



Travail pratique individuel (TPI)

Informaticien-ne CFC

Dossier d'inscription et description du travail (A remplir par le formateur)

Ce formulaire peut être téléchargé à l'adresse : <https://1drv.ms/f/s!AtCBFcUYDpzzgRdtwK3Jf6IXa7S->

Ce document sera connu du candidat uniquement au commencement du TPI. Il est interdit d'en communiquer le contenu au candidat avant la date de TPI convenue.

Candidat :

Nom : Freddi

Prénom : Sam

Société : CFPT Ecole Informatique

Objectifs du TPI

Les objectifs évalués de ce Travail Pratique Individuel sont les capacités du candidat :

- à réaliser et tester une application selon le cahier des charges ci-dessous dans les délais impartis
- à tenir un journal de bord durant le projet
- à rédiger la documentation demandée
- à présenter oralement le projet et à répondre aux questions des experts

Cahier des charges

1 Résolution d'un labyrinthe par la méthode d'un arbre de recherche

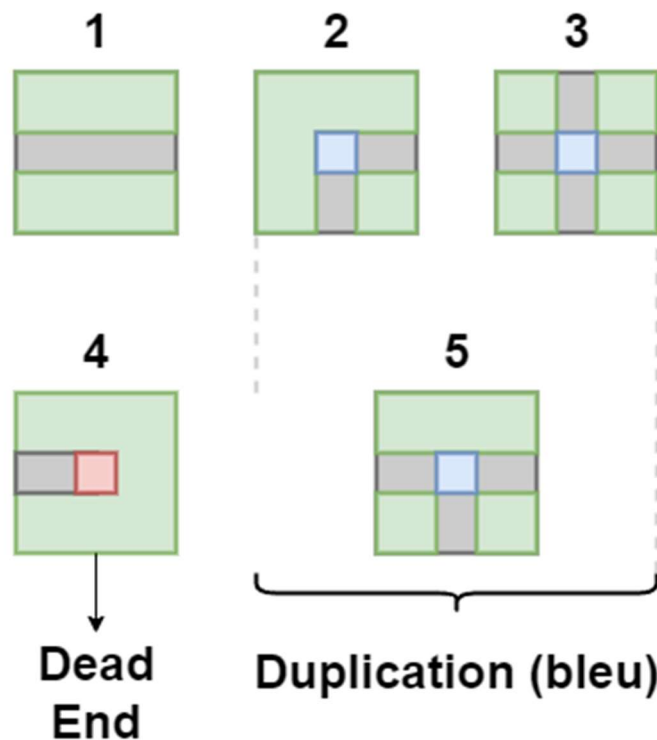
2 Matériel et logiciels à disposition

- Un PC standard école
- Windows 10
- Unity3D
- Carte Graphique Nvidia RTX3050 au min

3 Prérequis

- Compétences décrites dans le plan de formation, acquises durant la formation, aussi bien au niveau des modules que des ateliers
- Avoir achevé les éléments suivants dans les ateliers New Technologie et TPI.
 - La construction d'un labyrinthe automatiquement avec, dans l'idéal, une seule entrée et une seule sortie, avec cinq briques de base:

On doit pouvoir sauvegarder les labyrinthes intéressants et pouvoir les importer dans le projet, à des fins de test ou de démonstration, l'interface utilisateur devra donc contenir cette fonctionnalité.



- Le bot qui parcourra le labyrinthe, aura déjà les fonctions de bases, lui permettant de se déplacer (N, S, E, O), de détecter les différents nœuds du labyrinthe, sans se cogner.

4 Descriptif complet du projet

Le but du projet est de trouver le plus court chemin entre l'entrée et la sortie d'un Labyrinthe. L'utilisateur choisit l'entrée et la sortie. On n'impose pas une entrée et une sortie unique pour simplifier l'algorithme de création du labyrinthe. Toutefois, il faut avoir conscience que l'on s'expose à des solutions insolubles. Il faudra que le logiciel le détecte et le fasse savoir à l'utilisateur d'une manière ou d'une autre.

Pour cela, on utilisera un algorithme fondé sur la recherche dans un arbre, le but étant de visualiser, en directe, l'algorithme dans le labyrinthe.

Nous utiliserons, à cette fin, un algorithme assez simple. Un bot est constitué d'un capteur et d'une trace du parcours déjà effectué.

À chaque "nœud", on duplique le bot dans les trois directions.

Chaque bot dont le capteur touche un mur lorsqu'il est dupliqué et positionné meurt, de même que sa trace. Ne reste que le ou les bots qui ont résolu le problème. On affiche la trace en vert du premier qui réussit à trouver la sortie choisie.

