

Je parle les langues

App Inventor

Version 1

1

Hello World!

你好，世界！

¡Hola Mundo!



Talking in Languages 2.0 | by zinjixmaggir

DESCRIPTION DE L'APPLICATION

Cette application permet de dicter une phrase à son téléphone. Il va ensuite la traduire dans une autre langue puis lire la traduction.

Nous allons réaliser plusieurs versions

1. reconnaître la parole, traduire et lire, avec un bouton par étape
2. sélectionner la langue traduite dans une liste
3. enchaîner les opérations avec un seul bouton
4. permettre un dialogue avec le choix des deux langues et la traduction dans un sens ou l'autre selon que le est téléphone vertical ou horizontal.

Nota : à la date de rédaction de cette fiche, l'application fonctionne sans clef d'API Yandex pour la traduction, mais ça peut changer.

RÉALISATION D'UNE APPLICATION DE TRADUCTION (V1)

Que doit faire cette application ?

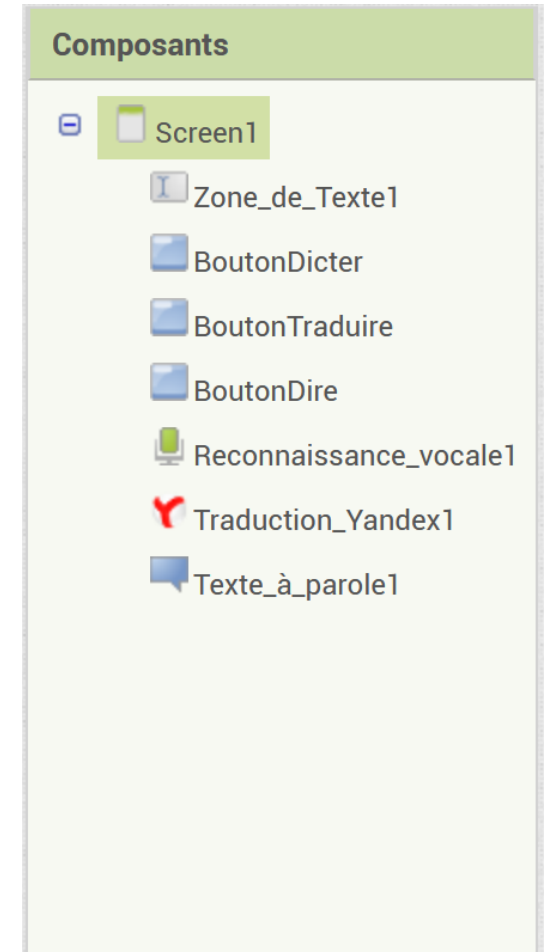
1. Ecouter une phrase et la convertir en texte
2. Traduire dans une autre langue
3. Lire ou dire la traduction



DESIGN : CHOIX DES COMPOSANTS

Que doit faire cette application ?

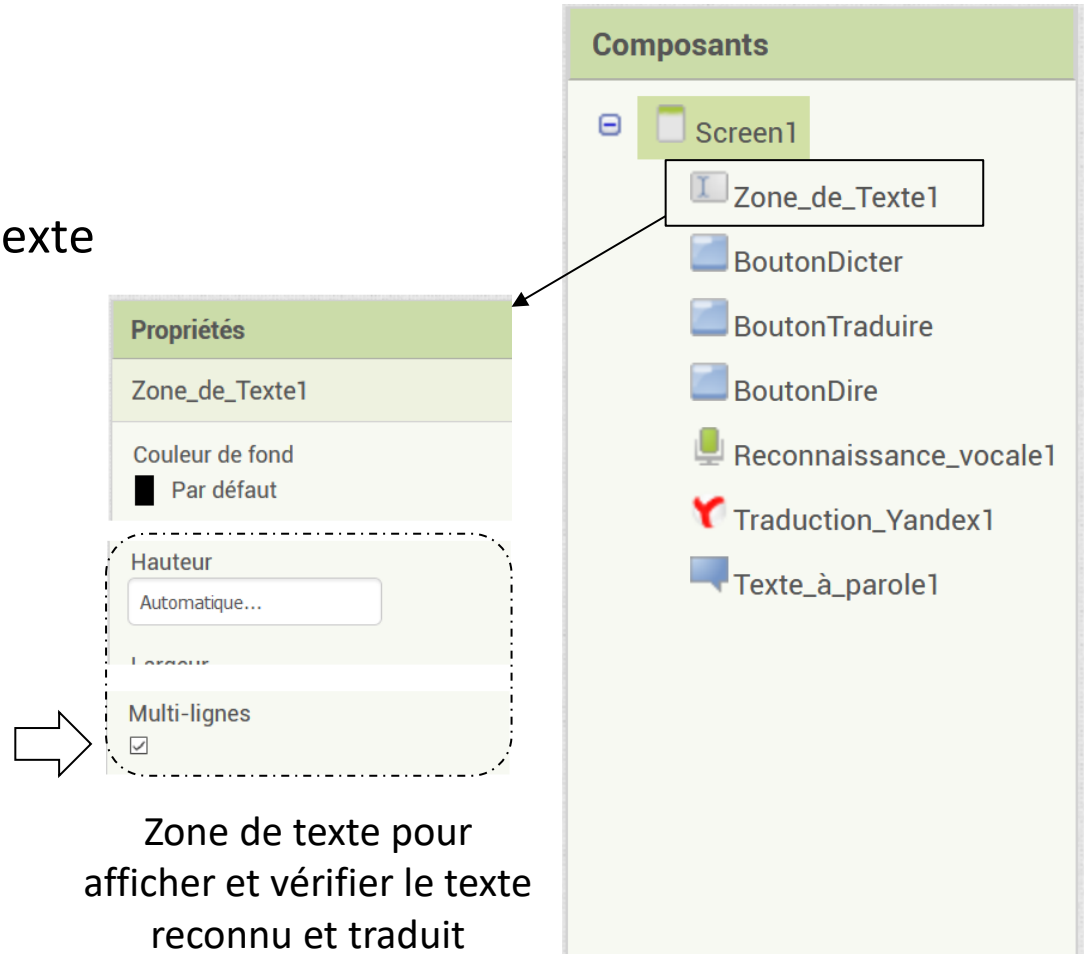
1. Ecouter une phrase et la convertir en texte
2. Traduire dans une autre langue
3. Lire ou dire la traduction



DESIGN : CHOIX DES COMPOSANTS

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte
2. Traduire dans une autre langue
3. Lire ou dire la traduction



