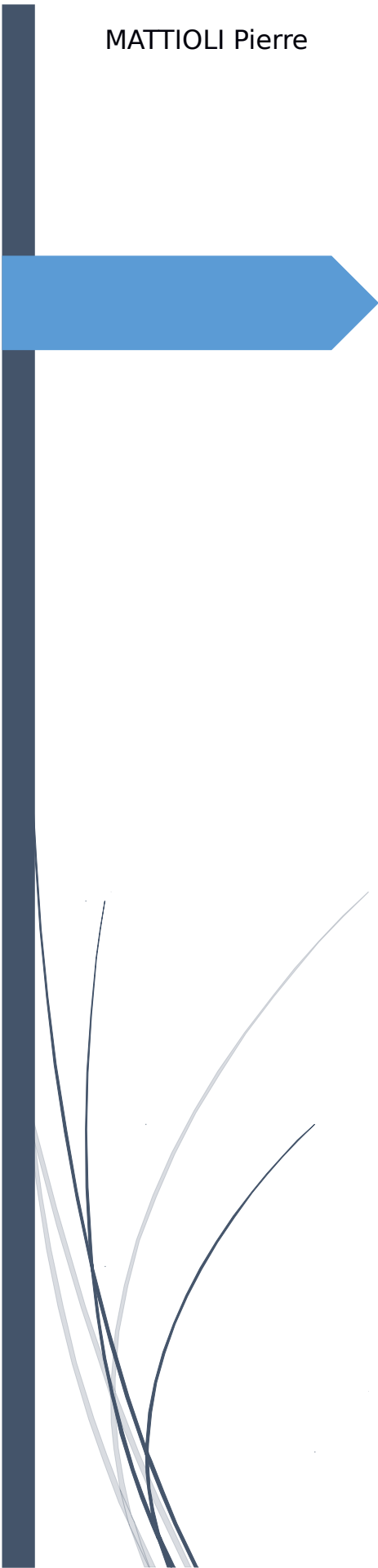


MATTIOLI Pierre

# Interface Homme- Machine

Simulateur Logique





# Introduction

Tkinter (de l'anglais Tool kit interface) est une librairie basique mais très simple d'utilisation pour construire rapidement des interfaces graphiques avec Python.

Elle nous sera très utile dans la création de notre IHM.

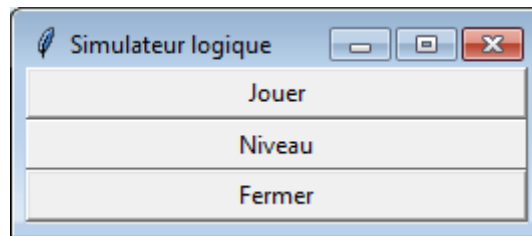
## **Objectif : réalisation d'une application de simulation de circuits logiques**

Le but de ce projet est la conception et la réalisation d'une application d'initiation aux circuits logiques dans un environnement ludique puisqu'il s'agit de permettre à un petit robot de faire le trajet d'un point A à un point B dans différentes configurations.

L'application doit permettre :

- de construire le robot,
- de créer l'environnement (et de le sauvegarder),
- de lancer la simulation (et de la suspendre),
- d'afficher diverses informations.

