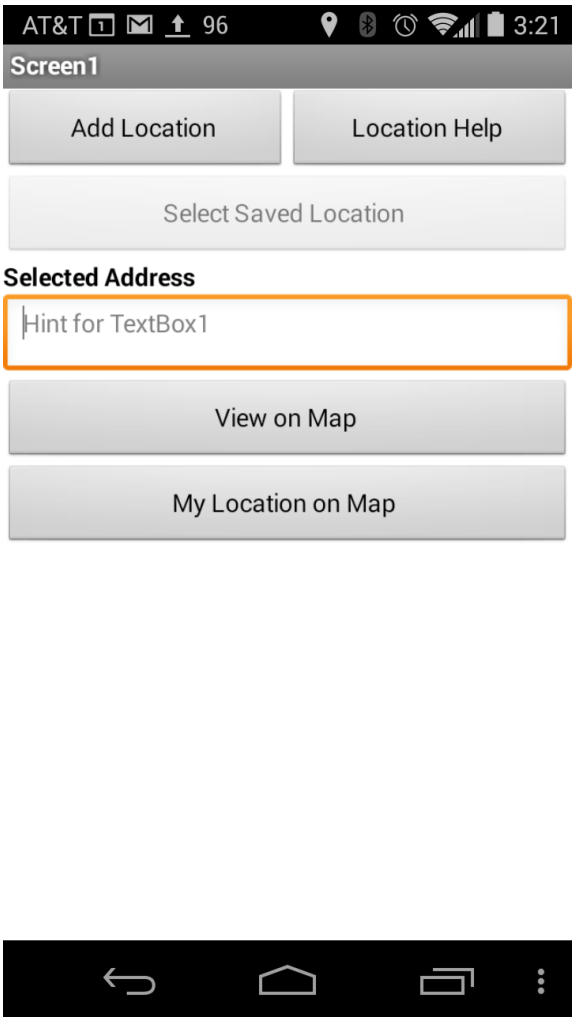
Map It : sauvegarder et afficher des lieux sur une carte Google

Imaginez que vous prévoyez de rencontrer de nouveaux amis dans une ville inconnue et qu'ils vous donnent des adresses de lieux de rencontre. C'est fastidieux de toutes les garder dans des messages e-mail ou texte. Donc, il serait bien d'avoir une appli où vous pouvez accéder facilement à tous les lieux. Nous pouvons construire l'appli qui le fera pour nous !

Quelles fonctions voudriez-vous inclure dans cette appli ?

- Un moyen de sauvegarder les lieux
- Un moyen de voir les lieux sauvegardés
- Voir les lieux sur Google Maps
- Voir votre emplacement actuel sur Google Maps

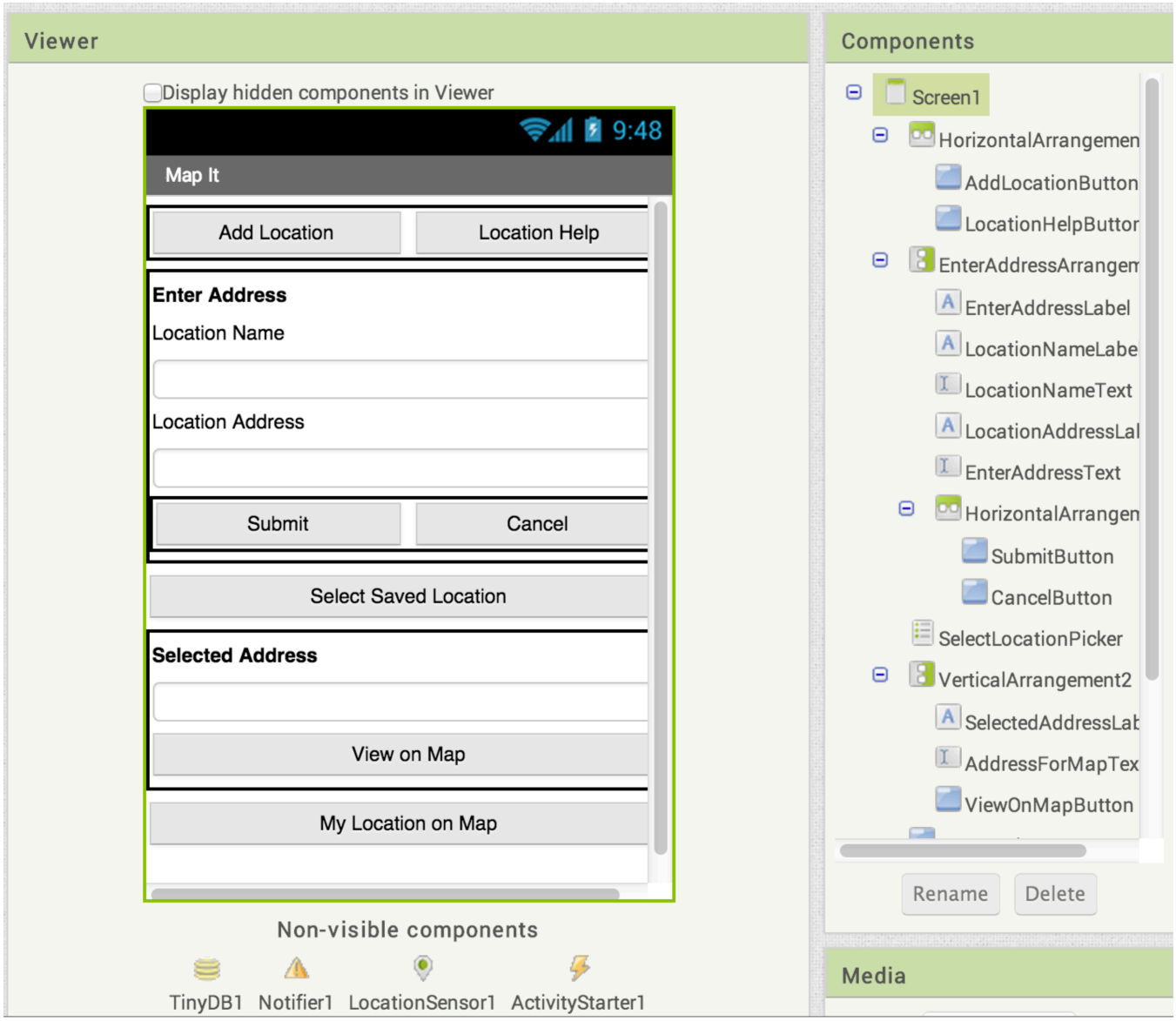
Concevons une interface utilisateur ressemblant à la figure à droite. Voici l'écran général que verra l'utilisateur, mais d'autres composants seront visibles lorsqu'il sélectionnera “Add Location”.



