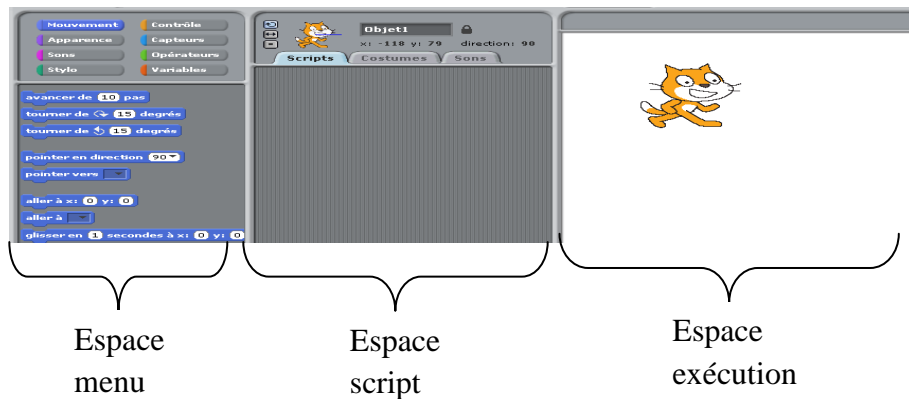


- 1°) Chez vous, vous devrez télécharger « Scratch » sur le site du lycée <http://www.lycee-la-cotiere.com>, puis onglets « enseignement disciplinaire » et enfin « Mathématiques » « 2nde ».
- 2°) Effectuer chacun des exercices de l'Étape 3, puis l'enregistrer dans le répertoire portant votre nom (sur le réseau) sous la forme Nom-TP1-Ex1. (Nom de lecteur : P :)
- 3°) Les exercices 3 à 5 devront être envoyés à l'adresse

Étape 1 : La base

Scratch est un logiciel qui permet, grâce à des briques, de construire des programmes mis en exécution sur des objets (ici notre objet est le chat).

L'écran est divisé en trois « espaces » : l'espace « menu » (où on « prend » les briques à emboîter pour faire le programme), l'espace « script » où on fait glisser les briques à utiliser) et enfin l'espace « exécution » (où notre « objet-chat » exécute le programme)



Remarque : dans l'espace script, il y a 3 onglets : « scripts », « costumes » et « sons » ; seul le premier nous servira.



Étape 2 : les menus

Menu Variable (en orange dans l'espace « menu »)

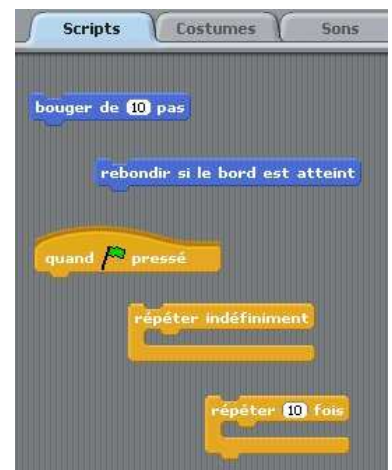
Il est important de commencer par définir les variables qui seront nécessaires dans l'algorithme. Pour cela, on utilise l'onglet « nouvelle variable », et on donne le nom de son choix à cette nouvelle variable.

Les différentes briques permettent alors d'attribuer des valeurs, d'afficher ou de cacher ces variables.

Le menu contrôle

Ce menu permet de contrôler l'exécution du programme. Il est constitué :

- de briques permettant d'exécuter le programme (soit en cliquant sur le drapeau vert, soit avec le clavier, soit avec un clic de souris sur l'objet « chat »),
- de briques permettant d'arrêter l'exécution du programme,
- de briques permettant de réaliser un temps d'attente, une répétition finie ou infinie d'opérations.
- Enfin de briques conditionnelles : Elles permettent aux objets de *prendre des décisions*. Elles nécessitent souvent l'emploi de briques des catégories CAPTEURS ou OPERATEURS.

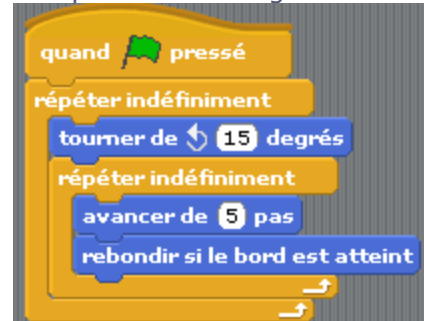


menu Mouvement

Ce menu permet de déplacer l'objet de différentes façons.

Exemple : Réaliser successivement les trois programmes suivants (dans l'espace « script ») , puis les faire fonctionner (en appuyant, comme c'est indiqué, sur le drapeau vert en haut à droite).

Pour prendre les objets, rendez-vous dans les différents menus puis les faire « glisser » à l'aide de la souris.



Stopper le programme (avec le symbole d'arrêt en haut à droite). Ramenez l'objet au centre. Recommencez plusieurs fois.

Menu opérateurs (espace « menu »)

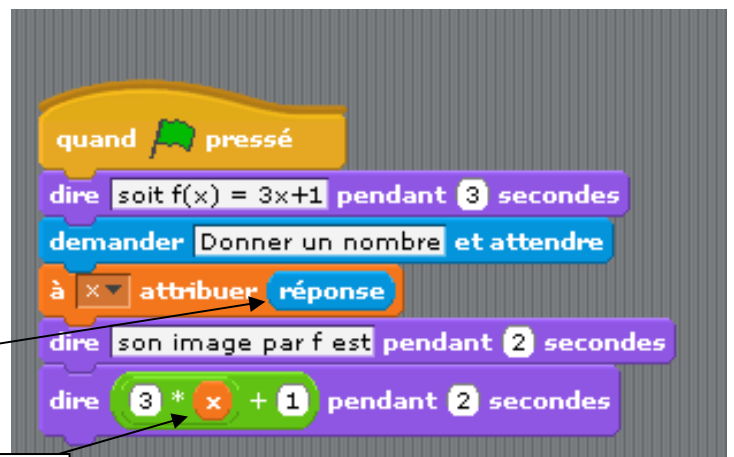
Il permet de réaliser beaucoup d'opérations mathématiques qui nous seront utiles dans les algorithmes.