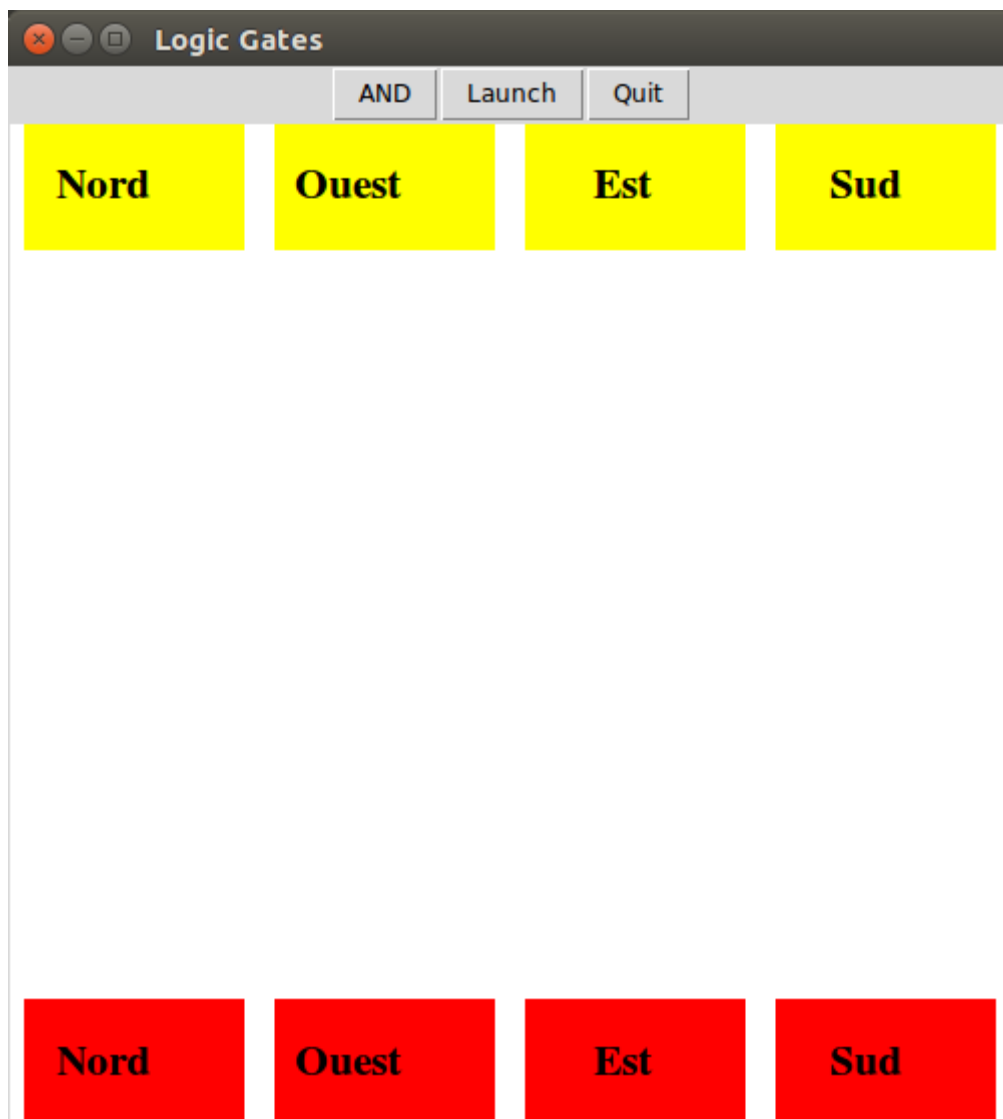
# Manuel utilisateur de l'application



**Pour lancer l'application : exécuter  
main.py**

Pour Jouer (BERENGUER) :

Cliquez sur « Jouer » => lance la simulation



Les modules jaunes en haut sont les capteurs (entrées) et ceux rouges en bas sont les moteurs (sorties) du robot.

