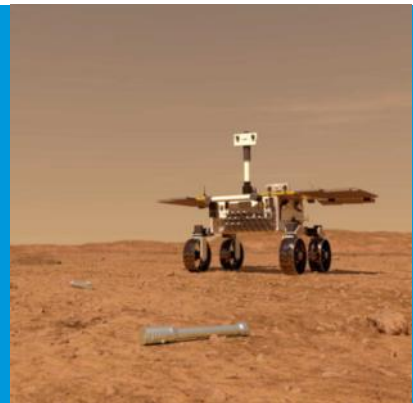


Fiche pour l'enseignant

# Reconnaissance d'images

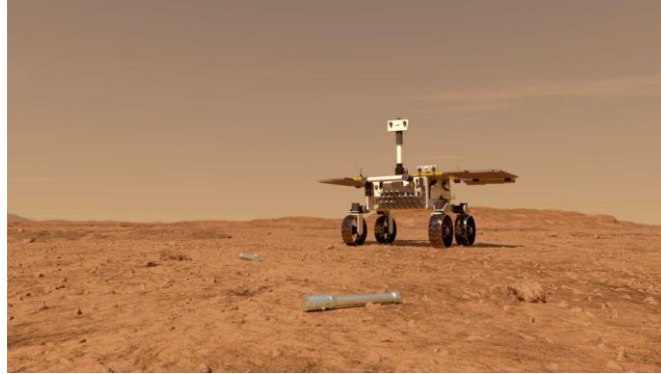
Phase 2 / Avec Scratch



Une fois que les élèves ont entraîné un modèle de reconnaissance d'images avec la Teachable Machine (voir fiche « Reconnaissance d'images, phase 1 – Entraînement du modèle »), cette activité montre comment l'utiliser dans un programme de façon simple avec Scratch pour programmer un rover à retrouver des tubes sur un sol martien.



# Intelligence Artificielle



---

Fetchbot : Rover martien intelligent

---

Age : 10-14 ans

---

## Reconnaissance d'images avec Scratch

Objectifs :

- ✓ Faire parler le rover Fetch pour qu'il dise ce qu'il voit avec sa caméra.

Notions abordées : Intelligence artificielle, classification d'images, programmation avec Scratch, condition, boucle.

- Phase 1 : Aperçu
- Phase 2 : Adacraft
- Phase 3 : Créer un projet Adacraft
- Phase 4 : Fais parler l'étoile
- Phase 5 : Changer l'étoile en rover martien
- Phase 6 : Conditions et boucles
- Phase 7 : Activation de la caméra et détection de mouvement
- Phase 8 : Reconnaissance d'images avec le modèle Teachable Machine

Durée : 4h00

Dispositif pédagogique : par groupe de deux.

### Matériel

- Un ordinateur/tablette par groupe de 2 avec connexion internet

---

### Prérequis

- Connaissances de bases de Scratch
- Avoir créé un modèle de reconnaissance d'image avec la Teachable Machine de Google ([Activité 1 - Reconnaissance d'images avec la Teachable Machine](#))

---

### Références & liens utiles

- Ressources complémentaires pour apprendre Scratch
  - [Ressources Scratch de la Communauté de l'Apprentissage de l'Informatique](#)
  - ["Je découvre Scratch", vidéo de 30 minutes de la Scientothèque](#)
- Autres ressources :
  - [Fondation La Main à la Pâte](#)
  - [CodeClub - Scratch Module 1](#)
  - [Fondation Raspberry - Introduction à Scratch](#)

---

### Droits d'auteur

Le contenu de cette fiche pédagogique est publié sous licence Creative Commons Attribution - Pas d'utilisation commerciale - Partage dans les mêmes conditions ([CC-BY-NC-SA](#)) :

**Attribution** [BY] (*Attribution*) : l'œuvre peut être librement utilisée, à la condition de l'attribuer à l'auteur en citant son nom : La Scientothèque. Cela ne signifie pas que l'auteur est en accord avec l'utilisation qui est faite de ses œuvres.

**Pas d'utilisation commerciale** [NC] (*Noncommercial*) : le titulaire de droits peut autoriser tous les types d'utilisation ou au contraire restreindre aux utilisations non commerciales (les utilisations commerciales restant soumises à son autorisation). Elle autorise à reproduire, diffuser, et à modifier une œuvre, tant que l'utilisation n'est pas commerciale.

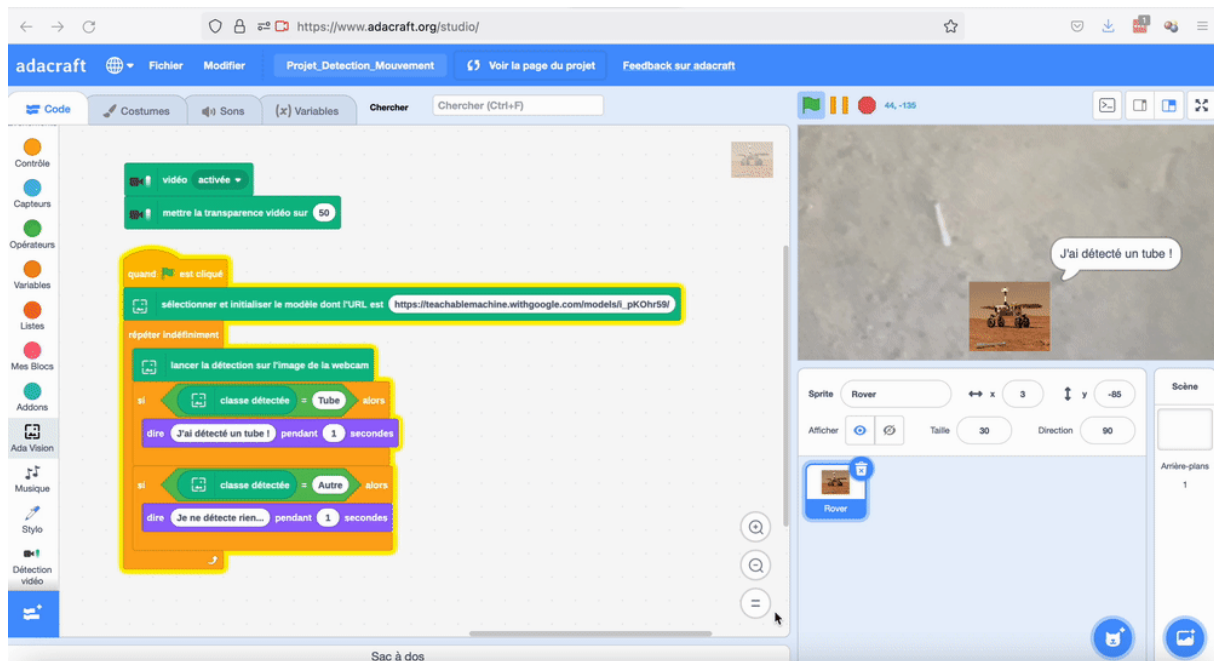
**Partage dans les mêmes conditions** [SA] (*ShareAlike*) : le titulaire des droits peut autoriser à l'avance les modifications ; peut se superposer l'obligation (SA) pour les œuvres dites dérivées d'être proposées au public avec les mêmes libertés que l'œuvre originale (sous les mêmes options Creative Commons).

## Description détaillée

Un protocole à destination des jeunes est disponible [en ligne](#). L'activité peut donc être réalisée en autonomie par les élèves.

### Phase 1 : Aperçu

Le programme Scratch final est illustré dans l'image animée ci-dessous. Le sprite (le dessin) du rover Fetch dit s'il détecte un tube (ou rien) sur l'image qui est présentée à la caméra.