DESIGN : 1 BOUTON POUR CHACUNE DES ÉTAPES

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

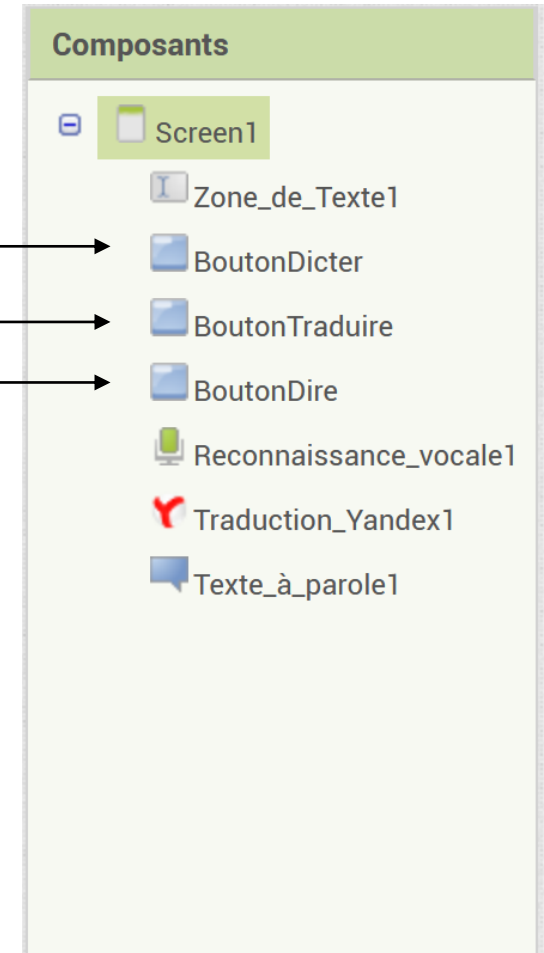
boutonDicter

2. Traduire dans une autre langue

boutonTraduire

3. Lire ou dire la traduction

boutonDire



DESIGN : COMPOSANTS DE RECONNAISSANCE, TRADUCTION, LECTURE

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

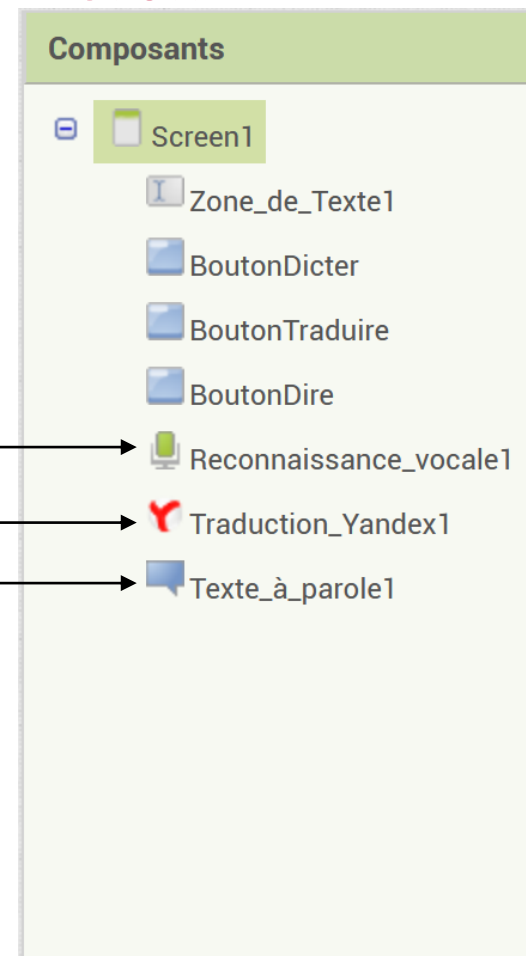
boutonDicter →  Reconnaissance vocale

2. Traduire dans une autre langue

boutonTraduire →  Traduction Yandex

3. Lire ou dire la traduction

boutonDire →  Texte à parole



DESIGN : COMPOSANTS DE RECONNAISSANCE, TRADUCTION, LECTURE

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

boutonDicter →  Reconnaissance vocale

2. Traduire dans une autre langue

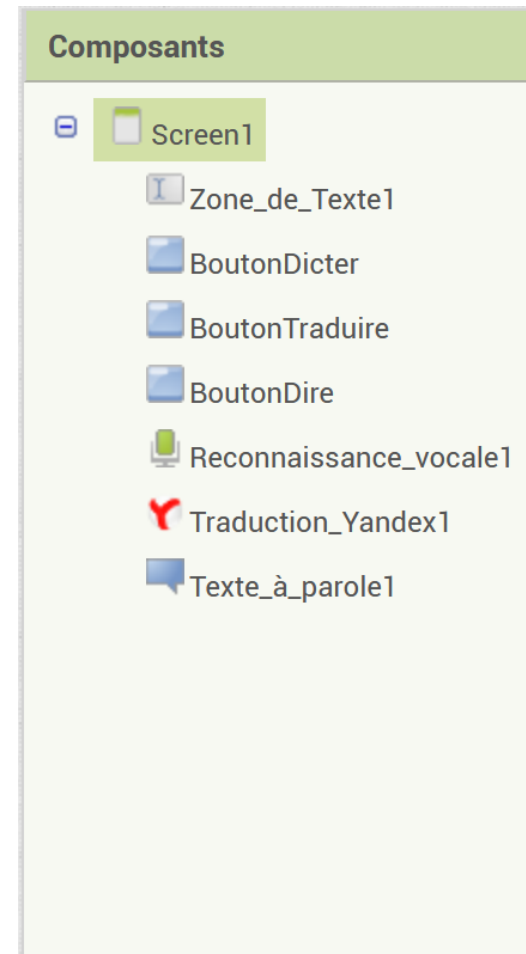
boutonTraduire →  Traduction Yandex



Yandex peut dans l'avenir demander une clef d'API

3. Lire ou dire la traduction

boutonDire →  Texte à parole



PROGRAMMATION 1 : RECONNAITRE LA PHRASE DICTÉE

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

boutonDicter →  Reconnaissance vocale

PROGRAMMATION 1 : RECONNAITRE LA PHRASE DICTÉE

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

boutonDicter →  Reconnaissance vocale



PROGRAMMATION 1 : RECONNAITRE LA PHRASE DICTÉE

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

boutonDicter →  Reconnaissance vocale



PROGRAMMATION 2 : TRADUIRE

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

boutonDicter →  Reconnaissance vocale

2. Traduire dans une autre langue

boutonTraduire →

 Traduction Yandex

```
quand BoutonDicter .Clic
faire appeler Reconnaissance_vocale1 .Obtenir texte
```

```
quand Reconnaissance_vocale1 .Après obtention texte
résultat partial
faire mettre Zone_de_Texte1 . Texte à obtenir résultat
```


PROGRAMMATION 2 : TRADUIRE

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

boutonDicter →  Reconnaissance vocale

2. Traduire dans une autre langue

boutonTraduire →  Traduction Yandex

```
quand BoutonDicter .Clic
faire appeler Reconnaissance_vocale1 .Obtenir texte
```

```
quand Reconnaissance_vocale1 .Après obtention texte
résultat partial
faire mettre Zone_de_Texte1 . Texte à obtenir résultat
```

```
quand BoutonTraduire .Clic
faire appeler Traduction_Yandex1 .Demander traduction
Langue à traduire à " en "
Texte à traduire Zone_de_Texte1 . Texte
```

PROGRAMMATION 2 : TRADUIRE

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

boutonDicter →  Reconnaissance vocale

2. Traduire dans une autre langue

boutonTraduire →  Traduction Yandex

```
quand BoutonDicter .Clic
faire appeler Reconnaissance_vocale1 .Obtenir texte
```

```
quand Reconnaissance_vocale1 .Après obtention texte
résultat partial
faire mettre Zone_de_Texte1 . Texte à obtenir résultat
```

```
quand BoutonTraduire .Clic
faire appeler Traduction_Yandex1 .Demander traduction
Langue à traduire à " en "
Texte à traduire Zone_de_Texte1 . Texte
```

```
quand Traduction_Yandex1 .Traduction reçue
Code réponse traduction
faire mettre Zone_de_Texte1 . Texte à obtenir traduction
```


PROGRAMMATION 2 : TRADUIRE

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

boutonDicter →  Reconnaissance vocale

2. Traduire dans une autre langue

boutonTraduire →  Traduction Yandex

```
quand BoutonDicter .Clic
faire appeler Reconnaissance_vocale1 .Obtenir texte
```

```
quand Reconnaissance_vocale1 .Après obtention texte
  résultat partial
faire mettre Zone_de_Texte1 . Texte à obtenir résultat
```

```
quand BoutonTraduire .Clic
faire appeler Traduction_Yandex1 .Demander traduction
  Langue à traduire à " en "
  Texte à traduire Zone_de_Texte1 . Texte
```

```
quand Traduction_Yandex1 .Traduction reçue
  Code réponse traduction
faire mettre Zone_de_Texte1 . Texte à obtenir traduction
```



Yandex renvoie une erreur (2202) si le code de la langue n'est pas reconnu, ou si le service n'est pas disponible. Yandex peut/pourra nécessiter une clé d'API

```
mettre Traduction_Yandex1 . Clé Api à
```

PROGRAMMATION 3 : LIRE LA TRADUCTION

Que doit faire cette application ?

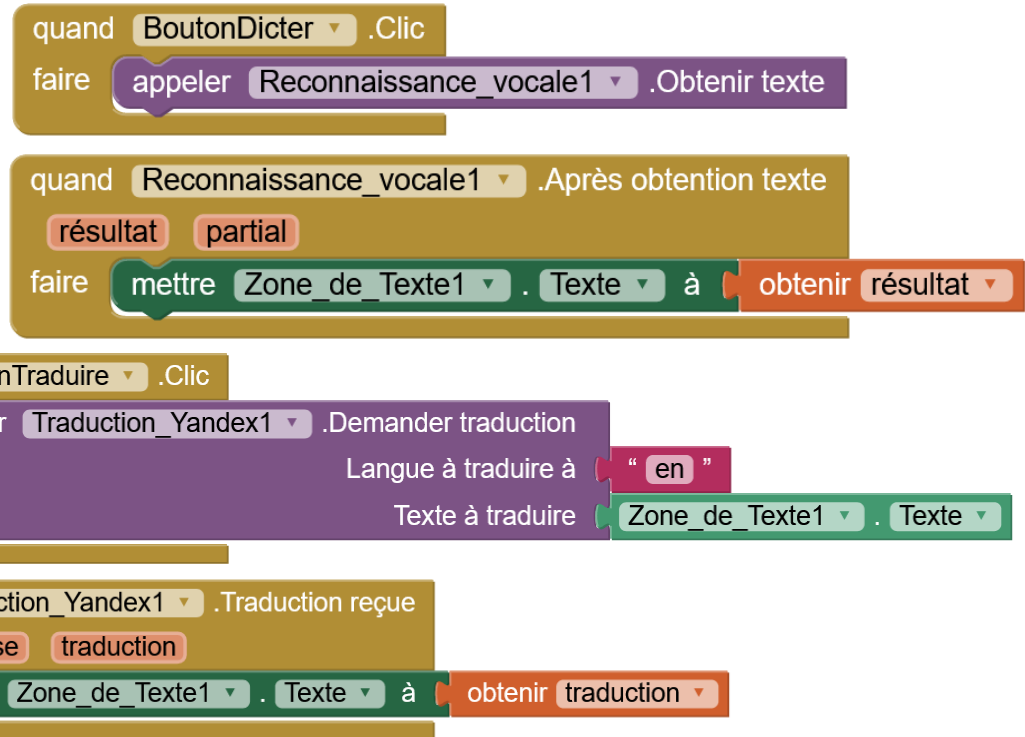
1. Ecouter une phrase et la convertir en texte