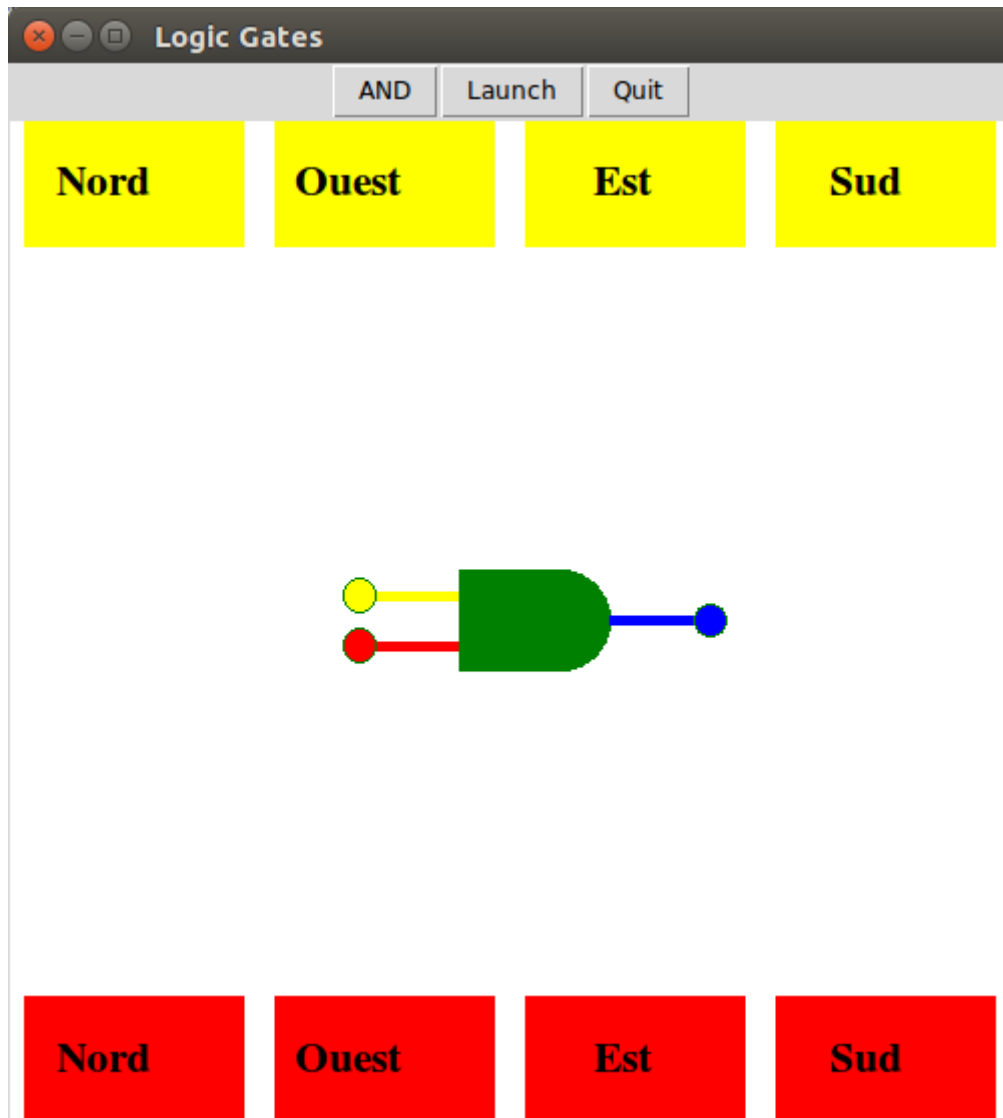
L'affichage n'est pas complet, car nous n'avons pas pu combiner cet écran avec celui du niveau pour faire bouger le robot.

Aussi, l'aspect interface n'a pas été poussé au maximum du fait de la concentration première dans le fonctionnement du projet, qui n'est pas arrivé au bout malheureusement.

Nous avons donc seulement trois boutons : AND pour créer une porte logique « et », Launch pour lancer la simulation, et Quit pour quitter.

#### Bouton AND :

Crée une porte logique « et » au centre du canvas :



