

Quiz et bonus de l'Étape 1



As-tu compris ?

- 1 Comment peux-tu démarrer/arrêter le programme ? quels blocs dois-tu utiliser ?
- 2 Dans quelle palette se trouvent les blocs pour fixer la position des lutins ou les faire bouger sur la scène ?
- 3 Quelle est la différence entre le bloc `aller à` et le bloc `glisser en` ?
- 4 Si tu veux faire apparaître un lutin en plein milieu de l'écran, quelles valeurs dois-tu utiliser dans le bloc `aller à x: y:` ?
- 5 Quelle est la différence entre un lutin et un costume ?
- 6 Pour animer un personnage (par exemple, Lila ou le chat Scratchy), quels blocs de la palette Apparence vas-tu utiliser ?

Entraîne-toi !

- 1 Fais flotter un drapeau dans le vent. Pour cela, ouvre le fichier Bonus1.sb2 qui est dans le dossier Bonus. Il contient déjà le lutin Drapeau. Tu peux utiliser les blocs `ajouter à l'effet tourner` et `ajouter à l'effet oeil de poisson` de la palette Apparence pour faire gonfler et tourner le drapeau dans un sens puis dans l'autre.
- 2 Enregistre un son pour l'aéroport. Enregistre ton annonce au haut-parleur : « embarquement immédiat, porte 47 ». Tu peux aussi essayer en anglais ! Fais jouer ce son quand Tom et Lila arrivent à l'enregistrement.
- 3 Crée un lutin Panneau comme dans les aéroports pour indiquer les départs avec la destination et le numéro des vols, avec 2 costumes : l'un indiquant l'enregistrement, l'autre l'embarquement immédiat. Tu peux le dessiner en utilisant le pinceau du menu Nouveau lutin située dans la liste des lutins. Ajoute ce lutin et ses deux costumes au fichier Etape1_complet.sb2 (la solution du jeu de l'Étape 1 qui est dans le dossier Jeux_complets).

- 4 Ajoute-lui un script permettant de le faire apparaître dans le hall de l'aéroport et de changer son costume quand l'enregistrement est fini pour indiquer l'embarquement immédiat.
- 5 Lance le fichier Bonus1_Lila.sb2 du dossier Bonus. Que dessine Lila ? Essaie de comprendre comment fonctionne le script.

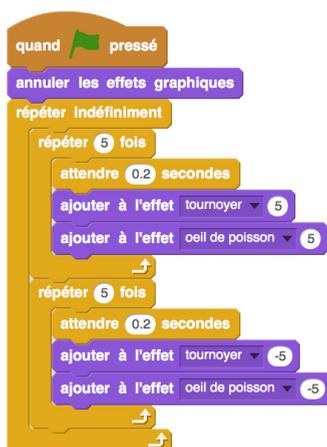
Réponses à « As-tu compris ? »

- 1 Tu démarres le programme en cliquant sur le drapeau vert en haut à droite de la scène. Cela correspond au bloc  de la palette Événements. Tu arrêtes le programme en cliquant sur le bouton rouge juste à côté du drapeau vert. Cela correspond au bloc  de la palette Contrôle.
- 2 Les blocs qui permettent de déplacer les lutins sont dans la palette Mouvement. Souviens-toi que tu peux lire les coordonnées du pointeur de souris en bas à droite juste sous la scène. Si tu es perdu, relis la page 13 du livre.
- 3 Le bloc  positionne immédiatement le lutin à une position définie par les valeurs des coordonnées x et y. Le lutin change de position instantanément. En revanche, le bloc  fait se déplacer le lutin depuis l'endroit où il se trouve vers la position définie par les coordonnées x et y. Tu vois le lutin se déplacer sur l'écran. Le temps mis pour glisser définit la vitesse du déplacement.
- 4 Pour que ton lutin soit au centre de la scène, il doit être positionné en $x = 0$ et $y = 0$. Attention, ce n'est vrai que si le dessin de ton lutin est bien centré dans son cadre. Tu peux définir le centre de ton lutin avec l'outil , qui est situé en haut à droite de la barre d'outils de dessin pour modifier un costume (voir page 15 du livre). Teste cet outil pour comprendre comment il fonctionne !
- 5 Un lutin est un objet ou un personnage pour lequel tu vas écrire des programmes (appelés aussi scripts), à partir des blocs de l'onglet Scripts. Chaque lutin peut avoir un ou plusieurs scripts qui lui sont associés. Un costume correspond à une des images associées au lutin. Un lutin peut avoir plusieurs costumes que tu peux voir et modifier à partir de l'onglet Costumes. Les costumes permettent de réaliser des animations des lutins, en changeant rapidement l'image (= un costume) associée au lutin.

