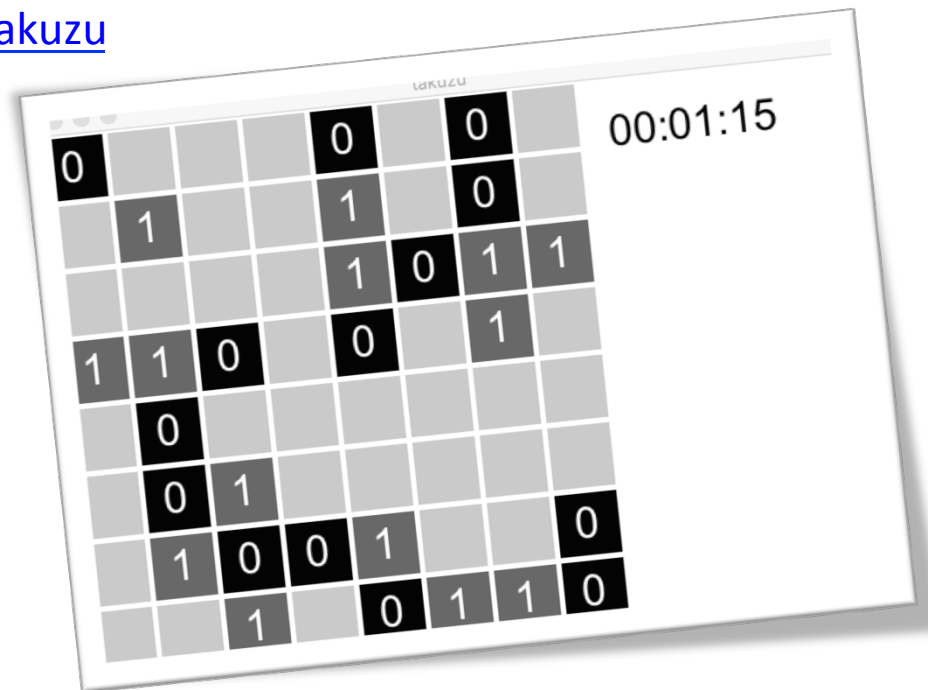
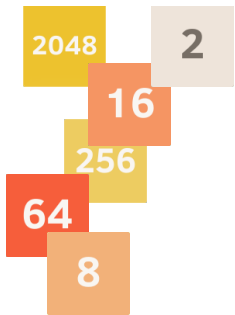


Devoir à rendre N° 3

Ecrire un programme permettant de compléter des grilles de Takuzu (ou binaïro, binero, bineiro, binary puzzle, brain snacks, zernerero)

<https://fr.wikipedia.org/wiki/Takuzu>





Principe du Takuzu

- Le Takuzu est un jeu de réflexion d'origine belge créé par Frank Coussement et Peter De Schepper en 2009, et déposé à l'époque sous le nom de Binaïro
- Chaque grille ne contient que des 0 et des 1, et doit être complétée en respectant trois règles :
 - autant de 1 et de 0 sur chaque ligne et sur chaque colonne
 - pas plus de 2 chiffres identiques côte à côte
 - 2 lignes ou 2 colonnes ne peuvent être identiques

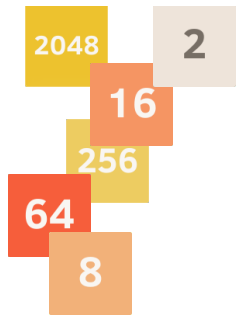
[Wikipedia, <https://fr.wikipedia.org/wiki/Takuzu>]

Exemple de
Grille 4x4 :

	1		0
		0	
	0		
1	1		0

Solution :