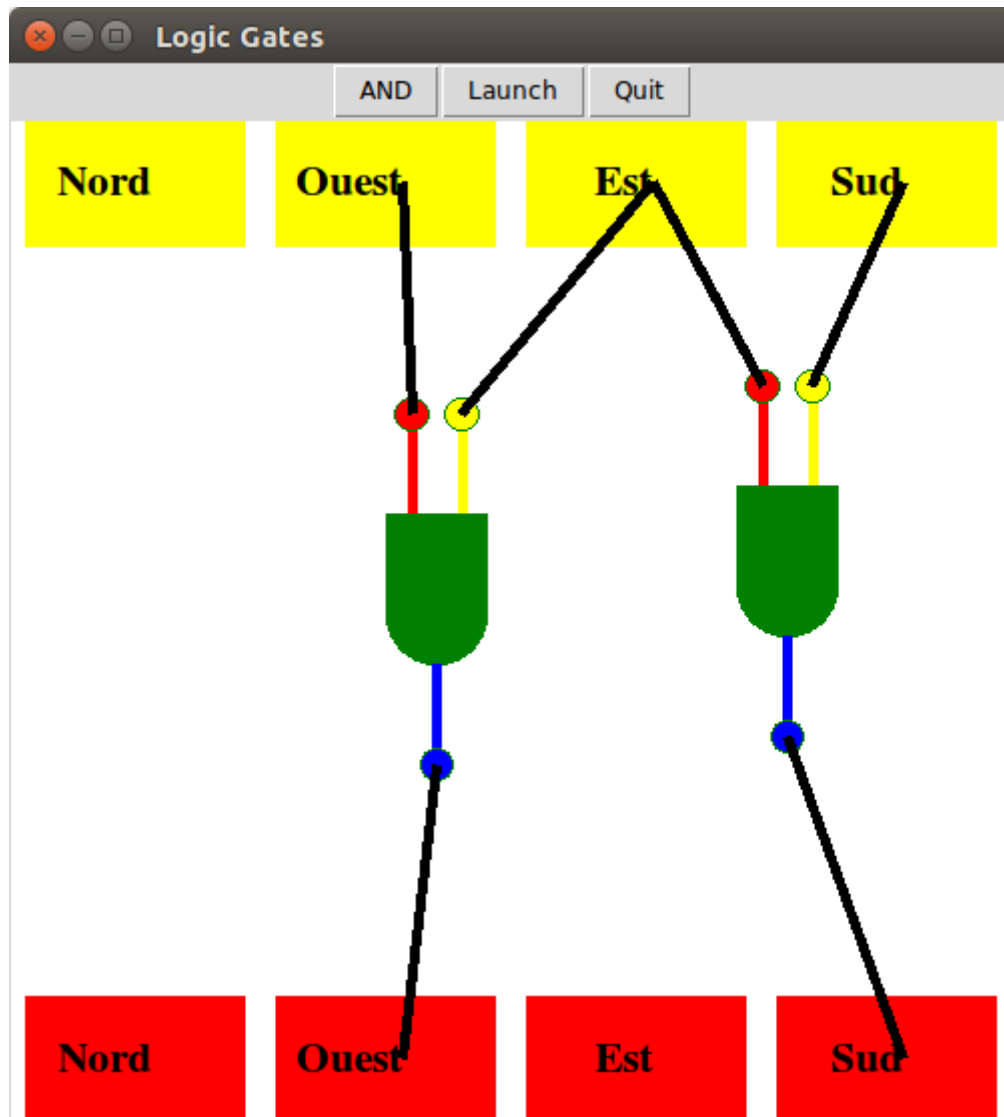
Les autres portes logiques n'ont pas été implantées par manque de temps.

Cette porte logique peut être bougée avec le click gauche et en bougeant la souris, et être pivoté à 90° en faisant un click droit.

La porte logique comporte trois connecteurs. Chacun d'entre-eux peut être relié via un fil à un capteur, un moteur, ou à un autre connecteur de la même porte logique ou d'une autre.

Ces fils bougent correctement selon la rotation ou le mouvement des portes logiques. Aussi, il peut y avoir des liaisons multiples, plusieurs liens peuvent se situer sur un seul connecteur.

