

---

# MICROPROJET

---

Le jeu du démineur

---

# Rapport Projet Démineur

---

## Table des matières

MICROPROJET .....	1
Rapport Projet Démineur.....	2
Introduction .....	3
Description des fonctionnalités .....	4
Présentation de la structure du programme et diagramme de classe .....	11
Explication mécanisme de sauvegarde .....	12
Algorithme de révélation de plusieurs cases .....	13
Conclusion personnelle .....	14

## Introduction

---

Le microprojet consiste en la réalisation du jeu du démineur grâce au langage informatique Java.

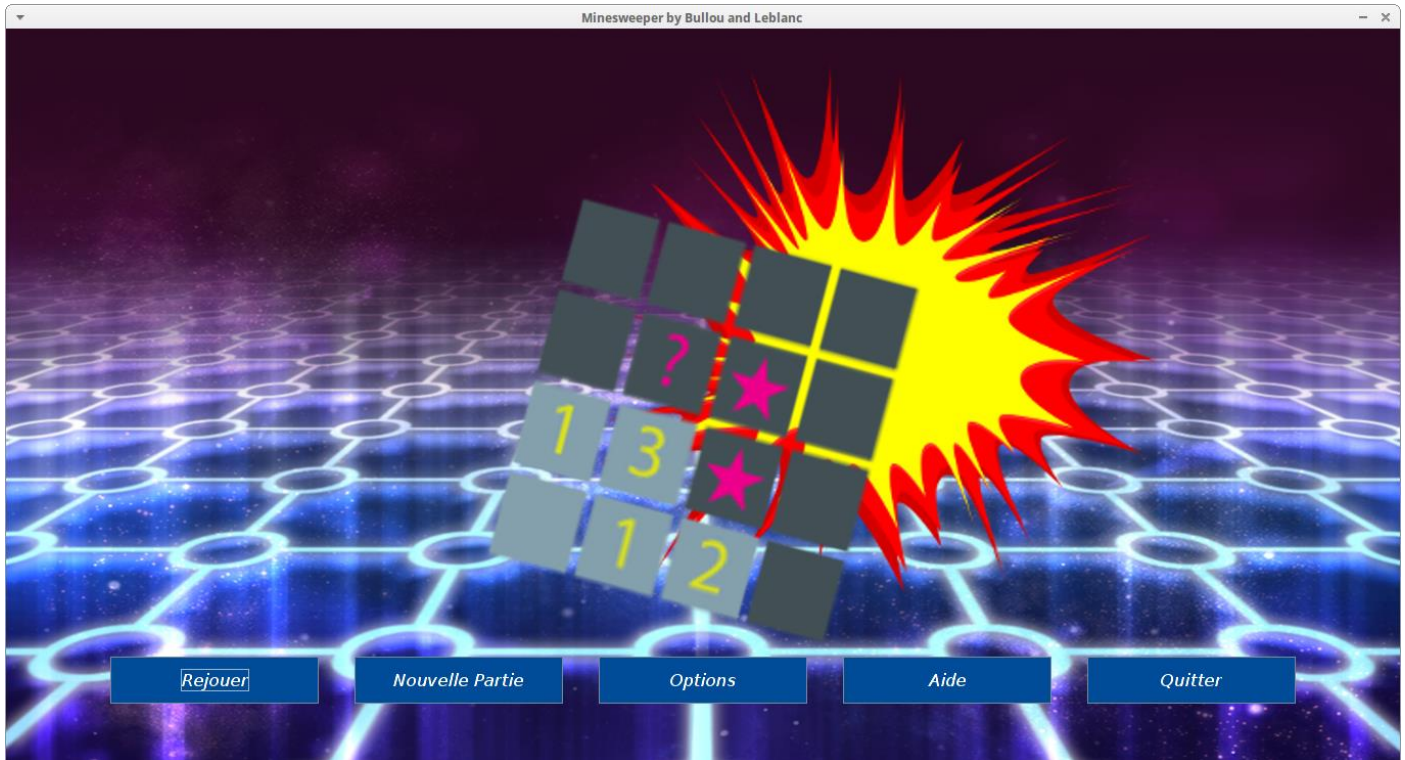
Le démineur est un jeu de réflexion composé d'une grille contenant des cases minées et des cases non minées. Le but du jeu est de découvrir toutes les cases non minées sans découvrir une seule case minée sinon c'est la fin de la partie.

Pour aider le joueur, les seules indications sont le nombre de mines autour de chaque case non minée et le nombre de mines restantes (case marquée par un drapeau)

