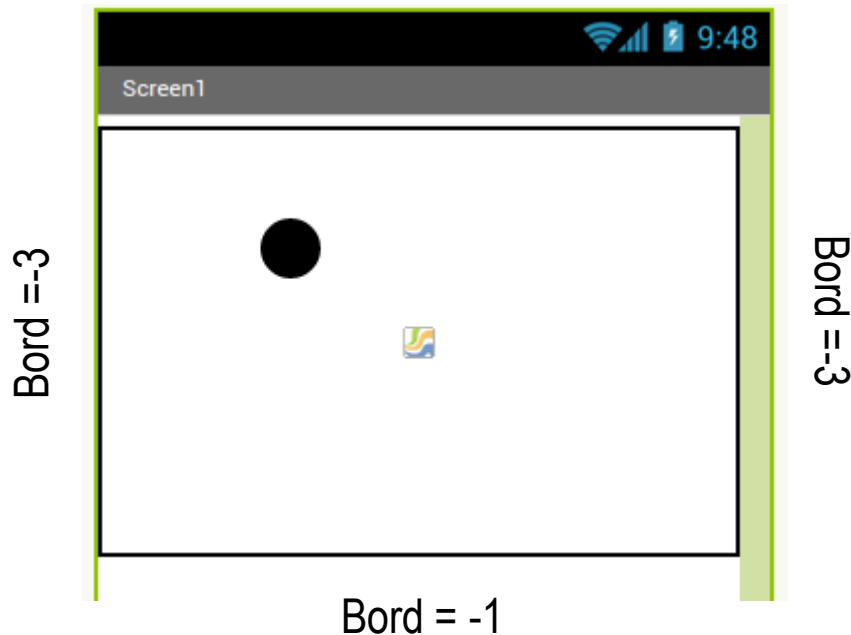




Faire rebondir une balle sur le bord du cadre

Faire rebondir une balle lorsqu'elle touche le bord d'un cadre.

Bord = 1



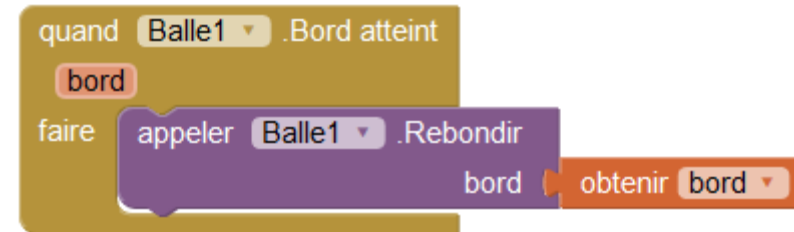
Des valeurs sont attribuées aux bords (1,3,-1,-3)
Pour le haut, la droite, le bas, la gauche

Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

- * Cadre
- * Balle

Éditeur de blocs



Ça veut dire quoi?

Ball1.BordAtteint détecte le moment où la Balle atteint le bord du cadre et fournit l'argument **bord**.

Chaque **bord** du cadre a une valeur numérique. Il suffit de renvoyer la même valeur **bord** dans l'appel de la procédure **Rebondir**

Ball1.Rebondir fait rebondir la Balle dans la direction opposée au mur (**Ball1.Orientation** est modifiée de 180 degrés.)



Ajouter du son

Ajouter du son soit en ajoutant des fichiers audio courts

Comme un son de crash lorsque deux objets entrent en collision

Soit des fichiers audio plus longs tels qu'une musique d'arrière plan



Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

- * Bouton
- * Son
- * Lecteur

Éditeur de blocs

```
quand Screen1 .Initialise  
faire appeler Lecteur1 .Commencer
```

```
quand Bouton1 .Clic  
faire appeler Son1 .Jouer
```

Ça veut dire quoi?

Quand le **Bouton1** est cliqué, le **Son1** est lu.
Quand **Screen1** s'affiche, la lecture de **MusiqueArrièrePlan** débute.



Mouvement programmé par un chrono

Déplacer l'image lutin en fonction du temps qui passe.

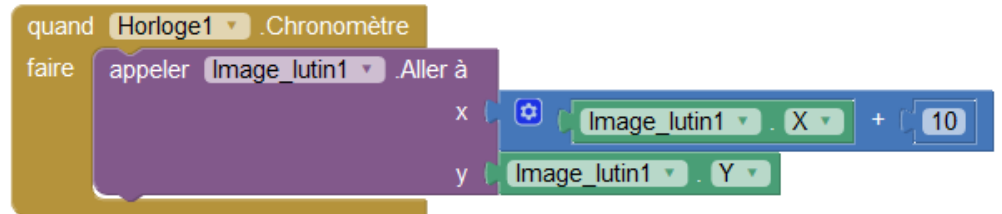


Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

- * Cadre
- * Image lutin
- * Horloge

Éditeur de blocs



Ça veut dire quoi?

Aller à déplace un objet à une position absolue sur le cadre et non à une position relative (déplace un objet à un point précis et pas d'une certaine distance depuis sa position initiale).