0	1	1	0
1	0	0	1
0	0	1	1
1	1	0	0

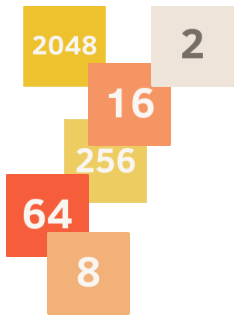


Travail à faire

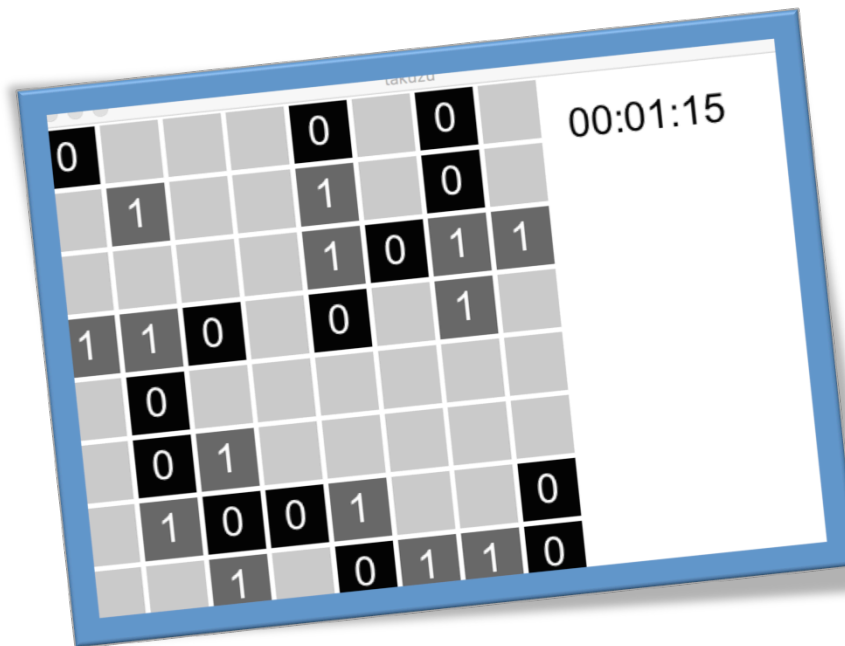
- L'objectif est de réaliser une interface permettant de jouer au jeu de Takuzu. Ce programme doit permettre de :
 - charger un fichier décrivant une grille de Takuzu de taille 8x8 ainsi que le fichier solution (voir diapo suivante) ;
 - compléter la grille en cliquant sur les cases pour :
 - indiquer un 0 par un premier clic
 - indiquer un 1 par un deuxième clic
 - revenir à une case indifférenciée à par un troisième clic
 - vérifier la solution :
 - soit automatiquement dès que la dernière case est complétée
 - soit manuellement en cliquant sur un bouton
 - Chronométrer le temps de résolution (à partir du 1^{er} clic dans la grille)

On peut avoir le même jeu avec des couleurs : <http://0hh1.com/>

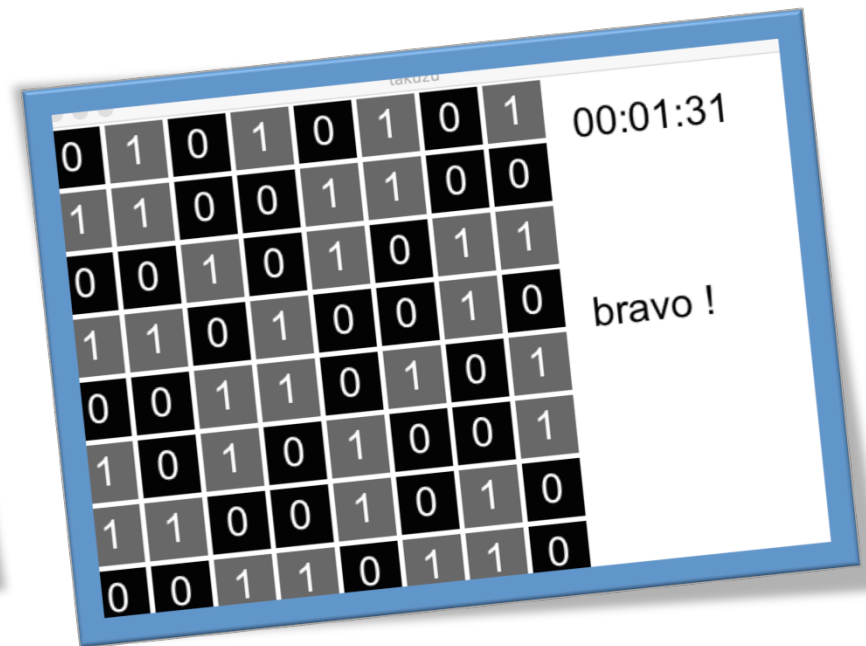
Ou avec des croix et des ronds : <https://scratch.mit.edu/projects/40867092/>



