

Nom :

Prénom :

TP SIN

Programmation

Support : App inventor 2

Support : App inventor 2

Pré requis (l'élève doit savoir):

- Savoir utiliser un ordinateur et un téléphone Android

Programme

Objectif terminal :

L'élève doit être capable de réaliser un programme Android et de le tester

Matériels :

- Logiciel App inventor
- Téléphone android

1. Travail demandé

Ressource : <http://appinventor.mit.edu/explore/ai2/molemash.html>

Dans le jeu MoleMash , une taupe apparaît à des positions aléatoires sur un terrain de jeu, et les joueurs marquent des points en frappant la taupe avant qu'elle saute loin. Ce tutoriel montre comment construire MoleMash comme un exemple d'un jeu simple qui utilise l'animation.

Voir le chapitre de livre NOUVEAU du App Inventor livre par Wolber, et al. (Comprend didacticiel vidéo)



Nom :

Prénom :

- Créer un nouveau projet sur app inventor

<http://ai2.appinventor.mit.edu/#6326882833268736>

- Récupérer l'image de la taupe ci-dessous

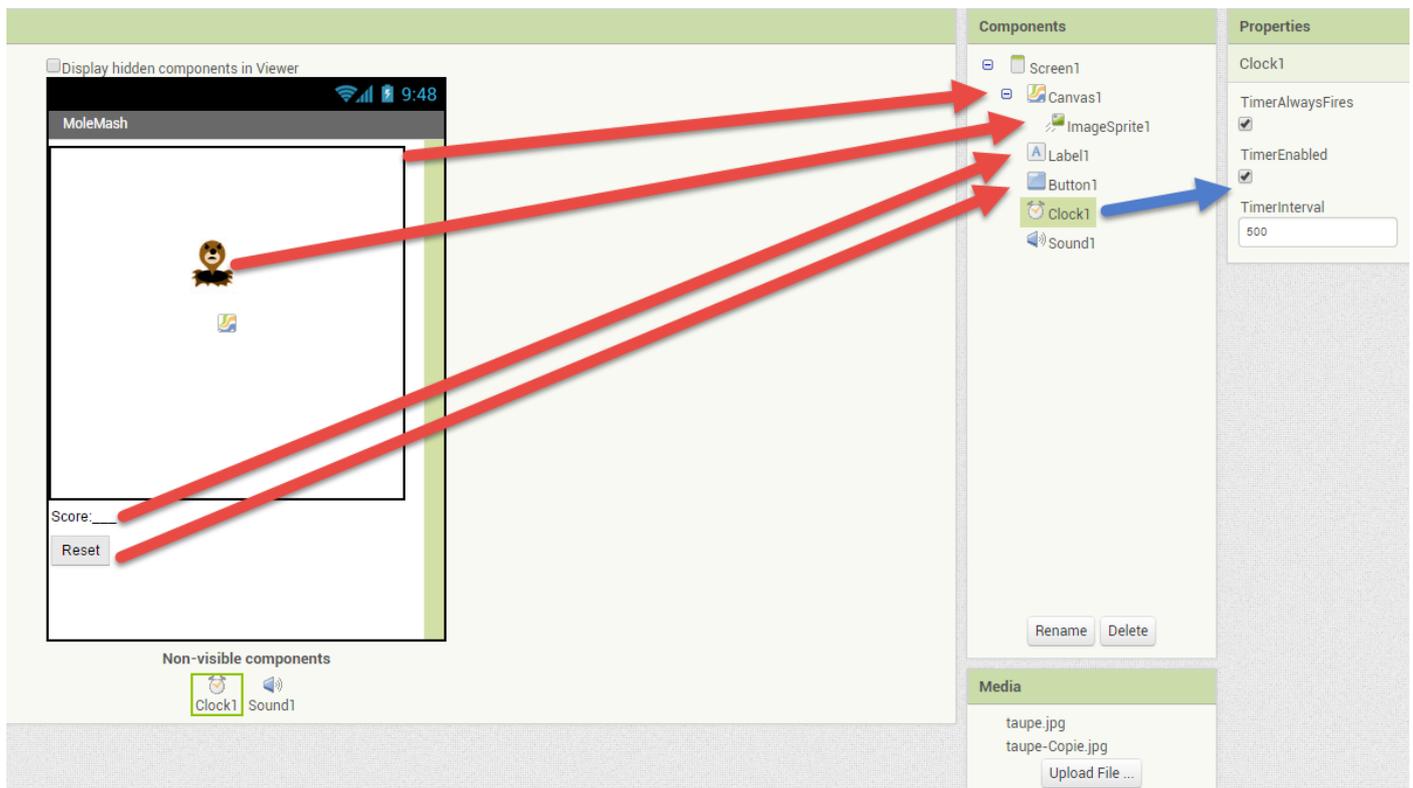


- On va concevoir le jeu de sorte que la taupe se déplace une fois toutes les demi-secondes. Si elle est touchée, le score augmente de un, et le téléphone vibre. En appuyant sur Reset on remet le score à zéro.

Ce tutoriel présente:

- sprites d'image
- minuteries et l'horloge composante
- procédures
- choisir un nombre aléatoire entre 0 et 1
- des blocs de texte
- typeblocking

- Faites glisser les composants de la Palette dans l'Observateur et affecter leurs noms. Mettez MyCanvas sur le dessus et définir ses dimensions de 300 pixels de large par 300 pixels de haut. Réglez le texte de ScoreLabel à "Note: ---". Réglez le texte du bouton RESET à "Reset". Ajoutez également un son élément et nommez-le "bruit". Vous utiliserez bruit pour faire vibrer le téléphone quand la taupe est frappée.



