/oauth\_clients/new</u> e l'ottenimento di una Comsumer Key e del Consumer Secret, sfruttare tutta una serie di azioni che vanno dalla ricerca degli ultimi tweet in rete, al messaggio diretto, al controllo dei follower e tante altre cosettine carine.

**ContactPicker**: e un bottone che se chliccato presenterà una lista di contatti da cui scegliere, inoltre sarà possibile estrapolare il ContactName, ovvero il nome del contatto, l'email, il ContactUri, ovvero l'indirizzo di dove si trova registrato questo dato all'interno del nostro dispositivo, e per finire, il Picture, che è l'immagine scelta per il profilo. Attenzione però, anche questo componente non funziona su tutti i dispositivi.

# Ai2 Tutorial – 7 – I componenti Storage

#### Storage



**Tiny db:** il -piccolo database- è un componente prezioso come l'acqua nel deserto. Esso rappresenta la porzione di memoria del telefono che destiniamo alla memorizzazione dei dati delle nostre app. Mi spiego meglio, avete presente quando settate il vostro gioco preferito con le impostazioni che preferite? ecco, il tiny db permetterà di salvare quelle impostazioni affinché possiate ritrovarle non appena riacceso il dispositivo o fatto partire l'app...infatti, a differenza delle variabili, il tinydb registrerà permanentemente le vostre informazioni, fino a quando, ovviamente, non le cancellerete, sovrascriverete o disinstallerete l'app dal vostro dispositivo.



Ricordo che inizialmente questo componente era difficile da capire...ma soltanto perchè veniva rappresentato come una cosa difficile, in realtà è molto semplice e offre infinite possibilità.

Per capire all'istante questo componente, pensate alla busta della spesa. Il nostro "tag" è la nostra busta. All'interno c'è un ottima mezza forma di <u>Dolce Sardo</u>, una confezione di <u>Sebadas</u> (si pronuncia Seadas e consiglio vivamente a tutti di mangiarla almeno una volta nella vita!), croccantini per cani e per gatti, parmiggiano, cereali per la colazione del nonno...ecc, ok, questi sono tutti i valori all'interno della busta.... (i value to store, all'interno dei tag...).

Complicato, vero? naaaaaaaaa....

call TinyDB1 .StoreValue	Supermecato ccc
valueToStore	<ul> <li>join</li> <li>Dolce Sardo</li> <li>Sebadas</li> <li>Parmiggiano</li> <li>croccantini per cani e gatti</li> </ul>
	croccantini per il nonno, gli stessi per cani e gatti, tanto non sente il sapore 🧃

Un'altra caratteristica del tinydb: anche se nella mia app inserisco 500 tinydb, il dispositivo non rallenterà, poichè in realtà il componente è uno solo, ma con diversi nomi.

Se	e volessi	recuperare	tutti	i	miei	prodotti	all'interno	della	bustadevo	solo	chiedere
	call (TinyDB1 ).GetV valueIfTagNotT	/alue tag <b>Constant</b> Supermeca here <b>Commi que</b>	to ccc ) * sto valore se	nel tag non t	rovi niente, "						

Ma se avessi già cancellato il tag? allora, questo blocco mi restituirà il valore "Dammi questo valore se nel tag non trovi niente" anzi, qui ho fatto un errore perchè **in realtà sarebbe "se non trovi il tag".** 



Potrò cancellare completamente tutta la memoria usando ClearAll (Pulisci tutti i tag) oppure solo uno di questi (ClearTag=Pulisci il Tag "Supermercato ccc").



Inoltre (capita sopratutto in fase di <u>Debug</u>), possiamo interrogare il sistema su tutti i tag presenti fino al momento della nostra richiesta, mi spiego meglio, capita che in una app creo ventidue Tag, tredici li cancello appena hanno esaurito la loro utilità, ovvero, non mi servono piu, poi ne creo altri

4879...ovviamente non posso ricordarmi quali di questi sono ancora presenti e quali sono i tredici che ho cancellato...ecco quindi che ci corre in aiuto questo blocco:



Interrogandolo, ci darà tutti i tag attualmente in uso, così che potremo gestirli e/o debuggare l'app.

**Tinywebdb:** funziona esattamente come il tinydb ma con la caratteristica di essere in cloud...ovvero i nostri dati sono registrati su un server al quale possiamo sempre accedere (fintanto che abbiamo una rete) per poter stoccare, aggiornare, sovrascrivere, e eliminare i dati. Ovviamente prima dovrete



configurare il servizio, nello specifico, cliccate sul link http://ai2.appinventor.mit.edu/reference/other/tinywebdb.html

Ad essere del tutto sinceri non ho mai provato questo componente, per cui, non sono in grado di offrirvi nient'altro che una perla di saggezza: Smanettate!!!