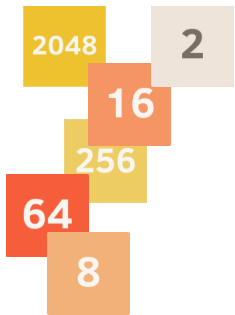
Exemple d'interface



En cours de résolution



En fin de résolution

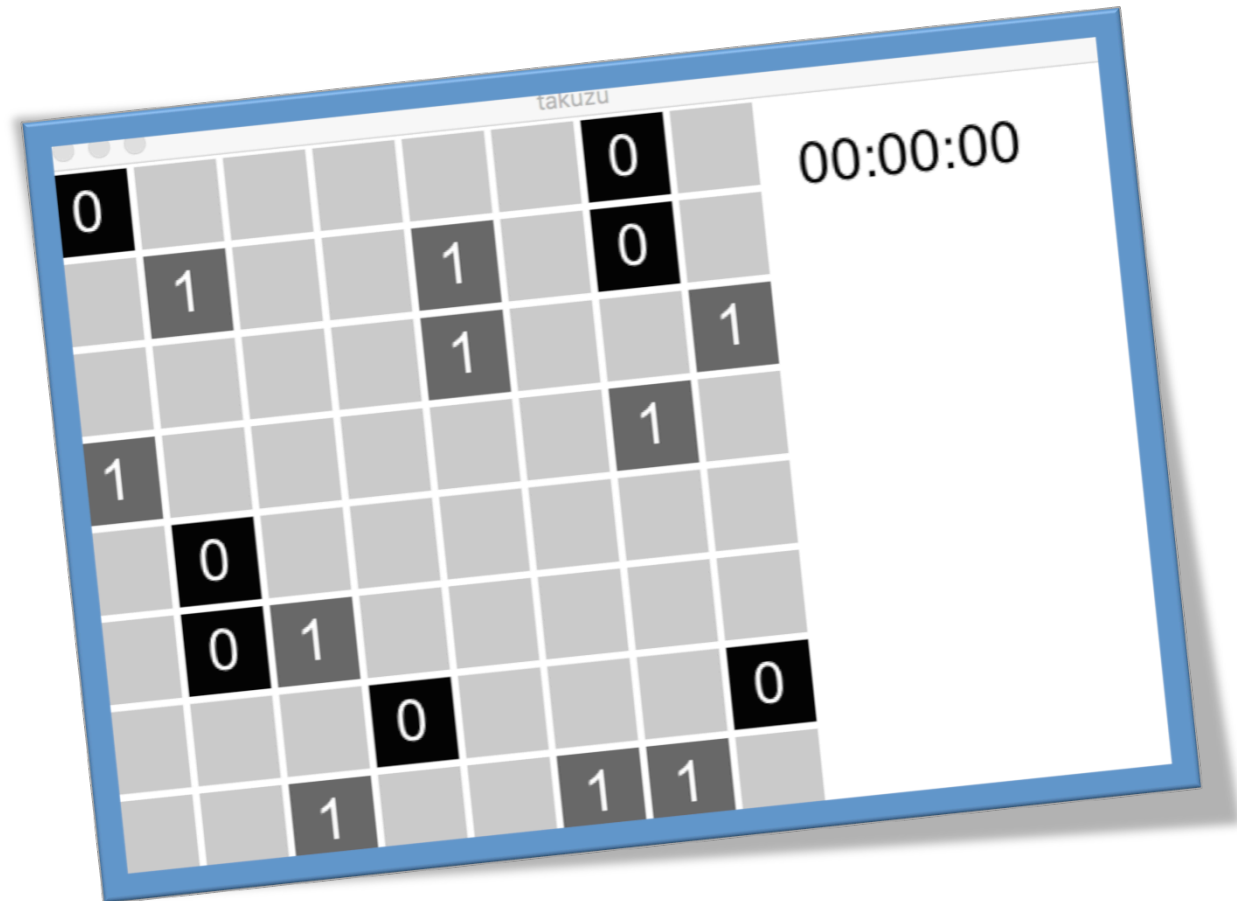


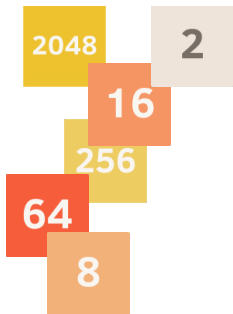
Fichiers de grilles

L'énoncé du problème

grille1.tak

```
8
02222202
21221202
22221221
12222212
20222222
20122222
22202220
22122112
```



