

Algorithmes sur Scratch

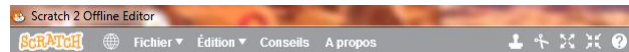
Objectif : comprendre ce qu'ont appris vos élèves de seconde au collège et partir de ce qu'ils savent pour construire des notions d'algorithmiques à partir de Python ou autres

1 Prise en main rapide

1.1 Prendre connaissance de l'environnement

prendre connaissance des différentes instructions pour comprendre l'environnement de travail :

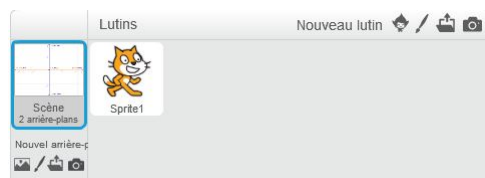
- les actions de fichiers (enregistrement, nouveau...), d'édition, de conseils...



- La fenêtres du déroulement des actions



- La scène et le(s) lutin(s)



conseil : choisissez des scènes (décors) qui donne du sens à vos activités, la page blanche est la plus fréquemment utilisée au début des manipulations.

- les différentes instructions classées par thèmes pour le(s) lutin(s)



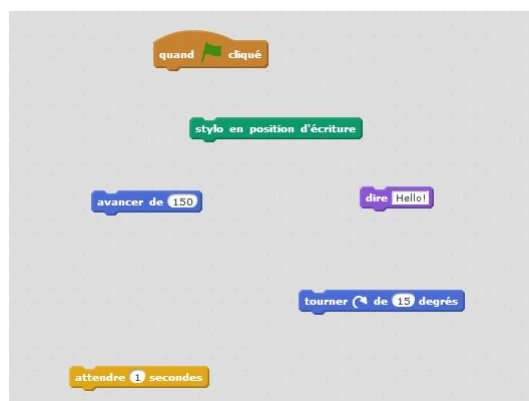
Remarques : il est possible de modifier les costumes des lutins, de modifier la scène et de gérer les sons. (voir les onglets à côté du script).

- la fenêtre où sont déposées les instructions relatives à l’animation du (des) lutins et de la scène.



1.2 Les premières intuitions

1. Retrouver les instructions de la fenêtre ci-dessous. Imbriquez-les de façon à créer une animation quelconque du lutin.



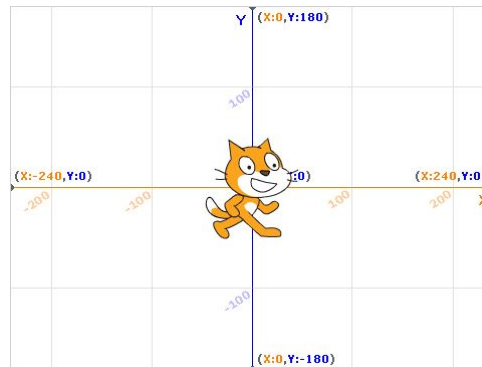
remarques :

Pour que le lutin exécute les instructions, il faut soit cliquer sur l’instruction ou le bloc d’instructions (à éviter), soit gérer des événements pour démarrer un bloc d’instructions (par exemple, la mise en place de l’événement du drapeau vert cliqué).

2. *Astuce* : Trouver une série d’au moins quatre instructions qui permettent de replacer le lutin à sa position initiale et d’effacer toutes les traces de ses déplacements (etc..). Ce bloc peut servir de ré-initialisation de la fenêtre des actions.

Remarque :

Le lutin se déplace dans un repère, ce repère peut être choisit comme scène :

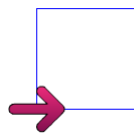


1.3 Les premières figures sur Scratch : découverte par contraintes

Réaliser les figures suivantes :

Note : pour bien repérer votre orientation, choisissez la flèche pour lutin

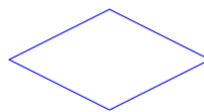
1. Construire un carré de côté 100 et faire revenir le lutin à sa position initiale.



