

Utiliser des variables

21 Compter des points

Objectif
Gérer des variables.

Je découvre

Les variables sont très utiles en programmation. On peut considérer une **variable** comme une « boîte » contenant une valeur. Chaque boîte a une étiquette : c'est le nom de la variable. On parle de « variable » car elle peut varier au cours du temps. Au début d'un jeu, une variable « score » peut valoir 0, puis être augmentée par la suite. On dit qu'elle est **incrémentée**.

Pour créer une variable, clique sur **Créer une variable** dans la catégorie **Données** et choisis **Pour tous les lutins**.

Créer une variable

Visualiser (ou non) la variable sur la scène.

Initialiser la variable avec un nombre ou du texte.

Augmenter ou incrémenter la variable score d'une valeur numérique k (ici $k = 1$).

Faire apparaître ou cacher la variable sur la scène lors du déroulement d'un script.

Pour certains projets, on peut utiliser **si < >** pour comparer des variables ou **+** pour effectuer des opérations. Ces briques sont proposées dans la catégorie **Opérateurs**. Par exemple, on peut ainsi tester si une variable dépasse une certaine valeur pour féliciter le joueur.

Les valeurs des variables sont conservées entre deux jeux, sauf si on les remet à 0 !



À toi de jouer !

Projet 26 suite Balle au mur

- Ouvre ton projet Balle au mur.
- a. Crée la variable « score » initialisée à 0 et affichée sur la scène.
b. Incrmente le score de 1 s'il y a rebond sur la raquette.
- Complète le script en suivant cet algorithme :

Si le score est égal à 10 alors :
Afficher « Gagné ! » durant 1 seconde.
Stopper le jeu.

- Teste et enregistre ton projet.

Exercice 1 Calcul de points

Calcule la valeur finale de la variable « points » :



Tu peux tester ce script comme un robot, pas à pas, en notant les valeurs intermédiaires.

22 Demander des réponses

Objectif
Interroger le joueur.

Je découvre

Avec Scratch, on peut s'amuser à interroger le joueur et jouer avec ses réponses.

Par exemple, ce script permet de **demandeur au joueur** de saisir 2 nombres puis d'en afficher le produit.

En regardant le script associé, on remarque plusieurs choses :

- la **réponse à la question** **demandeur et attendre** est aussitôt récupérée et stockée dans la brique **réponse** de la catégorie **Capteurs**.
- La brique **regroupe** est utilisée pour afficher deux textes côte à côte.

À toi de jouer !

Projet 29 début Opérations



- Crée ce projet avec Dee sur la scène de ton choix.
- a. Reproduis le script de la partie *Je Découvre* ci-dessus.
b. Modifie-le pour afficher le plus grand des deux nombres en utilisant la brique « si alors sinon ».
- Teste et enregistre ton projet.

Projet 29 BONUS Multiples ?

On voudrait savoir si « nombre 1 » est multiple de « nombre 2 ». Pour cela, on utilise **modulo** qui donne le reste de la division du premier nombre par le second.

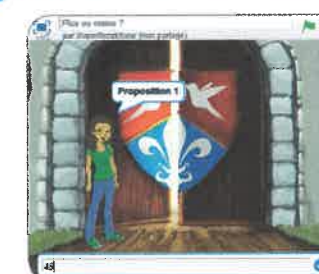
- Que vaut $16 \text{ modulo } 4$?

- a. Complète cet algorithme qui affiche « multiple » si le premier nombre est multiple du second, « non multiple » sinon :

Si alors :
.....
Sinon :
.....

- Crée le script associé.

Projet 30 début Plus ou moins ?



Abby propose au joueur de deviner en un minimum de coups, à quel nombre entre 1 et 100, elle pense.

- Crée ce projet avec Abby.
- Crée la variable « Mystère » (cachée) et le script qui lui associe un nombre entier aléatoire entre 1 et 100.
- Modifie le script en suivant cet algorithme :

Répéter indéfiniment :

Abby demande au joueur de taper un nombre qui est stocké dans la variable « Choix ».

Selon la situation, Abby affiche un message :

« Trop grand » si « Choix » > « Mystère » ;
« Trop petit » si « Choix » < « Mystère » ;
« Bravo ! » si « Choix » = « Mystère » (arrêt du jeu).

- Teste et enregistre ton projet.

Projet 30 BONUS Minimum d'essais

- Affiche le nombre d'essais du joueur pour trouver le nombre mystère.
- Quelle est la meilleure stratégie pour trouver en un minimum d'essais, le nombre « Mystère » ?