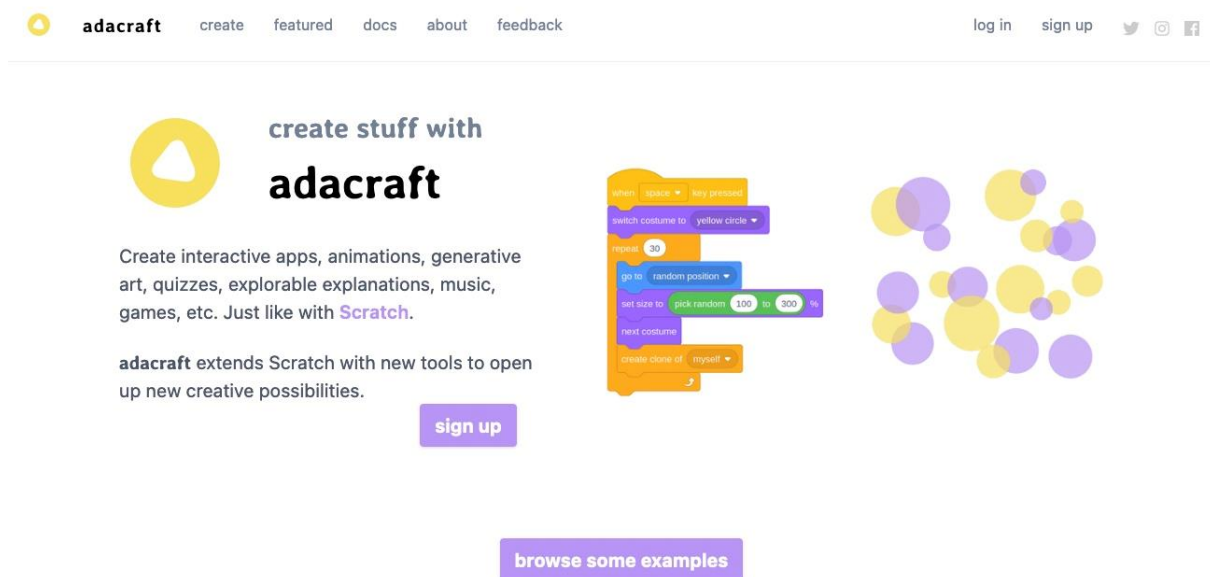


### Phase 2 : Adacraft

Cette activité utilise Adacraft, qui est une version modifiée de Scratch incluant des extensions pour l'intelligence artificielle. Nous utiliserons dans cette activité les extensions 'Détection Vidéo' et 'AdaVision'. L'extension 'Détection Vidéo' permet d'afficher les images venant de la caméra, et de détecter le mouvement. L'extension 'AdaVision' permet d'importer un modèle de reconnaissance d'image de la Google Teachable Machine, et d'associer des actions en fonctions de l'image reconnue à la caméra.

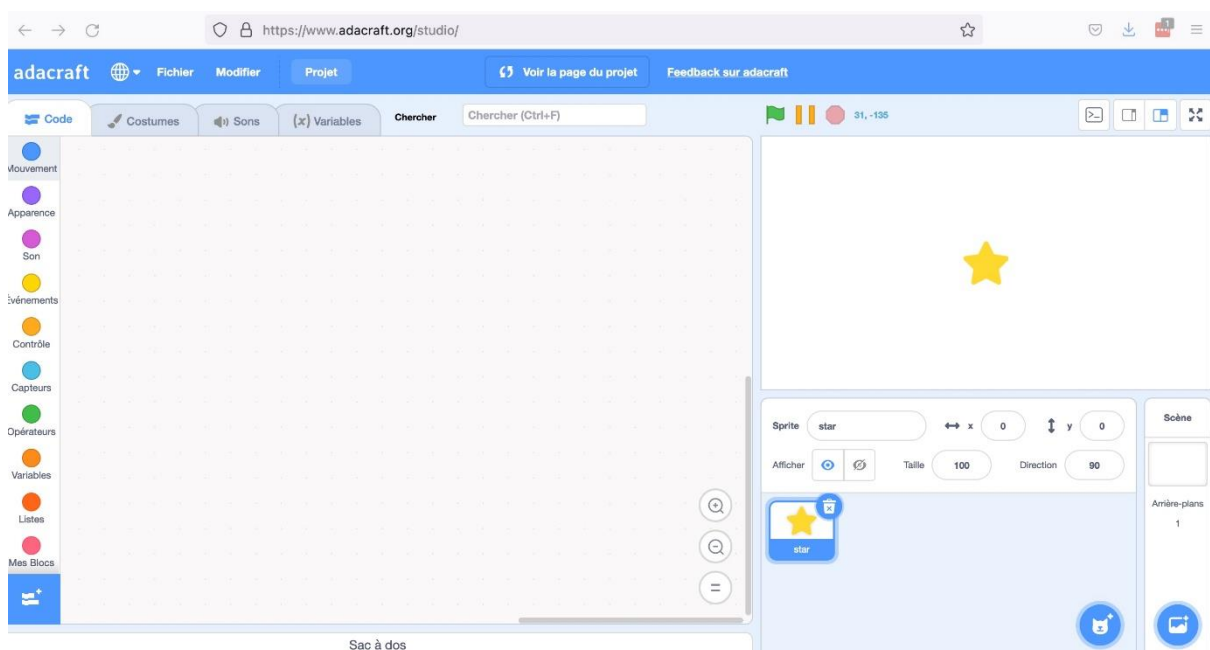
### Phase 3 : Créer un projet Adacraft

Les élèves doivent se rendre sur le site d'Adacraft : <https://www.adacraft.org>, puis cliquer sur 'Create' en haut à gauche.



S'ils ne sont pas inscrits, un message apparaîtra les informant que leur travail ne pourra pas être sauvegardé. Il est intéressant de s'inscrire sur le site (en cliquant sur 'Sign up') pour pouvoir sauvegarder ses programmes. Si vous souhaitez juste tester l'interface, cliquez sur 'Start creating without online saving'. Ils pourront dans tous les cas sauvegarder leur travail en le téléchargeant sur leur ordinateur.

Une fois ce message validé, ils arriveront sur l'interface du *studio* dans lequel ils pourront créer leurs programmes.