## Description des fonctionnalités

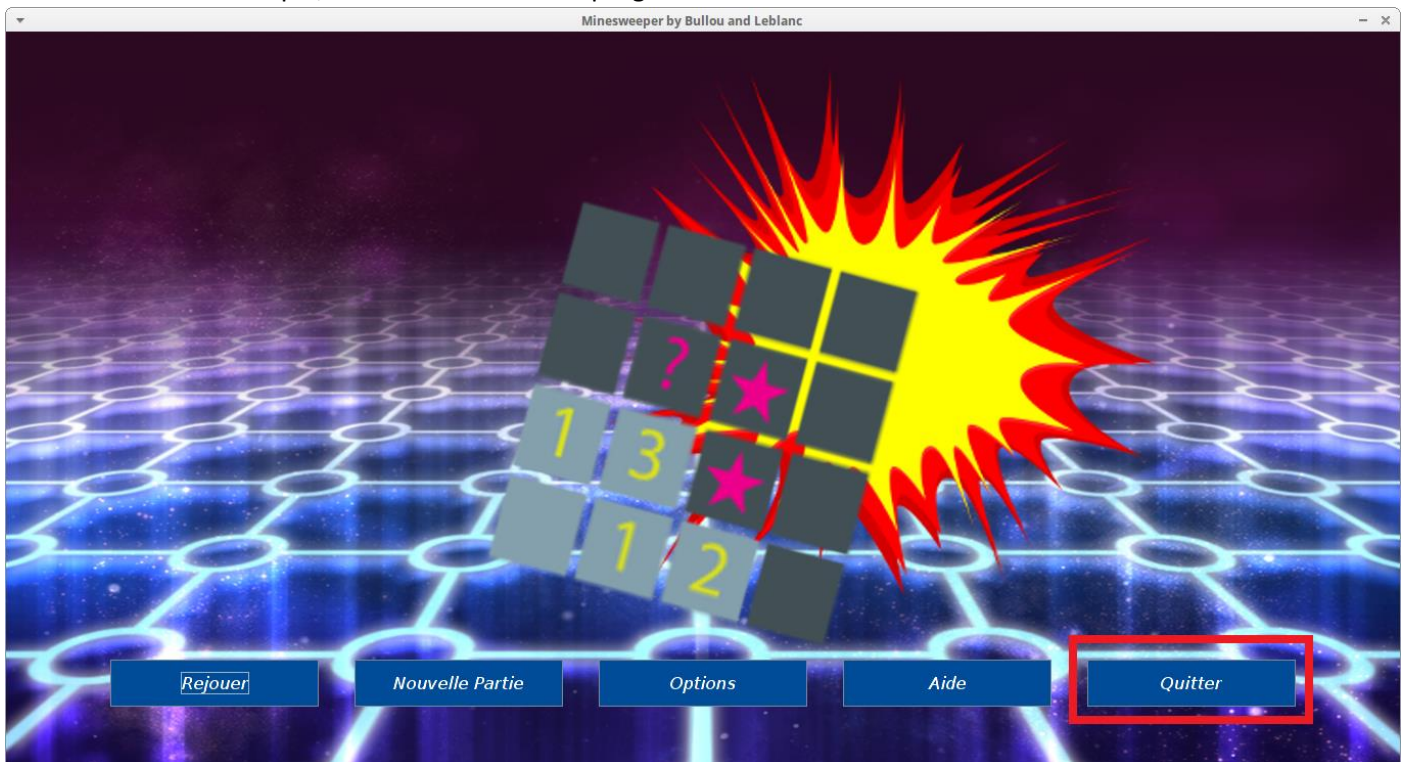
### Menu

Au lancement du programme l'utilisateur voit un premier menu. Il a le choix entre Rejouer, Nouvelle partie, Options, Aide, Quitter.



