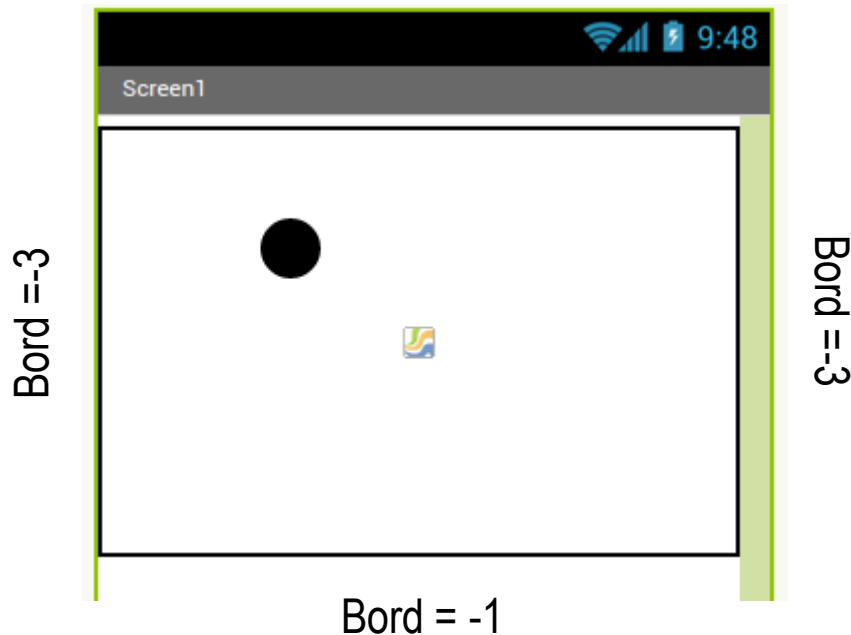




Faire rebondir une Balle sur le bord du cadre

Faire rebondir une Balle lorsqu'elle touche le bord du 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:

- * Canvas
- * Sprite

Éditeur de blocs

```
when Ball1 .EdgeReached
  edge
do call Ball1 .Bounce
  edge get edge
```

Ça veut dire quoi?

Ball1.EdgeReached détecte le moment où l'objet "Ball" atteint le bord du cadre et fournit l'argument **edge**.

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

Ball1.Bounce fait rebondir l'objet dans la direction opposée au mur (**Balle1.Heading** 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:

- * Button
- * Sound
- * Player

Éditeur de blocs

```
when Button1.Click  
do call Meow.Play
```

```
when Screen1.Initialize  
do call BackgroundMusic.Start
```

Ça veut dire quoi?

Quand **Button1** est cliqué, le **Son Meow** est lu.
Quand **Screen1** s'affiche, la lecture de **BackgroundMusic** débute.



Mouvement programmé par un chrono

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

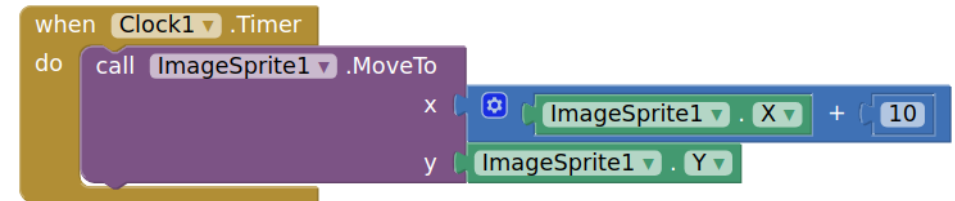


Préparation

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

- * Canvas
- * ImageSprite
- * Clock

Éditeur de blocs



Ça veut dire quoi?

MoveTo 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).

