|  |  |  |
| --- | --- | --- |
| **Projet Algorithme**  **2e année – Semestre 2**  **Date : 29 Avril 2020** |  |  |

|  |
| --- |
| Projet Mastermind |

**Etudiants :**

BECQUET Simon, Groupe n° 54

GHARBI Reda, Groupe n° 54

**Enseignants :**

Wolf Christian

* Cahier des Charges et principe de codes

Jouer au Mastermind :

- La sélection des couleurs se fait par un groupe de boutons pour permet au joueur de placer ses différentes couleurs.

-Ces boutons sont basés sur un groupe de boutons qui vont permettre d’interagir entre le joueur et les boutons.

-Evaluation de la combinaison avec des compteurs qui nous permettent de savoir le nombre de couleurs bien placées et le nombre de couleurs présentes dans la solution cachée.

-Le joueur pourra changer son choix de couleurs tant qu’il n’a pas fini sa combinaison. Après avoir validé son choix de combinaison il ne pourra plus changer les couleurs précédentes.

-Fin du jeu lorsqu’il a trouvé la combinaison exacte cachée ou dans le cas où il atteint le nombre maximal de tentatives. Cette fonctionnalité se fera par un Jdialog qui permet de stopper les interactions possibles avec les autres fenêtres lors de la fin du jeu.

Mode Ordinateur :

-Génération de toutes les combinaisons possibles en utilisant un incrémenteur (système de retenue pour avoir toutes les possibilités).

-Choix d’une combinaison aléatoire parmi toutes celles possibles et ensuite la mise en place de la procédure d’élimination des combinaisons incompatibles. Pour cela nous avons une méthode qui permet de passer d’un tableau de nombres à un tableau de couleurs.

- L’utilisation de thread qui permet un calcul en tache de fond ; la mise en place des linkelist et d’itérateur pour que le calcul se fasse rapidement. La gestion des threads s’est faite avec l’utilisation d’un exécuteur du type newSingleThreadExecutor.

-Possibilité d’activer le mode ordinateur à la suite de quelques tentatives.

-Temps d’arrêt entre chaque tentative de l’ordinateur qui permet à l’utilisateur de lire et comprendre un peu le fonctionnement de la résolution. Ce temps est variable selon la taille de combinaison.

Panneau Option :

-Possibilité de changer le nombre de couleurs dans la palette.

-Permettre l’utilisation de couleurs multiples dans une combinaison.

-Choisir la taille de la combinaison que l’on souhaite résoudre.

-Utilisation d’un bouton pour avoir les paramètres originaux du Mastermind.

Fonction en Jeu :

-Possibilité d’afficher un panneau d’aide qui permet d’aider le joueur lors de ces tentatives.

- Le mode ordinateur permet de résoudre le système.

-Affichage du temps de jeu et du nombre de coups pour une partie donnée.

-Bouton disponible pour recommencer une partie de zéro tout en restant sur la fenêtre de jeu.

-Bouton qui permet de révéler le code de la combinaison cachée. Ce dernier permet à l’utilisateur d’essayer de comprendre ses différentes erreurs lors de ces tentatives.

Interface Graphique des fenêtres :

-Toutes les fenêtres possèdent un redimensionnement dynamique. Ils ont été mis en place avec des layout qui permettent de bien positionner les différents composants de la fenêtre. Nous avons utilisées le code suivant «Toolkit.getDefauktToolkit().setDynamicLayout(false) ; » qui nous a permis de régler certains problèmes aux niveaux dynamiques.

- Affichage des règles s’effectuent au début de l’arrivée du joueur pour lui expliquer le fonctionnement ou dans l’accueil s’il a besoin de plus d’informations.

-Proportion minimale et maximale des différentes fenêtres qui sont défini avec la hauteur et la largeur de la fenêtre d’accueil.

-Toutes les fenêtres possèdent une taille minimum qui permet de voir dans de bonnes conditions quel que soit le redimensionnement fait.

-Redimensionnement dynamique des polices d’écritures notamment pour les titres et les commentaires dans certaines fenêtres.

-Utilisation de boutons circulaires avec la méthode contraint et la variables Shape.

Esthétique :

-Mise en place de boutons circulaires dans le panneau jeu.

-Utilisation de couleurs dégradées pour un meilleur visuel. Il a été fait à l’aide d’une méthode GradientPaint et LinearGradiantPaint. De plus pour chaque composant nous l’avons héritée de cette méthode pour avoir le même affichage entre tous les mêmes types d’objets. Pour cela nous avons juste à changer une ligne de codes entre les différents objets.