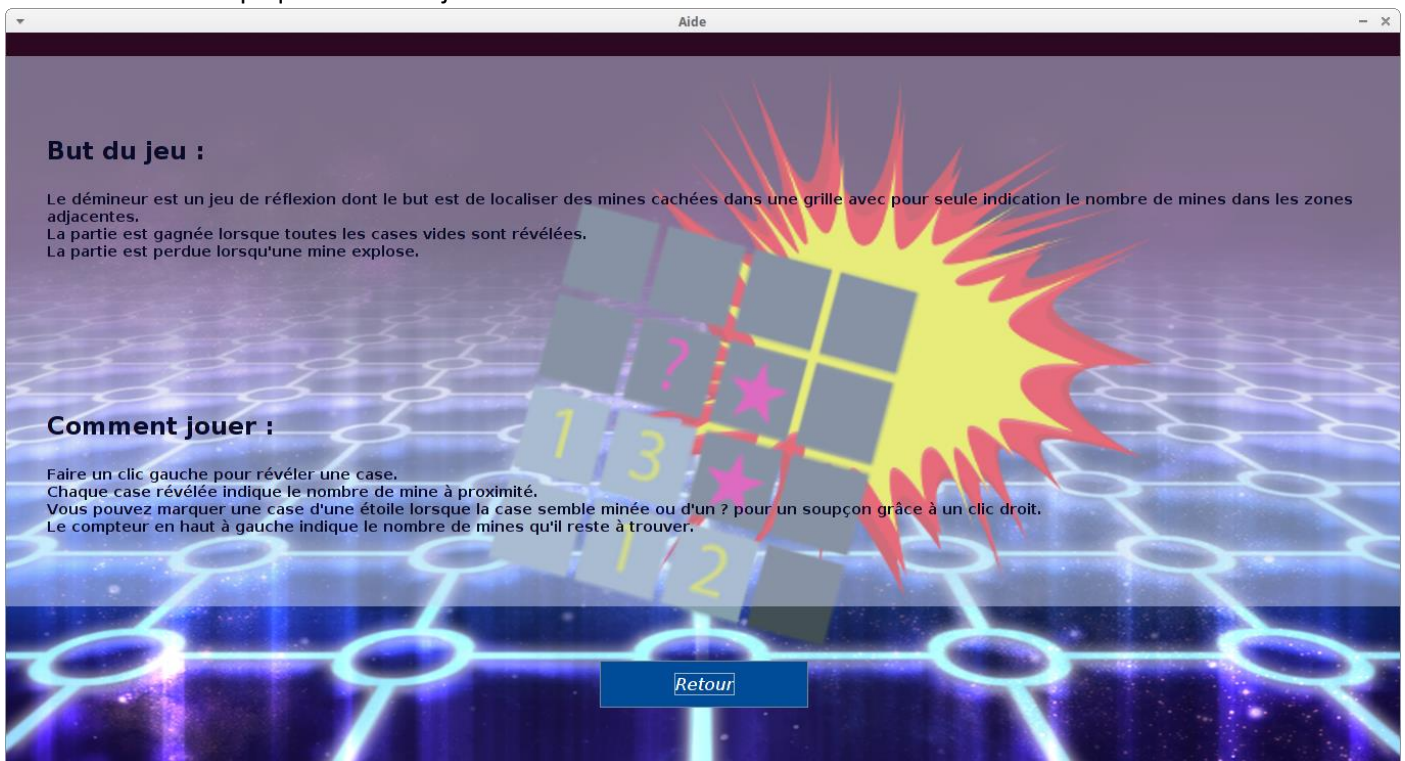
## Quitter

Comme son nom l'indique, ferme la fenêtre et le programme.



## Aide

La fenêtre d'aide explique comment jouer au démineur.





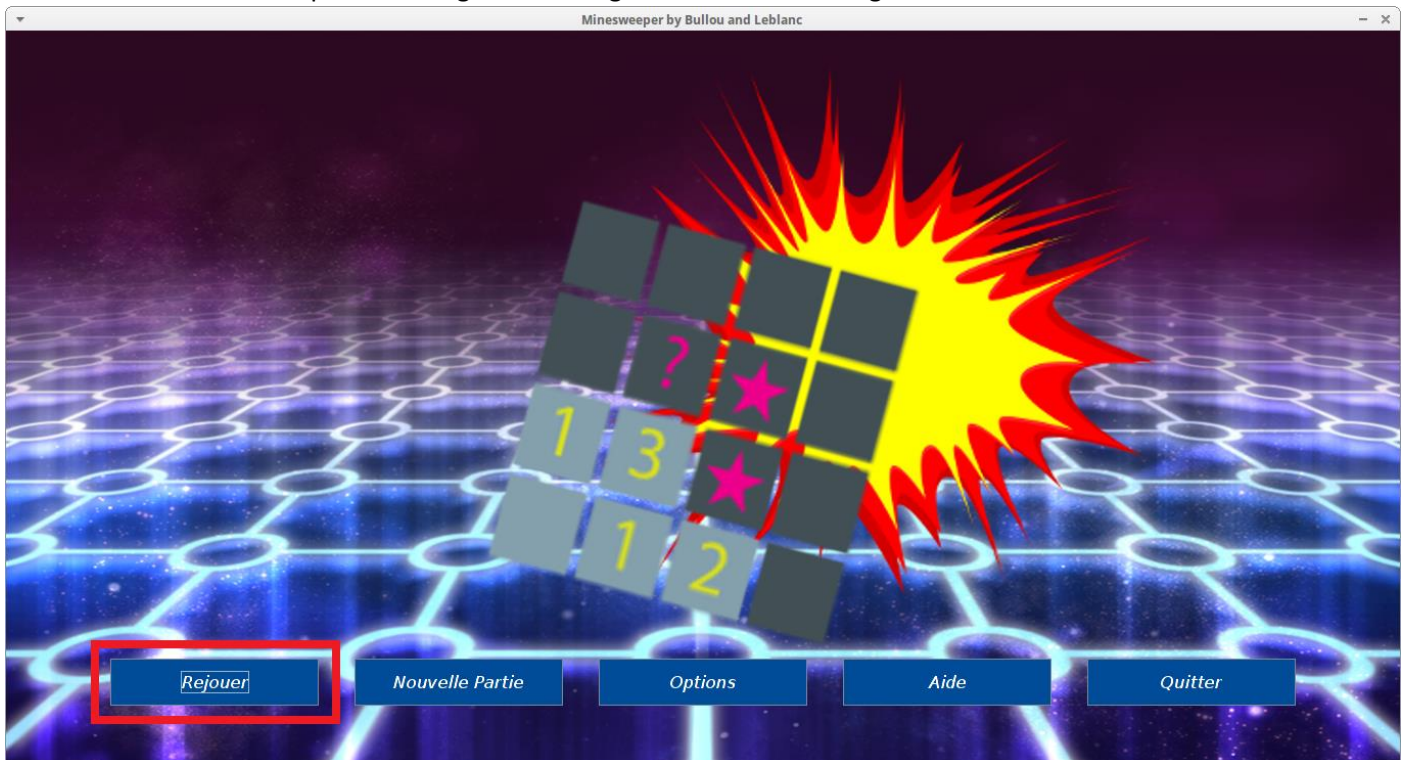
## Options

La fenêtre d'options permet de configurer l'interface utilisateur, pour l'instant il n'existe que le choix de la résolution. (Redémarrer pour appliquer).



