



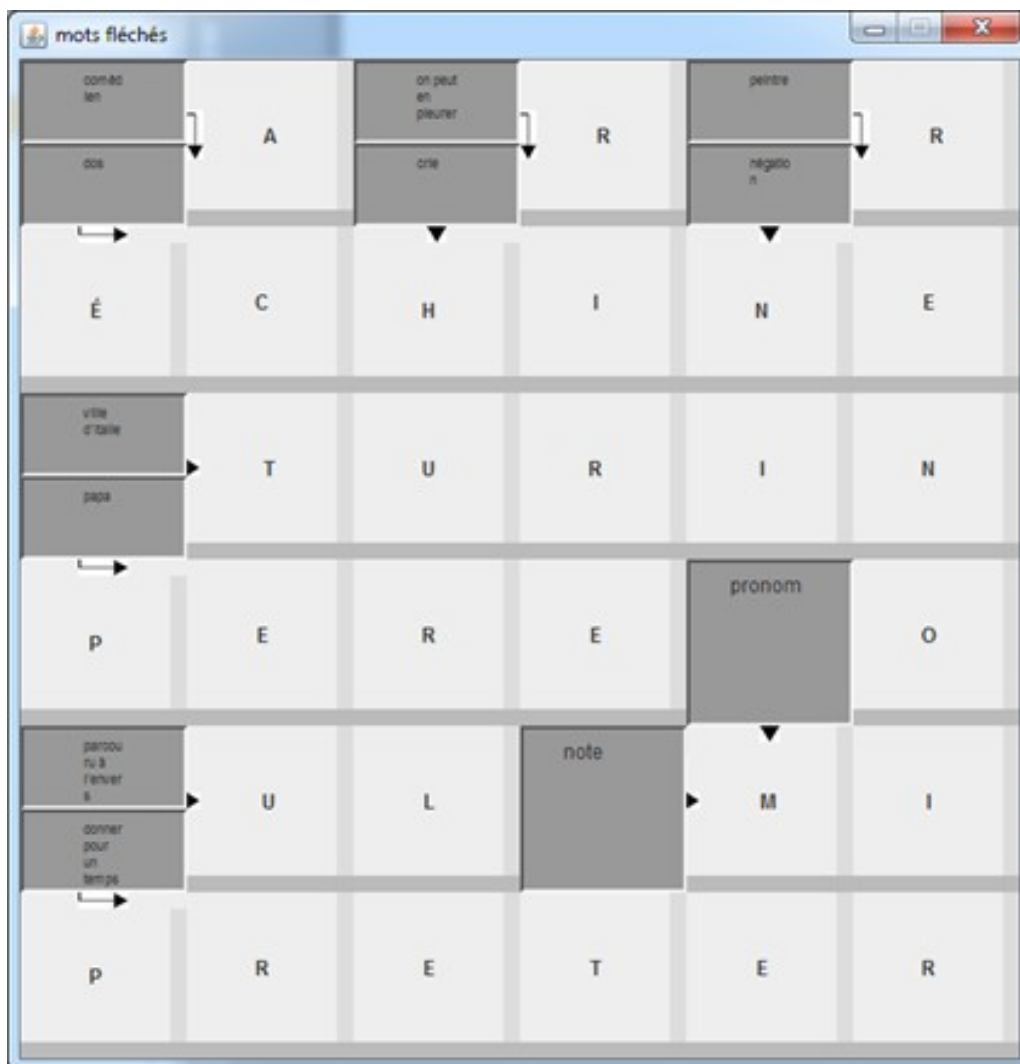
## - mots fléchés -

# Projet

Objectif du projet	Réalisation d'une application
Pré requis	Connaissances avancées en programmation
Ressources à disposition	Eclipse EDI – JetBrains IntelliJ IDEA
Capacités / compétences visées	Maîtriser les différentes étapes de réalisation d'un logiciel
Travail demandé	Lire l'énoncé, appliquer les consignes
Suivi du professeur	Suivi du déroulement du travail des étudiants

### Présentation

L'objectif du travail est la réalisation d'une application permettant d'accompagner la conception de grilles de mots fléchés. Quand une grille est terminée, l'application permet d'imprimer la grille sans les définitions afin qu'une autre personne puisse s'amuser à la remplir.



Les différentes fonctionnalités de l'application sont les suivantes :

### 1. Création d'une grille

Création d'une grille carrée de  $n \times n$  cases ou d'une grille rectangulaire de  $n \times m$  cases

Le positionnement des cases contenant les définitions de mots doit pouvoir être fait au hasard. Les cases de définition permettent d'indiquer si le mot s'écrit à droite de la case (à l'horizontale) ou en dessous de la case (à la verticale). La définition elle-même contient une indication (flèche) indiquant l'orientation du mot. A cette indication il est possible d'ajouter une définition textuelle.