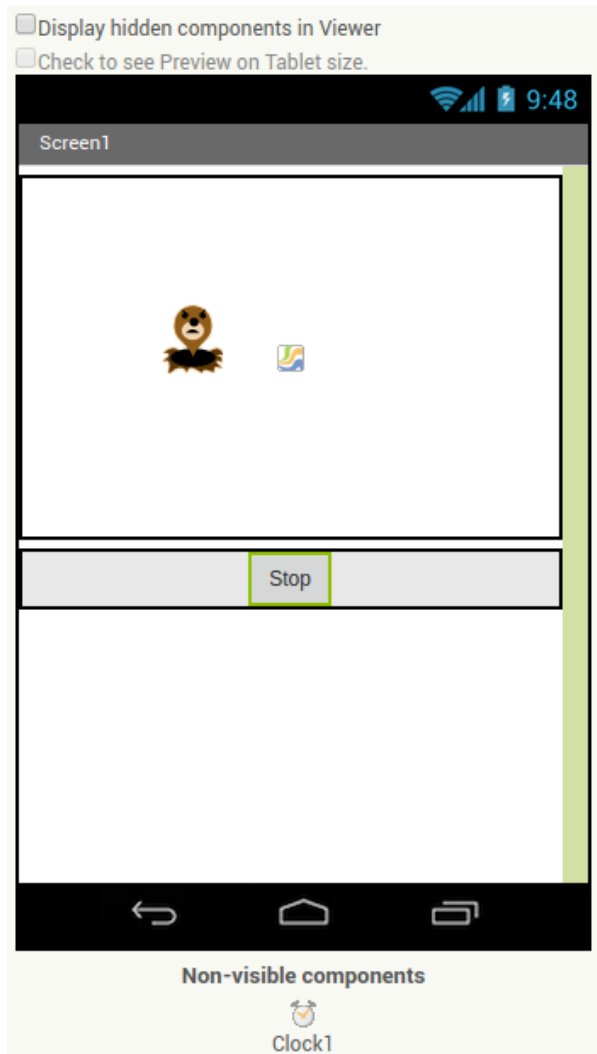
Horloge1 a un Intervalle qui indique la Fréquence à laquelle le Chronomètre doit se déclencher.

À chaque fois que **Horloge1.Chronomètre** se déclenche, l'Image lutin se déplace de 10 Pixels vers la droite étant donné qu'on ajoute 10 pas à chaque déclenchement du Chronomètre.



Commencer/Arrêter le mouvement programmé

Cela permet à l'utilisateur d'appuyer sur un bouton pour commencer ou arrêter le déplacement de l'Image lutin en fonction du temps.