boutonDicter →  Reconnaissance vocale

2. Traduire dans une autre langue

boutonTraduire →  Traduction Yandex

3. Lire ou dire la traduction



PROGRAMMATION 3 : LIRE LA TRADUCTION

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

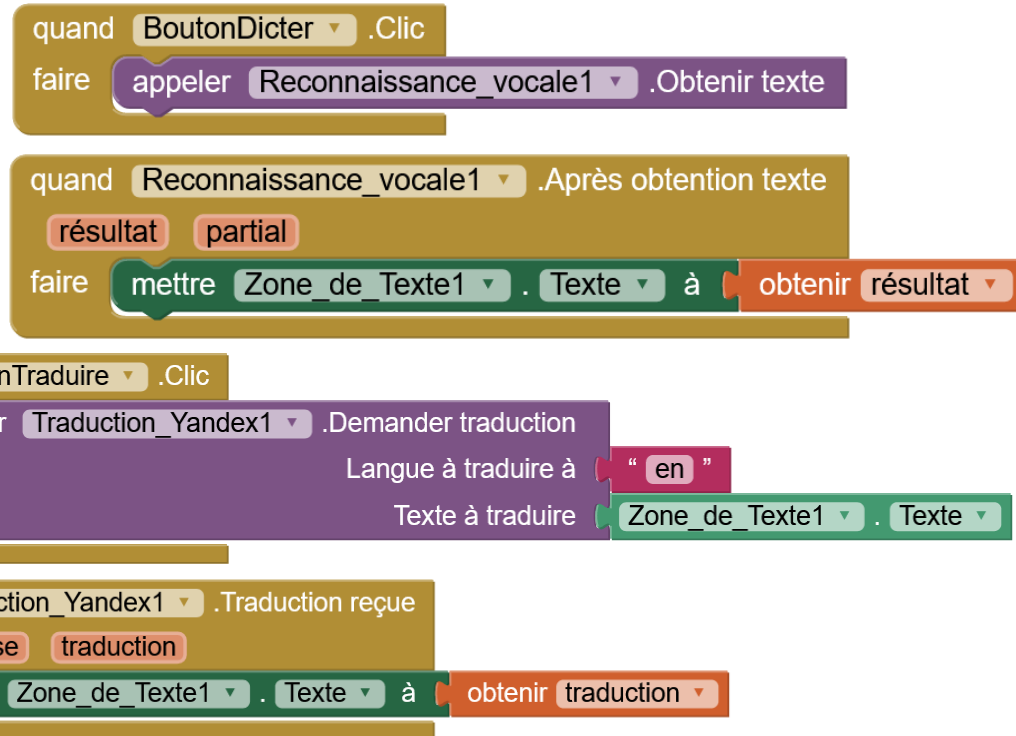
boutonDicter →  Reconnaissance vocale

2. Traduire dans une autre langue

boutonTraduire →  Traduction Yandex

3. Lire ou dire la traduction

boutonDire →  Texte à parole



PROGRAMMATION 3 : LIRE LA TRADUCTION

Que doit faire cette application ?

1. Ecouter une phrase et la convertir en texte

boutonDicter →  Reconnaissance vocale

2. Traduire dans une autre langue

boutonTraduire →  Traduction Yandex

3. Lire ou dire la traduction

boutonDire →  Texte à parole

```
quand BoutonDicter .Clic
faire appeler Reconnaissance_vocale1 .Obtenir texte
```

```
quand Reconnaissance_vocale1 .Après obtention texte
  résultat partial
faire mettre Zone_de_Texte1 . Texte à obtenir résultat
```

```
quand BoutonTraduire .Clic
faire appeler Traduction_Yandex1 .Demander traduction
  Langue à traduire à " en "
  Texte à traduire Zone_de_Texte1 . Texte
```

```
quand Traduction_Yandex1 .Traduction reçue
  Code réponse traduction
faire mettre Zone_de_Texte1 . Texte à obtenir traduction
```

```
quand BoutonDire .Clic
faire mettre Texte_à_parole1 . Langue à " en "
  appeler Texte_à_parole1 .Parler
  message Zone_de_Texte1 . Texte
```

Je parle les langues

App Inventor

Choix de la langue

2

Hello World!

你好，世界！

¡Hola Mundo!



Talking in Languages 2.0 | by zinjixmaggir

OBJECTIFS DE CETTE ÉTAPE

Dans cette étape, on va proposer à l'utilisateur de choisir la langue dans laquelle la phrase sera traduite.

Pour ça, on va

- utiliser un "spinner" ou "curseur animé" qui va proposer une liste de codes de langue : en,fr,it, ...
- dans laquelle l'utilisateur va sélectionner une langue
- le code de langue sélectionné sera utilisé
 - en paramètre de la traduction avec le composant Yandex
 - comme propriété de la lecture avec le composant texte à parole

CHOIX DE LA LANGUE DANS UNE LISTE : DESIGN

- Ajout d'un composant "spinner" ou "curseur animé"

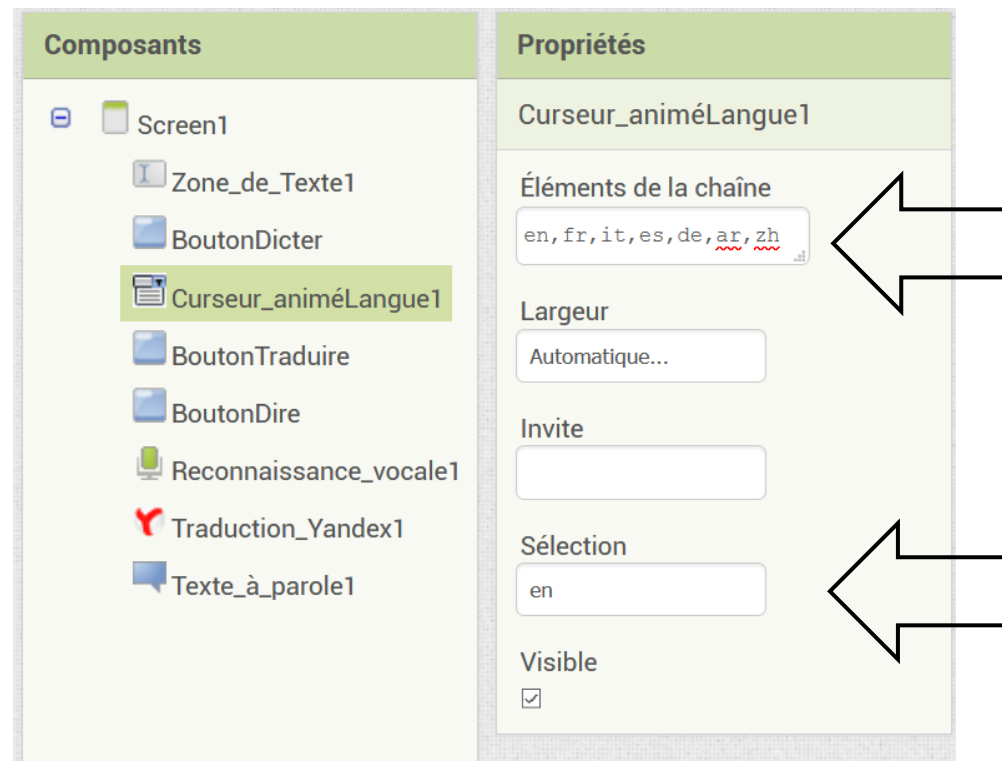
curseur_animéLange1

The screenshot displays the Android Studio IDE with the following components:

- Palette (Interface utilisateur):** Lists various UI components. The "Curseur animé" component is highlighted in green.
- Interface:** Shows a mobile app preview on a smartphone. The app has a text input field labeled "texte" and a spinner button labeled "Ajouter élém". Below the spinner are buttons for "Dicter", "Traduire", and "Dire".
- Composants:** Lists the components used in the interface. "Curseur_animéLange1" is highlighted in green.
- Propriétés:** Shows the properties for the selected "Curseur_animéLange1" component. The "Éléments de la chaîne" property is set to "en, fr, it, es, de, ru, ar, zh".

CHOIX DE LA LANGUE DANS UNE LISTE : DESIGN

- Ajout d'un composant "spinner" ou "curseur animé"
- et indiquer dans les propriétés:
 - une liste de codes de langues dans "Éléments de la chaîne" (séparés par des virgules et sans espace)
 - Le choix de la langue par défaut dans "sélection"



Codes de langue sur 2 caractères ou plus :

Pour la traduction yandex : <https://yandex.com/dev/translate/doc/dg/concepts/api-overview.html>

Pour la reconnaissance vocale : <https://cloud.google.com/speech-to-text/docs/languages>