Exemple :

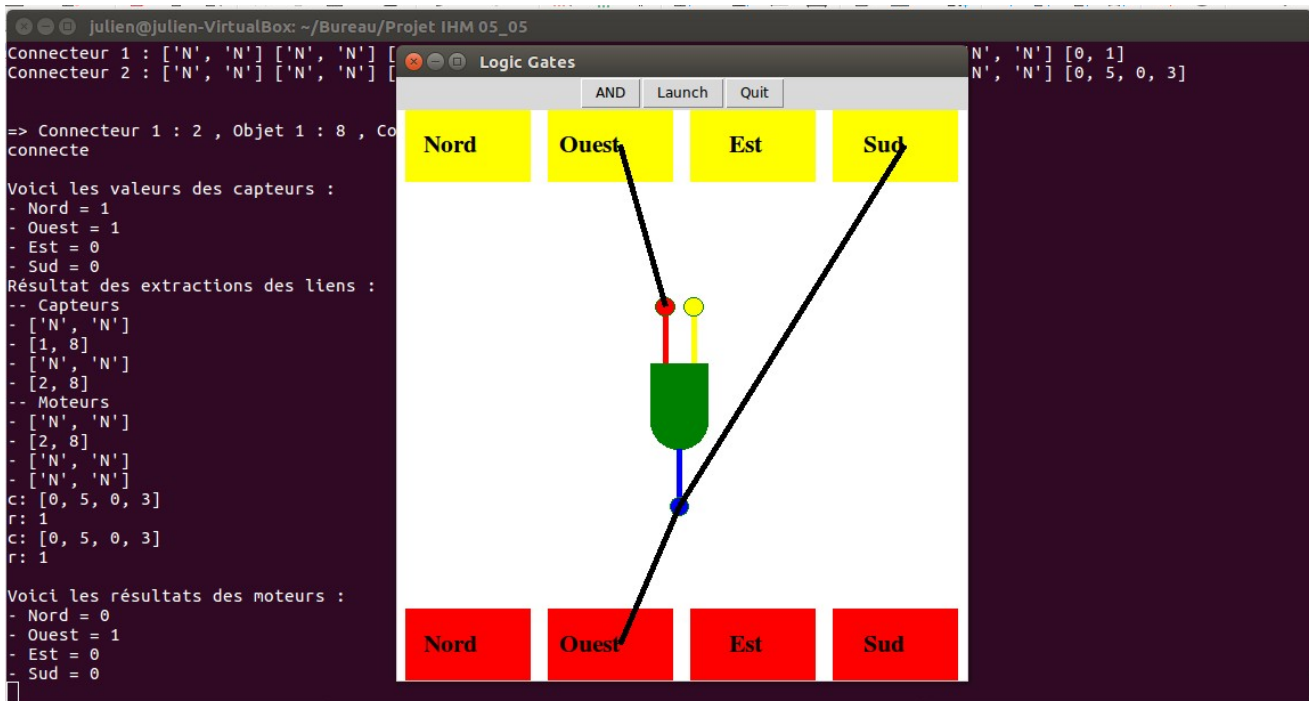


Ces fils sont des liens qui permettent de gérer la matrice (voir 'Bouton Launch').

Bouton Launch :

Permet de lancer la simulation :





Cette simulation fonctionne en liant les moteurs à des portes logiques, liées aux capteurs, ou directement à ceux-ci. Cela nous permet de transmettre la valeur des capteurs (1 ou 0) aux portes logiques, qui transformeront le signal jusqu'aux moteurs, qui ordonnent au robot de bouger.

Pour cela, nous utilisons deux modules qui gèrent une matrice : un qui la décrit, et l'autre qui l'utilise selon nos besoins. Cette matrice est composée de trois lignes, représentant les trois connecteurs. Chaque colonne représente un objet.

Chaque case est une liste de deux entiers, le premier représentant la ligne, le second la colonne. Lorsqu'un lien est effectué, la case du premier connecteur prend les coordonnées du second et inversement. Ceci nous permet de tous les relier entre-eux lors du lancement de la simulation. Aucune fonction ne permet après de supprimer un objet, donc cette gestion d'effacement dans la matrice n'est pas effective.

Nos cases gèrent les liens simples, mais aussi les liens multiples, chaque liste dans une case peut être augmentée.

A l'initialisation, la matrice est créée avec quatre capteurs et quatre moteurs. On a donc au début huit colonnes et les trois lignes, dont une seule est réellement utilisée vu que ces objets ne sont composés que d'un seul connecteur.

Cependant il y a de nombreux problèmes.

