



SCRATCH

INITIATION A SCRATCH



Table des matières

Initiation à Scratch	3
1. Présentation	4
2. Réalisation des mini-projets	8
2.1. Projet n° 1	8
2.2. Projet n° 2	9
2.3. Projet n° 3	12
2.4. Projet n° 4	13
2.5. Projet n° 5	16
2.6. Projet n° 6	17
2.7. Projet n° 7	19
2.8. Projet n° 8	21
2.9. Projet n° 9	23
2.10. Projet n° 10	25
2.11. Projet n° 11	27
2.12. Projet n° 12	29
2.13. Projet n° 13	31
2.14. Projet n° 14	33
2.15. Projet n° 15	34
2.16. Projet n° 16	36

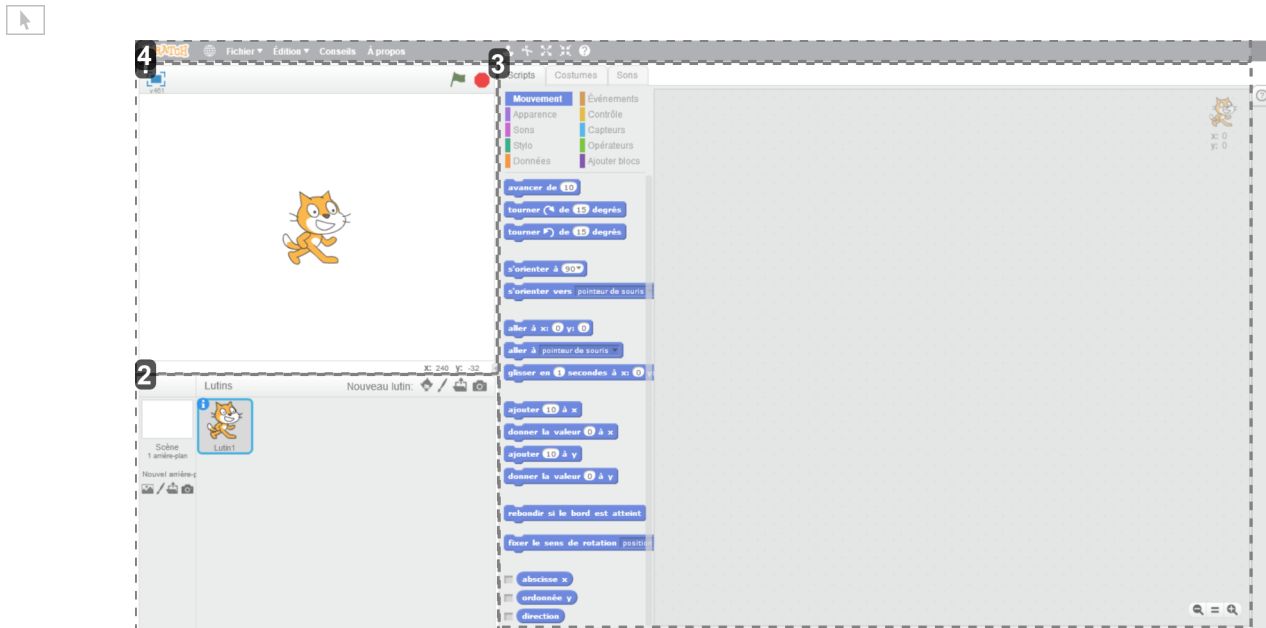
Initiation à Scratch

Initiation au logiciel de programmation **Scratch** sous forme de **16 petits projets**.

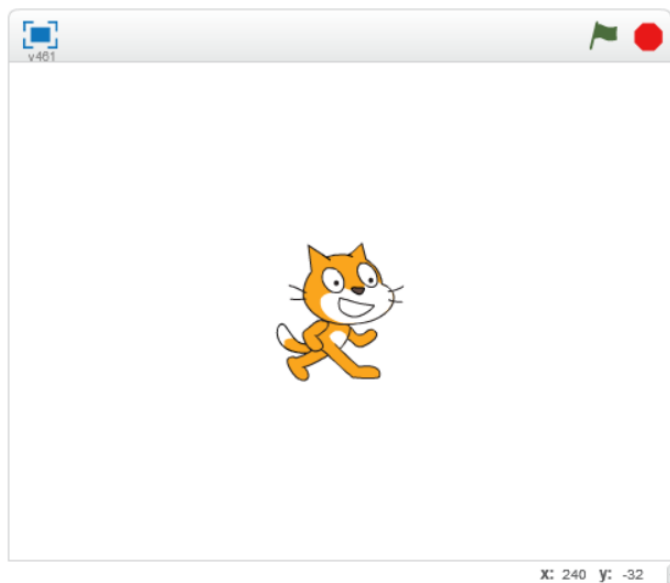


Scratch

1. Présentation



1 La scène



C'est la zone dans laquelle on pourra visualiser les effets du programme.

Dans la version « en ligne » il y a une ligne de titre à compléter (il faut remplacer le mot « untitled »).

Dans la version « hors ligne », si le fichier est nouveau, il faudra « l'enregistrer sous » et lui donner un nom.

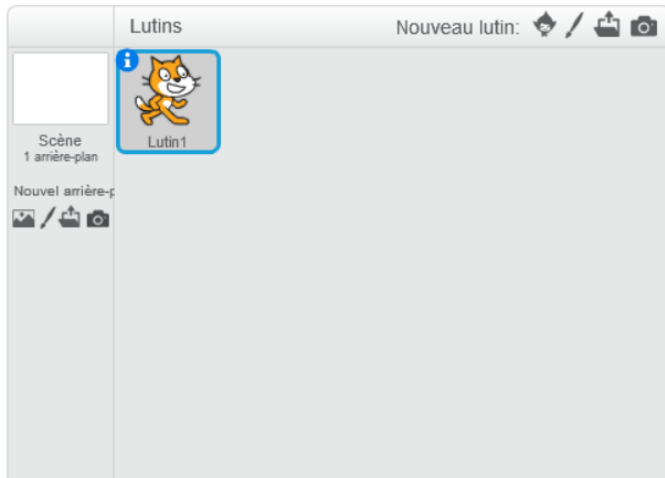
Le drapeau vert permet de lancer le programme (si la commande associée à ce bouton y a été insérée).

Le panneau rouge permet d'arrêter le programme.

Le rectangle bleu permet d'afficher la scène en plein écran. Cette opération est réversible depuis une icône similaire.

En bas de la scène, à droite, on trouve les coordonnées (x et y) du personnage qui a été sélectionné dans la zone de gestion des lutins.