La figure ci-dessous montre comment les composants sont organisés dans l'éditeur de conception. Elle montre également les composants non visibles (TinyDB, ActivityStarter, LocationSensor, Notifier) utilisés dans l'appli.

Obtenez tous les composants nécessaires que vous voyez ici et nous passerons ensuite aux blocs. Il n'est pas nécessaire de nommer vos composants comme ci-dessous, mais assurez-vous de les renommer de manière à vous souvenir de leur usage. Cela facilitera les choses lorsque nous travaillerons avec les blocs.

“Select Saved Location” est dans ce cas un composant ListPicker, pas un bouton.

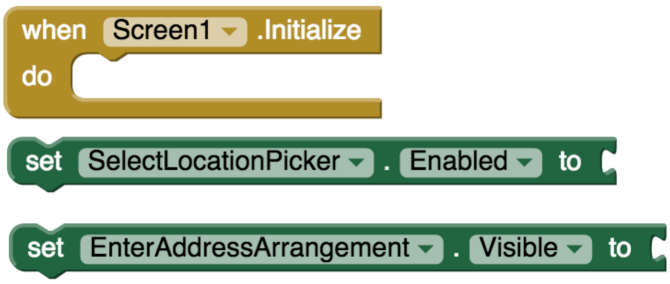


Initialiser l'écran

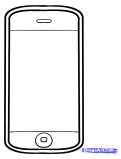
Voici ce que nous voulons faire à l'ouverture de l'appli :

- Masquer les composants pour que l'utilisateur saisisse une adresse à sauvegarder
- Désactiver “Select Saved Location” jusqu'à ce que l'utilisateur sauvegarde au moins une adresse

Tous les boutons ont une propriété “Enabled” et tous les composants ont une propriété “Visible”. Ces propriétés sont “vraies” si vous ne les modifiez pas, mais vous pouvez les définir sur "faux" afin d'arrêter le fonctionnement d'un bouton et de faire disparaître le composant de l'écran.



Avec ces deux propriétés et les blocs illustrés ci-dessous, comment initialisons-nous l'écran pour qu'il affiche les deux points cités plus haut ?



Testez !

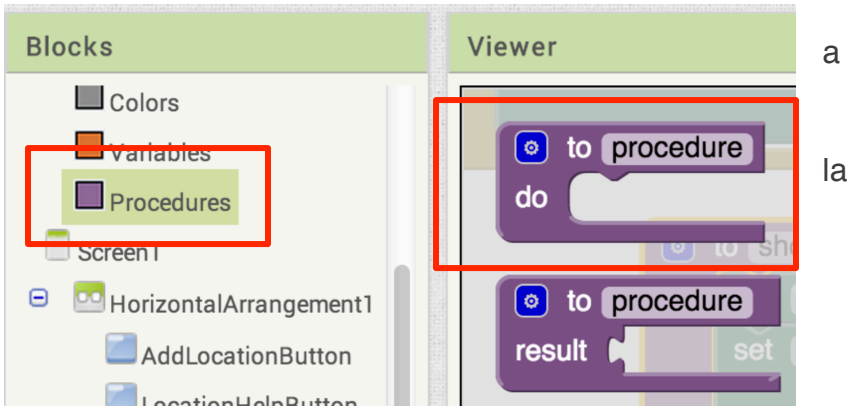
Voyez ce qui se produit quand vous définissez Enabled sur vrai ou faux et de même avec Visible.

