

# Tutoriel - App Inventor : balle rebondissante

TOUTES LES FICHES



 <b>PUBLIC</b>	 <b>PARTICIPANTS</b>	 <b>ANIMATEURS</b>	 <b>NIVEAU</b>	 <b>PRÉPARATION</b>	 <b>ACTIVITÉ</b>
--	--	--	--	---	--

## Description

---

–

## Objectifs

---

Nous allons ici réaliser une application pour Android où l'on doit détruire un ennemi avec une petite balle. Lorsque la balle touche l'ennemi, elle le détruit et le score augmente.

Commencez par vous connecter à App Inventor, créez un nouveau projet, et appelez-le « Balle Rebondissante ».

## Compétences travaillées

---

- Événement ; Variables
- Mathématiques
- Label ; Bouton ; Image Sprite

## Pré-requis

---

Connaître les bases d'App Inventor (avoir fait la fiche TalkToMe).

Avoir un appareil Android (attention certaines tablettes ne sont pas équipées de capteurs).

Avoir un compte Gmail.

## Matériel

---

- ordinateur avec connexion internet (optionnel)
- smartphone avec connexion internet

## Contenus utilisés

---

App Inventor : <http://ai2.appinventor.mit.edu/>

## WORKFLOW

### 1 Le design

#### Etape 1 : Paramétrer l'écran (Screen)

Tout d'abord, occupons-nous des paramètres de l'écran en allant dans la partie « Properties » tout à droite.

Il faut changer le titre de l'écran pour qu'il affiche « balle rebondissante », voici comment faire :



#### Etape 2 : Ajouter un texte (Label)

Allez dans « Palette/Drawing and Sensors » et importez un tableau (Canvas) et modifiez sa taille pour qu'il remplisse l'écran entier:



En modifiant la taille de la toile sur « Fill parent », celle-ci va occuper tout l'espace disponible de l'élément qui la contient (ici l'écran).

#### Etape 4 : Ajouter une balle (Ball)

Depuis la partie gauche Palette, ajouter une balle (ball) vous la trouverez dans la partie « Drawing and Animations ». Changez son rayon pour le mettre à 15 :

Radius

15

### Etape 5 : Ajouter un bouton cliquable (Button)

Depuis la partie gauche Palette/User Interface, **ajoutez un bouton cliquable** (Button) à votre application. Lorsque l'utilisateur appuiera dessus, le score se réinitialisera. **Changez son nom** en "Reset" dans Properties.

Text

Reset

### Etape 6 : Ajouter une image (ImageSprite)

Importez une image de petit monstre (venant de pixabay.fr par exemple) via le menu sur la droite:

Media

Upload File ...

Ensuite dans "Palette/Drawing and Sensors", importez une image (ImageSprite) changez "Picture" par votre image. Changez sa taille pour par exemple 50 par 50 pixels:

Height

50 pixels...

Width

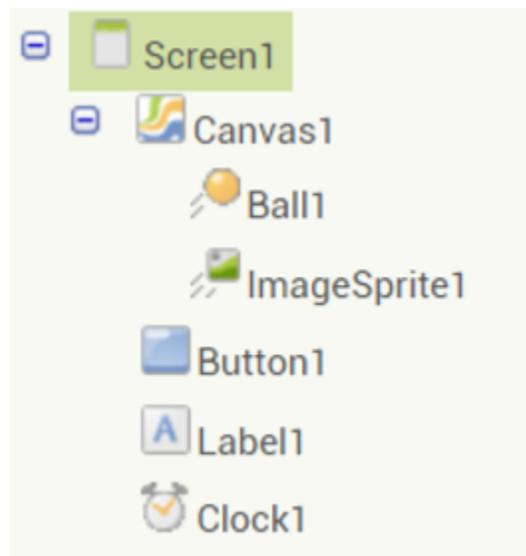
50 pixels...

Enfin allez dans Palette/Sensors et importez une horloge (Clock).

