

# OUTIL - LES ERREURS FRÉQUENTES DANS SCRATCH

CRÉATION DE CONTENU > 3.4 PROGRAMMATION

CONVIENT POUR	AGE	NIVEAU DE COMPÉTENCE	FORMAT	DROITS D'AUTEUR	LANGUE(S)
Animateurs	N/A	Niveau 2	Fiche de préparation	Creative Commons (BY-SA)	Français

Cette fiche présente des erreurs fréquentes d'utilisation de l'outil Scratch et les façons de les résoudre.

**Objectif général**                      Compétences

**Temps de préparation pour l'animateur**                      moins d'une 1 heure

**Domaine de compétence**                      3 - Création de contenu

**Ressource originellement créée**                      Français

## DÉROULÉ

### 1 L'absence de boucle

**Symptômes :**

Des collisions ne sont pas prises en compte, certains scripts avec des conditions ne fonctionnent pas etc.

**Cause :**



Le jeu ne marchera pas, car l'ordinateur ne vérifiera qu'une seule fois (lors du lancement) si la condition est vraie, puis le jeu va s'arrêter.

**Remède :**



En rajoutant une boucle, l'ordinateur vérifiera en permanence si la condition est vraie.

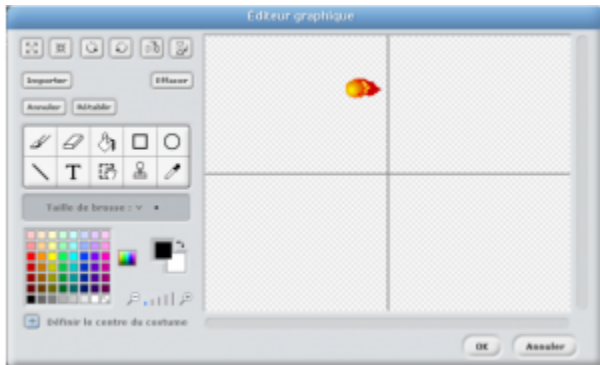
### 2 Le centre du dessin pas au bon endroit

**Symptômes :**

Le comportement d'un lutin est aberrant, les détections de collisions ne fonctionnent pas très bien...

**Cause :**

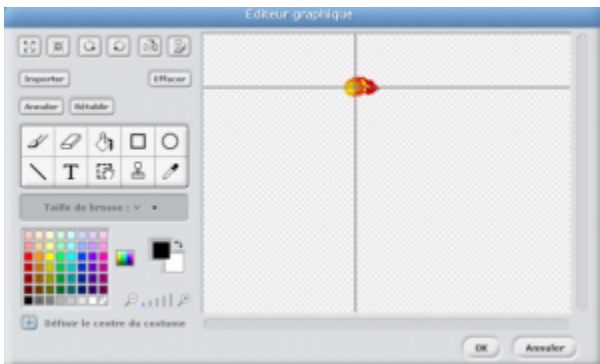
Le centre « physique » de l'objet (celui dont l'ordinateur se sert) ne correspond pas au centre graphique de l'objet (et il est généralement en dehors).



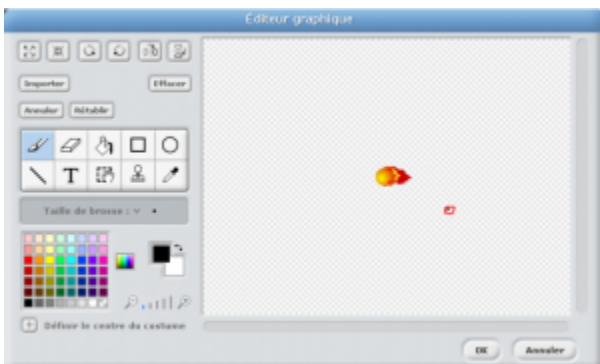
On peut le vérifier en cliquant sur « définir le centre du costume » dans l'outil d'édition.