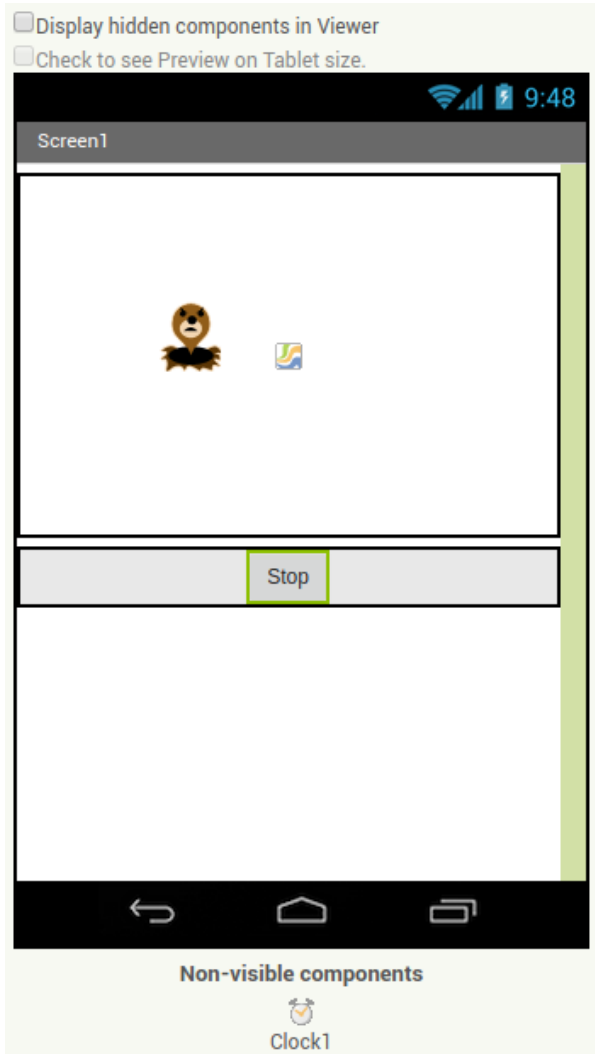
Clock1 a un Intervalle qui indique la Fréquence à laquelle le Chronomètre doit se déclencher.

À chaque fois que **Clock1.Timer** 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'Objet en fonction du temps.

