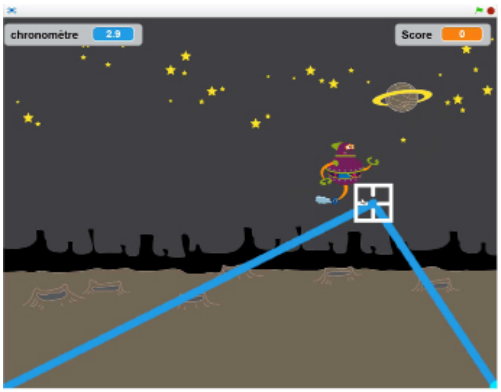
Robot Hunt

Vous allez créer une adaptation libre du jeu "Duck Hunt", sorti en 1985 sur console Nintendo.

Ce jeu demandera de programmer 4 lutins:



- Le lutin "Cible" contrôlé par le joueur.
- Le lutin "Robot" contrôlé par la machine.
- Le lutin "Game over" qui affiche la fin du jeu
- Le lutin "Laser" qui dessine le laser.

→ Ouvrez le fichier "BG2.sb2"

Partie 1 : les variables

Pour commencer, il faut créer et montrer les variables nécessaires. Il y en aura trois:

- "Score" qui comme son nom l'indique comptabilisera le nombre de robots détruits.
- "chronomètre" : cette variable existe déjà et se trouve dans le menu "Capteur". Il suffit ici de la montrer.

- "En mouvement" : cette variable prendra deux valeurs. Si elle vaut "oui", alors le robot pourra se déplacer. Si elle vaut "non", alors le robot ne bouge plus (ce qui se produira quand il sera touché par le laser).

Partie 2: la cible

Au lancement du programme, le lutin "Cible" se place en (0 ; 0) et passe au premier plan.
Puis, par l'intermédiaire d'une boucle de contrôle, il teste l'appuie des quatres flèches et se déplace en conséquence. Utilisez les blocs suivants pour construire le script.

Partie 3: Game Over

Au lancement du programme, le lutin "Game over" se cache. Il n'apparaîtra que lorsqu'il reçoit le message "Fin du jeu".. Écrivez le programme correspondant.

Partie 4: le robot en mouvement

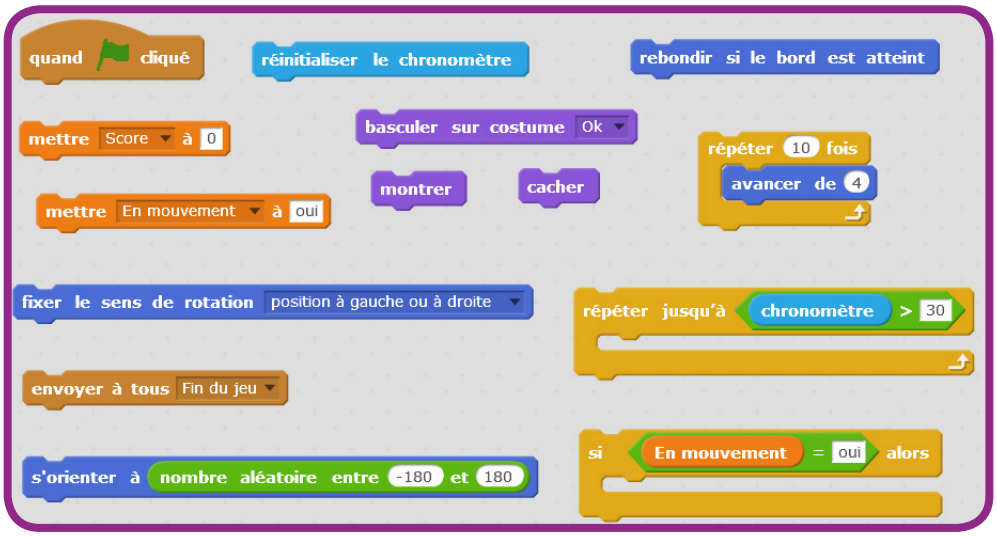
→ Au lancement du programme, c'est le robot qui initialise les variables. Il met le "score" à 0, la variable "En mouvement" à "oui", et réinitialise le chronomètre.

