Programmer des comportements dans Construct 3

# Introduction

Pour donner vie à votre jeu, vous devez créer un programme pour indiquer les règles du jeu. Votre programme sera composé d’une série d’instruction.

# Structure d’une instruction

Lorsqu’il se passe quelque chose, fait autre chose. Autrement dit :

Lors d’un **évènement**/**condition**, fait une/des **action(s)**.

Exemple

Lorsque la touche « w » est appuyée, avance le personnage principal.

# Évènements

Un **évènement** se déclenche une fois au moment exact qu’il se produit. Vous pouvez les repérer dans Construct 3 avec le symbole . Voici quelques exemples :



Au moment exact que la Tourelle tire…

Au moment exact qu’il y a collision avec BalleAlien

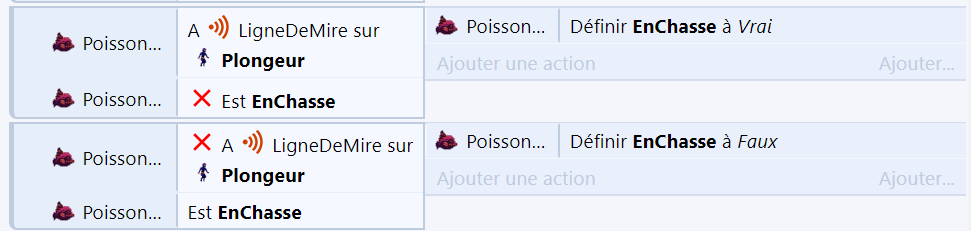
# Conditions

Une **condition** est testée 30x fois par seconde (la vitesse à laquelle le jeu s’exécute). Ça explique les comportements parfois étranges que vous avez quand vous codez avec des conditions. En effet, vos actions qui en dépendent peuvent être appelées à plusieurs reprises (30x par secondes!).

Contrairement aux évènements, il n’y aura pas d’icônes de flèches vertes pour ceux-ci.

## Transformer une condition en évènement

Prenez par exemple dans le Raton Coquin lorsque vous avez essayez détecter la proximité du raton avec le comportement *Ligne de mire*. Plusieurs ont été confronté au problème que le son du piège recommençait de jouer sans arrêt quand le raton approchait. Quand ça arrive, le truc pour contourner ce comportement est de se créer une variable booléenne comme avec le poisson rouge.



Par exemple, si je veux savoir exactement quand mon personnage arrive dans la ligne de mire du poisson, le premier code va être exécuté et la variable EnChasse sera à *Vrai*. Si le joueur s’éloigne, le deuxième code sera exécuté et la variable EnChasse sera à *Faux*.

# Actions

Vous savez comment fonctionnent les actions à ce stade.

# Variables

Les variables servent à entreposer des valeurs importantes. Par exemple, le nombre de vies de votre personnage, s’il est en train d’être endommagé, le pointage, etc.

## Variable globale

Lorsqu’on créé une variable globale, une seule variable est créée pour tout le programme. Une règle pour la création de variables globales : si je dois réinitialiser la scène et que j’ai encore besoin de la valeur, ça devrait être une variable globale. Par exemple : le meilleur score du joueur, le nombre de vies totale.

## Variable d’instance

Une variable d’instance est créée à partir d’un objet de la scène. Ce sera utile pour entreposer toute sorte d’information. Par exemple, le nombre de point de vies de mon joueur ou d’un ennemi, son état actuel (est en chasse ou non, est endommagé ou non, etc.). **On devrait préférer les variables d’instances car on peut avoir des problèmes lors de la réinitialisation de la scène.**