



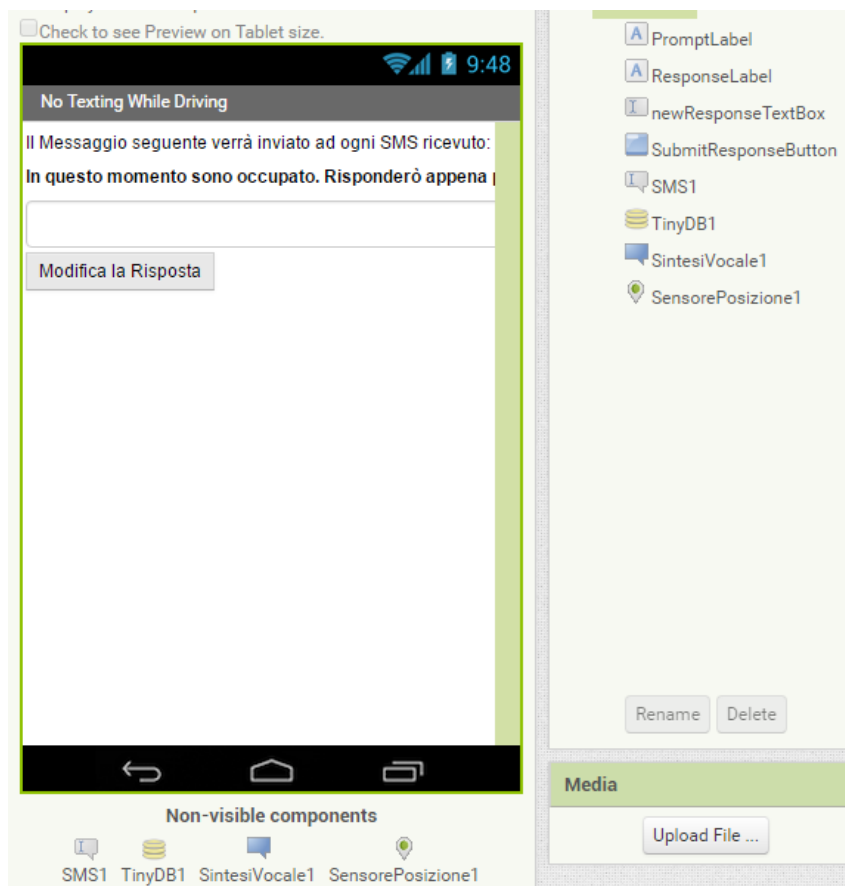
In questo tutorial andremo a creare un'app che risponderà per noi agli sms che ci arriveranno e sarà in grado di leggerne il contenuto (richiesto un cellulare).

Iniziamo!

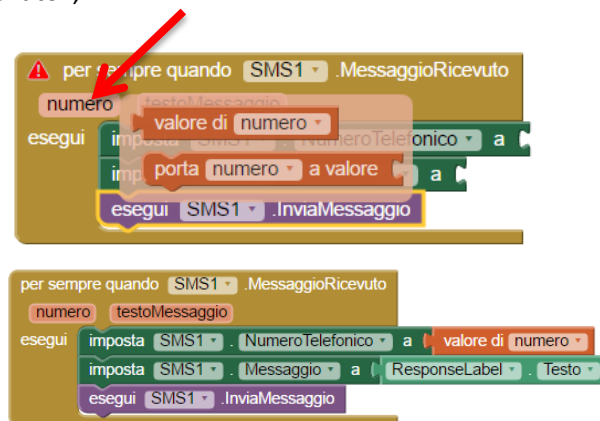
- Andiamo sul sito di AppInventor (<http://appinventor.mit.edu/>) accedendoci con il nostro utente e la nostra password;
- Creiamo quindi un nuovo progetto dal menù "Progetti\Avvio Nuovo Progetto..." e chiamiamolo "SegreteriaSMS";
- Se lo si desidera, impostiamo la lingua Italiano attraverso l'apposito menù in alto a destra;
- Inseriamo quindi nell'interfaccia i seguenti componenti:

Componente	Gruppo	Nome	Scopo
Etichetta	Interfaccia Utente	PromptLabel	Indica all'utente come funziona l'app
Etichetta	Interfaccia Utente	ResponseLabel	E' la risposta che verrà inviata a chi vi spedisce un SMS
CasellaDiTesto	Interfaccia Utente	newResponseTextBox	L'utente potrà personalizzare la propria risposta
Pulsante	Interfaccia Utente	SubmitResponseButton	L'utente lo seleziona per confermare la risposta
SMS	Social	SMS1	Elabora l'SMS
TinyDB	Archiviazione	TinyDB1	Salva la risposta in un database
SintesiVocale	Multimedia	SintesiVocale1	Legge il testo dell'SMS
SensorePosizione	Sensori	SensorePosizione1	Indica dove è il telefono

- Quindi modificare le proprietà come segue:
 - PromptLabel: nel campo **Testo** inserire il testo "Il Messaggio seguente verrà inviato ad ogni SMS ricevuto:";
 - ResponseLabel: nel campo **Testo** inserire "In questo momento sono occupato. Risponderò appena possibile.", poi mettere il **Grassetto**;
 - newResponseTextBox: modificare il campo **Larghezza** selezionando "Riempi contenitore";
 - SubmitResponseButton: modificare il campo **Testo** inserendo "Modifica la Risposta".
- Al termine dovremmo avere una situazione simile alla seguente:



- La prima parte, quella di design, è terminata! Andiamo a modificare il comportamento degli oggetti che abbiamo inserito;
- Il nostro primo obiettivo è quello di rispondere all'SMS ricevuto; raggiungiamo quindi SMS1 e selezioniamo "per sempre quando SMS1.Messaggio ricevuto";
- Quindi inseriamo al suo interno:
 - "imposta SMS1.NumeroTelefonico a " e, lasciando il mouse per qualche secondo su "numero" apparirà un pop-up dove potrete selezionare "valore di numero" da collegare al comando precedente;
 - Dopo il numero telefonico inseriamo il messaggio inserendo "imposta SMS1.Messaggio a" e, in "ResponseLabel", andare ad inserire la funzione "ResponseLable.Testo";
 - Infine, da SMS1, trasciniamo il comando "esegui SMS1.InviaMessaggio";
- **Salviamo!**
- A questo punto, una volta ricevuto un SMS, la nostra app sarà in grado di rispondere al mittente con un nostro messaggio;
- Adesso vogliamo anche aggiungere al messaggio la nostra posizione;
- Prima di tutto "inizializziamo", cioè mettiamo un valore iniziale, una variabile che chiameremo "lastKnowLocation";





- Andiamo quindi su “Variabili” e trasciniamo nella nostra area di lavoro il comando “inizializza variabile globale lastKnownLocation con valore” al quale collegheremo un valore di testo “Sconosciuta”;

```
inizializza variabile globale lastKnownLocation con valore " Sconosciuta "
```

- Procediamo in modo che possa contenere la nostra posizione andando su **SensorePosizione1** e scegliendo “per sempre quando SensorePosizione1.PosizioneCambiata”;
- Da Variabili, inseriamo al suo interno il comando “porta global lastKnownLocation a valore”, attaccandoci “SensorePosizione1.IndirizzoAttuale” che troviamo tra le opzioni di SensorePosizione1;
- Al termine, il nostro nuovo blocco si presenterà così:

```
per sempre quando SensorePosizione1 .PosizioneCambiata
  latitudine  longitudine  altitudine  velocità
  esegui  porta global lastKnownLocation a valore SensorePosizione1 . IndirizzoAttuale
```

- Ottimo! Non ci resta che aggiungere il valore di lastKnownLocation all’SMS di risposta...ma prima...
- **Salviamo!**
- Riprendendo il blocco dove diciamo alla nostra app di inviare solo il testo e aggiungiamo anche la posizione;