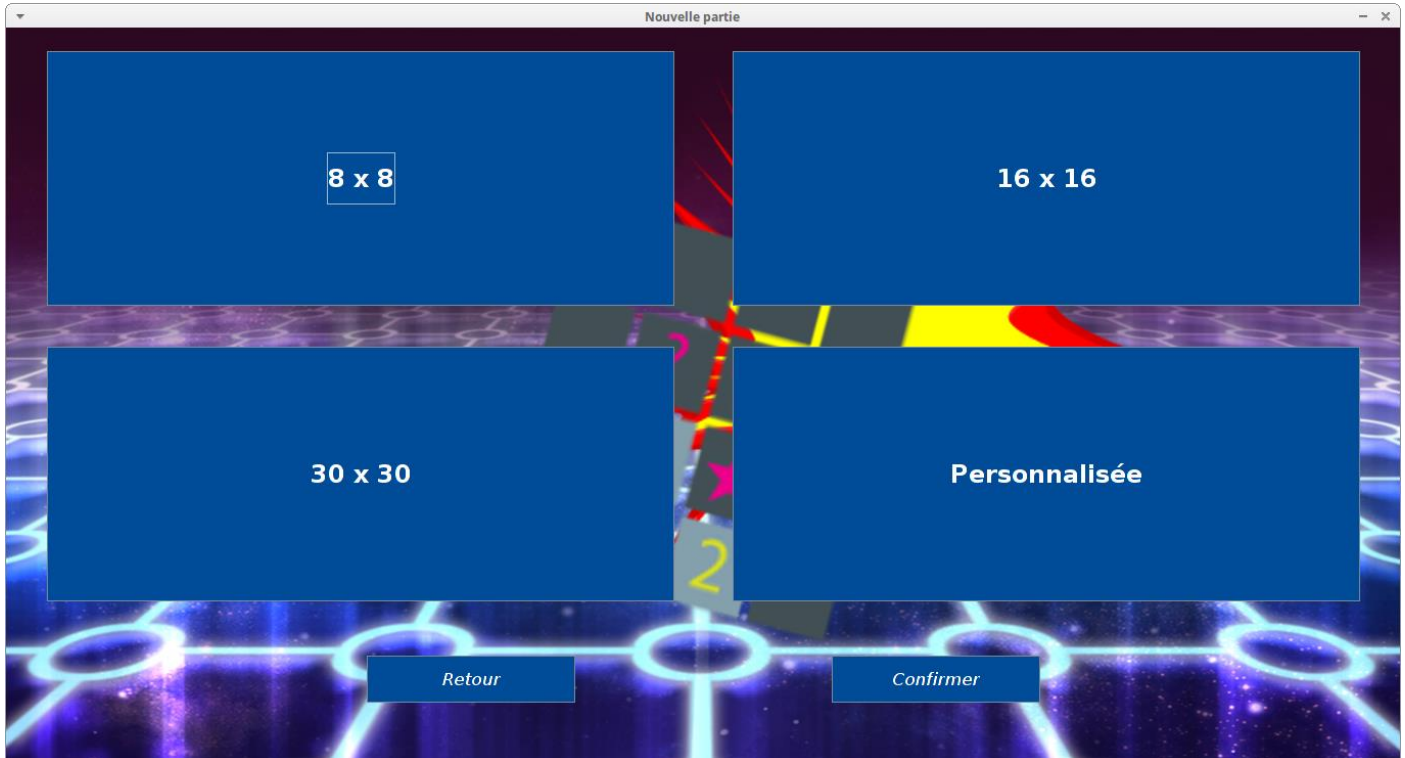
## Rejouer

Permet de continuer une partie sauvegardée. Charge un fichier de sauvegarde au format .dat.