E' cosi che si impara... oppure iniziate con una infarinatura grazie ad un video tutorial del grande Seblog! http://seblogapps.blogspot.it/2015/07/tutorial-11-difficile-tinywebdb-parte-1.html Per i novizi: in questo momento potrete anche saltare questo argomento, quando avrete maggiore esperienza, potrete ritornarci e sviluppare qualche bella app che sfrutti questo componente. Intanto, proseguiamo.



**File:** beh, questo è molto semplice, nelle nostre app possiamo sempre dedicare una cartella per infilarci un file di testo. Potrebbe risultare utile quando si deve lavorare con grandi file di testo oppure per salvare le impostazioni dell'app. La particolarità del componente è che potrò decidere di infilare la cartella nella directory principale (e parlo della cartella dell'app all'interno del nostro dispositivo) oppure nella sd card.



Semplicemente, qualora decida di mettere la cartella nella sd devo solo inserire uno slash davanti al

percorso in cui voglio mettere la cartella. Ad esempio: se volessi inserire la cartella /saluto.txt per ritrovarla dovremmo andare su /sdcard/saluto.txt se invece inserirsco solo saluto.txt allora la troverò dentro la cartella dell'app...alla quale non potremo accedere se non possediamo i <u>diritti di root</u> (che non è una parolaccia).

**Fusion Table**: Questo è il componente più difficile da capire e difficile da configurare, ma una volta creato, vi garantisco che la sua utilità è massima. Cosa è il Fusion Table? il fusion table è una tabella dalla quale e verso la quale è possibile inserire, interrogare, sostituire, aggiornare, condividere e cancellare dei record (dei dati, per capirci meglio) come indirizzi, coordinate, nomi, numeri, numeri di telefono, fotografie (o meglio, link alle fotografie) e altro ancora. La particolarità è che per poter interrogare e compiere tutte le azioni sopra descritte, dovrete formulare le vostre richieste sotto forma SQL (tranquilli, le sigle fanno paura ma sono solo fumo negli occhi...SQL=Structured Query Language). Sarò sincero, per crearla e configurarla dovrete cambiarvi molte volte la camicia, poichè non è facilissimo e basta anche una minima distrazione per farvi perdere molto lavoro.

Giusto per mettervi un pò di paura 👻 vi dico che ho perso una settimana dietro ad una Query che non voleva funzionare...alla fine ho scoperto che.....c'era uno spazio in piu......

Come per il tinywebdb, anche per la Fusiontable dovrete configurare un account. Ci sono due modi: si ottiene un API key e ogni volta che si tenterà di accedere alla fusiontable, bisognerà autenticarsi nei Google Server. Secondo metodo, si ottiene un Service Account Email Address che permetterà all'utente finale di accedere alla fusiontable senza eseguire il Login. Infatti l'account di servizio (Service Account Email Address) garantirà l'accesso per tutti. Vi consiglio principalmente di dare uno sguardo a questa App prefabbricata molto famosa, per iniziare ad avvicinarsi al mondo delle fusiontables: <a href="http://appinventor.mit.edu/explore/ai2/pizzaparty.html">http://appinventor.mit.edu/explore/ai2/pizzaparty.html</a>

Ricordate: questo servizio messo a disposizione da Google è gratuito fino a 10.000 chiamate al giorno, (se non vado errato) dopodicchè comincerà a costicchiare, (anche se poco, certo). Quindi, vi sarebbe necessità di avere i cosidetti in-app billing e gli abbonamenti in modo da potervi permettere il

servizio a pagamento ma, come ben sappiamo, app inventor ancora non li supporta, mentre non si può dire lo stesso per la versione modificata dal <u>Dott.</u> <u>Amerkashi</u> che si chiama "Ai2 Live complete" e in futuro la versione degli ex studenti del Mit: <u>Thunkable</u>.

Voi direte che 10.000 chiamate al giorno sono praticamente infinite...sbagliato, se doveste utilizzare questo servizio per una delle vostre app basterà che la vostra App riesca a raggiungere 1000 download e venga aperta per 10 volte che in breve avremo già raggiunto il limite...

Considerando inoltre che l'utilizzo della Fusiontable renderà le potenzialità della vostra app di gran lunga superiori rispetto ad un altra che non lo sfrutta, è gioco facile mettersi nei guai.