2 La zone de gestion des lutins et des arrière-plans

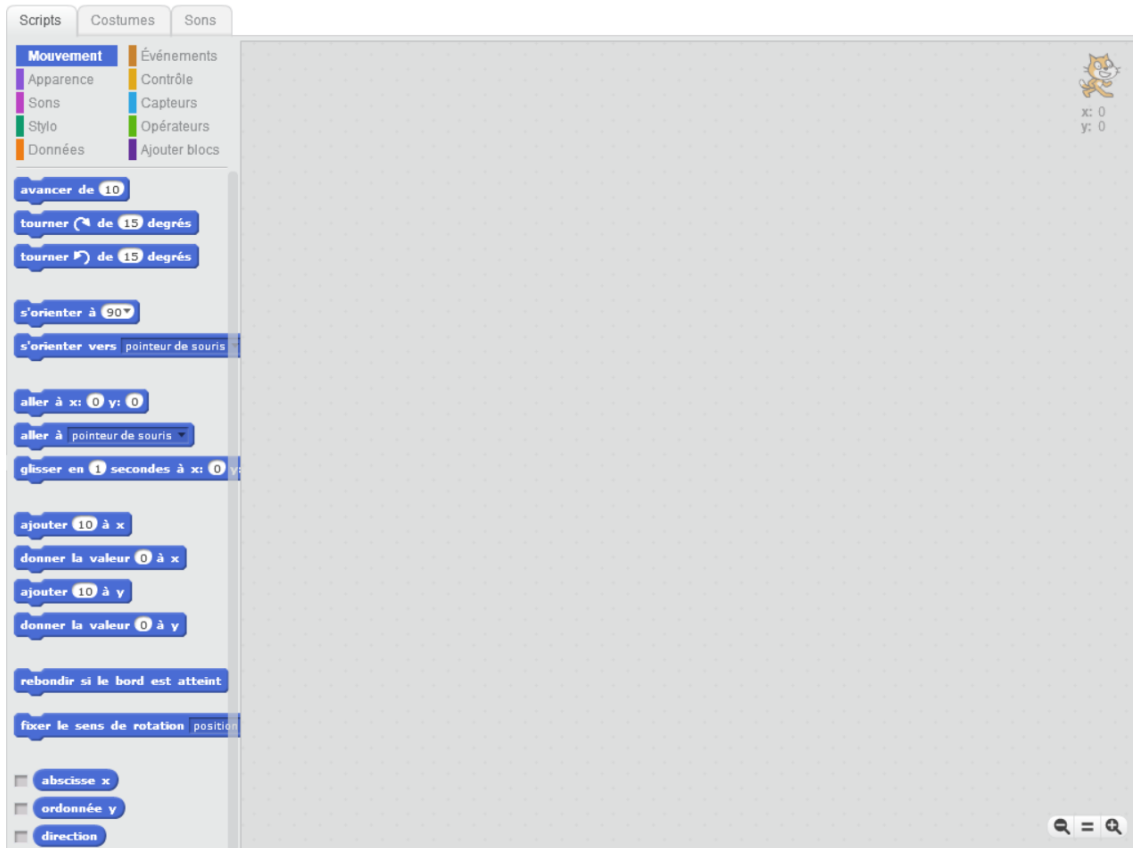


Chaque personnage est appelé « **lutin** ». Il est possible d'insérer plusieurs lutins qui pourront être gérés par des programmes différents. **Pour visualiser le programme attaché à un lutin, il est nécessaire de le sélectionner au préalable.**

Les lutins peuvent être supprimés, dupliqués, cachés. Il est possible d'en insérer de nouveaux, soit à partir de la banque proposée par Scratch, soit à partir de nos propres images (au moyen de l'icône « dossier »).

A côté de la zone des lutins (à gauche), se trouve celle des arrière-plans utilisés. Il est également possible d'en insérer de nouveaux, soit à partir de la banque proposée par Scratch, soit à partir de nos propres images (au moyen de l'icône « dossier »). Plusieurs arrière-plans peuvent être enregistrés

3 La zone des scripts



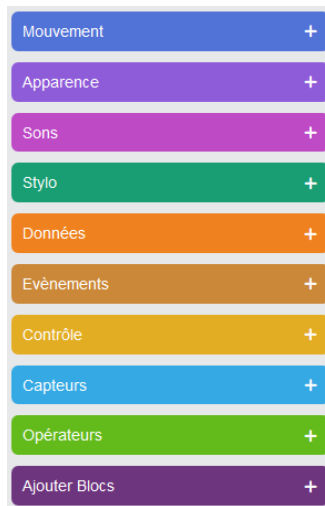
C'est la zone dans laquelle on écrira les différents programmes à partir des palettes de blocs proposés dans l'onglet « Scripts ». Il y a 10 palettes différentes, chacune est identifiée par une couleur particulière.

L'onglet « Costumes » présente les différentes apparences d'un lutin. L'utilisateur peut ajouter des costumes à un lutin en créant des variations de ce lutin avec un logiciel de dessin. Il suffira par la suite de les introduire dans l'onglet « Costumes » (au moyen de l'icône « dossier »).

L'onglet « Sons » permet de gérer les sons qui seront éventuellement utilisés dans les programmes.

Le bouton « Partager » permet de publier en ligne le projet réalisé.

Le bouton « Montrer la page du projet » permet de créer une fiche de présentation du projet publié.



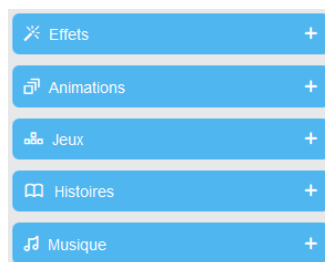
Palette

4 La barre de menu et la barre d'outils



Un clic sur la **planète** permet de modifier la langue de l'application. Il est assez intuitif d'utiliser les autres menus et outils.

Dans Scratch, les projets réalisés peuvent être classés dans **5 catégories**.



Catégories

Utiliser de nouvelles illustrations

Scratch est livré avec un certain nombre d'**arrière-plans et de lutins**, mais il est possible d'utiliser des images personnelles. Pour insérer de nouveaux arrière-plans, il est préférable d'utiliser des images qui ont une **dimension de 480x360 pixels**. Les lutins auront une taille inférieure qui pourra être directement redimensionnée depuis l'onglet « **Costumes** ».

2. Réalisation des mini-projets

2.1. Projet n° 1

Difficulté



Nom du projet

Faire voler le perroquet.

Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
01- Faire_voler.sb2

Objectif

Reproduire et modifier un programme pour créer une animation.



1 Description

La chauve-souris semble voler parce que le logiciel affiche 2 images, l'une après l'autre. Saurais-tu t'aider du programme de la chauve-souris pour reproduire cette illusion en réalisant un programme qui fasse voler le perroquet ?

2 Ce qu'il faut savoir

Chaque lutin est animé par un programme différent. Il faut vérifier que l'on a sélectionné le bon lutin avant de commencer le programme.

Pour connaître les costumes d'un lutin, il faut cliquer sur l'onglet « costumes ».

Le crochet « répéter indéfiniment » s'appelle une **boucle**. Il permet de reproduire la même action autant de fois qu'on le désire.

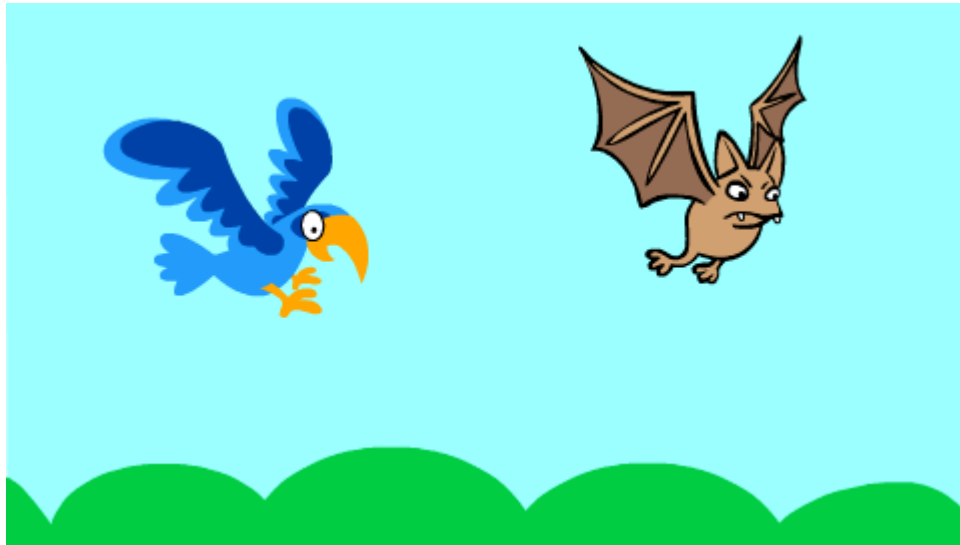
En cliquant sur le drapeau vert, le programme est lancé. En cliquant sur le panneau rouge, le programme est arrêté.