LISTE PARTIELLE DES CODES DE LANGUE DE TRADUCTION

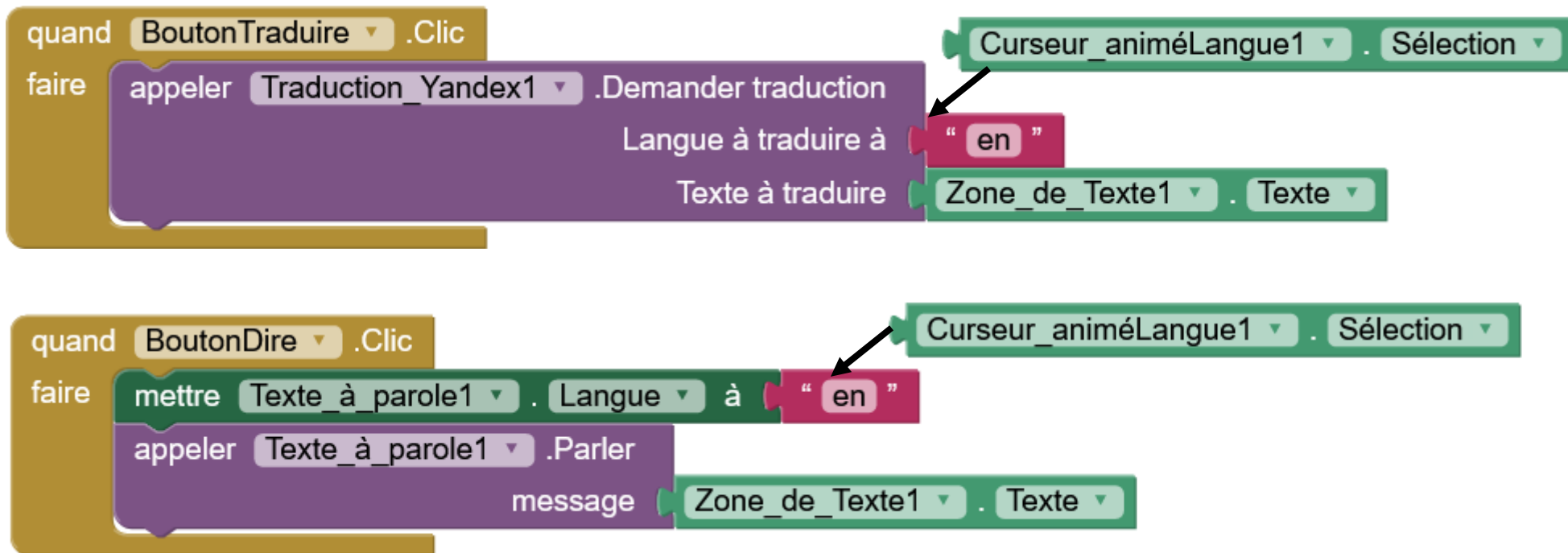
Le tableau suivant correspond à une partie des langues gérées par Yandex pour la traduction.

Tous ces codes ne sont pas nécessairement pris en compte par la reconnaissance de la parole et text to speech ([vérifier la doc et tester](#))

langue	Code	langue	Code	langue	Code	langue	Code
Azerbaijani	az	Georgian	ka	Malagasy	mg	Swahili	sw
Albanian	sq	Danish	da	Malayalam	ml	Sundanese	su
Amharic	am	Hebrew	he	Maltese	mt	Tajik	tg
English	en	Yiddish	yi	Macedonian	mk	Thai	th
Arabic	ar	Indonesian	id	Maori	mi	Tagalog	tl
Armenian	hy	Irish	ga	Mongolian	mn	Tamil	ta
Afrikaans	af	Italian	it	German	de	Tartar	tt
Basque	eu	Icelandic	is	Nepalese	ne	Telugu	te
Bashkir	ba	Spanish	es	Norwegian	no	Turkish	tr
Belarusian	be	Kazakh	kk	Punjabi	pa	Uzbek	uz
Bengal	bn	Kannada	kn	Papiamento	pap	Ukrainian	uk
Burmese	my	Catalan	ca	Persian	fa	Urdu	ur
Bulgarian	bg	Kirghiz	ky	Polish	pl	Finnish	fi
Bosnian	bs	Chinese	zh	Portuguese	pt	French	fr
Welsh	cy	Korean	ko	Romanian	ro	Hindi	hi
Hungarian	hu	Xhosa	xh	Russian	ru	Croatian	hr
Vietnamese	vi	Khmer	km	Cebuano	ceb	Czech	cs
Haitian	ht	Laotian	lo	Serbian	sr	Swedish	sv
Galician	gl	Latin	la	Sinhalese	si	Scottish	gd
Dutch	nl	Latvian	lv	Slovak	sk	Estonian	et
Greek	el	Luxembourg	lb	Slovenian	sl	Japanese	ja

CHOIX DE LA LANGUE DANS UNE LISTE : BLOCS

- Remplacement du code de langue par celui correspondant à la sélection du curseur animé (spinner)



CHOIX DE LA LANGUE DANS UNE LISTE : BLOCS

- Remplacement du code de langue par celui correspondant à la sélection du curseur animé (spinner)

```
quand BoutonTraduire .Clic
faire
  appeler Traduction_Yandex1 .Demander traduction
    Langue à traduire à Curseur_animéLangue1 . Sélection
    Texte à traduire Zone_de_Texte1 . Texte
```

```
quand BoutonDire .Clic
faire
  mettre Texte_à_parole1 . Langue à Curseur_animéLangue1 . Sélection
  appeler Texte_à_parole1 .Parler
    message Zone_de_Texte1 . Texte
```

Je parle les langues

App Inventor

Automatisation

3

Hello World!

你好，世界！

¡Hola Mundo!



Talking in Languages 2.0 | by zinjixmaggir

OBJECTIFS DE CETTE ÉTAPE

Dans cette étape, on va enchaîner automatiquement les opérations .

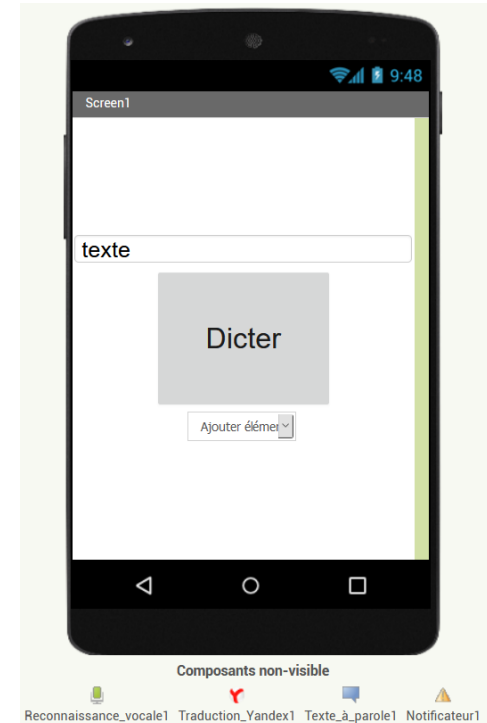
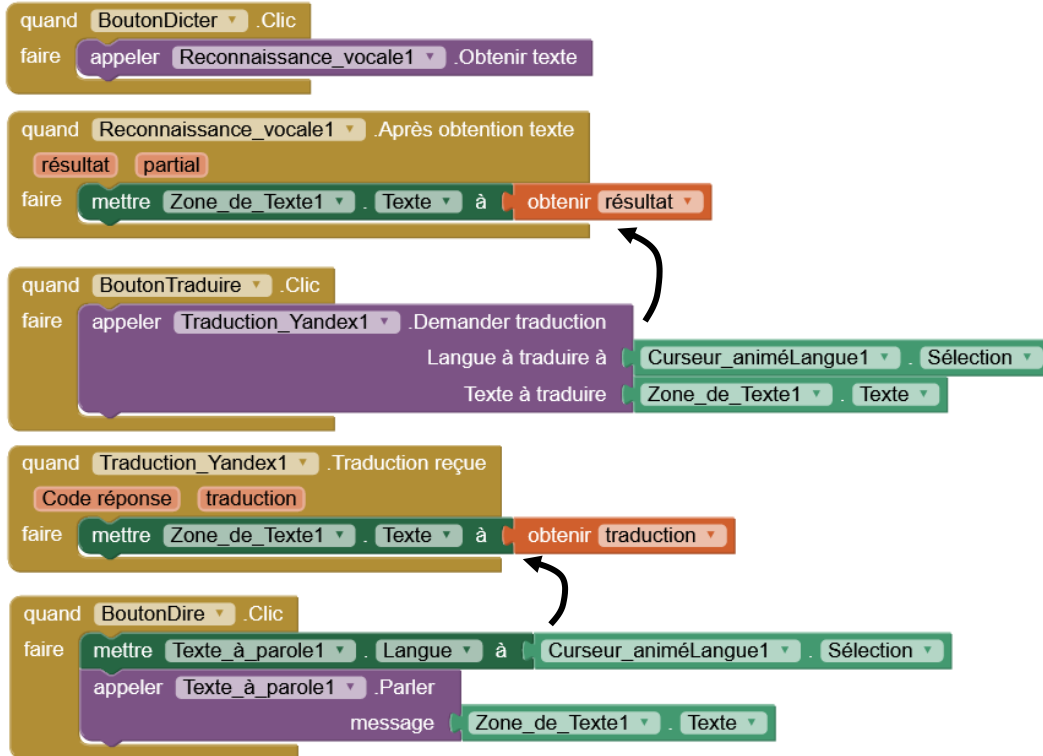
Dès que l'utilisateur a appuyé sur le bouton "Dicter"

- la reconnaissance vocale est appelée (comme avant)
- lorsque le texte est reconnu, la traduction est appelée (sans attendre)
- lorsque le texte est traduit, la lecture est appelée (sans attendre)

