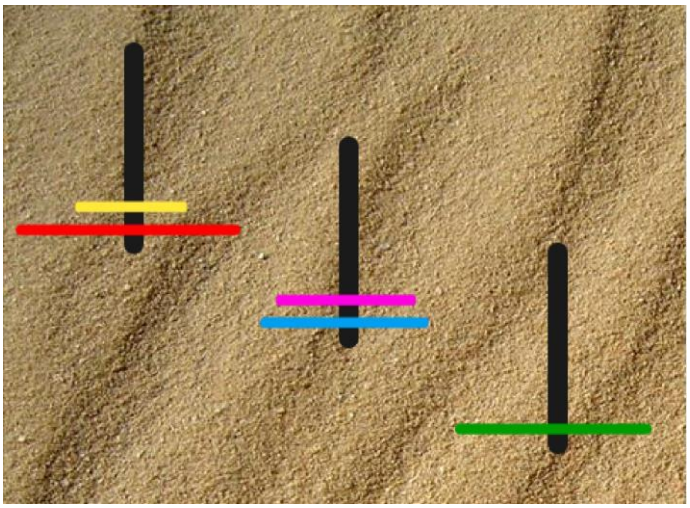
Les tours de Hanoï



Présentation / Règle du jeu :

Vous allez devoir programmer le jeu de réflexion des tours de Hanoï (voir vidéo)

Ces 3 tours peuvent comporter jusqu'à 8 anneaux.
Ces anneaux viennent s'empiler sur chacune des tours.

On peut déplacer un anneau d'une tour à une autre selon les règles suivantes :

- on ne peut déplacer qu'un anneau à la fois, celui situé en haut de chaque pile
- on ne peut déposer un anneau sur une tour que s'il est plus petit que celui du haut de la pile

Au début de la partie, les anneaux sont empilés sur la tour de gauche.

L'objectif pour le joueur est de parvenir à empiler les anneaux sur la tour de droite.

Commentaire :

Plus le nombre d'anneaux est élevé, plus la partie sera difficile et longue pour le joueur.
A titre indicatif, si on joue avec 5 anneaux, la partie pourra se gagner en 31 coups minimum, et si on joue avec 8 anneaux, il faudra au moins 255 coups.

La durée du jeu concerne le joueur, mais pas celui qui programme.

Méthode de travail :

Vous allez travailler par équipes de 3 ou 4, sur au moins 4 séances.
Le travail effectué par chacun devra être indiqué dans le tableau ci-après.
Le temps à prévoir pour réaliser ce projet dépasse le temps à y consacrer en classe.

Travail réalisé :

NOM	Séance 1	Séance 2	Séance 3	Séance 4

Le travail final sera évalué compte-tenu de l'implication de chacun, et du rendu global de l'équipe.
Il est possible qu'à l'issue du temps qui vous sera imparti, votre projet soit inabouti : ne cherchez pas absolument à le terminer, mais privilégiez la qualité du travail rendu à la quantité.

Le fichier de départ **Hanoï.sb2** vous est fourni avec :

- 1 arrière-plan comportant les 3 tours
- 8 lutins (les anneaux)
- 1 bloc de démarrage **(cliquer sur l'arrière-plan pour le voir)**
- 1 bloc-fonction « Positionner » avec deux paramètres
- 3 listes (Tour 1, Tour 2, Tour 3).

La plupart de votre code sera à écrire dans les scripts de l'arrière-plan.

Chaque liste comporte <u>les tailles</u> des anneaux empilés :

Ainsi, dans le fichier fourni, pour la tour 1 :

- l'élément n°1 a pour taille 8.
- l'élément n°2 a pour taille 7.
- etc.

Objectif n°1 : Déterminer quelle tour est cliquée

Pour commencer, créez une variable qui affiche le numéro de la tour cliquée (1, 2, 3, ou rien).

