Automates et dédales

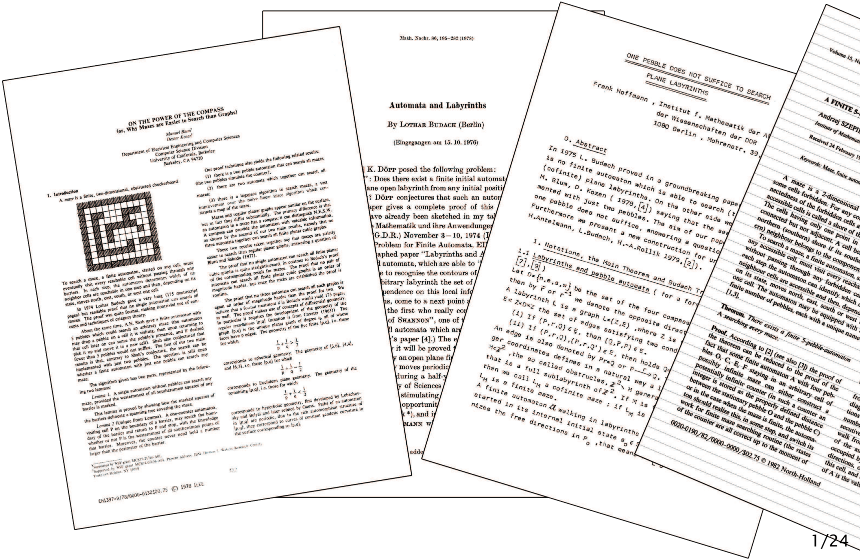
Nicolas Ollinger (LIFO, Université d'Orléans, France)

MeJ, Bourges — 22 février 2018

Claude Shannon (1916-2001)

Dans cet exposé

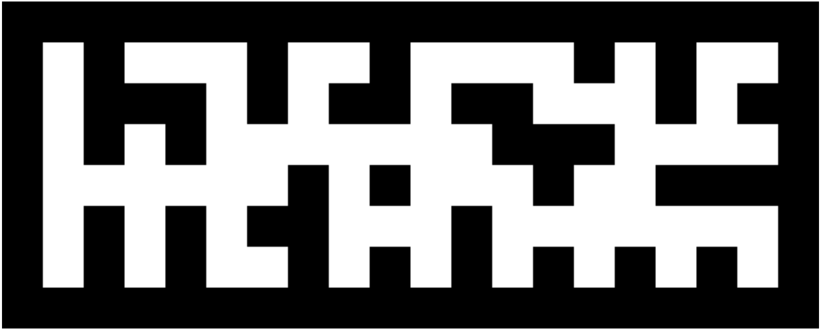
Quelques théorèmes autour de l'**exploration** de **dédales** par des **automates finis**.



Dédale

Dans cet exposé un dédale est une grille carrée de cases vides et de murs. Comment le définir **rigoureusement** ?

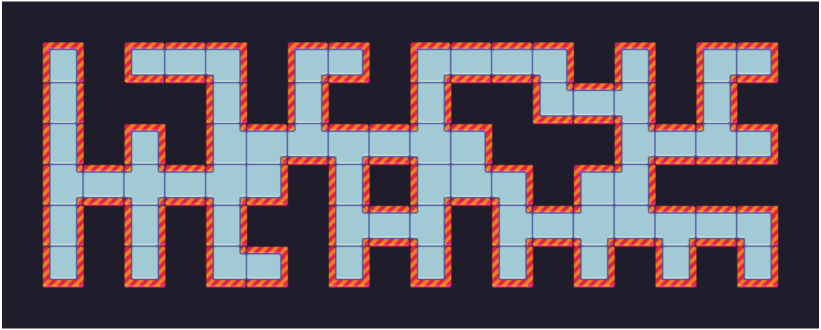
Définition Un **dédale** est une application $D : \mathbb{Z}^2 \to \{0, 1\}$ telle que $D(0) = 0$ et $D^{-1}(0)$ est **fini** et **simplement connexe**.



Dédale

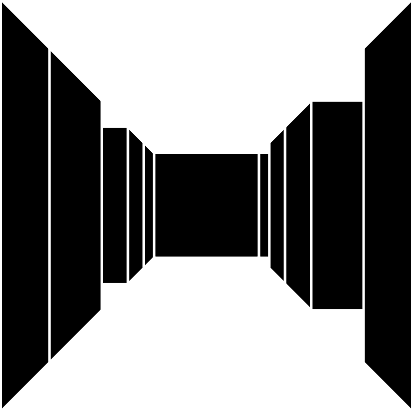
Dans cet exposé un dédale est une grille carrée de cases vides et de murs. Comment le définir **rigoureusement** ?

Définition Un **dédale** est une application $D : \mathbb{Z}^2 \to \{0, 1\}$ telle que $D(0) = 0$ et $D^{-1}(0)$ est **fini** et **simplement connexe**.



Exploration

Explorer un dédale c'est parcourir chacune des cases vides du dédale, en se déplaçant de case vide en case vide voisine.



Lors de l'exploration, on peut observer la case courante ainsi que les quatre cases voisines (N, S, E, O).

Automate fini

Un **automate** d'états **finis** choisit ses déplacements à partir de l'observation de son environnement et d'un état interne qui prend sa valeur dans un ensemble fini.

Q est l'ensemble fini des **états** de l'automate et
$\delta : Q \times \{0,1\}^{\boxplus} \to Q \times \boxplus$ est sa **fonction de transition**.

$$\text{avec} \quad \boxplus = \{\overset{C}{(0,0)}, \overset{N}{(0,1)}, \overset{S}{(0,-1)}, \overset{E}{(1,0)}, \overset{O}{(-1,0)}\}$$

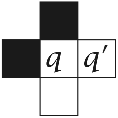
Transition

Une **transition** de l'automate est une règle qui définit son comportement.

Ainsi, $\delta \Big(q,$  $\Big) = (q', E)$ signifie

«dans l'état q, s'il y a des murs uniquement au nord et à l'ouest, alors passer dans l'état q' et se déplacer vers l'est. »

Notation compacte :

