

QUESTIONS POUR UN CHAMPION

RAPPORT

RIBEIRO AMANDINE
WILLAUME ARNAUD

LES REGLES DU JEU

1ere manche :

La 1^{ère} manche s'effectue en compagnie des 4 joueurs participants.

Le but de celle-ci est d'atteindre le seuil de 9 points afin d'accéder à la manche suivante.

Les questions sont diversifiées et couvrent plusieurs thèmes. Elles valent 1,2 ou 3 points en fonction du niveau de difficulté.

Un temps de 20 secondes est à votre disposition pour y répondre.

Au terme de celle-ci, les 3 premiers joueurs se retrouvent sélectionnés pour la phase suivante du jeu.

2^e manche :

Au cours cette manche, chaque participant aura pour mission de répondre correctement à 4 questions successives dans un délai de 90 secondes.

Un choix parmi 5 thèmes est offert aux joueurs parvenus au seuil des 9 points de la manche précédente.

Au fur et à mesure que les réponses sont correctes, la barre des scores augmente (1 puis 2 puis 3 etc.). Mais si une mauvaise réponse est ensuite donnée, la barre retombe à 0. Dès lors, il faut tout refaire : le plus haut score atteint reste enregistré.

Au terme de cette manche, les 2 meilleurs joueurs, c'est-à-dire ceux ayant répondu au plus de questions de manière successive sont qualifiés pour la phase finale.

3^e manche :

C'est le « face à face ». Le vainqueur de cette manche sera déclaré CHAMPION et remportera plein de dictionnaires qu'il pourra revendre sur ebay.

Vous disposez de 7 questions pour obtenir les 9 points nécessaires.

Avant chaque question, on propose alternativement à chaque joueur de « prendre » ou « laisser » la main.

Si vous prenez la main, vous pourrez marquer dans la zone des « 4 » ou des « 2 » points.

Si vous laissez la main, vous marquerez dans les zones valant « 3 » ou « 1 » points.

En cas de mauvaise réponse, la main passe à l'adversaire et il faut donc attendre qu'elle vous revienne pour pouvoir répondre.

DESCRIPTION DU PROJET

Dans un cadre universitaire de 3^e année de licence informatique, notre binôme a pour objectif de mettre à profit les notions apprises dans le cadre des cours de « Nouvelles technologies : Internet » dispensés par Monsieur Verlan.

Le but de ce projet est de développer, puis d'implémenter en langage Java, une version réseau du jeu « Questions Pour Un Champion » et de présenter le tout à l'aide d'un site Internet (créé en langage html). Vous pourrez trouver ce dernier à l'adresse : <http://questionspourunchampion.ifrance.com/>.

Le jeu se déroule en 3 manches et débute avec un nombre de participants égal à 4.

La première manche voit s'opposer 4 candidats. Une série de questions est posée, les joueurs doivent atteindre un palier fixé à 9 points afin d'obtenir leur billet pour la manche suivante. Seuls les 3 premiers candidats qualifiés pourront poursuivre le jeu. Un temps de 20 secondes s'écoule pour chaque question. Un barème est également fixé pour chacune d'entre elles, il oscille entre 1 et 3 points selon le niveau de difficulté.

La seconde manche se déroule avec 3 joueurs. Chacun leur tour, ils vont devoir répondre à une série de questions pendant 90 secondes. Le but de cette manche est de parvenir à répondre correctement à 4 questions de manière successive. En cas de mauvaise réponse, on garde le plus haut score atteint mais le joueur repart de 0. Les joueurs qualifiés sélectionnent chacun un thème de questions : la manche peut commencer. Au terme de cette manche du jeu, 2 candidats accèdent à la troisième et dernière phase du jeu.

Le « face-à-face » est la manche finale de la partie. Les deux derniers participants s'affrontent en vue du titre de champion. Le champion sera celui qui aura répondu à suffisamment de questions pour avoir les 9 points nécessaires. Cette manche a une particularité quant aux autres car elle nécessite de « prendre » ou « laisser » la main. En effet, les deux derniers candidats ne pourront pas répondre quand ils le souhaitent, un ordre est donné et le joueur devra attendre son tour. Le joueur qui prend la main pourra répondre dans les zones « 4 » et « 2 » points, celui qui la laisse aura les zones « 3 » et « 1 » points. En cas de mauvaise réponse de la part de l'un des candidats, la main passe à l'adversaire.

METHODE DE RESOLUTION

S'agissant ici d'un projet relativement complexe, le travail a dû être rigoureusement organisé afin d'être effectué dans les meilleures conditions. Pour la réalisation de celui-ci, nous avons opté pour un découpage en 4 parties.