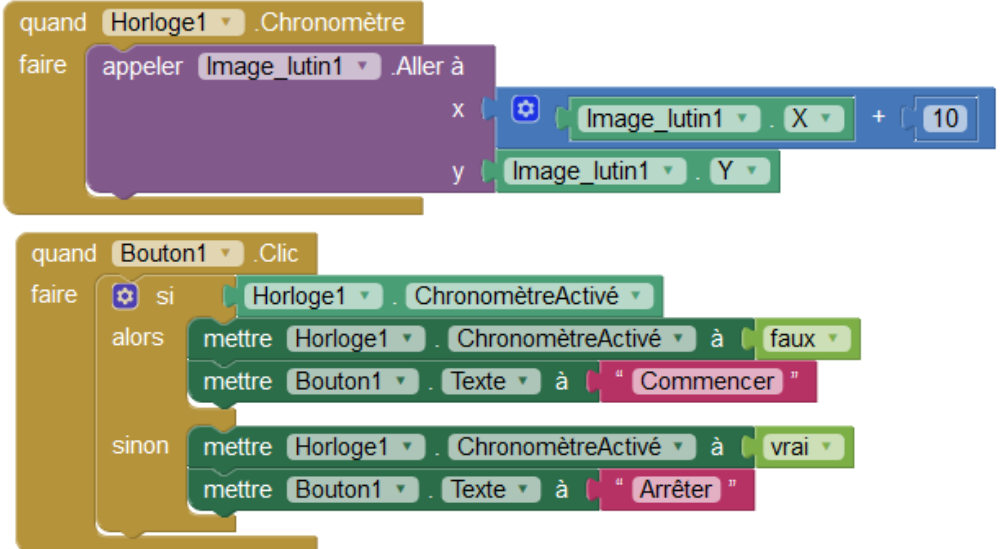


Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

* Cadre, Image lutin, Horloge, Bouton

Éditeur de blocs



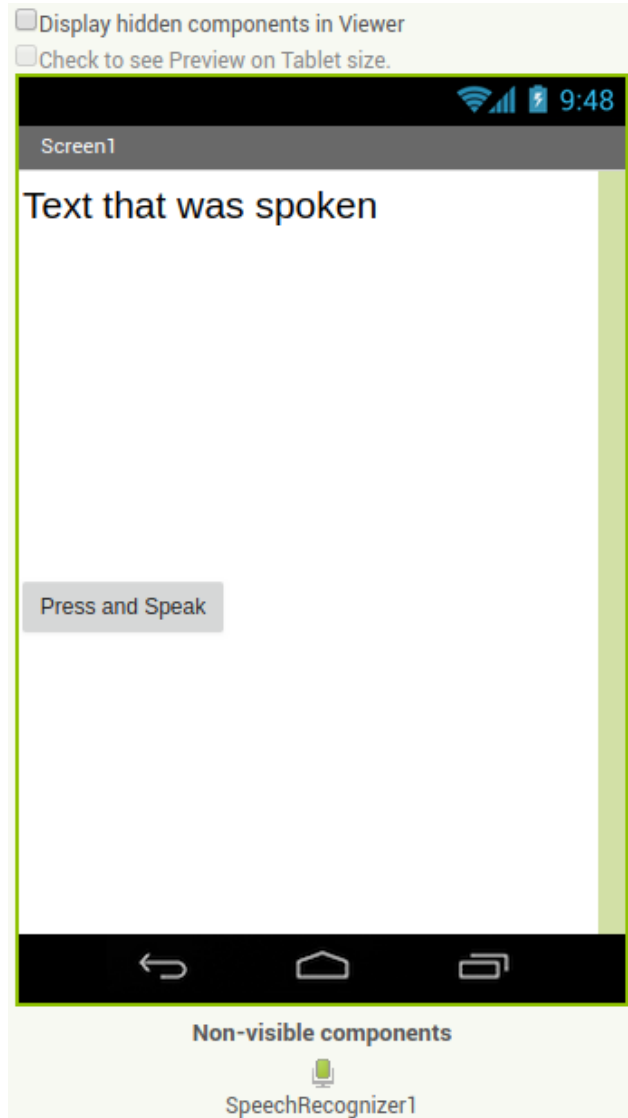
Ça veut dire quoi?

Quand **Bouton1.Click** est cliqué, si l'horloge est activée, alors arrêter le Chronomètre et afficher Start sur le bouton. Cela arrête le déplacement de l'Image lutin. L'inverse survient si le Chronomètre n'est pas en marche.



Reconnaissance vocale

Afficher ce qui est dit en texte sur l'écran de téléphone.



Préparation

Vous aurez besoin des composants suivants sur votre écran de conception

* Label, Bouton, Reconnaissance vocale

Éditeur de blocs



Ça veut dire quoi?

Quand vous cliquez sur **Bouton1** la **Reconnaissance vocale** est déclenchée et est prête pour que vous parliez.

Avant réception du texte est déclenchée avant que les paroles ne soient reçues et reconnues.

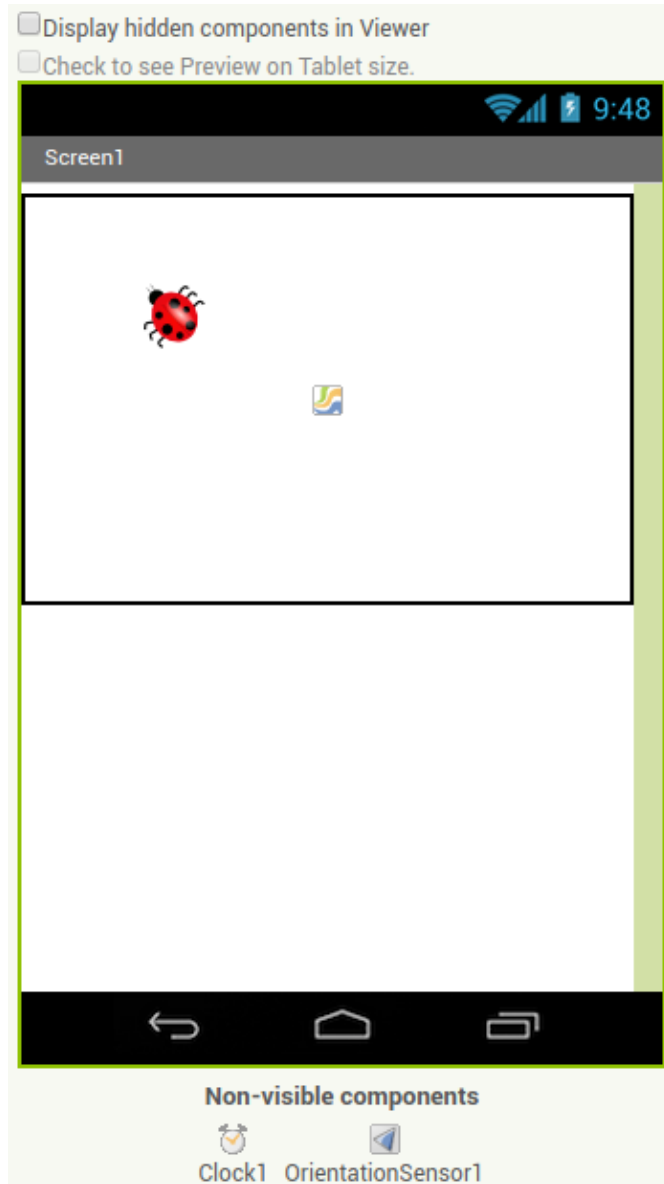
Le **Label** affiche un texte vide à l'écran.

Après obtention texte est déclenchée une fois les paroles reçues et reconnues. Le **Label** affiche alors le texte à l'écran.