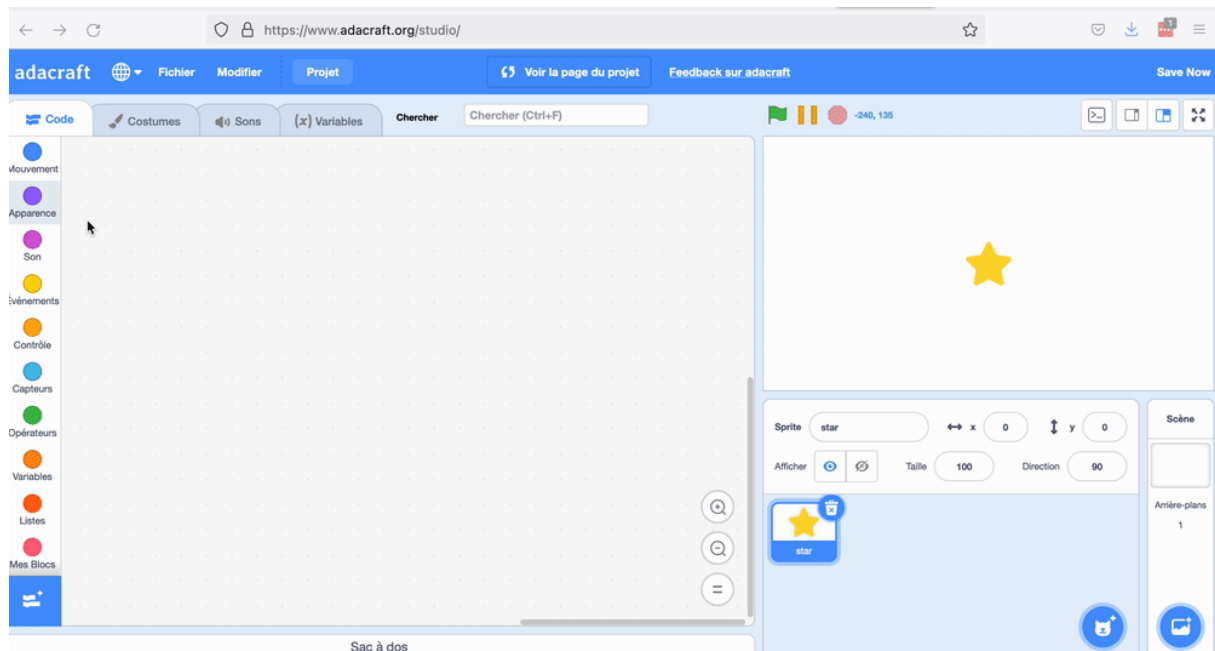


Vous pouvez choisir la langue en cliquant sur l'icône de planète en haut à gauche.

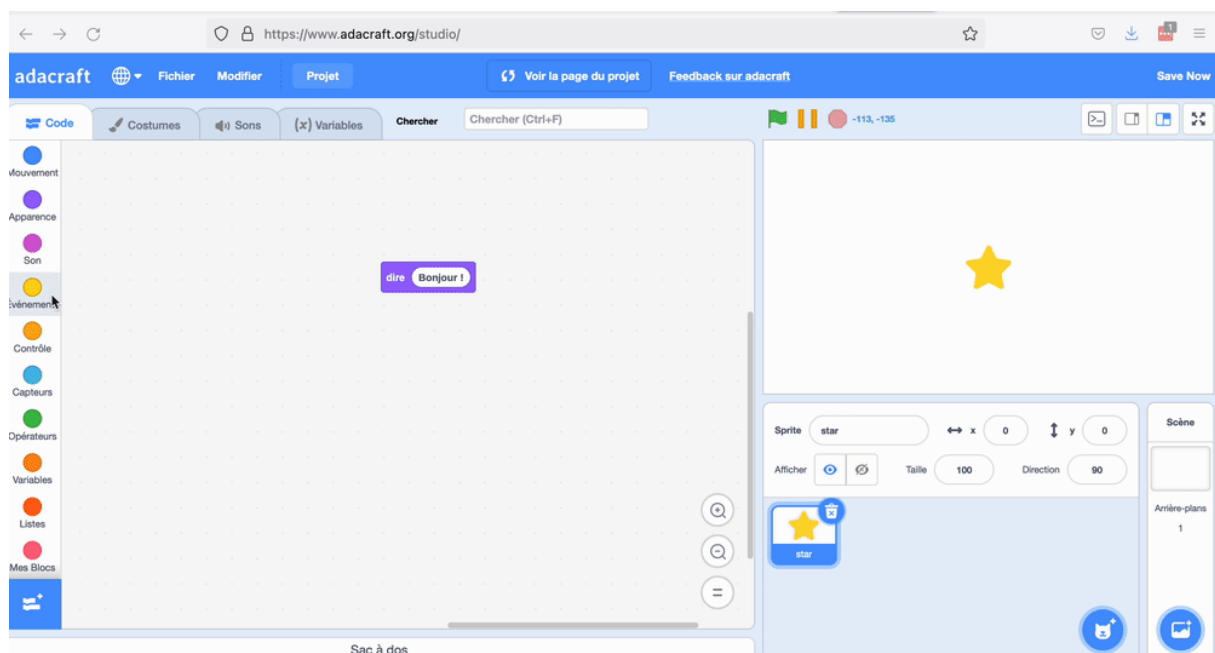
## Phase 4 : Faites parler l'étoile

Par défaut, le sprite affiché sur la scène est celui de l'étoile. Pour la faire parler, il faut utiliser le bloc 'Dire ...' qui se trouve dans la zone d'instruction à gauche, dans la catégorie 'Apparence'.

Déplacez le bloc 'Dire Bonjour' dans la zone du programme. Si vous cliquez sur ce bloc, vous verrez l'étoile dire 'Bonjour'.

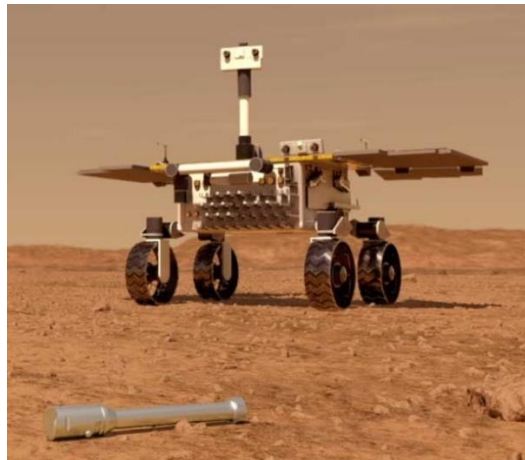


Vous allez maintenant faire démarrer ce programme en cliquant sur le drapeau vert (au-dessus de la scène) plutôt qu'en cliquant sur le bloc. Pour cela, sélectionnez le bloc 'Quand drapeau vert est cliqué' qui se situe dans la catégorie événements de la zone d'instructions, et placez le au-dessus du bloc 'Dire Bonjour'. Vous pourrez démarrer le programme en cliquant sur le drapeau vert au-dessus de la scène, et arrêter le programme en cliquant sur le 'stop' (l'hexagone rouge).



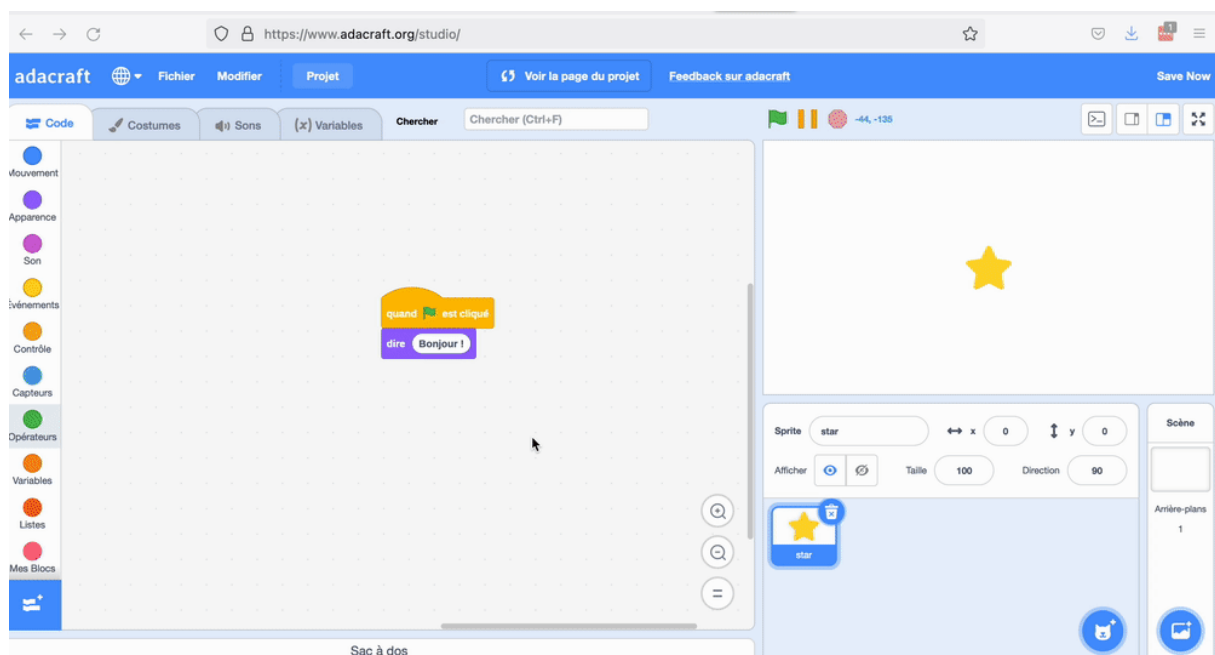
## Phase 4 : Changez l'étoile en rover martien

