<u>Aperçu</u>

Développements d'un questionnaire basé sur des questions définies par le programmateur. Les questions pourront dans une première phase être de 2 types. Soit à choix multiple soit avec réponse courte. Le nombre de questions est variable et au choix elles peuvent contenir ou non une image qui illustre la question.

Objectifs d'apprentissages

Ce tutoriel utilise en plus des composants usuels (Label, boutons, image,...) :

- Les listes afin de stocker les questions
- La visibilité des arrangements afin d'afficher les questions correctement
- Le notificateur

Etapes de réalisations

Pour réaliser cette application, place et configure les éléments suivants comme affiché.



Composant	Groupe	Nom	Utilité	Action
Arrangement	Disposition	VArrQuestion		Règle la largeur sur
Vertical				« Remplir parent » et
				l'alignement
				horizontal sur centré
Label	Interface utilisateur	LblQuestion	Afficher la question	Règle la largeur sur
				« Remplir parent »
Image	Interface utilisateur	ImgQuestion	Au besoin ajouter	
			une image à la	
			question	

Arrangement Vertical	Disposition	VArrChoixMultiple	Règle la largeur sur « Remplir parent » et l'alignement horizontal sur centré
Bouton	Interface utilisateur	BtnReponse1	
Bouton	Interface utilisateur	BtnReponse2	
Bouton	Interface utilisateur	BtnReponse3	
Bouton	Interface utilisateur	BtnReponse4	
Arrangement Vertical	Disposition	VArrRepondre	Règle la largeur sur « Remplir parent » et l'alignement horizontal sur centré
Zone de texte	Interface utilisateur	TxtReponse	
Bouton	Interface utilisateur	BtnRepondre	Affiche le texte : « Répondre »
Notificateur	Interface utilisateur	Notificateur1	

Maintenant que tu as tous les éléments nécessaires, passons à la programmation. Tout d'abord il faut stocker les questions. Chaque question est composée d'une liste comme ceci :

t	créer une liste	C	" (La question) "
		C	* Soit le nom d'une image qui illustre lq question, soit un texte vide) *
		C	* (La réponse correcte) *
		Ľ	* Au cas ou il s'agit d'une question à choix multiple, une réponse alternative fausse, sinon texte vide)*
		C	* Au cas ou il s'agit d'une question à choix multiple, une réponse alternative fausse, sinon texte vide 🔭
		C	* Au cas ou il s'agit d'une question à choix multiple, une réponse alternative fausse, sinon texte vide)*

Comme tu as plusieurs questions, il s'agit d'une liste de liste. Crée la structure suivante en y adaptant les questions de ton choix.



Veuille à ce que les noms d'images utilisé dans tes questions soient intégrés à ton projet.

Au besoin ici les images utilisées pour cet exemple: Gates.jp	g:
https://de.wikipedia.org/wiki/Datei:Bill_Gates_July_2014.jp	g

googleIMG.jpg : <u>http://www.atoute.org/n/local/cache-</u> vignettes/L549xH335/brinpage2001sepia-f7bff.jpg

Médi	a	
9	oogleIMG.jpg	
G	ates.jpg	
	Charger fichier	

Il te faut également une variable afin de savoir quelle question est actuellement affichée. Initialise la comme ci-dessous:

Pour accéder à une question tu as besoin des blocs suivants :



N'oublie pas, que chaque question est une liste composée de 6 éléments. Afin de répartir les différentes informations, crée une procédure que tu nommes AfficherQuestionSuivante.