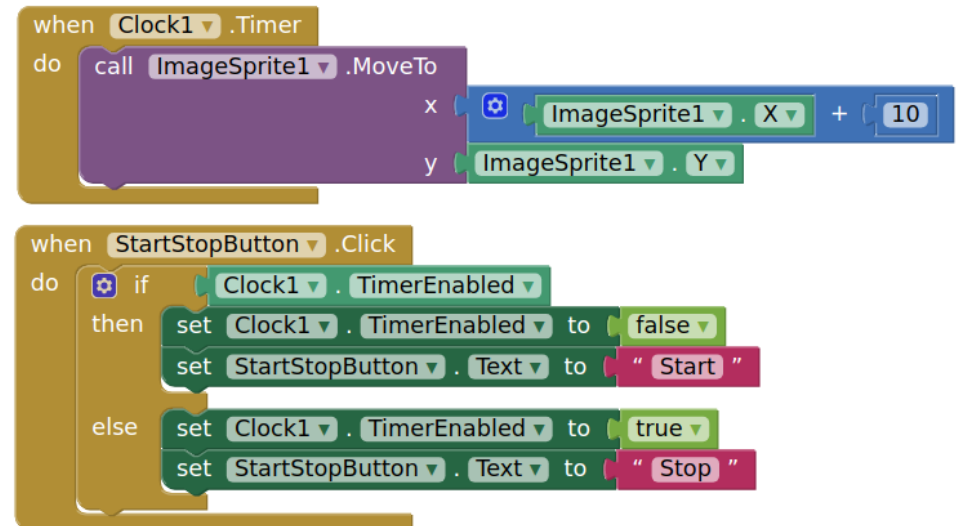


Préparation

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

* Canvas, ImageSprite, Clock, Button

Éditeur de blocs



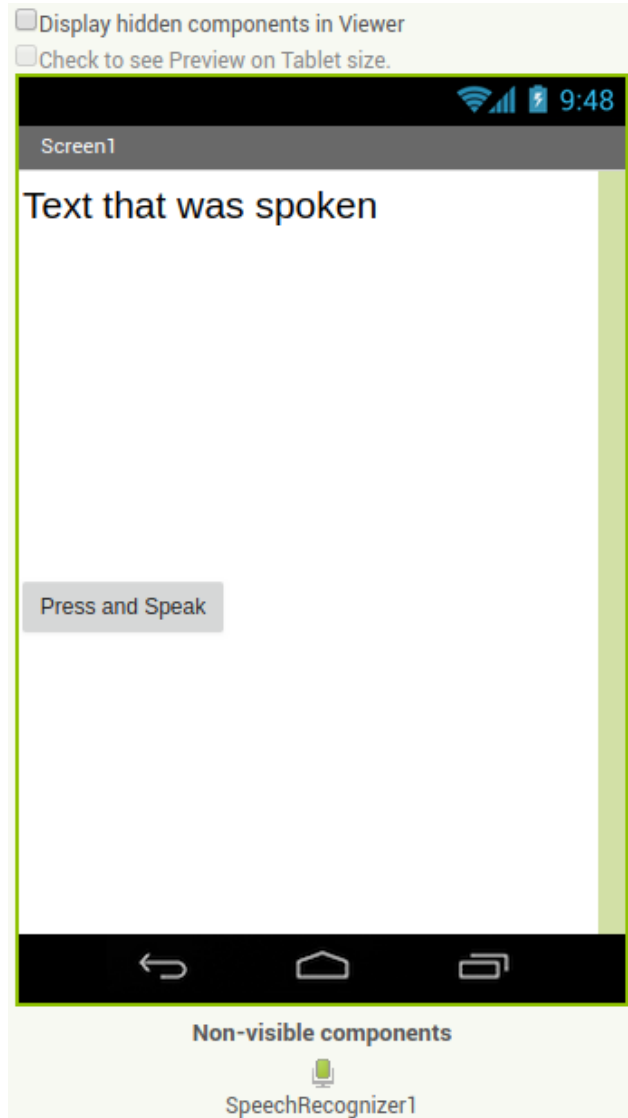
Ça veut dire quoi?

Quand **StartStopButton.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 du téléphone.



Préparation

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

* Label, Button, SpeechRecognizer

Éditeur de blocs

```
when PressAndSpeakButton .Click
```

```
do call SpeechRecognizer1 .GetText
```

```
when SpeechRecognizer1 .AfterGettingText
```

```
result
```

```
do set Label1 . Text to SpeechRecognizer1 . Result
```

```
when SpeechRecognizer1 .BeforeGettingText
```

```
do set Label1 . Text to ""
```

Ça veut dire quoi?

Quand vous cliquez **PressAndSpeakButton** le **SpeechRecognizer** est déclenché et est prêt pour que vous parliez.

L'instruction **BeforeGettingText** est déclenchée avant que les paroles ne soient reçues et reconnues.

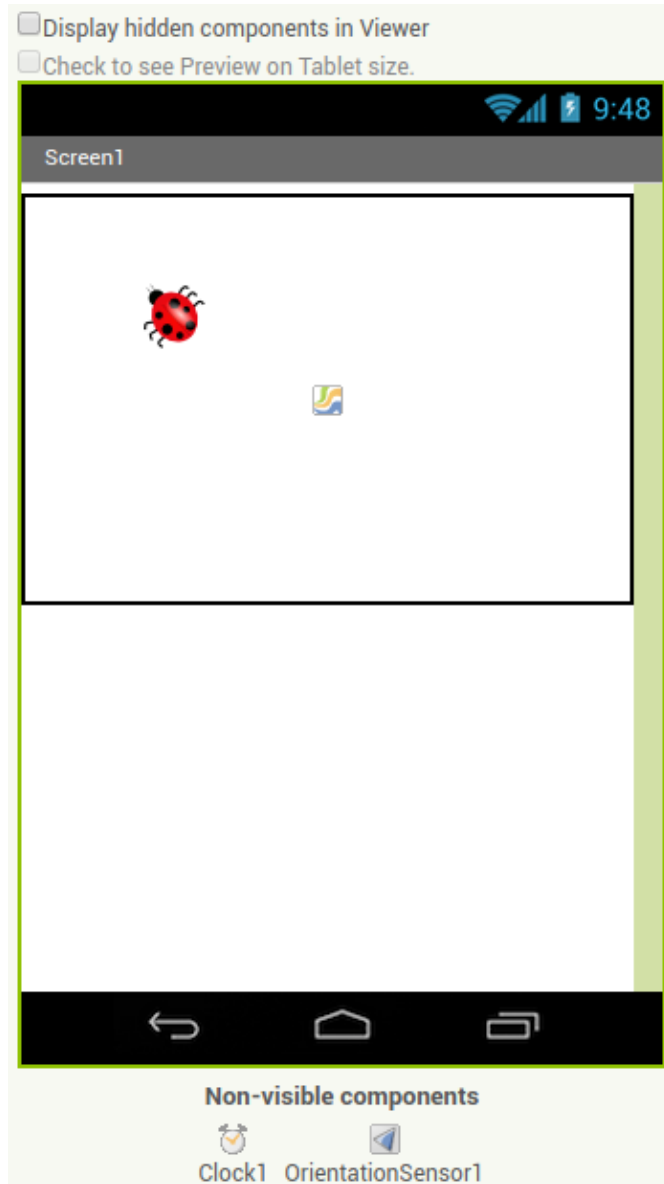
Le Label affiche un texte vide à l'écran.