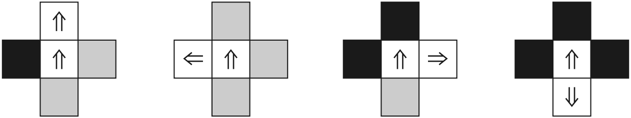


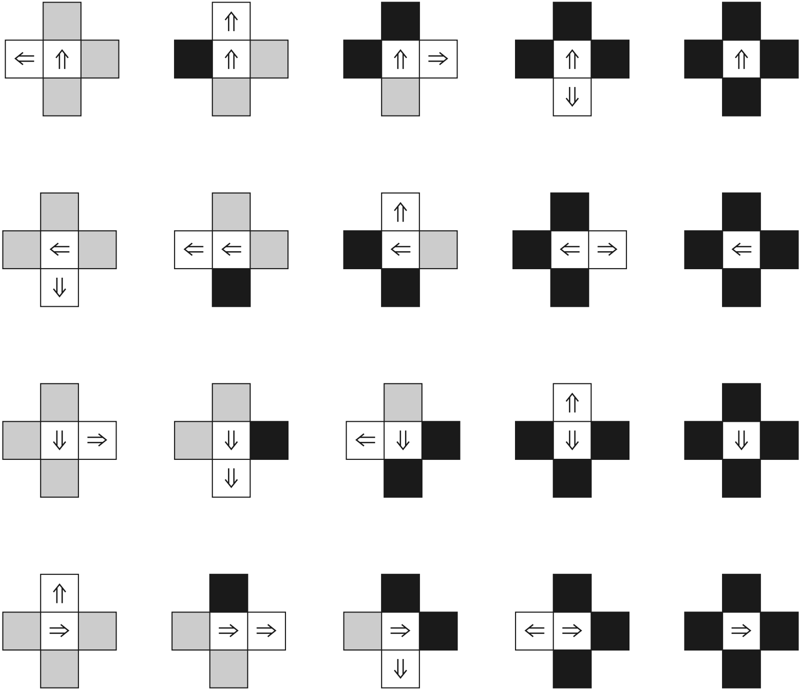
Longer le mur à main gauche

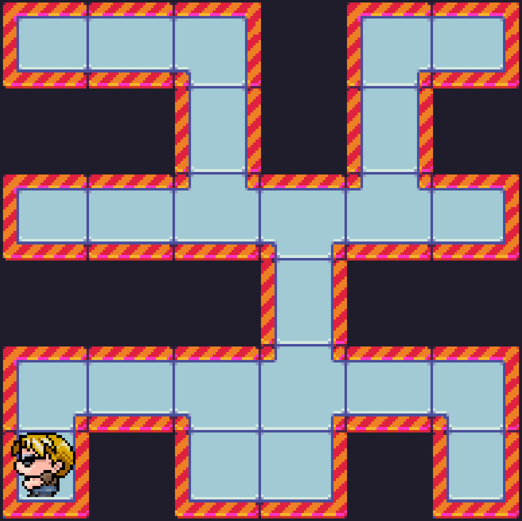
Première tentative d'exploration : construisons un automate qui *« garde la main gauche sur le mur »*.

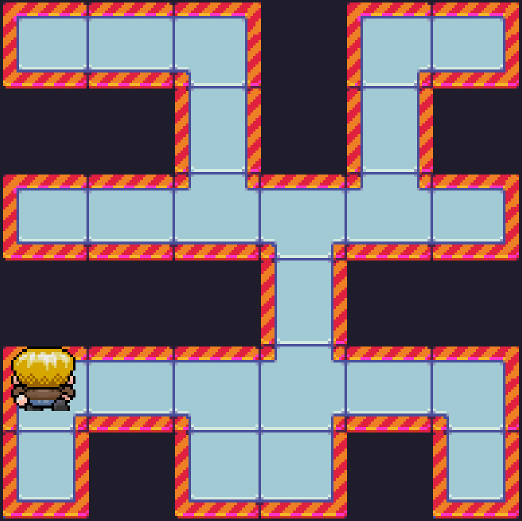
Q mémorise le côté traversé pour sortir de la case précédente, afin d'identifier le mur suivi.

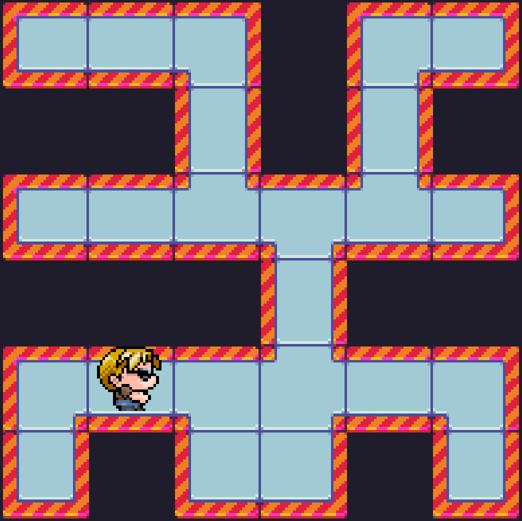
$$Q = \{\Uparrow, \Downarrow, \Leftarrow, \Rightarrow\}$$