Tout d'abord la gestion de la matrice fonctionne, mais la fonction récursive pour récupérer la valeur des capteurs via les liens n'est pas fameuse et peut mener à des erreurs. Actuellement, le meilleur moyen sûr de recevoir une valeur est un lien direct.

Aussi, lors d'un lien avec une porte logique au niveau de sa sortie, il faut récupérer le tag qui donne le type de la porte logique. Ce n'était pas prévu, et un 'include' du fichier 'jouer' pour qu'il renvoie le tag selon l'identifiant n'était pas possible, car ce fichier incluse justement le fichier qui gère la matrice. La solution n'a pas été trouvée.

Du fait que nous n'avons pas fusionné la partie portes logiques et celle du niveau dans le menu 'jouer', nous ne pouvons pas prendre les valeurs des capteurs (toutes mises à 1 par défaut dans le programme), ni à faire avancer le robot avec les valeurs de sorties.

Pour ce faire, nous aurions dû aussi modifier la partie du niveau pour y inclure des fonctions renvoyant la valeur des capteurs et celles permettant d'agir sur le robot. Aussi, il devrait y avoir la gestion du temps : avancer d'une case par seconde. Un test a été fait là-dessus ('play pause'), fourni avec le programme.

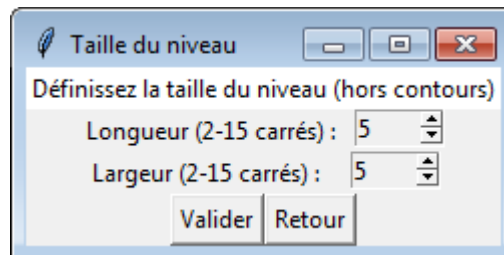
### Bouton Quit :

Dans les premières versions, fermait la fenêtre. Couplé au menu principal, fait revenir l'utilisateur à celui-ci.

Pour créer un niveau (MATTIOLI) :

Cliquez sur « Niveau » => « Créer »

Vous arrivez à cette fenêtre :

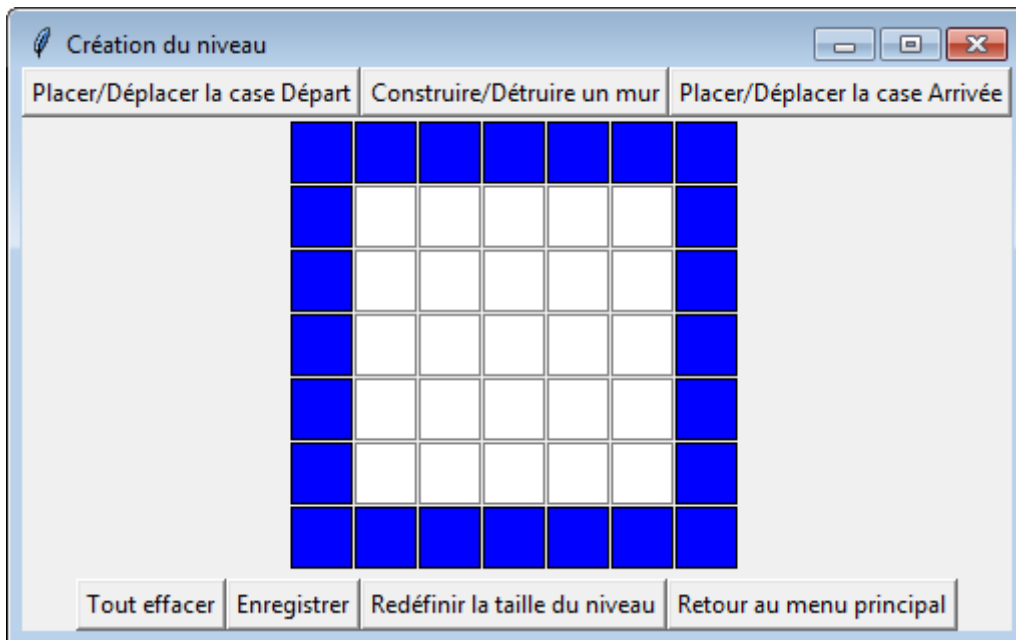


Ici, vous choisissez la taille de votre niveau (entre 2x2 et 15x15).