Vous aurez notamment besoin des capteurs 

et 

.
Pour simplifier, considérez que chaque tour est cliquable dans un rectangle qui l'entoure dont vous déterminerez les coordonnées.

Objectif n°2 : Positionnement des anneaux

Votre objectif va être de faire en sorte qu'en fonction des nombres donnés dans chaque liste (les nombres correspondent aux tailles), les anneaux se placent sur chaque tour, uniquement en cliquant sur le drapeau.

Pour cela, vous utiliserez le bloc-fonction **Positionner (Tour, Numéro anneau)** que vous devrez renseigner pour calculer la position, sur la **Tour** donnée de l'anneau de **numéro** indiqué, avant d'envoyer un message de positionnement à l'anneau en question.

Durant vos tests, vous répartirez les valeurs 1, 2, 3, etc. dans les différentes listes pour vérifier qu'en cliquant sur le drapeau les anneaux se placent correctement.

Veillez à ce que ces valeurs qui correspondent aux tailles ne soient pas données en doublon sur deux listes durant vos tests.

Formule de calcul des coordonnées d'un anneau sur une tour :

Si on appelle :
- **Tour** le numéro de la tour sur laquelle positionner un anneau,
- **Numéro anneau** à positionner (1 = en bas, 2 = celui du dessus, …)

Alors, si Tour = 1 :
- **Abscisse** = -148 - Taille anneau ÷ 2
- **Ordonnée** = 7 + Numéro anneau × 16

Si Tour = 2 :
- **Abscisse** = 3 - Taille anneau ÷ 2
- **Ordonnée** = -58 + Numéro anneau × 16

Si Tour = 3 :
- **Abscisse** = 150 - Taille anneau ÷ 2
- **Ordonnée** = -132 + Numéro anneau × 16

La taille de l'anneau se lit dans la liste correspondant à la tour.

Objectif n°3 : Déterminer si un déplacement est valide

Pour déplacer un anneau, il suffira de cliquer sur la tour de départ, puis sur la tour d'arrivée (Scratch ne gère pas facilement le déplacement des objets par glisser-déposer).
En utilisant une variable Tour de départ et une autre variable Tour d'arrivée, vous allez déterminer si le déplacement est valide (voir règle du jeu).

Objectif n°4 : Déplacement d'un anneau

Lorsqu'un déplacement est valide, effectuez-le en :

- retirant la taille de l'anneau de la liste où il était
- ajoutant la taille de l'anneau dans la liste où il arrive
- appelant la fonction Positionner pour dessiner l'anneau à sa nouvelle place

Objectif 5 : Fin du jeu

Gérez la fin du jeu : il se termine lorsque tous les anneaux sont sur la tour n°3, c'est-à-dire lorsque la taille de la liste Tour 3 est égale au nombre d'anneaux.

Variante 1 : Choix du nombre d'anneaux

Plutôt que de jouer toujours avec 8 anneaux, faites choisir à l'utilisateur son niveau de difficulté (entre 2 et 8 anneaux).

Variante 2 : Chronomètre et Score

Vous pouvez afficher un chronomètre à l'écran, ainsi que le nombre de coups effectués depuis le début de la partie.

Formule de calcul du nombre minimal de coups :

> Si on appelle **n** le nombre d'anneaux, alors le nombre minimal de coups nécessaires pour gagner vaut $\mathbf{2^n-1}$

Vous pouvez faire calculer le score qui s'affiche un pourcentage de réussite qui varie de 100 % lorsque le joueur a utilisé le nombre minimal de coups à 0% s'il en a utilisé au moins le double.

En appelant **x** le nombre de coups et **m** le nombre minimal de coups et **S** la fonction qui, à **x** associe le score, on a :

$$\begin{array}{ll} \text{Si } m \leq x \leq 2m : & S : x \mapsto -\frac{100}{m}x + 200 \\ \text{Si } x > 2m : & S : x \mapsto 0 \end{array}$$