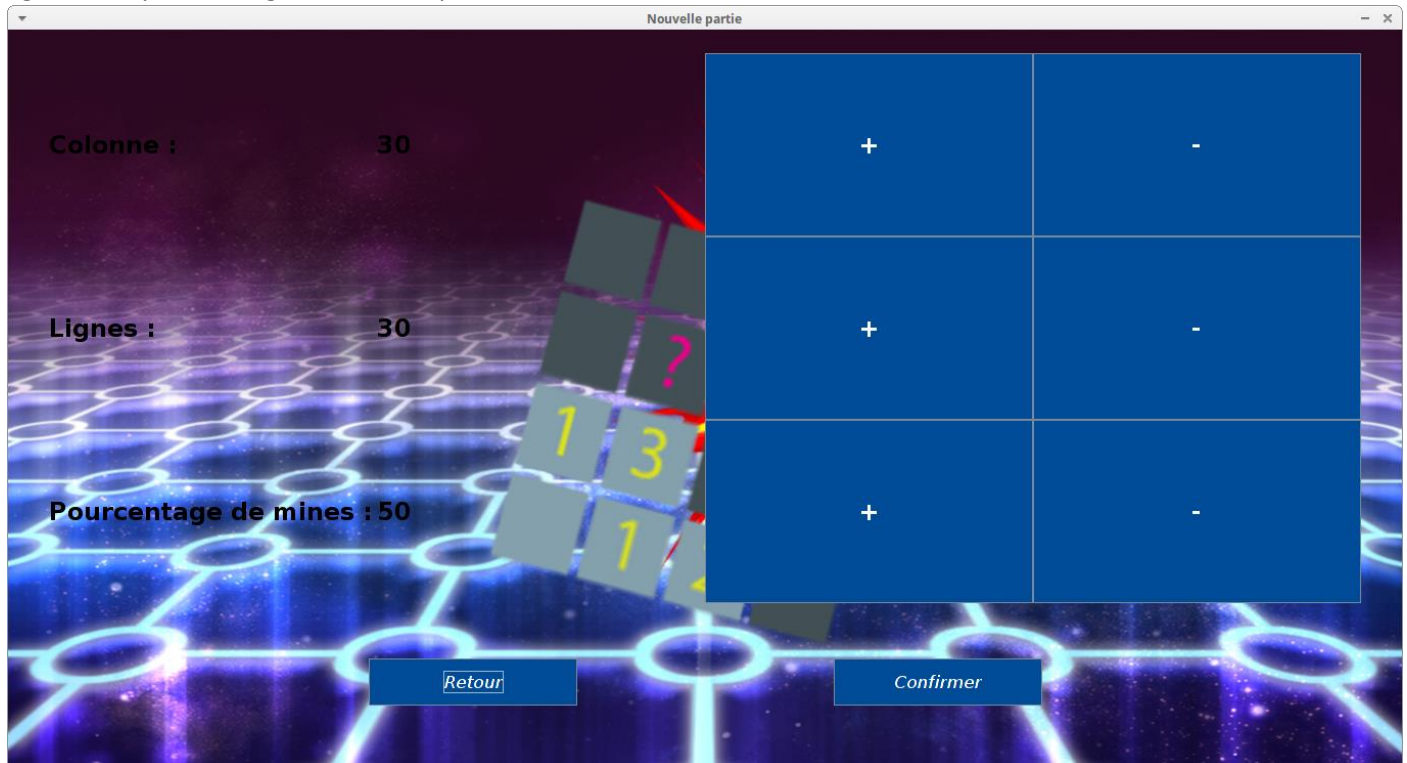
## Nouvelle partie

Un second menu apparaît demander à l'utilisateur le choix entre des grilles prédéfinies ou son choix personnel. Puis lorsque son choix est fait le jeu commence et affiche la grille nouvellement créée ainsi que le nombre de mine restantes (case marquée par un drapeau).



## Personnalisé

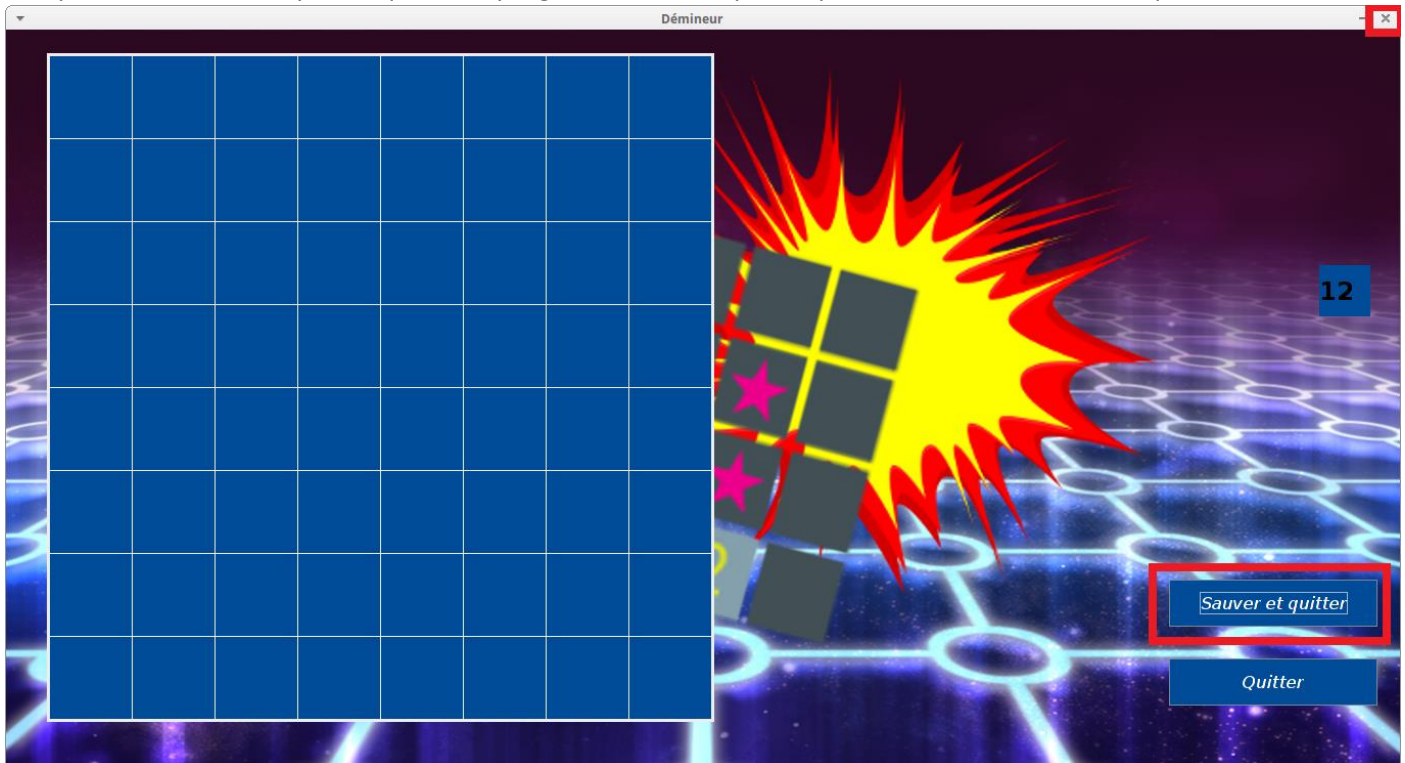
L'utilisateur choisit les dimensions de la grille de démineur, il peut modifier le nombre de colonnes, le nombre de lignes et le pourcentage de bombes présentent dans la mine.





