Logo

Description automatically generated with medium confidence

RAPPORT PROJET JAVA

# PROGRAMMATION ORIENTE OBJET

Théo Crépin

Jules Burette

Anthony Guerimand

Toufik Philippe

Groupe de TP A

Année 2021/2022

Table des matières :

1. Explications du programme
   1. Organigramme et structure du programme ……………………………………………………………3
   2. Détails des Classes et interface utilisées …………………………………………………………………4
   3. Détails des fonctions utilisées ………………………………………………………………………………..5

1. Problèmes rencontrés

1.Utilisation d’un fichier CSV …………………………………………………………………………………..6

2.Interface pour implémenter les phases de jeu ……………………………………………………..6

3.Gestion des exceptions ………………………………………………………………………………………..6

1. Bonus

1.Implémentation du timer ………………………………………………………………………………………7

2.Implémentation de l’IA…………………………………………………………………………………………..7

3.Implémentation du Grand Jeu………………………………………………………………………………..7

2.Implémentation du menu ……………………………………………………………………………………..7

1. Présentation des résultats.

1.présentation des résultats ……………………………………………………………………………………..8

1. Conclusion

1.Répartition des taches…………………………………………………………………………………………….9

2.Perspective d’amélioration……………………………………………………………………………………..9

Présentation du projet

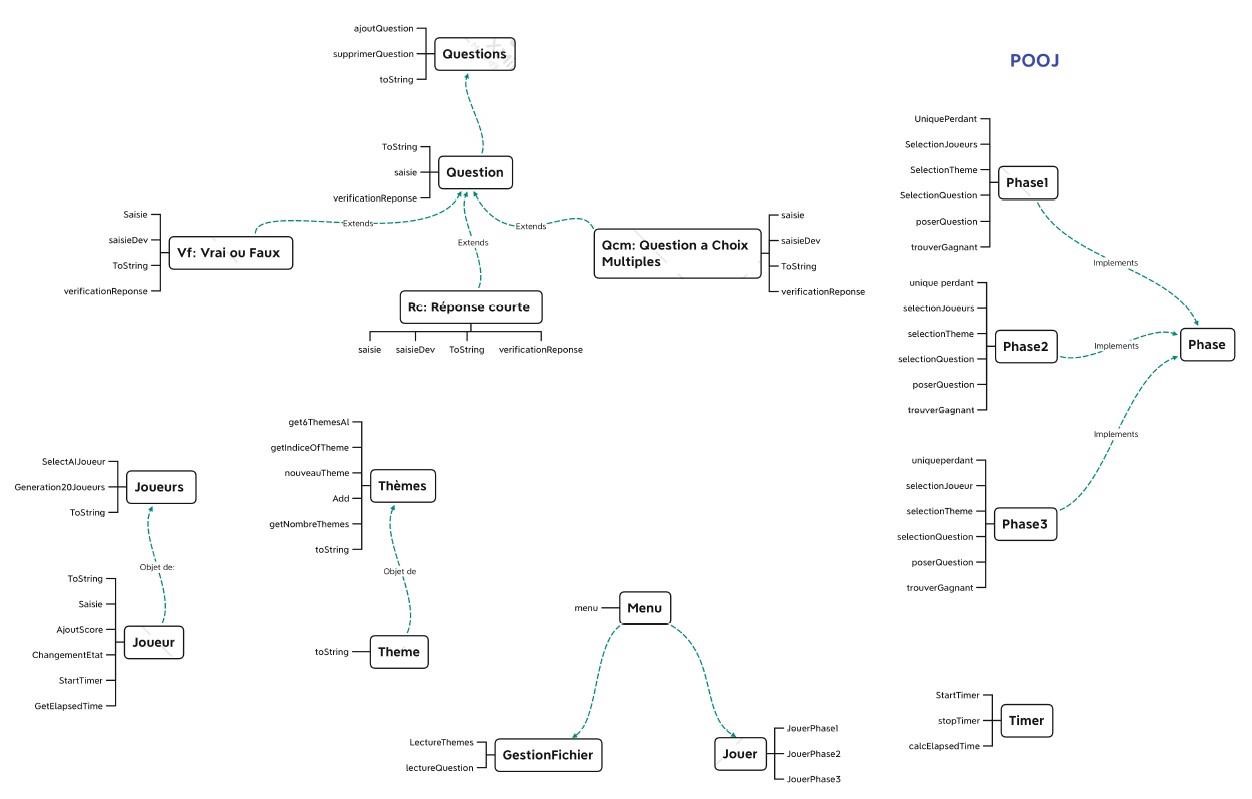
Ce projet de programmation orienté Java a pour but de créer un jeu dans la console de note IDE de question réponse. Il se base sur différents joueurs qui pourrons chacun leur tour au travers de la console répondre à des questions pour se sélectionner ou non pour les phases suivantes.

