CHELLE Raymond 02/09/2025

Rapport de TP : Classic Maze

1. Analyse du problème

L'objectif de ce TP est de programmer un personnage pour qu'il atteigne un objectif dans un labyrinthe avec une interface de programmation par bloc sur Code.org sur 20 parcours. Le problème est qu'il doit se déplacer dans ce labyrinthe en évitant des obstacles comme des murs par exemple, tout en utilisant des instructions de déplacement(avancer, tourner à droite, tourner à gauche) et des structures de contrôles (boucles et conditions) permettant de diminuer le nombre de bloc et par conséquent un gain de temps.

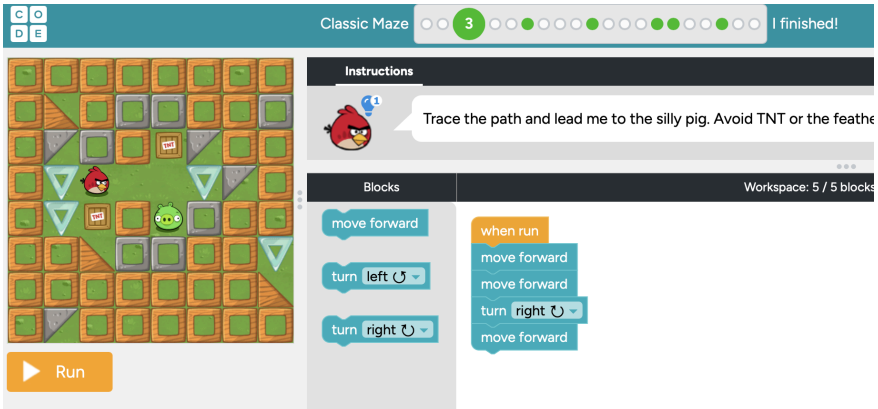
Cahier des charges :
Trouver une séquence d'instruction la plus courte et correcte afin que le personnage atteigne l'objectif, tout en respectant les contraintes du labyrinthes et les blocs donnés

2. Conception de la solution

La résolution de chaque labyrinthe peut être décomposée en plusieurs sous problèmes :
-Analyse spatiale du labyrinthe : Nous devons tout d'abord identifier le chemin optimal du départ à l'arrivée, pour s'y faire nous devons observer ce parcours et visualiser les déplacements
-Identifier des séquences qui se répètent permettant de diminuer le nombre de bloc, pour s'y faire il faut faire des programmes similaires jusqu'à respecter le nombre de bloc maximum pour un parcours
-Choisir les blocs appropriés : Il est en effet nécessaire de choisir les blocs adéquats notamment pour respecter la limite de blocs. Donc il faudra bien choisir entre séquentielle, boucle, et conditionnelle
-Assemblage de la solution finale : Il faut donc réussir à combiner tout ça grâce au bloc proposer

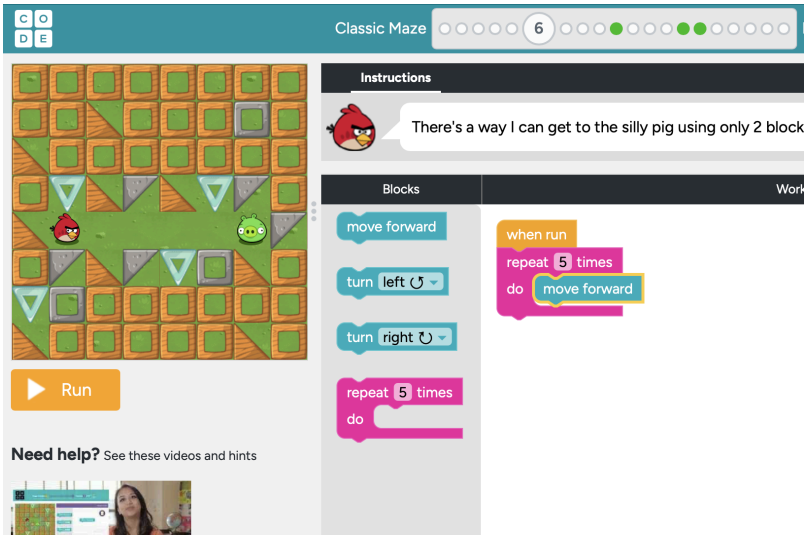
L'analyse spatiale influe directement sur le choix des blocs, déterminant ainsi le programme final