Vous allez maintenant remplacer l'étoile par un rover martien comme celui ci-dessous :

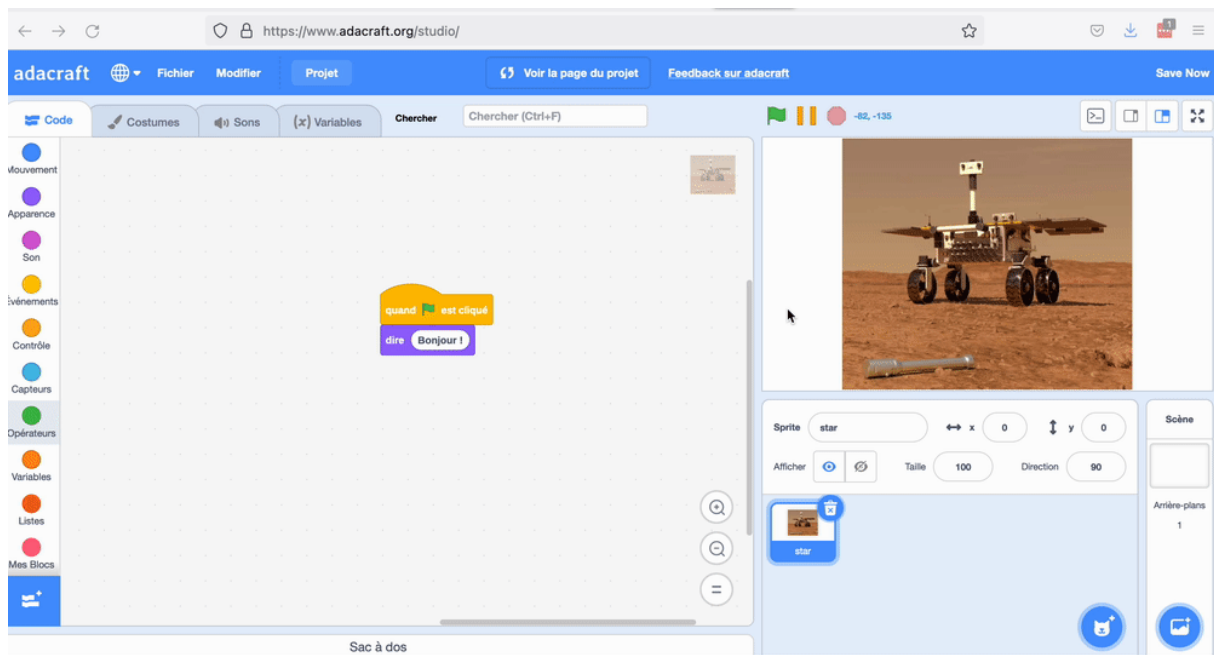


Vous pouvez télécharger l'image ici <https://github.com/lascientotheque/fetchbot-fr/raw/main/images/fetch-rover-small.jpg>.

Pour changer l'étoile avec cette image, il faut aller dans l'onglet 'Costumes', et placer la souris sur l'icône du chat en bas à gauche ('Choisir un costume'), puis sélectionner 'importer un costume'. Sélectionnez ensuite le fichier `fetch-rover-small.jpg` dans le répertoire où vous l'avez téléchargé (par défaut ; il s'agit du répertoire 'Téléchargements' de l'ordinateur). Revenez ensuite sur l'onglet 'Code'.



L'image est un peu trop grande et prend toute la scène. Réduisez sa taille pour qu'elle soit plus petite, et déplacez-la vers le bas de la scène. Vous pouvez changer la valeur de la taille dans la boîte 'Taille' située sous la scène. Choisissez une taille de '30', puis déplacez l'image vers le bas en la sélectionnant avec la souris et en la plaçant un peu plus bas.



Vous pouvez maintenant relancer le programme avec le drapeau vert, c'est Fetch qui vous dira bonjour !

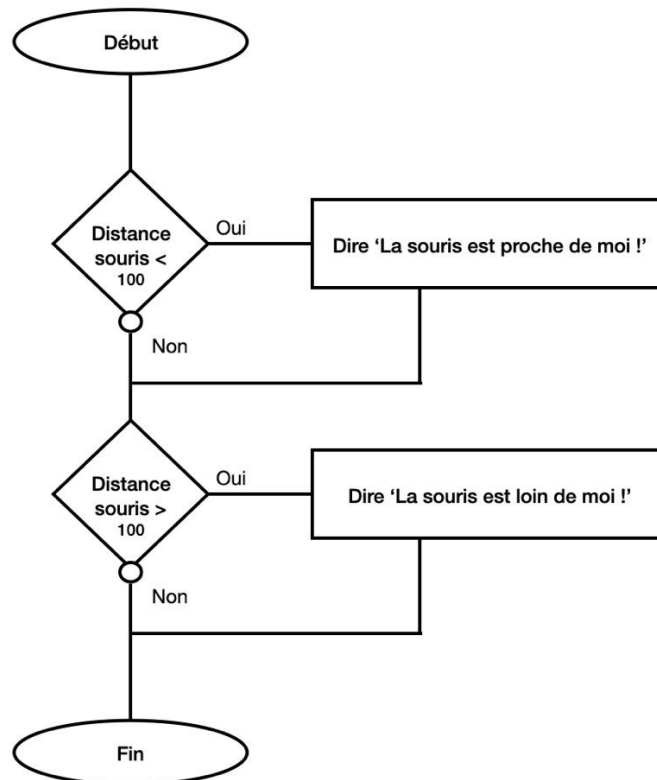
## Phase 6 : Conditions et boucles

Faites maintenant réagir ce que dit le rover en fonction de la position de la souris :

- Si la souris est proche du sprite de rover, alors le rover dit 'La souris est proche de moi !'
- Si la souris est loin du sprite du rover, alors le rover dit 'La souris est loin de moi!'

Le logigramme correspondant à ce programme est donné ci-dessous :