Présentation des fonctionnalités

Le projet présente ma majorité des fonctionnalités présentés et demandées dans le sujet. A savoir : l’affichage des thèmes, des joueurs et des questions, l’ajout et la suppression de question, l’ajout de Thèmes et de joueurs, lancer un jeu de trois phase impliquant 4 joueurs lancer un grand jeu de 3 fois 4 phase et d’un jeu entre les trois gagnants. Les trois bonus on aussi étaient réalisés, il y a une IA qui permet de donner des réponses aléatoires, un timer associé à chaque joueurs pour connaitre leur temps total de réponse et le grand jeu.

# I. Explications du programme

1. Organigramme et structure du programme



1. Détails des Classes et de l’Interface.

A) Classes utilisées.

GestionFichier : Cette Class permet de remplir le tableau de la Class « Questions » , en fonction des données du fichier « Donnees » il remplit également le fichier avec les nouvelles questions et idem avec les Thèmes.

Questions : Cette Class implémente dans un tableau toutes les valeurs pour les questions.

Question : Est une Class « abstract » définit ensuite la structure de chaque question en contenant les paramètres communs aux trois types de questions. (vf, qcm, rc).

QCM : Est une sous Class de Question, et permet de donner les paramètres des questions de types QCM.

Vf : Est une sous Class de Question, et permet de donner les paramètres des questions de types VF.

RC : Est une sous Class de Question, et permet de donner les paramètres des questions de types RC.

Jouer : Cette Class permet le bon déroulement des 3 phases lorsqu’elles sont appelées.

Joueur : Cette Class permet de définir l’entité « Joueur », en lui donnant son statut actuel ainsi que toutes ses propriétés.

Joueurs : Fais appel à la Class « joueur » elle contient la liste de tous les joueurs, et de toutes les fonctions d’affichages de cette sélection de joueur.

Phase 1 : Cette Class permet la mise en place de toute la structuration et le développement de la Phase 1.

Phase 2 : Cette Class permet la mise en place de toute la structuration et le développement de la Phase 3.

Phase 3 : Cette Class permet la mise en place de toute la structuration et le développement de la Phase 3.

Theme : Cette Class donne la structure de l’entité theme, et permet la création de celui-ci.

Themes : Permet de sélectionner un thème parmi la liste proposée et la mise en relation avec les phases.

B) Interfaces.

L’interfaces se nomme « Phase » il définit la signature de toutes les fonctions nécessaires au fonctionnement du jeu. Il contient plusieurs fonctions, qui appelle toutes les autres.

3. Détails des Fonctions utilisées.

Fonctions Importantes de GestionFichier :

lectureQuestion : Permet de lire le fichier « Donnees » et les répartis en fonction du type de questions.

Fonctions Importantes de Questions :

ajoutQuestion ajoute une question du Tableau. supprimeQuestion supprime une question des Question.

Fonctions Importantes de Question :

Question attribue à chaque question son theme et niveau, son nom,et savoir si elle a déjà été posée ou non.

Fonctions Importantes de Jouer :

jouerPhase 1 permet d’implémenter la première phase de jeu et de vérifier si les réponses sont juste et affiches également le score et sélectionne les vainqueurs.

jouerPhase 2 permet d’implémenter la phase 2 de jeu et de vérifier si les réponses sont juste et affiches également le score et sélectionne les vainqueurs.

jouerPhase 3 permet d’implémenter la dernière phase de jeu et de vérifier si les réponses sont juste et affiches également le score et sélectionne le vainqueurs.

Fonctions Phases : Permet de définir la signature des méthodes phase (1,2, ou 3).

Fonction importante de QCM : Ici, il permet la mise en place du qcm avec les différentes propositions de réponses, la saisie et l’affichage.

Même principe pour les fonctions de Vf et Rc .

II. Problèmes rencontrés.

1. Gestion du fichier CSV.

Lors de la première conception du projet, nous avions implementer toutes les valeurs concernant les questions QCM, VF, RC. Mais comme conseillé, nous avons décidé de stocker toutes ses valeurs qu’on peut assimiler à une base de données, dans un fichier de type CSV.