3 Palettes à utiliser



4 Blocs utiles



 Résultat

Résultat projet n°1

 Astuce

Il est nécessaire d'insérer des temps d'attente après le changement de costume pour que l'on puisse **voir** le programme les réaliser. Sans ces temps d'attente, les ordres sont bien respectés, mais (si l'ordinateur est puissant) ils le sont si rapidement que nos yeux n'arrivent pas à les voir.

 Pour aller plus loin ...

Un temps d'attente d'une seconde est trop long. Pour que les ailes battent plus vite, il est nécessaire de le raccourcir. Par exemple « 0.3 seconde » est plus rapide.

Tu peux insérer des temps différents pour voir le résultat de ces modifications (entre 0.1 et 0.9).

 Attention

Dans Scratch 0,3 s'écrit avec un point (0.3)

2.2. Projet n°2

 Difficulté **Nom du projet**

Faire danser la ballerine

 Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
02- Ballerine.sb2

Objectif

Reproduire et modifier un programme pour créer une animation.



1 Description

La danseuse attend que tu la programmes pour se mettre à danser. Tu peux commencer par t'aider du projet n°1 « Faire_voler » pour faire une boucle qui affiche les différents costumes de la ballerine.

Ensuite, tu peux choisir de n'afficher que certains costumes en utilisant le bloc « **basculer sur le costume...** » de la palette « **Apparence** » (n'oublie pas de cliquer sur le petit triangle noir pour choisir les costumes). Ce bloc peut être utilisé une fois pour chaque costume que tu souhaites faire apparaître (donc plusieurs fois).

Tu pourras donc agir sur les différents costumes de la ballerine.

2 Ce qu'il faut savoir

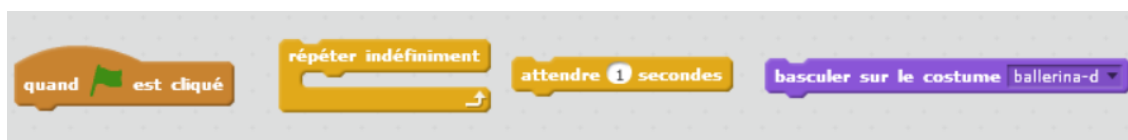
Pour connaître tous les costumes de la ballerine, tu peux cliquer sur l'onglet « **costumes** ».

En cliquant sur le drapeau vert, le programme est lancé. En cliquant sur le panneau rouge, le programme est arrêté.

3 Palettes à utiliser



4 Blocs utiles



Blocs utiles projet n°2

↳ Résultat



Résultat projet n°2

💡 Astuce

Ne pas oublier d'insérer au moins un temps d'attente après le changement de chaque costume pour que l'on puisse avoir le temps de les voir. Le temps d'une seconde est trop long, tu peux le raccourcir comme dans le programme « Faire_voler ». Il est également possible de varier le temps d'attente après chaque costume pour laisser la ballerine un peu plus longtemps sur certaines poses.

📁 Pour aller plus loin ...

Il est possible d'insérer de la musique à partir de la palette **Sons** « The_time_to_run ».

⚠ Attention

Dans Scratch 0,3 s'écrit avec un point (0.3)

📁 Question ?

Doit-on insérer un son avant la boucle ou bien à l'intérieur de la boucle ?

Tu peux rajouter d'autres sons plus brefs dans la boucle par la suite (par exemple le son « pop » après le costume « ballerina-d »).



Le morceau de musique « The time to run » composé par Dexter BRITAIN est publié sous licence libre BY NC SA

2.3. Projet n°3

Difficulté



Nom du projet

Voiture

Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
03- Voiture.sb2

Objectif

Créer un programme pour animer une image.



1 Description

Programmer la voiture pour qu'elle se déplace sur la route et qu'elle reparte en sens inverse lorsqu'elle touche le bord de l'écran. Tu pourras faire varier la vitesse de la voiture et faire évoluer ton programme pour introduire des pauses en appuyant sur la touche « **espace** ».

2 Ce qu'il faut savoir

Pour réaliser ce programme, il faut utiliser le bloc **avancer de 10**. Ce nombre peut être changé. C'est lui qui permet de régler la vitesse de déplacement de la voiture (essaye de choisir d'autres nombres).

Lorsque la voiture touche le bord, elle s'arrête. Pour la faire repartir dans l'autre sens, il faut utiliser le bloc **rebondir si le bord est atteint** depuis le menu **Mouvement**.

3 Palettes à utiliser

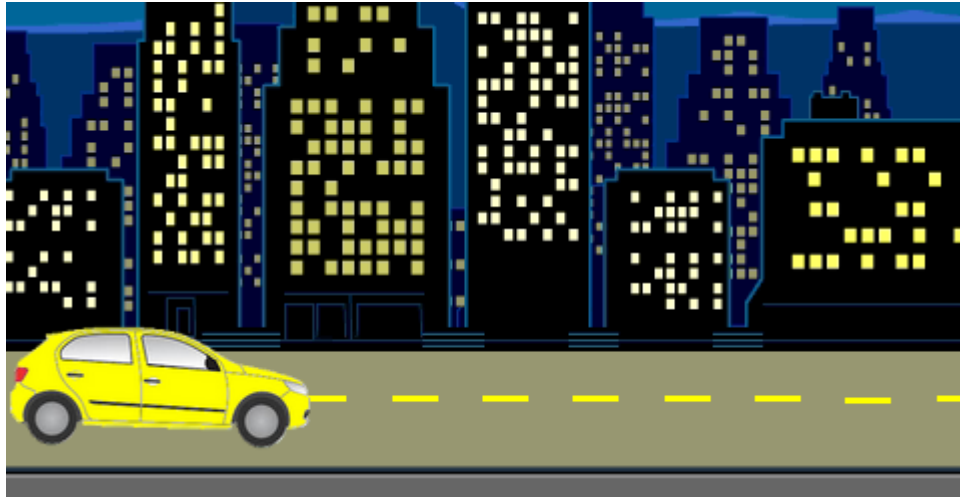


4 Blocs utiles



Blocs utiles projet n°3

↳ Résultat



Résultat projet n°3

💡 Astuce



voiture

x: -163 y: -83 direction: 90°

style de rotation:

peut glisser dans le lecteur

montrer: ☒

En cliquant sur le « i » du lutin « **voiture** » on fait

apparaître un menu qui permet d'empêcher le renversement de la voiture lorsqu'elle touche le bord (cliquer sur la double flèche).

📁 Pour aller plus loin ...

Modifier la boucle pour que la voiture s'arrête lorsqu'on appuie sur la barre **espace**. Pour changer ainsi le programme, il faut changer le bloc utilisé pour la boucle. Choisir le bloc

depuis la palette et glisser le bloc dans la zone vide depuis la palette .

2.4. Projet n°4

🗨 Difficulté



🗨 Nom du projet

Devinette

🗨 Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
04- Devinette.sb2

Objectif

Créer un programme pour présenter une devinette.



1 Description

Faire parler un personnage qui présentera la devinette suivante :

Sauras-tu trouver la réponse de cette devinette ? (phrase affichée pendant 3 secondes)

Il ne fait pas de bruit lorsqu'il se lève, mais il réveille tout le monde. (phrase affichée pendant 4 secondes)

Qui est-ce ? (phrases affichées pendant 2 secondes) - Temps d'attente supplémentaire de 2 secondes.