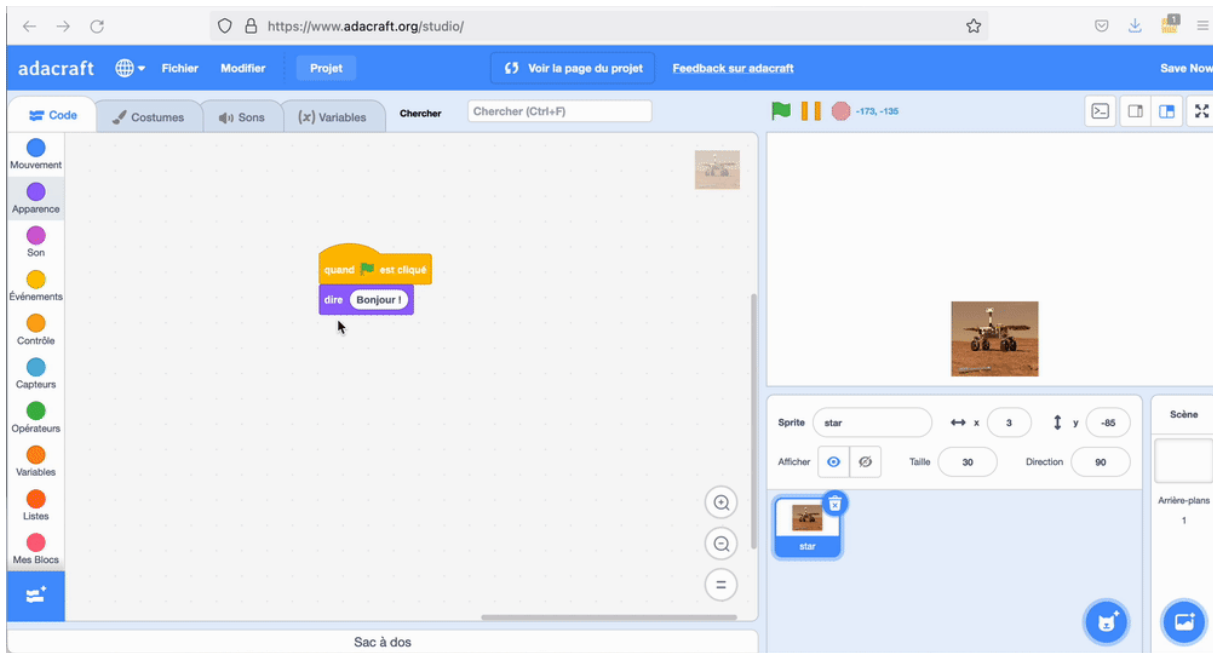
Lorsque le programme est lancé, on teste d’abord si la souris est proche du rover (une distance inférieure à 100 pixels). Si oui, alors le programme affiche ‘La souris est proche de moi !’. Ensuite on teste si la souris est loin du rover (une distance supérieure à 100 pixels). Si oui, alors le programme affiche ‘La souris est loin de moi !’.

Pour programmer ce logigramme en Scratch, utilisez un bloc de condition ‘si ... alors’ qui se trouve dans la catégorie ‘Contrôle’, un bloc de comparaison ‘... = ...’ qui se trouve dans la catégorie ‘Opérateurs’, et un bloc qui donne la distance entre la souris et le sprite, qui se trouve dans la catégorie ‘Capteurs’.

Voici le programme à obtenir :

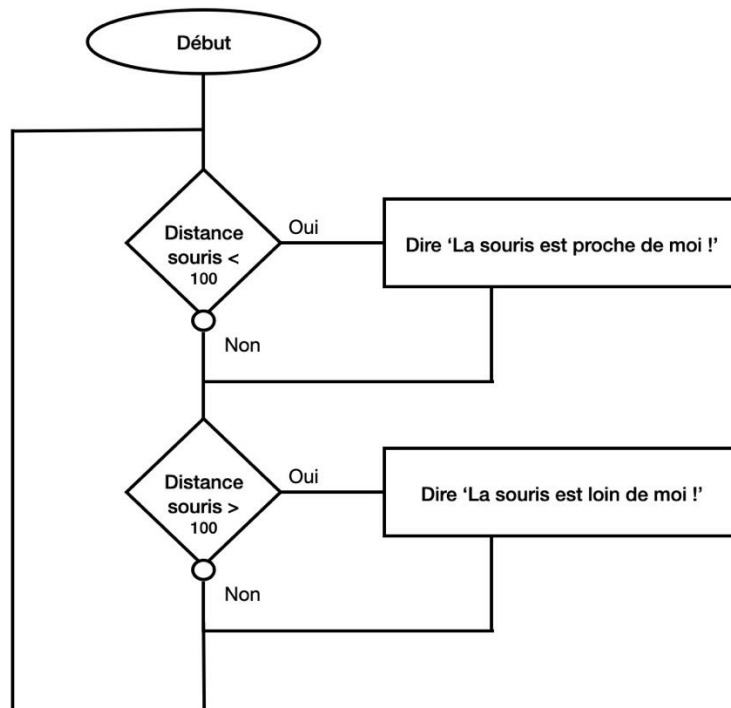


Les élèves doivent essayer de le faire par eux-mêmes. Si jamais ils ne trouvent pas un bloc, montrez-leur le détail pas-à-pas de ce qu’il faut faire ci-dessous (voir la version [en ligne](#) pour l’image animée) :



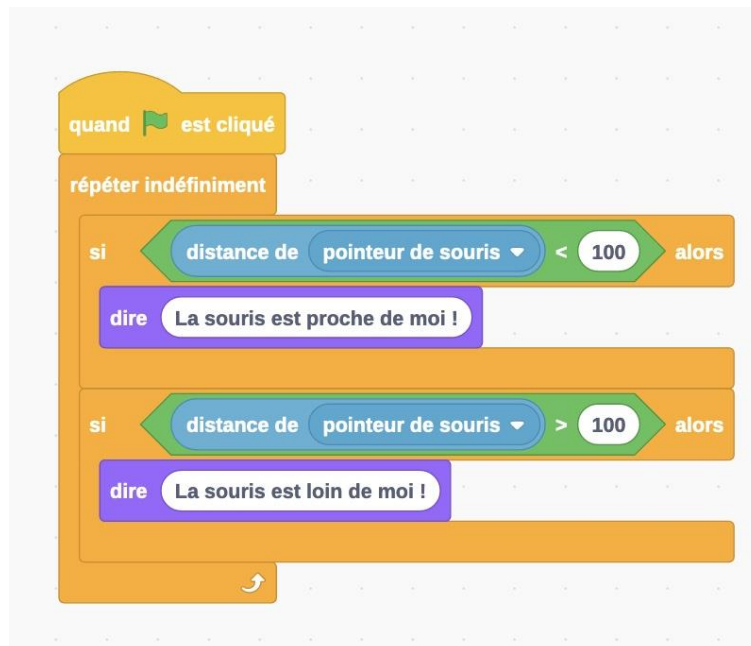
Si vous lancez le programme en cliquant sur le drapeau vert, le rover dira toujours 'La souris est loin de moi !'. En effet, le programme se termine tout de suite après que vous ayez cliqué sur le drapeau, et de drapeau est loin du rover.

Pour changer cela, ajoutez une boucle pour que le programme se répète indéfiniment. Le logigramme est modifié comme suit : Il n'y a plus de fin, et on revient au premier test lorsque l'on a terminé le deuxième test :