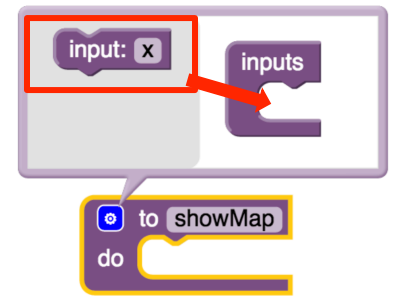
Voir l'emplacement actuel sur la carte

Nous avons inséré un bouton, MyLocationButton ‘My location On the Map’ sur lequel on peut cliquer pour voir son emplacement actuel. Quel est le premier bloc dont nous ayons besoin pour faire fonctionner un bouton à cliquer ? Un gestionnaire d'événements !

Nous devons écrire une procédure qui ouvrira l'appli de cartes. Sous les blocs intégrés, l'ensemble violet est appelé "Procedures". Allez dans celui du haut.

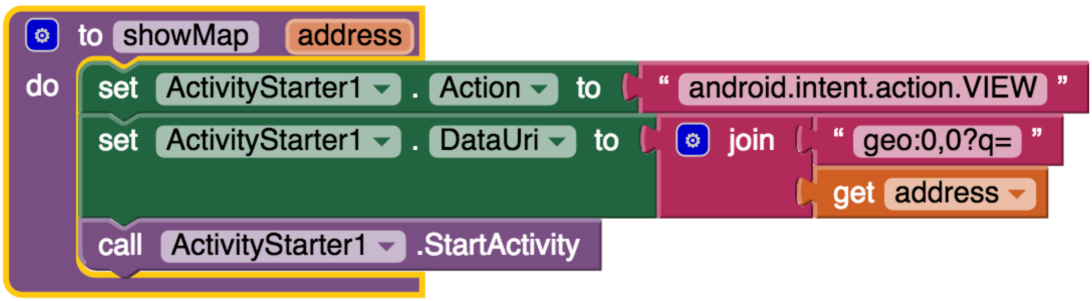
Appelons-le “showMap.” Remarquez qu'il y a une petite encoche en haut à gauche du bloc, comme dans le bloc if-then-else. Cliquez dessus et ajoutez une entrée pour la procédure.





Après avoir saisi l'entrée, une petite croix orange apparaît à côté de “showMap.” Cliquez sur la croix et renommez-la "address"

Voici une entrée pour la procédure. Cela signifie que nous pouvons réutiliser cette procédure pour afficher des cartes de différentes adresses. Nous devrons utiliser ActivityStarter ici pour démarrer l'appli Google Maps. Au final, votre procédure showMap doit ressembler à cela.

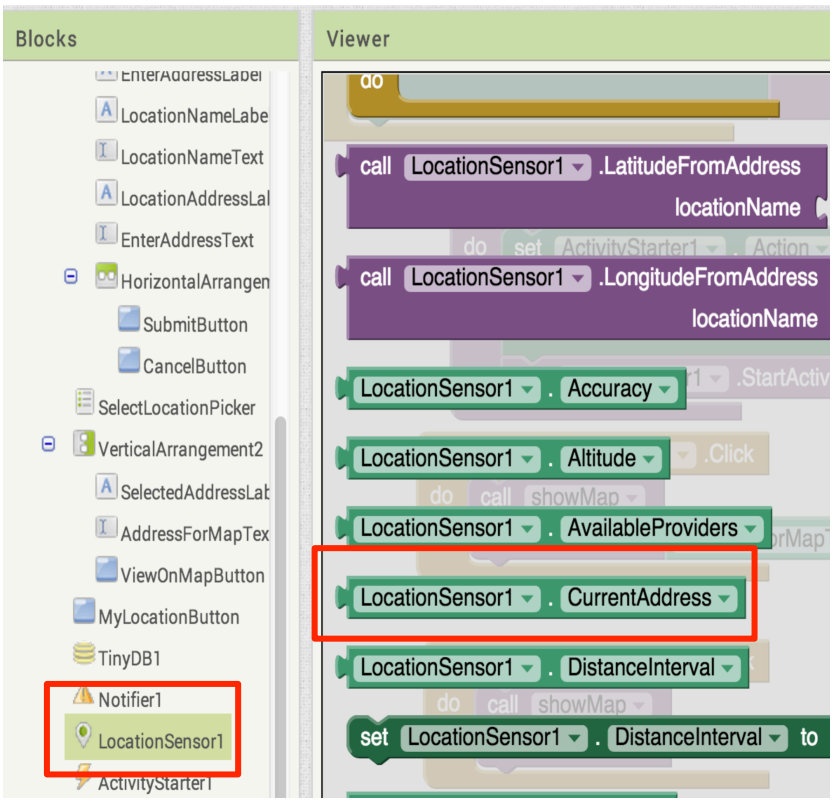


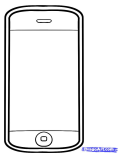
La combinaison de l'entrée de texte pour l'“Action” ainsi que pour “DataUri” indique à votre application de démarrer l'appli Maps. Si vous passez la souris sur address sans cliquer, vous verrez le bloc “get address” dont vous avez besoin pour la deuxième partie de l'entrée DataUri.

Maintenant que la procédure “showMap” a été créée, si vous cliquez sur “Procedures” sous les blocs intégrés, vous verrez un bloc pour “call showMap.” C'est ce que nous voulons en cliquant sur le bouton MyLocationButton.