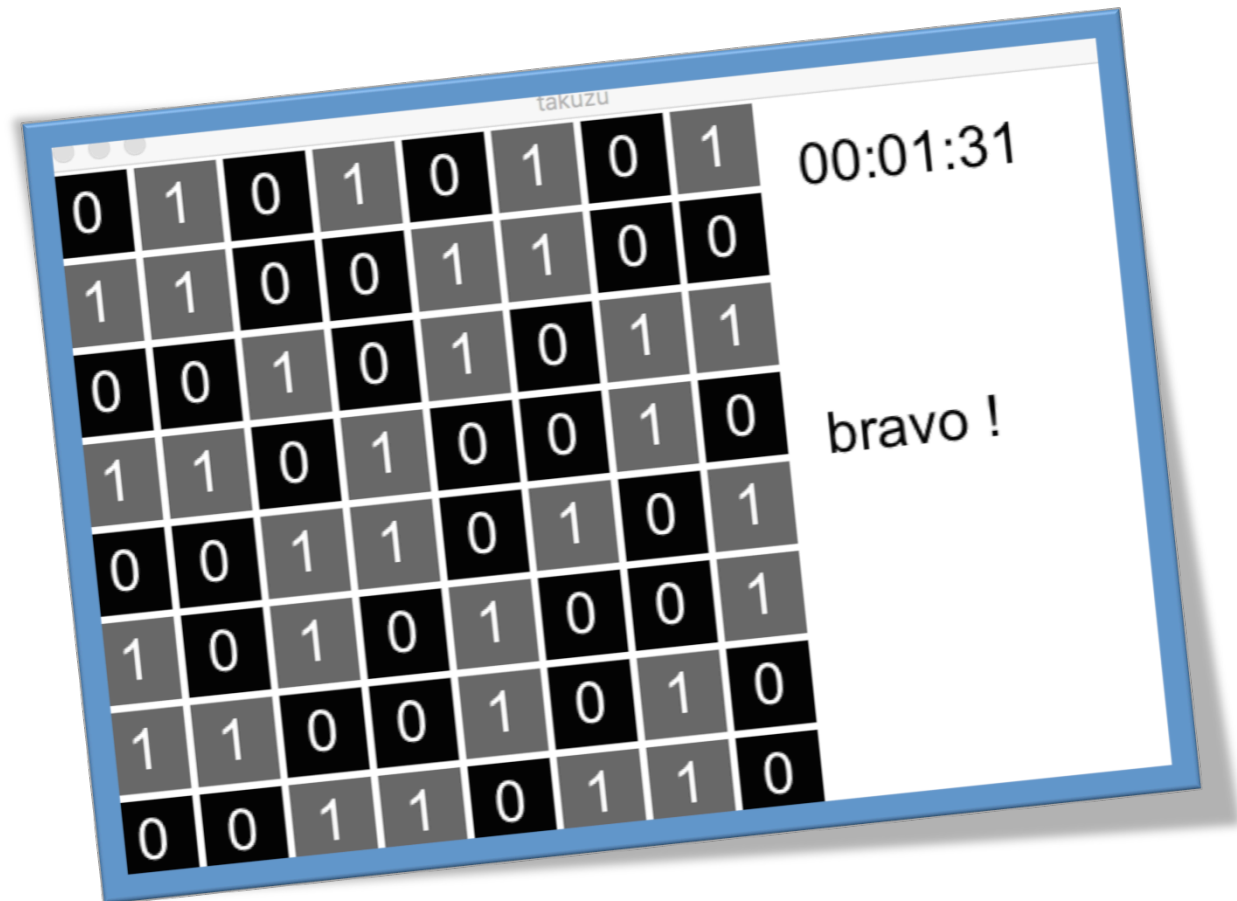


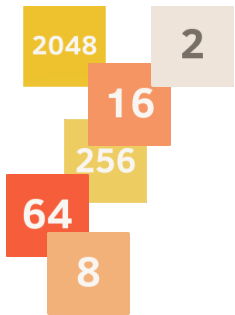
Fichiers de grilles

La solution du problème

grille1.sol

```
8
01010101
11001100
00101011
11010010
00110101
10101001
11001010
00110110
```



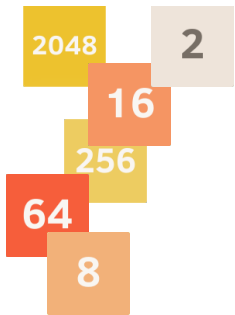


Pour lire un fichier

- **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(indice) :**
 - Renvoie le caractère à l'indice `indice` dans la chaîne de caractères
 - Exemple :

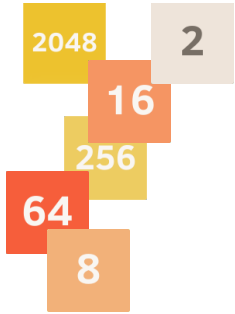
```
String ligne = "10101100";  
int val0 = ligne.charAt(0) - '0';  
// val0 vaut 1 (la valeur correspondant au premier  
// caractère de la chaîne ligne)
```



Pour choisir un fichier

- **`selectInput("message", callback)` :**
 - Ouvre une boîte de dialogue pour sélectionner un fichier, et appelle la fonction « callback » en lui transmettant le fichier choisi
 - Exemple :

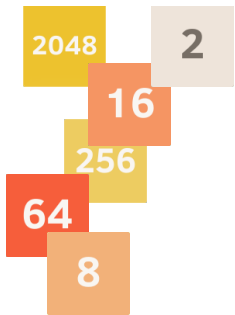
```
void setup() {  
    selectInput("Sélectionnez la grille", "fileSelected");  
}  
  
void fileSelected(File selection) {  
    if (selection != null) {  
        println(selection.getAbsolutePath());  
    }  
}
```

Pour chronométrer