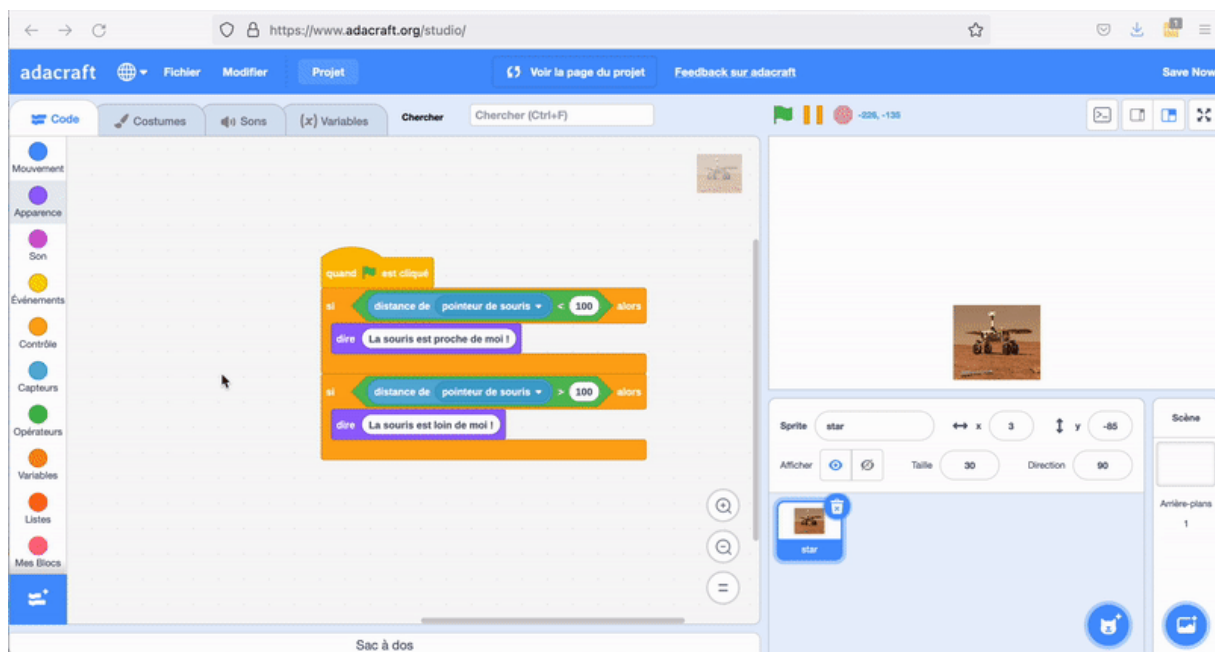


Pour faire cela dans Scratch, ajoutez un bloc 'Répéter indéfiniment' (qui se trouve dans la catégorie 'Contrôle') pour inclure les autres blocs.

Voici le programme à obtenir :



Et ici le détail pas-à-pas pour obtenir ce programme (voir la version [en ligne](#) pour l'image animée).



Maintenant, si vous cliquez sur le drapeau vert, vous verrez le texte changer lorsque vous approchez la souris du sprite !

## Phase 7 : Activation de la caméra et détection de mouvement

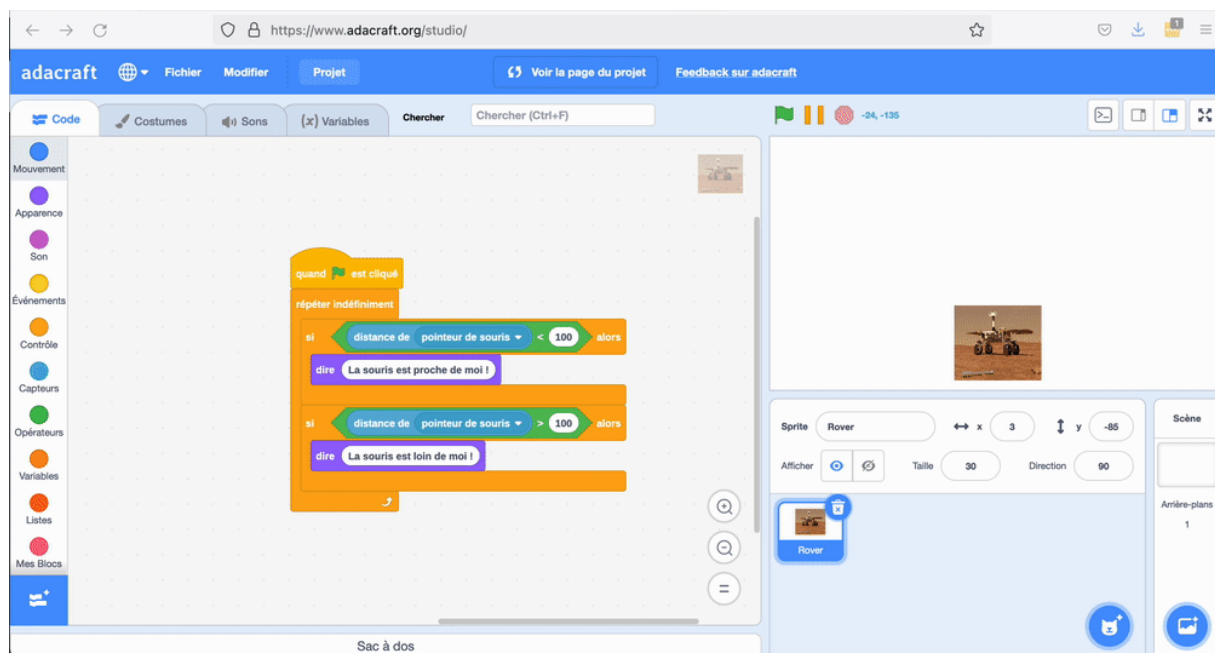
Vous allez maintenant faire un programme très similaire. Au lieu de détecter si la souris est proche du rover, le programme détectera s'il y a du mouvement devant la caméra au niveau du rover.

### 2.7.1. Activation de la caméra

Vous devez tout d'abord activer la caméra.

Les blocs de programmation pour la caméra et la détection de mouvement se trouvent dans une extension. Pour y accéder, il faut cliquer sur l'icône en bas à gauche 'Ajouter une extension'. Ceci vous amènera à une page avec toutes les extensions disponibles. Descendez dans la page pour trouver l'extension 'Détection vidéo' et cliquez dessus. Vous pourrez voir l'extension dans la zone d'instructions.

Le navigateur vous demandera s'il peut accéder à la caméra. Autorisez-le. Vous verrez alors la vidéo venant de la caméra s'affichera en arrière-plan dans la scène.



### 2.7.2. Contrôle de la caméra et de la transparence

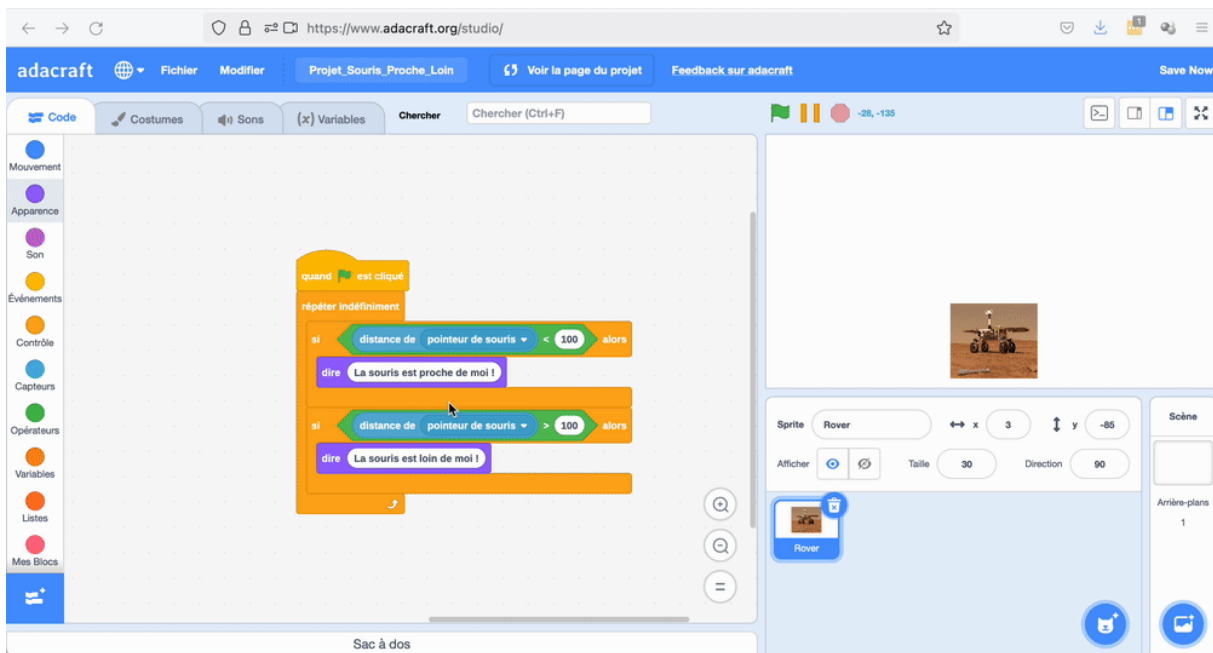
Il est possible d'allumer et d'éteindre la caméra grâce au bloc 'Vidéo', et choisir la transparence grâce au bloc 'Mettre la transparence vidéo sur ...'.