Il ne nous manque plus désormais que l'entrée d'adresse pour la procédure showMap. Cette adresse viendra du Location Sensor. Voyons quels blocs sont disponibles ici.

Il y a un bloc du Location Sensor qui nous donne CurrentAddress. Utilisez ce bloc pour compléter le gestionnaire d'événements MyLocationButton.Click.





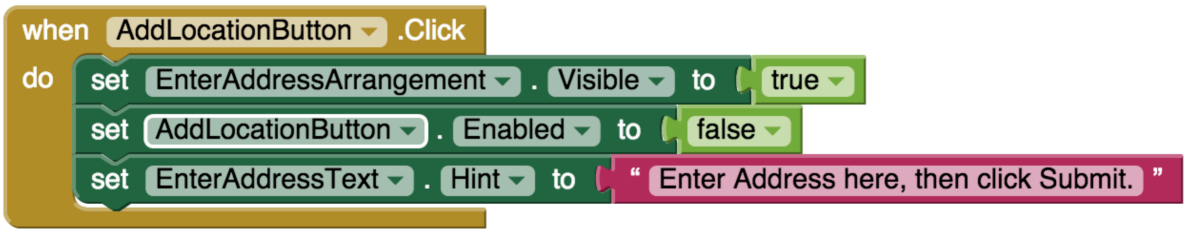
Testez !

Voyez si vous pouvez trouver votre emplacement actuel sur Google Maps à partir de l'appli.

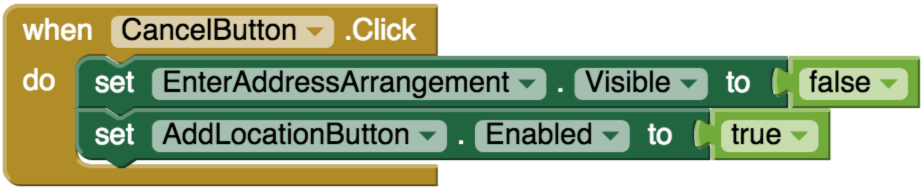
Ajouter une nouvelle adresse

Nous utilisons le bouton AddLocationButton pour ajouter une nouvelle adresse et le bouton CancelButton pour annuler le processus d'ajout et le bouton LocationHelpButton pour afficher les types d'adresses qui peuvent être saisies. Les blocs pour réaliser ces actions fonctionnent comme ci-dessous.

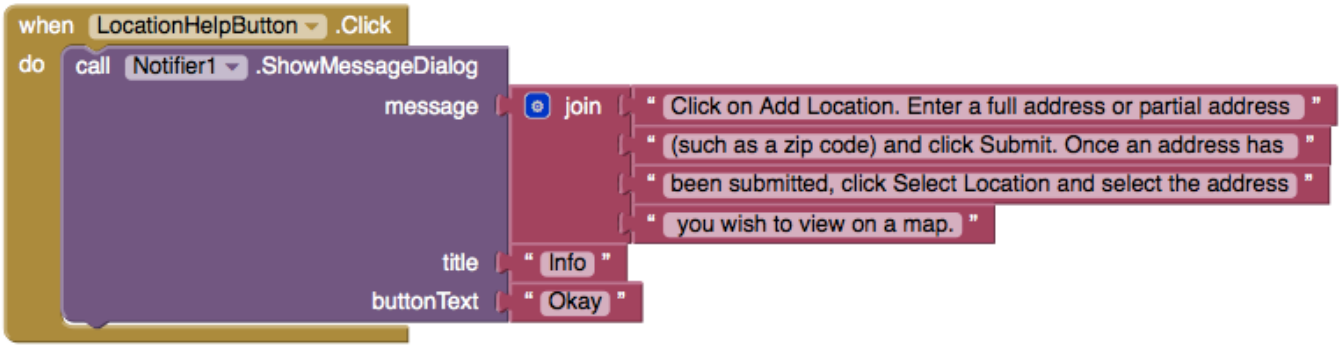
Lorsque nous avons initialisé l'écran, nous avons défini le réglage EnterAddress sur faux, mais nous devrons le voir quand nous devrons saisir une adresse à sauvegarder. Cela signifie que nous devons changer “Visible” sur vrai. Nous changerons “Enabled” sur faux comme nous avons déjà cliqué sur le bouton. Nous définirons également texte pour la Address Text box.

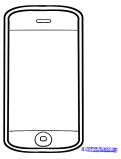


Si l'utilisateur clique sur Cancel, le réglage sera masqué de nouveau. Avec cette approche, l'écran reste simple et nous évitons les confusions. Nous fournissons également le bouton SubmitButton pour permettre à l'utilisateur d'indiquer qu'il veut stocker les données.



Si l'utilisateur n'est pas certain de savoir utiliser l'appli, nous pouvons définir le LocationHelpButton pour qu'il affiche un message de notification comme ci-dessous.



