BOULANGER Raphaël et TURPIN Aurélien

Spécifications :

1. Au démarrage de l’application, une fenêtre console s’ouvre contentant un menu avec la liste des 3 sections (Fise1, Fise2, Fise3). L’utilisateur peut entrer un numéro (entre 1 et 3) pour choisir sa section. Dans le cadre de notre projet, nous aurons accès uniquement à la section FISE1. De plus, dans un souci de confort pour l’utilisateur, l’affichage de la console est réinitialisé si l’utilisateur entre un numéro autre que le 1.
2. Si la Fise1 est choisie, un 2nd menu s’affiche avec la liste des groupes de TD de cette section. L’utilisateur peut alors entrer le numéro correspondant au groupe de TD dont il veut faire l’appel (entre 1 et 5 car il y a 5 groupes en Fise1). L’utilisateur peut aussi retourner au menu précédent en entrant le numéro 0 : cette fonction a été rajoutée dans l’optique où l’on ait accès aux sections Fise2 et Fise3 pour pouvoir retourner au menu précédent si l’utilisateur s’était trompé de section.
3. Après avoir choisi un groupe de TD, une fenêtre graphique s’ouvre dans laquelle s’affiche une page composée de vignettes correspondant à chaque étudiant du groupe avec sa photo, son nom et son prénom. L’utilisateur peut ainsi changer l’état de présence d’un étudiant en cliquant sur la vignette qui lui est associée (par défaut tous les étudiants sont notés « présent »), ce qui permet à l’utilisateur de corriger aisément cet état s’il se trompe. Une croix rouge indique que l’étudiant sera noté absent, et un V vert indique qu’il sera noté présent, (ces symboles sont affichés en haut à droite des vignettes étudiantes), le changement d’état de présence d’un étudiant entraine un rafraichissement de la page active pour permettre l’actualisation de ces symboles. Chaque page affiche au maximum 10 étudiants, pour naviguer vers les autres pages, l’utilisateur peut cliquer sur des boutons NEXT et/ou PREVIOUS situés dans les coins inférieurs des pages. A la dernière page, un bouton END est disponible dans le coin inférieur droit pour que l’utilisateur puisse signifier qu’il a terminé la saisie.
4. Après avoir cliqué sur le bouton END, une page apparait indiquant que la saisie des élèves absents est terminée et que la fiche d’absence (en format .txt) a été créée. Celle-ci s’ouvre peu après la fenêtre graphique et indique quelques statistiques et les élèves absents durant la séance.

Explications des classes :

Les 4 classes Eleve, Groupe, Section, et Telecom s’occupent chacune d’un niveau de répartition des élèves et stockent les caractéristiques qui y sont liées, et les objets plus restrictifs qui les composent. Par exemple Telecom contient les sections qui le compose, Section contient les groupes qui lui appartiennent et Groupe contient tous les élèves qui le compose. La classe la plus « basse » dans cette hiérarchie, la classe Eleve, va contenir des champs correspondant aux caractéristiques de chaque élève du groupe étudié, récupérées dans le fichier texte (à savoir, le nom, prénom, groupe et nom du fichier photo), et les classes au-dessus dans la hiérarchie contiennent des vecteurs avec les objets qui lui leur sont directement inférieurs hiérarchiquement : un Groupe possède un vecteur d’Eleves par exemple. La particularité de la classe Telecom est qu’elle va également s’occuper de l’affichage dans les fenêtres consoles en fonction des choix, des noms de groupe, etc. Les méthodes de Telecom vont alors descendre dans la hiérarchie des classes pour aller chercher les informations nécessaires pour l’affichage dans la fenêtre console.

Telecom ne va pas jusqu’à chercher les informations des élèves, ce sont les classes qui s’occupent de l’affichage des élèves sur les fenêtres graphiques qui vont faire cela : ces classes sont Image, Page, et Seance. De même ici, on a une hiérarchie à 3 membres qui s’imbriquent les uns dans les autres : Image s’occupent de la création de chaque vignette qui correspond à chacun des élèves du groupe sélectionné, Page de la création d’une page de 10 vignettes (donc de 10 objets Image), et Seance qui va lancer l’affichage de ces pages, permettre de naviguer entre elles, et enfin de sauvegarder les informations nécessaires dans le fichier texte. Encore une fois, on a un vecteur d’objets Page dans Seance, et dans Page un vecteur d’objets Image ainsi qu’un vecteur d’objets Eleve pour associer les objets Image aux objets Eleve pour aller récupérer les informations de chaque Eleve pour les afficher ensuite.

La dernière classe est un peu à part, c’est la classe Memoire, utilisée pour des ajouts facultatifs que l’on a faits : les statistiques sur les fiches d’absence. Cette classe permet de regrouper toutes les fonctions allant lire les fichiers texte des séances précédentes et calculant les statistiques à afficher. Créer une classe ici n’était pas absolument nécessaire mais cela permettait de regrouper toutes ces fonctions en un objet. Cette classe communique avec la classe Seance uniquement, puisque que c’est elle qui s’occupe d’aller écrire dans le fichier texte où l’on sauvegarde la fiche d’absence. Ici la relation liant Memoire et Seance n’est donc qu’une association simple, au contraire des classes citées plus haut qui étaient reliées par des relations de composition (souvent multiples) comme indiqué dans le diagramme de classes.