Placez les deux blocs l'un sur l'autre dans la zone de programmation. Vous pouvez changer les valeurs pour le bloc vidéo de façon à allumer ou éteindre la caméra. La transparence peut être changée avec le bloc 'Mettre la transparence vidéo sur ...' en changeant la valeur, qui doit être entre 0 et 100. 0 correspond à l'image de caméra, et 100 à une image transparente (on ne la verra plus).

Voici les deux blocs à ajouter :



Vous pouvez ensuite allumer et éteindre la caméra en cliquant sur le bloc 'Vidéo'.



Une transparence de 50 est un bon choix, mais faites vos propres tests !

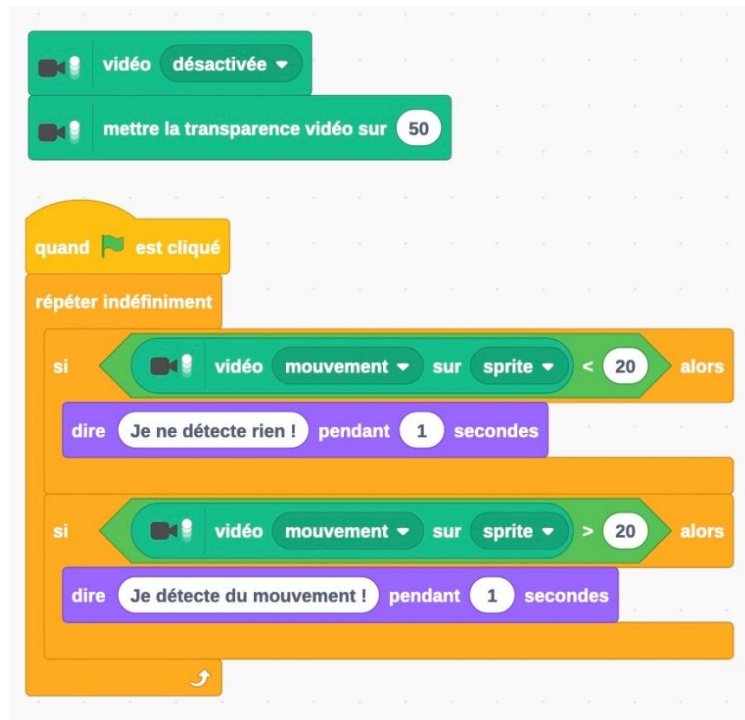
### 2.7.3. Détection de mouvement

La détection de mouvement sur un sprite peut être ajoutée au programme grâce au bloc 'Vidéo mouvement sur sprite' de l'extension 'Détection vidéo'.

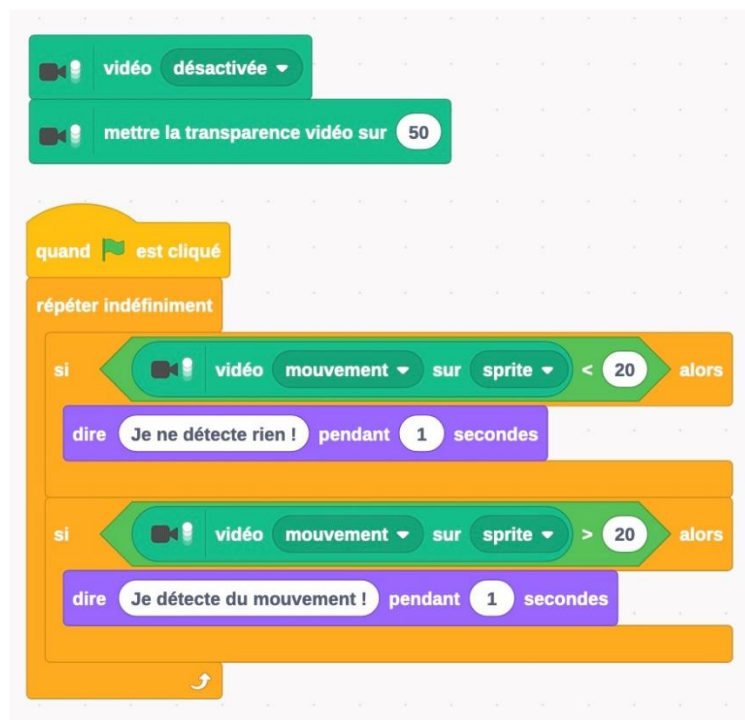
Modifiez le programme écrit précédemment en :

- Remplaçant les blocs 'distance de pointeur de souris' par les blocs 'Vidéo mouvement sur sprite'
- En changeant la valeur 100 en 20 (le seuil de détection de mouvement est compris entre 0 et 100)
- En changeant le texte pour faire dire au rover s'il voit du mouvement devant lui ou pas. Utilisez le bloc 'Dire ... pendant ... seconds', avec les textes 'Je détecte du mouvement !' quand le seuil de détection est supérieur à 20, et 'Je ne détecte pas de mouvement !' lorsque le seuil de détection est inférieur à 20. Affichez le texte pendant 1 seconde.

Voici le programme à obtenir :



Le rover peut maintenant détecter lorsque quelque chose bouge devant lui et le dire !



Vous pouvez changer le seuil de détection de mouvement pour modifier la sensibilité de la détection.

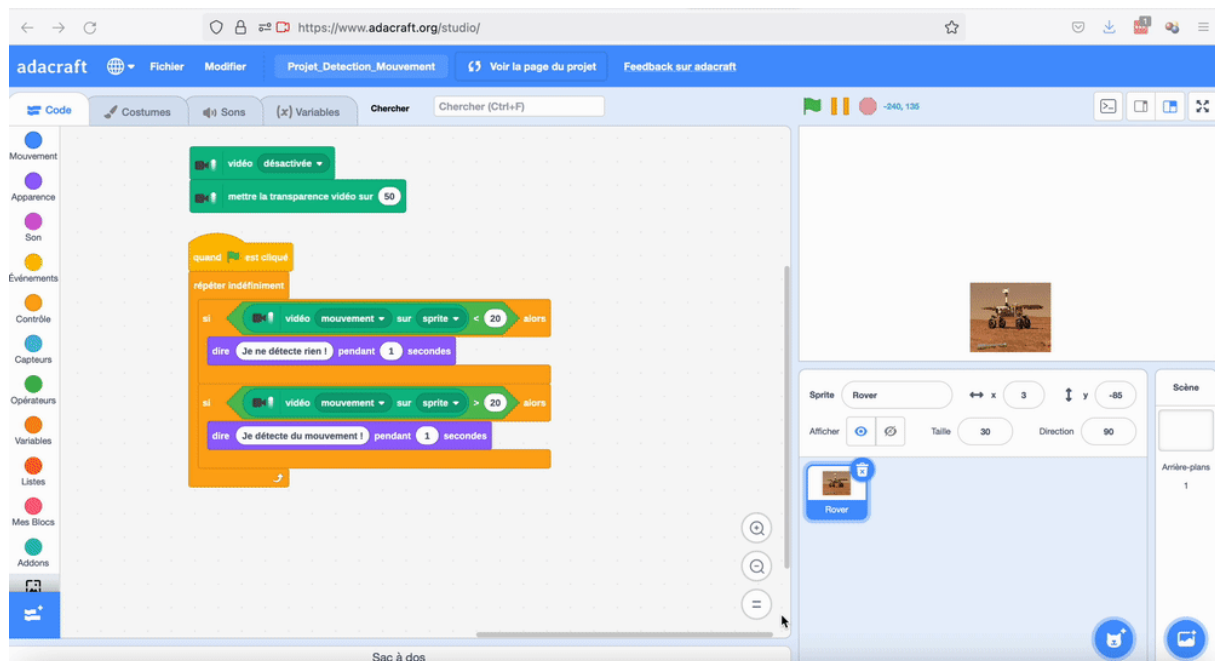
## Phase 8 : Reconnaissance d'images avec le modèle Teachable Machine

Les élèves vont maintenant programmer le rover avec de la reconnaissance d'images. Vous utiliserez pour cela un modèle de prédiction que vous aurez créé auparavant avec la Teachable Machine (Activité 1).