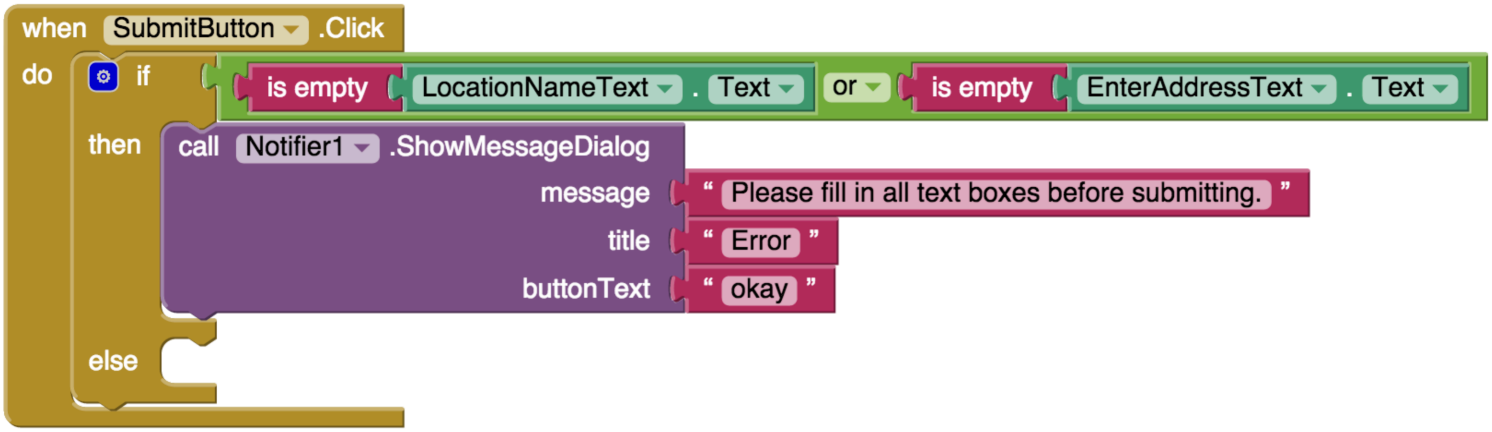


Testez-le ! Voyez si les boutons Add Location, Location Help et Cancel fonctionnent tous correctement.

Amusez-vous avec les options vrai/faux et les différentes zones de textes également.

Soumettre de nouvelles adresses

Nous voulons avoir la certitude que quand l'utilisateur clique sur le bouton SubmitButton, une adresse et un nom de lieu ont été soumis. Pour ce faire, nous utilisons de nouveau le bloc if-then-else comme nous l'avons fait dans les autres tutoriels. Si les zones de texte ne sont pas renseignées, nous affichons un message d'erreur à l'aide du Notifier. Nous ajoutons un nouveau lieu à notre liste sauvegardée. Une fois l'adresse ajoutée, nous masquons notre réglage de nouveau. Voici les blocs pour afficher un message d'erreur à l'aide du Notifier.



TinyDB

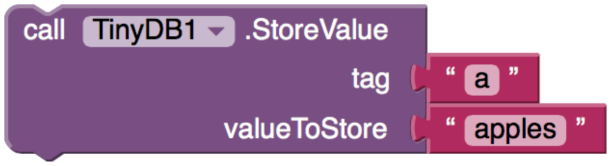
TinyDB est similaire aux listes dans la mesure où il stocke l'information pour vous. Cependant, TinyDB doit être ajouté auparavant comme un composant du concepteur.

En outre, vous pouvez spécifier la balise afin que la liste d'éléments que vous sauvegardez ne soit pas forcément une liste numérotée. Par exemple, en continuant notre liste d'épicerie, nous pouvons procéder ainsi pour sauvegarder notre liste.