- 6 Pour créer une animation, tu vas basculer rapidement d'un costume à l'autre. L'un des premiers programmes Scratch que tu peux écrire consiste à faire courir le chat Scratchy en basculant rapidement entre ses deux costumes où ses pattes ne sont pas dans la même position. Tu dois donc utiliser les blocs **basculer sur costume** ou **costume suivant** de la palette Apparence. Attention, pense à ajouter un bloc **attendre 0.2 secondes** (de la palette Contrôle) entre chaque costume, sinon ça ira tellement vite que tu ne verras rien !

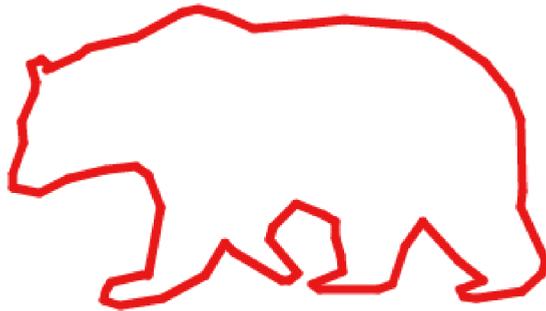
Réponses à « Entraîne-toi ! »

- 1 Utilise le bloc **ajouter à l'effet tourner** de la palette Apparence. Répète ce bloc plusieurs fois en changeant la valeur de l'effet (+2 par exemple, puis -2) et tu verras le lutin se déformer comme s'il tournait. Pour un drapeau, ça donne l'impression qu'il flotte au vent. Tu peux aussi utiliser l'effet œil de poisson pour « gonfler » ou « dégonfler » le drapeau. Pour annuler l'effet graphique et revenir à l'image du départ, ajoute au début de ton programme le bloc **annuler les effets graphiques**.

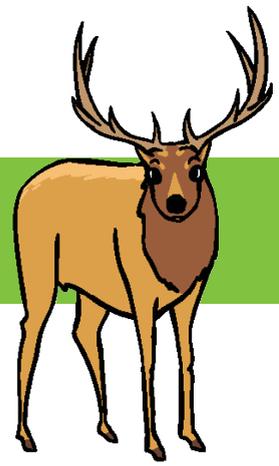


- 2 Ce programme appelé Bonus1_solution.sb2 se trouve dans le dossier Bonus.
- 3 Pour ajouter un son, il faut que tu ailles dans l'onglet Sons pour le lutin de Lila, par exemple. Tu peux enregistrer le son avec le micro de ton ordinateur ou utiliser un son déjà enregistré. Il existe des banques de sons et de bruitages gratuits sur internet.

- 4 Indice : tu dois utiliser les blocs `envoyer à tous message1` et `quand je reçois message1` de la palette Évènements. Tu fais apparaître le lutin avec le bloc `montrer` quand il reçoit le message `Aéroport`, puis tu le fais changer de costume (avec le bloc `basculer sur costume`) lorsqu'il reçoit le message `Enregistrement`.
- 5 Dans le script du fichier Bonus1_Lila.sb2, Lila dessine un ours noir. Le programme utilise plusieurs blocs de la palette Stylo pour dessiner l'ours, point par point. On ajoute des blocs `attendre secondes` pour que le dessin se fasse petit à petit.



Quiz et bonus de l'Étape 2



As-tu compris ?

