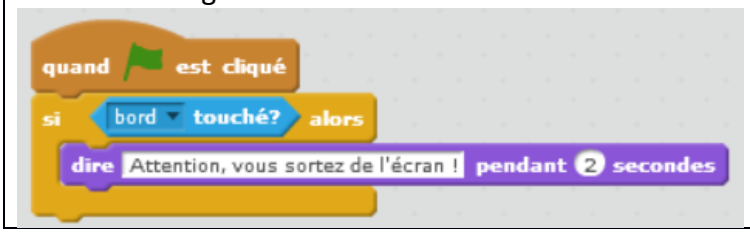
Gérer la temporalité sur Scratch

Introduction et premières approches

Une difficulté fréquemment rencontrée est celle de la temporalité : c'est-à-dire la prise en compte de la vitesse de lecture de l'algorithme, l'ordre d'exécution des instructions ou même la prise en compte du temps lui-même.

Prenons cet algorithme : 	Un déplacement au clavier étant programmé par ailleurs dans le script. (Voir par exemple le paragraphe « avec des flèches » de cette page ou un peu plus loin dans ce document)

Une première lecture peut laisser penser qu'il correspond à :

```
Si bord touché
        Dire « Attention … »
FinSi
```

Pourtant, lorsqu'on clique sur le drapeau vert pour commencer et que l'on déplace le lutin au clavier, le message ne s'affiche pas lorsque le lutin touche un bord.

Cela est dû à la vitesse de lecture du programme. À peine a-t-on cliqué sur le drapeau vert que le test conditionnel est lu. Lorsque le lutin arrive au bord de la scène, il est trop tard pour déclencher l'action.

Une première piste pourrait être :

Autres pistes plus pérennes

Deux autres pistes sont offertes par Scratch.

-I-

La première proposée ici est trompeuse : il s'agit du « répéter indéfiniment ». Il s'agit d'une instruction fort peu académique en informatique mais bien pratique ici. Il est vrai que Scratch possède un bouton d'arrêt externe qui corrige cette imperfection :

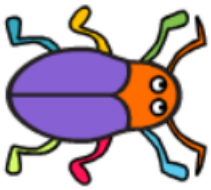
Le test va donc être réalisé en continu pendant que des commandes mettant en œuvre une interaction avec le clavier permettront le déplacement du lutin.

Autre programme :

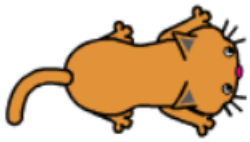
Ce programme va gérer des déplacements dans les 4 directions (droite, gauche, haut, bas).

S'il n'y a pas d'interaction avec le clavier, le lutin reste immobile.

On peut noter les orientations qui peuvent être visuellement sympathiques pour des lutins vus de dessus tels que :


Beetle

ou


Cat2

par exemple.

D'un point de vue informatique, on a une seule instruction pour démarrer un programme : le drapeau vert.

Cette commande  va d'ailleurs vider la variable  , ce qui ne sera pas le cas avec les instructions que l'on va présenter ici et qui correspondent à des interactions avec l'utilisateur sur un programme déjà démarré.

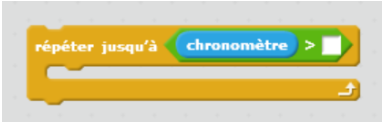
Ce qui peut donner ceci :

Par exemple dans un programme « complet », c'est-à-dire démarré avec  :

On pourra alors facilement gérer des interactions entre lutins ou avec les bords de la scène.

Une des dernières difficultés qui peut apparaître lors de la conception d'un algorithme peut intervenir avec l'instruction  qui se présente sous l'aspect d'une variable.

Cette instruction affiche le temps écoulé, en secondes, depuis l'ouverture du logiciel sans jamais s'arrêter. Cela implique que pour l'utiliser, il faut être attentif à toujours le réinitialiser au début du programme et être conscient qu'on ne pourra pas tester une égalité de temps mais seulement une inégalité à l'aide des instructions suivantes :



Un exemple d'utilisation du chronomètre : Attention freinage d'urgence

Yannick Danard
Damien Rivière