p→pommes

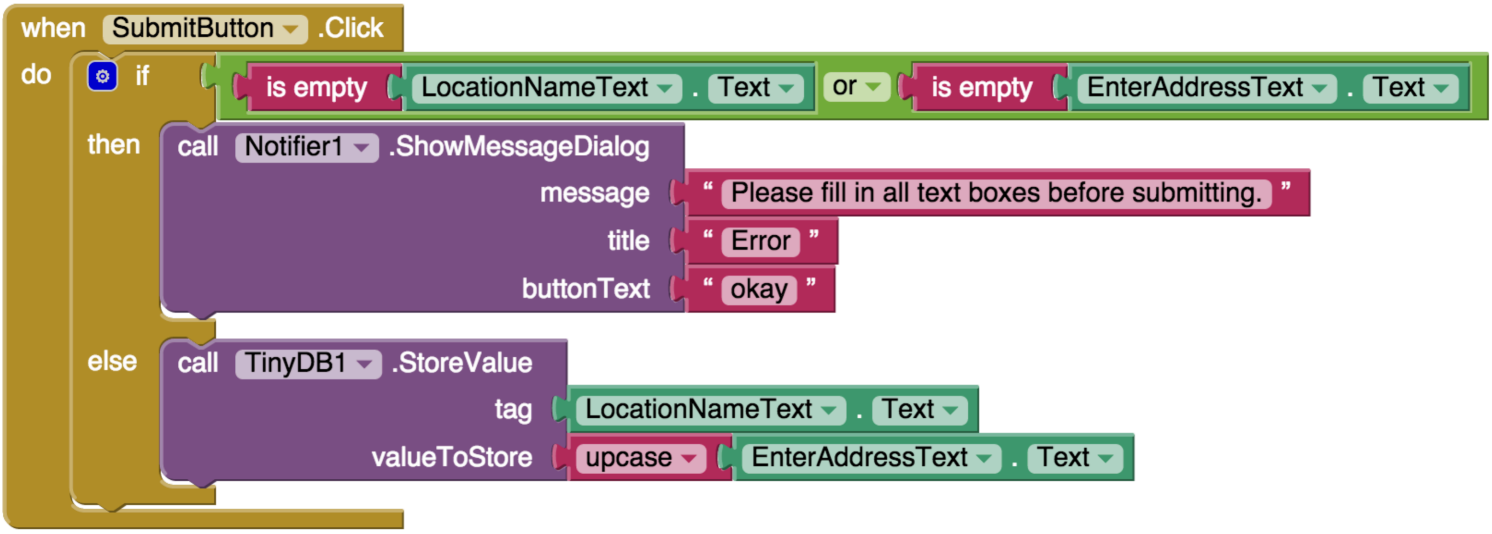
o→oranges

c→citrons

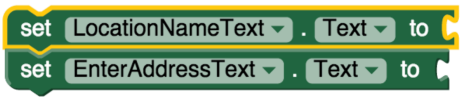


Balises	Valeurs
p	pommes
o	oranges
c	citrons

La balise est juste une façon pour vous de sauvegarder et récupérer la valeur que vous aviez sauvegardée. Si vous demandez la valeur associée à “p” vous obtiendrez “pommes.” Pour notre appli ici, nous pouvons utiliser TinyDB pour conserver le nom d'un emplacement comme une balise et l'adresse comme valeur.

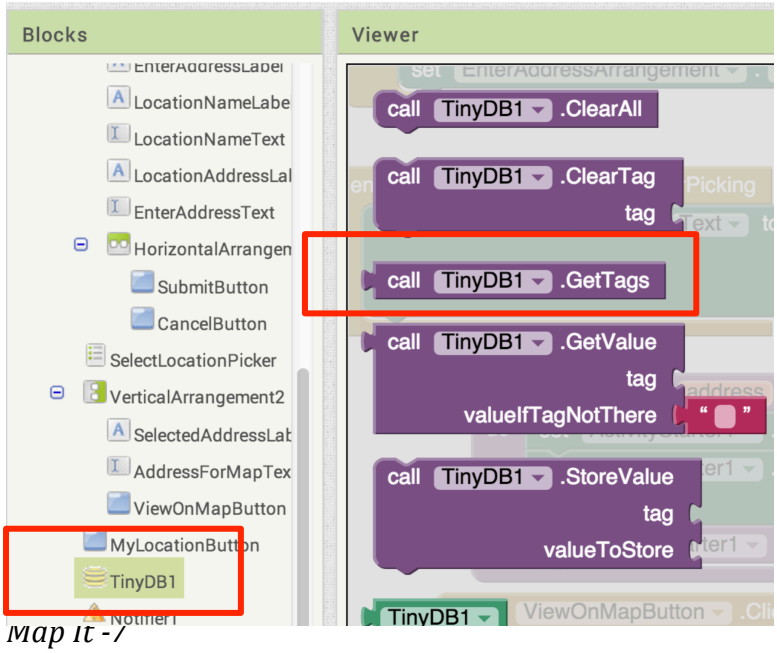


Nous devons également ajouter des blocs pour effacer le texte de LocationNameText et EnterAddressText après la soumission d'une adresse. Qu'est-ce que vous devez ajouter aux blocs suivants (et où devez-vous les mettre) pour effacer le texte de ces champs ?