### Remède :

Avec ce même bouton, on peut redéfinir le centre de l'objet



### Variante : le pixel isolé

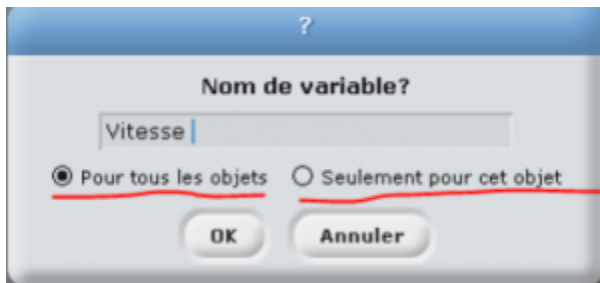


### 3 La variable globale

**Symptômes** : Une variable n'apparaît pas, ou ne fonctionne pas

**Cause** :

Lors de la création d'une variable, on a deux choix : l'affecter à tous les objets (variable globale) ou seulement à cet objet (variable locale). Généralement le souci survient lorsqu'on a affecté une variable locale.



**Remède** : recréer une nouvelle variable

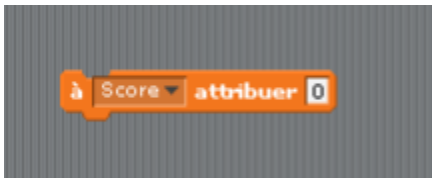
### 4 Mettre et ajouter sur une variable

**Symptômes** :

La valeur d'une variable est aberrante, elle ne se remet pas à zéro lors du lancement,

**Cause** :

Sur la version 1.4 de Scratch, il y a 2 blocs différents :



Ce bloc met la variable « Score » à 0, par exemple pour remettre les scores à 0 au début du jeu

A Scratch block with the text "changer Score par 1".

Celui-ci rajoute 1 au score actuel, par exemple si l'on prend un bonus.

**Remède :**

Choisir le bon bloc

## 5 Importer son / jouer son

**Symptômes :**

Le son n'est pas joué.

**Cause :**

Pour jouer un son, il faut d'abord aller dans l'onglet son de l'objet et importer (ou enregistrer) un son.



Une fois qu'il est stocké en mémoire, il faut préciser dans le programme quand le son doit être joué.



**Remède :** (voir ci-dessus)

## 6 Script des scènes / objets

### Symptômes :

Il manque des blocs dans la bibliothèque.



### Cause :

Vous êtes actuellement dans la partie Scène, et comme les scènes ne bougent pas, elles n'ont pas de bloc Mouvement.

### Remède :

Re-cliquer sur un objet.

## 7 Problème de détection de collision

### Symptômes :

Vous avez beau mettre des blocs de détection de collision par la couleur, votre objet ne les détecte pas ou mal...

### Cause :

Avec un bloc de ce type, il suffit que la couleur utilisée pour la détection se retrouve un peu dans votre arrière-plan, pour que la condition soit vraie.



## Remède :

Évitez autant que possible les détections par couleur ou choisissez des couleurs qui ne seront pas dans votre arrière plan. (Exemple le vert fluo, ne se retrouve pratiquement pas dans la nature, c'est pour cela qu'on l'utilise au cinéma)

## 8

# Quelques conseils divers pour trouver et résoudre les erreurs

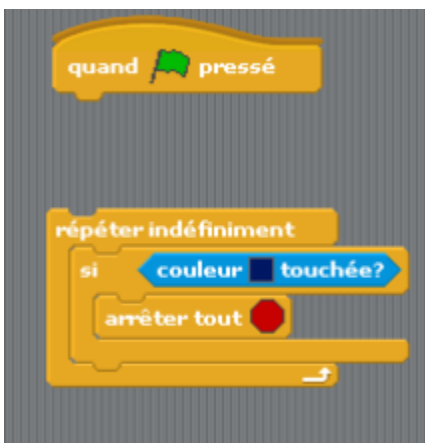
- **Etre au clair** sur ce que vous voulez faire :

Si vous ne savez pas exactement ce que vous voulez programmer, il y a peu de chance que l'ordinateur puisse le comprendre

- Bien **identifier** qui fait quoi :

Quel script est responsable de quelle partie du jeu, quels objets gèrent les détections etc.

- Ne pas hésiter à « **casser** » le programme :



En déplaçant les blocs ainsi, ils ne sont plus actifs. Ce qui est souvent pratique quand on veut identifier la provenance d'une erreur.

- Ne pas hésiter à écrire son code/ algorithme sur un **morceau de papier** :  
Les erreurs de logique apparaissent parfois mieux lorsque l'on sort de l'écran.