Votre application devrait ressembler à cela :



Les composantes de votre application

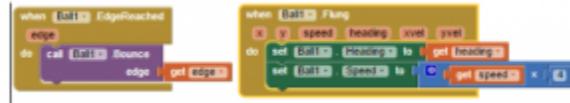


## 2 La programmation

Nous allons maintenant programmer l'application. Rendez-vous dans la partie Blocks. Si vous ne trouvez pas un des blocs utiles, n'hésitez pas à consulter l'annexe.

## Etape 1 : Gérer le comportement de la balle

Nous allons d'abord nous occuper d'avoir une balle qui bouge quand on tire dessus et qui rebondit contre les murs, créez les blocs suivant :



Quand la balle touche une extrémité (Edge), on la fait rebondir (Bounce) dessus. Quand on glisse le doigt sur la balle, on récupère sa position (x et y), sa vitesse (speed), sa direction (Heading) et la vitesse horizontale et verticale du doigt (xvel et yvel). On la fait se diriger dans le sens du mouvement (Heading) puis on modifie sa vitesse (Speed), nous la multiplions par 4 pour rendre les tirs plus impressionnantes.

## Etape 2 : Interaction avec le petit monstre

Quand le petit monstre est touché par la balle, il disparaît et on augmente le score. Toutes les secondes, le petit monstre change d'endroit:



On initialise le score à 0. Quand l'image rentre en collision avec la balle (CollidedWith), on la cache (Visible = false), on augmente le score de 1 et on change l'étiquette qui montre le score. Chaque seconde (When Clock1.Timer do), on change la position de l'image via les deux premières lignes vers une position aléatoire (random integer renvoie un entier aléatoire entre 1 et le deuxième chiffre, ici la taille de l'écran ) puis on la rend visible.

## Etape 3 : Remettre à zéro le score

On utilise le bouton "Reset" pour remettre le score à 0:



Quand on clique sur le Bouton 1 (Click), on remet le score à 0 et on met à jour l'étiquette de score.

Voilà votre application est terminée !

### 3 Pour développer

Voici une liste (non exhaustive bien sûr) des améliorations possibles :

- Améliorer l'interface graphique.
- Ajouter d'autres ennemis.
- Modifier la vitesse de la balle, de l'ennemi.
- Rajouter un parcours.

### 4 Annexe

Nom du bloc	Où le trouver ?
<ul style="list-style-type: none"> <li>• When Ball1 .EdgeReached do</li> <li>• When Ball1 .Fling do</li> </ul>	Ball1
<ul style="list-style-type: none"> <li>• Call Ball1 .Bounce edge</li> </ul>	
<ul style="list-style-type: none"> <li>• Set Ball1 .Heading to</li> <li>• Set Ball1 .Speed to</li> </ul>	

<ul style="list-style-type: none"> <li>• Get edge</li> </ul>	Appuyez sur le bouton « edge » du bloc « When Ball1 .EdgeReached do »	
<ul style="list-style-type: none"> <li>• Get heading</li> <li>• Get speed</li> </ul>	Appuyez sur les boutons « heading » et « speed » du bloc « When Ball1 .EdgeReached do »	
<ul style="list-style-type: none"> <li>• Pour les symboles mathématiques</li> <li>• Pour les chiffres</li> <li>• Random integer from</li> <li>• Initialize global score to</li> <li>• Set global score</li> <li>• Get global score</li> </ul>	Math	
<ul style="list-style-type: none"> <li>• When ImageSprite1 .CollidedWith do</li> </ul>	ImageSprite1	
<ul style="list-style-type: none"> <li>• Set ImageSprite1 .X to</li> <li>• Set ImageSprite1 .Y to</li> <li>• Set ImageSprite1 .Visible to</li> </ul>		
<ul style="list-style-type: none"> <li>• Set Label1 .Text to</li> </ul>		Label1

<ul style="list-style-type: none"> <li>• Join</li> <li>• « »</li> </ul>	Text
<ul style="list-style-type: none"> <li>• When Clock1 .Timer do</li> </ul>	Clock1
<ul style="list-style-type: none"> <li>• Screen1 .Width</li> <li>• Screen1 .Height</li> </ul>	Screen1
<ul style="list-style-type: none"> <li>• True</li> </ul>	Logic