L'instruction **AfterGettingText** 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:

* Canvas, ImageSprite, OrientationSensor, Clock

Éditeur de blocs

```
to MoveBug
do
  set Bug . Heading to OrientationSensor1 . Angle
  set Bug . Speed to OrientationSensor1 . Magnitude × 100

when Clock1 . Timer
do
  call MoveBug
```

Ça veut dire quoi?

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

Le **OrientationSensor.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 **OrientationSensor.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 l'instruction **Clock1.Timer** est déclenchée, l'instruction **MoveBug** débute.



Nombres aléatoires

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

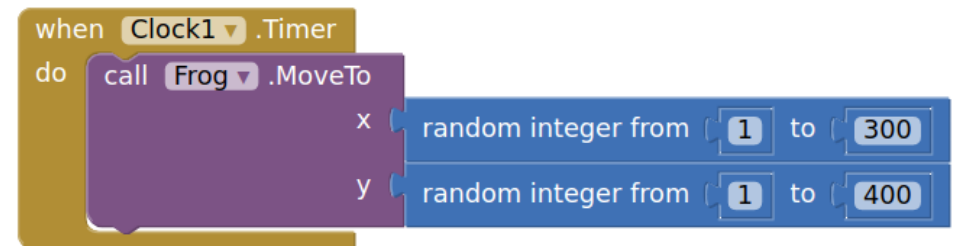


Préparation

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

- * Canvas
- * ImageSprite
- * Clock

Éditeur de blocs



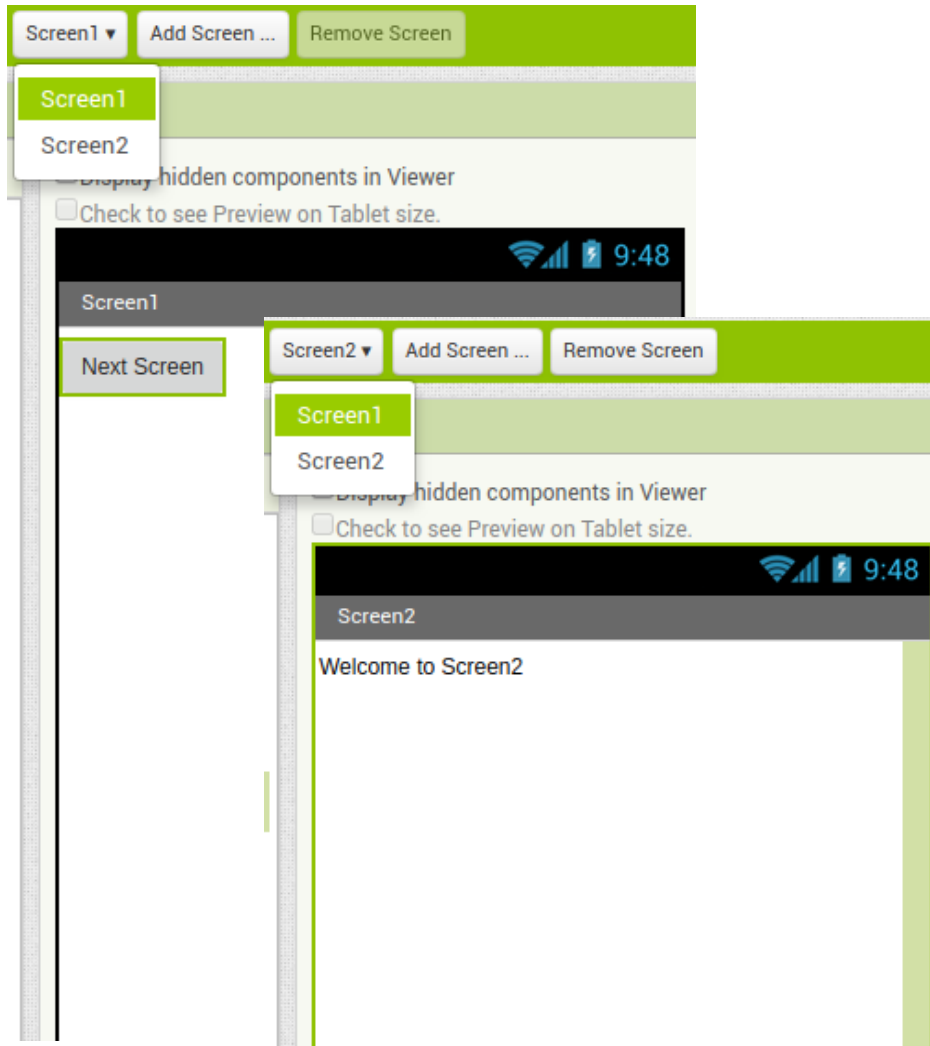
Ça veut dire quoi?