- On doit modifier les paramètres de l'horloge pour que la taupe puisse sauter régulièrement. Le composant de l'horloge fournit diverses opérations. Ici, vous allez utiliser le composant comme une minuterie qui déclenche à intervalles réguliers. L'intervalle de mise à feu est déterminé par la propriété **TimerInterval**. Nommez-la "MoleTimer". Définissez son TimerInterval à 500 millisecondes pour faire déplacer la taupe toutes les demi-secondes. Assurez-vous que **TimerEnabled** est cochée.
- Modifier les paramètres de l'imageSprite

Nom :

Prénom :

- Chaque Sprite possède une **vitesse** et un **cap**, et aussi un **intervalle** qui détermine la fréquence des sprites. Les Sprites peuvent aussi détecter quand ils sont touchés. Dans **MoleMash**, la taupe a une vitesse nulle, de sorte qu'elle ne se déplace pas par elle-même. Au lieu de cela, on doit définir la position de la taupe suivant la minuterie. Définir ces propriétés :
 - **Enabled**: checked
 - **Interval**: 500
 - **Heading**: 0
 - **Speed**: 0.0
 - **Visible**: checked
 - **Width**: Automatic
 - **Height**: Automatic

Vous devriez voir le **x** et **y** propriétés déjà remplis. Ils ont été déterminés quand vous avez placé la taupe lorsque vous l'avez fait glisser sur **MyCanvas**.

- Définissez les deux procédures:
 - **MoveMole** déplace la taupe à une nouvelle position aléatoire sur la toile.
 - **UpdateScore** montre le score, en modifiant le texte de la **ScoreLabel**

The image shows two Scratch code blocks. The top block is a 'to procedure' block named 'MoveMole'. It contains two 'do' blocks. The first 'do' block has a 'set ImageSprite1 . X to' block followed by a 'random fraction' block multiplied by '(Canvas1 . Width - ImageSprite1 . Width)'. The second 'do' block has a 'set ImageSprite1 . Y to' block followed by a 'random fraction' block multiplied by '(Canvas1 . Height - ImageSprite1 . Height)'. The bottom block is a 'to procedure' block named 'UpdateScore'. It contains a 'do' block with a 'set Label1 . Text to' block followed by a 'join' block with 'Score: ' and 'get global score'. Two red arrows point from the 'to procedure' block on the right to the 'MoveMole' and 'UpdateScore' blocks.