# Ai2 tutorial – 8 – Componenti Connectivity

Componenti Connectivity



Bene, anzi benissimo, perchè siamo al penultimo articolo sulla spiegazione dei componenti, cosi, presto ci addentreremo nella parte più interessante del nostro percorso...

iniziamo con la spiegazione dei componenti Connectivity:

**ACTIVITY STARTER:** l'Activity Starter è un componente molto potente che permette, tramite opportune stringhe di codice, di avviare attività preimpostate nel nostro dispositivo. Aprire un determinato video su YouTube, o una intera categoria, chiamare un numero di telefono o avviare una chiamate sequenziale ad un gruppo di persone, creare un Qrcode con un link personalizzato, vedere la lista delle chiamate frequenti, o l'ID di un nostro contatto in rubrica, abilitare il wifi o rendere visibile il Bluetooth ai dispositivi vicini, sono solo alcune delle cose che è possibile fare con queste

piccole stringhe di codice... attenzione però, per poter funzionare, queste stringhe devono essere inserite esattamente come ci vengono fornite... maiuscole e minuscole, punti o altro....

Alcune di queste sono descritte nell'app<u>App Inventor Master Tools</u> dalla quale prenderemo qualche spunto a titolo di esempio.

Selezioniamo come activity "ricerca un video specifico su YouTube" e avremo questa schermata:

🛇 🜵 🖻 🞧	🛜 🔏 100% 🗖 14:36
Scrollare il monitor per vec	dere maggiori informazioni.
Seleziona	Avaia Activity
Seleziona	Avvia Activity
Ma	ain
android.media.action.MEDIA	A_PLAY_FROM_SEARCH
Activit	ty Class
	,
Activity	/ Package
Data	Туре
Dat	a Uri

S 🕆 🖬 🖬 👀	Î 1	100% 💼 14:36
Ac	tivity Package	
	Data Type	
	Data Uri	
call on me	Extra value	
query	Extra key	
Activity starter result r Activity starter result: Activity starter result <sup>-</sup> Activity starter resultU	name: No result No result Type.: No result Jri:No result	

Quindi, se desideriamo visualizzare il video "Call on me" basterà inserire il componente ActivityStarter nel

nostro progetto, passare al blocks Editor e prendere il blocco ActivityStarter Action (Main stà per Action) e prendendo spunto dall'app App InventorMaster Tools, impostarlo su **android.media.action.MEDIA\_PLAY\_FROM\_SEARCH,** il blocco ActivityStarter Extra value su <u>call on me</u> ed infine il blocco ActivityStarterExtraKey su **query**. Ultima cosa, ma importantissima, dovete utilizzare il blocco "Start Activity" che darà il via a tutto!

0	set ActivityStarter1 . Action . to ( android.media.action.MEDIA_PLAY_FROM_SEARCH .
	set ActivityStarter1 . ExtraValue to 🛟 call on me
	set ActivityStarter1 . ExtraKey to ( query *
	call ActivityStarter1 . StartActivity

Vi consiglio di dare uno sguardo al video....merita!

#### **Bluetooth Server e Bluetooth Client**

In generale la comunicazione tra esseri umani, avviene grazie a due cose: una bocca e due orecchie.

#### <u>cit.: Zenone di Cizio</u>

Ipotizziamo che il Bluetooth Server sia "le orecchie" e il Client "la bocca". Perchè non il contrario? Perchè il saggio ascolta...non a caso abbiamo un numero doppio di orecchie...ma questa è un altra storia...tanto per capirci, il Server rimane in ascolto fino a chè.....

TOC TOC...Permesso? sono il Client, desidero richiedere l'accesso al tuo dispositivo, per comunicare...

Il Server, sentendosi un pò solo, accetterà la comunicazione...ma ascolterà soltanto....questo perchè?

Perchè per poter parlare, il dispositivo che funge da Server dovrà diventare Client....ed ecco quindi come far comunicare due dispositivi...tramite il bluetooth, ma a volte non serve che la comunicazione sia a doppio senso... a volte possiamo semplicemente avere un Papà Client che parla ad un Figlio Server che deve imparare "come mangiare il gelato senza sporcarsi"... Semplice no?

#### Web Component

Questo importante e misconosciuto strumento è in realtà molto utile, infatti, tramite questo componente sarà possibile interrogare la rete per capire molte informazioni delle pagina web che ci interessano, oppure per verificare se è presente la rete nel dispositivo.