Quand l'évènement **Clock1.Timer** est déclenchée, **Frog.MoveTo** déplace la grenouille 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 dans votre application.
Aller à l'écran suivant en appuyant sur un bouton.



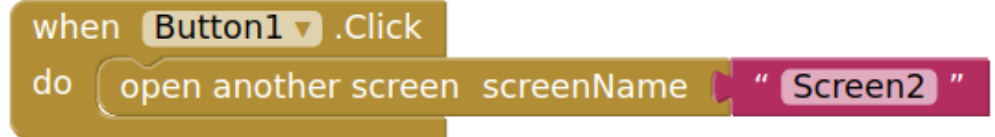
Préparation

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

- * Screen1: Button
- * Screen2: Label

Éditeur de blocs

Pour Screen1:



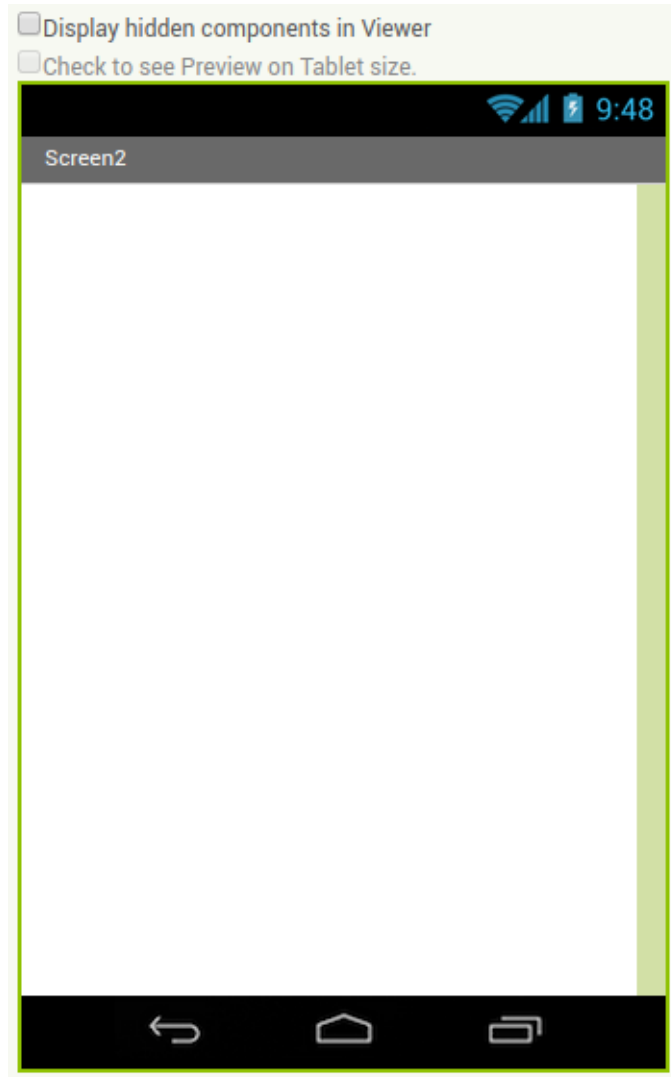
Ça veut dire quoi?

