

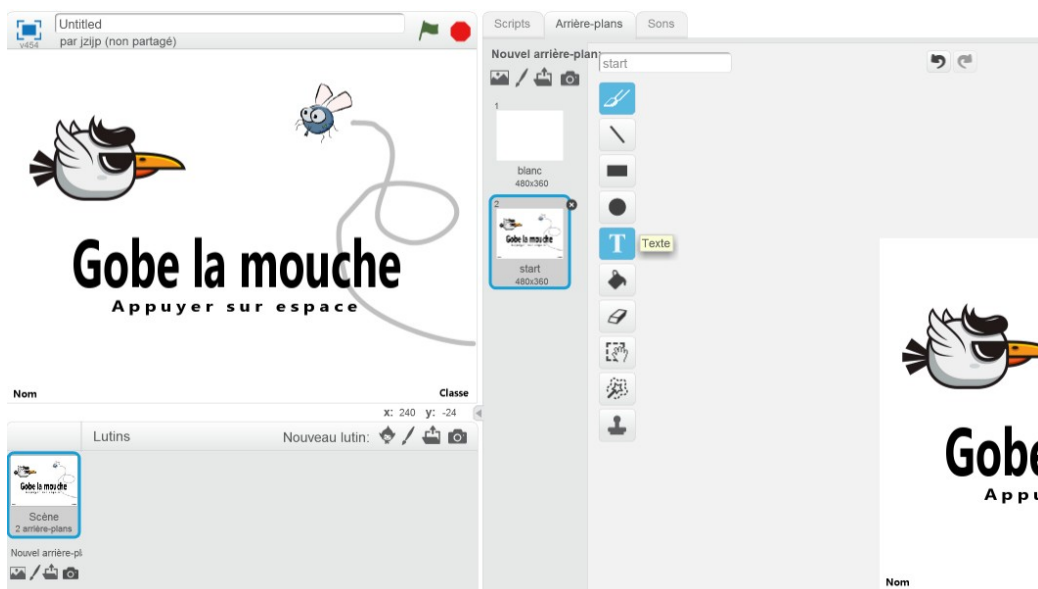


L'objectif est de réaliser « gobe la mouche », une version évoluée Flappy Bird. Laisse toi guider par ce document, mais n'hésites pas à adapter les valeurs et à corriger les bugs que tu pourras constater tout au long du codage : n'oublie pas de faire des tests.

## Écran de démarrage /2

1/ Supprime l'arrière plan existant et crée un arrière plan blanc que tu nommeras « blanc ».

2/ Récupère l'image start (enregistre-là sur ton ordinateur) . Cette image sera l'image de présentation au lancement du jeu. Place la comme deuxième arrière-plan de scène. Modifie la pour mettre ton nom et ta classe.



3/ Lorsque le drapeau sera cliqué (phase d'initialisation du jeu), il faudra afficher cette image. Crée donc un script qui effectue cette action



4/ Le joueur lancera le jeu en appuyant sur la touche espace. A ce moment c'est l'arrière-plan « blanc » qui doit s'afficher.



## Mouvements de l'oiseau /2

1/ Récupère l'image « bird » et place là comme premier lutin



2/ A l'initialisation du jeu (quand l'arrière plan bascule sur « start »), le lutin doit être caché.



3/ Lorsque le jeu commence (l'arrière plan bascule sur « blanc »), l'oiseau doit apparaître et venir se placer à la bonne taille sur la gauche de l'écran



4/ L'oiseau doit tomber indéfiniment. Il faut donc le déplacer vers le bas (vers les y négatifs) de façon continue.