1. Interfaces pour implémenter les phases de jeu.

Au début, nous avions fait une Class qui implémentait l’interface pour chaque phase de jeu, mais nous avons ensuite décider de faire chaque phases dans une classe differente (et de simplifier l’interface en ne lui demandant d’appeler que les fonctions nécessaires au fur et à mesure).

1. Gestion des exceptions.

Les exceptions nous ont permis d’éviter de faire crasher le programme dès qu’une valeur fausses ou non attendu survenais lors des réponses. Elles nous ont permis de gérer également les problèmes entre la sélection des joueurs.

# III. BONUS

1.Implementation du Timer.

Nous avons décidé d’implémenter le timer pour les questions, lorsque la question est affiché à l’écran, le timer du joueur en question est lancé et s’arrête une fois une réponse fournie. Le timer de chaque joueur représente donc le temps de réponse total.

3. Implémentation de l’IA

Nous avons décidé d’implémenter une sorte d’IA que permet de faire répondre les joueur de manière autonome et ce peu importe le type de question. Lorsque qu’un joueur est créé, on demande à l’utilisateur de choisir si ce dernier est contrôlé par l’utilisateur ou par l’IA, le choix des thèmes lorsque cela est possible reste à la charge de l’utilisateur.

3. Implémentation du grand jeu

Nous avons décidé d’implémenter le grand jeu, dans le menu utilisateur, dans la section jeu, l’utilisateur choisi de soit lancer une jeu classique (Phase 1 , Phase 2, Phase 3 sur 4 joueurs) soit un grand jeu qui effectue 2 jeux classiques puis ensuite un jeu entre les trois gagnant pour désigner un grand gagnant.

4. Implémentation du menu

Nous avons implémenté le menu, il permet de guider le déroulement du jeu et permet à l’utilisateur de faire des choix. Il permet le lancement du jeu, l’affichage des questions / thèmes et permet aussi de saisir de nouvelles questions et de nouveaux thèmes.

# IV. Présentation des résultats

Les résultats sont que le jeu est totalement fonctionnel, lors de l’exécution de la fonction main du fichier testJeux.java du package testApplication, on à la génération automatique des questions et des thèmes des fichiers Donnees.txt et Themes.txt on peut ensuite naviguer dans les menu pour jouer (menu jeu) pour afficher les joueurs/themes/question dans le menu settings ou pour gérer les données dans le menu développer ou encore quitter l’exécution

Pour utiliser le programme il faut ouvrir le dossier PROJETS5POOJ dans un IDE (VisualStudioCode chez nous) et ouvrir le fichier TestJeux.java dans le dossier testApplication pour exécuter la fonction main de ce dernier

Pour éviter des problèmes de lecture de fichier à l’exécution du programme, les fichier Donnees.txt et Themes.txt qui contiennent ces informations ont étaient dupliqué dans les sous dossiers pour éviter les problèmes de lien.

Text

Description automatically generated

# V. Conclusion

1. Répartition des tâches.

Théo : Programmation de l’interface, des phases de jeu, des thèmes, de l’IA et du grand jeu.

Jules : Programmation des joueurs, du menu, du timer et du grand jeu.

Anthony : Programmation des types de questions et gestion du fichier CSV.

Toufik : Programmation des exceptions.

1. Perspectives d’améliorations.

Nous pouvons améliorer le programme en essayant de le rendre plus lisible et en essayant de l’optimiser.

Tous les bonus et fonctionnalité demandées dans le sujet ont étaient codées et fonctionnent,

Il reste encore certaines exceptions qui n’ont pas étaient géré, elles sont la partie d’administration.

Ce projet nous a beaucoup appris sur le plan technique avec la gestion de beaucoup de classe et d’interface mais aussi sur le plan humain du fait de certaines grosses difficulté auxquelles nous avons fait faces. Nous avons aussi appris à nos dépend que repousser les rendus et les réunions entre l’équipe nous mène à devoir travailler dur et longtemps sur le fil du rasoir pour suivre les délais.

Participations :

|  |  |
| --- | --- |
| Toufik PHILIPPE | Participation à 20% |
| Jules BURETTE | Participation à 30% |
| Anthony GUERIMAND | Participation à 20% |
| Théo CREPIN | Participation à 30% |

fin