03/09/2025

Bloc 1 – TP01

Code sur screatch

Sommaire

Introduction 1
I. Conception 1
Premier niveau : l'oiseau et le cochon 1
Niveau suivant : Plantes vs Zombies 2
II. Réalisation 2
III. Test 3
IV. Retour d'expérience 3
Conclusion 3

Fait par : Mattéo MOURANCHON – Groupe 2

Introduction

Le **jeu Bloc sur Code.org** est un jeu éducatif en 2D qui permet d'apprendre la logique de la programmation à l'aide de blocs de commandes visuelles.

L'objectif est de déplacer un personnage (oiseau, plante ou zombie) d'un point de départ vers un objectif (cochon, fleur, etc.), en contournant les obstacles grâce à une suite d'instructions.

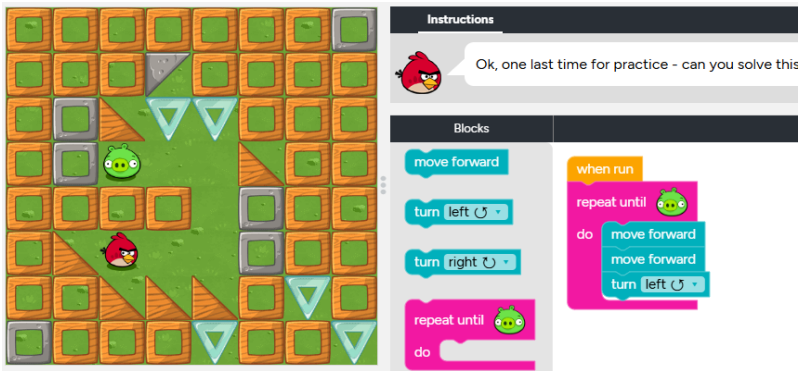
Ce TP avait pour but de comprendre les bases de la **programmation par blocs** et d'expérimenter les notions fondamentales de la logique algorithmique : les **boucles** et les **conditions**.

I. Conception

Premier niveau : l'oiseau et le cochon

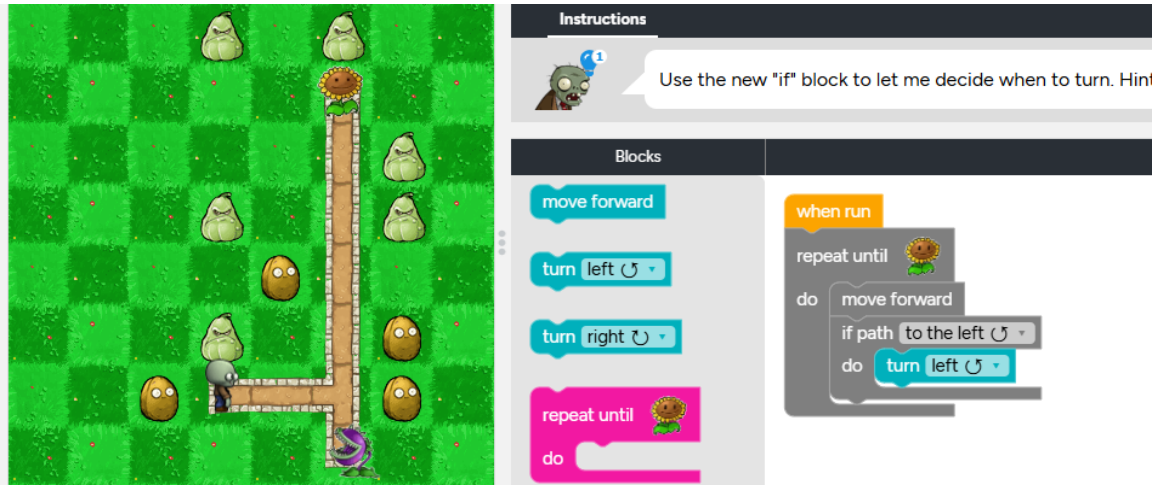
Dans le premier niveau, l'objectif était de déplacer l'oiseau jusqu'au cochon en contournant les obstacles.
Pour cela, j'ai utilisé :

« repeat until do » : répète les instructions jusqu'à ce que le but soit atteint.
« move forward » : fait avancer l'oiseau d'une case.
« turn left » ou **« turn right »** : permet de changer la direction du personnage pour éviter un obstacle.



Niveau suivant : Plantes vs Zombies

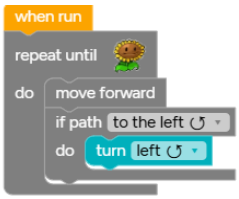
Dans ce niveau, deux nouveautés apparaissent :



Les blocs **gris**, imposés par le jeu et impossibles à supprimer.
La commande conditionnelle « **if path do** », qui permet de tester si un chemin est libre ou bloqué. Si le chemin est libre, le personnage exécute l'action prévue (par exemple tourner à gauche). Sinon, il continue tout droit.

Ces outils offrent plus de flexibilité et permettent de rendre le personnage **autonome** dans ses déplacements.

II. Réalisation



Découverte de l'interface : prise en main de l'environnement Code.org et compréhension du fonctionnement des blocs.

Analyse des niveaux : observation du point de départ, du chemin, des obstacles et de l'objectif.

Construction du programme : assemblage des blocs pour diriger correctement le personnage.

Nouveaux concepts : utilisation des boucles pour répéter les actions et des conditions pour gérer les obstacles.

Améliorations : ajustement du code lorsqu'il ne donnait pas le résultat attendu

III. Test

Une fois le code écrit, chaque niveau a été exécuté pour vérifier le bon fonctionnement du programme :

- **Niveau Angry Birds** : l'oiseau a bien atteint le cochon après avoir avancé de deux cases puis tourné.
- **Niveau Plantes vs Zombies** : grâce à la condition if path do, le zombie a pu s'adapter au terrain et trouver le bon chemin vers la fleur.

IV. Retour d'expérience

Ce TP m'a permis de découvrir la programmation par blocs et de comprendre l'utilité des boucles et des conditions. J'ai vu qu'avec quelques instructions simples, on peut déjà résoudre des problèmes et rendre un programme plus intelligent. Les blocs imposés m'ont parfois limité, mais cela m'a obligé à chercher d'autres solutions et à mieux réfléchir à la logique du parcours. J'ai appris qu'il est important de tester son programme étape par étape et de corriger ses erreurs pour arriver à une solution efficace.

Conclusion

Ce travail m'a permis d'acquérir de nouvelles compétences en **programmation par blocs** et de mieux comprendre le rôle des instructions de contrôle comme les boucles et les conditions. Grâce aux différents niveaux proposés (Angry Birds, Plantes vs Zombies, Âge de glace), j'ai pu progresser de façon progressive et amusante.

Ce TP m'a montré qu'avant de résoudre un problème, il est essentiel d'analyser la situation, de planifier les actions à exécuter et de tester régulièrement son programme pour corriger les erreurs. C'est une première approche très utile qui me servira pour aborder des langages plus complexes comme **Java** ou **Python**, et pour développer des compétences solides en logique et en programmation.