- **millis()** :
 - Renvoie le nombre de millisecondes écoulées depuis le démarrage du programme
 - Exemple :

```
int start = millis();
```

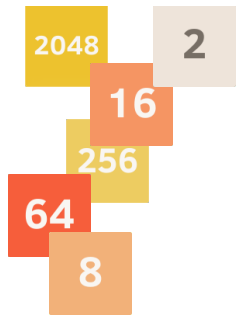


« Originalité » du programme

- 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 :

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

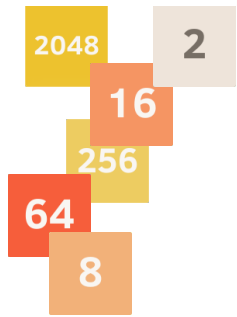




Critères de notation (1/2)

Respect des contraintes

- **(2 points)** rendre son DM en temps et en heure en respectant les contraintes :
 - Date de rendu : **vendredi 5 janvier 2018 à 23h55** (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
- **(1 point)** fournir un manuel utilisateur
- **(1 points)** structurer son programme en fonctions
- **(2 points)** utiliser des tableaux et des boucles dès que c'est possible

Fonctionnalités

- **(2 points)** sélectionner et charger un problème (fichiers avec l'extension « .tak »)
- **(2 points)** modifier l'état d'une case
- **(1 point)** implémenter un chronomètre
- **(2 points)** visualiser la grille
- **(2 points)** vérifier la solution (automatiquement ou manuellement)

Qualité

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