Il existe de nombreux logiciels de création d'applications permettant de produire du code sans en écrire une seule ligne (AuthorWare et Gazelle sont parmi les précurseurs, Mediator est encore d'actualité …).
Parmi les solutions d'assistance à la création d'applications pour smartphones ou tablettes, App Inventor, en version 2 actuellement, joue la carte de la construction d'interface par glisser-déposer et de la programmation par blocs.
Pour en savoir plus sur les origines de AI2, une adresse :
http://appinventor.mit.edu/explore/about-us.html

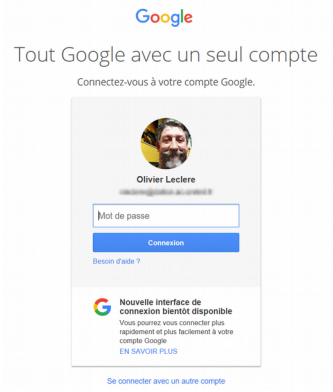
App Inventor 2 (AI2) est une application en ligne, nul besoin de téléchargement, permettant de construire des applications pour les dispositifs tournant sous Android (génération d'un fichier APK). Il permet un dévelopement visible, en temps réel, sur un dispositif sous Android connecté à l'ordinateur ou sur un émulateur. Pour ces 2 modes de fonctionnement, des modules seront à télécharger et installer sur l'ordinateur de création.
La « collaboration » MIT AI2 et Google exige d'avoir accés à un compte Google (à créer si besoin!). Et il semblerait que Chrome est conseillé comme navigateur à utiliser (perso je travaille plutôt avec FF).

Premiers pas

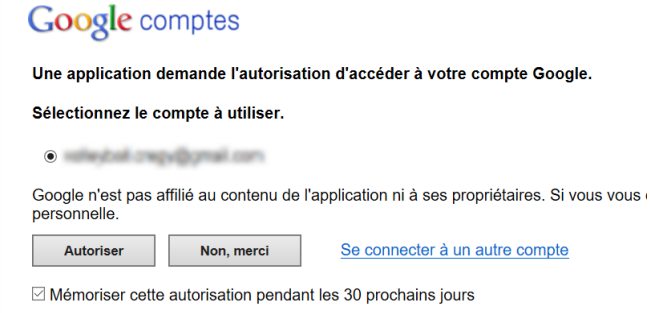
Vous disposez d'un smartphone, ou d'une tablette, sous Android (et de son cable de liaison USB), il est conseillé de télécharger App Inventor 2 Setup version macOs ou Windows sur l'ordianteur de travail, et l'application AI2 Companion présente sur le store Google.
La solution conseillée par l'éditeur, de transfert en temps réel par Wifi, a très peu de chance de fonctionner à la fac.
Pour le téléchargement : http://appinventor.mit.edu/explore/ai2/setup-device-usb.html
et les instructions d'installation, in english of course.

Vous n'avez pas de dispositif sous Android, il vous reste la possibilité de faire tourner l'émulateur, lui aussi en temps réel, pour voir et tester (attention toutefois, certaines instructions peuvent engendrer des erreurs ou des réponses inattendues, sur l'émulateur, mais seront fonctionnnelles sur les dispositifs. Le téléchargement de cet émulateur, et les instructions pour son installation sont disponibles à cette adresse : http://appinventor.mit.edu/explore/ai2/setup-emulator.html

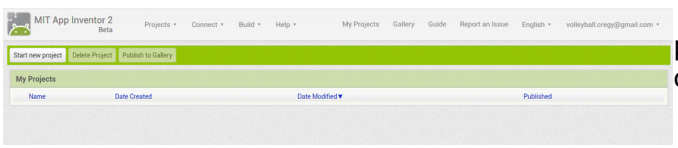
Maintenant que vous avez le moyen de « visualiser » votre travail, il est temps de passer aux choses sérieuses ! http://ai2.appinventor.mit.edu
Evidemment, il faut vous identifier à l'aide de votre compte Google !



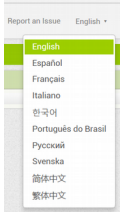
Et autoriser cette nouvelle application à utiliser votre compte :



Et bien sûr accepter les conditions d'utilisation … remplir, ou non, un questionnaire ...
Une fois ces tracasseries réglementaires réglées, il est enfin possible d'accéder pleinement à AI2 :



Encore un petit réglage : passer l'application en français (ou en coréen !)