Si tu ne trouves pas, clique sur le bouton vert ! (phrase affichée pendant 2 secondes)

La réponse sera affichée en cliquant sur le bouton vert  (**C'est le soleil**, phrase affichée pendant 3 secondes)

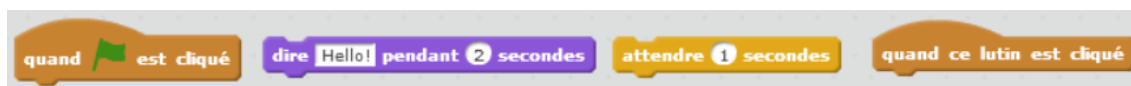
2 Ce qu'il faut savoir

Dans la palette **Apparence**, prendre le bloc **dire Hello! pendant 2 secondes**. Une fois que le bloc est déposé, en cliquant dans les zones blanches, il est possible de changer ce que le lutin dit et le temps d'affichage de ce qui est dit. Le même bloc peut être utilisé plusieurs fois.

3 Palettes à utiliser



4 Blocs utiles



Blocs utiles projet n°4

↳ Résultat



Résultat projet n°4

💡 Astuce

Le temps d'affichage dépend de la longueur des phrases. Plus la phrase est longue, plus il faudra de temps pour la lire. Entraîne-toi à mesurer combien de temps il faut pour lire les phrases affichées.

📁 Problème à résoudre ...

Comment cacher le bouton vert pendant que la devinette est posée ? (Cela éviterait de donner la réponse avant que la devinette n'ait été entièrement posée). Pour ce faire, il faudra utiliser les

blocs **envoyer à tous** `message1` et **quand je reçois** `message1` dans la palette **Événements**, et les blocs **cacher** et **montrer** dans la palette **Apparence**.

📁 Pour aller plus loin ...

Tu peux enrichir les programmes créés :

- en faisant varier les costumes du lutin en fonction des phrases affichées
- en faisant apparaître une image de la réponse à la fin (dans l'onglet « Costumes » tu peux ajouter le costume « **Sun** » et utiliser le bloc **basculer sur le costume** `sun` de la palette

Apparence)

2.5. Projet n°5

Difficulté



Nom du projet

Raconter une histoire

Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
05- Histoire.sb2

Objectif

Créer un programme pour présenter le dialogue de deux personnages.



1 Description

Faire dialoguer deux personnages qui raconteront l'histoire suivante :

- Le Prince : **Chère Princesse, j'aimerais vous emmener en voyage.** (phrase affichée pendant 3 secondes)
- La princesse : **Quelle bonne idée ! Où allons-nous ?** (phrase affichée pendant 3 secondes)
- Le Prince : **Nous allons visiter un château en Écosse.**
- La Princesse : **Je trouve qu'il fait froid en Écosse.**
- Le Prince : **Préférez-vous aller dans une île ?**
- La Princesse : **Non, il fera trop chaud.**
- Le Prince : **Et si nous allions dans cette université pour apprendre à programmer avec Scratch ?**
- La Princesse : **Quelle bonne idée ! J'aime tellement Scratch !** (La photographie représente le campus de l'université du MIT Massachusetts USA)

2 Ce qu'il faut savoir

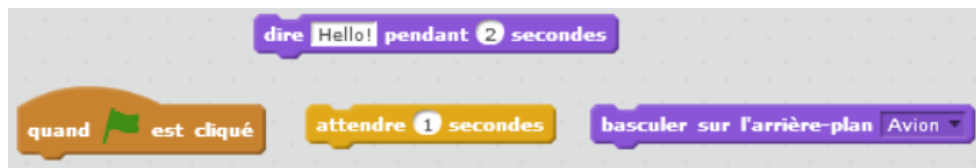
Pour éviter que les personnages ne parlent en même temps, il va falloir insérer des temps d'attente pour arriver à afficher ce qu'ils disent au moment voulu. N'hésite pas à faire plusieurs essais pour arriver résultat recherché, mais il est plus rapide de faire un calcul...

Avant que le Prince ne propose une nouvelle destination (et donc un nouvel arrière-plan, que l'on appelle scène dans « Scratch »), il faut l'insérer avec le bloc  en sélectionnant l'arrière-plan désiré (avec le triangle noir).

3 Palettes à utiliser



4 Blocs utiles



Blocs utiles projet n°5

➡ Résultat



Résultat projet n°5

📁 Pour aller plus loin ...

Tu peux modifier cette histoire en insérant d'autres arrière-plans et un autre personnage (par exemple le pilote de l'avion à la fin de l'histoire avec la commande « clic droit + montrer). Si tu l'utilises, n'oublie pas de remettre l'arrière-plan de l'avion...

2.6. Projet n°6

🗨 Difficulté



🗨 Nom du projet

Arc-en-ciel

Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
06- Arc-en-ciel.sb2

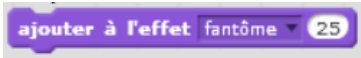
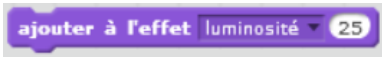
Objectif

Programmer différents effets dans des illustrations et savoir les faire varier en changeant leurs paramètres.



1 Description

Trois lutins sont proposés : un **nuage**, un **soleil** et un **arc-en-ciel**. Nous allons modifier l'apparence de chacun d'eux à l'aide d'un effet :

- Le nuage va disparaître progressivement (à l'aide de l'effet ).
- Le soleil, un peu terne, va devenir plus brillant (à l'aide de l'effet ).
- L'arc-en-ciel va scintiller en faisant varier toutes ses couleurs (à l'aide de l'effet ).

2 Ce qu'il faut savoir

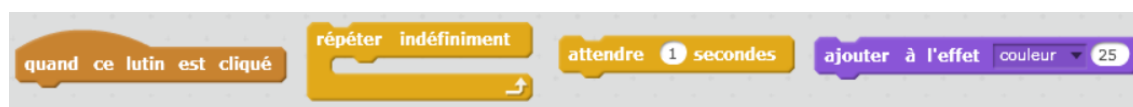
Avant de commencer un programme, il faut s'assurer d'avoir sélectionné le lutin concerné.

Pour pouvoir voir les effets programmés, il est souvent nécessaire d'ajouter un temps d'attente en utilisant le bloc . Ce temps doit souvent être plus petit. Pour cela, il faut utiliser un nombre décimal. Scratch utilise un point à la place de la virgule. Par exemple : « 0,5 » s'écrit « 0.5 ».

3 Palettes à utiliser



4 Blocs utiles



Blocs utiles projet n°6

↳ Résultat



Résultat projet n°6

💡 Astuce

Le nombre indiqué dans le bloc **ajouter à l'effet** couleur 25 représente la puissance de l'effet. Plus il sera grand, plus l'effet sera prononcé. Il faudra souvent qu'il soit faible pour qu'il puisse être visible progressivement.

📁 Pour aller plus loin ...

Il y a d'autres effets que tu peux explorer. Tu peux essayer de remplacer les lutins présents par d'autres lutins auxquels tu appliqueras les autres effets disponibles.

2.7. Projet n°7

💬 Difficulté



💬 Nom du projet

Sortir du château

💬 Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
07- Sortir_du_château.sb2

Objectif

Savoir changer l'arrière-plan (appelé « scène » dans Scratch) après avoir cliqué sur un lutin.



1 Description

Dans ce projet, plusieurs programmes sont déjà faits :

- Le programme du personnage qui explique la situation (il ressemble à celui du projet n°4).
- Les programmes qui font bouger la chauve-souris (ce sont les mêmes que ceux des projets n°1 et n°3).

Le but du projet est de programmer la chauve-souris pour qu'elle disparaisse lorsqu'on clique sur elle. Sa disparition doit entraîner l'ouverture de la porte du château.