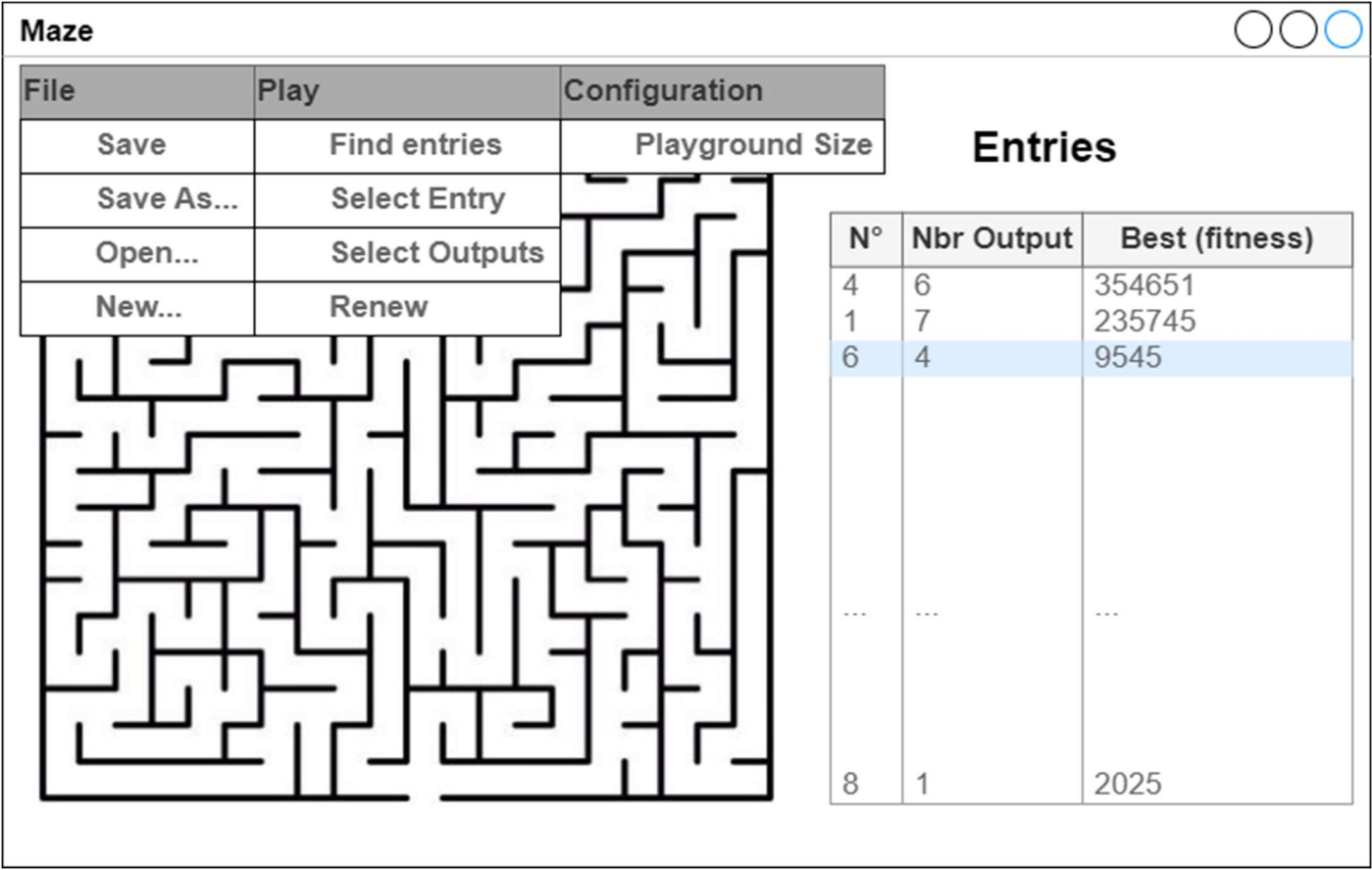
La trace est en bleu lorsque les bots cherchent la solution. Elle est en rouge pendant une-demie seconde avant de disparaître, lorsqu'ils sont en collision avec un mur. Enfin, en jaune pour la trace qui est la plus proche de la sortie choisie.

Si la solution n'est pas trouvée, le programme l'indique explicitement et demande à l'utilisateur s'il veut reprendre en sélectionnant une autre sortie, s'il recommence depuis le début ou s'il préfère récupérer un labyrinthe déjà existant.

Un dernier point à mettre en place est de tester pour une entrée donnée toutes les sorties possibles et indiquer quelles sont les solutions possibles, après avoir effectué l'algorithme.

On classera les résultats du plus simple au plus compliqué (du plus court chemin au plus long, du plus grand nombre de sorties au plus bas) en les affichant dans un tableau. La sélection dans le tableau permet d’afficher le chemin directement sur le labyrinthe en vert. Toutes les autres traces restent et sont affichées en bleu.

Voici un exemple d’interface utilisateur proposée pour le projet, à compléter/modifier en fonction de la complexité de mise en œuvre:



Attention, l’image du labyrinthe ci-dessus n’est pas représentative de ce que l’on attend réellement dans le projet (voir les briques de constructions plus haut).

Un point important : nous souhaitons conserver une trace des 10 meilleurs labyrinthes créés, classés selon leur performance (la meilleure trace). Il est essentiel de pouvoir les sauvegarder, les classer et les ouvrir pour les comparer.

5 Livrables

- Rapport de projet
- Manuel utilisateur
- Résumé du rapport du TPI
- Journal de travail

6 Points techniques évalués spécifiques au projet correspondants aux points A14 à A20 du formulaire d’évaluation

- A14 : La possibilité de sauvegarder et d’ouvrir un labyrinthe est présente dans l’interface utilisateur,
- A15 : Les traces sont de la bonne couleur et indiquent les chemins parcourus par les bots,
- A16 : Le candidat utilise correctement le modèle objet proposé par Unity3d,
- A17 : La ou les solutions sont explicitement identifiées, s’il y en a plusieurs, la meilleure en vert,

A18 : Utilisation d'un capteur "laser" pour déterminer la présence d'un mur devant le bot lors de la duplication,
A19 : L'utilisateur peut choisir les entrées et les sorties directement sur l'interface graphique,
A20 : Le classement des meilleures solutions permet de sélectionner et d'afficher le chemin spécifique.