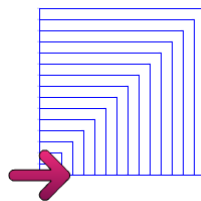
2. Construire un parallélogramme de côté 100 et 80 d'angle 60 degré :



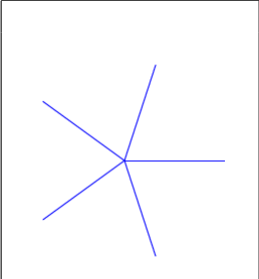
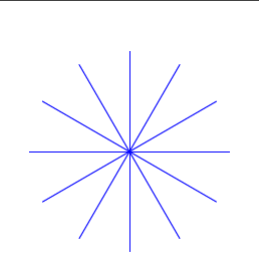
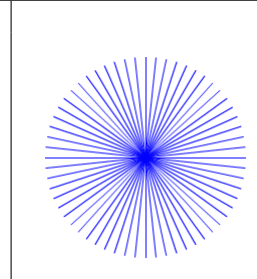
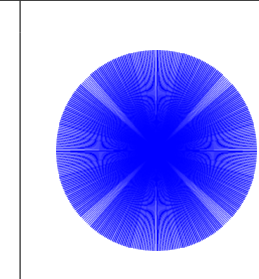
3. Construire un losange dont les longueurs des diagonales sont déterminées par l'utilisateur, et le losange sera centré au centre de l'image.



4. Améliorer l'algorithme du carré pour construire des carrés imbriqués.

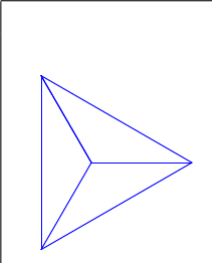
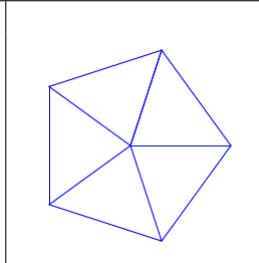
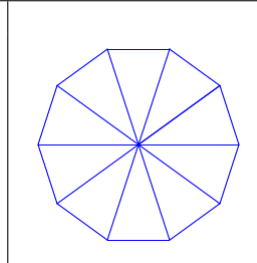
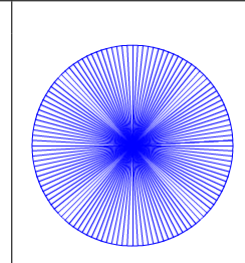


5. (a) Construire des suites de segments dont l'extrémité commune est le centre de l'image, la longueur est 100, et l'angle entre deux segments consécutifs est un entier diviseur de 360 degrés. Le premier segment est défini par les points de coordonnées (0 ; 0) et (100 ; 0).

5 segments	12 segments	60 segments	360 segments
			

- (b) Compléter l'algorithme précédent par :

- L'obtention d'une liste des points qui forment le polygone régulier défini par les extrémités des segments.
- Construire un algorithme dans la même feuille (une procédure ou fonction) pour tracer ce polygone :

3 côtés	5 côtés	10 côtés	120 côtés
			

2 Ce qui peut se faire au collège

2.1 Prolongement du sujet zéro

Algorithmiques : Procédures, boucles

1. Répondre aux questions du sujets zéro et testé avec Scratch.



- 1 – Pour réaliser la figure ci-dessus, on a défini un motif en forme de losange et on a utilisé l'un des deux programmes A et B ci-contre. Déterminer lequel et indiquer par une figure à main levée le résultat que l'on obtiendrait avec l'autre programme.

```

définir Motif
stylo en position d'écriture
avancer de 40
tourner ⤵ de 45 degrés
avancer de 40
tourner ⤵ de 135 degrés
avancer de 40
tourner ⤵ de 45 degrés
avancer de 40
tourner ⤵ de 135 degrés
relever le stylo
    
```

Programme A

```

quand est cliqué
cacher
effacer tout
mettre la taille du stylo 1
aller à x: -230 y: 0
s'orienter à 90 degrés
répéter 8 fois
Motif
avancer de 65
    
```