à Affic	cher	QuestionSuivante					
re me	ettre	LblQuestion * .	(Texte 🔹 à 🕻		choisir liste élément	liste (,	🕻 choisir liste élément 🛛 liste 🖡 obtenir (global Questions 🕥
							index 🍃 obtenir [global Nr_Question_Actuel 🔻]
						index ()	
me	ettre	ImgQuestion 🔹	(Image 🔹 à	C	choisir liste élément	liste	e 🗘 choisir liste élément 🛛 liste 🏌 obtenir (global Questions 🔹
							index 👂 obtenir [global Nr_Question_Actuel *
							x (2
me	ettre	BtnReponse1 •	. Texte 🔹 à	C	choisir liste élément	liste	e 📙 choisir liste élément 🛛 liste 🍃 obtenir 👩 lobal Questions 🕥
							index 👂 obtenir global Nr_Question_Actuel 🔻
							x () [3]
me	ettre	BtnReponse2 *	. (Texte 🔹 à	C	choisir liste élément	liste	e 🕻 choisir liste élément 🛛 liste 🖡 obtenir global Questions 💌
							index 👂 obtenir global Nr_Question_Actuel 💌
							x () (4)
me	ettre	BtnReponse3 *	. (Texte 🔹 à	C	choisir liste élément	liste	e 🕻 choisir liste élément 🛛 liste 🍃 obtenir Global Questions 🕥
							index 👂 obtenir global Nr_Question_Actuel 🔹
							x () 5
me	ettre	BtnReponse4 *	. (Texte 🔹 à	C	choisir liste élément	liste	e 🜔 choisir liste élément 👘 liste 🔰 obtenir 👩 lobal Questions 💌
							index 🖡 obtenir global Nr_Question_Actuel 🔹
							x (6

Ajoute le bloc suivant et faits un premier essai en connectant ton téléphone à l'aide de l'ai2 Companion.



Pourquoi nous faut il la condition si ?

Il faut à présent ajouter quelques blocs afin de choisir quelle question est affiché. Afin d'afficher les questions dans l'ordre, ajoute à ta procédure les blocs suivants :



Comme nous avons différent type de questions (A choix multiple et à réponse courte), tu dois adapter ton affichage au type de question. Modifie à cette fin ta procédure comme suite :

💿 à	Afficher	QuestionSuivante						
faire	mettre	global Nr_Question_Actu	el 🔹 à 🕻 🖸 🖕	obtenir global Nr_Que	estion_Actuel * +			
	🧿 si	obtenir (global Nr	_Question_Actuel	🕥 🏊 🖒 taille de la	aliste liste 🖡 obter	nir (global Questio	ns 🔹	
		mettre global Nr_Questi	on_Actuel 🔹 à 🕻	1				
	mettre	LblQuestion). (Texte)	🗋 à 🚺 choisir lis	te élément liste ("	choisir liste élémen	t liste i obteni	r global Questions 🔹	
						index 🕻 obteni	r global Nr_Question_	Actuel 🔹
				index (1			_
	mettre	ImgQuestion • . Image	💌 à 🖡 choisir	liste élément liste	Choisir liste éléme	ent liste 🕻 obte	nir global Questions •	2
					2	index obte	enir global Nr_Question	n_Actuel *
	o si	C 😧 estvide 🖒 cour	oer I choisir liste	e élément liste L c	hoisir liste élément	liste 🕻 obtenir	global Questions 🔹	
						index 🕻 obtenir	global Nr_Question_A	ctuel 🕥
				index 🗘 🖪	3			
		mettre VArrChoixMultipl	e 🔹 . Visible 🔹	à l faux 🔹				
		mettre (VArrRepondre •). (Visible 🔪 à	(vrai •				
		mettre (VArrChoixMultipl	e 🔹 . (Visible 🔹	à I <mark>vrai v</mark>				
		mettre (VArrRepondre •). (Visible 🔹 à	faux				
		mettre (BtnReponse1 •). Texte 🔹 à 🕻	choisir liste élément	liste (C choisir lis	te élément liste	obtenir (global Que	Ougetion Actual -
					index (3	IIIdex		Question_Adder
		mettre (BtnReponse2 •). Texte 🔹 à 🕻	choisir liste élément	liste (choisir lis	te élément liste	🚦 obtenir (global Que	estions *
						index	obtenir (global Nr_	Question_Actuel *
					index 🔰 🚺			
		mettre (BtnReponse3 •). (Texte 🔹 à 🕻	choisir liste élément	liste (C choisir lis	te élément liste	obtenir global Que	estions •
					index / 6	Index	obtenir (global Nr_	Question_Actuel
		mettre (BtnReponse4 •). (Texte 🔻 à 🌘	choisir liste élément	liste choisir lis	te élément liste	obtenir global Que	estions 🔹
							obtenir global Nr_	Question_Actuel *
					index () 6			

De même chaque question ne contient pas une image, ajoute une condition si afin de n'afficher l'image que si nécessaire.