-Utilisation de tableaux de boutons et de labels pour uniformiser les différents composants et réduire la taille du code. Il nous a également aidé pour la rapidité du placement des objets. Ces méthodes ont été faites avec des InnerClass.

-Possibilité de passer en mode sombre. (Couleur plus sobre)

Problèmes :

-Les couleurs par défaut sont différentes selon le système d’exploitation. Nous pouvons corriger ce problème à l’aide du LookAndFieel ou de codées les couleurs particulières en hexadécimales.

-Sur les ordinateurs Apple, nous avons un problème d’affichage avec l’apparition de carrée rectangulaire derrière les boutons circulaires.

-La police par défaut (Sérif) est non présente sous les versions Linux cela crées de petits gènes.

-Latence lors de l’activation du mode ordinateur si la puissance de l’ordinateur n’est pas suffisante ou lorsqu’on utilise une grande taille de combinaison. (Problème réglé avec une meilleure utilisation des threads)

-Problème aux niveaux du timer avec un peu de freeze lors de la fin du jeu.

-Affichage de la couleur des boutons lorsqu’ils ont activé « isPress » (couleurs anormales que dans le mode sombre)

-Affichage d’un pixel blanc aux niveaux des boutons ronds.

Améliorations possibles :

-Organisation par package des différentes classes.

-Mode jeu à rajouter dans le code. Ce mode se baserait sur un système ou plus l’on joue aux jeux plus la difficulté augmente. Par exemple nous pouvons réduire le nombre de tentatives, augmenter le nombre de couleurs dans la palette, augmenter la taille de la combinaison …

-Mise en place d’un tableau de scores selon différentes catégories par exemple le nombre de coups fait pour résoudre le jeu, mise en place du mode multicolore, temps… Ce tableau permettra de comparer les meilleurs joueurs du Mastermind ; ce classement est stocké dans un fichier externe (JTable).

-Mode de jeu a rajouté qui se base sur un mode multijoueur qui consiste à un duel entre deux joueurs. Ce système serait basé en deux manches gagnantes où le joueur doit déterminer la combinaison en le moins de coups possible. Le joueur adversaire choisirait la combinaison que le joueur doit trouver.

-Raccourci claviers avec l’utilisation des fonctions KeyListener, KeyReleaesed, SetUndecorated(true) ou encore MaximisedBoth qui permettrait de passer en plein écran ou d’autres fonctionnalités.

-Icone du programme avec Tray.

-Affichage des crédits en splash.

-Ajout de musique de fonds.

-Mise en place d’un set de couleurs pour les daltoniens.

-Gestion des langues pour les langues les plus parlées dans le monde.

-Mettre en place des captures d’écrans pour mieux expliquer les règles.

-Possibilité de personnaliser les JSliders et les JCheckbox du panneau options.

* Bibliographie non exhaustive

Vous avez une bibliographie exhaustive dans le fichier readme.

FontDynamicResizing and personalization:

<http://java-sl.com/tip_adapt_label_font_size.html>

<https://docs.oracle.com/javase/8/docs/technotes/guides/2d/spec/j2d-fonts.html>

<https://stackoverflow.com/questions/5652344/how-can-i-use-a-custom-font-in-java>

<https://www.developpez.net/forums/d32258/java/interfaces-graphiques-java/font-charger-police-d-fichier/>

<https://stackoverflow.com/questions/2715118/how-to-change-the-size-of-the-font-of-a-jlabel-to-take-the-maximum-size>

DynamicResizing and JFrame Ratio:

<https://stackoverflow.com/questions/2303305/window-resize-event>

<https://docs.oracle.com/javase/7/docs/api/java/awt/Toolkit.html>

<http://www.java2s.com/Questions_And_Answers/Swing/JFrame/Resize.htm>

<https://stackoverflow.com/questions/27544569/java-how-to-control-jpanel-aspect-ratio>

<https://www.codeproject.com/Tips/5247835/Resizing-Frame-and-Controls-with-the-Same-Ratio-in>

<https://stackoverflow.com/questions/16075022/making-a-jpanel-square>

<http://www.aidewindows.net/materiel_ecran-taille.php>

<https://stackoverflow.com/questions/30198953/how-to-dynamically-control-auto-resize-components-in-java-swing>

<https://stackoverflow.com/questions/27544569/java-how-to-control-jpanel-aspect-ratio>