PRIMA

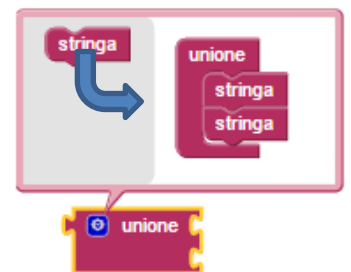
```
per sempre quando SMS1 .MessaggioRicevuto
  numero  testoMessaggio
  esegui  imposta SMS1 . NumeroTelefonico a valore di numero
          imposta SMS1 . Messaggio a ResponseLabel . Testo
          esegui SMS1 .InviaMessaggio
```

DOPO

```
per sempre quando SMS1 .MessaggioRicevuto
  numero  testoMessaggio
  esegui  imposta SMS1 . NumeroTelefonico a valore di numero
          imposta SMS1 . Messaggio a unione ResponseLabel . Testo
          " La mia ultima posizione è: "
          valore di global lastKnownLocation
          esegui SMS1 .InviaMessaggio
```



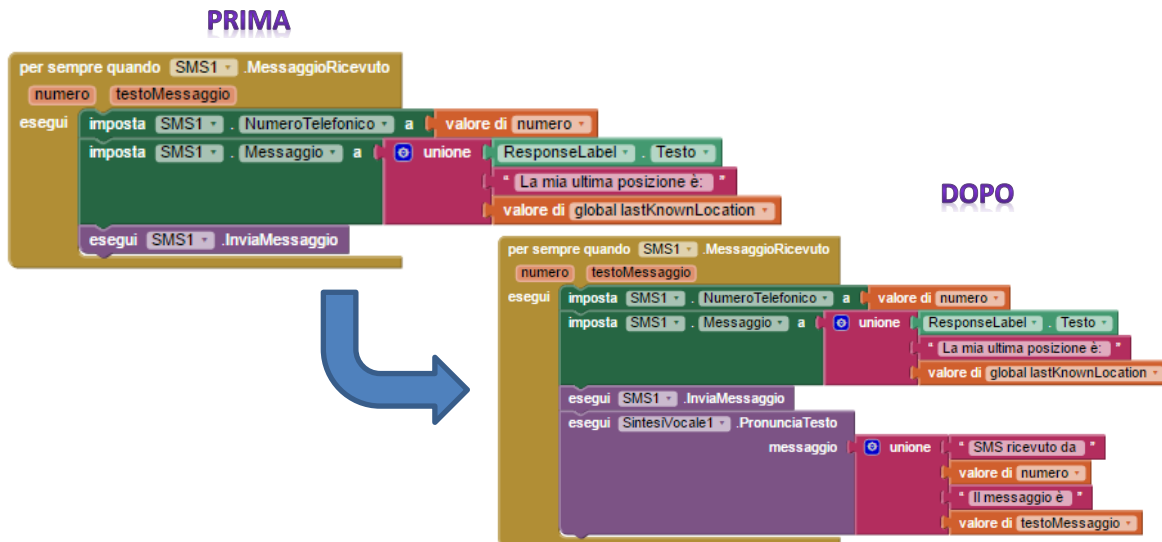
- Cosa abbiamo aggiunto?
- Prima di tutto il comando “unione” che troviamo all’interno di Testo;
- Attaccandolo a “imposta SMS1.Messaggio a “ vi accorgete vi permetterà di unire solo due testi, per permettere di aggiungere tre o più sarà sufficiente cliccare su  e trascinare “stringa” all’interno di unione quante volte si desidera, nel nostro caso occorrerà aggiungerne solo una in più;



- Nella prima posizione rimettiamo “ResponseLabel.Testo”;
- Al centro, da “Testo”, aggiungiamo un testo vuoto nel quale scriviamo “La mia ultima posizione è: “
- Come ultimo, da “Variabili”, inseriamo “valore di global lastKnownLocation”;
- **Salviamo!**
- Ora aggiungiamo la “Voce” alla nostra app, ovvero, ogni qual volta ci arriva un messaggio, oltre a rispondere con un altro messaggio vogliamo che ci venga letto il numero di chi lo ha spedito e il contenuto dell’sms stesso.