→ Il bascule sur son costume "ok", se montre, et fixe son sens de rotation en "gauche-droite" pour ne pas se retrouver la tête en bas en changeant de direction.

→ Il va ensuite répéter un déplacement aléatoire jusqu'à ce que le chronomètre dépasse 30 secondes. Pour cela, il va tester si la variable "En mouvement" vaut "oui", et si c'est le cas, il s'oriente dans une direction aléatoire comprise entre -180 et 180 degrés. Il avance alors de 40 unités de longueurs en 10 étapes. Puis, s'il atteint le bord de la scène, alors il rebondit..

→ Une fois les 30 secondes dépassées, il se cache et envoie à tous le message "Fin du jeu".

Utilisez les blocs suivants pour construire le programme:



Partie 5 : le laser

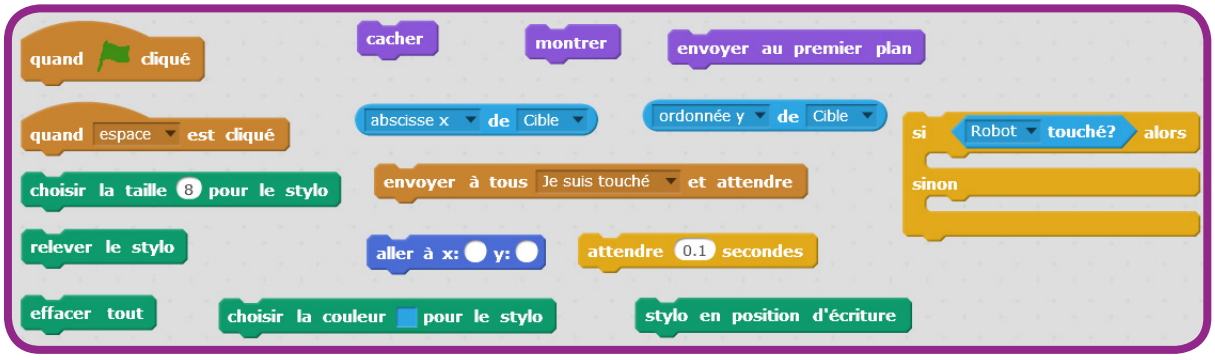
Le lutin "laser" va servir à deux choses: - dessiner les deux lasers vers la cible
- tester si ce lutin touche alors le robot.

→ Au lancement du programme, le lutin se cache.

→ Quand l'utilisateur appuie sur la touche "espace", alors il trace les lasers jusqu'à la cible:
- Il se montre et se place au premier plan.
- Il initialise son "stylo" en effaçant tout, choisissant une couleur bleue et une épaisseur de 8.
- Il se place en bas à gauche en (-240 ; -180) et se met en position écriture.
- Il va alors aux coordonnées de la cible, puis en bas à droite en (240 ; -180), où il se relève.
- Après une attente de 0,1 seconde, il efface tout.

→ Enfin, après s'être de nouveau placé aux coordonnées de la cible, il teste s'il la touche.
Dans ce cas, il se cache et envoie à tous le message "je suis touché".
Dans le cas contraire, il se cache simplement et le robot n'est pas touché.

Utilisez les blocs suivant pour programmer le laser:



Partie 6: le robot est touché

Quand le robot reçoit le message "je suis touché", il effectue les actions suivantes:

→ Il bascule sur son costume "boom", met la variable "En mouvement" sur "non", et ajoute 1 au score.

→ Il lance une animation de zoom qui s'efface progressivement.

→ Après une seconde d'attente, il se replace dans la scène à une postion aléatoire.

→ Il reprend alors son état initial, à savoir une taille de 45%, son costume "ok", une annulation des effets graphiques, et une remise à "oui" de la variable "En mouvement".

Utilisez les blocs ci-dessous pour écrire ce programme: