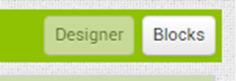
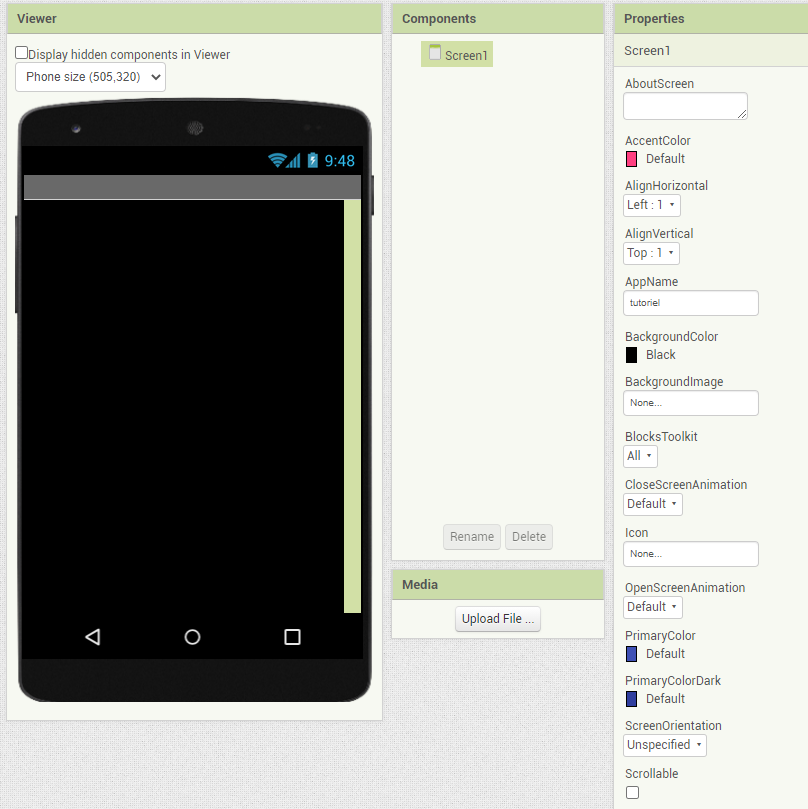
# Tutoriel pour créer un programme d’interface Bluetooth SUR MIT app inventor.

Tout d’abord, lorsqu’on crée un nouveau projet dans l’app inventor on a deux modules différents à gérer le module Designer et le Module Blocks.