### 2.8.1. Ajout de l'extension AdaVision et chargement du modèle

Pour la reconnaissance d'image, vous aurez besoin de l'extension AdaVision. Comme pour la détection de mouvement, allez dans les extensions Adacraft, et sélectionnez l'extension 'AdaVision'. L'extension contient des blocs pour charger un modèle de reconnaissance d'images, et pour ensuite détecter ce qui se trouve dans une image.

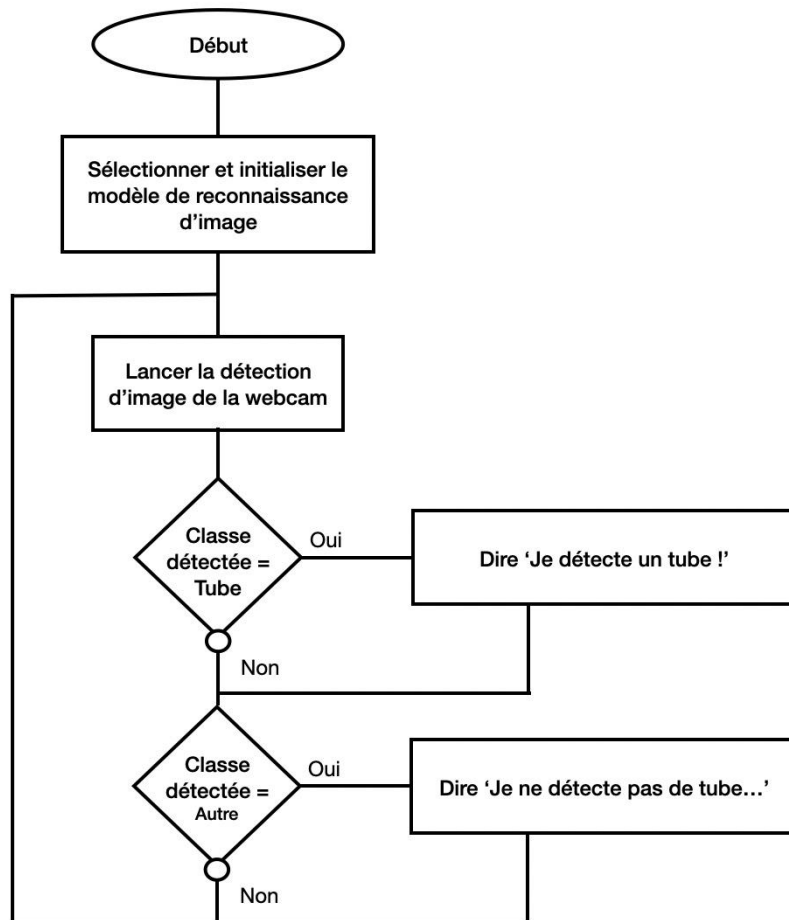
Le bloc pour charger un modèle est le premier dans la catégorie 'AdaVision', qui s'appelle "Sélectionner et initialiser le modèle dont 'l'URL est ...". Sélectionnez-le et placez le juste après le bloc "Quand le drapeau est cliqué".



Remplacez ensuite l'URL avec celle du modèle de reconnaissance de tube que vous avez créé dans l'activité 1.

### 2.8.2. Détection de tubes

Le programme pour la détection de tube est très similaire à celui pour la détection de mouvement : Au lieu de tester s'il y a du mouvement devant la caméra, vous testerez si un tube est présent. Cela donne le logigramme suivant :

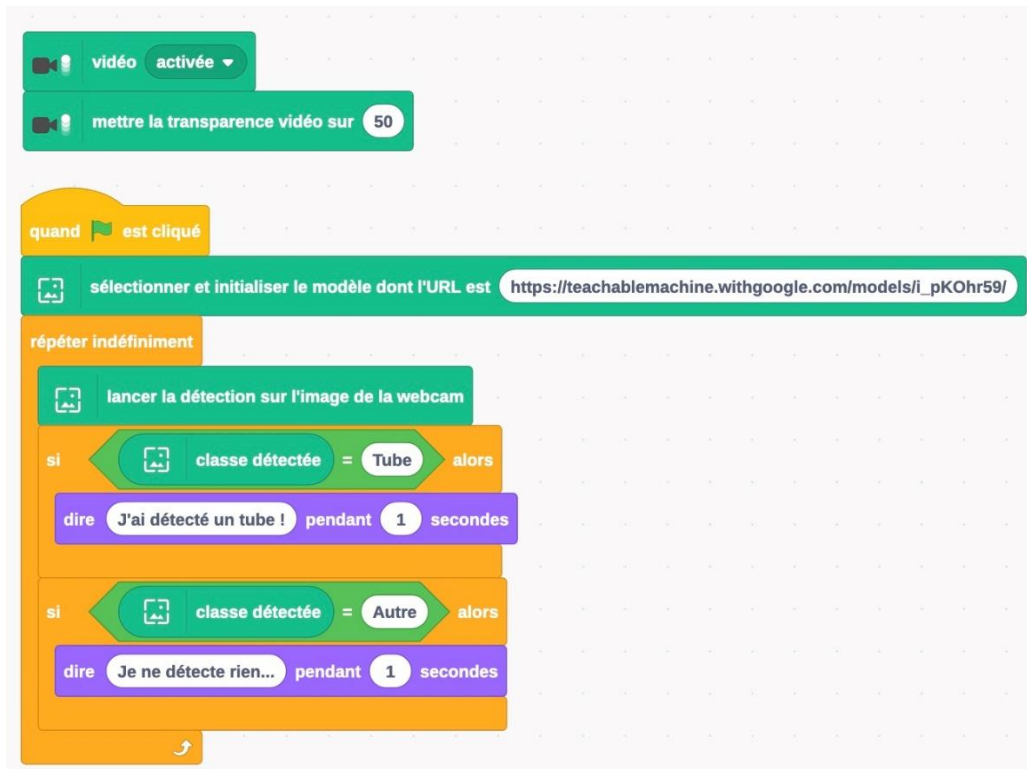


Modifiez le programme de détection de mouvement comme suit :

- Ajoutez un bloc “Lancer la détection sur l’image de la webcam” après le bloc “Répéter indéfiniment”
- Dans les blocs de conditions, utilisez des blocs de comparaisons d’égalité “... = ...” pour tester si l’image reconnue est un tube, ou autre chose
- Dans les blocs de comparaison, vous mettrez à gauche le bloc “Classe détectée”, et à droite la classe “Tube”, ou “Autre”.
- Changez le texte de ce que le rover en fonction de ce qui est détecté. Par exemple,
  - “J’ai détecté un tube !” lorsque la classe détectée est “Tube”, et
  - “Je ne détecte rien...” lorsque la classe détectée est “Autre”.

Voici le programme à obtenir :





Attention : Dans le bloc "Sélectionner et initialiser le modèle dont l'URL est ...", c'est l'URL du modèle que vous avez créé qu'il faut mettre !

Lancez ensuite le programme. Votre rover peut maintenant détecter si un tube se trouve devant lui !!!