Programme B

```

quand espace est pressé
cacher
effacer tout
mettre la taille du stylo 1
aller à x: 0 y: 0
s'orienter à 90 degrés
répéter 8 fois
Motif
tourner ⤵ de 45 degrés
    
```

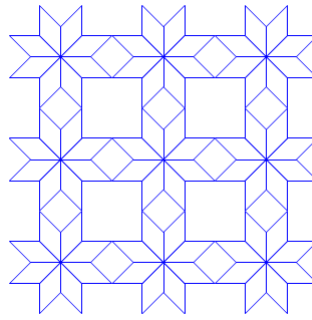
- 2 – Combien mesure l'espace entre deux motifs successifs ?

- 3 – On souhaite réaliser la figure ci-dessous :



Pour ce faire, on envisage d'insérer l'instruction `ajouter 1 à la taille du stylo` dans le programme utilisé à la question 1. Où faut-il insérer cette instruction ?

2. Prolonger le sujet zero à partir de l’algorithme B pour réaliser ce pavage :



Remarque : Une figure sur GeoGebra permet de comprendre le pavage.

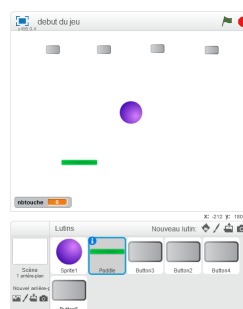
2.2 Jeu de pong : programmation événementielle

Événements, boucles, variables

On donne les instructions de deux lutins, le paddle et la balle.

<pre> quand est cliqué répéter indéfiniment avancer de 10 rebondir si le bord est atteint si couleur touchée? alors s'orienter à nombre aléatoire entre -90 et 90 si ordonnée y de Sprite1 < -150 alors aller à x: 0 y: 0 s'orienter à 0 stop tout </pre>	<pre> quand est cliqué répéter indéfiniment si touche flèche gauche pressée? alors avancer de 10 si touche flèche droite pressée? alors avancer de -10 </pre>

- Après avoir choisi vos lutins dans la bibliothèque, retrouver et placer les instructions des algorithmes associés à chaque lutin.
Note : le choix de la couleur se fait en cliquant sur une couleur n'importe où dans la fenêtre scratch
- Premier prolongement possible : compléter le code pour savoir combien de fois la balle a touché le paddle.
- Deuxième prolongement possible : en haut de la fenêtre placer des lutins "brique", chaque fois que la balle touche une brique elle disparaît de la fenêtre.