2 Ce qu'il faut savoir

Il y a plusieurs façons d'atteindre un objectif. Par exemple, il y a plusieurs blocs qui permettent de faire disparaître un lutin. Il est donc possible de créer des programmes différents qui conviendront parfaitement. Les blocs proposés ci-dessous ne seront donc pas tous nécessaires, il faudra faire des choix selon l'effet que l'on recherche : soit faire disparaître d'un seul coup la chauve-souris, soit la faire disparaître progressivement.

3 Palettes à utiliser



4 Blocs utiles



Blocs utiles projet n°7

➡ Résultat



Résultat projet n°7

💡 Astuce

Pour ouvrir la porte du château, il faut changer l'image d'arrière-plan et montrer une image sur laquelle la porte a été enlevée. Une image de ce genre a déjà insérée dans les scènes. Pour afficher les scènes disponibles, il faut cliquer sur le bouton « **Scène** » puis sur l'onglet « **Arrière-plans** ». Il est même possible de modifier les scènes avec l'outil de dessin de Scratch.

📁 Pour aller plus loin ...

Après avoir fait disparaître la chauve-souris d'un seul coup, essaye de la faire disparaître progressivement en utilisant une boucle limitée et le bloc **ajouter à l'effet fantôme** 25 .

📁 Question ?

Additionne les nombres utilisés pour indiquer la force de l'effet fantôme.
Que se passe-t-il lorsque le total atteint ou dépasse 100 ?

2.8. Projet n°8

💬 Difficulté



💬 Nom du projet

L'armoire

Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
08- Armoire.sb2

Objectif

Programmer le déplacement d'objets pour habiller un personnage.



1 Description

Dany a besoin de ton aide pour choisir des vêtements dans l'armoire et s'habiller. Tu vas donc programmer les vêtements qui viendront se placer sur Dany lorsqu'on cliquera sur le bloc

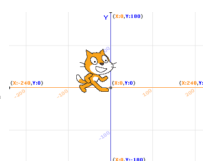


. On choisira la lettre « a » pour la tenue « a », la lettre « b » pour la tenue « b » et ainsi de suite...

2 Ce qu'il faut savoir

Les lutins ont des positions qui peuvent être définies par des coordonnées que l'on appelle x et y. La position au centre correspond à $x = 0$ et $y = 0$.

D'autres positions sont indiquées sur le schéma ci-contre.

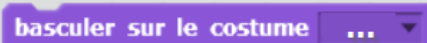


Repère-les : $x = -240$, $y = 0$ $x = 240$, $y = 0$ $x = -240$, $y = -180$ $x = 240$, $y = 180$

Lorsqu'on place un lutin à un endroit donné, si on clique sur le « i » du lutin (ou bien clic droit + info) on découvre qu'elle est sa position « x » et « y ».

Dans ce projet, Dany fait partie de l'arrière-plan pour qu'il ne soit pas déplacé. Les lutins ne seront que les habits qui vont le recouvrir.

Chaque lutin dispose d'un ou plusieurs costumes. Si tu prends le temps de les regarder, tu auras davantage de possibilités pour habiller Dany en utilisant le bloc



3 Palettes à utiliser



4 Blocs utiles



Blocs utiles projet n°8

↳ Résultat



Résultat projet n°8

💡 Astuce

Tous les habits présentés dans l'armoire peuvent être rangés immédiatement en cliquant sur la barre **espace**. Un programme a déjà été placé sur chaque lutin pour réaliser cette action.

📁 Pour aller plus loin ...

Les costumes des personnages peuvent être modifiés à l'aide des outils de Scratch. Affiche le lutin, puis clique sur l'onglet « **Costumes** », il est alors possible de changer les couleurs à l'aide du pot de peinture après avoir sélectionné la couleur désirée.

2.9. Projet n°9

💬 Difficulté



💬 Nom du projet

Pommes

💬 Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
09- Pommes.sb2

Objectif

Programmer un lutin dans un jeu pour qu'il puisse se déplacer.



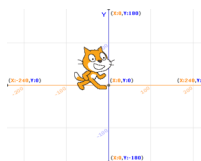
1 Description

C'est l'automne, les pommes tombent des arbres. Isaac Newton aimerait éviter de recevoir une pomme sur la tête, mais il ne peut pas se déplacer. Saurais-tu le programmer pour qu'il puisse se déplacer vers la droite et vers la gauche, en utilisant les flèches de ton clavier ?

2 Ce qu'il faut savoir

Les lutins ont des positions qui peuvent être définies par des coordonnées que l'on appelle x et y. La position au centre correspond à $x = 0$ et $y = 0$.

D'autres positions sont indiquées sur le schéma ci-contre.



Repère-les : $x = -240, y = 0$ $x = 240, y = 0$ $x = -240, y = -180$ $x = 240, y = 180$

Pour se déplacer vers la droite et vers la gauche, il faudra changer la valeur de x, soit pour l'augmenter, soit pour la diminuer.

3 Palettes à utiliser



4 Blocs utiles



Blocs utiles projet n°9

Dans le premier bloc, clique sur le triangle noir pour sélectionner la « flèche gauche » ou la « flèche droite ». Dans le deuxième bloc, pour changer la valeur de x, soit on ajoutera « 10 » pour le faire grandir et déplacer le lutin vers la droite, soit on indiquera « -10 » pour le faire diminuer et déplacer le lutin vers la gauche.

➡ Résultat



Résultat projet n°9

💡 Astuce

N'oublie pas d'insérer le bloc  pour éviter que Newton ne disparaisse de l'écran !

📁 Pour aller plus loin ...

Comment accélérer les programmes pour rendre le jeu plus difficile ?

Sais-tu qu'Isaac Newton a réellement existé et qu'il a fait une importante découverte scientifique en recevant une pomme sur la tête ?

2.10. Projet n°10

💬 Difficulté



💬 Nom du projet

Chat et souris

💬 Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :

10- Chat_et_souris.sb2

Objectif

Programmer un lutin dans un jeu pour pouvoir gagner.



1 Description

La souris aimerait bien manger le fromage, mais elle ne peut se déplacer que vers la droite et vers la gauche.

Comment la programmer pour qu'elle puisse monter vers le fromage ?

2 Ce qu'il faut savoir

C'est en changeant les valeurs de x et y que l'on déplace la souris.

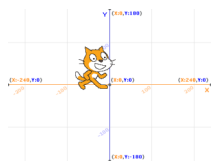
En changeant les valeurs de x, on dirige la souris vers la droite ou vers la gauche. En changeant les valeurs de y, on fait monter ou descendre la souris.

Au début du jeu :

- la souris se trouve en $x=0$, $y=-180$

- le fromage se trouve en $x=0$ et $y=180$.

Il te faudra compléter le programme pour que lorsqu'on appuie sur la flèche « **vers le haut** » la valeur de y grandisse (en lui ajoutant « 5 » par exemple).



