

# Programmation Web Avancée Côté Client

## TP n° 8

Licence Informatique 2ème année  
Université de Nice-Sophia Antipolis

### Introduction

Dans cette séance, nous reprenons le code du (TP n° 7) et nous ajoutons les fonctionnalités qui permettront aux joueurs de déplacer les tuiles.

### 1 Déplacement des tuiles

Dans cet exercice, nous réaliserons les fonctionnalités nécessaires pour permettre à l'utilisateur de déplacer les tuiles du labyrinthe. À ce but, il faudra que l'utilisateur puisse pivoter la tuile en plus pour choisir l'orientation souhaitée et qu'il puisse la saisir et la déposer sur une case externe du labyrinthe pour la pousser dans le plan de jeu. La règle du jeu impose que l'on ne puisse passer une tuile que dans les rangées (horizontales ou verticales) impaires, c'est-à-dire celles qui ne contiennent que des tuiles mobiles. Lorsque la tuile en trop est poussée dans le plan de jeu, toute la rangée doit se déplacer pour lui faire de la place et la nouvelle « tuile en trop » doit ressortir du côté opposé. Il y a, cependant, une exception : « la carte qui vient de sortir d'une rangée ne peut pas être poussée au même endroit par le joueur suivant ». Donc, il faut garder mémoire d'où vient la tuile en trop et empêcher à l'utilisateur de la pousser au même endroit.

1. Ajoutez à l'objet **Labyrinthe** une propriété pour stocker la provenance de la tuile en trop.
2. Gérez les cliques sur la tuile en trop pour faire en sorte que à chaque clique la tuile pivote de 90 deg dans le sens inverse des aiguilles d'une montre.
3. Ajoutez à l'objet **Labyrinthe** une méthode **pousser** qui prend les coordonnées  $(i, j)$  d'une case, vérifie que la case soit une case valide où l'utilisateur est autorisé à pousser la tuile en trop (en cas contraire lancer une exception) et modifie le labyrinthe selon la règle : la tuile en trop est poussée dans le plan de jeu à la case  $(i, j)$ , les tuiles de la rangée sont déplacées d'une case et la tuile qui ressort de l'autre coté devient la nouvelle tuile en trop.
4. Ajoutez à la méthode **pousser** du code pour gérer correctement le déplacement du pion : le pion se déplace avec la tuile sur laquelle il est posé ; si la tuile sort du jeu, le pion doit être placé sur la tuile qui vient d'être poussée dans le plan de jeu.

## 2 Glisser-déposer

Utilisez l'interface de programmation de HTML5 pour réaliser une fonctionnalité qui permette à l'utilisateur de glisser-déposer la tuile en trop sur la case souhaitée. Vous devez gérer les événements `ondragstart` et l'attribut `draggable` de la table contenant la tuile en trop, ainsi que les événements `ondragover` et `ondrop` des cellules du tableau qui affiche le labyrinthe.