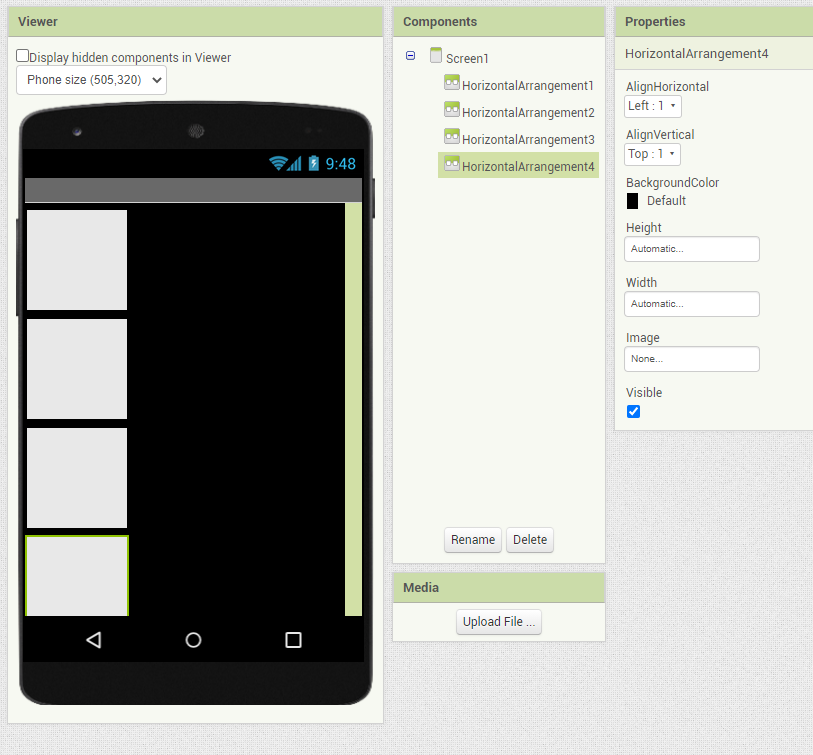


**Module Designer**

Dans le module designer on retrouve l’interface d’utilisateur, dans ce module on va décider les boutons, les afficheurs de texte, des fonctionnalités avec lesquels l’utilisateur pourra interagir, l’aspect esthétique, entre autres.

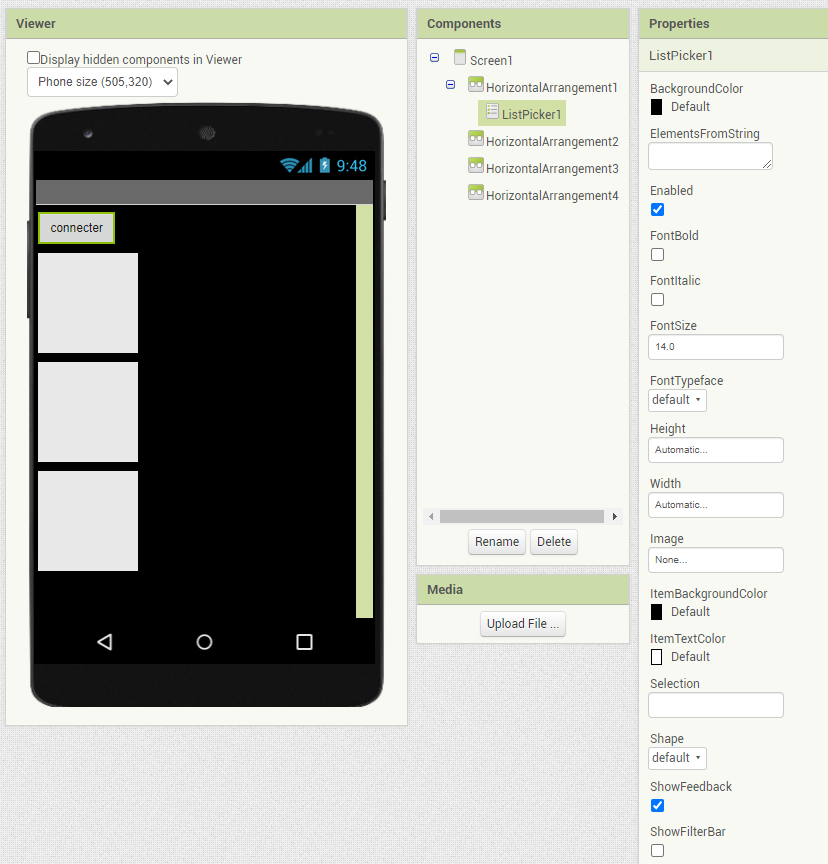
1. Scren : dans ce menu on peut décider les différentes propriétés de l’écran comme la couleur du fond, les marges, l’emplacement des boutons (droite, gauche et centre). Dans notre cas, on a décidé de mettre un fond noir, enlever le titre en haut (screen 1) et mettre l’emplacement des boutons à gauche.
2. Layout arrangement :

La fonction de ces boîtes est de contenir et d'organiser différents composants de l'interface utilisateur, si nous n'utilisons pas les boîtes, il serait impossible de mettre deux éléments au même endroit. Dans notre cas nous utilisons quatre boîtes.