Les boutons traduire et dire sont devenus inutiles et sont supprimés

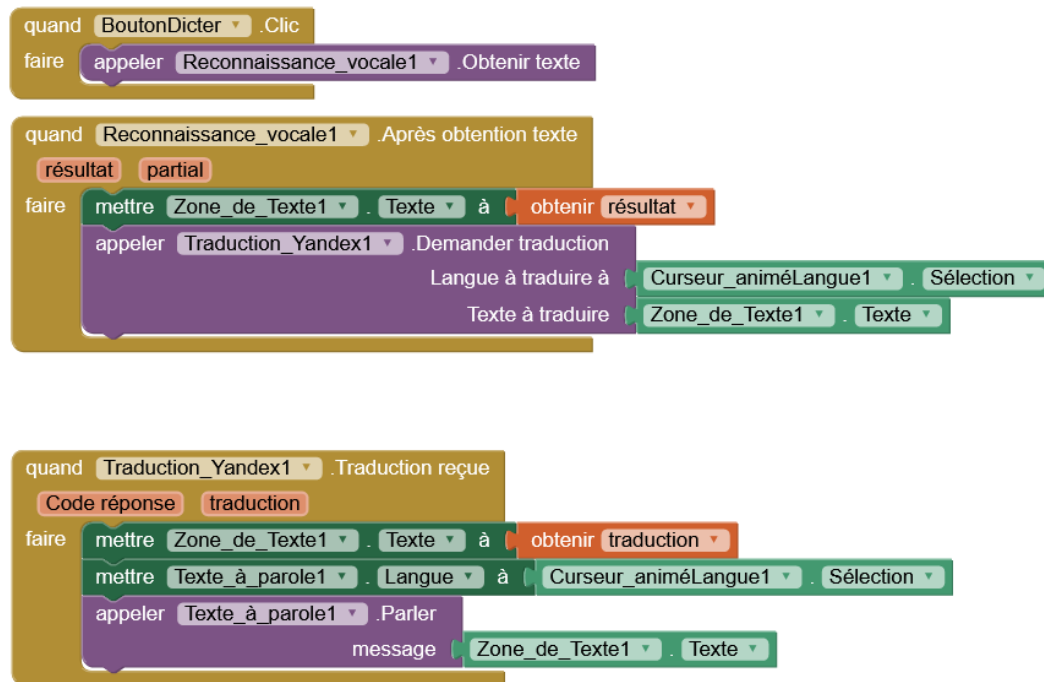
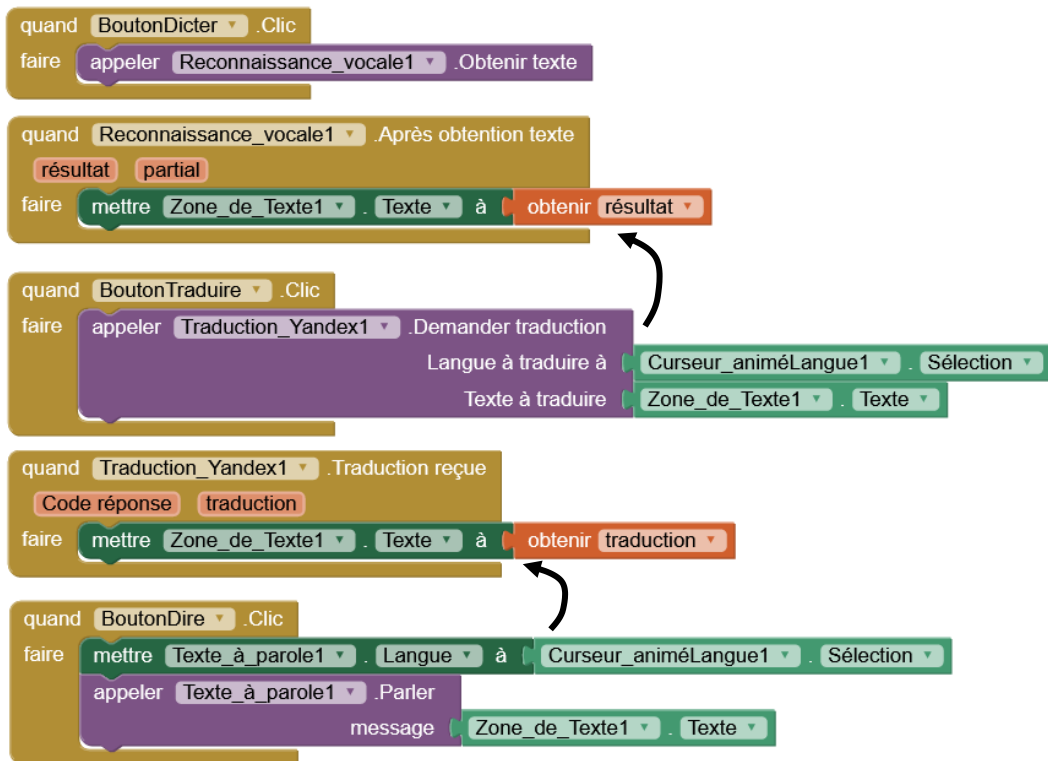
ENCHAINEMENT AUTOMATIQUE DES TACHES

Appel de la traduction à réception de la reconnaissance et du texte à parole à réception de la traduction



ENCHAINEMENT AUTOMATIQUE DES TACHES

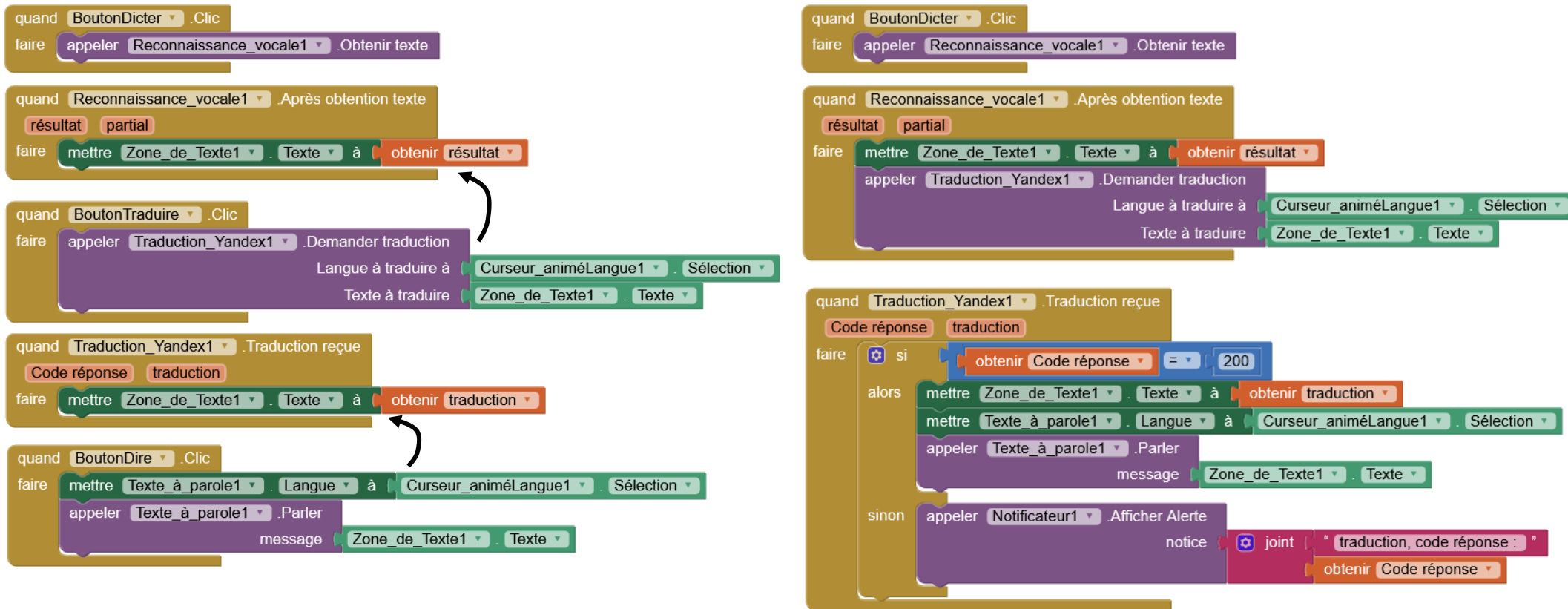
Appel de la traduction à réception de la reconnaissance et du texte à parole à réception de la traduction



On peut ensuite supprimer les boutons inutiles et vérifier le code réponse Yandex

