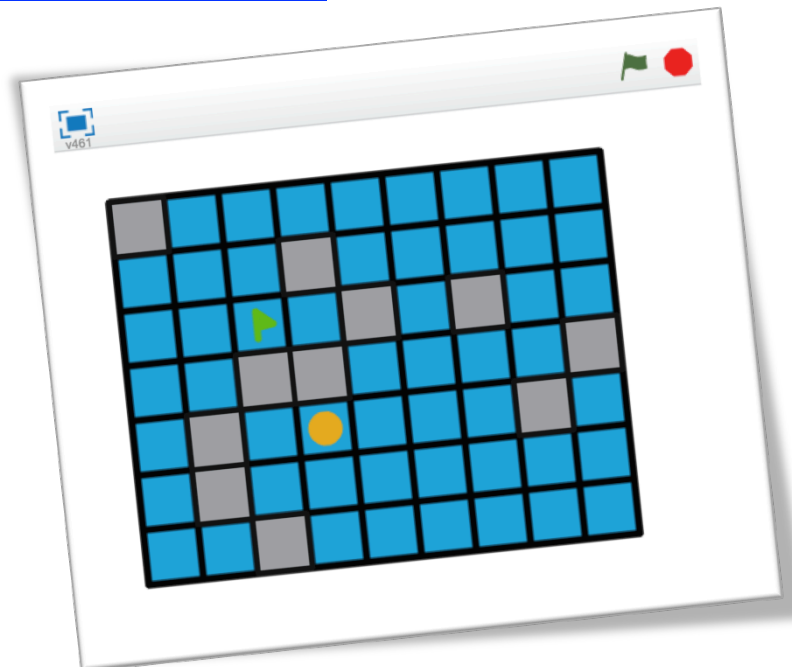


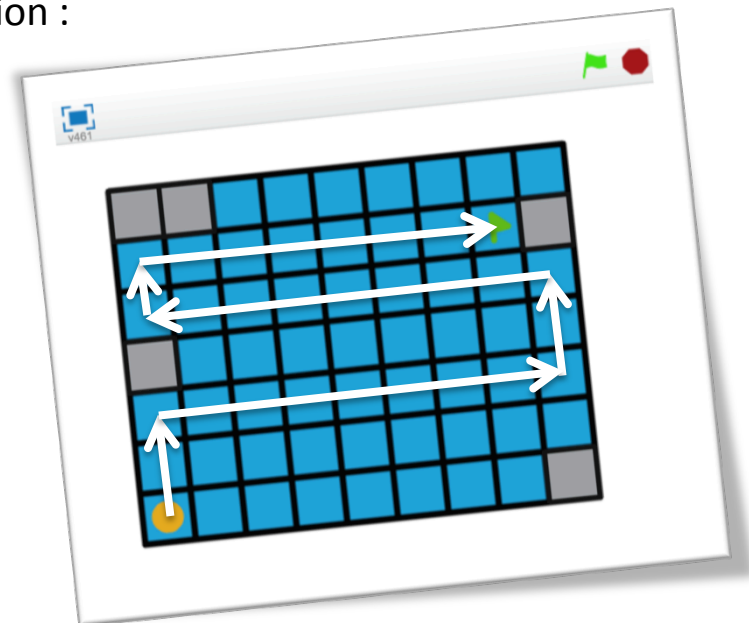
# Devoir à rendre N° 3

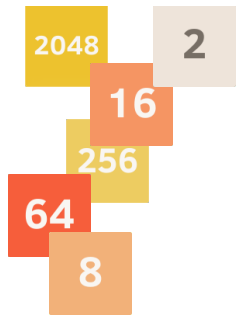
Ecrire un programme **en Processing** permettant de jouer à un jeu ressemblant au jeu « IceWalker » disponible sur le site scratch :