- 1 Quels blocs peux-tu utiliser pour démarrer un script si une condition est vraie ? Dans quelles palettes sont-ils ?
- 2 Quelle est la différence entre le bloc `si` `alors` et le bloc `si` `alors` `sinon` ?
- 3 Quel bloc de la palette Opérateurs dois-tu utiliser si tu veux tester deux conditions qui doivent être vraies en même temps ? Par exemple, pour écrire, il faut un crayon et une feuille de papier. Sans crayon ou sans feuille, c'est difficile !
- 4 Quel bloc de la palette Opérateurs dois-tu utiliser si tu veux créer un test où il suffit qu'une condition parmi plusieurs soit vraie ? Par exemple, tu es content si tu manges du gâteau au chocolat ou si tu vas voir un film au cinéma (remarque, dans ce cas, que tu seras sûrement content si tu manges du gâteau ET que tu vas au cinéma...).
- 5 Quel bloc de la palette Opérateurs dois-tu utiliser si tu veux créer un test qui vérifie qu'une variable est plus grande qu'une certaine valeur fixée ? Par exemple, pour aller sur les manèges de la fête foraine, tu dois avoir plus de 7 ans.
- 6 Pourquoi doit-on (presque) toujours mettre les blocs `si` `alors` et `si` `alors` `sinon` à l'intérieur d'un bloc `répéter indéfiniment` ?

Entraîne-toi !

Ton défi : ajouter au programme complet de l'étape 1 un avion qui suit la trajectoire entre Paris et Denver. Comme dans un vrai avion (as-tu déjà pris l'avion ?), tu vas suivre la progression de sa trajectoire pendant le voyage sur une carte.

Ouvre le fichier Bonus2.sb2. Si tu regardes avec attention le costume du lutin Avion, tu remarques que son « nez » est d'une couleur légèrement différente (marron foncé). Ici, il faut que tu utilises le bloc  (3^e bloc en partant du haut dans la palette Capteurs) comme condition dans le bloc   pour que le nez de l'avion (marron) ne s'éloigne jamais de la ligne de la trajectoire (rouge). Dans ce bloc , tu peux choisir les couleurs en cliquant dans le petit carré du bloc puis en cliquant sur un pixel de la couleur que tu veux.

Un indice supplémentaire : tu dois faire avancer l'avion lentement vers la gauche (, par exemple) et à chaque fois qu'il a avancé, tu dois tester si le nez touche bien la trajectoire. Si c'est le cas, tu continues à avancer, sinon tu dois le faire tourner légèrement (bloc ) dans un sens puis dans l'autre (bloc ), en testant à chaque fois si tu es bien revenu sur la trajectoire. Maintenant, à toi de jouer !

Réponses à « As-tu compris ? »

- 1 Tu peux utiliser les blocs , ,  ou . Tous ces blocs se trouvent dans la palette Contrôle. Ils servent à contrôler qu'une partie du script ne peut être réalisée que lorsque la condition testée est vraie. Ils possèdent une case en forme d'hexagone, dans laquelle tu dois glisser un bloc (de la palette Capteurs ou Opérateurs) qui a également une forme d'hexagone et permet de définir la condition à vérifier.
- 2 Avec un bloc , tu peux définir ce qui se passe si la condition est vraie et ce qui se passe si elle est fausse. Avec un bloc , tu définis seulement ce qui se passe si la condition est vraie.
- 3 Tu dois utiliser l'opérateur .
- 4 Tu dois utiliser l'opérateur .
- 5 Tu dois utiliser l'opérateur .
- 6 Le bloc  permet de tester une condition pendant toute la durée du programme (ou du jeu). Par exemple, si tu écris un programme pour déplacer un personnage avec les touches Flèches du clavier, il faudra qu'à chaque fois que tu appuies sur une flèche, le personnage se déplace. Mais

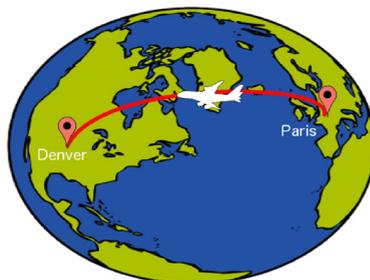
si tu écris seulement `si touche flèche droite pressée? alors avancer de 10`, sans mettre le tout dans une

boucle `répéter indéfiniment`, alors ton programme ne fonctionnera pas. Il testera une seule fois si la touche Flèche droite est appuyée, puis il s'arrêtera.

