

## Algorithmique : Exemple de repères de progressivité

Programmation séquentielle : exécution d'une suite d'instructions dans un ordre programmé. Une fois lancé, nous n'avons plus la main sur les actions. Il n'y a pas d'interaction possible.

Programmation événementielle : le programme réagit à des événements extérieurs (aux touches, à la souris, aux réponses données...) ou intérieurs (interactions avec d'autres lutins, valeur d'une affectation).

### 6ème



Programmation débranchée (décrire un chemin)  
Puis un seul lutin, programme séquentiel avec le stylo en position d'écriture (notion de direction...)  
Notion d'algorithme, de temporalité.

### 5ème



Plusieurs lutins, programmes séquentiels \* actions en parallèle non événementielles. Premières boucles.



Un seul lutin avec programme événementiel \* instructions conditionnelles, notion de variable.

### 4ème

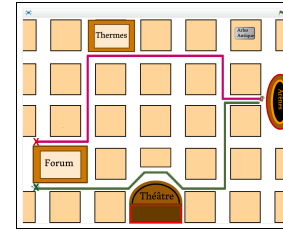


Plusieurs lutins, programmation événementielle \* actions en parallèle, interactions avec des événements intérieurs (entre lutins) et extérieurs (action des flèches...)

### 3ème

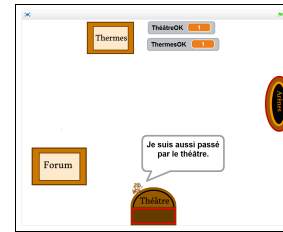


Mêmes compétences qu'en 4ème. Les instructions conditionnelles peuvent être imbriquées. Les structures sont plus complexes. On ajoute des blocs, de l'aléatoire, on répartit les tâches d'un projet.

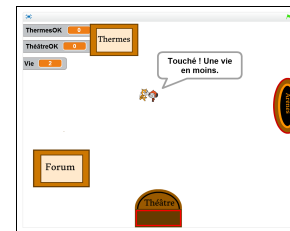


Programmation débranchée.

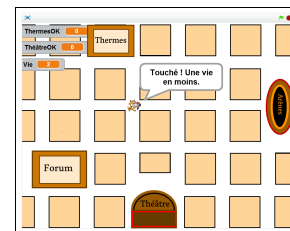
Puis avec Scratch :  
Un seul lutin, description d'un chemin, d'une figure, avec un programme séquentiel, notion de direction. **Stylo en position d'écriture.**  
Même chose avec deux lutins.  
Les deux programmes séquentiels sont en parallèle. Boucles simples.



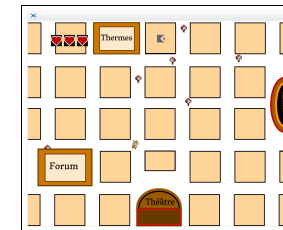
Un lutin dirigé par les quatre flèches. Programme en événementiel : interaction avec l'extérieur (les touches) et l'intérieur (reconnaissance des autres lutins statiques)  
Instructions conditionnelles simples.



Deux lutins dirigés par les touches.  
On commence par un décor neutre (comme le précédent)...



Puis on ajoute un décor de type labyrinthe.  
Les variables d'affectation (sous forme de compteur par exemple) apparaissent clairement, les messages également.



On ajoute des contraintes, des blocs.  
Les instructions d'affectation sont plus complexes.  
Démarche de projet.