Dans une première partie, nous nous sommes consacrés à l'interface graphique. Le jeu se déroulant en 3 manches, l'interface graphique n'est pas la même durant toute la partie.

Nous nous sommes ensuite penchés sur la réalisation des questions pour le jeu ainsi que les règles à proprement dites, le tout en langage Java.

Nous présentons de plus ce jeu dans une version réseau adaptée pour 4 joueurs : cela nécessite donc la réalisation d'un serveur acceptant des clients.

Enfin, la réalisation d'un site Internet présentant l'ensemble de ce projet, le rapport ainsi que le jeu a été effectué en langage HTML.

1. Questions Pour Un Champion - L'interface graphique

L'interface graphique a un rôle important dans un jeu. Nous avons donc fait pour le mieux afin de la rendre la plus agréable possible pour l'utilisateur. Nous avons opté pour une présentation simplifiée voire épurée mais ayant l'avantage d'être claire et non confuse ou trop chargée.

Notre programme comporte 5 classes à caractère graphique.

La première classe est la classe de « *Bienvenue* ». C'est la page d'accueil du jeu, et en l'occurrence la première fenêtre que le joueur voit en phase de jeu.

Le candidat est invité dans la partie, et peut entrer son nom. En attendant le début de la première manche et la connexion des autres joueurs, le joueur verra une barre de progression en bas de fenêtre. Il a également la possibilité d'entrer l'adresse IP du serveur en cliquant sur l'onglet « *Option* », crée grâce à un *JOptionPane*.

La classe « *ChoixCat* » intervient après la première manche, seuls les joueurs qualifiés pour la manche suivante la verront. Elle annonce le passage à la 2^{ème} manche et permet au joueur dont c'est le tour de jouer de choisir le thème pour les questions auxquelles il devra répondre par la suite. Les thèmes sont présentés grâce à l'intermédiaire de *JButtons*.

Les classes qui suivent sont les classes des différentes manches.

On retrouve la partie graphique de la 1^{ère} manche dans la classe « *GraphNiveau1* ». Elle consiste en une série de questions posées aux candidats. Nous avons opté pour la réalisation d'une *caseQuestion* à l'aide d'un *JTextArea* et d'une *caseReponse* avec un *JTextField*.

Par ailleurs, cette manche nécessite un comptage de points. Chaque joueur doit pouvoir connaître son score mais aussi celui des autres. Ainsi à gauche des cases de questions

et de réponses, le joueur voit son score. Le but de la manche étant d'atteindre un score de 9 neufs points, nous avons donc créé 1 *JPanel*, lui-même scindé en 9 *JPanels*. Au fur et à mesure des points acquis, ces différentes sections seront teintées en jaune. Du côté droit de la fenêtre on retrouve les scores de l'ensemble des joueurs, affichés grâce aux *JLabels* mis en place.

Chaque candidat a 20 secondes pour répondre à la question posée, nous avons donc instauré un chronomètre que l'on place sous le cadre question-réponse, il s'agit d'une étiquette *JLabel*.

Une autre étiquette a été également créée sous ce même cadre, elle indique au joueur si sa réponse est bonne ou non, et si le temps de réponse est dépassé.

La classe « *GraphNiveau2* » concerne la 2^{ème} manche du jeu. On y retrouve les *caseQuestion* et *caseReponse*. De même, l'étiquette du chronomètre (fixé ici à 90 secondes) ainsi que celle portant un commentaire quant à la fausseté de la réponse.

Le score cette fois-ci est comptabilisé dans un *JPanel* scindé en 5 autres *JPanel*, numérotés de 0 à 4. Ces divisions seront colorées au fur et à mesure des bonnes réponses des joueurs. Le but de cette manche est de parvenir à répondre à 4 questions successivement, ainsi si le joueur donne une mauvaise réponse il se verra repartir de 0. Cependant on coloriera le plus haut niveau atteint durant toute la partie.

La 3^{ème} manche est implémentée dans la classe « *GraphNiveau3* ». La *caseQuestion* est sous divisée en 6 parties représentées par des *JLabel*. En effet, dans cette manche les questions sont présentées sous la forme de définitions qu'il faut faire défiler progressivement, ainsi nous avons jugé utile la mise en place de 6 étiquettes pour effectuer cette tâche.

Comme précédemment, sont présents *caseReponse*, le chronomètre et l'étiquette de commentaire.

2. Questions Pour Un Champion - L'implémentation java du jeu

On retrouve l'ensemble des classes dans un projet intitulé GoingForGold (titre original du jeu). Le langage Java étant une programmation orientée objet, cela explique le fait du nombre important de classes. Au fur et à mesure de la conception du jeu, ce nombre n'a fait que s'incrémenter.

