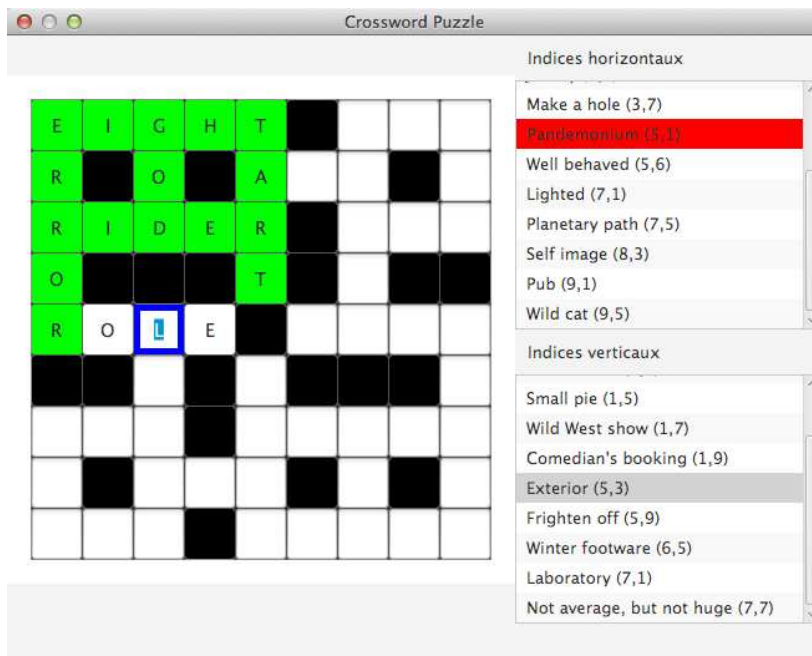


PRA – Mots croisés



Les fonctionnalités de l'IHM :

- Afficher une menu proposant le choix d'une grille aléatoire de mots croisés.
- Afficher une grille de mots croisés de taille variable sous forme d'un GridPane rempli de Label insérés dynamiquement.
- Avoir une case courante, avec le focus de l'application (encadré en bleu sur l'image, en utilisant une feuille de style CSS),
- Afficher des ListView avec les indices horizontaux et verticaux.
- Sélectionner dans les deux listes les indices des mots passant par la case courante. Si un seul mot passe par la case courante, aucun indice ne doit être sélectionné dans l'une des deux listes.
- Conserver une direction courante (horizontale à droite ou verticale vers le bas seulement).
- Afficher l'indice de la direction courante en rouge (en utilisant une autre la feuille de style CSS), et celui de l'autre direction en couleur par défaut.

Les actions sont les suivantes :

- Un clic de la souris sur une case blanche transforme cette case en case courante, et met à jour les indices courants dans les deux listes.
- Un clic de la souris sur une case noire ne doit rien modifier.
- Un clic de la souris sur un indice met la case courante sur la première lettre du mot concerné, et met à jour l'autre indice courant. La direction courante conserve la direction de l'indice sélectionné.
- Un clic sur une touche entre "A" et "Z" affiche cette lettre dans la case courante et avance la case courante dans la direction courante (à droite ou en bas), quand c'est possible.

- Il est évidemment impossible de rentrer plusieurs lettres dans une case.
- La lettre tapée doit apparaître en s’agrandissant (utiliser la classe `ScaleTransition` en mettant un `ChangeListener` à `square.textProperty()`).
- Un clic sur une flèche déplace la case courante sur la case suivante dans la direction sélectionnée, autant que possible. La direction courante est sauvegardée.
- Un clic sur la touche “Entrée” colorie en vert le fond de chacune des cases dans lesquelles la lettre proposée coïncide avec la solution (bien sûr, la couleur verte doit disparaître si une nouvelle lettre est saisie dans cette case plus tard.)
- Un clic sur le bouton “effacer” (`KeyCode.BACK_SPACE`) efface la lettre de la case courante, tandis que le curseur se déplace dans la direction opposée à la direction courante (quand c’est possible bien sûr).
- “Ctrl” + “W” doit fermer la fenêtre courante.

```
1 public class Clue {
2     private String clue;
3     private int row;
4     private int column;
5     private boolean horizontal;
6
7     public Clue(String clue, int row, int column, boolean horizontal)
8
9     public String getClue()
10
11     public int getColumn()
12
13     public int getRow()
14
15     @Override public String toString()
16 }
```

```
1 public class Crossword extends Grid<CrosswordSquare> {
2
3     private ObservableList<Clue> verticalClues;
4     private ObservableList<Clue> horizontalClues;
5
6     private Crossword(int height, int width)
7
8     public StringProperty propositionProperty(int row, int column)
9
10    public boolean isBlackSquare(int row, int column)
11
12    public void setBlackSquare(int row, int column, boolean black)
13
14    public char getSolution(int row, int column)
15 }
```

```
16  public void setSolution(int row, int column, char solution)
17
18  public char getProposition(int row, int column)
19
20  public void setProposition(int row, int column, char proposition)
21
22  public String getDefinition(int row, int column, boolean horizontal)
23
24  public void setDefinition(int row, int column, boolean horizontal,
25                          String definition)
26
27  public ObservableList<Clue> getVerticalClues()
28
29  public ObservableList<Clue> getHorizontalClues()
30
31  public static Crossword createPuzzle(Database database,
32                                     int puzzleNumber)
33  public void printProposition()
34
35  public void printSolution()
36  }
```

Bien sûr, étant donné que la taille de la grille du jeu n'est pas connu d'avance, elle doit être remplie dans le fichier java.

Les packages doivent **impérativement** s'appeler `nometudiant1.nometudiant2.crossword` et il faut envoyer l'archive zippée du répertoire `src` du projet.

La date limite est le vendredi 22 mars.