Réponses à « Entraîne-toi ! »

Voici le script de la solution, qui est dans le fichier Bonus2_solution.sb2 du dossier Bonus. Au début, tu orientes l'avion dans la bonne direction et tu le positionnes sur ses coordonnées de départ (Paris) sur l'arrière-plan. Puis tu vas tester si le nez de l'avion touche la trajectoire et dans ce cas tu le fais avancer vers la gauche (de -5). Si le nez ne touche pas la trajectoire, tu tournes légèrement l'avion dans le sens des aiguilles d'une montre (de 3°). Si dans ce cas le nez est de nouveau en contact avec la trajectoire, alors l'avion avance (de -5) sinon on tourne dans l'autre direction, soit -6° (on doit en effet revenir à 0 en enlevant 3, puis enlever encore 3). La condition `abscisse x < -130` permet d'arrêter l'avion quand il arrive à Denver, sinon le script fait faire un tour complet à l'avion qui repart vers Paris !

```
quand flag pressé
  s'orienter à 90
  aller à x: 179 y: 55
  répéter jusqu'à abscisse x < -115
    si couleur touche ? alors
      avancer de -5
      attendre 0.2 secondes
    sinon
      tourner de 5 degrés
      si couleur touche ? alors
        avancer de -5
        attendre 0.1 secondes
      sinon
        répéter jusqu'à couleur touche
          tourner de -5 degrés
```



Quiz et bonus de l'Étape 3



As-tu compris ?

- 1 Comment peux-tu créer une variable dans Scratch ? Souviens-toi, une variable est comme une petite boîte où tu stockes de l'information : un nombre ou un mot, par exemple.
- 2 Avec quel bloc peux-tu stocker de l'information dans une variable ? Par exemple, donne la valeur 10 à la variable Age que tu créeras.
- 3 Quels blocs de la palette Capteurs dois-tu utiliser pour savoir l'heure qu'il est ?
- 4 Comment peux-tu faire dire l'heure à un lutin ?
- 5 Quel bloc de la palette Capteurs dois-tu utiliser pour poser une question et attendre la réponse ? Dans Scratch, comment peux-tu accéder à la réponse que le joueur a tapée au clavier ?
- 6 À quoi sert le bloc `nombre aléatoire entre 1 et 10` de la palette Opérateurs ? Et ce même bloc avec différentes valeurs, par exemple entre 1 et 6 ?
- 7 À quoi sert le bloc `arrondi de` de la palette Opérateurs ?

Entraîne-toi !

Ton défi : créer un chronomètre avec trois boutons Début, Stop et Remise à zéro. Ouvre le fichier Bonus3.sb2 qui contient déjà les lutins.

Crée d'abord trois variables : Secondes, Minutes et Stop. Les deux premières serviront à stocker le temps qui s'écoule à partir du moment où tu appuies sur le bouton Début. La dernière servira à arrêter le programme lorsqu'on clique sur le bouton Stop. Pour programmer tes lutins boutons, tu vas utiliser le bloc `quand ce lutin est cliqué` de la palette Événements, exactement comme dans le livre page 32 pour déclencher l'éruption.

Pour le lutin du bouton Début, il faut que tu utilises une boucle qui contient le bloc **attendre 1 secondes** puis le bloc **ajouter à Secondes**. Attention, tu dois utiliser un bloc **si** **alors** pour tester si on dépasse 60 secondes, c'est-à-dire une minute ! Dans ce cas, tu dois utiliser un bloc **mettre Secondes à 0** pour remettre la variable Secondes à 0 et un bloc **ajouter à Minutes** pour ajouter 1 à la variable Minutes. Attention, au début du programme, les variables Secondes et Minutes doivent être mises à 0. Enfin, quand tu cliques sur le bouton Remise à zéro, pense à remettre les variables Secondes et Minutes à 0, sans arrêter le chronomètre.