Open another screen prend comme argument un bloc de texte. Le texte de ce bloc est le nom d'un autre écran. Quand le bouton est cliqué, l'écran2 s'ouvre.



Créer votre propre couleur

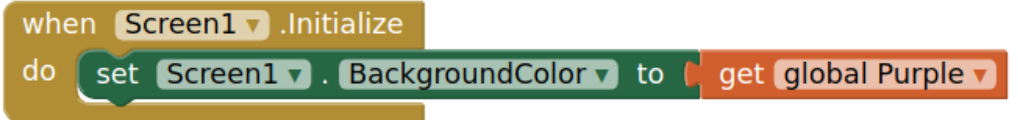
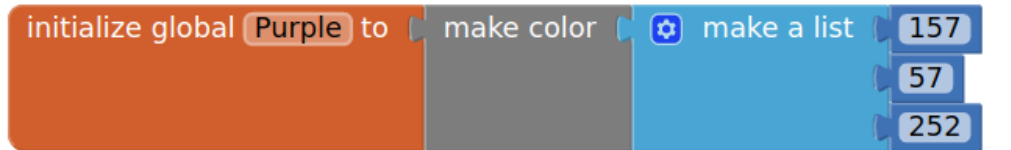
Créer vos propres couleurs en utilisant le bloc "make color"



Préparation

Aucun composant nécessaire pour utiliser "make color"

Éditeur de blocs



Ça veut dire quoi?

Make color 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.

Purple est créé en utilisant 157 = R, 57 = G, 252 = B et 100 = alpha.

Quand **Screen1** 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'Image lutin en balayant l'écran avec vos doigts.



Préparation

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

- * Canvas
- * ImageSprite

Éditeur de blocs

```
when PirateSprite .Flung
  x y speed heading xvel yvel
do
  set PirateSprite . Heading to get heading
  set PirateSprite . Speed to get speed
```

Ça veut dire quoi?

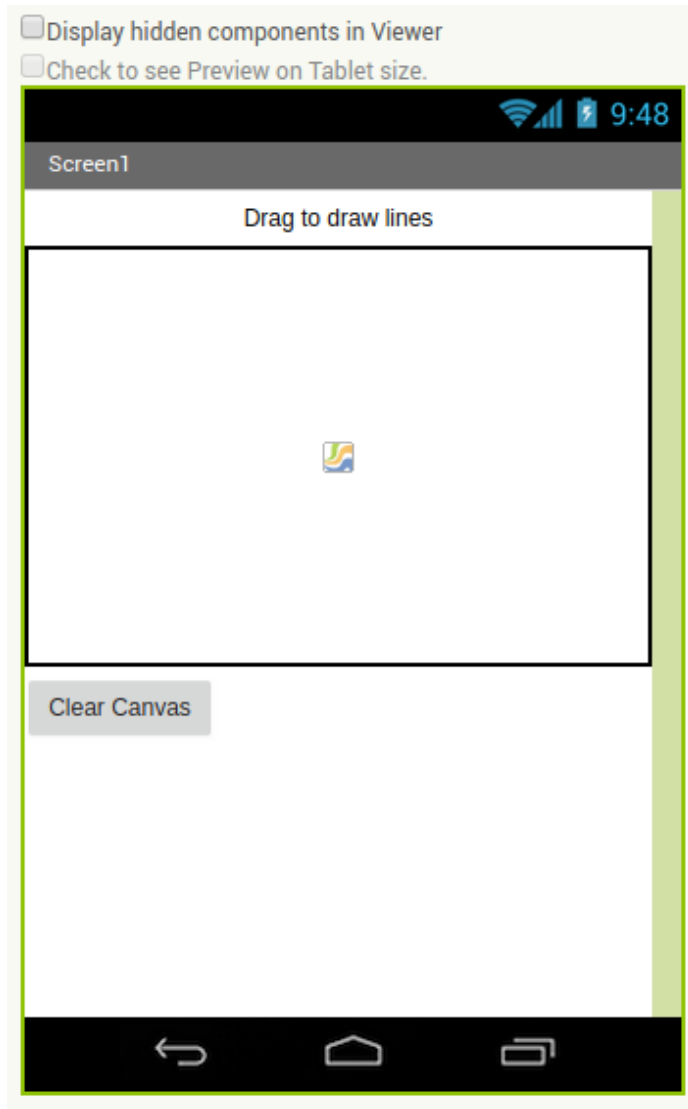
Flung détecte quand l'utilisateur effectue un mouvement de balayage avec l'Image lutin sur l'écran.