3. Réalisation

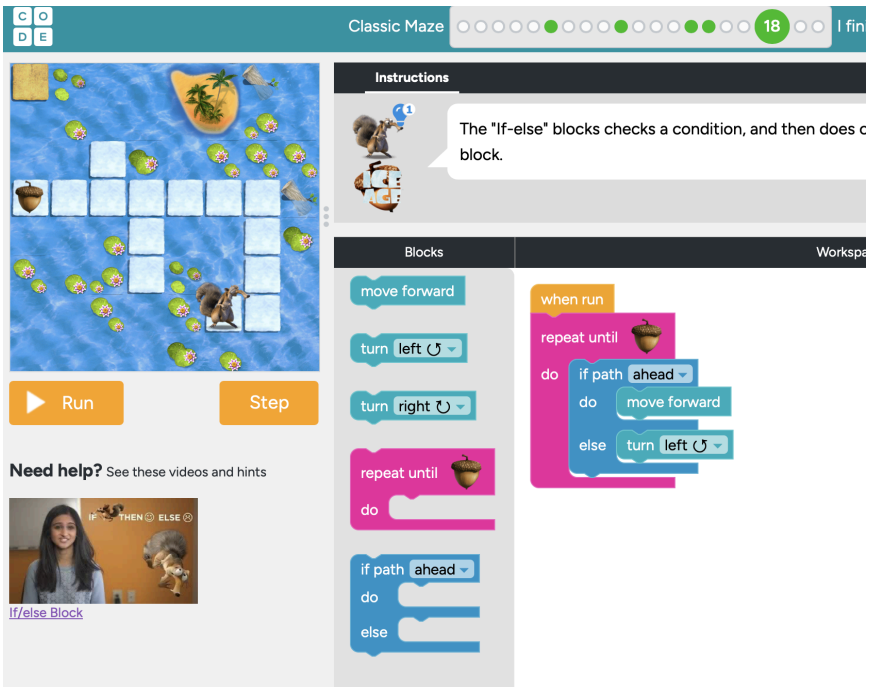


Dans le parcours 3, on fait face à des blocs **séquentielles** avec une série de 5 instructions, par une rapide analyse spatiale, j'ai remarqué que le personnage devait seulement avancer 2 fois, puis tourner à droite puis avancer une fois. Par conséquent, j'ai mis 2 "move forward", puis turn right, puis move forward.

Donc ce parcours a été réussi avec succès.



On remarque que dans ce parcours 6, on a l'apparition d'un **système de boucle**, donc au lieu de mettre 5 fois “move forward”, on a tout simplement à mettre le nouveau bloc en n'oubliant pas de rentrer la valeur 5, désignant le nombre de fois que l'instruction sera répété



Dans ce parcours 18, on a un **système de condition** qui est introduit, c'est-à-dire ici, si le chemin est dirigé vers l'avant, alors on avance une fois vers l'avant, sinon le personnage tourne à gauche.
Et pour mettre le moins de bloc possible j'ai aussi mis une boucle qui se répète jusqu'à l'arrivée à l'ensemble du programme.

4. Conclusion

Ce TP m'a permis de comprendre l'importance d'analyser rigoureusement un problème avant de commencer à coder. Le principal défi pour moi a été d'optimiser la solution avec des boucles ou encore avec les blocs de conditions

Bilan des efforts :
J'ai passé environ 30 minutes à analyser le labyrinthe et à visualiser une séquence d'instructions puis 30 autres minutes pour placer les blocs et tester. Mais une meilleure planification m'aurait permis de réduire le temps passé.