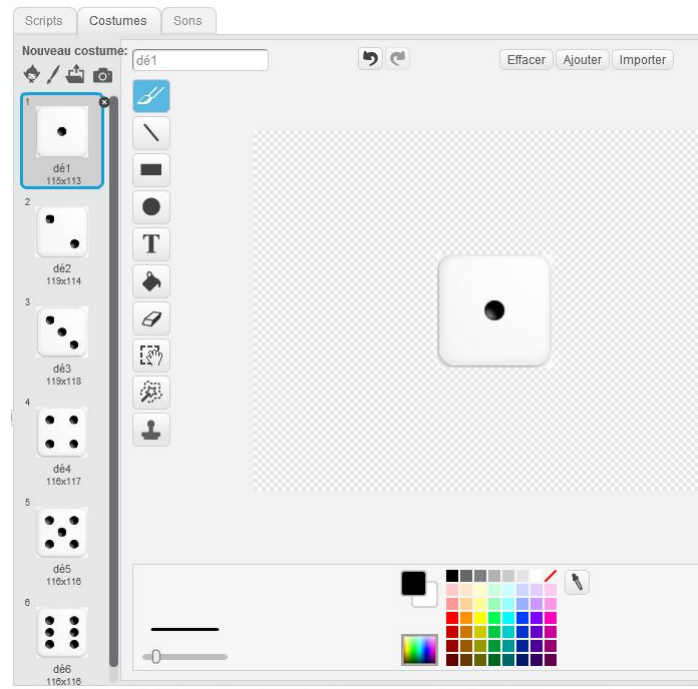
2.3 Lancer de deux dés

Algorithmiques : événements, conditions, boucles

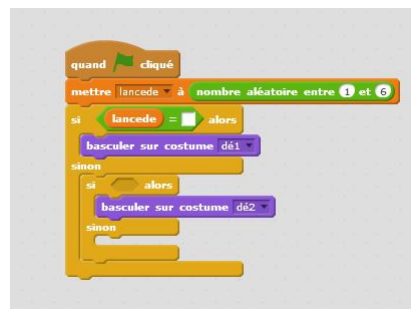
L'activité se décline en trois exercices, il s'agit de lancer un dé à six faces et de constater les probabilités, on peut alors prolonger l'exercice par le lancer de deux dés et chercher les sommes.

1. Le lancer d'un dé, conditions

- (a) Prévoir six costumes pour lancer le dé : récupérer six photos, les faces du dé.



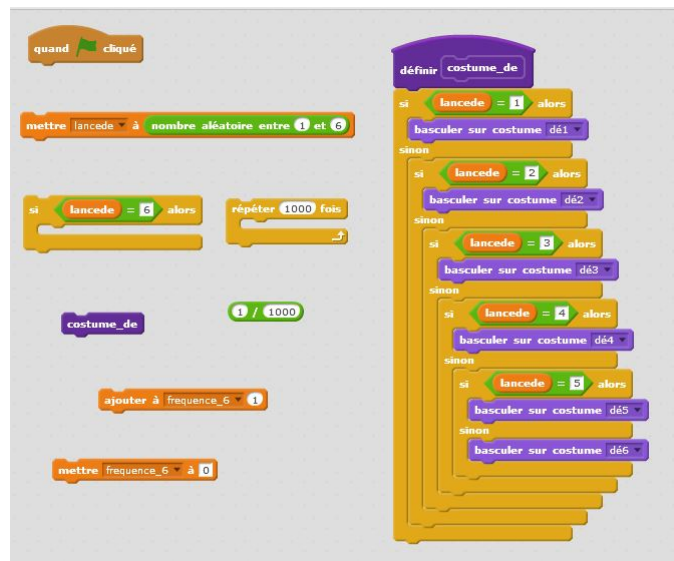
- (b) Recopier et compléter le script suivant pour obtenir le lancer d'un dé



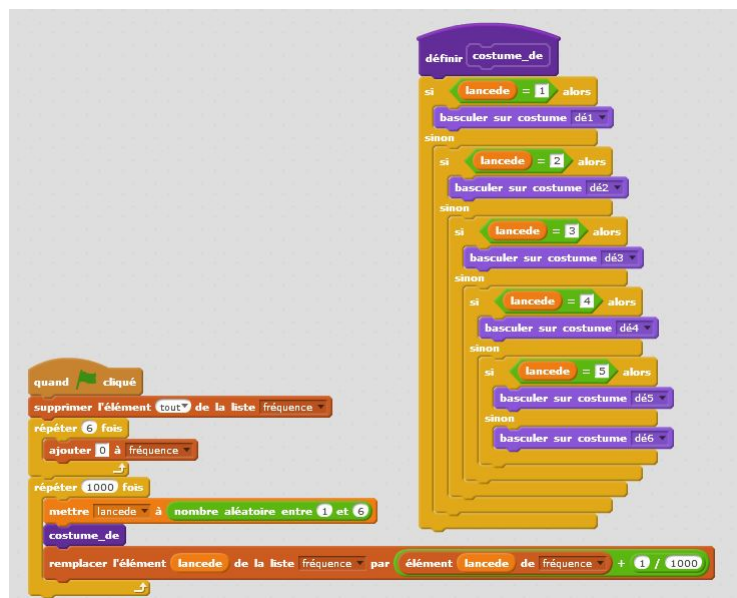
- (c) À l'aide d'une boucle, simplifier les instructions pour éviter les conditions imbriquées.