### *Sauver et quitter*

Le programme sauvegarde la partie en cours dans un fichier de sauvegarde au format .dat et ferme le programme lorsque l'utilisateur accepte de quitter le programme ou lorsqu'il clique sur le bouton Sauver et quitter.



### *Affichage du nombre de mines restantes*

On montre au joueur combien de mines il reste (cases marquées d'une étoile) .

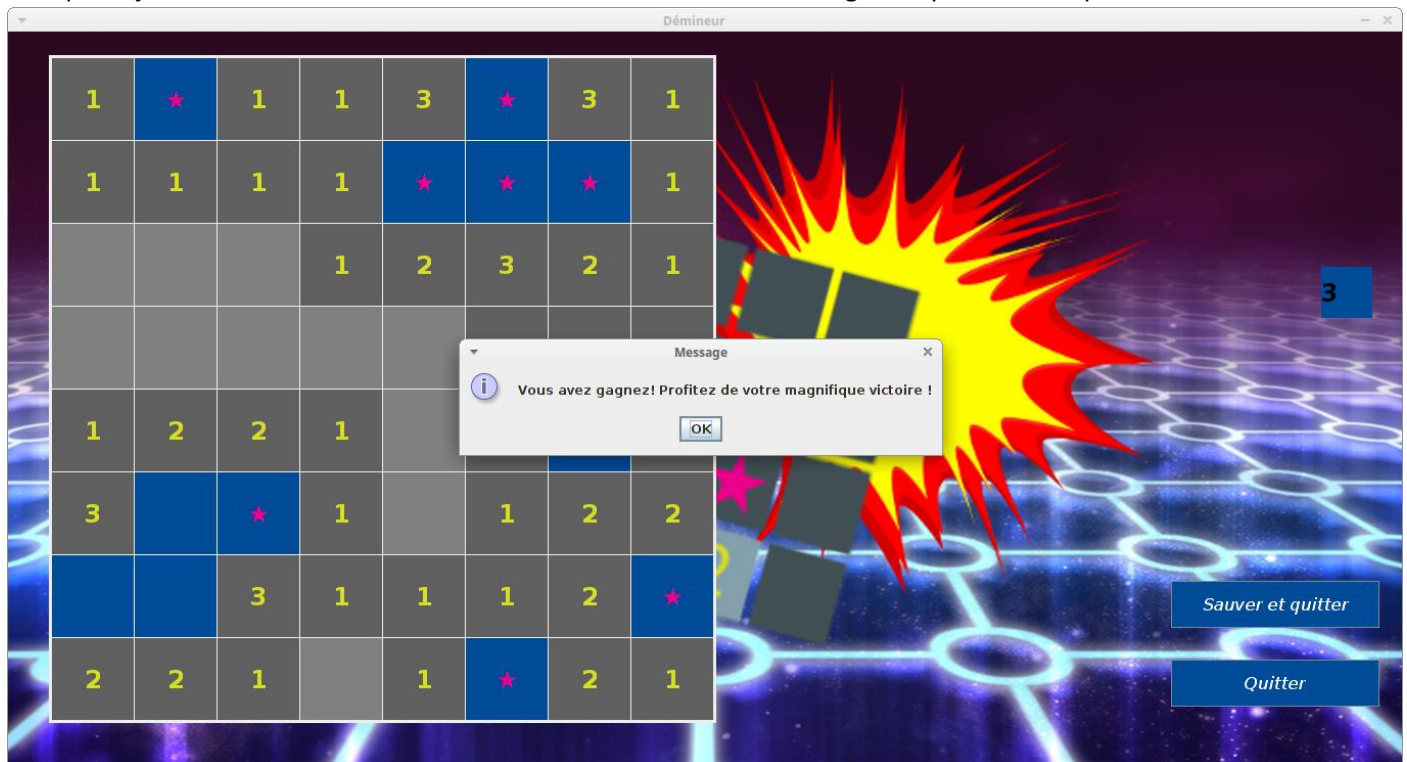
### *Les boutons "Retour"*

Lors de la navigation dans le programme l'utilisateur pourra voir des boutons Retour, Annuler, ils permettent de revenir une fenêtre précédant celle où il se situe.