**Évolution :** *(uniquement si les fonctionnalités de bases sont opérationnelles)*

- à la place de la définition textuelle prévoir un dessin correspondant au mot à trouver
- Le positionnement des cases contenant les définitions de mots peut avec une certaine régularité, avec un pourcentage donné de définitions par rapport au nombre de cases total

### 2. Définitions

La grille de travail permet d'ajouter et de supprimer des définitions selon le contexte des cases alentour, qu'elles soient remplies ou pas. Vous trouverez en annexe les différents cas de figure qui peuvent être rencontrés.

### 3. Support d'un dictionnaire

L'application se base sur l'utilisation d'un fichier contenant l'ensemble des mots qui pourront être utilisés. Un fichier initial est fourni, il contient des mots triés dans l'ordre alphabétique. L'objectif est que l'utilisateur puisse chercher un mot dans ce fichier selon certains critères, par exemple, un mot d'un nombre de lettres donné ( $n$  lettres), un mot commençant par une lettre donnée (lettre  $i$ ), un mot d'un nombre de lettres donné et contenant une certaine lettre à une position donnée ( $n$  lettres avec la lettre  $i$  en position  $p$ ), un mot avec les lettres  $i_1, i_2, i_n$  en position  $p_1, p_2, p_n$ .

### 4. Support graphique

La conception d'une grille de jeu nécessite l'utilisation d'un environnement graphique permettant de positionner une définition, indiquer sa direction et indiquer le nombre de lettres.

Votre application utilisera les librairies Swing ou Java FX

**Évolution :**

- Des fonctionnalités supplémentaires seraient très intéressantes comme la possibilité de déplacer une définition et le mot associé.
- D'autres fonctionnalités plus complexes peuvent être imaginées telle que la vérification des lettres lors du déplacement d'une définition (ex : on déplace le mot horizontal « képi » vers le mot vertical « kiwi », l'application doit vérifier que la lettre au croisement des mots est commune comme « k » ou le premier ou le deuxième « i »).
- Une autre fonctionnalité peut être la possibilité d'agrandir la grille dans un sens ou dans l'autre.

## 5. Évolutivité

Votre application devra mettre en œuvre le patron MVC afin que votre application puisse évoluer de différents points de vue :

- Logique métier : la réutiliser pour une autre application
- Modèle de données : remplacer le dictionnaire sous forme de fichier par une base de données
- Remplacement de la vue Swing ou Java FX par une vue HTML/CSS

### ATTENTION :

- Le développement de la vue ne sera autorisé par le professeur que dans la mesure où les parties modèle et contrôleur sont déjà fonctionnelles

**Remarques :** Différences entre la vue et le modèle :

- les mots disponibles dans le dictionnaire font partie du modèle de données
- les mots qui apparaissent dans la grille graphique (*la vue*) sont une représentation **visuelle** d'un tableau de données (*le modèle*) : la « grille visuelle » est mise à jour selon les informations présentes dans ce tableau de données qu'on appellera « grille virtuelle ». De cette façon, le contrôleur peut aller chercher des données dans *le modèle* pour demander de rafraîchir l'affichage des données dans *la vue*. Il fera cela via l'utilisation de méthodes spécifiques au *modèle* et à la *vue*. Ainsi, le fonctionnement est maintenu même si le format des données change où si l'interface graphique change

## 6. Étapes

- **L'ordre des étapes donné doit être respecté**
  - Pour chaque étape, votre document contiendra des schémas ou diagrammes. Un schéma réalisé sur papier est tout à fait suffisant dans la mesure où il est lisible
  - Chaque document
    - devra être remis à la fin du créneau dans lequel il est défini et fera l'objet d'une note
    - Devra être remis au format électronique
    - terminé en avance devra être validé par le professeur qui vous autorisera à passer à la suite
1. Création d'un document d'analyse :
    1. Il devra décrire les différents scénarios afin de faire apparaître les fonctionnalités qui seront nécessaires
    2. Il devra reprendre le présent document et lister l'ensemble des fonctionnalités à développer comme des futurs livrables
    3. il contiendra un diagramme de cas d'utilisation
  2. Création d'un document de conception :
    1. Il devra recenser les classes nécessaires, par exemple une classe « grille » avec des attributs hauteur et largeur, une classe « mot » avec un attribut longueur, une classe « définition » avec un attribut intitulé et un attribut direction, etc.