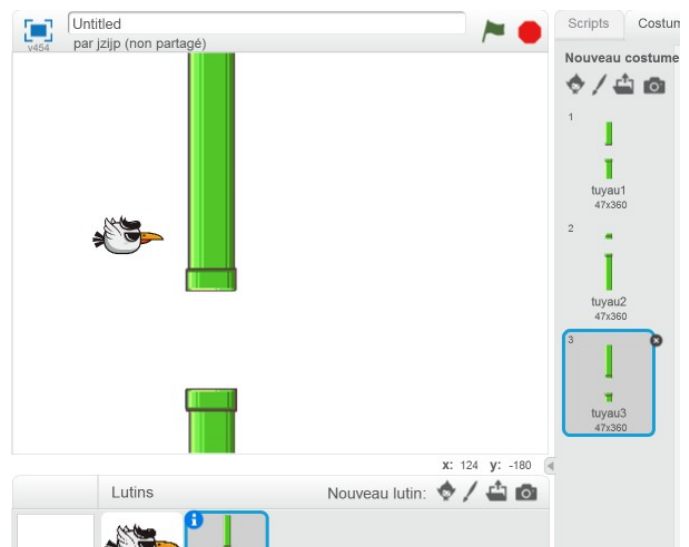


5/ Le joueur fait remonter l'oiseau en appuyant sur la touche espace



## Mouvement du tuyau /2

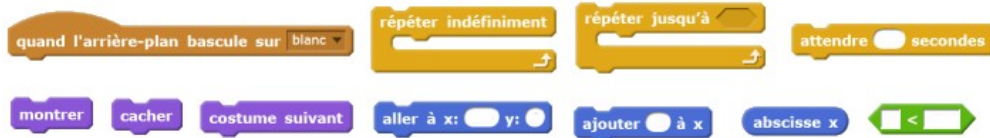
1/ Récupère les images tuyau1, tuyau2 et tuyau3 que tu définira comme 3 costumes d'un lutin que tu appellera « tuyau »



2/ A l'initialisation du jeu (arrière plan sur « start »), le tuyau doit être caché.

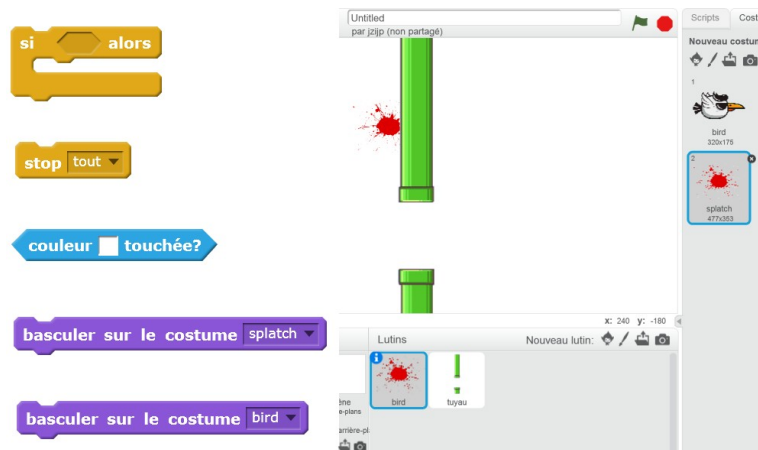
quand l'arrière-plan bascule sur start **cache**

3/ Lorsque l'arrière plan bascule sur « blanc », le tuyau doit se déplacer indéfiniment de la droite de l'écran vers la gauche de l'écran. On va donc répéter indéfiniment la séquence suivante : cacher > costume suivant (pour changer la hauteur de l'ouverture) > aller à droite de l'écran > attendre un moment > montrer > déplacer vers la gauche tant que la position du tuyau n'est pas complètement à gauche de l'écran.



## Fin du jeu /2

1/ Lorsque l'oiseau entre en collision avec le tuyau le jeu est fini. Pour détecter cette collision on va utiliser « si couleur ... touchée » dans la boucle qui fait tomber l'oiseau, on basculera le costume de l'oiseau sur splatch et on stoppera tous les scripts. Penser à bien mettre le costume de l'oiseau sur bird à l'initialisation.



## Score /2

Crée une variable « score », cache la et met la à 0 à l'initialisation. Montre la au lancement du jeu et ajoute lui 1 à chaque passage du tuyau.

**cache la variable score** **montre la variable score** **mettre score à 0** **ajoute à score 1**

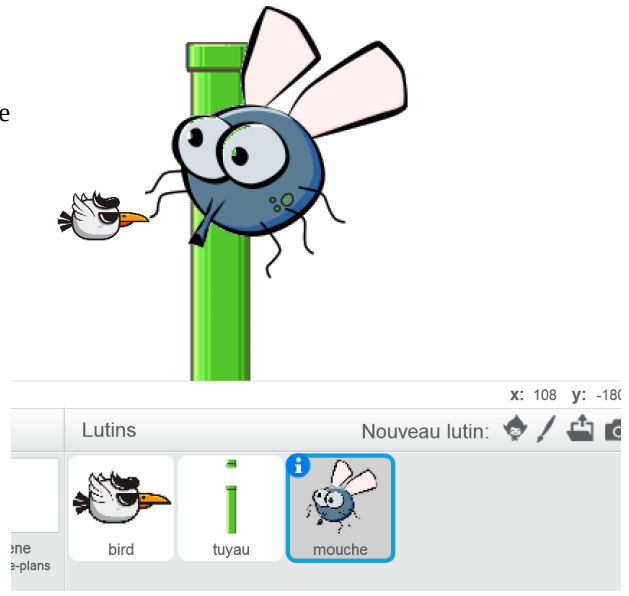
## Mouvements de la mouche /2

1/ Récupère l'image mouche et place là comme lutin que tu appellera « mouche ». A l'initialisation du jeu la mouche doit être cachée et mise à la bonne taille

quand l'arrière-plan bascule sur start

cache

mettre à 100 % de la taille initiale



2/ Le mouvement de la mouche est très similaire au mouvement du tuyau (dont tu peux t'inspirer) : Lorsque l'arrière plan bascule sur « blanc », la mouche doit se déplacer indéfiniment de la droite de l'écran vers la gauche de l'écran, mais aléatoirement :