3 Palettes à utiliser

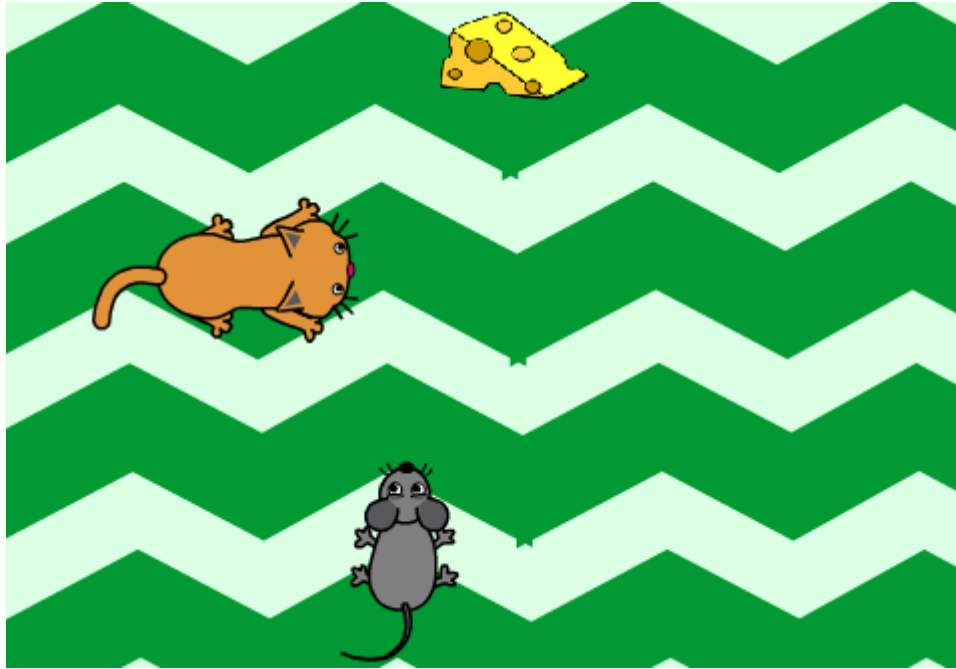


4 Blocs utiles

Pour réaliser ce projet, on utilisera les blocs habituels.

Les blocs  et  permettent d'insérer une instruction qui

posera une condition. Dans ce cas, si le lutin touche une couleur connue (le jaune représente le fromage et l'orange représente le chat) alors l'ordre suivant est respecté (soit c'est gagné, soit c'est perdu). Il est nécessaire de conserver ces instructions dans le bloc que tu vas ajouter pour que le jeu puisse fonctionner. Si tu les oublies, tu verras que le jeu ne fonctionne plus.

 Résultat*Résultat projet n°10* Astuce

Il est possible de dupliquer des portions de programmes en faisant un clic droit et en choisissant la commande « dupliquer ». Ensuite, il ne reste qu'à modifier le code qui a été copié pour programmer la flèche « vers le haut » et faire monter la souris en appuyant sur cette touche.

 Pour aller plus loin ...

Comment changer le programme du chat pour rendre le jeu plus facile ou plus difficile ?

Saurais-tu programmer la souris pour qu'elle puisse aussi descendre avec la flèche « **vers le bas** » ?

2.11. Projet n°11

 Difficulté Nom du projet

Tracer un carré

 Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :

11- Tracer_carre.sb2

Objectif

Programmer le tracé d'une figure géométrique.



1 Description

Pour représenter la pointe d'un stylo, on a rétréci un disque qui est devenu un point et on l'a colorié en rose. C'est ce point qui tracera les formes que l'on programmera. On utilisera des blocs qui se trouvent dans la palette **Stylo** pour définir la couleur et l'épaisseur du trait, puis pour commander au stylo de se placer en position d'écriture. La position de départ du stylo sera ($x = -120$ et $y = -60$).

2 Ce qu'il faut savoir

Pour commencer le tracé du carré, on utilisera le bloc **s'orienter à 90** pour avancer dans la direction voulue. En choisissant la valeur 0, on orientera le tracé vers le haut.

Il est utile d'insérer des temps d'attente pour éviter que le tracé soit réalisé trop rapidement. Ainsi on pourra mieux voir chaque étape du tracé

Rappels : Un carré a 4 côtés identiques et 4 angles droits. Un angle droit mesure 90° . L'angle droit est toujours orienté vers la droite. La surface disponible pour le tracé mesure 480 pixels de longueur sur 360 pixels de largeur.

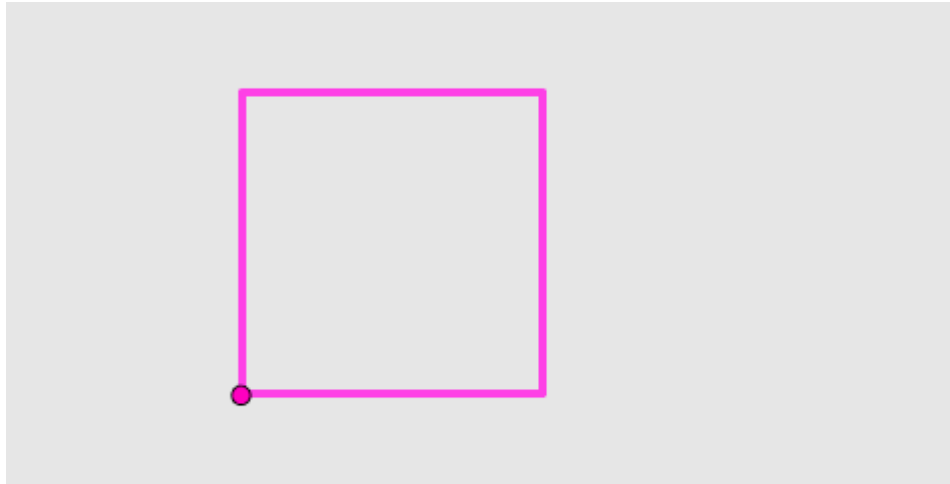
3 Palettes à utiliser



4 Blocs utiles



Blocs utiles projet n°11

 Résultat*Résultat projet n°11* **Astuce**

Pour permettre de recommencer le tracé du carré, on a ajouté un petit programme qui efface tout et replace la pointe du crayon à son point de départ lorsqu'on appuie sur la barre d'espace. Ceci te permettra d'observer les modifications de ton programme si tu choisis de changer l'épaisseur du trait ou la couleur du crayon...

 Pour aller plus loin ...

Après avoir programmé le tracé d'un carré, tu peux modifier ton programme pour tracer un rectangle. Il faudra prendre en compte la largeur et la hauteur de la surface disponible. Une fois que tu sais programmer le tracé d'un carré et d'un rectangle, tu peux essayer de programmer le tracé d'autres formes. Par exemple pour tracer un cercle le crayon doit faire un tour complet sur 360 °...

2.12. Projet n°12

 Difficulté **Nom du projet**

Labyrinthe1

 Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
12- Labyrinthe1.sb2

 Objectif

Programmer les déplacements du chat Scratch pour qu'il puisse sortir d'un labyrinthe.



1 Description

Scratch se trouve à l'entrée du labyrinthe. Il veut sortir du labyrinthe mais pour l'instant, il ne peut que se déplacer vers le haut.

Sauras-tu programmer ses déplacements avec les 4 **flèches** de ton clavier ?

2 Ce qu'il faut savoir

Pour que Scratch se déplace réellement dans le labyrinthe, il faut l'empêcher de passer au travers des murs. Il faudra donc lui interdire les mouvements qui lui font toucher les murs. Regarde le code utilisé pour programmer la touche « **flèche vers le haut** », lorsque la couleur noire est touchée (les murs du labyrinthe sont noirs), on demande à Scratch de respecter l'ordre inverse pour qu'il reste à la même place. On a alors l'impression qu'il est arrêté par le mur. Pour que Scratch puisse se déplacer, il faut que les valeurs de x et de y changent lorsqu'on appuie sur les flèches (comme dans les projets précédents). Pour faire grandir x ou y, on ajoute un nombre à x ou y. Pour diminuer x ou y, on ajoute un nombre négatif (c'est un nombre avec le signe -). Par exemple, pour enlever 3 on ajoute -3.

3 Palettes à utiliser



4 Blocs utiles

Pour réaliser ce projet, on utilisera les blocs suivants qui sont présentés entre crochets :

[si ---- alors] [couleur ☐ touchée ?] [ajouter ---- à x] [ajouter ---- à y]

➡ Résultat



Résultat projet n° 12

Astuce

Pour écrire le programme, tu n'es pas obligé de glisser les blocs un par un. Il est possible de dupliquer des portions de programmes en faisant un clic droit et en choisissant la commande « dupliquer ». Ensuite, il ne reste qu'à modifier certains blocs et à supprimer ceux qui sont inutiles.