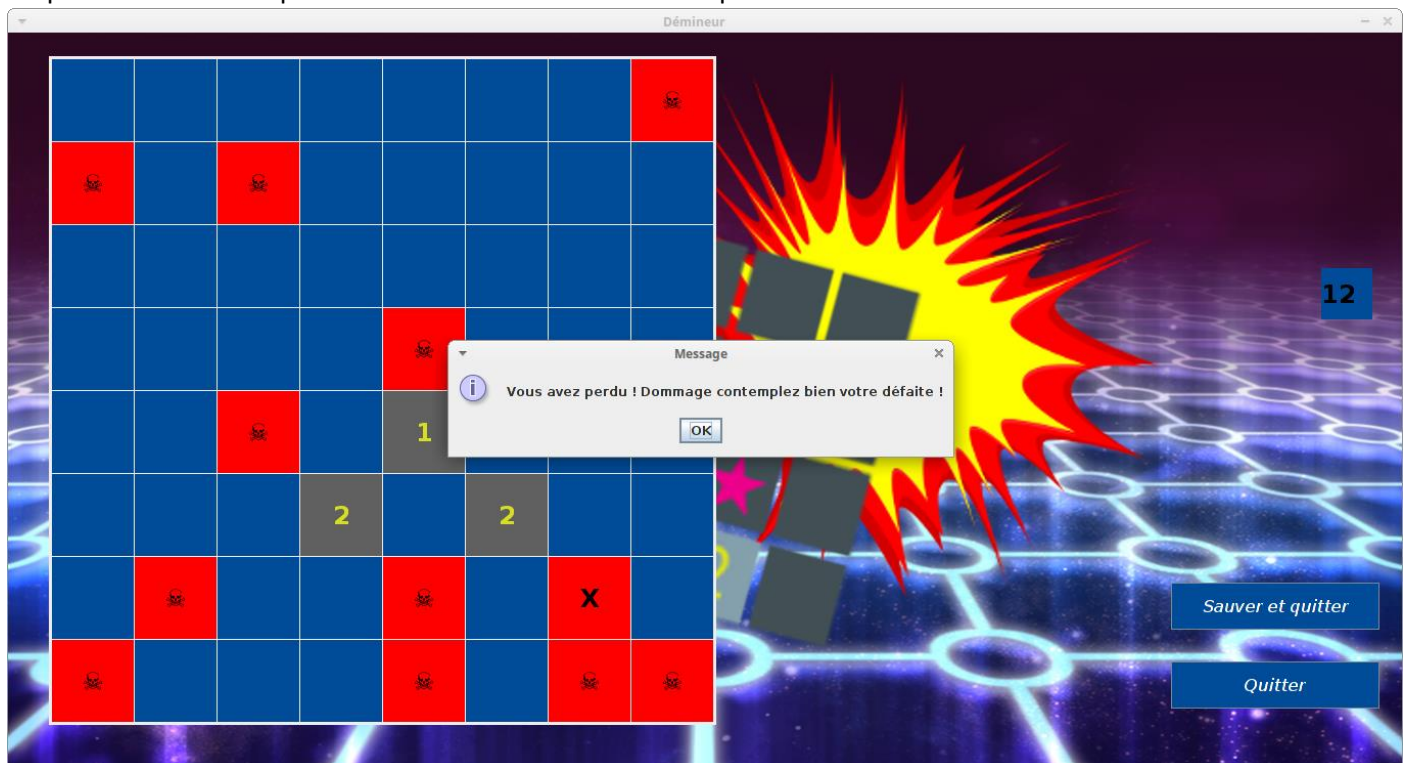


## Victoire et Défaite

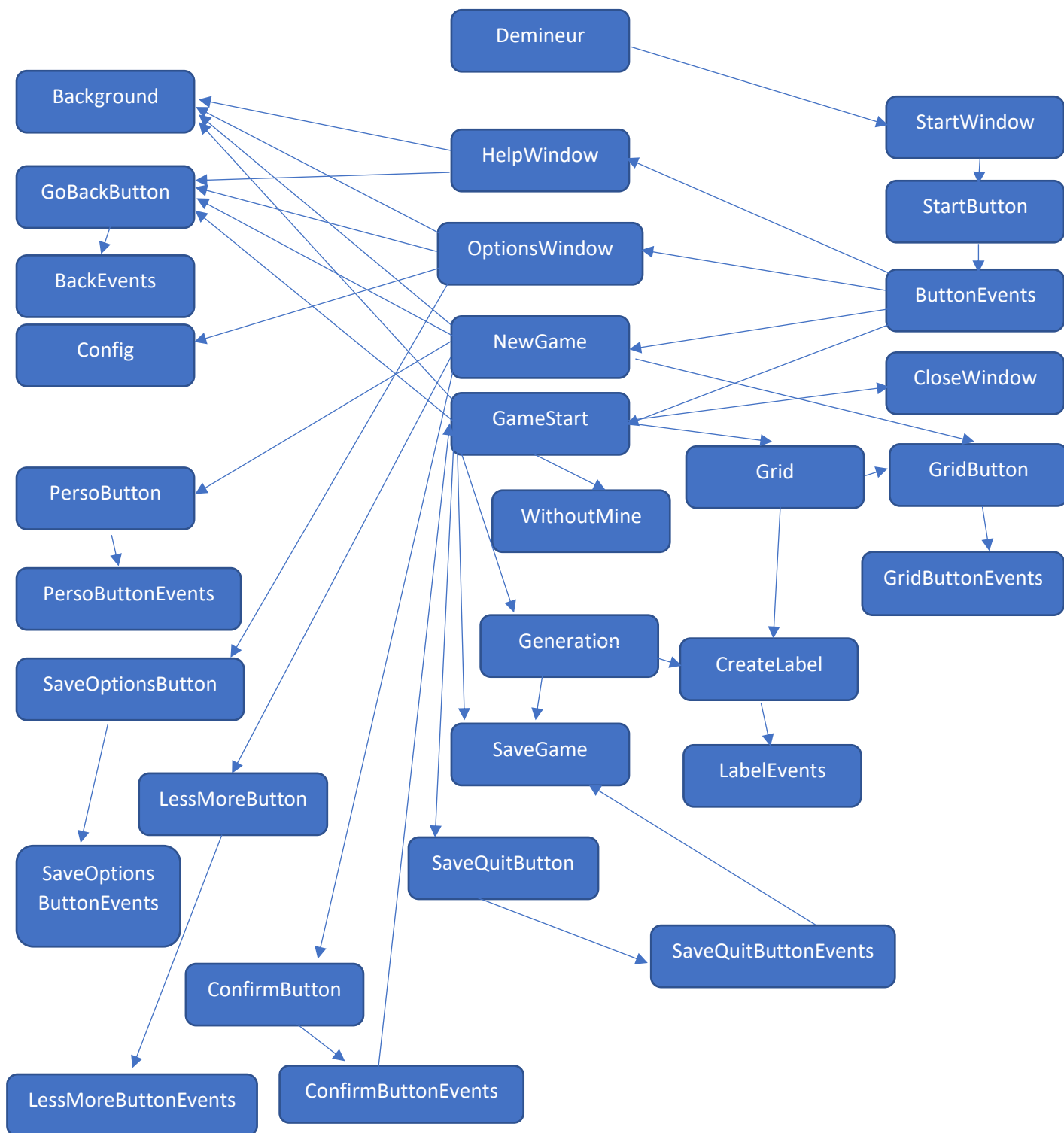
Lorsque le joueur a découvert toutes les mines on lui affiche un message et il peut contempler sa victoire.