2. il contiendra
  1. un diagramme de classes montrant la structure générale de votre application
  2. un diagramme de classes détaillant les méthodes prévues sur un deuxième document si la taille du premier ne permet pas de le faire
3. Planning prévisionnel

	<b>Matin</b>	<b>Après midi</b>
Jour 1	cours/TP	<ul style="list-style-type: none"><li>• Document d'analyse</li><li>• diagramme de cas d'utilisation</li></ul>
Jour 2	cours/TP	<ul style="list-style-type: none"><li>• Document de conception</li><li>• diagramme de classes</li></ul>
Jour 3	<ul style="list-style-type: none"><li>• Développement de la classe dictionnaire</li><li>• tests unitaires</li></ul>	<ul style="list-style-type: none"><li>• Développement de la classe de gestion de grille virtuelle</li><li>• tests unitaires</li></ul>
Jour 4	cours/TP	Développement de la classe de grille graphique et gestion des événements
Jour 5	<ul style="list-style-type: none"><li>• Développement du contrôleur</li><li>• liaison du modèle avec la vue</li></ul>	<ul style="list-style-type: none"><li>• Développement du contrôleur</li><li>• liaison du modèle avec la vue</li><li>• remise de l'application</li></ul>

**ATTENTION** : l'idée est que votre travail suive un déroulement itératif et incrémental afin que votre application soit fonctionnelle tout au long du cycle de développement. En conséquence il est nécessaire de construire des modules opérationnels puis de les compléter par de nouveaux modules

## 7. Annexe

Voici des exemples de construction d'une grille qui devront être mis en œuvre :



Figure 1 Le clic sur la case 1,1 propose les actions contextuelles associées



Figure 2 Le clic sur la case 4,5 propose des actions relatives au contexte de la grille