<https://scratch.mit.edu/projects/247985310/>



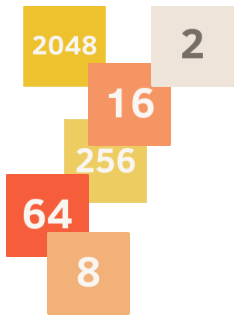
- ## Premier niveau





# Travail à faire

- L'objectif est de réaliser une interface permettant de jouer au jeu Ice Walker. Ce programme doit permettre de :
  - charger un fichier décrivant un niveau (voir diapo suivante)
  - jouer au jeu
    - Flèches directionnelles (gauche, droite, haut, bas) pour diriger la balle
    - 'R' pour redémarrer le niveau (= replacer la balle à sa position de départ)
    - vérification que la balle a atteint la position du drapeau et chargement du niveau suivant



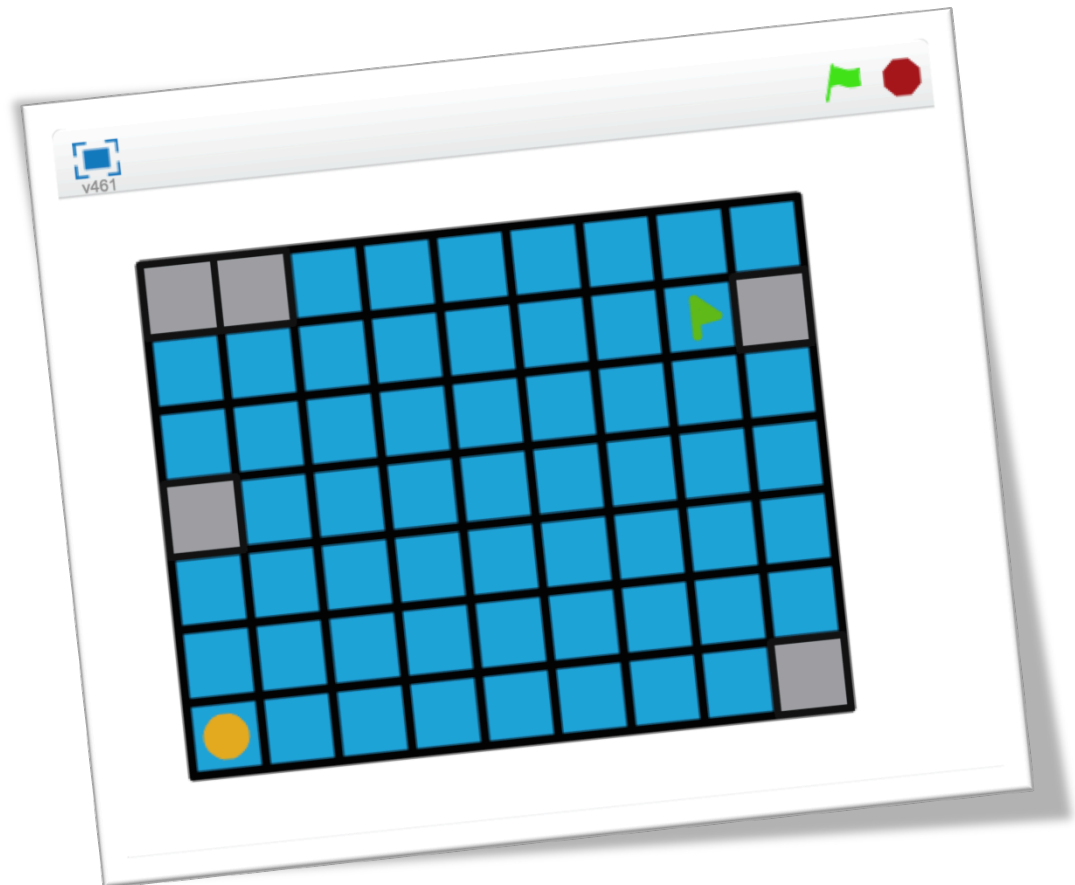
# Fichiers de niveaux

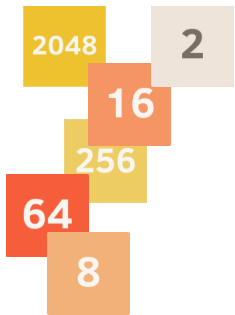
Les niveaux sont décrits dans des fichiers texte, ligne par ligne, avec le codage suivant :