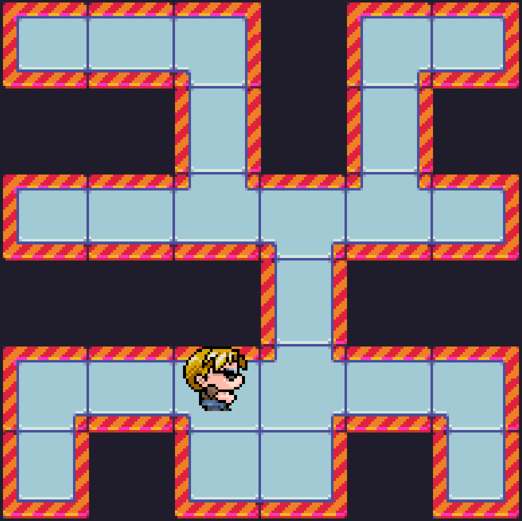


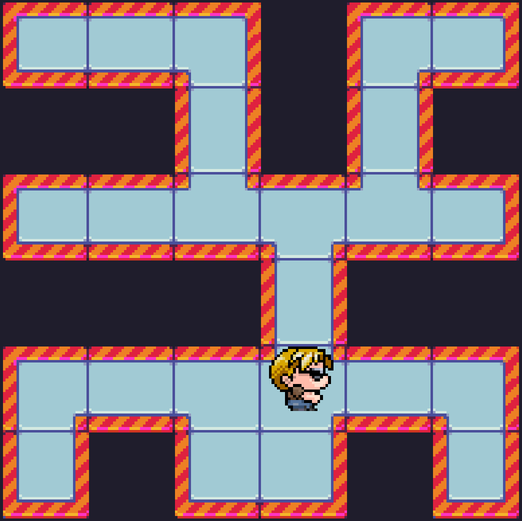


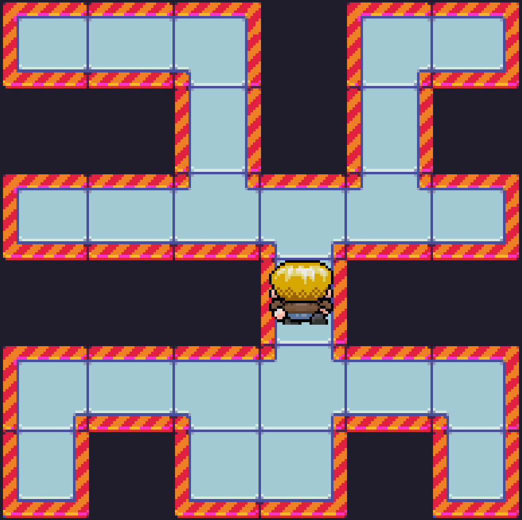


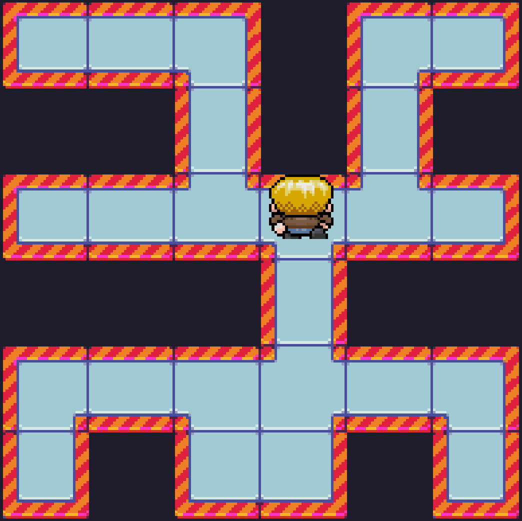


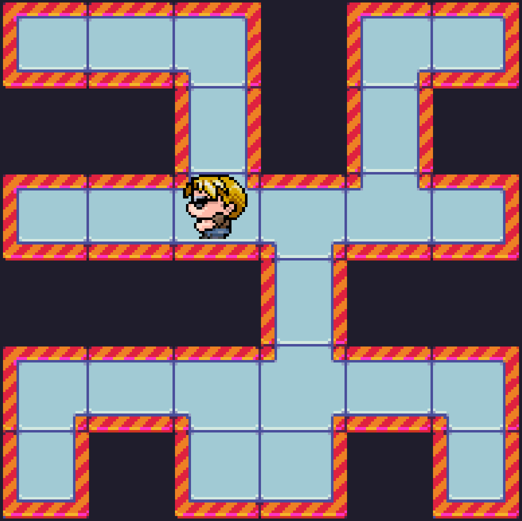


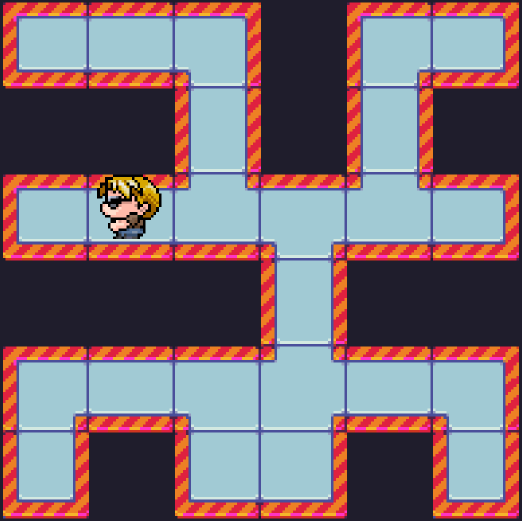


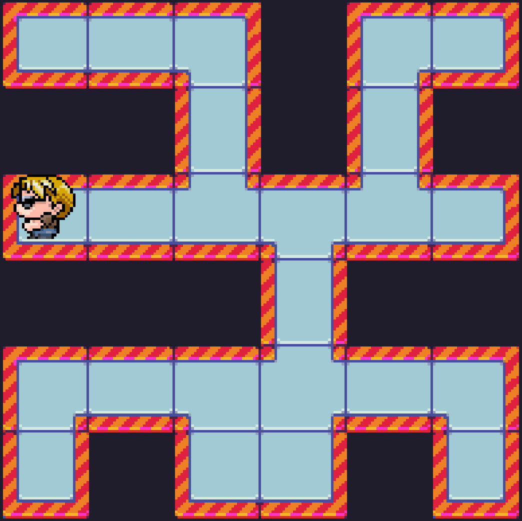


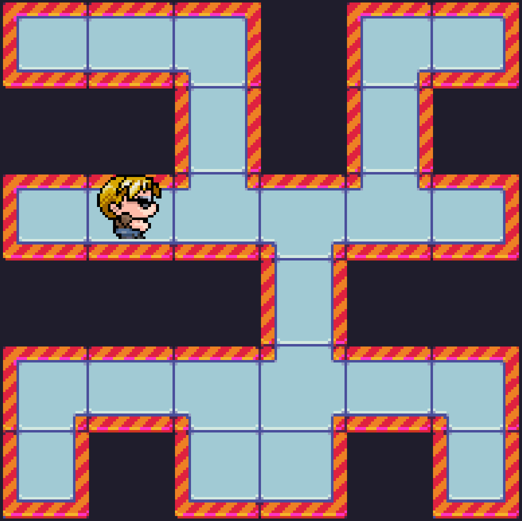


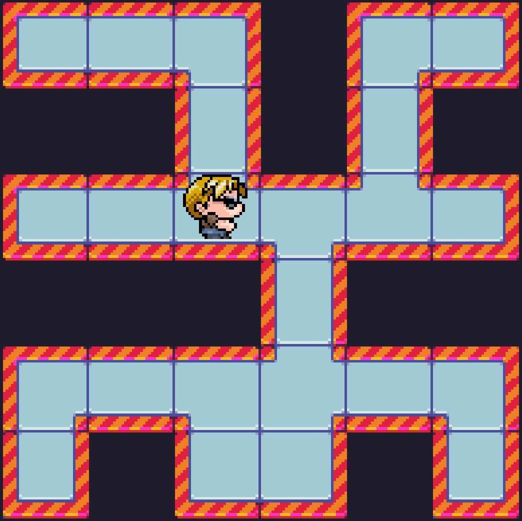


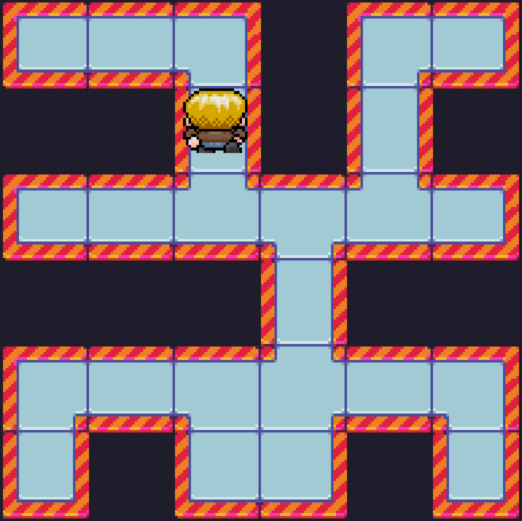


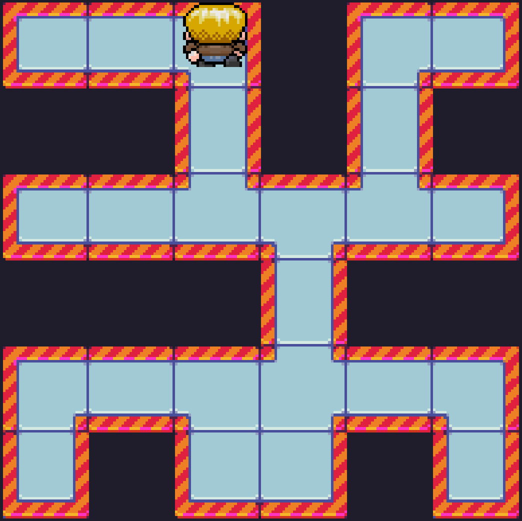


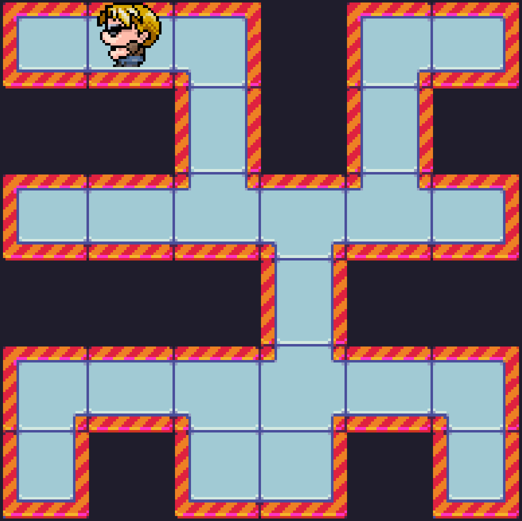


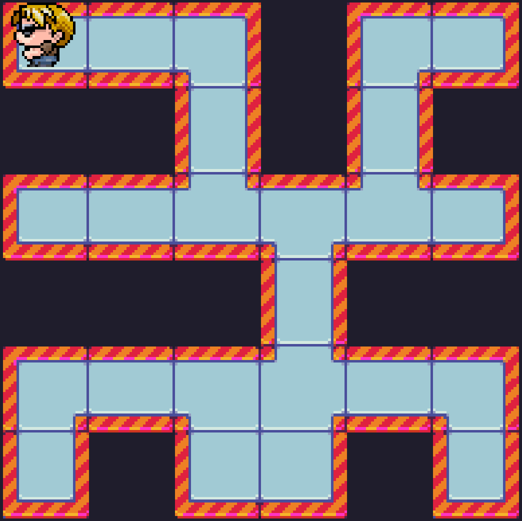


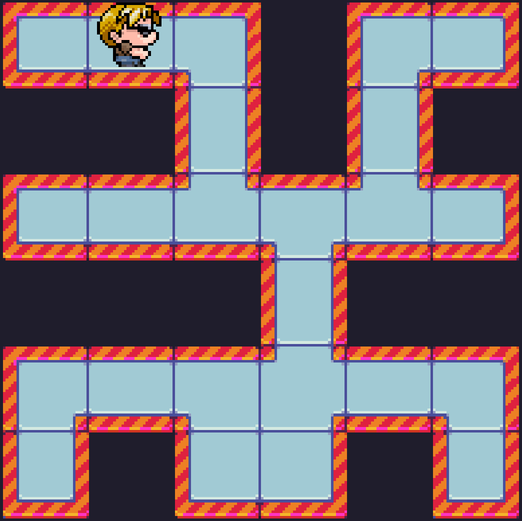


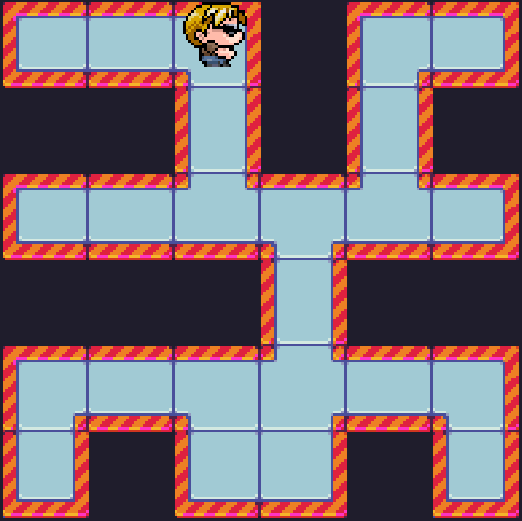


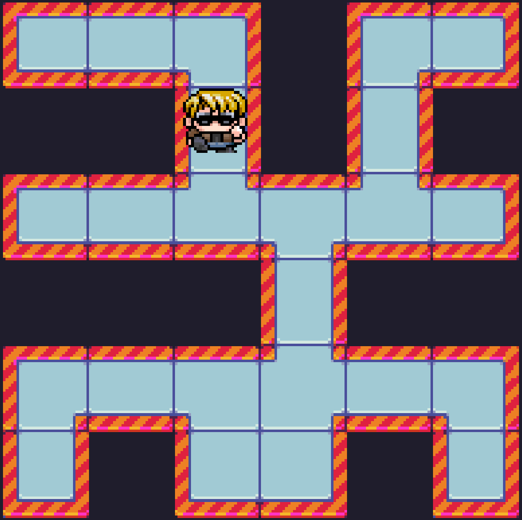


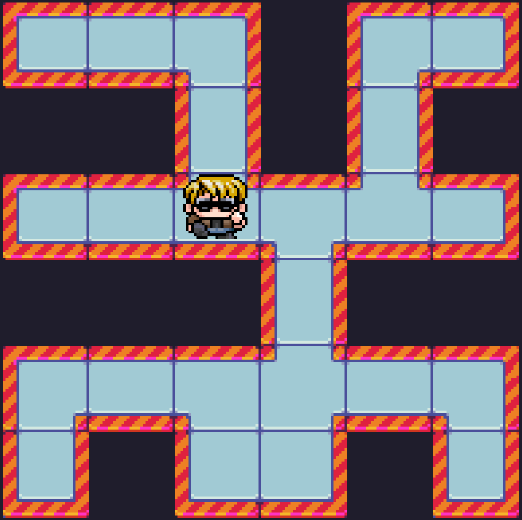


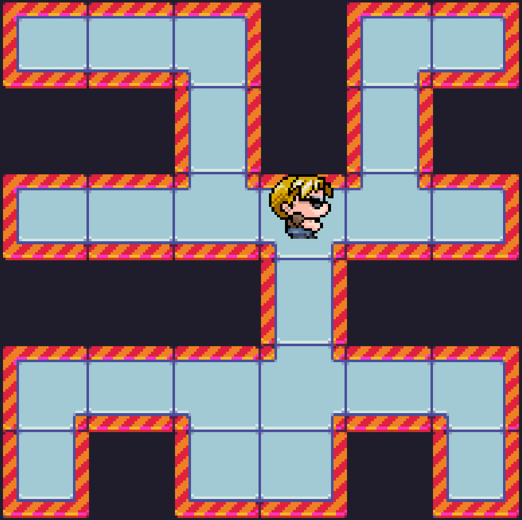


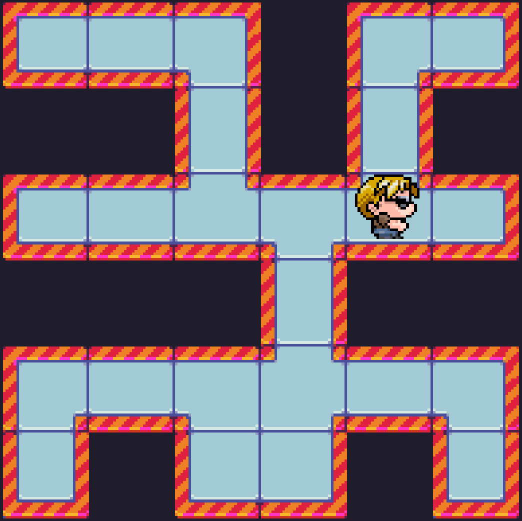


