

Cahier des charges : Dideblocks

Objectif

Nous devons programmer un environnement de programmation visuelle, c'est-à-dire un langage de programmation graphique tel que Scratch, Snap ou Blockly.

Description fonctionnelle

Une fenêtre composée de trois zones :

- les CHOIX de commandes qui sont sous forme de blocs pouvant s'emboîter suivant leurs différentes formes.
- l'ÉDITION du programme, la zone vers laquelle l'utilisateur dépose les blocs qui constituent le corps de son programme.
- l'EXECUTION où l'on pourra visualiser le résultat du programme. Un sprite se trouvera dans cette zone.

La possibilité de glisser-déposer des blocs de la zone CHOIX vers la zone EDITION.

Une catégorisation des commandes suivant que c'est une variable, une expression booléenne, une fonction, une boucle, un événement. On utilisera un code couleur et un code forme pour les blocs.

Si une association de deux blocs est impossible, le bloc que l'on vient de déplacer retourne dans la zone CHOIX et l'action est annulée.

La possibilité de créer ses propres fonctions et ses propres variables qui pourront être réutilisées dans le programme (en créant une entrée dans la zone CHOIX).

Les actions élémentaires (void) possibles seront les suivantes :

- Déplacer le sprite.
- Changer l'apparence du sprite (peut être des animations prenant plusieurs frames).
- « Lever » ou « baisser » le crayon : le sprite pourrait laisser une trace derrière lui quand il se déplace, ce qui permettrait de faire des dessins.

Les événements qui peuvent déclencher l'exécution des fonctions seront les suivants :

- Manuellement, en cliquant sur un bouton associé à chaque fonction de la zone EDITION.
- Suite à une input de l'utilisateur (au clavier ou autre).
- On pourrait penser à des conditions compliquées, sur la position où l'état du sprite.

Puis, si on a le temps :

- La sauvegarde des programmes pour pouvoir les rouvrir ultérieurement.
- Pour l'instant, l'utilisateur n'a pas de but précis et peut s'amuser à sa guise avec le sprite (un esprit bac à sable). On peut envisager de créer des mini-jeux que l'utilisateur doit gagner grâce à la programmation (exemple : un labyrinthe).