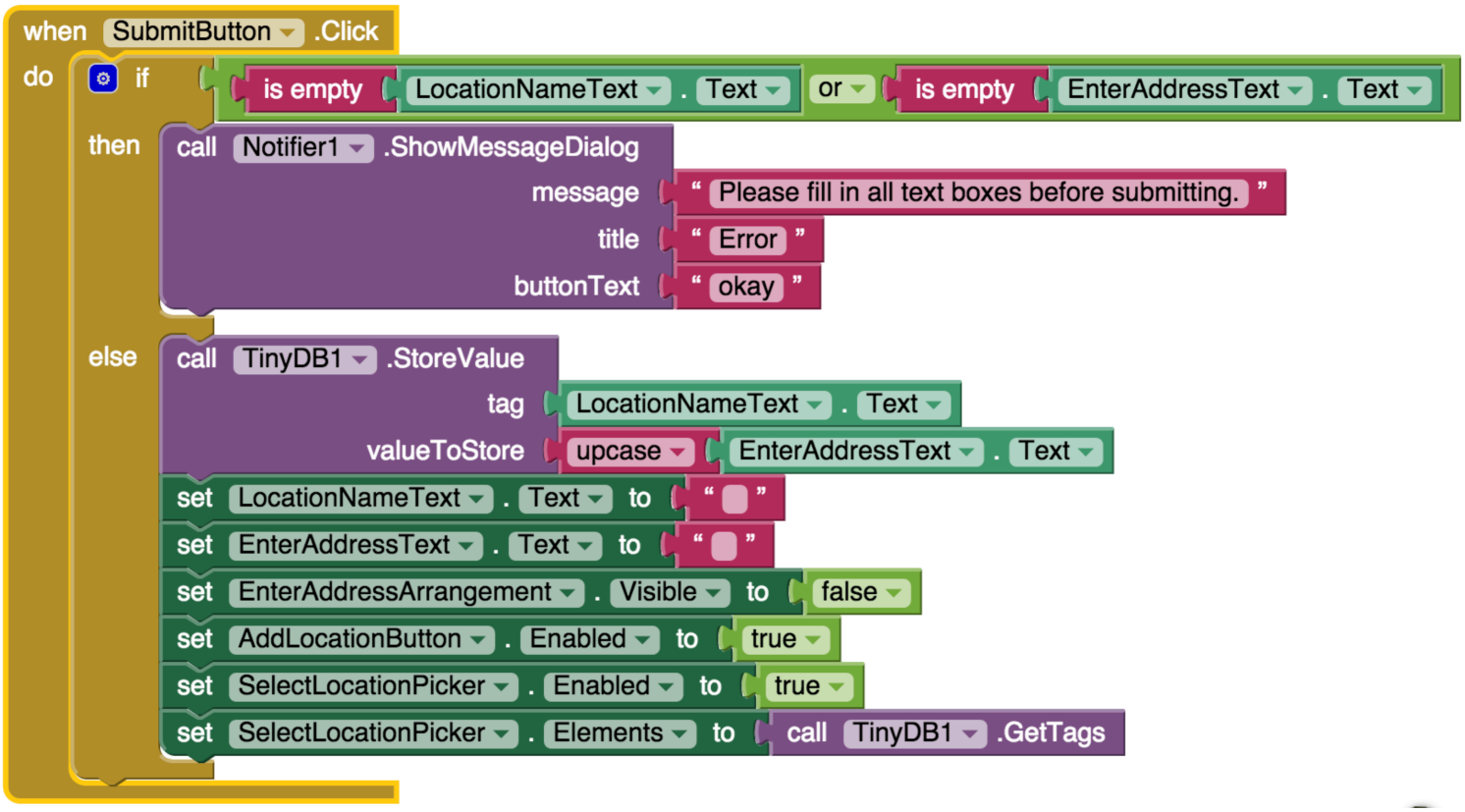
Aussi, comme avec le bouton Cancel, nous devons masquer le réglage et activer de nouveau le bouton Add Location après la soumission. Regardez votre gestionnaire d'événements “when CancelButton.Click” pour trouver les blocs à réutiliser ici.

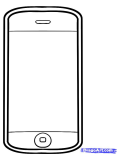
Une fois que tous les aspects ont été pris en compte dans le gestionnaire d'événements, nous devrons également activer notre composant ListPicker. Nous devons également entrer nos Location Names dans le ListPicker afin de les retrouver plus tard. Comment pouvons-nous obtenir les Location Names pour le ListPicker? Nous savons que les Location Names sont utilisés comme des balises dans la TinyDB, alors voyons ce qui est disponible.



Il semble que nous puissions avoir toutes les balises de TinyDB dans la procédure “call TinyDB1.GetTags.”

Un ListPicker affiche des éléments. Nous devons les associer aux balises que nous obtenons de la TinyDB. Le gestionnaire d'événements SubmitButton.Click final devrait avoir cette apparence.





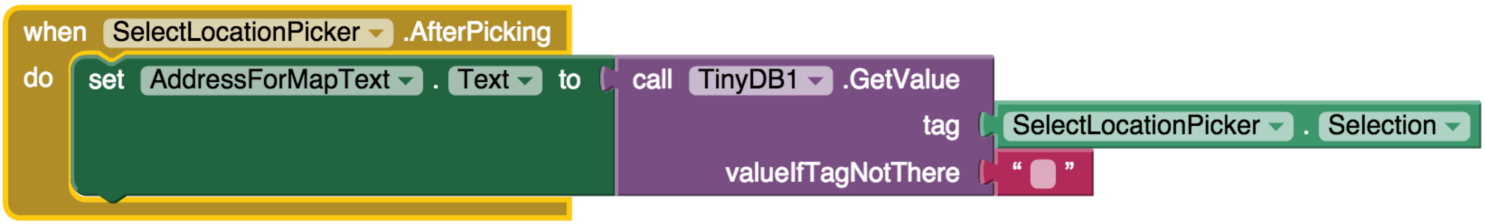
Testez !

Vérifiez si vos lieux ont été sauvegardés en cliquant sur “Select Saved Location” après avoir soumis une adresse.

Sélectionner des adresses sauvegardées

Lorsque l'utilisateur clique sur le List Picker ‘Select Saved Location, il est en mesure de sélectionner un lieu à partir du nom utilisé pour l'adresse.

