

Calculateur de la loi des gaz	11 ^e année – Chimie
Document	

Ce document décrit en partie la façon de créer une application de calculateur qui applique les lois des gaz apprises en 11^e année en utilisant MIT Scratch comme environnement de programmation.

La façon dont Scratch est structuré est que des scripts sont joints à des sprites : des acteurs individuels dans le jeu. Donc nous avons un sprite pour chaque loi des gaz qui invoquera les sprites associés aux variables qui s'appliquent à cette loi et un sprite qui servira de bouton « calculer » qui contiendra le script qui exécute le calcul.

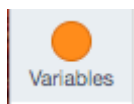
Le jeu est disponible sur Scratch à l'adresse suivante :
<https://scratch.mit.edu/projects/451122922>

Nous n'avons besoin que d'une seule vraie variable pour ce script : le numéro atomique N. Chaque script de Scratch commence avec un sprite (un chat souriant) et une variable (nommée « my variable » [ma variable]). Commençons par renommer cette variable.

Renommer et créer des variables

En programmation, la bonne façon de faire les choses est toujours de définir les variables avant de faire quoi que ce soit d'autre. Pour les lois de Boyle-Mariott, de Gay-Lussac et de Charles nous aurons besoin de six variables : V1, V2, P1, P2, T1 et T2.

Clique sur le cercle orange à la gauche de la fenêtre Scratch qui indique « variables ».



Une variable existe par défaut. Nous allons la renommer. Cliquez avec le bouton droit sur le bloc étiqueté *my variable* et sélectionnez l'option *rename variable (renommer la variable)*. Nommez-la *V1*.

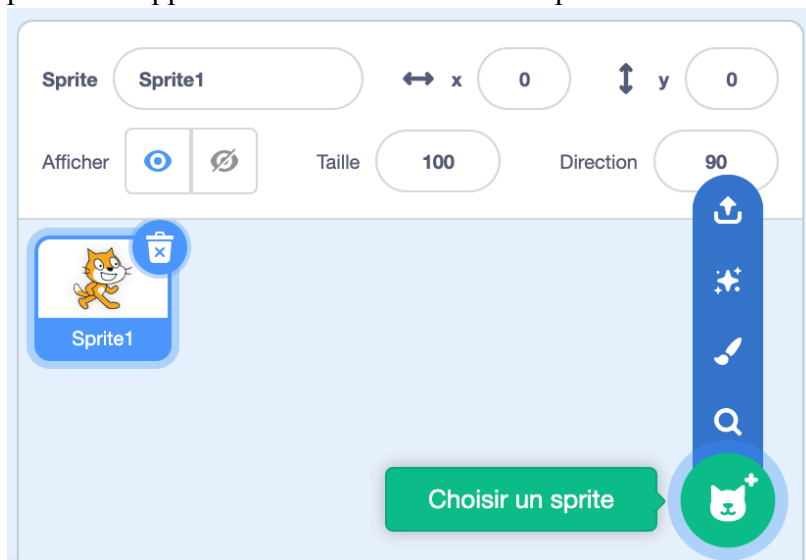


Ensuite, clique sur le bouton *Make a Variable (Créer une variable)* et crée les autres variables.

Avec les variables définies, il est temps de préparer le terrain pour notre jeu.

Sprites

Nous devons créer un certain nombre de sprites. Il y a un sprite par défaut, le chat Scratch. Nous pouvons supprimer le chat avec l'icône de poubelle.



Chaque sprite aura son propre code associé avec, contrôlant à la fois le sprite et le comportement des variables globales.

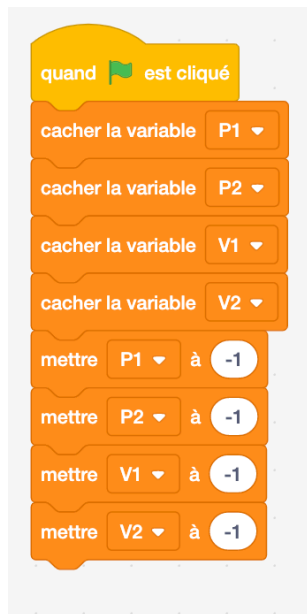
Bouton des lois des gaz

Chacun des boutons des lois des gaz sera extrêmement semblable, donc nous aborderons seulement celui de la loi de Boyle-Mariott ici. Il peut être reproduit pour créer des boutons pour la loi de Charles et la loi de Gay-Lussac.

Crée un nouveau sprite; nous avons utilisé l'élément « bouton-3 ». Pour ajouter du texte à notre bouton, nous avons d'abord sélectionné le sprite en cliquant dessus dans le panneau des sprites (mis en évidence par son cadre en bleu), puis en cliquant sur l'onglet « costumes » dans le coin supérieur gauche.



Utilisez l'outil de texte pour ajouter du texte à votre bouton dans la police de votre choix. Vous pouvez également être plus créatif et téléverser une image de votre ordinateur.



Programmer le bouton est plutôt simple. Au lancement du programme (indiqué par le moment auquel on clique sur le drapeau vert), nous voulons nous assurer que toutes les variables associées à cette loi des gaz sont cachées et fixées à une variable physiquement impossible, comme -1.