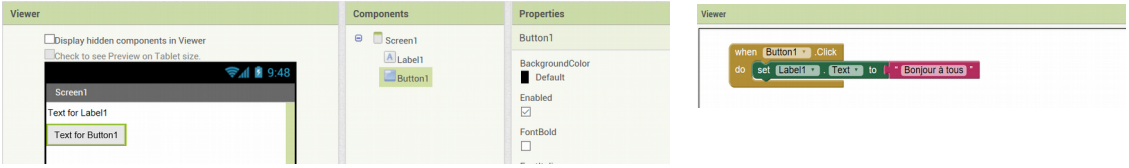


Premier projet

La terminologie d'AI2 fait état de projets pour les applications à créer. Donc première étape, création d'un nouveau projet. Pour faire simple, le premier projet s'appelle Appli1 …
L'interface est « assez » simple :

Le projet est livré avec un premier « écran » que l'on pourrait aussi comparé à une page, vierge, représenté dans la zone de travail, attention, il ne s'agit pas vraiment d'un aperçu (l'aperçu est disponible sur l'appareil connecté s'il est présent, ou dans l'émulateur).
La zone de gauche regroupe tous les éléments qu'il est possible d'utiliser, ils sont regroupés par type, et sont directement utilisables par un simple glisser-déposer.
Les propriétés de l'élément sélectionné dans la liste des éléments utilisés (ou déposés) sont modifiables via la zone correspondante.
L'accés au mode « programmation » se fait via le bouton Blocs, cette présentation étant celle du mode conception graphique, comme le montre la capture suivante :

Là aussi, la programmation se fait intégralement par glisser-déposer, des blocs disponibles dans le réservoir de gauche vers la zone de travail.
La zone des instructions contient en haut à gauche un « bloc-note » capable de retenir des blocs, des assemblages de blocs ou des jeux entiers de blocs pour une réutilisation rapide … ainsi qu'une corbeille pour les blocs jugés inutiles.
C'est parti pour un grand classique :
Glisser-déposer une « étiquette » ainsi qu'un bouton dans la zone de travail graphique …
Glisser-déposer un bloc « Quand le bouton est cliqué ... », puis un bloc « mettre le texte de l'étiquette à ... » et fermé l'assemblage avec le texte « Bonjour à tous ».



Utilisation de l'émulateur

L'émulateur installé (cf Premiers pas) il est accessible en lancant, depuis la barre de menu, la commande Connecte / Emulateur (AI starter doit être déjà lancé …).Lors du premier lancement, il faut patienter le temps de créer les différents éléments et de les charger dans la fenêtre, puis un pseudo smartphone apparaît, avec les éléments déposés dans la zone de travail, le bouton marqué Text for Button1 est cliquable, ce qui permet de modifier le texte de l'étiquette, de Text for Label1 en Bonjour à tous.
Tant que l'émulateur n'est pas fermé, ou réinitialisé, les modifications apportées à la zone de travail graphique ou de programmation sont « immédiatement » répercutées dans notre pseudo smartphone.

Utilisation d'un dispositif Android raccordé

Si l'installation du companion sur l'Android et du AI2 Setup (cf Premiers pas) est faite correctement, l'écran de la tablette ou du smartphone est le reflet de l'application en cours de construction. Les boutons sont opérationnels … si les blocs correspondants sont bien en place !

Réalisation d'une application

Le but de ce TP est de réaliser une application de conversion de température , de Fahrenheit en °Celsius.

Création du nouveau projet

Cette application va faire appel à deux écrans, un d'accueil et un pour les calculs et l'affichage.