- La mouche n'apparaît pas directement à droite après avoir disparue à gauche : elle attend un temps aléatoire
- La mouche doit apparaître à droite à une hauteur aléatoire,
- Dans son déplacement, la mouche tourne aléatoirement

quand l'arrière-plan bascule sur blanc

répéter indéfiniment

attendre 1 secondes

répéter jusqu'à

nombre aléatoire entre 1 et 10

nombre aléatoire entre 1 et 10

nombre aléatoire entre 1 et 10

cache

montrer

abscisse x

tourner de 1 degrés

avancer de 1

aller à x: 1 y: 1

## Gobage de la mouche /2

1/ Crée une variable gobage de la mouche que tu cacheras et mettras à 0 à l'initialisation, et que tu montreras au lancement du jeu

cache la variable gobages

mettre gobages à 0

montrer la variable gobages

2/ Dans la boucle qui fait avancer la mouche, ajoute « si la distance à l'oiseau est inférieure à une certaine valeur » alors ajoute 1 à la variable « gobage » et 3 à la variable « score », cache la mouche et déplace là complètement à gauche (essaie d'expliquer pourquoi cette dernière instruction est nécessaire)

si distance de bird < 10 alors

ajouter à gobages 1

ajouter à score 3

cache

donner la valeur 0 à x

## Niveaux de jeu /2

1/ Créer une variable « passages » qui comptera le nombre de passages du tuyau : cacher cette variable, la mettre à 0 à l'initialisation du jeu puis ajouter 1 à passages lors de chaque passage du tuyau.



2/ On va tester la variable « passages » après chaque passage du tuyau, et si elle est un multiple de 10 on va envoyer un message « level » à tous les scripts pour indiquer qu'on change de niveau. Pour tester qu'un nombre x est un multiple d'un nombre y, on utilise l'opérateur « x modulo y », qui donne le reste de la division de x par y.



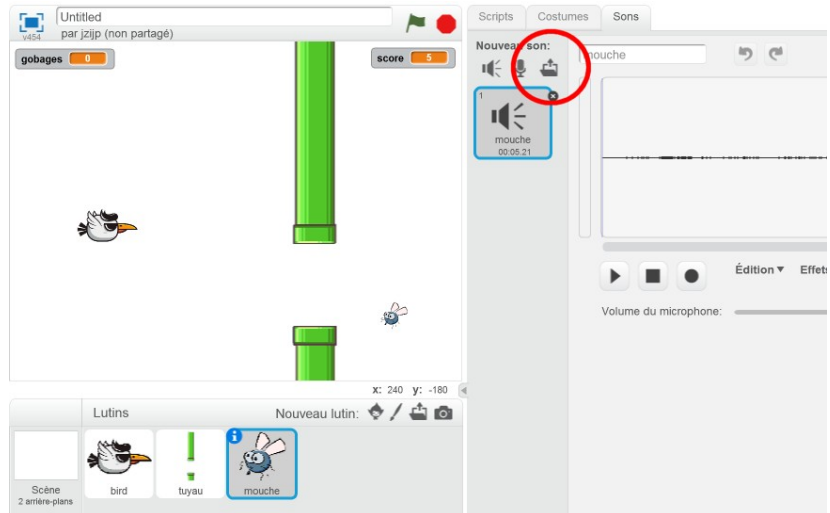
2/ Augmente légèrement la taille de l'oiseau et change la couleur du fond à chaque nouveau niveau



## Sons /2

Ajoute tous les sons « Boing » (chaque fois que tu fait remonter l'oiseau), « Snap » (lorsqu'il y a une collision entre l'oiseau et le tuyau), « Gober » (Lors une mouche est gobée) et « Mouche » (Lors du passage d'une mouche)

Exemple pour le son « Mouche » : Tu dois jouer le son mouche lorsque la mouche apparaît, et l'arrêter lorsque la mouche disparaît



## Bonus /2

Réalise une amélioration du jeu de ton choix.