- Andremo quindi a modificare il blocco “per sempre quando SMS1.MessaggioRicevuto” come segue:



- Cosa è cambiato? In sostanza dopo “esegui SMS1.InviaMessaggio” abbiamo aggiunto un nuovo comando “esegui SintesiVocale1.PronunciaTesto” che trovate tra le opzioni disponibili in **SintesiVocale1**;
- Quindi abbiamo inserito il comando “unione” all’interno di **Testi**, impostandolo affinché potesse contenere 4 oggetti così come abbiamo fatto precedentemente con il messaggio SMS di risposta;
- Infine abbiamo messo alla prima posizione un oggetto “stringa di testo” (in **Testi**) con la frase “SMS ricevuto da”; alla seconda il “valore di numero”; alla terza un’altra “stringa di testo” con “il messaggio è”; infine il “valore di testoMessaggio”;
- Se non ricordate come recuperare i valori di “numero” e “testoMessaggio” andate a vedere alla pagina 2;
- Ricapitolando, in questo momento la nostra app, quando riceviamo un messaggio, è in grado di rispondere all’SMS con un altro che conterrà la nostra risposta e la nostra ultima posizione, inoltre saremo in grado di sentire il messaggio letto proprio dalla nostra app;
- **Salviamo!**
- Se volessimo personalizzare il messaggio di risposta? All’inizio avevamo impostato “In questo momento sono occupato. Risponderò appena possibile.”
- Per far sì che questo sia possibile bisogna salvare il nuovo messaggio da qualche parte così che, ogni volta che aprirò la mia app, il messaggio sia quello desiderato ed impostato l’ultima volta;
- Procediamo quindi inserendo sulla nostra area di lavoro, da **SubmitResponseButton**, “per sempre quando SubmitResponseButton.Cliccato”;
- Al suo interno spostiamo il contenuto di newResponseTextBox in ResponseLabel con il comando “imposta ResponseLabel.testo a” (in **ResponseLabel**) al quale attaccheremo “newResponseTextBox.testo” (in **newResponseTextBox**);
- Andiamo ora a cancellare il valore all’interno di **newResponseTextBox** aggiungendo “imposta newResponseTextBox.Testo a “ al quale verrà aggiunto, da Testo, il valore “vuoto” (primo elemento);
- Per fare in modo che il nuovo testo sia disponibile anche le volte successive continuiamo utilizzando “esegui TinyDB1.MemorizzaValore” (**che si trova in TinyDB1**);
- Ad “etichetta” inseriamo un Testo “responseMessage”, mentre in valoreDaMemorizzare “ResponseLabel.Testo”



- Il nuovo blocco di codice dovrebbe essere come sotto:

```

per sempre quando SubmitResponseButton .Cliccato
esegui
  imposta ResponseLabel . Testo a newResponseTextBox . Testo
  imposta newResponseTextBox . Testo a " "
  esegui TinyDB1 . MemorizzaValore
    etichetta "responseMessage"
    valoreDaMemorizzare ResponseLabel . Testo
  
```

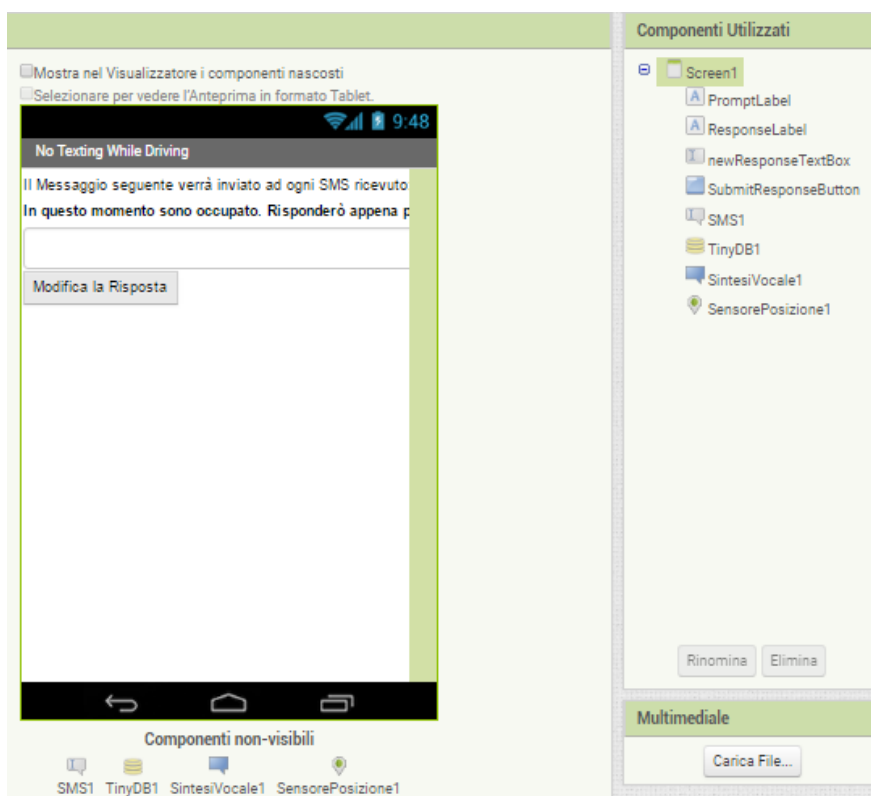
- **Salviamo!**
- Ultimo, ma non meno importante, occorre che all’avvio della nostra applicazione il messaggio che abbiamo personalizzato sia ricaricato, procediamo;
- Da **Screen1** trascinate “per sempre quando Screen1.Inizializza”;
- All’interno da **ResponseLabel** “imposta ResponseLabel.Testo a” e a seguire “esegui TinyDB1.OttieniValore” (all’interno di **TinyDB1**); in “etichetta” inserire “responseMessage” (ovvero l’etichetta del valore che abbiamo salvato nel blocco precedente), mentre in “valoreSeEtichettaNonPresente” imposteremo il valore di default “In questo momento non posso rispondere. Risponderò appena possibile.”;
- Il blocco completo sarà:

```

per sempre quando Screen1 . Inizializza
esegui
  imposta ResponseLabel . Testo a " "
  esegui TinyDB1 . OttieniValore
    etichetta "responseMessage"
    valoreSeEtichettaNonPresente "In questo momento non posso rispondere. Risponderò appena p
  
```

- **Salviamo un’ultima volta!**
- Finito, non resta che compilarlo e poi scaricarlo sul nostro cellulare

Per riepilogare, nella sezione Designer dovremmo avere una situazione molto simile alla seguente:





Mentre nella sezione Blocchi

The screenshot shows four event-driven code blocks in AppInventor2:

- per sempre quando SMS1 - .MessaggioRicevuto**:
 - inizializza variabile globale lastKnownLocation con valore " Sconosciuta "
 - esegui:
 - imposta SMS1 - . NumeroTelefonico - a valore di numero -
 - imposta SMS1 - . Messaggio - a unione:
 - ResponseLabel - . Testo -
 - " La mia ultima posizione è: "
 - valore di global lastKnownLocation -
 - esegui SMS1 - .InviaMessaggio
 - esegui SintesiVocale1 - .PronunciaTesto:
 - messaggio unione:
 - " SMS ricevuto da "
 - valore di numero -
 - " Il messaggio è "
 - valore di testoMessaggio -
- per sempre quando SensorePosizione1 - .PosizioneCambiata**:
 - latitudine longitudine altitudine velocità
 - esegui: porta global lastKnownLocation a valore SensorePosizione1 - . IndirizzoAttuale -
- per sempre quando Screen1 - .Inizializza**:
 - esegui:
 - imposta ResponseLabel - . Testo - a
 - esegui TinyDB1 - .OttieniValore:
 - etichetta " responseMessage "
 - valoreSeEtichettaNonPresente " In questo momento non posso rispondere. Risponderò appena p...
- per sempre quando SubmitResponseButton - .Cliccato**:
 - esegui:
 - imposta ResponseLabel - . Testo - a newResponseTextBox - . Testo -
 - imposta newResponseTextBox - . Testo - a " "
 - esegui TinyDB1 - .MemorizzaValore:
 - etichetta " responseMessage "
 - valoreDaMemorizzare ResponseLabel - . Testo -

...BE COOL!!!!