Ecran Screen1 :

Glissez-déposez un élément de type image et, en dessous, un élément de type bouton

Puis, la l'intermédiaire de la fenêtre des propriétés, réglez les différents paramètres des éléments comme suit :

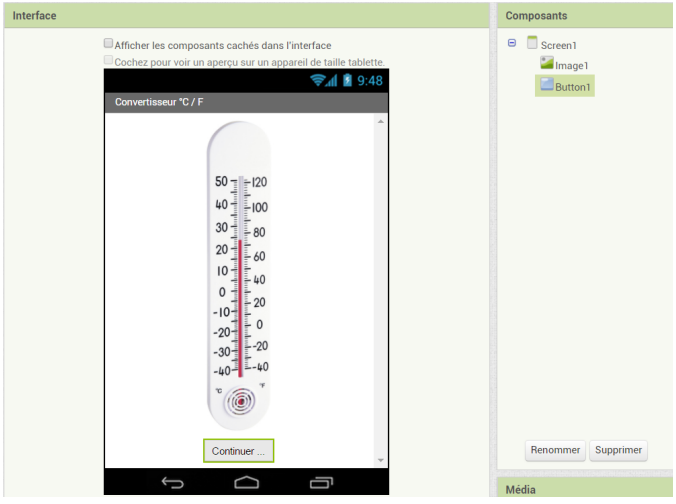
Screen1 :
Alignement horizontal :centré

Alignement vertical : bas

Image1 :

Image : Choisir Charger fichier et pointez un fichier prélablement téléchargé depuis des images de thermomètre (libres de droit dans la mesure du possible ...). Le fichier sera téléversé et disponible dans la petite fenêtre nommée Média en dessous de celle des Objets déposés ;
Hauteur : Remplir parent …
Largeur : Automatique …
Button1 :
Taille de police : 14
Hauteur : Automatique …
Largeur : Automatique …
Texte : Continuer ...
Alignement texte : Centre 1

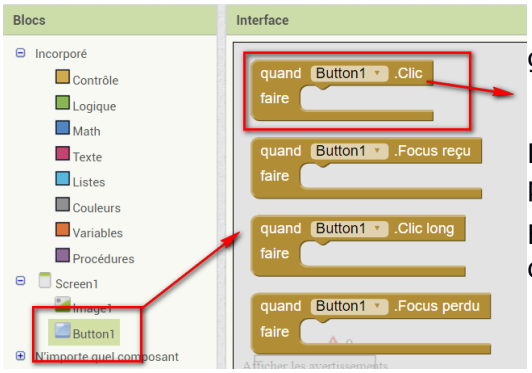
Votre écran1 devrait ressembler à ça :



Quelques blocs

Pour que le bouton « marche », il va falloir assembler quelques blocs …

Dans la liste des éléments disposés sur l'écran Screen1, cliquez sur Button1, s'ouvre alors la liste des blocs disponibles pour cet objet :



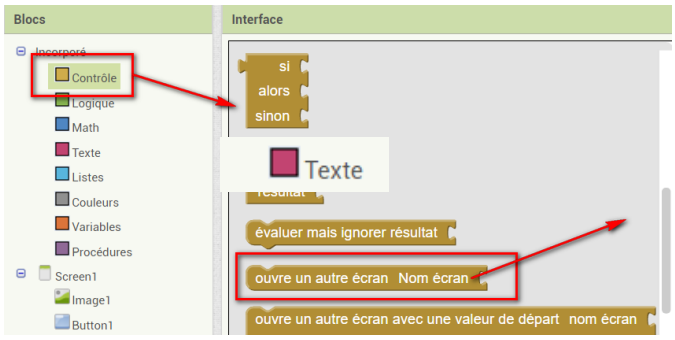
C'est le bloc Quand .... .Clic faire qui sera utilisé donc un glissé-déposé de ce bloc dans la fenêtre de travail s'impose.

L'événement Clic sur Button1 est en place, mais il ne se passera rien puisque l'espace faire est encore vide !

Faites apparaître les blocs de contrôle d'un clic sur l'entrée du même nom dans la liste des blocs :

Il nous faudrait passer à un deuxième écran lors du clic sur le bouton …

Dans la liste des éléments disponibles, c'est un un contrôle qu'il faut : bouton Contrôle puis D&D du contrôle ouvre un autre écran …



Le bloc n'est toujours pas fermé, il manque le nom de l'écran à ouvrir :

Pour ajouter une chaine de caractères, il faut faire appel au bouton Texte

Il faut faire un peu de saisie pour donner le nom de l'écran sur lequel on veut basculer. Compléter la case avec Calculateur par exemple :

Le bloc est maintenant fermé, il n'y a plus de point d'accrochage disponible.

Mais attention, en l'état de l'application, le bouton va « fonctionner », mais il n'existe pas d'écran nommé Calculateur !