Un deuxième script est créé qui sera exécuté lorsque l'on clique sur le bouton (dans Scratch, *when this sprite is clicked*). Il diffusera un message d'événement que les autres sprites peuvent utiliser pour déclencher leur comportement. Nous devons créer ce message d'événement au moyen du menu déroulant. Le sprite sera aussi configuré pour prononcer (*say*) un rappel de ce que l'utilisateur doit faire : saisir trois variables en cliquant sur leur bouton.

(Remarque : La seule différence entre ce sprite et ceux des autres lois des gaz sera les variables contrôlées et le nom de la diffusion.)

Boutons de variables

De nouveau, puisqu'ils sont virtuellement identiques, les boutons associés à chaque variable ne seront pas tous abordés. Nous ne discuterons que du bouton pour la variable V1.

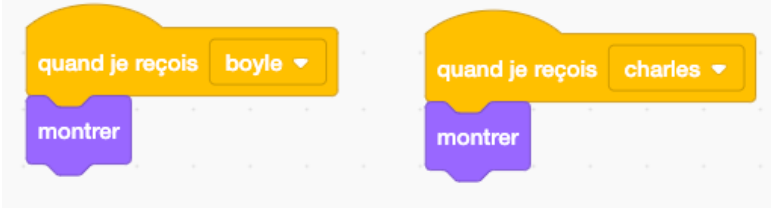


Comment précédemment, modifie le sprite au moyen de l'outil de peinture de l'onglet « Costumes » ou téléverse ta propre image.

La première chose est de s'assurer que le bouton se cache lorsqu'il est initialisé.



Puis qu'il se montre lorsque l'un des messages des lois des gaz applicables est diffusé.



(Remarque que ce sont tous des scripts séparés, déconnectés et indépendants!)



Enfin, nous avons besoin d'un script pour contrôler le comportement du bouton lorsque l'on appuie dessus : nous voulons

(Remarque que nous voulons placer la variable maintenant affichée quelque part d'évident; dans l'exemple, elle est placée pour se trouver au-dessus du bouton.)

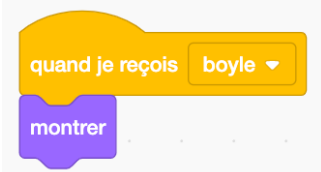
Boutons de calcul

Bien que cela ne soit pas visuellement évident dans la démonstration, chaque loi des gaz a son propre sprite pour fournir un bouton de calcul qui contient les scripts pour son calcul. Encore, nous aborderons seulement la loi de Boyle-Mariott, puisqu'il n'y a aucune différence majeure.

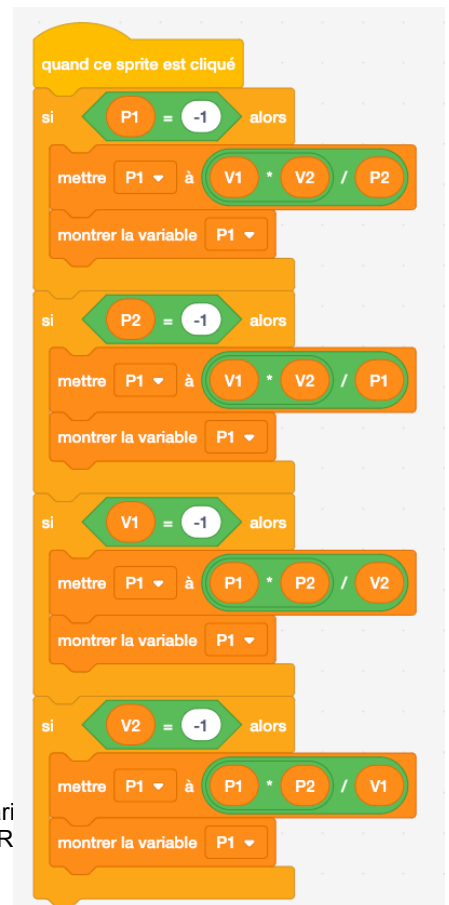
Nous ne voulons pas que le bouton soit montré lorsque le programme est lancé, donc il doit être caché au démarrage.



Et il doit se montrer lorsqu'il reçoit le message.



Les calculs ne sont pas, en eux-mêmes, très compliqués. La seule anicroche est de s'assurer que les bons calculs sont faits. C'est



pourquoi le programme est configuré pour initialiser chaque variable avec une valeur de -1 : afin de pouvoir déterminer pour quelle variable l'utilisateur n'a pas saisi de valeur. La variable sans valeur saisie par l'utilisateur est celle que nous voulons calculer. Ainsi, le code est une tour d'instructions SI qui vérifient chaque variable une après l'autre pour voir si elle a une valeur de -1. Si, par exemple, P1 = -1, alors elle est calculée par la loi de Boyle-Mariott et la variable est affichée. Autrement le code vérifie P2 et ainsi de suite.

Voici quelques questions pour ceux plus intéressés par la programmation que la chimie :

Qu'arrive-t-il si l'utilisateur saisit -1 pour une ou plusieurs valeurs?

Qu'arrive-t-il s'il tape des nombres réguliers pour les quatre valeurs?

Croyez-vous que cela respecte les bonnes pratiques de programmation et, sinon, comment changerez-vous le code?