Mouvement avec des Capteurs

Déplacez l'Image lutin en inclinant votre téléphone.

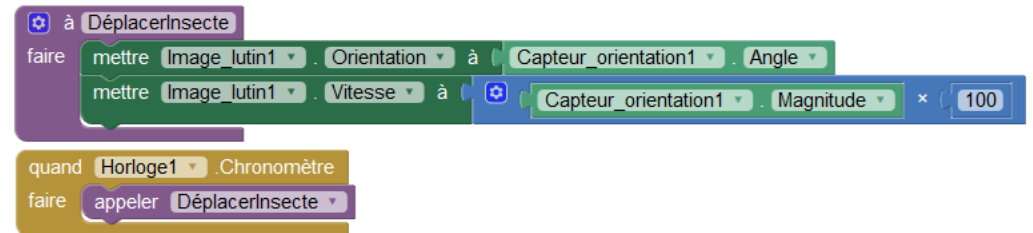


Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

* Cadre, Image lutin, Capteur d'orientation, Horloge

Éditeur de blocs



Ça veut dire quoi?

Une procédure appelée **DéplacerInsecte** a été créée et déplace l'insecte dans la direction vers laquelle le téléphone est incliné.

Le **Capteur_orientation1.Angle** est utilisé pour dire à l'insecte dans quelle direction se déplacer en se basant sur l'angle d'inclinaison de votre téléphone.

Le **Capteur_orientation1.Magnitude** est utilisé pour dire à l'insecte à quelle vitesse se déplacer en se basant sur le degré d'inclinaison de votre téléphone.

Quand **Horloge1.Chronomètre** est déclenché, l'instruction **DéplacerInsecte** débute.



Nombres aléatoires

Générer des nombres aléatoires pour faire apparaître l'Image lutin à des Coordonnées (x.y) aléatoires.

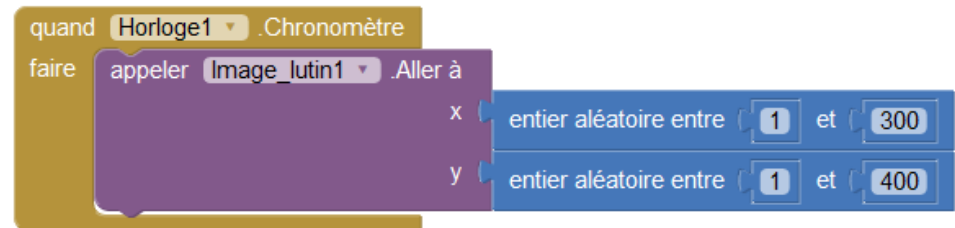


Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

- * Cadre
- * Image lutin
- * Horloge

Éditeur de blocs



Ça veut dire quoi?

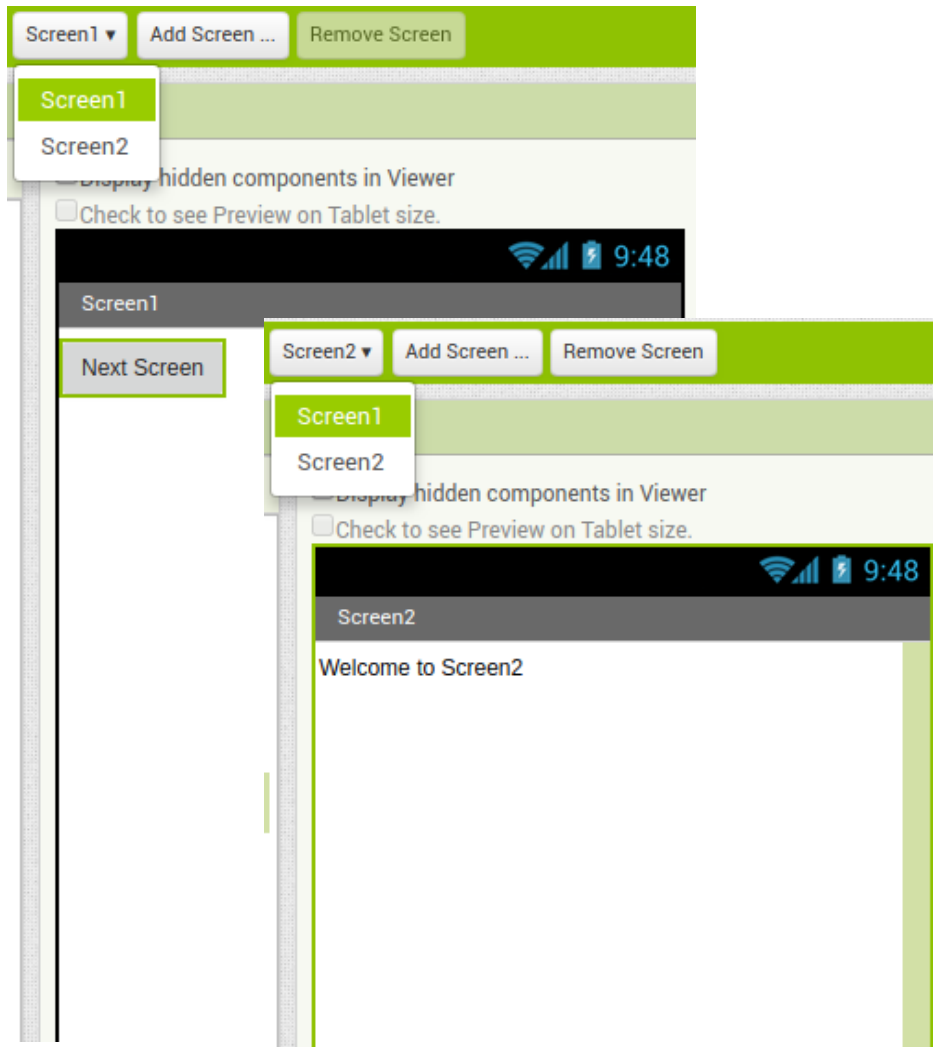
Quand l'évènement **Horloge1.Chronomètre** est déclenché, **Image_lutin1.Aller à** déplace l'Image lutin à des coordonnées aléatoires sur des valeurs comprises entre 1 et 300 pour l'abscisse x et 1 et 400 pour l'ordonnée y.