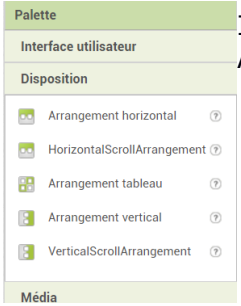
Retour en mode conception, ajout d'un nouvel écran, nommé Calculateur … ouf, sauvé !

Testez votre application via l'émulateur ou directement sur votre smartphone/tablette …

Plus complexe

Ce second écran va devoir contenir bien plus d'éléments, et il pourrait être utile de les organiser …

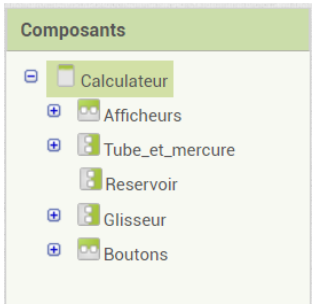
Il faut déjà créer 5 zones containeurs pour nos objets, à l'intérieur de notre écran. La liste des dispositions va aider :



Il faut disposer 2 Arrangements horizontaux, en haut et en bas, et entre, 3 Arrangements verticaux.

Il est possible et pratique de nommer ces « boîtes » via le bouton Renommer placé en bas de la zone des composants placés.

Chaque boîte aura son jeux de propriétés réglées pour un affichage optimal, suivant les indications du tableau suivant.



Propriétés des boîtes

	Afficheurs	Tube et mercure	Reservoir	Glisseur	Boutons
Align. Hori.	Centrer:3	Centrer:3	Centrer:3	Centrer:3	Centrer:3
Align. Vert.	Bas:3	Bas:3	Haut:1	Centre:2	Bas:3
Hauteur	80 pixels	190 pixels	40 pixels	80 pixels	60 pixels
Largeur	Remplir parent	40 pixels	40 pixels	Remplir parent	Remplir parent
Image	Aucun ...	tube.png	bas.png	Aucun ...	Aucun ...

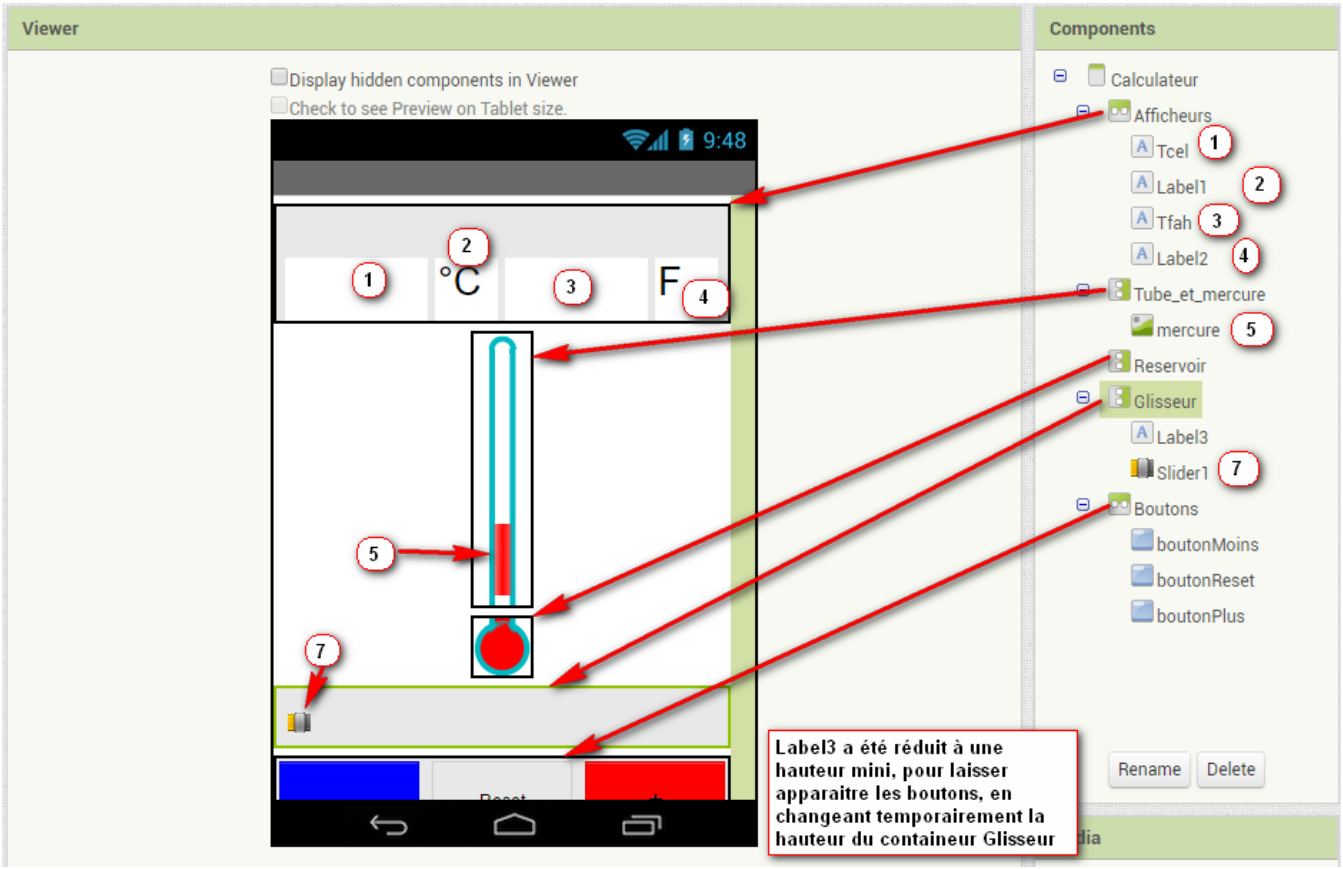
Elles ont toutes en commun, une couleur de fond par défaut et une visibilité activée.