L'utilisateur transmet la trajectoire et la vitesse du mouvement de balayage aux fonctions **Heading** et **Speed** 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:

- * Canvas
- * Button
- * Label

Éditeur de blocs

```
when Canvas1 .Dragged
  startX  startY  prevX  prevY  currentX  currentY  draggedAnySprite
do
  call Canvas1 .DrawLine
    x1 get prevX
    y1 get prevY
    x2 get currentX
    y2 get currentY
```

```
when ClearButton .Click
do
  call Canvas1 .Clear
```

Ça veut dire quoi?

Quand **Canvas1.Dragged** 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 **Clear.Click** 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, Sound, AccelerometerSensor, Label

Éditeur de blocs

```
when AccelerometerSensor1 .Shaking
do
  call Meow .Play
  call Meow .Vibrate
  milliseconds 20
```

Ça veut dire quoi?

AccelerometerSensor.Shaking détecte quand le téléphone est secoué. Le Son (Meow) 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:

- * Canvas
- * Sprite
- * Button

Éditeur de blocs

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

```
when LadyBug .CollidedWith  
  other  
do set Aphid . Visible to false
```

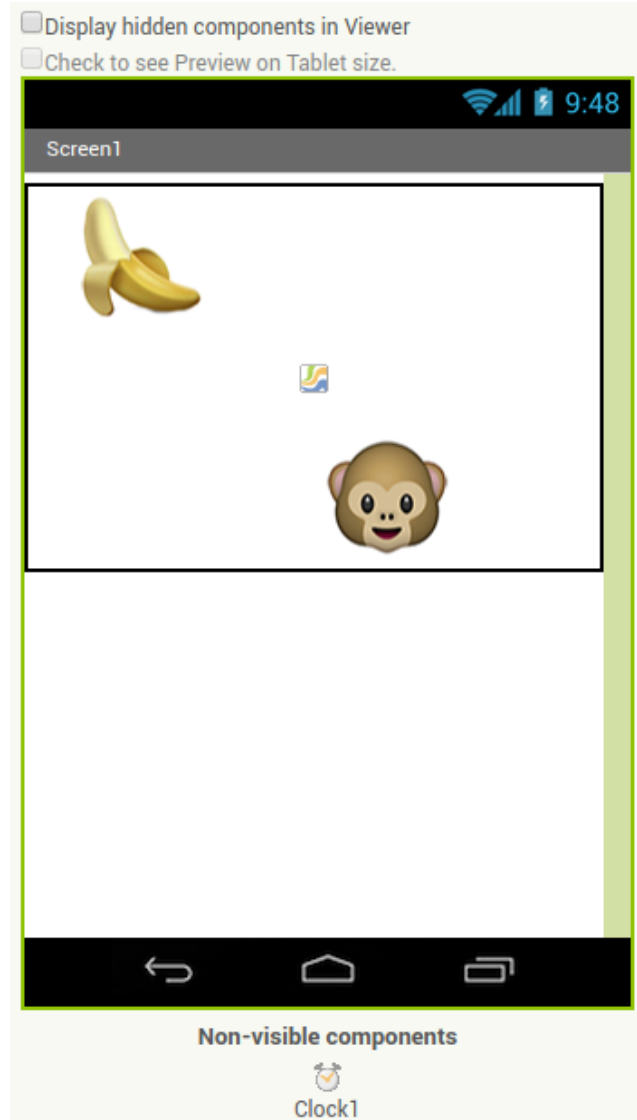
Ça veut dire quoi?