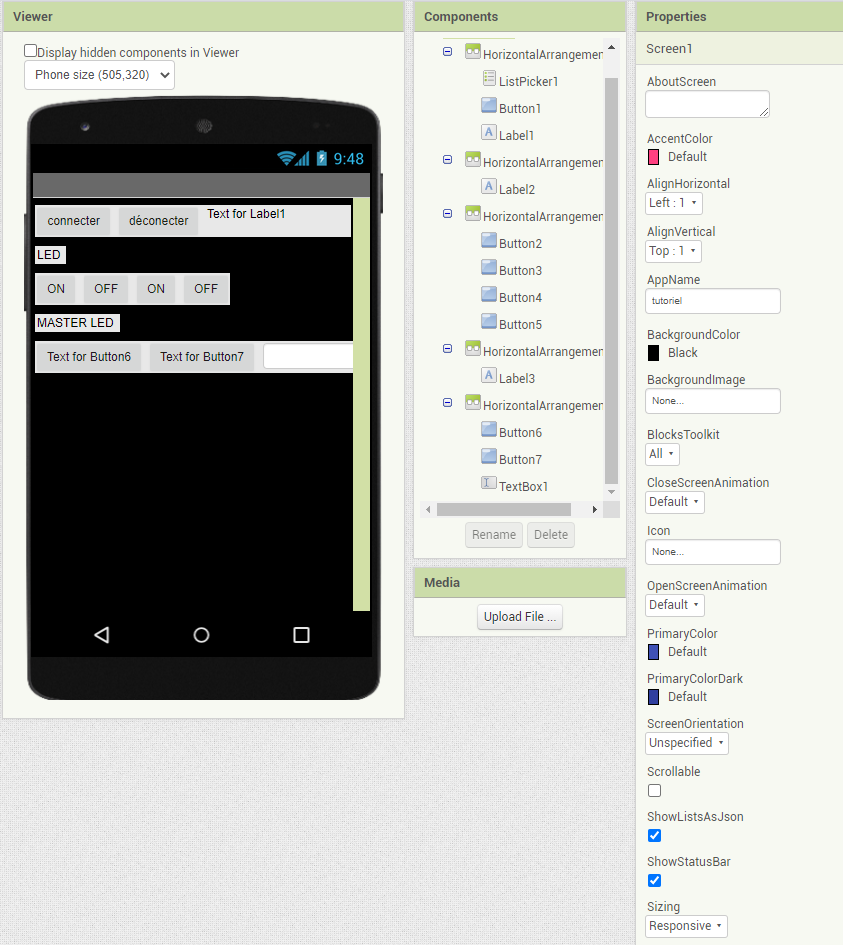
1. ListPicker :

Le listePicker est un type de bouton qui contient une liste des boutons à l’intérieure. Cet outil sera le bouton de connexion. Alors, à l’intérieur les choix seront les adresses des dispositifs Bluetooth captées par le téléphone.



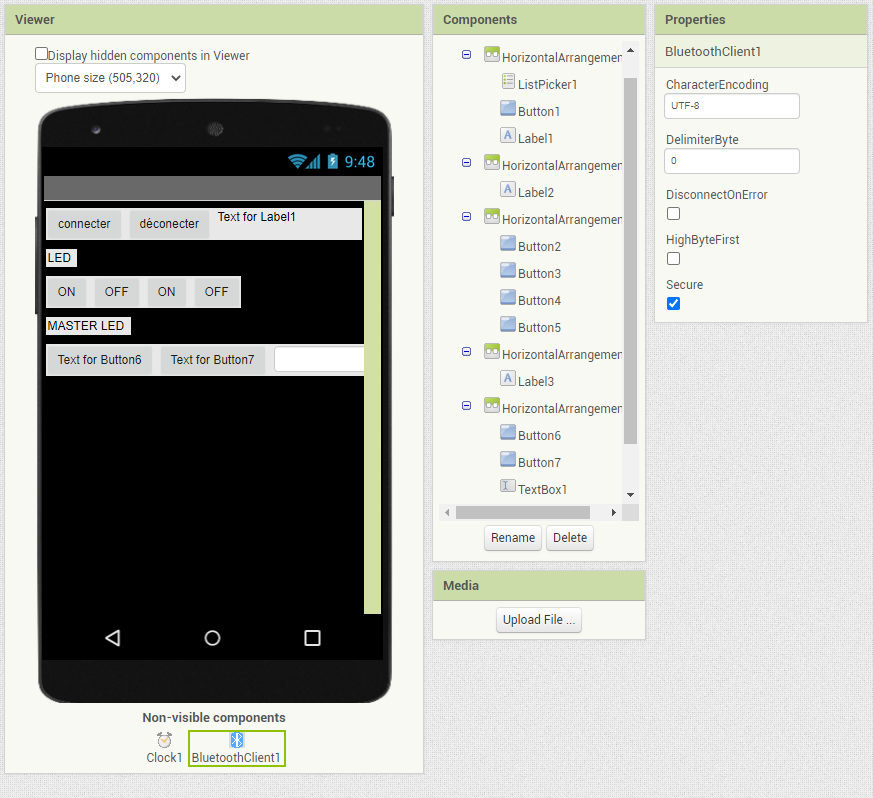
1. Les autres boutons, le Label, textbox :