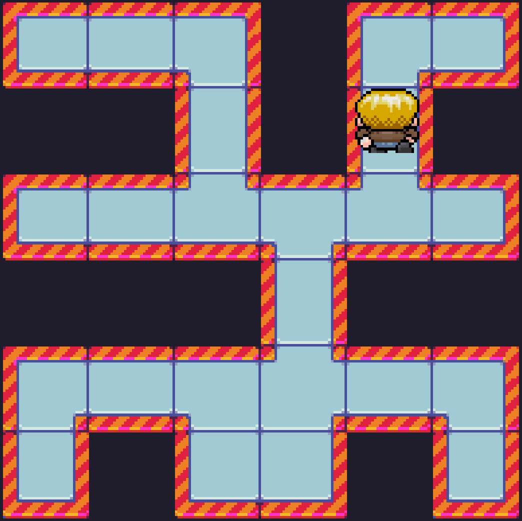


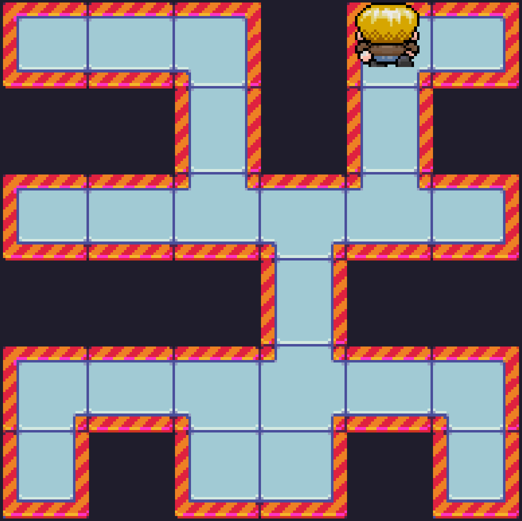


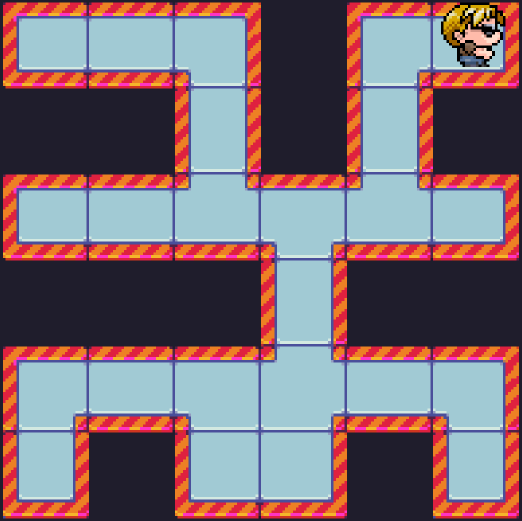


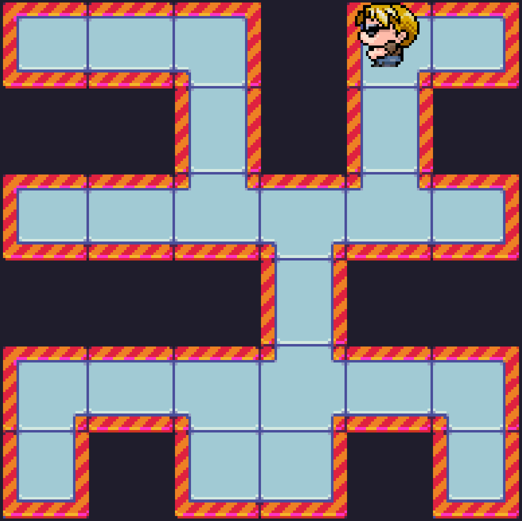


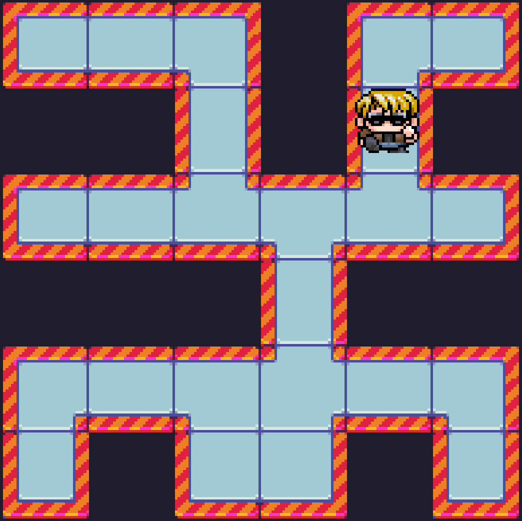


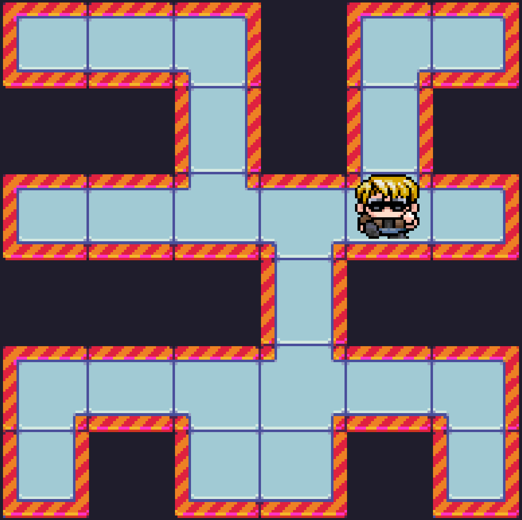


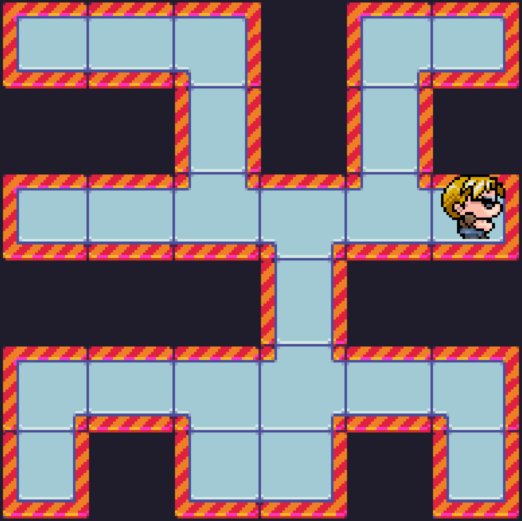


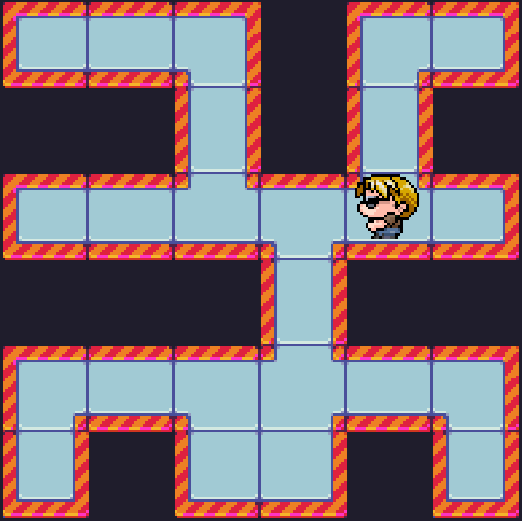


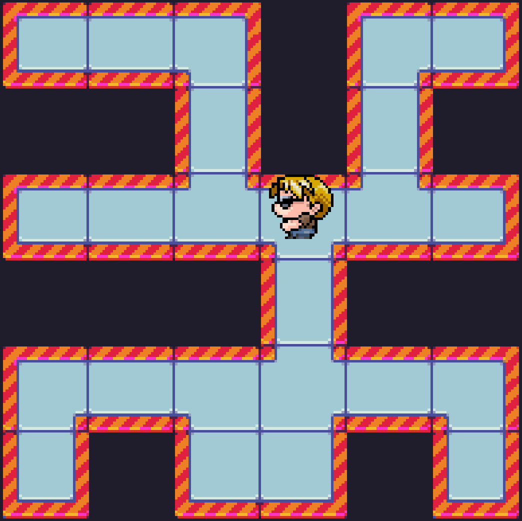


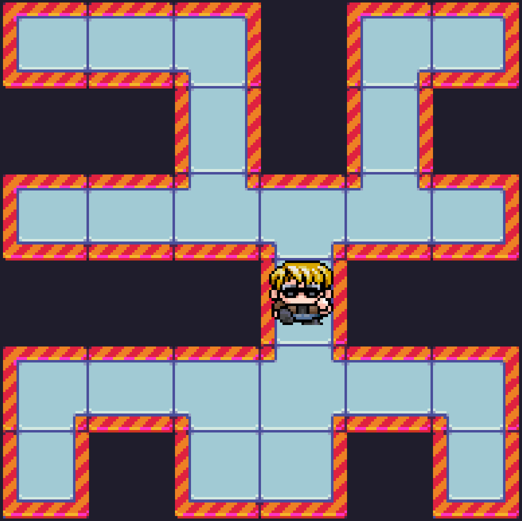


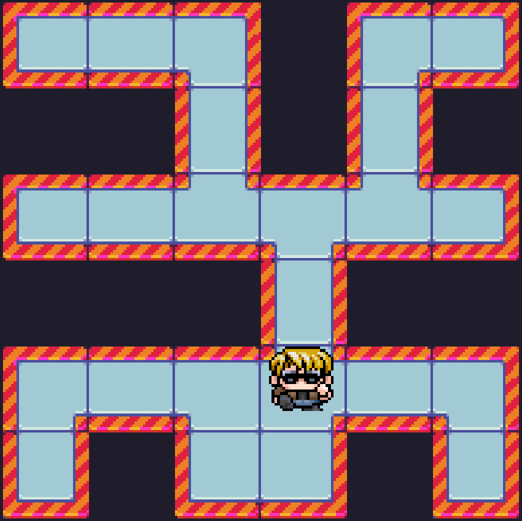


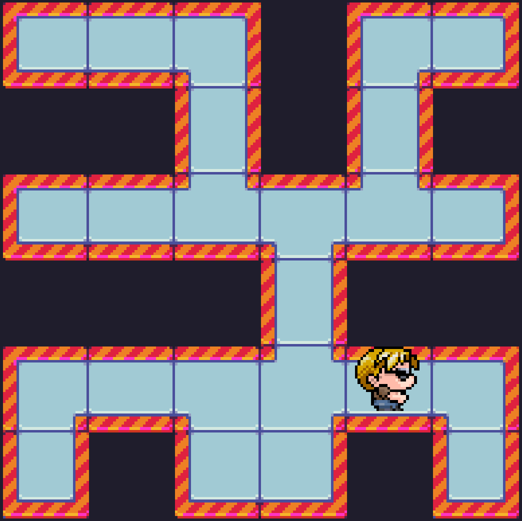


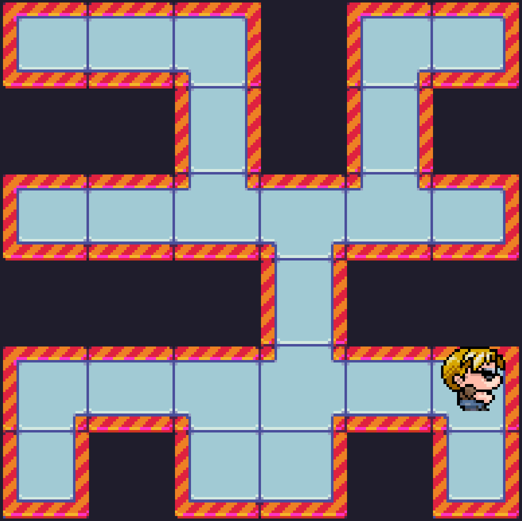


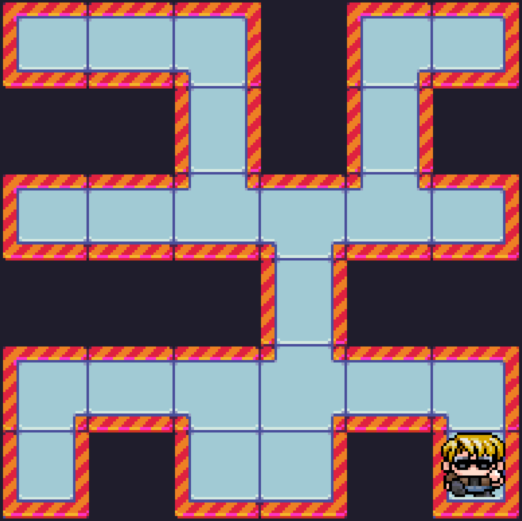


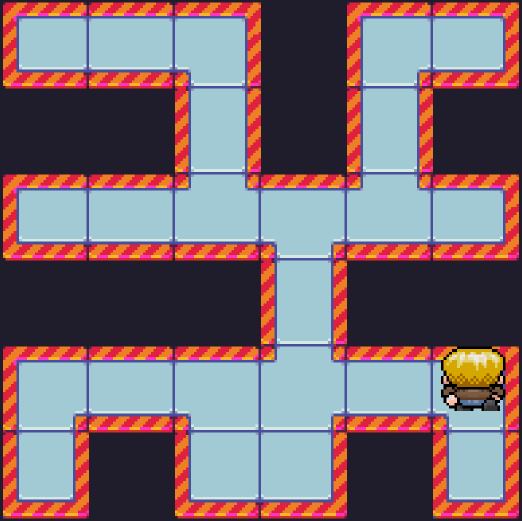


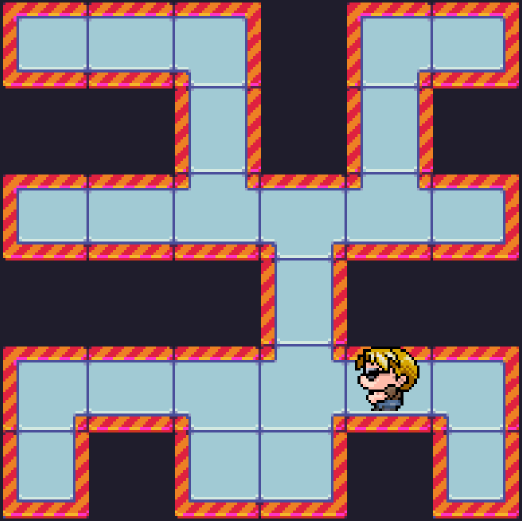


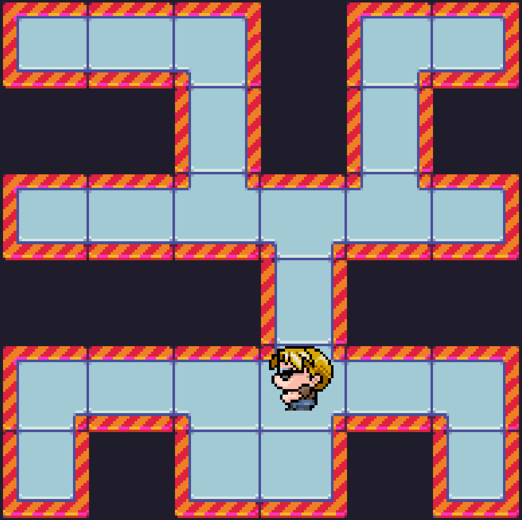


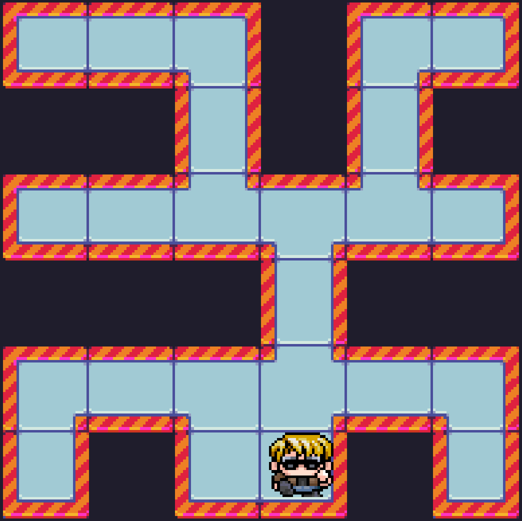


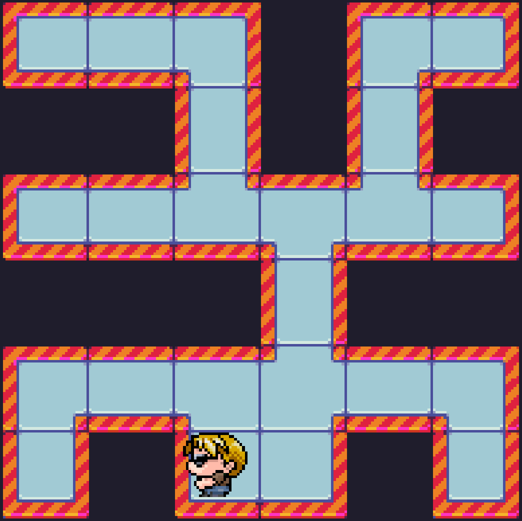