Écrans multiples

Utiliser de multiples écrans (Screens) dans votre application.

Aller à l'écran suivant en appuyant sur un Bouton.



Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

- * Screen1 (écran1): Bouton
- * Screen2 (écran2): Label

Éditeur de blocs

Pour Screen1 (écran1)



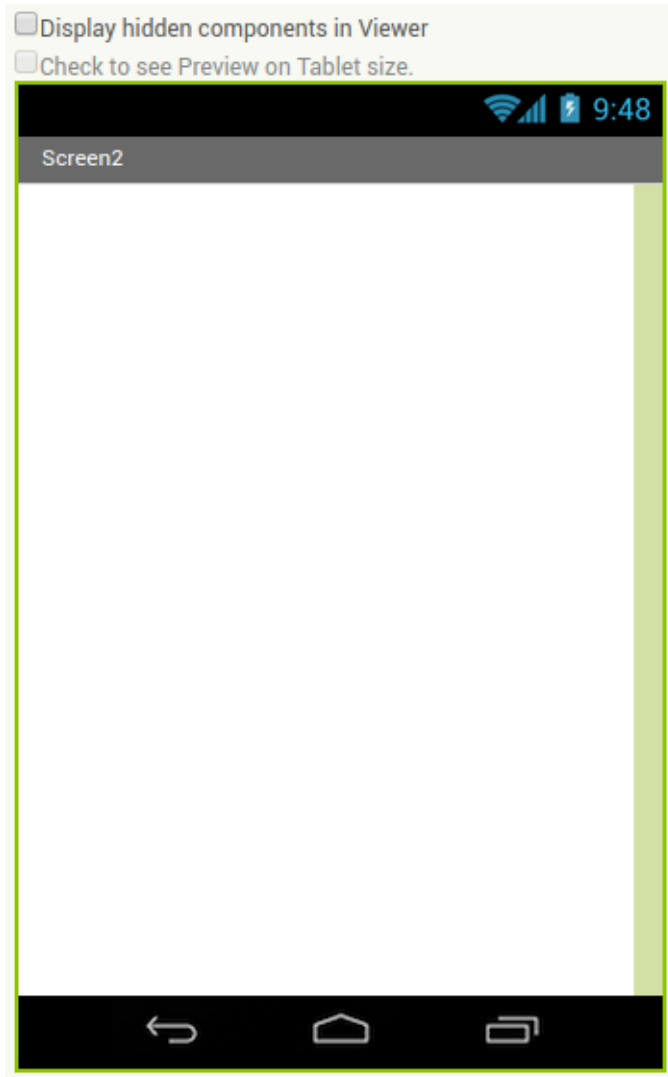
Ça veut dire quoi?

Ouvre un autre écran prend comme argument un bloc de texte. Le texte de ce bloc est le nom d'un autre écran. Quand le bouton est cliqué, Screen2 (écran2) s'ouvre.



Créer votre propre couleur

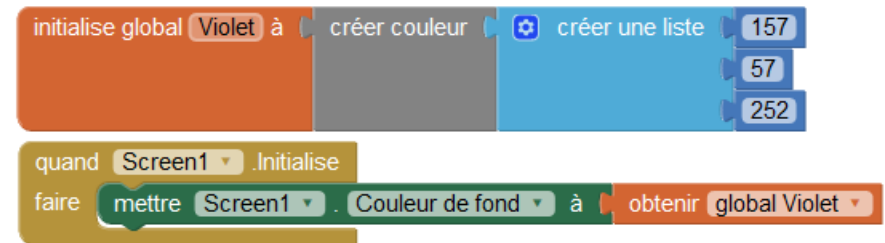
Créer vos propres couleurs en utilisant le bloc "créer une couleur"



Préparation

Aucun composant de conception n'est nécessaire pour utiliser "créer une couleur"

Éditeur de blocs



Ça veut dire quoi?