De même lorsque le joueur tombe sur une case minée, on lui affiche un message et il peut ...contempler... sa défaite. De plus on lui montre quelle mine a causé sa défaite et l'emplacement de toutes les autres.



## Présentation de la structure du programme et diagramme de classe



## Explication mécanisme de sauvegarde

---

Le mécanisme de sauvegarde est tout simple :

On va dans un premier temps sauvegarder les dimensions de la grille (longueur puis largeur) puis le nombre de mines dans cette grille, ensuite on parcourt la grille de tous les JLabel la composant et on écrit dans l'ordre :

- la variable indiquant si la mine est minée ou non (true / false)
- la variable indiquant l'état de la mine (normal, drapeau, soupçon ou révélée).

Et voilà notre fichier de sauvegarde se tient de la forme suivante :

[longueur grille] [largeur grille] [mine case(x)] [état case(x)] [mine case(x + 1)] [état case(x + 1)] [mine case(x + 2)] [état case(x + 2)] ... [mine case(longueur grille \* largeur grille - x + 1)] [mine case(longueur grille \* largeur grille - x)]

Pour reconstruire la grille on va lire dans le fichier créé en suivant la même façon que pour l'écriture dans l'ordre :

- lire la longueur de la grille
- lire la largeur de la grille

Pour i = 0 allant à i < longueur grille

    Pour j = 0 allant à j < largeur grille

- lire si minée ou non
- lire état de la case

    j++

    FinPour

  i++

FinPour

## Algorithme de révélation de plusieurs cases

---

### *Classes et méthodes utilisées :*

Classe :

LabelEvents

Méthodes :

Reveal, FillCase

Classe :

CreateLabel

Méthode :

Reveal

### *Code utilisé :*

Fichier LabelEvents.java

(Tout le fichier)

Fichier CreateLabel.java

```
private int State; // 0 Normal, 1 Drapeau, 2 Soupçons, 3 révélée
```

```
/**
```

```
 * Parametre l'etat de la case a 3 (revelee)
```

```
 */
```

```
public void RevealCase(){
```

```
    this.State = 3;
```

```
}
```



## Conclusion personnelle

---

### *Bullou Julien*

Ce projet fut pour moi plus un qu'un simple projet. De par le niveau de connaissance du langage qu'il demande et par les connaissances et la logique mathématique nécessaire pour pouvoir réaliser la récursivité de révélation des cases. J'ai eu la chance d'avoir un coéquipier qui a su m'aider car mon niveau était parfois insuffisant. Grâce à notre travail, le projet a pu être réalisé dans les temps et ce malgré tous les problèmes qui ont pu survenir au cours de la conception et la réalisation de celui-ci.

### *Leblanc Kévin*

La réalisation de ce projet m'a beaucoup plu malgré des heures passés dessus à résoudre les problèmes d'affichage, de gestion des données, et surtout le temps passé à chercher le moyen de faire telle ou telle chose puisque ce que nous avons en tête n'est pas forcément faisable avec le niveau que nous avons. De plus java est un langage très riche et rechercher la méthode qui correspond à ce que nous souhaitons réaliser peut paraître complexe.

Avec mon camarade nous avons essayé de nous partager les tâches mais chacun avait besoin des données de l'autre pour continuer ce qui a fait que j'ai plutôt travaillé sur la gestion des données du programme global, des menus et comportement du programme ainsi que des lectures/écritures dans les fichiers.

J'ai tout fait pour respecter la mise en forme des méthodes vue en cours mais je trouve qu'il y a beaucoup d'améliorations qui peuvent être faites.