Le principe du jeu étant de poser des questions, nous avons d'abord procédé à la création d'une base de données de questions. Cette base de données est elle-même sous divisée en 3 parties correspondant à chaque manches du jeu.

Pour la première manche, nous avons attribué à chaque question une réponse ainsi qu'un nombre de points (évalués en fonction de la difficulté de la question).

Pour la seconde manche, les questions sont cette fois-ci regroupées par thèmes, par exemple « cinéma », « culture » ou encore « musique ». Chaque question rapporte le même nombre de points.

Pour la troisième manche, chaque question se présente sous la forme d'une définition.

Nous avons par la suite décidé de regrouper toutes ces questions dans la classe « *ListeQuestions* ». Nous avons créé des *ArrayList* qui vont toutes les contenir. En effet manipuler une *ArrayList* en Java comporte beaucoup d'avantages étant donné qu'au cours des différentes manches du jeu, nous serons amenés à les parcourir constamment.

Lorsqu'un utilisateur devra entrer une réponse, il n'aura pas à se soucier de l'entrer en caractères minuscules ou majuscules de par l'utilisation de la méthode *toUpperCase* qui convertit automatiquement les caractères.

Les classes « *QuestionsManche1* », « *QuestionsManche2* », et « *QuestionsManche3* » ont toutes le même principe. Elles définissent la structure des listes de questions correspondantes. Ainsi pour la première manche on définit un tableau de 2 cases : la première pour la question, la seconde pour la réponse. Pour la deuxième manche, on déclare cette fois un tableau de 3 cases : le thème, la question, la réponse. Et enfin pour la dernière manche, le tableau est plus important du fait du découpage de la question en plusieurs parties pour le défilement.

On y déclare de plus une variable *alreadyUsed* permettant de vérifier si la question a déjà été ou non utilisée et ce afin dans le but de ne pas réemployer plusieurs fois une même question au cours d'une même manche.

Il a fallu ensuite mettre en place un chronomètre, le temps imparti aux joueurs étant limité et différent pour chaque manche. On retrouve ce dernier dans les classes « *ChronometreManche1* », « *ChronometreManche2* » et « *ChronometreManche3* ». Nous avons dès lors dû aborder la notion de *Thread* nécessaire à l'implémentation de ces différentes classes. On déclare ainsi un thread que l'on va faire démarrer (et s'endormir) et qui va nous permettre d'afficher le chronomètre et surtout de le faire défiler.

La classe « *GraphNiveau1* » s'intéresse à la première manche du jeu, on a décrit son rôle d'ordre graphique précédemment, mais c'est également elle qui comporte grossièrement les règles de la manche. A l'aide d'une variable *hasard*, on parcourt la liste de questions disponible et on pose les questions aux joueurs. En cas de bonne réponse, le point est comptabilisé, sinon un message s'affiche pour indiquer au joueur qu'il a commis une erreur. Une question utilisée au cours d'un jeu ne sera pas réutilisée au cours du même jeu.

Cette classe implémente l'interface *KeyListener*. En effet, dans un souci de facilité d'utilisation, lorsque le joueur appuie sur la touche ENTREE de son clavier, sa réponse est validée et donc testée.

La classe « *GraphNiveau2* » reprend le même principe que la classe « *GraphNiveau1* » à quelques différences près.

Les questions posées vont dépendre du choix de thème effectué auparavant par le candidat. Une fois ce choix établi, une variable *hasard* parcourt les questions de la liste concernée.

Contrairement à la manche précédente, le joueur a le droit de changer de question pour ne pas perdre de temps (le temps étant global pour la manche), il a alors la possibilité d'appuyer sur la touche ECHAP de son clavier pour passer à la question suivante (changer de questions équivaut au niveau du comptage des points à une mauvaise réponse de sa part). On utilise donc de même un appel à l'interface *KeyListener*.

La classe « *Jeu* » est la classe exécutée par les clients pour le jeu en lui-même. A chaque lancement, un client est créé. Dès lors la fenêtre de « bienvenue » s'affiche : elle permet au joueur de s'identifier et de se déclarer prêt au commencement du jeu. Lorsque le client entre son nom et clique sur « commencer », celui-ci se retrouve connecté au serveur. Quand les 4 clients ont effectué ceci, cela a pour conséquence de faire démarrer le jeu.

La classe « *Joueur* » gère les informations importantes relatives au joueur. Celles-ci nous serviront au fur et à mesure du jeu, comme notamment son nom ou ses points gagnés au cours des différentes manches du jeu.

3. Questions Pour Un Champion - Programmation réseau

Nous proposons ici une version réseau du jeu Questions Pour Un Champion. Ainsi 4 joueurs vont pouvoir s'affronter au cours d'une partie.

Le but d'un réseau est de permettre à plusieurs utilisateurs de communiquer entre eux et donc de jouer ensemble. Pour cela, nous mettons en place un serveur prêt à accepter et gérer des clients, en l'occurrence 4 par partie.

La partie Serveur sera regroupée dans les classes « *MainServer* » « *QPUCServer* » et « *QPUCServerThread* ». La partie Client sera quant à elle regroupée dans les classes « *QPUCClient* » et « *QPUCClientThread* ».

Le serveur gère les connexions Client. C'est lui qui détermine si une partie peut commencer ou non. Il attend les clients, les comptabilise et leurs attribue un numéro de joueur. Une fois le nombre de 4 clients atteint, la partie peut commencer.

La classe « *MainServer* » est l'exécutable de la partie serveur.

La classe « *QPUCClient* » permet d'enregistrer l'adresse IP à laquelle le client va se connecter, de même que le port.

Ces informations sont ensuite transmises à la classe « *QPUCClientThread* » dont la méthode *run* gère en quelque sorte le jeu.

La classe « *QPUCServer* » concerne le graphique de la fenêtre que le serveur affiche. On y voit les clients qui se connectent et une fois le nombre de clients suffisant elle donne le top départ du jeu.

Cette classe fixe également le port utilisé par les clients et attend la connexion de ces derniers. Dès lors un *socket* est créé puis transmis grâce à *QPUCServerThread* qui gèrera les communications.

La classe « *QPUCClientThread* » gère la communication entre le serveur et le client. Elle contrôle les flux d'entrées et sorties du client.

Dans cette classe, les clients transmettent leur score obtenu au cours de la manche 1 au serveur, celui-ci les redistribue à chacun afin d'afficher dans leurs fenêtres respectives les scores obtenus par tous.

La classe « *QPUCServerThread* » gère également la communication entre le serveur et le client. Elle contrôle les flux d'entrées et sorties du serveur.

C'est la partie communicante avec la classe « *QPUCClientThread* ». En fait ce sont leur méthode « *run* » qui dialoguent.

4. Questions Pour Un Champion - Le site

La présentation de ce projet se fait sous la forme d'un site Internet, créée au format HTML.

Pour sa réalisation, il a fallu établir un plan de site.

Nous avons opté pour une page d'accueil avec un logo du site, et un lien direct vers le sommaire, sommaire qui permet à l'utilisateur de naviguer à travers tout le site.

Chaque page du site présente un lien vers celui-ci, de même un lien vers nos mails respectifs.

Le site est donc découpé en plusieurs parties :

- la page d'accueil
- le sommaire
- les règles du jeu
- la description du projet
- la méthode de résolution
- conclusion
- les sources java
- les captures d'écran
- le jeu (les fichiers .rar)

Le site a été conçu dans un effort d'esthétique et de facilité de navigation.

Afin de profiter pleinement du site, l'utilisateur devra installer dans son système informatique les polices utilisées pour la création des pages web : Andy et Matisse (fichiers à installer dans votre dossier C:\WINDOWS\Fonts).

La page d'accueil est constituée d'un logo du jeu, remanié à l'aide du logiciel Photoshop, elle permet un accès direct au sommaire du jeu.

La page de sommaire recoupe comme son nom l'indique un plan du site avec un lien vers les différentes pages nommées.

Les règles du jeu sont une explication du déroulement du jeu manche par manche.

Le projet est ensuite décrit et la méthode de résolution développée, suivie d'une conclusion.

On trouve également une page donnant accès à toutes les sources java constituant le programme.

La page de captures d'écran donne un aperçu du jeu.

Arrive ensuite la page permettant au visiteur d'obtenir ce fameux jeu, il y trouvera le fichier .jar lui permettant de jouer.

CONCLUSION

Réaliser ce projet fut une expérience enrichissante et un véritable challenge pour nous, et ce, sur plusieurs points. La création d'un jeu a toujours des côtés ludiques mais permet de mettre concrètement en œuvre les notions apprises au cours de notre cursus universitaire. Il en est de même pour la création du site Web.

Evidemment, nous avons été confrontés à de très nombreuses difficultés. La gestion du temps et l'organisation du travail n'a certainement pas été optimisée comme elle l'aurait pu et dû. Notre maîtrise concernant les threads ou bien le réseau, avec les sockets, les communications inter-clients, client-serveur ou serveur-client étant loin d'être parfaites, nous sommes penchés longuement sur le sujet.

De très nombreux tests ont été effectués au cours de cette réalisation.

Notre programme est constitué de deux exécutable : un .jar pour le serveur et un .jar pour la partie client (qui sera à exécuter 4 fois).

On part du principe que le serveur est lancé une fois sur une machine, et que l'adresse IP de cette machine est connue par les 4 clients lançant le jeu sur leurs machines respectives.