💿 si	🔹 🕻 est vide 🖞 couper 🕻 choisir liste élément 🛛 liste 📫 choisir liste élément 🖉 liste 👔 obtenir 🚺 liste 🚺 couper 🚺
	index global Nr_Question_Actuel >
	index 🚺 2
alors	mettre (ImgQuestion •). Visible •) à // faux •)
sinon	mettre (ImgQuestion v). (Visible v) à (vrai v)
	mettre (ImgQuestion •). (Image •) à 🖡 choisir liste élément 🛛 liste 🦕 choisir liste élément 🛛 liste 🖕 obtenir (global Questions •)
	index 🖕 obtenir (global Nr_Question_Actuel 🗠
	index 🗘 🔁

L'affichage des questions fonctionne à présent correctement. Connecte to téléphone et vérifie que tout fonctionne sans erreur.

Il te faut à présent ajouter une méthode de réaction aux différents boutons. Commence par le cas du choix multiple. Pour chacun des 4 boutons, programme la méthode de réaction suivante :

quand	BtnRe	ponse1 Slic
faire	🖸 si	comparer textes 🕻 EtnReponse 1 🔹 . Texte 🔹 💷 🕻 choisir liste élément 🛛 liste 🖞 choisir liste élément 🛛 liste 🖞 obtenir global Questions 🔹
		index 🗘 obtain global Nr_Question_Actual 🕤
		index 🖟 3
		appeler Notificateur1 . Afficher fenêtre message
		message (Réponse correcte, Bravo)
		Titre (* Exact *
		Texte bouton (Colored and Col
		appeler AfficherQuestionSuivante *
		appeler Notificateur1 • Afficher fenêtre message
		message 👔 🕻 C'est faux, réessaye 📩
		Titre 🖡 🖕 Titreur *
		Texte bouton 🕴 😶 🖸

Le cas des réponses courtes est un peu plus difficile. Crée la réaction du bouton mais afin d'éviter des différences de majuscules ou d'espaces, compare les textes sans espaces et sans majuscules. En plus, cache le clavier lorsque l'utilisateur à fini.

quand	BtnRe	tepondre 🕤 Clic
faire	🗿 si	comparer textes (couper (minuscule), TxReponse), Texte = 0 couper (minuscule) couper
		index 🖡 obtenir (global Nr_Question_Actuel
		index 🖬 🖬
		appeler (Notificateuri ••) Afficher fenêtre message
		message (Féponse correcte, Bravo)
		Titre (<mark>* Exact *)</mark>
		Texte bouton
		appeler Screent • Cacher clavier
		appeler AfficherQuestionSuivante •
		appeler Notificateuri 🖬 Afficher fenêtre message
		message / Cest faux, réessaye. Attention aux fautes d'orthographe
		Titre (Erreun *
		Texte bouton

Test ton application. Ci-dessous l'intégralité du code :



Fichiers

.aia: <u>https://drive.google.com/file/d/0B_b0KFDNNII_N1c2N3NIU205cTA/view?usp=sharing</u> .apk: <u>https://drive.google.com/file/d/0B_b0KFDNNII_SS1zaGpoM3R0aXc/view?usp=sharing</u>

Aller plus loin :

• Compte le nombre de fausse réponses afin d'afficher un score à l'utilisateur.

- Lors des questions à choix multiple, la bonne réponse est toujours affichée sur le 1^{er} bouton, comment pourrais-tu changer l'ordre d'affichage ?
- Ajoute un bouton « Donner sa langue au chat » Celui-ci affiche la réponse correcte dans le Notificateur et passe à la question suivante.