

PROJET Le singe gourmand



Un petit singe très gourmand adore les bananes ! Pour en ramasser le plus possible, il devra éviter les pastèques qui risquent de lui tomber sur la tête et l'assommer...

Étape 1 Créer un fond pour le jeu et gérer le déplacement du singe

- Choisir l'arrière-plan dans la bibliothèque du logiciel Scratch et sélectionner le lutin Monkey2 également dans la bibliothèque.
- Programmer Monkey2 pour qu'il apparaisse en bas au centre de l'écran à 50 % de sa taille et qu'il se déplace vers la droite ou vers la gauche grâce aux flèches de déplacement du clavier, mais sans jamais sortir de l'écran.

Coup de pouce

On peut vérifier lors de l'appui sur l'une des deux flèches ◀ et ▶ que l'abscisse du lutin n'a pas dépassé une certaine valeur pour éviter qu'il ne sorte de l'écran.



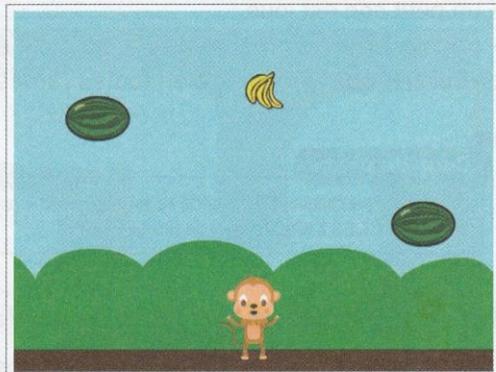
Étape 2 Faire tomber des bananes et des pastèques

- Rajouter un lutin *banane* (à 50 % de sa taille initiale) et un lutin *pastèque* (à 80 % de sa taille initiale).
- Programmer le lutin banane pour, qu'indéfiniment, cette banane apparaisse aléatoirement en haut de l'écran et glisse verticalement en un temps aléatoire jusqu'en bas de l'écran, où elle disparaît.
- Programmer de même le lutin pastèque, mais mettre une pause aléatoire avant qu'elle apparaisse.
- Dupliquer ensuite ce lutin pour augmenter le nombre de pastèques qui tombent.

Coup de pouce

Pour programmer une pause, on peut utiliser l'instruction

attendre nombre aléatoire entre 0 et 2 secondes



Étape 3 Gérer les contacts avec les bananes

Programmer le lutin banane pour qu'il se cache lorsqu'il touche le singe et qu'il ajoute 1 au nombre de bananes attrapées.

Coup de pouce

Créer une variable <BANANES> qu'il faudra remettre à zéro au début de chaque partie. Il est préférable de tester le contact dans une autre boucle infinie afin d'éviter quelques erreurs de comptage.



Étape 4 Gérer les contacts avec les pastèques

- Chaque contact avec une pastèque doit enlever une vie. Il faut donc créer une variable <VIES> qui démarre toujours à 5, par exemple, et à laquelle on enlève 1 chaque fois qu'une pastèque touche le singe.
- Le singe changera 3 fois de costume pour montrer qu'il a pris un coup.

Coup de pouce

Pour faire changer de costume, l'instruction `costume suivant` utilisée dans une boucle avec un tout petit temps d'attente est assez efficace.



Étape 5 Ajouter la possibilité de gagner des vies

- Ajouter un lutin en forme de cœur qui devra tomber comme les bananes et les pastèques et qui fera ajouter 1 aux vies lorsque il est attrapé par le singe.
- En revanche, plus le singe a de vies et plus les apparitions des cœurs doivent être rares.

Coup de pouce

On peut simuler le tirage au sort toutes les secondes d'un nombre entre 1 et 6 plus le nombre de vies pour raréfier les apparitions du cœur. Celui-ci n'apparaissant que quand le 1 sort par exemple.



Étape 6 Programmer la fin de partie

- La partie se termine quand le singe n'a plus de vies.
- Le singe annonce alors le nombre de bananes que le joueur a réussi à attraper.

Coup de pouce

On peut envoyer un message à tous les lutins lorsque le nombre de vies est égal à 0 pour leur dire de se cacher et de stopper leurs scripts.



Quelques évolutions possibles à ce projet

- Intégration d'un temps limite de jeu.
- Mémoriser le meilleur score quand le joueur a perdu.
- Faire un jeu à 2 joueurs où 2 singes pilotés au clavier doivent attraper le plus de bananes possibles.
- Ajouter d'autres fruits qui tombent, certains valent plus de points que d'autres...
- À vos idées !...