Louis THIDET - Lucie DELAY

Université Paris Cité

Rapport du projet Java

Construction du jeu-vidéo « Versus »

PRESENTATION DU PROGRAMME

Présentation du jeu

Notre projet a consisté à réaliser une sorte de jeu de combat au tour par tour opposant deux joueurs en local. Son objectif a été de reproduire les combats que l'on peut retrouver dans divers jeux-vidéos, tels que les Final Fantasy ou les Dragon Quest. Chaque joueur dans notre jeu dispose de quatre personnages, dont il choisit les classes, et son objectif est de vaincre son adversaire avec les ressources qu'il choisit et dont il dispose : outre les compétences propres aux personnages, les joueurs disposent d'un certain nombre d'objets qui permettent d'augmenter les statistiques de ces derniers. Les classes disposent chacune de leurs propres spécificités (statistiques, sorts, apparence).

Contenu du programme

Le programme a été construit à l'aide de l'IDE Netbeans, de Ant et de la librairie graphique Swing. Il utilise également la librairie JACo MP3 Player, qui permet de charger de la musique, et la librairie KGradientPanel, qui permet de créer des JPanels dont le fond consiste en un dégradé de deux couleurs. Le programme est composé de trois paquets : Interface, CharClasses et Items.

- Interface contient le corps du programme. Il contient la classe Versus, qui est à la fois le main et la JFrame dans laquelle se lance le programme, et les classes MainMenu, Options, Preparation et Combat, qui sont des JPanels composant les différents menus.
- CharClasses comprend toutes les classes relatives à la construction des personnages. Chaque classe de personnage correspond à une classe du programme. Tandis que certaines d'entre elles héritent directement de la classe CharClass, qui définit la structure de tous les personnages, d'autres héritent de la classe Mage, ou encore de la classe OffensiveMage. Les 6 classes de personnages sont : Warrior, Monk, Thief, RedMage, WhiteMage et BlackMage
- Items comprend toutes les classes relatives à la construction des objets que les joueurs utilisent dans leurs combats

Lorsque le programme se lance, l'utilisateur voit le JPanel MainMenu, depuis lequel il peut accéder soit au JPanel Options, soit au JPanel Preparation. Le JPanel Options permet de changer les couleurs des différents sous-panels du jeu, tandis que le JPanel Preparation est celui depuis lequel les deux joueurs vont choisir les classes de leurs personnages mais aussi les objets pour les améliorer. C'est depuis le JPanel Preparation que l'on accède au JPanel Combat, où le jeu démarre.

FONCTIONNEMENT ET LIMITES

Fonctionnement du programme

Au lancement du programme sont créées pour chaque classe du paquet Interface une instance, que le programme utilisera tout le long de la session. Ensuite est crée ou chargé (s'il existe déjà) le fichier versus.ini, dans lequel figurent les diverses propriétés qui gèrent les couleurs des menus. L'existence de ce fichier évite la réassignation des options choisies à chaque démarrage du jeu.

Lorsque les utilisateurs – que l'on suppose être deux, puisque le jeu est fait pour deux personnes devant un seul ordinateur – parviennent au JPanel Preparation, après avoir appuyé sur le bouton playButton du JPanel MainMenu, ils font face à des JComboBox permettant la sélection des personnages de leurs équipes. Lorsqu'ils glissent la souris sur les JLabel qui représentent leurs personnages, une description de ceux-ci apparaît à l'écran. Une fois qu'ils ont terminé de choisir leurs personnages, les joueurs ont la possibilité de sélectionner des objets via des JList. Chaque objet coûte un certain nombre de crédits et les joueurs disposent chacun d'un maximum de 10 crédits. Une fois le choix des objets et personnages accomplis, les joueurs cliquent chacun sur « Ready », puis sur le bouton preparation_fightButton, ce qui les dirige vers le JPanel combat.

Le jeu repose sur un certain nombre de variables globales qui ont besoin d'être réinitialisées à chaque combat, et par conséquent elles ne sont initialisées ni dans le constructeur du JPanel Combat ni lors de leur déclaration, mais à l'intérieur d'une méthode appelée gameSetUp(), qui se déclenche avec le bouton preparation_fightButton.

Les combats se déroulent de la manière suivante : un joueur est sélectionné au hasard et ses quatre personnages peuvent effectuer une action. Ensuite, c'est au tour de l'autre joueur de faire jouer ses personnages. Le combat se poursuit tant que les deux équipes ont toujours au moins un personnage en vie.

Les combats reposent sur la méthode triggerGameUpdate(), qui se déclenche à chaque fois qu'un joueur produit une action. Chaque tour est composé de quatre sous-tours (un pour chaque personnage), qui sont complétés lorsqu'est effectuée une action ou lorsqu'un personnage est déjà mort. Une vérification du nombre de sous-tours effectués se produit à chaque lancement de triggerGameUpdate(), pour que le programme sache si c'est au tour de l'autre joueur ou non de jouer, et si un joueur a gagné ou non.

Il existe quatre actions : les attaques simples, les sorts (ou abilités), la défense et l'usage des objets, lesquelles existent par le biais des méthodes attackedCharacter(), castSpell(), defendAlly() et useItem() et les variales globales booléennes isAttacking, isCasting, isDefending et isUsingItem.

Les personnages des joueurs sont tous représentés par un JLabel, sur lequel il faut cliquer pour accomplir une action. Lorsqu'un clic est produit sur le JLabel d'un personnage, se déclenche la méthode targetedCharacter(), qui vérifie si l'une des variables isAttacking, isCasting, isDefending et isUsingItem est passée sur true, et gère le lancement des méthodes attackedCharacter(), castSpell(), defendAlly() et useItem(). Si l'une des variables booléennes citées est sur true et que toutes les conditions requises pour que l'action lui correspondant s'accomplisse sont enclenchées (Par exemple, si un joueur cherche un défendre un personnage, il faut que le personnage défenseur soit allié du personnage à défendre), alors le clic sera un succès, et un sous-tour sera passé.

Limites

- Nous avions initialement intégré une introduction vidéo au début de notre jeu, mais nous l'avons retirée parce qu'elle utilisait la librairie VLCJ, laquelle pour fonctionner nécessite impérativement que soit installé sur l'ordinateur le logiciel VLC.
- Nous souhaitions intégrer la possibilité de changer la dimension de la fenêtre du jeu, mais nous n'avons pas eu le temps de trouver un moyen efficace pour mettre en place cette fonctionnalité pour tous les menus (Il fallait non seulement redimensionner les images, mais l'intégralité des éléments affichés à l'écran). C'est la raison pour laquelle les boutons de changement de dimension du menu Options sont désactivés.
- Le programme produit un fichier de configuration versus.ini. Si le programme est lancé sur Windows il sera rangé dans un dossier Versus crée dans le repertoire des Documents. S'il est lancé sur MacOS ou Linux, il sera rangé dans un dossier caché .versus dans le home de l'utilisateur. On ne savait pas où ranger le fichier dans le cas des autres OS, donc si le programme est lancé sur un OS autre que ceux mentionnés, il se fermera automatiquement. Le programme charge directement les valeurs contenues dans versus.ini lors de son lancement.