Réponses à « As-tu compris ? »

- 1 Pour créer une variable, il faut cliquer sur la palette Données, puis sur le bouton Créer une variable. Tu peux alors taper au clavier le nom de la variable. Tu peux choisir si la variable est « disponible » pour tous les lutins (c'est ce qu'on a choisi pour toutes les variables du livre) ou seulement pour le lutin actif.
- 2 Tu peux stocker de l'information avec le bloc **mettre** qui est dans la palette Données.
- 3 Tu dois utiliser deux blocs **actuel**, en choisissant heure pour le premier et minutes pour le second grâce au menu déroulant accessible par la petite flèche noire. Tu remarques que tu peux aussi obtenir le mois, le jour de la semaine, l'année...
- 4 Tu dois d'abord créer deux variables : Heure et Minutes. Puis tu stockes dans chacune les valeurs obtenues avec deux blocs **actuel** (voir la réponse précédente). Pour que le lutin dise l'heure, tu dois utiliser un bloc **dire** de la palette Apparence. Enfin, tu dois utiliser plusieurs fois le bloc **regroupe** de la palette Opérateurs pour assembler les différents morceaux de la phrase que doit dire ton lutin.



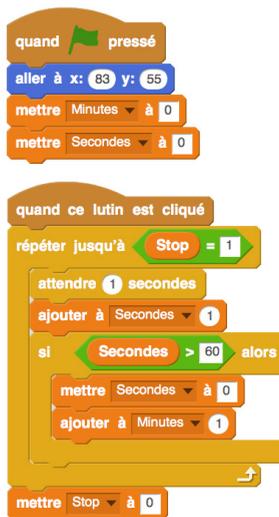
- 5 Tu dois utiliser le bloc **demande et attends** de la palette Capteurs. La réponse entrée au clavier est stockée dans la variable **réponse** qui est déjà créée dans Scratch et disponible dans la palette Capteurs.

- 6 Le bloc **nombre aléatoire entre 1 et 10** permet de tirer un nombre au hasard entre 1 et 10. À chaque tirage, tu as autant de chance de tomber sur un chiffre ou sur un autre entre 1 et 10. Si tu choisis un nombre aléatoire entre 1 et 6, c'est exactement comme si tu lançais un dé.
- 7 Quand tu appliques une opération mathématique, comme une division, tu peux arriver sur un nombre qui n'est pas entier : par exemple, si tu divises 5 par 4, tu trouves 0,8. Le bloc **arrondi de** permet de trouver le chiffre entier le plus proche. Ici, le bloc **arrondi de 5 / 4** donne le plus proche nombre entier de 0,8, c'est-à-dire 1. Essaie avec différentes valeurs pour bien comprendre.

Réponses à « Entraîne-toi ! »

Voici les scripts de la solution, qui sont dans le fichier Bonus3_solution.sb2 du dossier Bonus.

Scripts du lutin bouton Début :



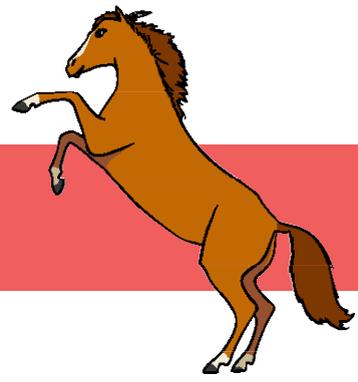
Script du lutin bouton Stop :



Script du bouton Remise à zéro :



Quiz et bonus de l'Étape 4



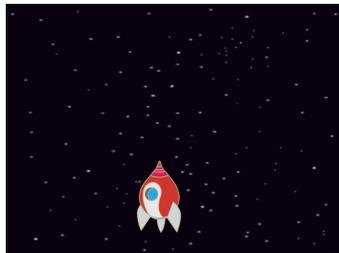
As-tu compris ?