Les autres boutons que l’on utilise dans le projet n’ont pas des ListPicker, le fonctionnement de ces boutons sera défini dans le deuxième module (Blocks), on pourrait les comparer avec un bouton-poussoir dans un circuit ou un switch. Le Label c’est du texte qui peut contenir des instructions, les conséquences d’une action programmée ou toute simplement un texte prédéfini. Le textbox juste comme le label est une boite qui peut contenir du texte, la seule différence est qu’on peut décider si on veut écrire du texte ou seulement l’afficher.



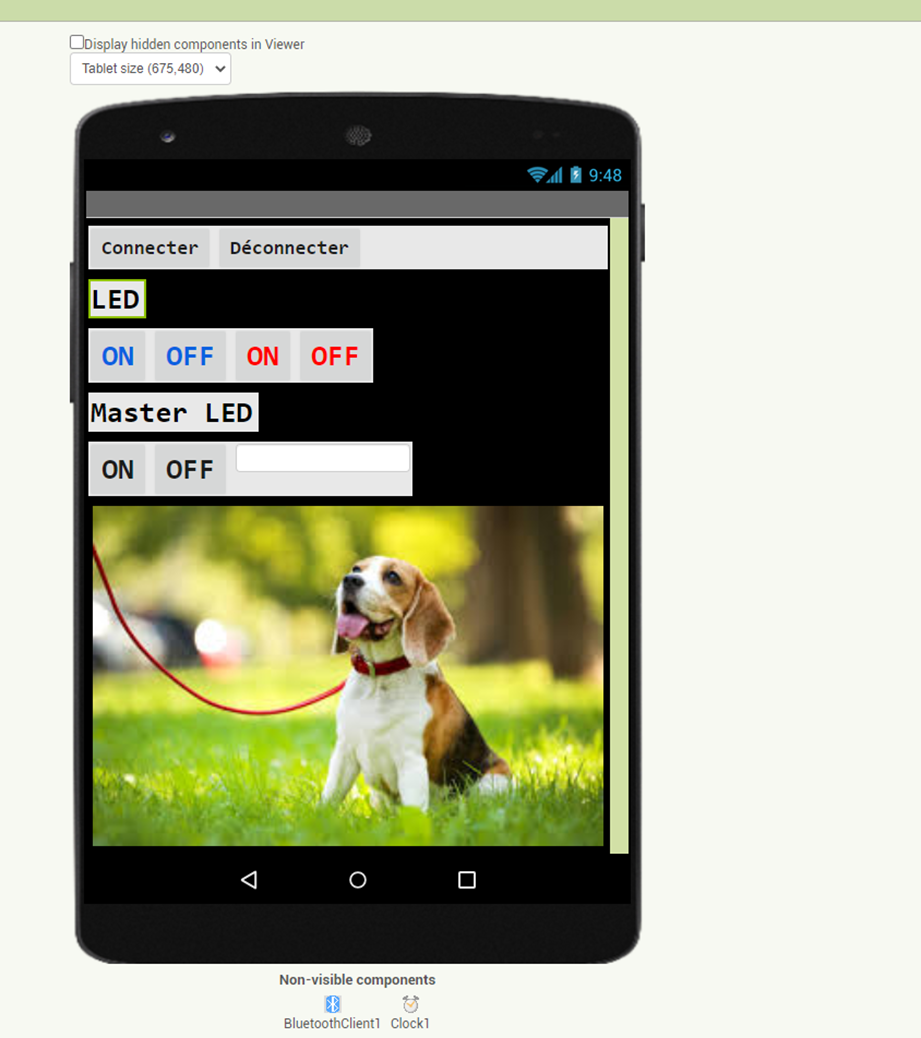
1. Capteurs et connectivité :