- Créer une variable score qu'on initialisera à 0. Cette variable va être incrémentée chaque fois que la taupe sera touchée. On doit la créer avant de remplir la fonction UpdateScore.

The image shows two Scratch code blocks. The top block is an 'initialize global' block with 'score' as the variable name and '0' as the value. A blue arrow points from the word 'Variable' to the 'score' variable. The bottom block is a 'to procedure' block named 'UpdateScore'. It contains a 'do' block with a 'set ScoreLabel . Text to' block followed by a 'join' block with 'Score: ' and 'get global score'.

- La prochaine étape est de faire avancer la taupe en continu. On va utiliser **MoleTimer**. Ce composant de l'horloge à un gestionnaire d'événement appelé **when ... Timer** qui déclenche de façon répétée à un taux déterminé par le **TimerInterval**.

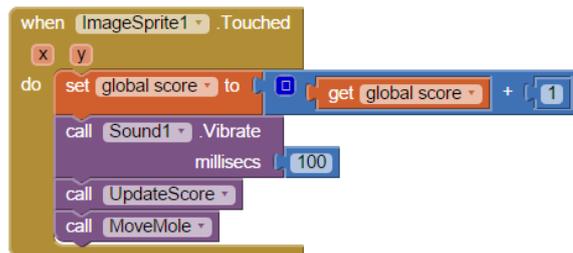
Nom :

Prénom :



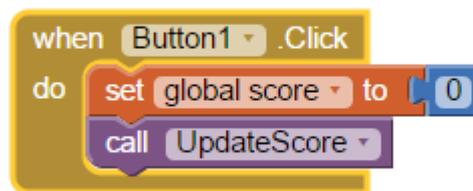
```
when Clock1.Timer
do call MoveMole
```

- Le programme doit augmenter le score à chaque fois que la taupe est touchée. Créer un gestionnaire d'événement tactile pour :
 - Incrémente le score.
 - Appels `UpdateScore` pour montrer le nouveau score.
 - Fait vibrer le téléphone pour 1/10 de seconde (100 millisecondes).
 - Appels `MoveMole` pour que la taupe se déplace tout de suite, plutôt que d'attendre la minuterie.



```
when ImageSprite1.Touched
do
  set global score to get global score + 1
  call Sound1.Vibrate 100
  call UpdateScore
  call MoveMole
```

- Réaliser la procédure de remise à zéro du score.



```
when Button1.Click
do
  set global score to 0
  call UpdateScore
```

- Programme fini

Nom :

Prénom :

```
to MoveMole
do
  set ImageSprite1 . X to random fraction × Canvas1 . Width - ImageSprite1 . Width
  set ImageSprite1 . Y to random fraction × Canvas1 . Height - ImageSprite1 . Height
```

```
to UpdateScore
do
  set Label1 . Text to join " Score: "
  get global score
```

```
when Button1 . Click
do
  set global score to 0
  call UpdateScore
```

```
initialize global score to 0
```

```
when Clock1 . Timer
do
  call MoveMole
```

```
when ImageSprite1 . Touched
do
  set global score to get global score + 1
  call Sound1 . Vibrate
  milliseconds 100
  call UpdateScore
  call MoveMole
```



• Travail supplémentaire :

- Faire varier la vitesse de la taupe en réponse à la façon dont le joueur joue. Plus le score augmente, plus la vitesse de déplacement de la taupe augmente.
- Gardez une trace du moment où le joueur frappe la taupe et lorsque le joueur rate la taupe, et montrer un score avec deux affichages réussites et échecs. Pour ce faire, vous aurez besoin de définir des gestionnaires touchés à la fois pour la taupe, et pour MyCanvas.