<https://docs.oracle.com/javase/tutorial/uiswing/events/componentlistener.html>

Multicolor JLabel:

<https://stackoverflow.com/questions/6856120/building-html-in-java-code-only>

<https://htmlcolorcodes.com/fr/>

<https://www.tutorialspoint.com/how-can-we-implement-a-jlabel-text-with-different-color-and-font-in-java>

Layout Managers:

<http://imss-www.upmf-grenoble.fr/prevert/Prog/Java/swing/gestionPlace.html>

<https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

<https://docs.oracle.com/javase/tutorial/uiswing/layout/layoutlist.html>

<http://www.fredosaurus.com/notes-java/GUI/layouts/42boxlayout-spacing.html>

<https://www.cs.tut.fi/lintula/manual/java/tutorial/uiswing/layout/box.html>

<https://www.tutorialspoint.com/swing/swing_springlayout.htm>

<https://stackoverflow.com/questions/21622148/how-can-i-center-align-a->

<https://perso.telecom-paristech.fr/hudry/coursJava/interSwing/box.html>

<https://docs.oracle.com/javase/tutorial/uiswing/layout/problems.html>

<https://stackoverflow.com/questions/22010658/boxlayout-inside-a-borderlayout>

<http://www.java2s.com/Tutorial/Java/0240__Swing/HorizontalAlignmentCENTER.htm>

ContentPane and GUI:

<https://www.lri.fr/~anab/teaching/DevLog/cours5-IG.pdf>

<https://www.jmdoudoux.fr/java/dej/chap-swing.htm>

<http://echo.nextapp.com/content/echo2/doc/api/2.1/public/app/index.html?nextapp/echo2/app/ContentPane.html>

<https://docs.oracle.com/javase/tutorial/uiswing/painting/closer.html>

<https://www.bogotobogo.com/Java/tutorials/javagraphics3.php>

Round Button:

<https://docs.oracle.com/javase/7/docs/api/javax/swing/ButtonModel.html>

<https://happycoding.io/examples/java/swing/circle-button>

<https://www.roseindia.net/tutorial/java/swing/createRoundButton.html>

GitHub:

<https://openclassrooms.com/fr/courses/2342361-gerez-votre-code-avec-git-et-github?status=published>

<http://johnatten.com/2012/09/08/basic-git-command-line-reference-for-windows-users/#BASH-NAVIGATE-FILE-SYSTEM>

<https://github.com/collab-uniba/socialcde4eclipse/wiki/How-to-import-a-GitHub-project-into-Eclipse>

Eclipse

<https://github.com/collab-uniba/socialcde4eclipse/wiki/How-to-import-a-GitHub-project-into-Eclipse>

<https://waytolearnx.com/2018/11/eclipse-les-raccourcis-clavier-pour-programmer-rapidement.html>

<http://patatos.over-blog.com/article-todo-list-sous-eclipse-45699065.html>

Mode PleinEcran :

<https://docs.oracle.com/javase/7/docs/api/java/awt/Frame.html>

<https://stackoverflow.com/questions/11570356/jframe-in-full-screen-java>

<https://www.tutorialspoint.com/how-to-set-fullscreen-mode-for-java-swing-application>

* Diagramme UML
* Carnet de route et Implication du groupe

Nous avons utilisé le logiciel Github avec des commit ceux qui nous permet de dresser un carnet de route très précis :

-Création des boutons ronds et début de la classe combinaison.

-Elaboration de la grille de boutons pour le panneau jeu.

-Début de l’interface graphique avec la création des panneaux options et d’accueil.

-Conception de la classe CarreEval avec principalement l’aspect graphique des carrées.

-Réflexion sur les méthodes pour la classe CarreEval principalement axée sur les compteurs.

-Mise en place de l’interaction des panneaux.

-Création des méthodes HiddenCombiPanel et de la palette de couleurs qui sont présentes dans le panneau jeu.

-Fin des panneaux options et accueil en termes d’implémentation des boutons.

-Génération des méthodes pour le mode ordinateur dont la création des combinaisons possibles.

-Esthétique de tous les panneaux et les widgets.

-Résolution du mode ordinateur.

- Mise en place de la correction des petits problèmes de codes.

La répartition du travail s’est faite avec le logiciel GitHub qui nous a permis de bien avancer de notre côté. Ensuite, nous faisons des récapitulatifs de notre travail sur discord ceux qui étaient bien pratiques pour comprendre les avancées de notre camarade.

Pourcentage de travail :

* Becquet Simon %
* Gharbi Reda %