- **Créer couleur** utilise une liste de 4 nombres. Les trois premiers nombres représentent les valeurs RGB (Red, Green, Blue = Rouge, Vert, Bleu). Le dernier nombre est l'alpha et détermine l'intensité de la couleur.
- **Violet** est créé en utilisant 157 = R, 57 = G, 252 = B et 100 = alpha.
- Quand **Screen1 (écran1)** est initialisé, la couleur de l'arrière-plan s'ajuste à la couleur que l'on vient de créer dans la variable "Purple".



Mouvement de balayage

Modifier la trajectoire et la vitesse de l'Objet en balayant l'écran avec vos doigts.

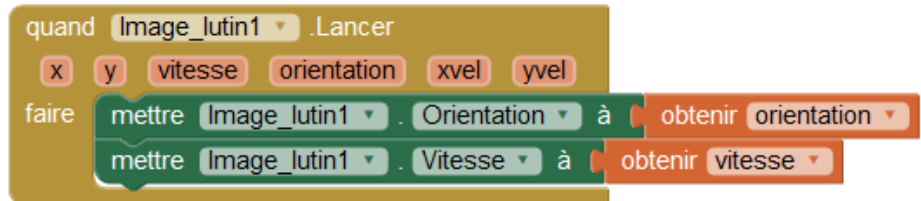


Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

- * Cadre
- * Image lutin

Éditeur de blocs



Ça veut dire quoi?

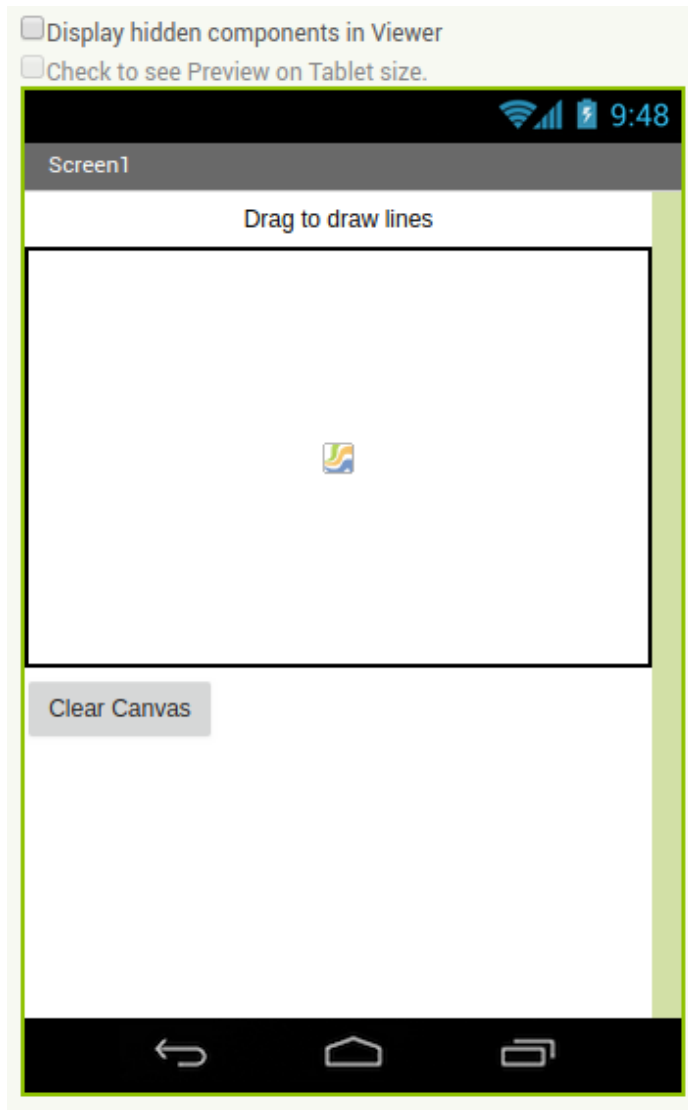
Lancer détecte lorsque l'utilisateur effectue un mouvement de balayage avec l'Image lutin sur l'écran.

L'utilisateur transmet la trajectoire et la vitesse du mouvement de balayage aux fonctions **Orientation** et **Vitesse** de l'Image lutin.



Dessiner dans un cadre

Faites glisser vos doigts sur l'écran pour Dessiner une courbe le long du chemin tracé par votre doigt.



Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

- * Cadre
- * Bouton
- * Label

Éditeur de blocs

```
quand Cadre1 .Glissé
  X début  Y début  X précédent  Y précédent  Xactuel  Yactuel  draggedAnySprite
faire
  appeler Cadre1 .Dessiner ligne
    x1 obtenir X précédent
    y1 obtenir Y précédent
    x2 obtenir Xactuel
    y2 obtenir Yactuel

quand Réinitialiser .Clic
faire
  appeler Cadre1 .Supprimer
```

Ça veut dire quoi?

Quand **Cadre1.Glissée** est déclenché, une courbe est dessinée de là où votre doigt est parti jusqu'à l'endroit où il a été amené.

Quand le bouton **Réinitialiser.Clic** est cliqué, le cadre est effacé.



Secouer le téléphone

Déclencher quelque chose quand vous secouez votre téléphone.