Pour les images tube.png et bas.png, elles sont disponibles à l'adresse suivante :

http://www.zoliv.fr/AppThermo/ suivie du nom de fichier

Chaque boite contient bien évidemment des objets, sauf celle nommée Reservoir qui est vide, mais utilise une image en guise de remplissage …

Pour l'organisation des différents éléments, référez vous à la capture suivante :



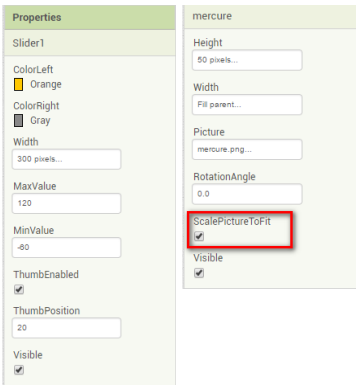
Les textes du containeur Afficheurs font tous 40 pixels de haut, Label1 et Label2 font également 40 pixels de large, tandis que Tcel et Tfah font 30 % de largeur.

Toutes ces propriétés ressemblent furieusement à ce que le CSS3 peut gérer !

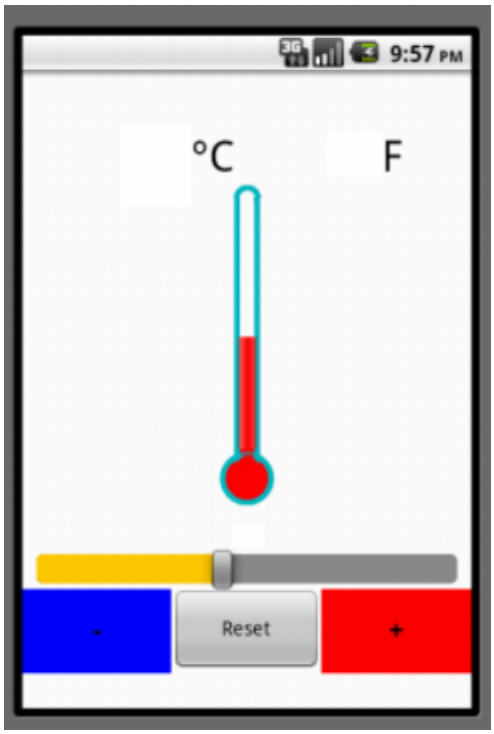
Les boutons du containeur inférieur se partagent tout l'espace disponible sur la largeur et sur la hauteur.

Petit astuce, concernant l'image mercure.png , il lui faut une adaptation à la taille du containeur.

Pour le Slider1, cf les réglages sur la capture ci-contre.