LadyBugCollidedWith est déclenché quand la Coccinelle (Ladybug) entre en collision avec le Puceron (Aphid). 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:

* Canvas, ImageSprite, Clock
Éditeur de blocs

```
when MonkeySprite .Dragged
  startX  startY  prevX  prevY  currentX  currentY
do
  call MonkeySprite .MoveTo
    x  get currentX
    y  MonkeySprite . Y
```

Ça veut dire quoi?

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

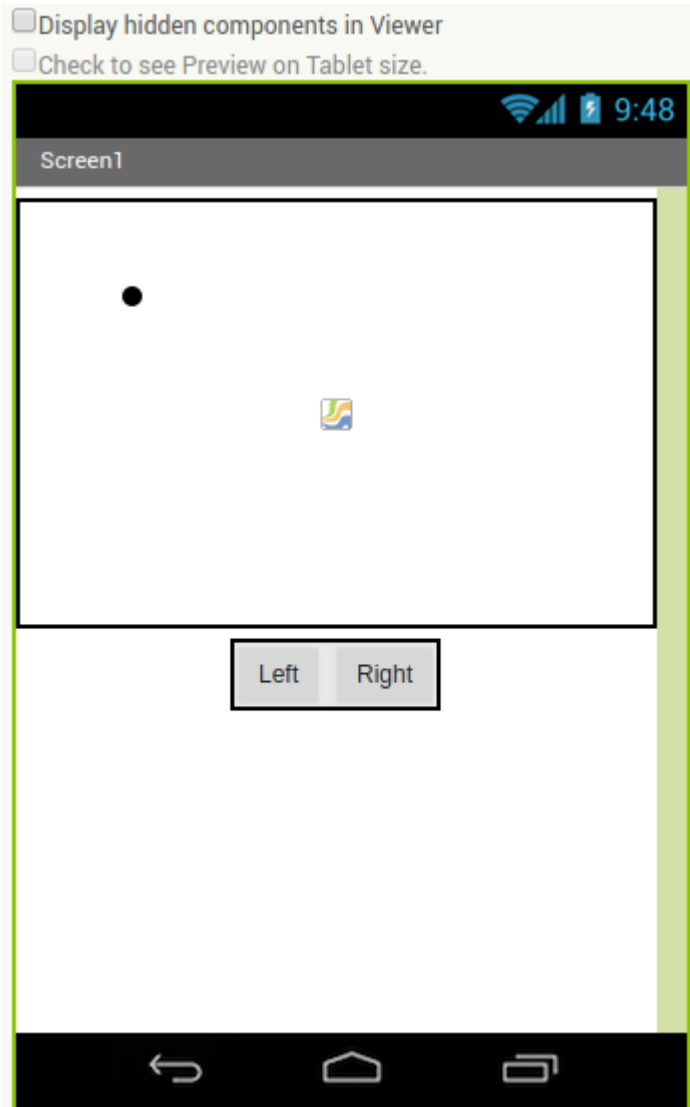
- **startX** et **startY**, où l'utilisateur a initialement touché l'écran.
- **currentX** et **currentY**, où l'utilisateur touche actuellement l'écran.
- **prevX** et **prevY** détiennent les valeurs qui étaient dans **currentX** et **currentY** à l'appel (call) précédent de l'évènement. (Lors du premier appel de cet évènement, **prevX** and **prevY** sont identiques à **startX** et **startY**.)

Quand l'utilisateur déplace l'Image lutin (Singe), elle est déplacée vers la nouvelle position **X**, **currentX**. 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:

* Canvas, Sprite, Button

Éditeur de blocs

```
initialize global speed to 1
when Left.Click
do set Ball1.X to Ball1.X - get global speed
when Right.Click
do set Ball1.X to Ball1.X + get global speed
```

Ça veut dire quoi?

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

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

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