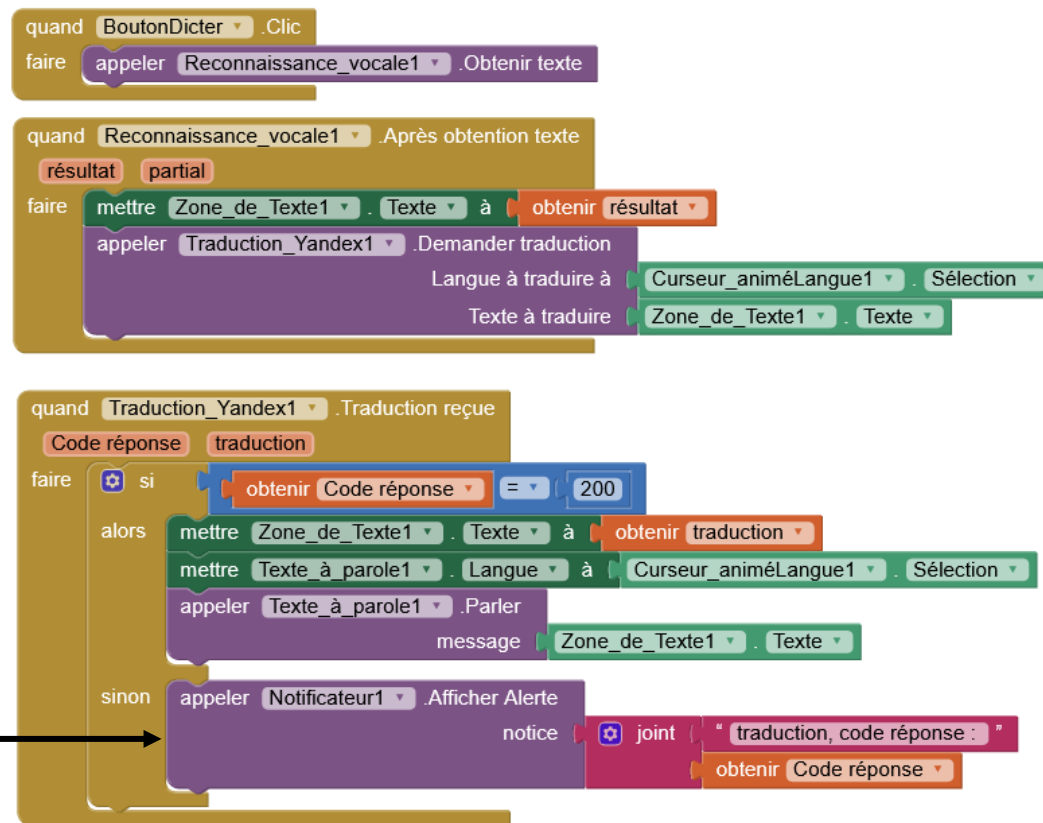
ENCHAINEMENT AUTOMATIQUE DES TACHES

Appel de la traduction à réception de la reconnaissance et du texte à parole à réception de la traduction



ENCHAINEMENT AUTOMATIQUE DES TACHES

Appel de la traduction à réception de la reconnaissance et du texte à parole à réception de la traduction



Ajouter un composant "notificateur"
et afficher un message si le code
réponse de la traduction n'est pas égal à
200 (réponse OK)



Je parle les langues

App Inventor

Choix de la langue à traduire

4

Hello World!

你好，世界！

¡Hola Mundo!



Talking in Languages 2.0 | by zinjixmaggir

OBJECTIFS DE CETTE ÉTAPE

Dans cette étape, on va choisir la langue de départ, celle qui est dictée

- on choisit avec un nouveau curseur animé : curseurUtilisateur
- problème : le composant "speech to text" ne propose pas ce choix :
 - la langue est par défaut celle du smartphone
 - et la détection automatique de la langue ne marche pas bien
- Solution : utiliser un composant proposé en extension à App Inventor
 - le speech recognizer de Taifun / puravida
 - ce composant permet le choix de la langue à reconnaître



- on va importer ce composant et l'utiliser à la place du précédent

NOUVEAU CURSEUR ANIMÉ POUR CHOISIR LA LANGUE DICTÉE

- ajouter un curseur animé " Curseur_utilisateur" pour la langue de l'utilisateur avec les mêmes paramètres
- renommer le 1° curseur en "Curseur_interlocuteur« pour la langue traduite
- et intégrer ces deux curseurs dans un arrangement horizontal



EXTENSION : SPEECH RECOGNIZER DE TAIFUN / PURAVIDA

Cette extension comprend les mêmes blocs que SpeechToText et ajoute le choix de la langue à reconnaître



<https://downloads.sourceforge.net/project/puravidaapps/com.puravidaapps.TaifunSpeechRecognizer.aix>

ou

<https://onvaessayer.org/appinventor?res=SpeechRecognizer>

Ce bloc évite la reconnaissance automatique de la langue qui ne marche pas toujours très bien



App Inventor Extensions

Free to install + Easy to use + Over 84,000 happy users + Rating 4.7 out of 5.

INSTALL



Speech Recognizer Extension

See the [App Inventor Extensions document](#) about how to use an App Inventor Extension.

For questions about this extension or bug reports please start a new thread in the [App Inventor Extensions](#) forum. Thank you.

For feature requests [please contact me by email](#). To be a sponsor of a new method already is possible starting from only 10 USD! With your contribution you will help the complete App Inventor community. Thank you.

Nov 5th, 2016: Version 1: initial version

July 18th, 2019: Version 2: Fixing the Supported Languages method for newer Android versions

Description

Speech Recognizer extension.

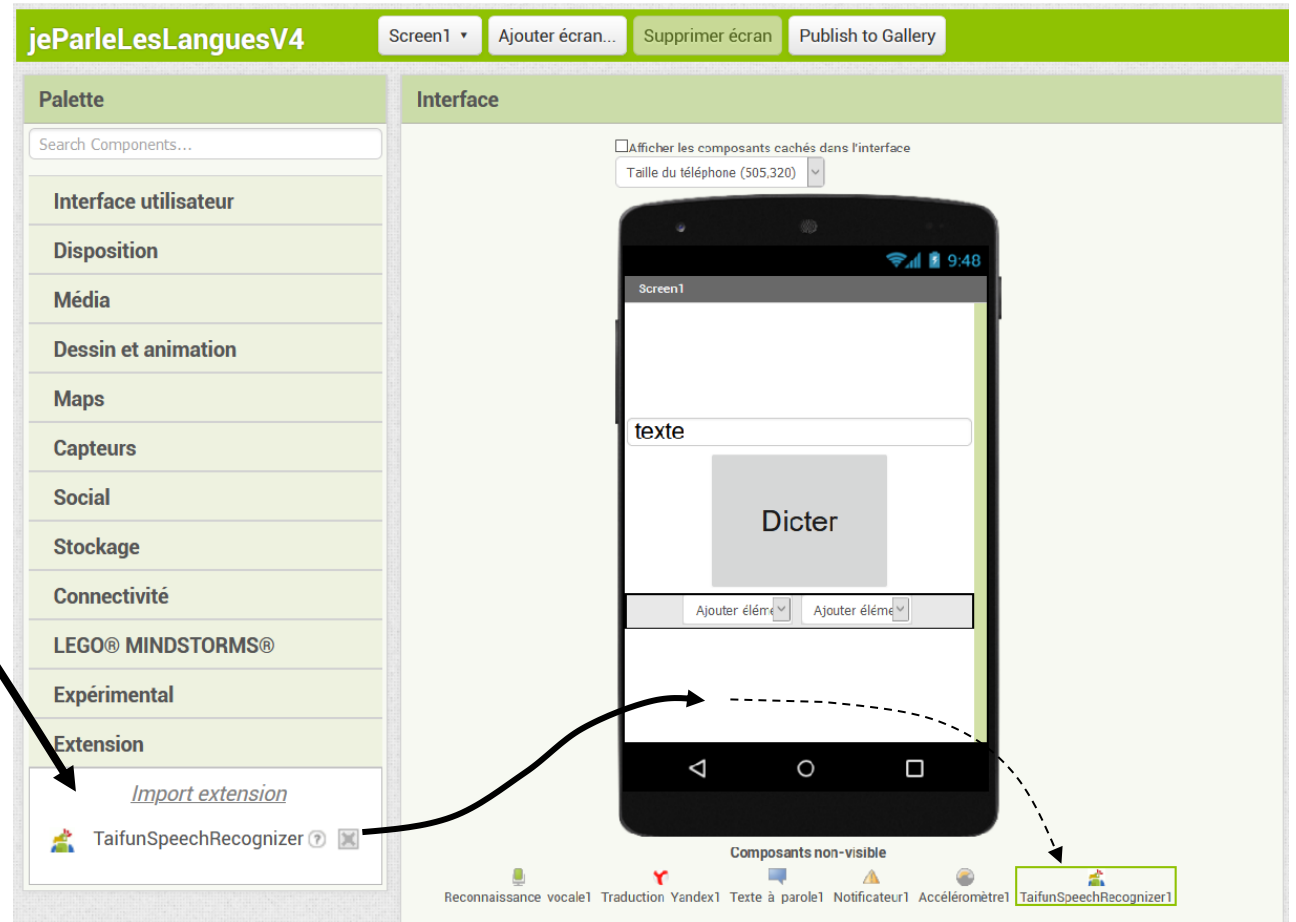
Required permissions: android.permission.INTERNET

Thank you Olibres for being the sponsor of this extension.