Quand aux autres menus, nous vous laissons le soin de les découvrir par vous-même, notamment à travers les exemples qui suivent.

Etape 3 : A vous !

Exercice 1 : Vous reproduirez le programme ci-contre, en essayant de comprendre sa logique, puis vous le ferez fonctionner.

Vérification : si l'utilisateur « rentre » le nombre 4, la réponse donnée (l'image de 4 par f) doit être



Attention, ici ne pas taper « réponse », mais utiliser la brique « réponse » située dans le menu « capteur »

Attention, ici ne pas taper « x », mais utiliser la brique « x » située dans le menu « variable »

Exercice 2 : modifier ce programme pour qu'il nous donne sur commande les images d'un nombre x entré par l'utilisateur de la fonction $f(x)=7x^2-4x$. (Attention : Scratch ne connaît pas les priorités opératoires : il traitera en premier l'opération « la plus en avant »)

Vérifier que si l'utilisateur « rentre » le nombre 2, la machine « renvoie » 20 !

Exercice 3 : On vous donne ci-dessous un algorithme écrit de deux façons :

Écrit en langage naturel

*Initialiser une variable de type nombre « code », et une variable « essai ».

*L'objet choisi un nombre entier aléatoire entre 1 et 99, et l'affecte à la variable « code »

* L'objet demande « A quel nombre est-ce que je pense ? », et saisi la réponse dans « essai ».

*Répéter jusqu'à ce que « code » = « essai » :

- Si « code » > « essai », alors afficher « trop petit » et demander « essaye encore » ; affecter la réponse à « essai »

- Sinon afficher « trop grand » et demander « essaye encore » ; affecter la réponse à « essai »

*A la fin de la répétition, afficher « enfin trouvé »

Écrit sous forme codée

Entrée : Saisir « essai »

Initialisation : « code » prend un nombre entier aléatoire entre 1 et 99.

Traitement :

Tant que « essai » > « code », répéter:

- Si « code » > « essai » alors afficher « trop petit » puis demander « autre essai » ; saisir « essai ».

-Sinon afficher « trop grand » puis demander « autre essai » ; saisir « essai »

-FinSi

FinTant que

Sortie :

Afficher « enfin ! »

Écrire cet algorithme en langage « scratch » (en emboitant les briques), puis le faire fonctionner.

Enregistrer ce fichier sous le nom « scratchTP1 » dans votre répertoire, puis réduire le logiciel, et copier puis coller ce document dans le répertoire « devoir/Maths/maseconde »

Exercice 4 : Améliorer l'algorithme précédent pour qu'il « compte » le nombre d'essais faits par l'utilisateur. (Cela s'appelle « la pose d'un compteur »)

Exercice 5 : Voici un programme écrit sous forme codée.

- 1) Faire fonctionner cet algorithme avec $n=10$.
- 2) Expliquer à quoi sert cet algorithme.
- 2) Programmer cet algorithme avec scratch.

Écrit sous forme codée

Entrée : Saisir « n »

Initialisation : S prend la valeur 0

Traitement :

Pour i de 1 jusqu'à n

S prend la valeur $S+i$

FinPour

Sortie : Afficher S

ANNEXE

The image displays four panels of Scratch code blocks, organized by function. Each panel has a header with category tabs: Mouvement (Motion), Apparence (Appearance), Sons (Sound), Stylo (Pen), Contrôle (Control), Capteurs (Sensors), Opérateurs (Operators), and Variables.

- Panel 1 (Control):** Contains blocks for event-driven actions like "quand [drapeau] pressé", "quand [espace] est pressé", "quand [Objet1] pressé", "attendre [1] secondes", "répéter indéfiniment", "répéter [10] fois", "envoyer à tous", "envoyer à tous et attendre", "quand je reçois", "répéter indéfiniment si", "si", "si", "sinon", "attendre jusqu'à", "répéter jusqu'à", "arrêter le script", and "arrêter tout".
- Panel 2 (Variables):** Contains blocks for variable management: "Nouvelle variable", "Supprimer une variable", "à attribuer [0]", "changer [x] par [1]", "afficher la variable", "cacher la variable", and "Créer une liste".
- Panel 3 (Motion):** Contains blocks for movement: "avancer de [10] pas", "tourner de [15] degrés", "tourner de [15] degrés", "pointer en direction [90]", "pointer vers", "aller à x: [0] y: [0]", "aller à", "glisser en [1] secondes à x: [0] y: [0]", "remplacer x par [10]", "mettre x à [0]", "remplacer y par [10]", "mettre y à [0]", "rebondir si le bord est atteint", and checkboxes for "position x", "position y", and "direction".
- Panel 4 (Operators):** Contains blocks for mathematical and logical operations: "+", "-", "*", "/", "nombre aléatoire entre [1] et [10]", "<", "=", ">", "et", "ou", "non", "regroupe [salut] [monde]", "lettre [1] de [monde]", "longueur de [monde]", "mod", "arrondir", and "racine de [10]".