2. Simuler le lancer d'un dé 1000 fois, fréquences des résultats associés, gestions de listes

- (a) Pour faire la simulation de 1000 lancers d'un dé, modifier l'algorithme précédent et compléter le en remettant le script dans l'ordre (le bloc costume dé n'est pas à modifier, il est laissé à droite), on souhaite obtenir la fréquence du 6 :

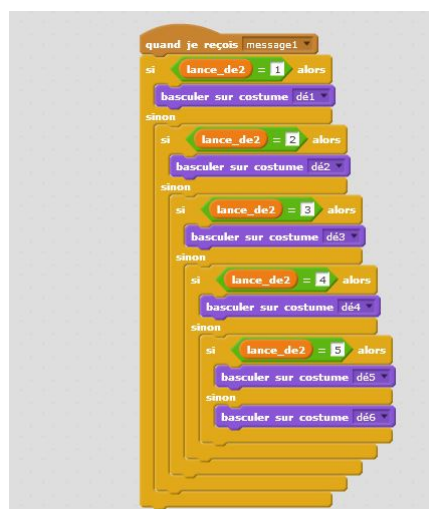
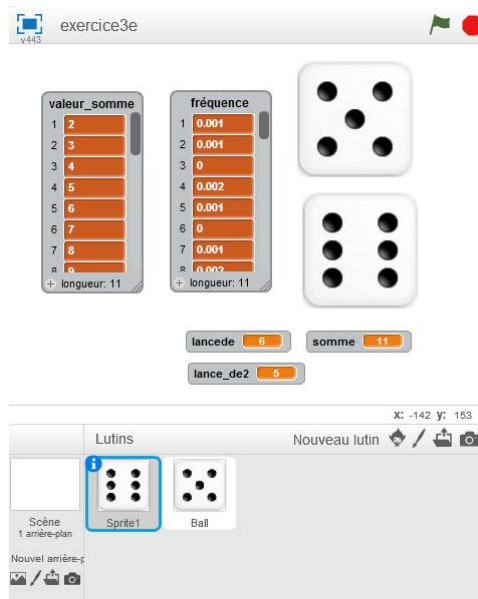


- (b) Recopier et compléter l'algorithme suivant pour simuler 1000 lancers d'un dé. La liste des fréquences donne la fréquence de chaque numéro :



Sur ce même principe, créer la simulation de 1000 lancers de deux dés, et donner les résultats avec la fréquence des sommes observées.

Correction, on utilise deux lutins, un script possible pour chacun :



(c) *Remarque* : il est possible d'exporter les listes sur tableau (clic droit sur la liste affichée dans la scène Scratch) et d'obtenir des diagrammes en bâtons des résultats.

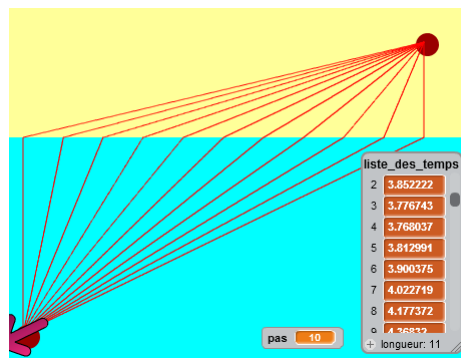
3 Pour aller plus loin, notion de temps, optimisation et simulation

Exemple de problème :

Soit un point A dans un milieu où la vitesse de déplacement est 0,005 seconde par unité et un point B dans un milieu B où la vitesse de déplacement est 0,01 seconde par unité.

Le point A a pour coordonnées $(180 ; 140)$ et le point B a pour coordonnées $(-220 ; -150)$, la droite d'équation $y = 45$ sépare les milieux.

1. Optimiser par balayage le parcours du lutin qui va de A vers B . On donnera le parcours que doit décrire le lutin, en arrondissant si besoin à l'unité.



2. Résoudre le problème (on pourra utiliser Géogébra ou Xcas pour appuyer l'illustration).