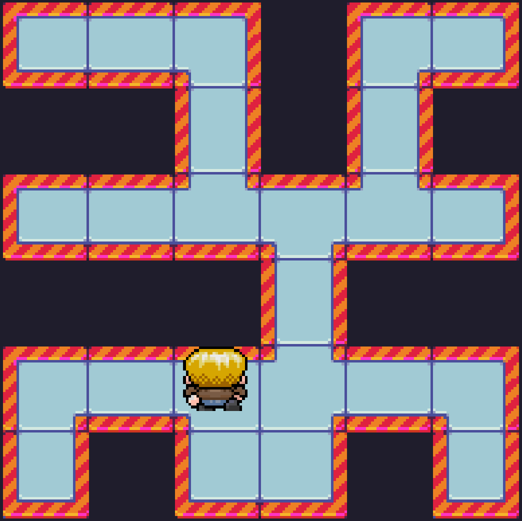


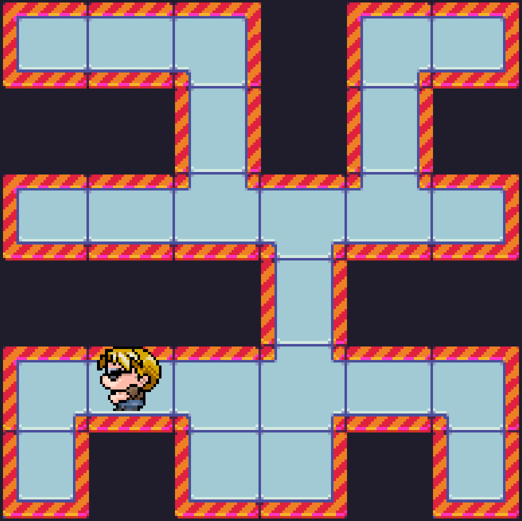


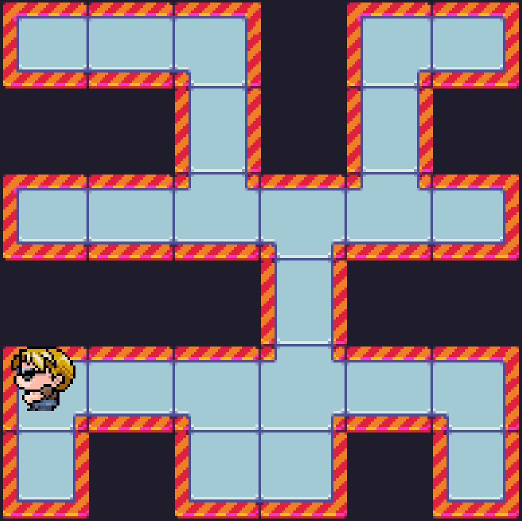


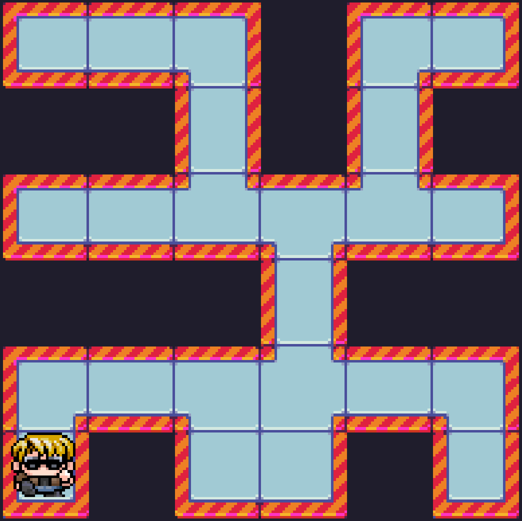










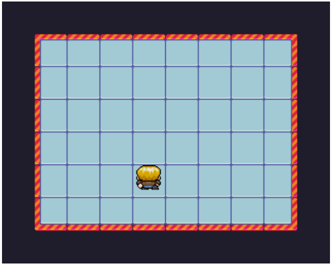


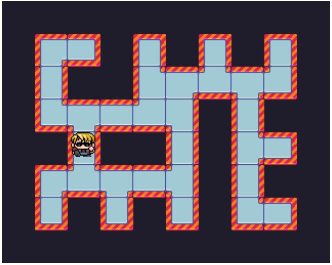
Limites de cet automate

Cet automate fini n'explore pas tous les dédales.

① Le dédale doit avoir des couloirs fins.

② Les murs doivent être eux-même simplement connexes.

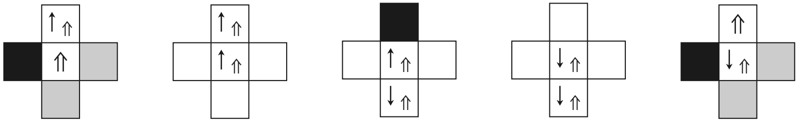




Lâcher le mur de temps en temps

Pour résoudre le problème ①, on ajoute un **balayage vertical** entre chaque transition.

$$Q' = Q \cup Q \times \{\uparrow, \downarrow\}$$



Proposition Il existe un automate fini qui explore tous les dédales dont les murs sont simplement connexes.

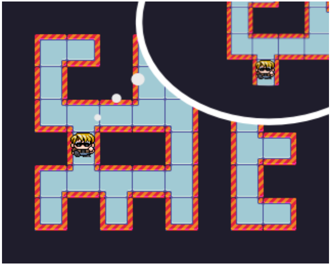
Et pour le problème 2 ?