Les capteurs et la connectivité sont des fonctionnalités qui existent déjà dans le téléphone, elles peuvent être incluses et utilisées dans le programme, dans notre cas, nous incluons l’horloge et le système Bluetooth client.



1. Personnalisation :

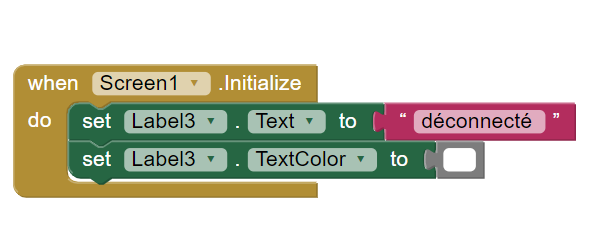
Comme déjà mentionné on peut personnaliser les boutons, le texte, mettre une image ou toute autre chose au besoin.

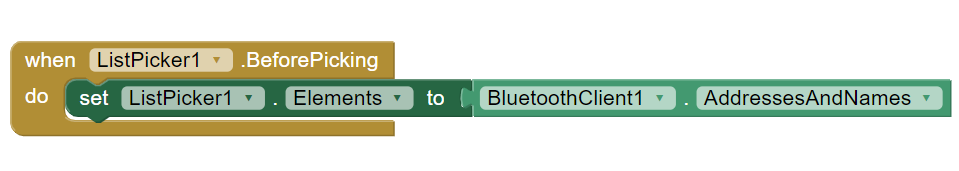


**Module Blocks**

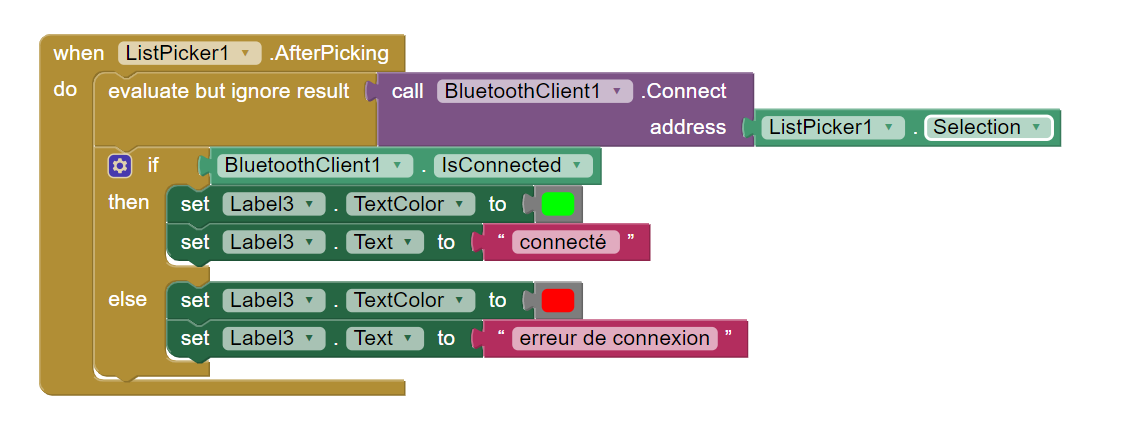
Dans le module Blocks, on crée le programme de notre application. Ce dans ce module qu’on va décider le rôle de chaque bouton, boite de texte et fonctionnalité du programme. La programmation en MIT app Inventor ressemble un peu à celle en LabVIEW, mais avec une synthèse plus proche de la programmation en C et il est important de garder les séquences.