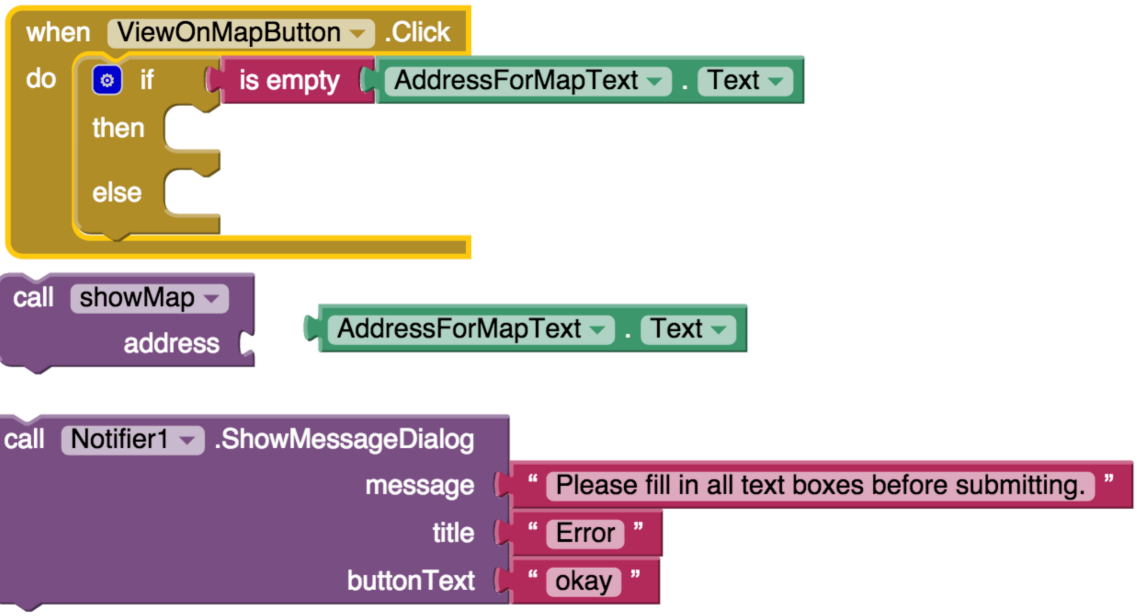
Toutefois, après avoir sélectionné un lieu, nous voulons que l'adresse puisse afficher le lieu sur la carte. Pour cela, nous devons récupérer l'adresse sauvegardée dans TinyDB. Nous pouvons le faire à l'aide du bloc “call TinyDB1.GetValue” avec Picker Selection comme entrée pour la balise. La valeur retournée sera une adresse et nous la placerons dans la zone de texte AddressForMapText box. Les blocs complets figurent ci-dessous.



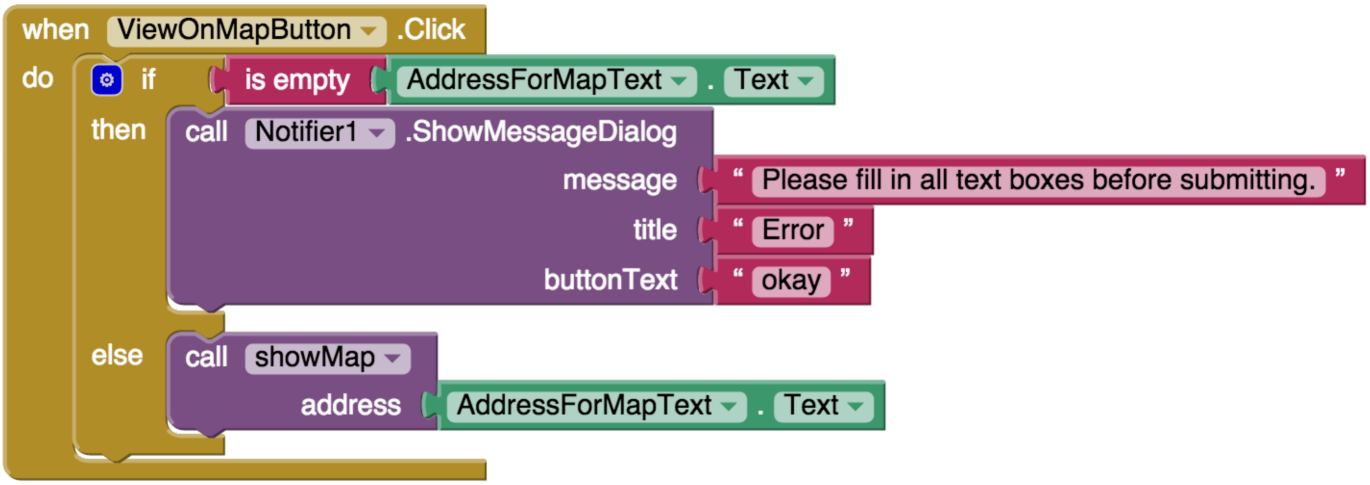
L'adresse apparait-elle dans la zone de texte AddressForMapText box ? Si tel est le cas, ajoutons la fonctionnalité au bouton View On Map.

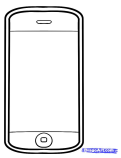
Lorsque l'on clique sur le bouton ViewOnMapButton, nous nous assurons qu'une adresse figure dans la zone de texte AddressForMapText box. Dans le cas contraire, un message d'erreur est affiché. Pour ce faire, nous devrons utiliser de nouveau le bloc if-then-else pour rechercher le texte manquant. Pour le message d'erreur, nous pouvons réutiliser le même bloc Notifier que nous avions utilisé pour le message d'erreur SubmitButton.Click.

En fonction des blocs suivants, comment complétons nous ce gestionnaire d'événements ?



Bloc final





Testez !

Sauvegardez plusieurs lieux et voyez si vous pouvez tous les visualiser individuellement sur la carte.