Note: This extension offers the functionality of the [Speech Recognizer component](#). Additionally the following blocks have been added:

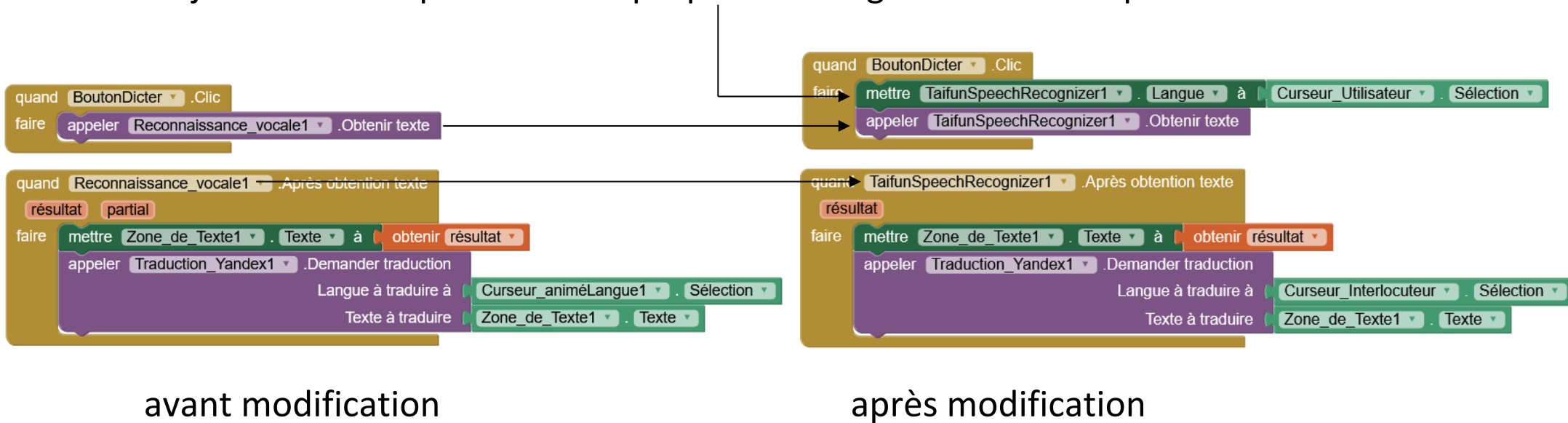
EXTENSION : SPEECH RECOGNIZER DE TAIFUN / PURAVIDA

- Importer le composant dans la catégorie des extensions
- Puis l'ajouter dans l'interface



REEMPLACER LES APPELS AU COMPOSANT SPEECH TO TEXT

- Au lieu d'utiliser les évènements et fonctions du composant de reconnaissance vocale initial, on utilise le composant : **TaifunSpeechRecognizer**
- Et on ajoute un bloc qui modifie la propriété « langue » de ce composant



Je parle les langues

App Inventor

Dialogue A/R

5

Hello World!

你好，世界！

¡Hola Mundo!



Talking in Languages 2.0 | by zinjixmaggir

OBJECTIFS DE CETTE ÉTAPE

Dans cette étape, on va permettre un dialogue :

- si l'utilisateur parle, il tient le téléphone vertical, la phrase est alors traduite dans la langue de l'autre
- si l'autre parle, le téléphone est tendu vers lui à l'horizontal, la phrase est traduite dans la langue de l'utilisateur

On va utiliser

- l'accéléromètre pour détecter la position du téléphone
- un autre composant (extension) pour pouvoir choisir la langue dictée
- un 2° curseur animé pour choisir la 2° langue

DIALOGUE ENTRE DEUX PERSONNES

Pour gérer un dialogue entre deux personnes on doit

1. définir les 2 langues :

- celle de celui qui parle :
- celle de celui qui écoute la traduction :

initialise global langueIN à " fr "

initialise global langueOUT à " en "

DIALOGUE ENTRE DEUX PERSONNES

Pour gérer un dialogue entre deux personnes on doit

1. définir les 2 langues :

- celle de celui qui parle :
- celle de celui qui écoute la traduction :

initialise global langueIN à " fr "

initialise global langueOUT à " en "

2. choisir la langue de l'utilisateur et celle de son interlocuteur

- langue de l'utilisateur (nouveau) :
- langue de l'interlocuteur :

Curseur_Utilisateur . Sélection

Curseur_Interlocuteur . Sélection

DIALOGUE ENTRE DEUX PERSONNES

Pour gérer un dialogue entre deux personnes on doit

1. définir les 2 langues :

- celle de celui qui parle :
- celle de celui qui écoute la traduction :

initialise global langueIN à " fr "

initialise global langueOUT à " en "

2. choisir la langue de l'utilisateur et celle de son interlocuteur

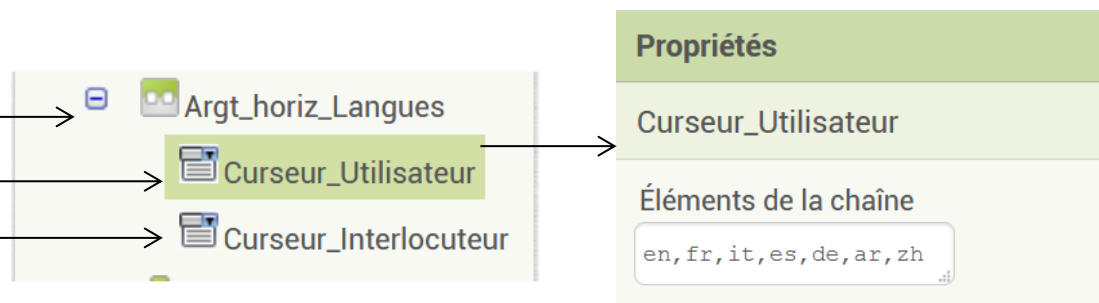
- langue de l'utilisateur (nouveau) :
- langue de l'interlocuteur :

Curseur_Utilisateur . Sélection

Curseur_Interlocuteur . Sélection

Dans le design,

- ajout d'un arrangement horizontal
- ajout d'un curseur animé utilisateur
- renommer le 1° curseur



DIALOGUE ENTRE DEUX PERSONNES


Pour gérer un dialogue entre deux personnes on doit

1. définir les 2 langues :

- celle de celui qui parle :
- celle de celui qui écoute la traduction :



initialise global langueIN à " fr "



initialise global langueOUT à " en "

2. choisir la langue de l'utilisateur et celle de son interlocuteur

- langue de l'utilisateur (nouveau) :
- langue de l'interlocuteur :



Curseur_Utilisateur . Sélection



Curseur_Interlocuteur . Sélection

3. savoir qui parle et qui écoute la traduction :

DIALOGUE ENTRE DEUX PERSONNES

Pour gérer un dialogue entre deux personnes on doit

1. définir les 2 langues :

- celle de celui qui parle :
- celle de celui qui écoute la traduction :



initialise global langueIN à " fr "



initialise global langueOUT à " en "

2. choisir la langue de l'utilisateur et celle de son interlocuteur

- langue de l'utilisateur (nouveau) :
- langue de l'interlocuteur :



Curseur_Utilisateur Sélection



Curseur_Interlocuteur Sélection

3. déterminer qui parle et qui écoute la traduction :

- traduire l'utilisateur si il tient le téléphone vertical devant lui,
- traduire l'autre si il tend le téléphone à l'horizontale vers son interlocuteur.

L'accéléromètre permet de distinguer ces 2 cas.



obtenir zAccel < 6

DESIGN : AJOUTER UN ACCÉLÉROMÈTRE

The screenshot displays the 'jeParleLesLanguesV5' design tool interface. The top bar includes 'Screen1', 'Publish to Gallery', 'Designer', and 'Blocs'. The interface is divided into four main sections:

- Palette:** A sidebar on the left containing a search bar and a list of components under the 'Capteurs' (Sensors) category. The 'Accéléromètre' (Accelerometer) component is highlighted in green.
- Interfa:** The central workspace showing a mobile phone mockup. The screen contains a text input field with 'texte', a 'Dicter' button, and two 'Ajouter élém...' buttons. A checkbox 'Afficher les composants cachés dans l'interface' is checked, and a dropdown menu shows 'Taille du téléphone (505,320)'. At the bottom, a 'Composants non-visible' (Hidden Components) bar lists various elements, with 'Accéléromètre1' highlighted in a green box.
- Composants:** A tree view on the right showing the hierarchy of components on the screen, including 'Screen1', 'Zone_de_Texte1', 'BoutonDicter', 'Argt_horiz_Langues', 'Curseur_Utilisateur', 'Curseur_Interlocuteur', 'Traduction_Yandex1', 'Texte_à_parole1', 'Notificateur1', 'TaifunSpeechRecognizer1', and 'Accéléromètre1'.
- Propriétés:** A panel on the right showing the properties for 'Accéléromètre1', including 'Activé' (checked), 'LegacyMode' (unchecked), 'Intervalle Minimum' (400), and 'Sensibilité' (modéré).

