



Conception détaillée

Nom de l'entreprise : UPJV

Nom du projet : Grille de rappel

Personnes à contacter : Mme Rosselle Maryline

Sommaire

I.	Introduction.....	3
II.	Conception détaillée.....	3
III.	Un peu de code.....	4
IV.	Cahier de tests.....	6
V.	Conclusion.....	6

I. Introduction

Dans ce livrable nous allons désormais rentrer plus en profondeur dans le projet. Nous allons premièrement présenter les choix que nous avons réalisés à la suite du cahier des charges et de la partie analyse. Puis nous montrerons un peu de code que nous jugeons pertinent de notre application. Enfin nous réaliserons un cahier de tests afin de montrer à notre client comment mettre en pratique certaines fonctionnalités de l'application.

II. Conception détaillée

La grille de rappel est programmée en JAVA pour des raisons de facilité que nous avons évoquée lors de la phase d'analyse.

Deux boutons sont présents sur l'écran d'accueil : un pour ouvrir un fichier existant, l'autre pour créer un fichier de questions.

Les questions sont enregistrées dans un fichier texte que l'application génère. Le joueur rentre ses questions une par une dans ce fichier. Elles sont enregistrées dans une ArrayList . Elles sont écrites dans le fichier à l'aide de la fonction Path file (voir partie codage).

Si un fichier existe déjà, il est possible de le récupérer via une fonction permettant d'ouvrir le répertoire du périphérique sur lequel l'application est lancée.

La classe RandomAccessFile permettra de récupérer des lignes, de les lire et de les afficher à partir d'un fichier.

La résolution de l'application est de 1300/800.

Le timer se situe dans la partie supérieure droite de l'écran lorsque le jeu est lancé. Le temps s'écoule jusqu'à atteindre 00 : 00.

Il s'affiche de la façon suivante :



Temps restant : 04:53

Un score apparaîtra sur l'écran de jeu. Il s'incrémentera en fonction du nombre de points que rapporte la question et du choix de proposition.

L'événement MouseListener permet d'ouvrir une petite fenêtre, par-dessus la fenêtre du jeu, qui contient toutes les propositions à une question (exemple : « Je ne sais pas », ...) lorsque l'utilisateur réalise un clic gauche dans un rectangle qui contient une question. Les rectangles sont réalisés avec la classe « Graphics ».

Les réponses de l'utilisateur sont enregistrées dans le fichier texte correspondant à la réponse choisie. A la fin le contenu de ces fichiers est affiché sur l'écran de fin de partie. Lorsque la partie sera terminée il suffira de supprimer ces fichiers, puis de les recréer avec les mêmes noms.

III. Un peu de code

```
public void actionPerformed(ActionEvent ae) {
    String fileName = field.getText();
    String encoding = "UTF-8";
    try{
        PrintWriter writer = new PrintWriter(fileName, encoding);
        writer.close();
        JOptionPane.showMessageDialog(null,
            "Fichier créer !");
        valid = true;
    }
    catch (IOException e){
        System.out.println("An error occurred.");
        e.printStackTrace();
    }
}
```

La fonction New_File permet de créer un fichier texte. Nous allons récupérer le texte du JTextField et à l'aide de la fonction PrintWriter un nouveau fichier vierge est créer.

```
try {
    Path file = Paths.get(fileName);
    Files.write(file, lines, StandardCharsets.UTF_8);
}
catch (IOException e){
    System.out.println("An error occurred.");
    e.printStackTrace();
}
```

La fonction Write_File permet d'écrire dans le fichier. Nous allons récupérer le nom du fichier à l'aide de la classe Path. La fonction File.write, prend en paramètre le nom du fichier la liste des questions et l'écriture en UTF8. Ainsi un fichier texte remplis des 12 questions en 12 lignes est créé.

```

public void Count() {

    timer = new Timer(1000, new ActionListener() {

        @Override
        public void actionPerformed (ActionEvent e) {

            seconde--;
            SSeconde = Format.format(seconde);
            SMinute = Format.format(minute);
            l.setText("Temps restant : " + SMinute + ":" + SSeconde);

            if (seconde==0) {
                seconde= 59;
                minute--;
                SSeconde = Format.format(seconde);
                SMinute = Format.format(minute);
                l.setText("Temps restant : " + SMinute + ":" + SSeconde);
            }
            if (minute==0 && seconde == 0) {
                timer.stop();
                System.exit(1);
            }
        }
    });
}

```

La fonction Count() est la fonction qui gère le timer. Elle s'occupe de faire les modifications nécessaires. Par exemple lorsque le timer arrive à 04 : 00, il doit faire passer le timer à 03 : 59 ! Pour le moment l'application se ferme lorsque le timer arrive à 00 : 00 (cela sera modifié plus tard). Les affichages se font dans une autre partie du code !

```

public class FileChooser1 {

    public static void main(String[] args) {

        JFileChooser jfc = new JFileChooser(FileSystemView.getFileSystemView().getHomeDirectory());

        int returnValue = jfc.showOpenDialog(null);

        if (returnValue == JFileChooser.APPROVE_OPTION) {
            File selectedFile = jfc.getSelectedFile();
            System.out.println(selectedFile.getAbsolutePath());
        }

    }

}

```

La fonction fileChooser1() est la fonction qui s'occupe d'obtenir le répertoire du périphérique sur lequel se trouve l'utilisateur actuellement. On peut alors afficher le chemin du fichier sélectionné.

IV. Cahier de tests

Imaginons qu'un étudiant utilise notre application pour réviser un cours d'informatique. Il accède à la page d'accueil ou deux choix s'offre à lui :

- Créer une session
- Ouvrir une session

Il décide de créer une session. Une nouvelle page s'affiche et il rentre le nom du fichier texte qu'il veut créer et une fois cela fait il rentre les questions une par une en prenant soin de valider à chaque fois. Par exemple il rentre le nom « RevisionMO.txt » et rentre plusieurs questions comme « Qu'est ce qu'un diagramme d'activité ? », « Comment construire un diagramme d'activité » et ainsi de suite.

Retour au menu ou il ouvre une session, il choisit le fichier qui vient de créer et la page de questions s'ouvre. 12 cases cliquables apparaissent en cliquant sur chacune des cases une petite fenêtre s'ouvre faisant apparaître les 4 choix de réponses de la question. Une fois les 5 minutes écoulées ou les 12 questions répondues les révisions prennent fin et le score s'affiche ainsi que les questions rangées dans un tableau en fonction de la réponse de l'étudiant. Il a ensuite la possibilité de revenir au menu et recommencer.

V. Conclusion

A travers ce livrable le client peut avoir une idée beaucoup plus précise du projet. Les fonctionnalités ainsi que l'application sont décrites avec des détails « en profondeur ». On se penche désormais sur le cœur même du sujet. Ainsi il est maintenant possible de se faire une idée de à quoi ressemblera l'application en général.