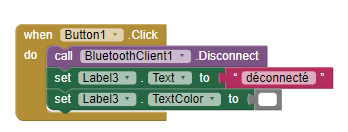
1. Lorsque l’application est initialisée le téléphone n’est pas encore connecté au module Bluetooth HC-05. Alors le label texte affiche le message déconnecté.



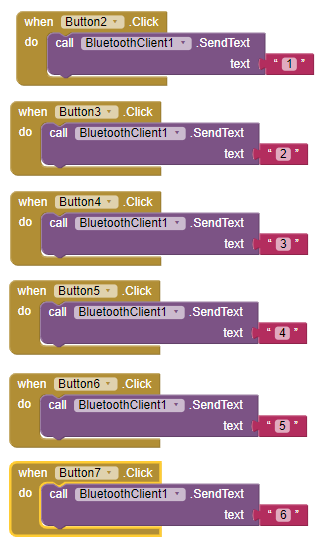


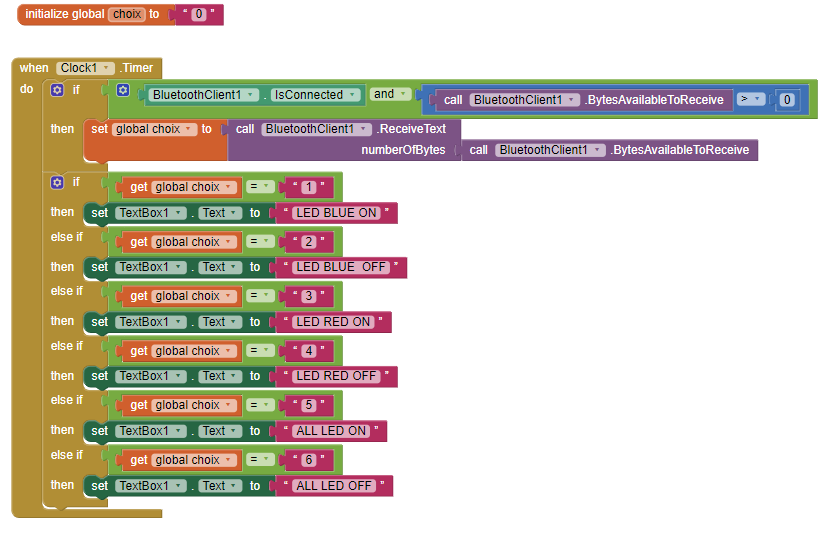
1. Le ListPicker montre les adresses Bluetooth avec lesquelles on peut faire une connexion.
2. Après avoir choisi l’adresse, le téléphone fait une tentative de connexion, si la connexion est faite un message vert (connecté) est affiché, si la connexion échut un message rouge d’erreur est affiché.



1. Le bouton déconnexion ferme la connexion entre les deux dispositifs et affiche un message de déconnexion.

1. Transmission Bluetooth : Les boutons ON OFF envoient un message Bluetooth lorsqu’on les appuis.



1. On crée une variable qui va contenir le message envoyé par le Mbed. Tout d’abord, ces IF seront activés avec l’horloge de façon synchrone, cela veut dire que peu importe dans quel moment du programme on se trouve le délai sera toujours respecté (comme une interruption). Ensuite, la première condition dit que si on est connecté avec un autre dispositif et qu’on a des bytes dans la mémoire la variable choix aura la valeur du byte reçue. Dans l’autre IF cette valeur sera comparée afin d’afficher un message en dépendant de la LED qui a été allumée ou éteinte.