Dans ce projet, tout peut être réalisé en un seul programme, il n'est pas utile de multiplier les programmes et de t'éparpiller. Il faudra quand même faire attention aux imbrications des conditions...

Pour aller plus loin ...

Tu peux insérer des éléments qui seront programmés comme étant des pièges pour Scratch. Lorsqu'il les touchera, il reviendra au point de départ. Pour obtenir ce résultat, tu peux utiliser le bloc [aller à x : ---- y : ----]. Bien entendu, tu laissera un chemin pour sortir du labyrinthe.

2.13. Projet n° 13

Difficulté



Nom du projet

Labyrinthe 2

Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
13- Labyrinthe2.sb2

Objectif

Programmer les déplacements du chat Scratch pour qu'il puisse sortir d'un labyrinthe.



1 Description

Scratch se retrouve dans le même labyrinthe, mais cette fois une chauve-souris l'empêche de passer. Si Scratch touche la chauve-souris la partie est perdue.

Commence par vérifier que Scratch ne peut pas passer. C'est fait ?

Ta mission consiste à programmer Scratch pour qu'il puisse passer par un endroit qui lui permette d'éviter de toucher la chauve-souris. Par exemple, on pourrait imaginer que Scratch passe par un tunnel qui le rapprocherait de la sortie...

2 Ce qu'il faut savoir

L'entrée du tunnel est représentée par un rond gris cerclé de violet. La sortie du tunnel est représentée par un rond gris cerclé de rouge.

Dans cette version du jeu, c'est le programme de Scratch qu'il va falloir modifier. Lorsque Scratch va toucher l'entrée du tunnel, il sera envoyé à l'emplacement occupé par la sortie du tunnel. Avant

de modifier le programme, tu dois repérer les coordonnées x et y de la sortie du tunnel.

3 Palettes à utiliser

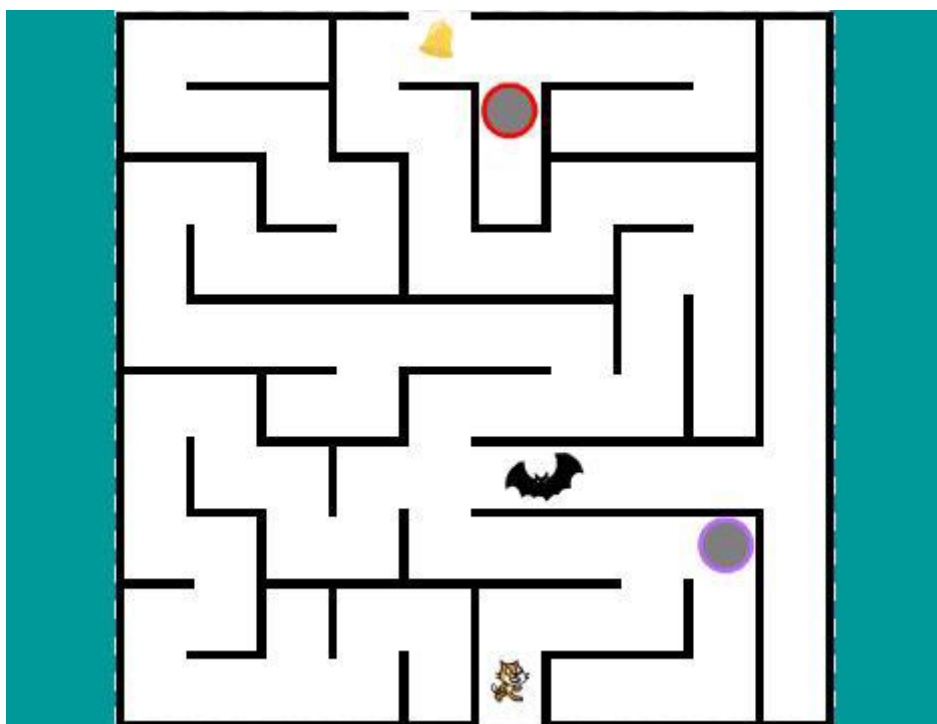


4 Blocs utiles

Pour réaliser ce projet, on utilisera les blocs suivants qui sont présentés entre crochets :

[si ---- alors] [----▼ touché ?] [aller à x : ---- y : ----]

↳ Résultat



Résultat projet n°13

💡 Astuce

Pour repérer les coordonnées x et y de la sortie du tunnel, tu peux déplacer le pointeur de la souris à l'endroit désiré (x et y sont affichés sous le labyrinthe à droite).

Tu peux aussi afficher les coordonnées de la sortie du tunnel en cliquant sur le « i » du lutin.



Le bloc **aller au premier plan** a été ajouté dans les programmes du chat et de la chauve-souris pour qu'ils soient sur le même plan et puissent se toucher, sinon le chat pourrait traverser la chauve-souris sans perdre.

📁 Pour aller plus loin ...

Tu peux inventer d'autres histoires et trouver d'autres solutions pour permettre à Scratch de sortir du labyrinthe. Par exemple, il pourrait toucher un bouton qui ralentirait beaucoup le déplacement de la chauve-souris. Ou bien, il pourrait y avoir des portes secrètes dans le labyrinthe (certaines parties des murs ne seraient pas vraiment noires) etc. Partage tes idées avec tes camarades...

2.14. Projet n° 14

Difficulté



Nom du projet

Fusée

Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
14- Fusée.sb2

Objectif

Programmer des objets pour simuler le décollage d'une fusée.



1 Description

Tu es le chef de la nouvelle mission spatiale et ta mission consiste à programmer le lutin « Fusée » pour déclencher son décollage lorsqu'on appuie sur le bouton vert.

2 Ce qu'il faut savoir

La fusée commence par décoller lentement, elle gagne de la vitesse au fur et à mesure qu'elle avance. Il faudra donc envisager d'utiliser plusieurs boucles (par le bloc ) dans lesquelles la fusée avancera à des vitesses différentes, lentes au début puis de plus en plus rapide.

Rappel : L'effet de vitesse est provoqué en indiquant des distances de plus en plus grandes dans le bloc .

Il y a de nombreuses façons de réaliser cette mission. Il est donc intéressant de regarder comment tes camarades ont conçu leur programme...

3 Palettes à utiliser



4 Blocs utiles

Le bouton vert est déjà programmé pour envoyer le message « **Décollage** » au lutin « **Fusée** » lorsqu'on clique dessus.

Dans le programme de la fusée, le bloc [quand je reçois Décollage ▼] déclenche le programme qui suit (lorsque le bouton vert est cliqué). C'est lui qui portera les instructions pour la fusée.

Les autres blocs utilisés sont :

la boucle [répéter ---- fois] [avancer de ----] [costume suivant] [attendre ---- secondes] [cacher]

↳ Résultat



Résultat projet n° 14

💡 Astuce

Pour simuler le crépitement du feu des réacteurs de la fusée, il est possible d'afficher à tour de rôle les 2 costumes de la fusée et en insérant un temps d'attente très court entre les deux.

Rappel : un temps plus court qu'une seconde commence par un zéro suivi d'un point [0.]

Le bloc [cacher] permet d'éviter qu'un petit bout de fusée reste affiché en haut de l'écran !

📁 Pour aller plus loin ...

Tu peux imaginer plusieurs améliorations du programme.

Par exemple, il est possible d'insérer un compte à rebours avant le décollage (soit avec des sons enregistrés, soit avec des chiffres qui s'affichent à l'écran). Tu peux aussi envisager d'autres animations sur la fusée...