- 0 = glace
- 1 = mur
- 2 = position de départ
- 3 = drapeau

**niveau1.iwk**

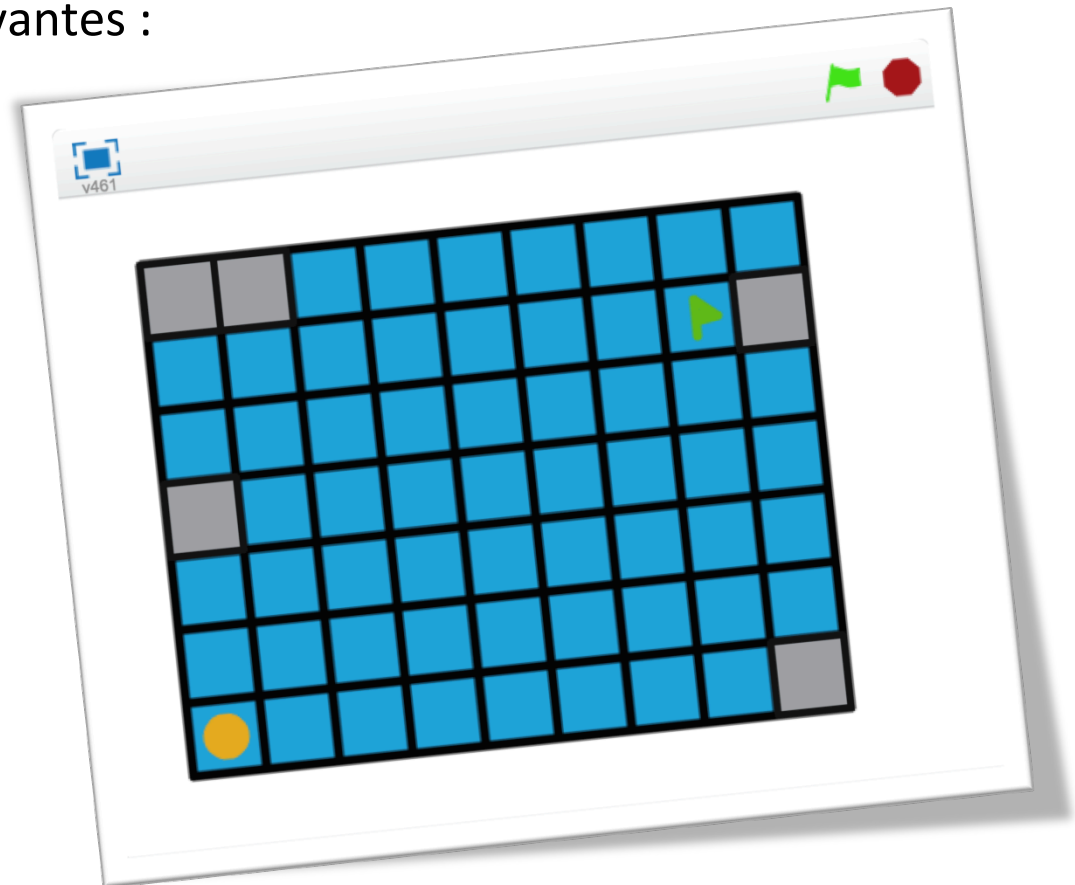
```
1100000000
0000000031
0000000000
1000000000
0000000000
0000000000
0000000000
2000000001
```

