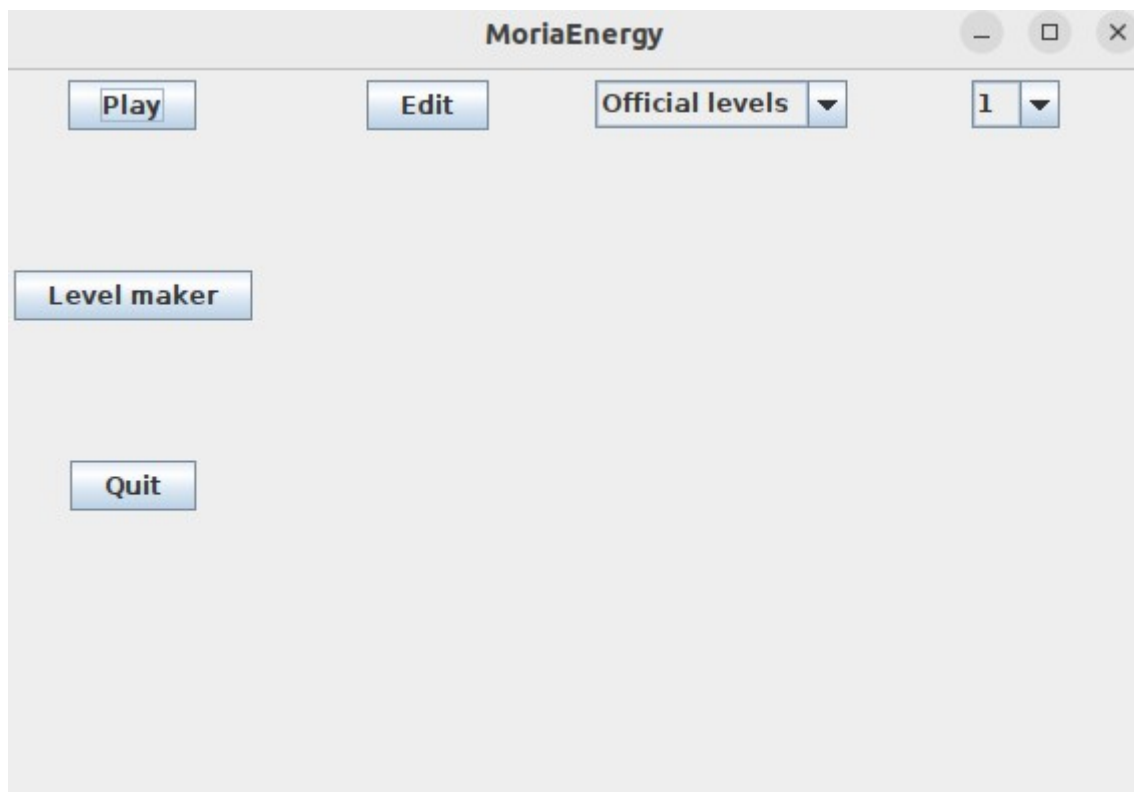


# Rapport d'Interface graphique par Virgile Guinet et Arbi Tazouev

## I Le menu principal

L'utilisateur peut lancer le projet en effectuant la commande gradlew run cela lance le menu principal.



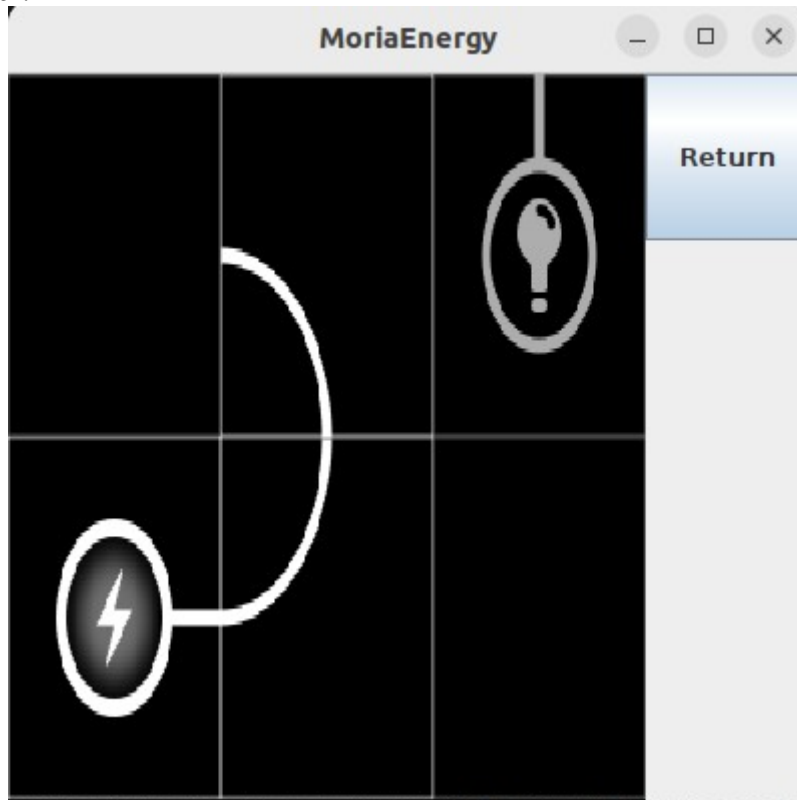
Ce menu offre 6 interactions au joueur :

- Le bouton 'Quit' permet de fermer le menu et de quitter l'application.
- Le bouton 'Level maker' permet de créer son propre niveau
- 'Edit' permet de modifier le niveau actuellement sélectionné et de l'enregistrer en tant que niveau personnalisé
- 'Play' permet de jouer le niveau sélectionné
- La combobox 'Official levels' permet de sélectionner les niveaux officiel ou les niveaux personnalisé
- La seconde combobox indique le niveau choisis par l'utilisateur

Le menu est un objet de type MainMenu il est composé de 2 gridbagLayout inséré l'un dans l'autre  
Nous avons mis les JPanels dans des JButtons pour réduire leurs tailles

## II Le level player

Voici l'écran de jeu :



Le joueur peut cliquer sur une case pour la faire pivoter. Lorsque la source est reliée à des cases les composants s'illuminent. Une fois toutes les lampes allumées on passe au niveau suivant. Le bouton 'return' permet de revenir au menu principal. Lorsque le dernier niveau est complété on revient au menu principal.

Dans notre code nous avons créé une classe mère Cell et deux classes fille Square et Hexagon représentant les cases carrées et hexagonales. On stock dans chaque cellule la liste de ces connexions avec ses voisins.

Pour représenter un niveau nous avons créé une classe Map constitué d'un tableau de tableau de cellules. Lorsque le joueur lance un niveau on lit le fichier correspondant et on crée une Map. On applique un nombre de rotation aléatoire à chaque cellule. Ensuite on dessine les cellules. Pour pouvoir dessiner on délimite la taille d'une cellule en fonction de leur nombre et de la taille de l'écran. On récupère l'image correspondant à la cellule souhaitée et on la dessine.

Lorsque l'utilisateur clique on récupère les coordonnées et on regarde quelle case est la plus proche. On pivote la case la plus proche. On redessine alors le JPanel.

A chaque rotation on vérifie à l'aide d'un algorithme si toutes les lampes sont reliées à des sources, si oui on passe au niveau suivant.

Nous avons implémenté une fonction qui permet de propager l'énergie aux cellules voisines si elles sont reliées. Ainsi l'algorithme de victoire n'a uniquement besoin que de parcourir la Map et de regarder si toutes les lampes sont allumées.

### III Le level maker

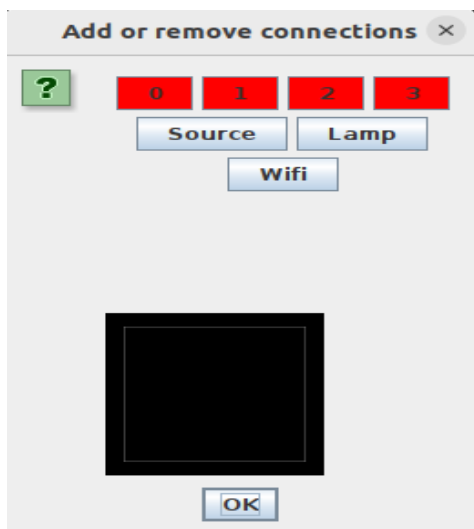
Le level maker permet au joueur de créer ses propres niveaux.

Lorsque l'utilisateur appuie sur le bouton 'level maker' un pop up apparaît et demande la taille ainsi que la forme des cellules voulu par le joueur.



Les différents boutons sur la droite (de haut en bas) permettent à l'utilisateur :

- faire pivoter les cellules lorsqu'il clique dessus
- ajouter une nouvelle cellule là où il clique



- d'effacer entièrement le niveau
- de sauvegarder le niveau à condition qu'il soit dans une position gagnante
- de revenir au menu principal