2.15. Projet n° 15

💬 Difficulté



💬 Nom du projet

Aquarium

Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
15- Aquarium.sb2

Objectif

Modifier le programme d'un jeu pour le rendre plus intéressant.



1 Description

Dans ce jeu, le poisson est obligé de partager l'aquarium avec une méduse qui ne l'aime pas. Alors, il doit éviter de la toucher, sinon la partie est perdue. Pour gagner, il doit manger la nourriture pour poisson qu'on lui donne (**Miam1, Miam2 et Miam3**). Le poisson se déplace avec les **flèches** habituelles. La méduse est programmée pour se déplacer « au hasard ».

Si tu joues à ce jeu, tu vas vite découvrir qu'il est très facile de gagner la partie. Commence par jouer une partie en cliquant sur le drapeau vert, puis sur la barre **espace**.

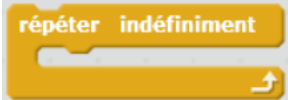
Ta partie est terminée ? C'était trop facile, n'est-ce pas ? Ta mission consiste à changer le programme pour que le jeu soit plus difficile, mais qu'il reste possible de gagner la partie.

2 Piste de travail proposée

Tu peux observer qu'un lutin a été masqué et n'a pas été utilisé jusqu'à maintenant, il s'agit de l'araignée de mer. Pour rendre le jeu plus difficile, tu peux programmer cette araignée pour que la partie soit perdue si le poisson la touche. L'araignée ne se déplacera pas, par contre tu peux la dupliquer 2 fois (une fois que son programme sera terminé) pour obtenir 3 araignées qui fonctionneront avec le même programme.

3 Palettes et blocs à utiliser

Les palettes et les blocs à utiliser sont ceux que tu utilises d'habitude, mais il y a 2 deux nouvelles palettes qui sont utilisées : la palette **Opérateurs** qui est utilisée dans le programme de la méduse et la palette **Données** qui est utilisée dans le programme des morceaux de nourriture. Normalement, tu ne devrais pas avoir besoin d'utiliser ces palettes une nouvelle fois. Par contre, il faudra que tu repères les blocs qui font perdre le poisson dans le programme de la méduse.

La boucle  est une instruction nécessaire. Si elle n'est pas utilisée, les instructions ne seront exécutées qu'une seule fois et le jeu ne fonctionnera pas.

➤ Résultat



Résultat projet n° 15

💡 Astuce

La disposition des araignées dans l'aquarium aura une grande importance sur la difficulté du jeu. Elle peut même rendre le jeu impossible à gagner, ce qui n'est pas le but recherché.

📁 Pour aller plus loin ...

Cherche d'autres idées pour améliorer ce jeu. Cela peut être l'ajout de nouveaux éléments, des sons qui guident le joueur. Il est également possible d'augmenter le nombre de morceaux de nourriture à manger, d'introduire une deuxième méduse, de changer les vitesses des déplacements, etc.

2.16. Projet n° 16

🗨 Difficulté



🗨 Nom du projet

Robot calculateur

🗨 Fichier à utiliser

Le fichier se trouve dans le dossier **Travail** de votre espace **se3** :
16- Robot_calculateur.sb2

Objectif

Modifier un programme pour l'adapter aux besoins rencontrés.



1 Description

Le robot calculateur peut vérifier toutes les additions de 2 nombres que tu lui indiques. Saurais-tu modifier son programme pour qu'il fasse des multiplications à la place des additions ?

Essaye de comprendre le fonctionnement du programme et lance-toi. Si tu fais des erreurs ce n'est pas grave, ce sera même nécessaire pour atteindre ton objectif. Le plus important c'est d'arriver à les corriger !

2 Ce qu'il faut savoir

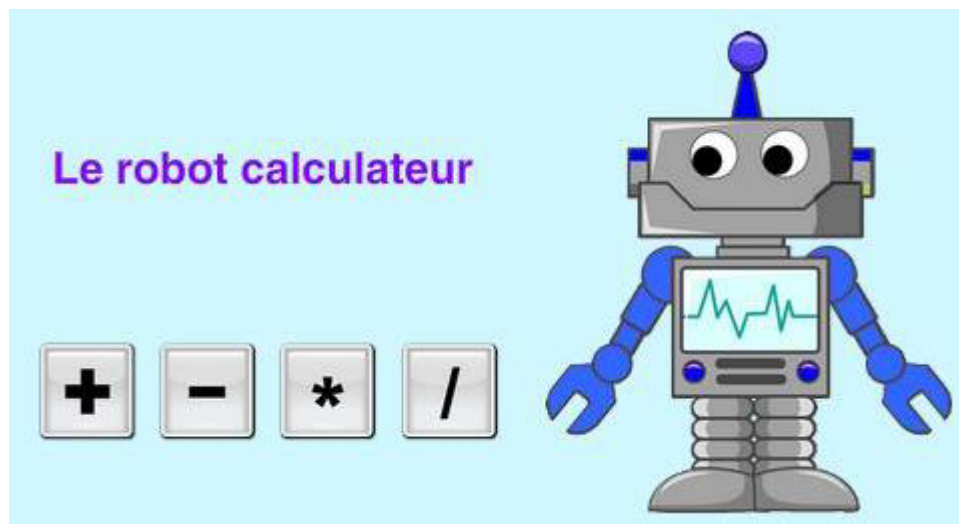
Tu commences à bien connaître Scratch. Dans ce projet, l'objectif est d'arriver à comprendre le fonctionnement du programme pour être capable le modifier.

Lis attentivement le programme et cherche à comprendre à quoi sert chaque bloc. Essaie de repérer les éléments qui devront être modifiés. Selon ton niveau, peut-être que ton enseignant acceptera de t'indiquer le nombre de blocs qui doivent être modifiés...

3 Palettes et blocs à utiliser

Tu commences à connaître les palettes et les blocs les plus courants de Scratch. Dans ce projet, les palettes **Données** et **Opérateurs** joueront encore une fois un rôle important. Par exemple, la palette **Opérateurs** contient les blocs qui permettent de faire des opérations sur les nombres. Attention, sur ton clavier et dans Scratch, le signe de la multiplication est *****.

Résultat



Résultat projet n°16

Astuce

Ce genre de travail va te demander un grand nombre d'essais. Tu risques donc de beaucoup changer le fichier d'origine. Pour pouvoir le retrouver, il est préférable de travailler sur une copie de

ce fichier en prenant soin de l'enregistrer sous un autre nom (en utilisant la commande « Sauvegarder comme copie » ou « Enregistrer sous » selon la version de Scratch utilisée). Par exemple, tu peux le renommer « Robot_calculateur_test1 » et ainsi de suite...

Pour aller plus loin ...

Relève ce nouveau défi :

Modifie ce programme pour que le robot donne le choix de l'opération qui sera faite : addition, soustraction, multiplication ou division. Les lutins qui représentent ces opérations sont déjà affichés. Ton enseignant peut te guider pour t'expliquer comment programmer l'addition. En comprenant cette étape, tu pourras terminer la programmation des autres opérations. Ce projet est un peu plus difficile que ceux que tu as faits jusqu'à présent. Pour le réussir, tu auras sans doute besoin d'une séance supplémentaire avec l'aide de ton enseignant. Les blocs nécessaires sont :



Quelques indices pour t'aider :

1°) « message 1 » sera remplacé par le nom d'une des 4 opérations.

2°) Le bloc  permet d'indiquer qu'avant d'envoyer le message qui déclenche l'opération, il faudra attendre que le « **Nombre 2** » soit enregistré (ce qui se produit lorsque le **Nombre 2** > 0).