Théorème [Budach 1974] Pour tout automate fini il existe un dédale **piège** qui n'est pas exploré par cet automate.

Si nous voulons être capables d'explorer n'importe quel dédale avec un unique automate, il faut augmenter ses capacités.

Quel **gadget** raisonnable, le plus simple possible, peut-on ajouter aux automates finis pour rendre l'exploration possible ?

Avec une mémoire



Avec une mémoire 2D sur laquelle se déplace une tête de lecture/écriture, l'exploration devient très facile.

Hélas, cette mémoire donnerait à notre automate le pouvoir de calcul d'une **machine de Turing**. Il pourrait exécuter **n'importe quel algorithme**.

Avec de la peinture ou avec une corde

En permettant à l'automate de modifier le dédale avec de la peinture (sur les cases vides), ou encore avec une corde que l'automate déroule derrière lui dans le dédale, on peut effectuer un parcours arborescent du dédale.

Cette technique nécessite une quantité de peinture ou de corde qui **dépend de la taille du dédale à explorer**.

Peut-on faire mieux ?

Avec des galets

On confie des galets à l'automate, qu'il peut déposer et ramasser sur le sol des cases du dédale.

$$\delta : Q \times \{0,1\}^{\boxplus} \times \mathbb{N}^2 \to Q \times \boxplus \times \mathbb{Z}$$

Dès lors, $\delta\left(q, \boxplus, 2, 0\right) = (q', E, 1)$ signifie

«dans l'état q, s'il y a des murs uniquement au nord et à l'ouest et qu'il y a 2 galets dans ma poche et 0 galet au sol, alors transférer 1 galet de la poche au sol, passer dans l'état q' et se déplacer vers l'est. »

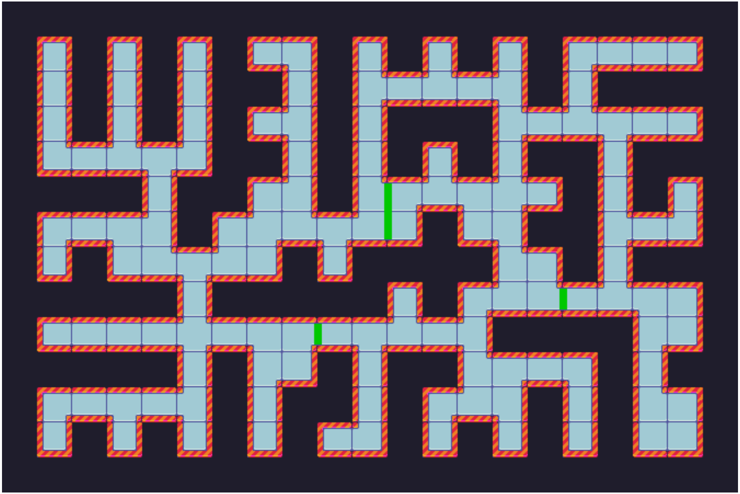
Que peut-on faire avec des galets ?