Cliquez sur « Valider » pour confirmer votre choix.

Problème obtenu : après avoir implanté le module Jouer, les spinboxs de la fenêtre ont commencé à retenir de mauvais entiers. Du coup, en appuyant sur la fleche haute ou la fleche basse de chaque spinbox, la longueur ou la largeur sera incrementé de 1 et ne pourra pas dépasser 15.

Vous arrivez maintenant à cette fenêtre ci-dessous :

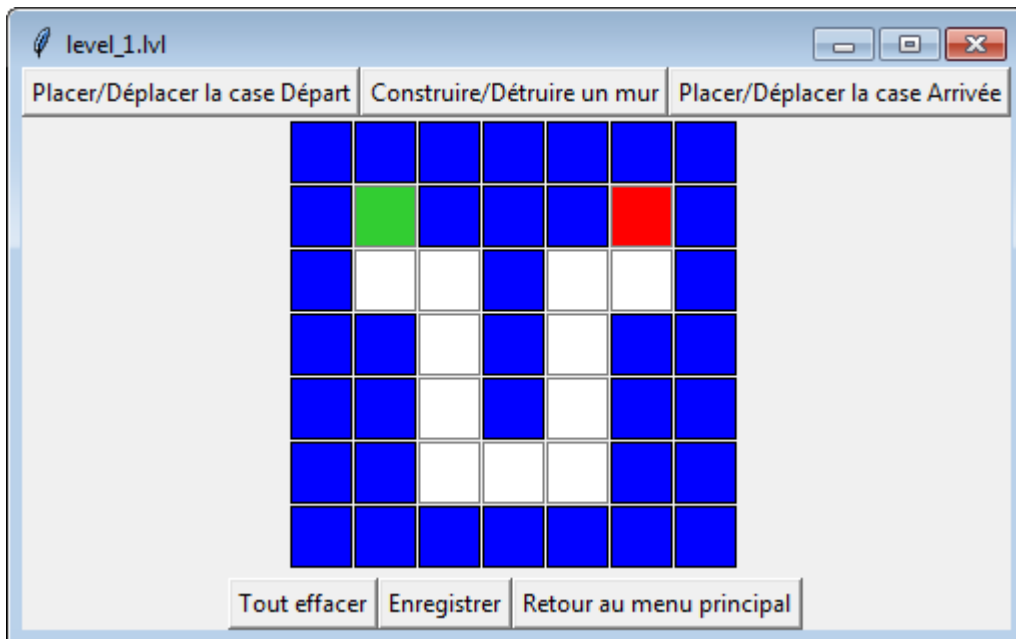


- Pour placer/déplacer la case départ, cliquez sur le bouton du même nom, puis sur un des carrés blanc du niveau.
- Même chose pour la case arrivée.
- Pour construire un mur, cliquez sur le bouton « Construire/Détruire un mur », puis sur un des carrés blanc du niveau.
- Pour détruire un mur, cliquez sur le même bouton, puis sur un mur déjà construit.
- En cliquant sur « Tout effacer », vous remettrez le niveau à zéro (tous les carrés modifiables redeviendront blancs)
- Pour sauvegarder votre niveau, cliquez sur « Enregistrer ».
- Cliquer sur « Redéfinir la taille du niveau » vous fera revenir à la fenêtre précédente
- Le bouton « Retour au menu principal » permet comme son nom l'indique de retourner au menu principal

#### Pour modifier un niveau :

Cliquez sur « Niveau » => « Modifier »  
 => Choisissez un fichier (d'extension .lvl)

Vous arrivez ici :



Les boutons sont les mêmes que précédemment.

Cliquez sur « Enregistrer » pour sauvegarder pour vos modifications.

Pour fermer l'application :

Cliquez sur « Fermer »

# Problèmes rencontrés