

# Rapport L<sup>A</sup>T<sub>E</sub>X

Kyrian Balavoine

7 avril 2024

## Résumé

**Mosaic** est un jeu créé par *Simon Tatham*. C'est un jeu type puzzle semblable à un démineur

## Table des matières

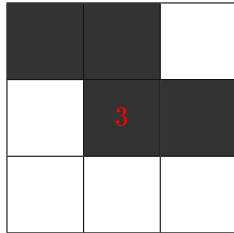
<b>1</b>	<b>Le jeu</b>	<b>1</b>
1.1	Règles . . . . .	1
1.2	La présentation . . . . .	2
1.3	notre jeu . . . . .	3
<b>2</b>	<b>Le code</b>	<b>3</b>
2.1	réduction . . . . .	3
2.2	solveur . . . . .	4

## 1 Le jeu

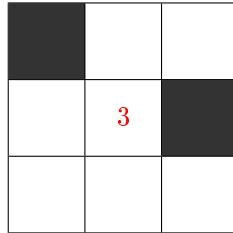
### 1.1 Règles

- Chaque case du jeu peut être coloré en noir ou en blanc
- La contrainte d'une case est le chiffre indiqué dedans. Il concerne les cases dans le carré 3x3 autour de la contrainte.
- Une case est dite "satisfaite" si le nombre de cases noires dans son entourage correspond à sa contrainte, et toutes les autres cases sont blanches
- Le jeu est gagné quand toutes les cases sont satisfaites

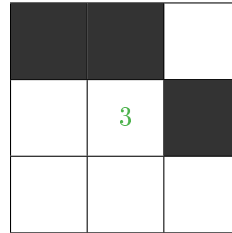
Example :



(a) too much black



(b) too much white



(c) perfect

## 1.2 La présentation

Par défaut, le jeu ressemble à ça quand on lance une partie :

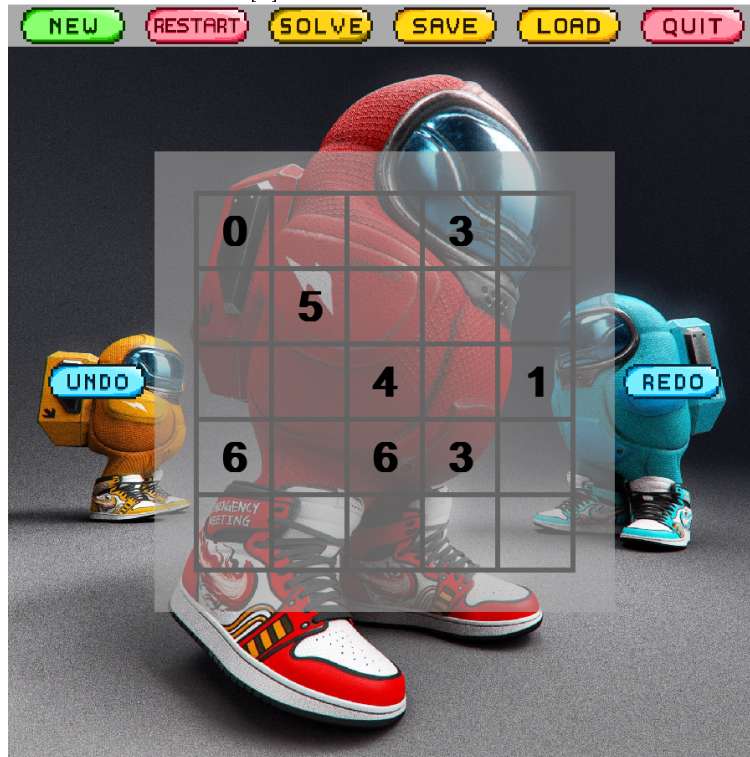
0			3	
	5			
		4		1
6		6	3	

Cette même partie, une fois gagnée, ressemble à ça :

0			3	
	5			
		4		1
6		6	3	

### 1.3 notre jeu

avec mon binôme[?], nous avons fait ce jeu :



## 2 Le code

### 2.1 réduction

certaines fonctions étaient beaucoup trop longues, avec une série de switch casque qui faisait plus de 40 lignes. Grâce à une liste, il a été raccourci à 4 lignes :

```
1 int mov_i[9] = {0, -1, 1, 0, 0, -1, -1, 1, 1};
2 int mov_j[9] = {0, 0, 0, -1, 1, -1, 1, -1, 1};
3 i = ((signed int)i + mov_i[ dir ]);
4 j = ((signed int)j + mov_j[ dir ]);
```

## 2.2 solveur

Le plus grand obstacle auquel j'ai fait face dans la programmation du jeu, c'est le solveur. en effet, un jeu 5x5 à  $2^{25}$  possibilités. Après de nombreuses épreuves, j'ai réussi à réduire le nombre de possibilité au nombre de victoire possible, soit 1 dans le jeu ci-dessus (1.2)

## Références

[1] Lucas Rivoal. porte-parole. 2024. groupe b15.