Pour le reste, vous devriez pouvoir trouver des réglages agréables !Votre simulateur pourrez alors ressembler à cela :



Graphiquement, votre application est prête, reste à passer aux « Legos ».

Les éléments déclencheurs sont :

- clic sur boutonPlus

- clic sur boutonMoins

- clic sur boutonReset

- déplacement de Slider1

Les réponses sont :

- affichage de la valeur de la température en °Celsius dans l'étiquette Tcel

- affichage de la valeur de la température en Fahrenheit dans l'étiquette Tfah après calculs

- affichage de la valeur de la température en °Celsius dans l'étiquette Label3 du Glisseur

- augmentation de la température de 5°C lors de l'appui sur boutonPlusmoin

- diminution de la température de 5°C lors de l'appui sur boutonMoins

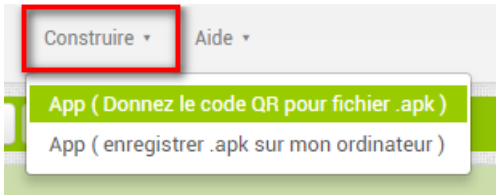
- remise à 20°C de la température lors de l'appui sur boutonReset

Pour le bon fonctionnement de l'ensemble, il faut une variable, de type global, nommée température et initialisée à la valeur de 20 pour rester cohérent.

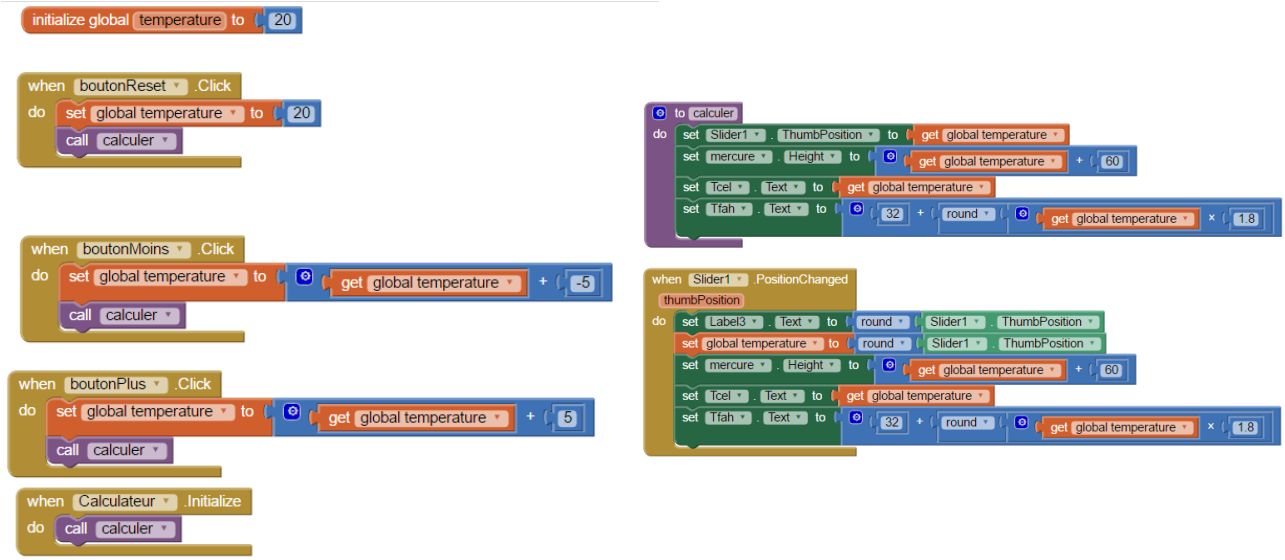
Essayez de construire les différents éléments avec les blocs à votre disposition …

Dernière étape

Tout fonctionne, parfait, il faut maintenant passer à la dernière étape, la construction du fichier APK pour distribution sur des terminaux Android.



Dans le menu, l'entrée Construire permet de choisir entre un QR-code (valable 2 heures seulement) pour charger l'App et un fichier APK à télécharger sur son ordinateur pour une distribution ultérieure.

Dans les deux cas, il ne s'agit plus de simulation, l'application, suite au téléchargement ou au transfert devient réellement disponible sur votre dispositif mobile.

Pour ceux qui sèchent ou ont un peu de mal, voici à quoi ressemble une des solutions :