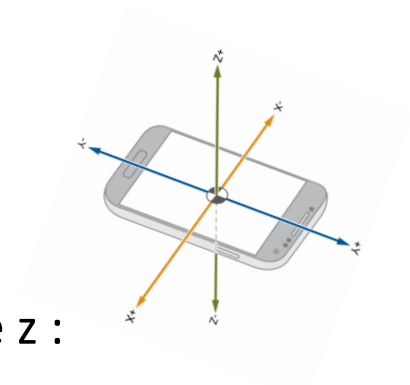
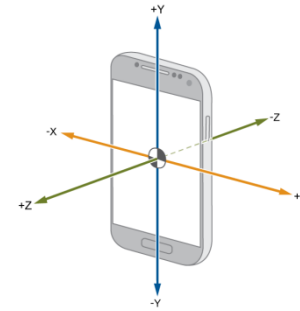
UTILISATION D'UN ACCÉLÉROMÈTRE

- Faites des essais sur son fonctionnement : afficher les mesures

```
quand Accéléromètre1 . Accélération changée
  faire
    mettre Zone_de_Texte1 . Texte à
      joint ( " acc X = "
            Accéléromètre1 . XAccel
            joint ( " , Y = "
                  Accéléromètre1 . YAccel
                  joint ( " , Z = "
                        Accéléromètre1 . ZAccel
```

- l'accéléromètre mesure (surtout) la gravité : 9,81

	X Accel	Y Accel	Z Accel
Smartphone vertical	~ 0	~ 9,8	~ 0
Smartphone à plat	~ 0	~ 0	~ 9,8



- L'orientation du smartphone peut être déduite de la valeur lue sur l'axe z : si $zAccel > 6$ le smartphone est vertical, sinon il est horizontal

CHOIX DE LA LANGUE IN ET OUT EN FONCTION DE L'ACCÉLÉROMÈTRE

Quand accélération change

si z accel < 6 (vertical)

langueIN = langue utilisateur

langueOUT = langue interlocuteur

écran vert

sinon (horizontal)

langueIN = langue interlocuteur

langueOUT = langue utilisateur

écran bleu

```
initialise global langueIN à " fr "
initialise global langueOUT à " en "

quand Accéléromètre1 .Accélération changée
  xAccel yAccel zAccel
  faire
    mettre Zone_de_Texte1 . Te...
    si
      obtenir zAccel < 6
      alors
        mettre global langueIN à Curseur_Utilisateur . Sélection
        mettre global langueOUT à Curseur_Interlocuteur . Sélection
        mettre Screen1 . Couleur de fond à [vert]
      sinon
        mettre global langueIN à Curseur_Interlocuteur . Sélection
        mettre global langueOUT à Curseur_Utilisateur . Sélection
        mettre Screen1 . Couleur de fond à [bleu]
```

PRISE EN COMPTE DES LANGUES CHOISIES

```
quand BoutonDicter .Clic
faire
  mettre TaifunSpeechRecognizer1 . Langue à Curseur_Utilisateur . Sélection
  appeler TaifunSpeechRecognizer1 .Obtenir texte
```

```
quand TaifunSpeechRecognizer1 .Après obtention texte
résultat
faire
  mettre Zone_de_Texte1 . Texte à obtenir résultat
  appeler Traduction_Yandex1 .Demander traduction
    Langue à traduire à Curseur_Interlocuteur . Sélection
    Texte à traduire Zone_de_Texte1 . Texte
  obtenir global langueOUT
```

```
quand Traduction_Yandex1 .Traduction reçue
Code réponse traduction
faire
  si obtenir Code réponse = 200
  alors
    mettre Zone_de_Texte1 . Texte à obtenir traduction
    mettre Texte_à_parole1 . Langue à Curseur_Interlocuteur . Sélection
    appeler Texte_à_parole1 .Parler
      message Zone_de_Texte1 . Texte
  sinon
    appeler Notificateur1 .Afficher Alerte
      notice joint " traduction, code réponse : "
      obtenir Code réponse
```


PRISE EN COMPTE DES LANGUES CHOISIES

```
quand BoutonDicter .Clic
faire
  mettre TaifunSpeechRecognizer1 . Langue à Curseur_Utilisateur . Sélection
  appeler TaifunSpeechRecognizer1 .Obtenir texte
```

```
quand TaifunSpeechRecognizer1 .Après obtention texte
résultat
faire
  mettre Zone_de_Texte1 . Texte à obtenir résultat
  appeler Traduction_Yandex1 .Demander traduction
    Langue à traduire à Curseur_Interlocuteur . Sélection
    Texte à traduire Zone_de_Texte1 . Texte
```

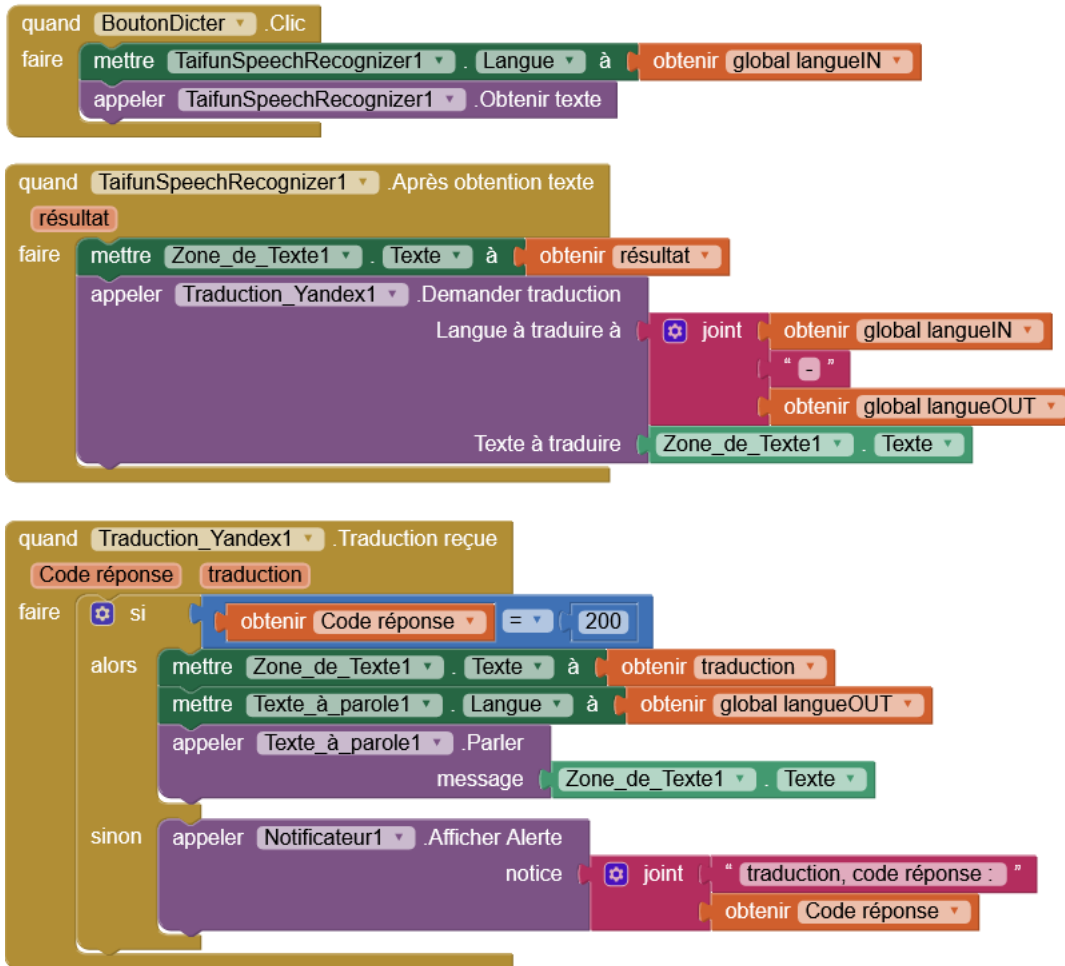
```
quand Traduction_Yandex1 .Traduction reçue
Code réponse traduction
faire
  si obtenir Code réponse = 200
  alors
    mettre Zone_de_Texte1 . Texte à obtenir traduction
    mettre Texte_à_parole1 . Langue à Curseur_Interlocuteur . Sélection
    appeler Texte_à_parole1 .Parler
      message Zone_de_Texte1 . Texte
  sinon
    appeler Notificateur1 .Afficher Alerte
      notice joint " traduction, code réponse : "
      obtenir Code réponse
```

Dans la traduction, on peut indiquer la langue en entrée et en sortie, séparées par "-"

```
joint
  obtenir global langueIN
  " - "
  obtenir global langueOUT
```

```
obtenir global langueOUT
```

PRISE EN COMPTE DES LANGUES CHOISIES



RÉSULTAT FINAL

```
quand BoutonDicter .Clic
faire
  mettre TaifunSpeechRecognizer1 . Langue à obtenir global langueIN
  appeler TaifunSpeechRecognizer1 .Obtenir texte

quand TaifunSpeechRecognizer1 .Après obtention texte
  résultat
  faire
    mettre Zone_de_Texte1 . Texte à obtenir résultat
    appeler Traduction_Yandex1 .Demander traduction
      Langue à traduire à joint obtenir global langueIN
      Texte à traduire Zone_de_Texte1 . Texte
      obtenir global langueOUT

quand Traduction_Yandex1 .Traduction reçue
  Code réponse traduction
  faire
    si obtenir Code réponse = 200
    alors
      mettre Zone_de_Texte1 . Texte à obtenir traduction
      mettre Texte_à_parole1 . Langue à obtenir global langueOUT
      appeler Texte_à_parole1 .Parler
        message Zone_de_Texte1 . Texte
    sinon
      appeler Notificateur1 .Afficher Alerte
        notice joint " traduction, code réponse : "
        obtenir Code réponse
```

```
initialise global langueIN à " fr "
initialise global langueOUT à " en "

quand Accéléromètre1 .Accélération changée
  xAccel yAccel zAccel
  faire
    mettre Zone_de_Texte1 . Te...
    si obtenir zAccel < 6
    alors
      mettre global langueIN à Curseur_Utilisateur . Sélection
      mettre global langueOUT à Curseur_Interlocuteur . Sélection
      mettre Screen1 . Couleur de fond à [vert]
    sinon
      mettre global langueIN à Curseur_Interlocuteur . Sélection
      mettre global langueOUT à Curseur_Utilisateur . Sélection
      mettre Screen1 . Couleur de fond à [bleu]
```