Théorème [Hoffman 1981] Pour tout automate à 1 galet il existe un dédale **piège** qui n'est pas exploré par cet automate.

Théorème [Blum et Kozen 1978] Il existe un automate à 2 galets qui peut explorer n'importe quel dédale et s'arrête.

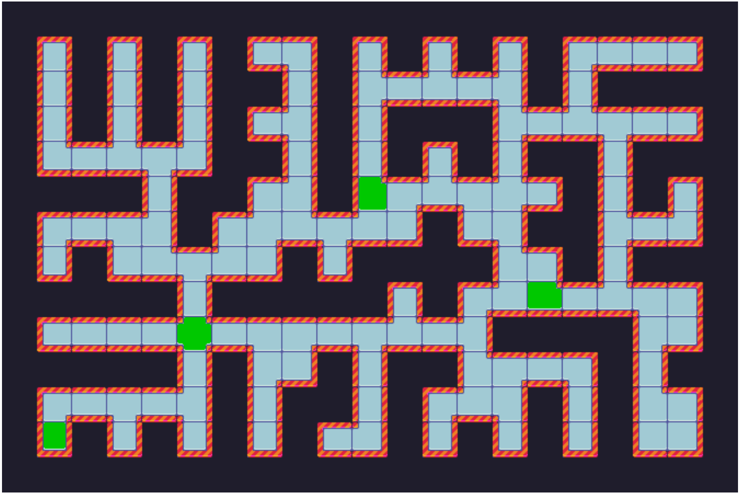
Nous allons esquisser une démonstration de ce théorème.

Murs virtuels



Ajouter des murs virtuels pour retrouver un dédale avec un unique bord à longer.

Cases vertes



Une case est **verte** si elle est adjacente au sommet le plus à gauche parmi les sommets les plus bas du contour d'une composante connexe de murs.

Automates et cases vertes

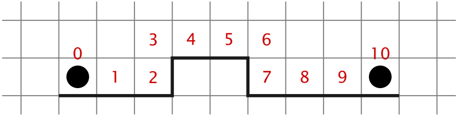
Proposition Il existe un automate fini qui explore n'importe quel dédale dont on a marqué les cases vertes.

$$\delta : Q \times \{0, 1, 2\}^{\boxplus} \to Q \times \boxplus \times \mathbb{Z}$$

Comment faire pour **reconnaître** les cases vertes à l'aide de 2 galets ?

Compter avec deux galets

En espaçant deux galets le long d'un contour, on peut compter.



Un automate fini à deux galets peut :

- déplacer le compteur le long du bord ;
- incrémenter et décrémenter le compteur ;
- tester si le compteur vaut zéro.

Automate fini à compteur

Un automate à compteur est un automate fini muni d'un compteur entier.

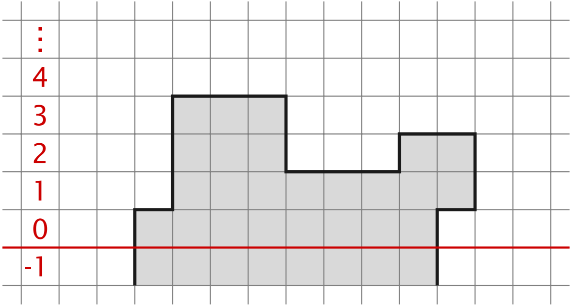
$$\delta : Q \times \{0,1\}^{\boxplus} \times \{Z,P\} \to Q \times \boxplus \times \{-1,0,1\} \times \mathbb{Z}$$

Proposition Il existe un automate à compteur qui reconnaît les cases vertes.

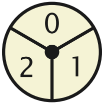
Remarque Attention, le long du bord nos compteurs sont bornés par le périmètre.

Altimètre

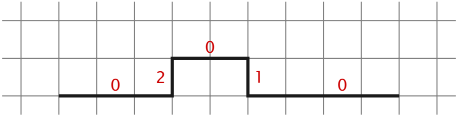
L'automate utilise son compteur pour mémoriser l'altitude atteinte depuis le point de départ de sa reconnaissance.



Compte-tour

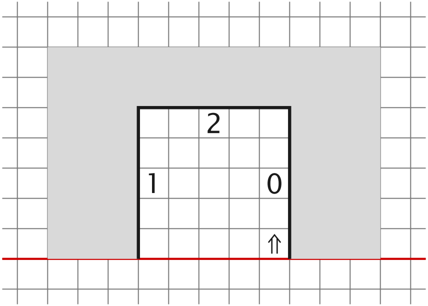


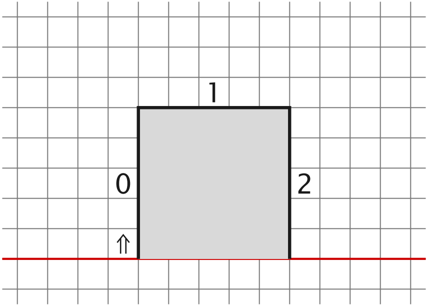
L'automate mémorise de plus l'index de rotation (le nombre de virage à droite moins le nombre de virage à gauche), modulo 3.

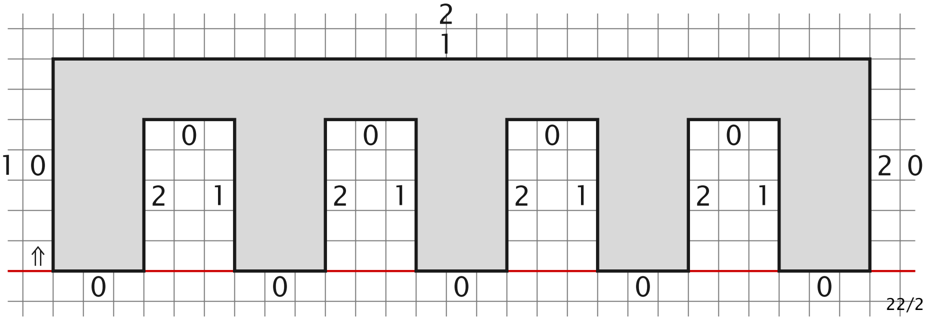


Théorème [Hopf-Riemann] L'index de rotation d'un chemin simplement clos dans le plan vaut $\pm 2\pi$ selon que la zone à main gauche est bornée ou non.

Détecter les cases vertes







Récapitulons

avec deux galets on simule un compteur
avec un compteur on reconnaît les cases vertes
avec les cases vertes on reconnaît les murs virtuels
avec les murs virtuels on explore tout dédale

Théorème [Blum et Kozen 1978] Il existe un automate à 2 galets qui peut explorer n'importe quel dédale et s'arrête.

Remarque Son nombre d'état est *a priori* énorme et son temps de parcours, bien que polynomial, l'est aussi.

Une dernière chose pour la route

Et si on considère des dédales infinis ?

Théorème [Szepietowski 1982] Il existe un automate à **5 galets** qui peut explorer n'importe quel dédale, fini ou infini, et s'arrête après avoir exploré un dédale fini.