

420-KBE, application Trivial Pursuit

Objectifs :

1. Réaliser un diagramme de cas d'utilisation
2. Décrire les cas d'utilisation
3. Écrire les user-stories
4. Estimer et prioriser les user-stories
5. Produire un backlog de produit.
6. Produire un plan de Release.
7. Travailler en équipe.

Nous voulons implémenter une petite application C# ADO.NET pour le jeu Trivial Pursuit. Trivial Pursuit est un jeu de questionnaire qui se joue de 2 à 6 joueurs.

Description du jeu :

- Le jeu est un jeu de questionnaires.
- Le jeu se joue de 2 à 6 joueurs. C'est un jeu tour à tour. **Pour l'application on prend uniquement deux joueurs.**
- Nous avons **4** catégories : sport, histoire, géographie, art – culture. Vous pouvez choisir d'autres catégories.
- Chaque catégorie a une couleur.
- La couleur blanche ne correspond à aucune catégorie.
- Chaque catégorie a un ensemble de questions environ une **dizaine** .
- Chaque question a 4 choix de réponses, dont une bonne réponse. Et une seule bonne réponse.

Déroulement du jeu :

On tourne une roulette, lorsque la roulette tombe sur la couleur correspondant à la catégorie, une question est choisie au hasard dans cette catégorie. Le joueur répond à la question. Si sa réponse est bonne il continue. Il tourne la roulette à nouveau, sinon il passe son tour à un autre joueur.

Lorsque la roulette tombe sur la couleur blanche, alors le joueur peut choisir sa catégorie.

Un joueur est déclaré gagnant lorsqu'il a répondu (a donné la bonne réponse) à 3 questions de chaque catégorie. Ce chiffre peut être paramétrable. (2 à N).

Indication concernant la base de données :

La base de données doit contenir entre autres les différentes catégories, les questions, les réponses, les joueurs, le pointage des joueurs et les statistiques de chacun des joueurs.

- Les joueurs sont identifiés par un numéro unique, un alias unique, un nom et un prénom.
- Les catégories ont un numéro ou un code catégorie unique, une couleur et un nom.
- Les questions ont un numéro unique, un énoncé, un Flag indiquant si la question est déjà posée. Le flag est initialement à zéro. Il prend la valeur 1 lorsque la question est posée.
- Les réponses ont un numéro unique. Un énoncé et un attribut indiquant si c'est la bonne réponse.
- Chaque question appartient à une seule catégorie. Une catégorie à 1 à plusieurs questions.
- Chaque question à 4 réponses. Une réponse concerne une seule question.
- Un joueur peut gagner plusieurs catégories. Et une catégorie peut être gagnée par plusieurs joueurs.

Indication concernant les requêtes à la base de données :

La base de données doit répondre au moins à ces questions :

- Chercher une question en fonction de la catégorie;
- Chercher les réponses d'une question;
- Valider la bonne réponse;
- Mettre à jour le pointage d'un joueur;
- Afficher pour chaque joueur (Alias) la liste des catégories (nomCatégorie) qu'il a gagnées;
- Pour un joueur, afficher la catégorie la plus faible;
- On doit pouvoir ajouter des questions et leurs réponses;

Indication concernant l'interface.

Vous ne devez en aucun cas implémenter une roue. Ce n'est pas le but

Conseil : utiliser un nombre aléatoire entre 1 et 5 pour déterminer la catégorie. Lorsque 1 est tiré, cela veut dire qu'il s'agit de la catégorie Art, lorsque 2 est tiré cela veut dire qu'il s'agit de la catégorie Sport, ainsi de suite

Nombre aléatoire	Catégorie	Couleurs
1	Art	
2	Sport	
3	Culture générale	
4	Histoire	
5	Choix de la catégorie	Blanc

Contraintes de conception

- La Base de données est une base MS SQL Server;
- Toutes les requêtes doivent être écrites **en procédures stockées**;
- L'application doit être implémentée en C# ADO.NET;