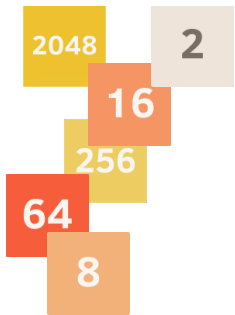




# Interface graphique

- Vous pouvez choisir les couleurs que vous voulez
- Si vous voulez reproduire exactement l'aspect du jeu d'origine, les couleurs (R, V, B) sont les suivantes :
  - Glace : (23, 159, 215)
  - Mur : (156, 158, 162)
  - Balle : (225, 169, 26)
  - Drapeau : image fournie



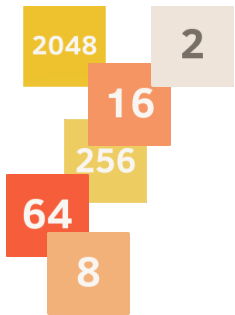


# HOWTO : lire un fichier

- **loadStrings(nom\_fichier) :**
  - Lit le fichier dont le nom est passé en paramètre et retourne un tableau de String dont chaque élément correspond à une ligne du fichier
  - Exemple :

```
String lignes[] = loadStrings("grille1.tak");
```
- **charAt(idx) :**
  - Renvoie le caractère à l'indice `idx` dans la chaîne de caractères
  - Exemple :

```
String ligne = "10101100";  
int val0 = ligne.charAt(0) - '0';  
// val0 vaut 1 (la valeur correspondant au premier  
// caractère de la chaîne ligne)
```

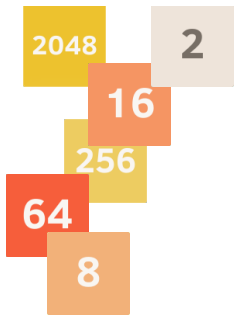


# « Originalité » du programme

- Le travail est à faire **seul** ou en **binôme** !
- Pour les binômes, les 2 partenaires du binôme doivent être **dans le même groupe de TD**, et de **niveau comparable**
- Tout « emprunt » à un autre binôme ou à un programme sur Internet doit être explicité
  - Les programmes seront comparés entre eux grâce à un logiciel anti-plagiat
  - Une note d'originalité entre 0 (plagiat complet) et 1 (travail original) sera donnée
  - La note finale sera calculée selon la formule :

$$\text{note}_{\text{finale}} = \text{note}_{\text{brute}} * \text{note}_{\text{originalité}}$$



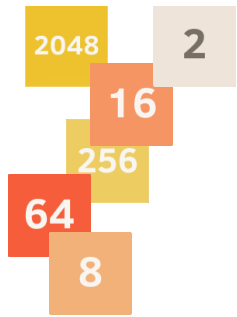


# Critères de notation (1/2)

## Respect des contraintes

- **(2 points)** rendre son DM en temps et en heure en respectant les contraintes :
  - Date de rendu : **vendredi 4 janvier 2019 à 23h** (tout rendu impossible après cette date)
  - Rendu sur e.media **exclusivement**
  - Programme dans un fichier .pde **unique** nommé :
    - Pour un monôme : **GxDM3\_NOM\_Prenom** (même convention de nommage que pour les DM1 et DM2)
    - Pour un binôme : **GxDM3\_NOM1\_Prenom1\_NOM2\_Prenom2**
  - Rendu du dossier correspondant **compressé au format zip (à l'exclusion de tout autre format !)**





# Critères de notation (2/2)

## Forme

- **(1 point)** indenter et commenter correctement son programme
- **(2 points)** structurer son programme en fonctions
- **(2 points)** utiliser des tableaux et des boucles dès que c'est possible

## Fonctionnalités

- **(3 points)** charger un niveau (fichiers avec l'extension « .iwk »)
- **(2 points)** visualiser le niveau
- **(2 points)** mettre en place un écran d'aide au démarrage
- **(1 point)** mettre en place l'interaction clavier
- **(3 points)** calculer le déplacement de la balle sur la glace et vérifier l'arrivée de la balle sur le drapeau (à la fin d'un déplacement)

## Qualité

- **(4 points)** design visuel et sonore, jouabilité, originalité, extensions, investissement...