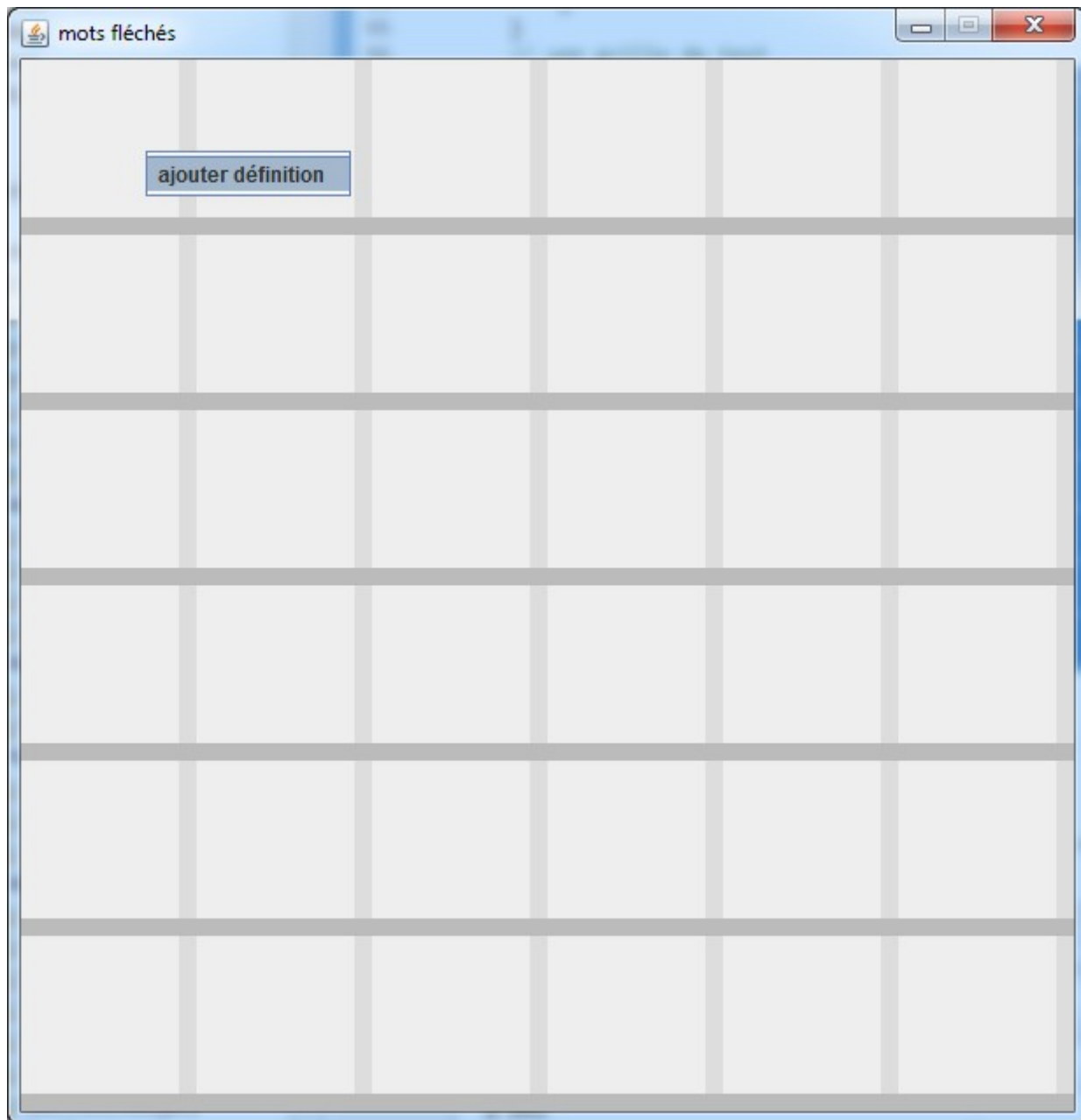
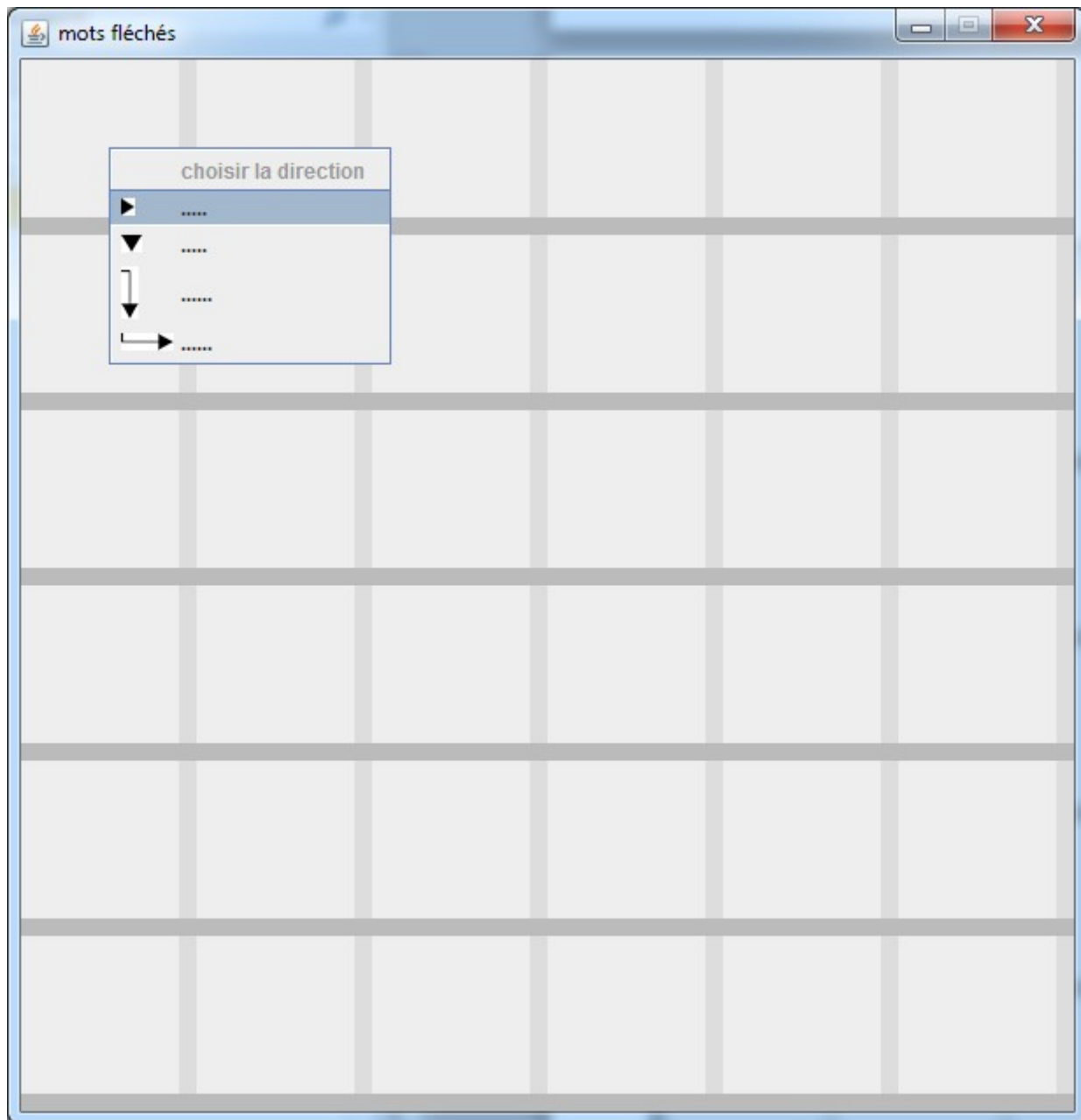


Figure 3 le clic sur la case vide 1.1 propose d'ajouter une définition



**Figure 4**La case 1,1 étant vide, les 4 possibilités possibles d'ajout de définitions sont proposées