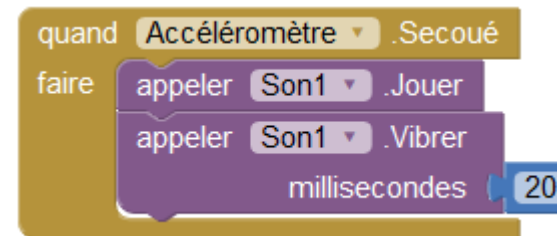


Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

* Image, Son, Accéléromètre, Label

Éditeur de blocs



Ça veut dire quoi?

L'instruction **Accéléromètre.Scoué** détecte quand le téléphone est secoué. Le Son est alors lu et le téléphone vibre pendant 20 millisecondes.



Détection de collision

Déclencher quelque chose quand l'Image lutin 1 entre en collision avec l'Image lutin 2.



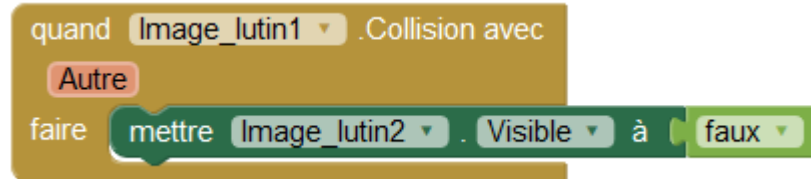
Préparation

Vous aurez besoin des composants suivants sur votre écran de conception

- * Cadre
- * 2 Images lutin
- * 4 Boutons

Éditeur de blocs

Astuce: Pour faire bouger votre coccinelle (Image_lutin1) en cliquant sur les Boutons, consulter les cartes de Mouvement.



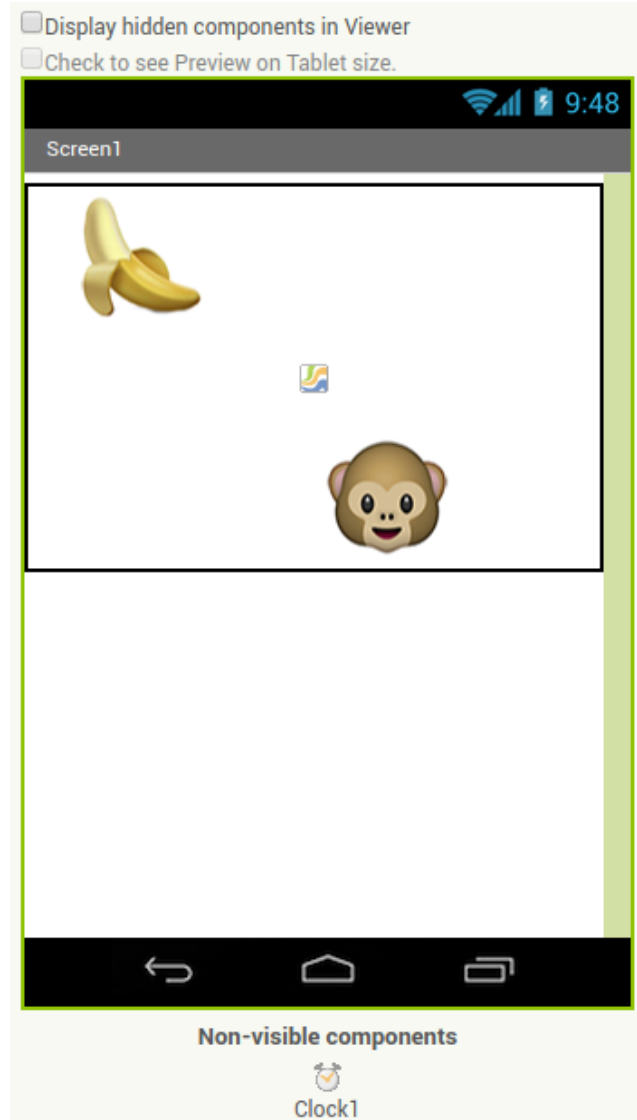
Ça veut dire quoi?

Image_lutin1.Collision est déclenché quand la Coccinelle (Image_lutin1) entre en collision avec le Puceron (Image_lutin2). Cela fait disparaître le Puceron.



Déplacer l'Image lutin

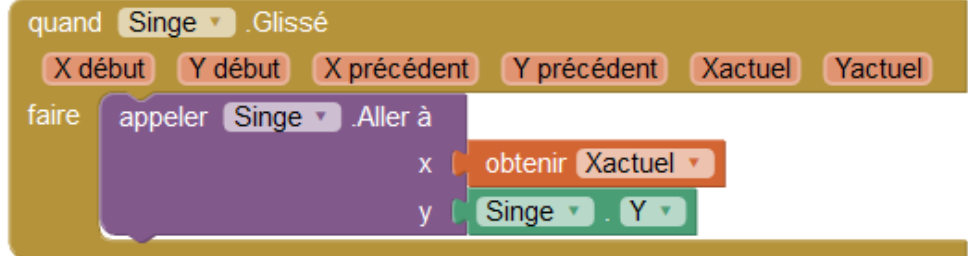
Déplacer l'Image lutin d'un côté à l'autre en faisant glisser votre doigt dessus.



Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

* Cadre, Image lutin, Horloge
Éditeur de blocs



Ça veut dire quoi?

Quand l'utilisateur déplace l'Image lutin (Singe) avec le doigt, l'instruction **Singe.Glissé** est déclenchée à de multiples reprises. Chaque appel (call) a ici 6 arguments:

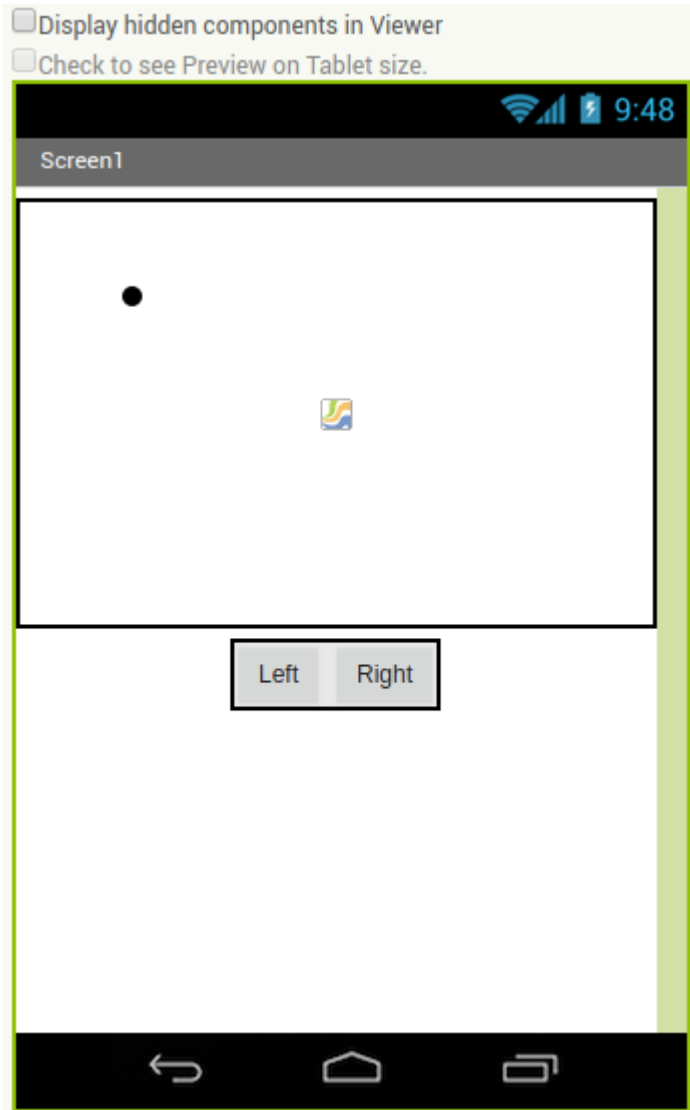
- **X début** et **Y début**, où l'utilisateur a initialement touché l'écran.
- **Xactuel** et **Yactuel**, où l'utilisateur touche actuellement l'écran
- **X précédent** et **Y précédent** détiennent les valeurs qui étaient dans **Xactuel** et **Yactuel** à l'appel précédent de l'évènement. (Lors du premier appel de cet évènement, **X précédent** et **Y précédent** sont identiques à **X début** et **Y début**.)

Quand l'utilisateur déplace l'Image lutin (Singe), elle est déplacée vers la nouvelle position **X**, **Xactuel**, du déplacement. Le **Y** reste le même de manière à ce que le Singe ne puisse se déplacer que dans la direction **X**.



Mouvement avec des Boutons

Déplacer la Balle en touchant des boutons.

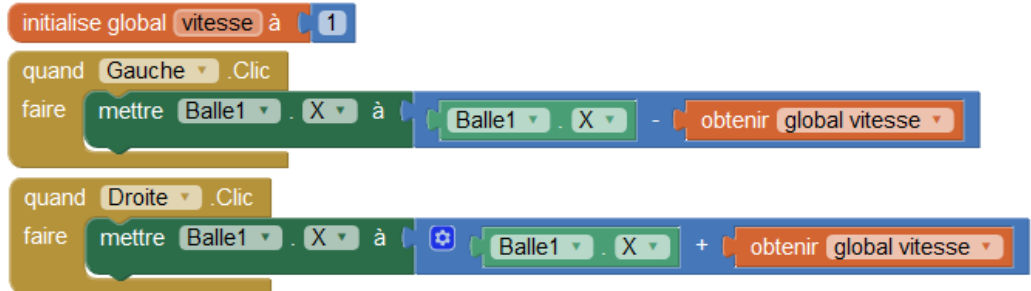


Préparation

Vous aurez besoin des composants suivants sur votre écran de conception:

* Cadre, Balle, 2 Boutons

Éditeur de blocs



Ça veut dire quoi?

Définir une variable **vitesse** à 1 pour déterminer la distance parcourue par la balle à chaque fois que le bouton est cliqué.

L'évènement **Gauche.Clic** déplace la balle vers la gauche à chaque fois que le bouton est cliqué.

L'évènement **Droite.Clic** déplace la balle vers la droite à chaque fois que le bouton est cliqué.