- 1 Quels blocs de la palette Capteurs peux-tu utiliser pour vérifier qu'un lutin touche un autre lutin ? (il y a au moins deux réponses possibles)
- 2 Dans quelle palette se trouve le bloc **chronomètre** ? Quel est le bloc qui permet de remettre le chronomètre à zéro ?
- 3 Quel bloc de la palette Capteurs dois-tu utiliser pour tester si une touche du clavier a été enfoncée ?

Entraîne-toi !

Ton défi : écris les scripts pour créer une fusée qui se déplace dans l'espace. Ouvre le fichier Bonus4.sb2 qui contient déjà les lutins Fusée, Etoiles1 et Etoiles2.

Commence par programmer le déplacement de la fusée avec les touches du clavier : si tu appuies sur la touche Flèche droite, tu dois te déplacer vers la droite, et ainsi de suite avec les quatre flèches. Ensuite, tu dois utiliser les deux lutins Etoiles1 et Etoiles2 pour faire défiler la galaxie et donner l'impression du mouvement comme dans le jeu de l'étape 4 avec les lutins Piste1 et Piste2. La seule différence ici est que tu dois faire défiler le décor du haut vers le bas et pas de droite à gauche comme dans le livre. Il faut que tu crées une variable Défilement puis que tu lui donnes les bonnes valeurs. Pour t'aider, tu peux copier et modifier les scripts de la page 44 du livre.



Réponses à « As-tu compris ? »

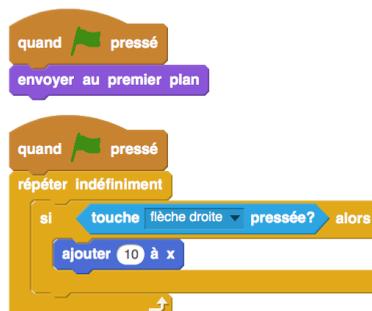
- 1 Tu dois utiliser un bloc **si** **alors** et bien choisir la condition dans la palette Capteurs.

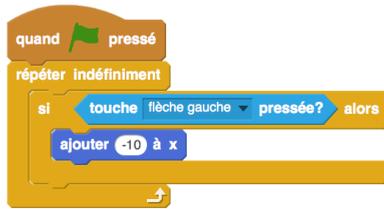
 - La condition peut être donnée par le bloc **touché?**. Tu choisis le lutin testé à l'aide du menu déroulant accessible par la petite flèche noire.
 - Tu peux aussi utiliser le bloc **distance de** qui est une variable qui contient la distance entre le lutin dont tu écris le script et un autre lutin du programme. Ici aussi, tu peux choisir le lutin grâce à un menu déroulant accessible par la petite flèche noire. En testant si la distance est plus petite que quelques pixels, tu peux savoir si ton lutin touche l'autre.
 - Tu peux enfin utiliser le bloc **couleur touche ?** de la palette Capteurs, en choisissant comme couleurs celles des bords des deux lutins. Choisis les couleurs en cliquant sur les petits carrés de couleur du bloc, puis sur le pixel de la couleur de ton choix n'importe où sur l'écran de ton ordinateur !
- 2 Le chronomètre se trouve dans la palette Capteurs. Le bloc **réinitialiser le chronomètre** permet de le remettre à 0.
- 3 Tu dois utiliser un bloc **si** **alors** et glisser dedans la condition donnée par le bloc **touche** **pressée?**. Avec le menu déroulant, tu peux choisir quelle touche du clavier tu veux utiliser.

Réponses à « Entraîne-toi ! »

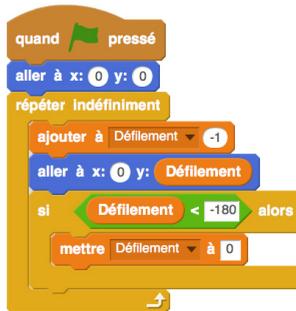
Ouvre le fichier Bonus4_solution.sb2. Voici les différents scripts à écrire. Tu peux maintenant programmer un jeu à partir de ce fichier : à toi de jouer !

Scripts du lutin Fusée :





Scripts du lutin Etoiles1 :



Script du lutin Etoiles2 :

