

Projet de synthèse

Ce projet doit vous permettre de vivre une mise en situation professionnelle.

Vous travaillerez en équipe avec une grande autonomie.

Vous devrez mettre en œuvre les bases de la programmation avec pour objectif de produire dans le respect de l'état de l'art un composant logiciel conforme aux spécifications.

Vous découvrirez tout au long de cette semaine de nouveaux outils et méthodes accompagnant le cycle de vie d'un projet et visant à professionnaliser votre démarche :

- GIT : Outil collaboratif de gestion de versions
- Externalisation des ressources (en vue éventuellement de l'internationalisation du logiciel)
- Tests unitaires
- Outil de vérification de code (revue automatique)

Les spécifications fonctionnelles

Vous devez développer un jeu qui permet de faire découvrir un ou plusieurs mots.

Ces mots sont proposés aléatoirement par le programme.

Le joueur propose à chaque fois une lettre. Si la lettre est présente, elle est découverte. Si la lettre est absente, une erreur est comptabilisée. Le but du jeu est de découvrir ces mots en un minimum de temps et d'essais erronés.

Ce jeu se joue en 2 à 5 manches. Un mot par manche.

Pour chacune des manches du jeu, les points sont comptabilisés ainsi :

- Le nombre de secondes dont le joueur a eu besoin pour trouver le mot. 1 point par seconde
- Le nombre d'essais incorrects : 7 points par lettre absente.

Ainsi, si vous avez mis 160 secondes pour trouver le mot ALERTE et commis 5 erreurs, vous comptabilisez $(1 \times 160) + (7 \times 5)$, soit 195 points. L'objectif est d'obtenir le nombre de points le plus faible. Le classement s scores obtenus est faible as, meilleur est le résultat.

Une manche peut être déclarée perdue si le nombre d'essais en erreur est égal à 9. Une manche perdue implique la perte de la partie.

Le score d'une partie est la moyenne des points obtenus lors des différentes manches gagnées.

Les 10 meilleurs scores sont sauvegardés avec le score obtenu et le pseudo du joueur.

Règles relatives aux mots

Un mot ne doit pas comporter moins de 5 ou plus de 25 lettres.

Les caractères accentués et la différence de casse ne sont pas pris en compte.

Le jeu de caractères utilisé ne comportera donc que des majuscules non accentuées.

Les mots sont extraits d'un fichier alimenté à partir de divers fichiers texte.

L'application proposera un composant permettant d'alimenter ce fichier. Au-delà des règles mentionnées ci-dessus, il sera nécessaire de réaliser un contrôle visuel des mots devant être insérés dans le fichier afin d'en exclure les termes impropres.

Normes et standards

Vous devez coder vos composants en respectant l'état de l'art du développement objet et les conventions d'écriture définies pour le langage C#.

Les composants développés devront respecter les règles de séparation de la couche métier de celle relative à l'interface utilisateur.

Ainsi, vos objets métier, qui régissent les propriétés et les méthodes propres au jeu, pourront faire l'objet d'implémentation diverses :

- Interface Windows Forms
- Interface Windows Presentation Foundation
- Interface Web de type WebForms ou Asp.mvc
-

Vous devez ici produire des composants d'interface de type Windows Forms.

L'utilisateur souhaite retrouver des comportements similaires à ceux des outils bureautiques utilisés au quotidien. Vous devrez donc respecter les standards des interfaces graphiques sous Windows.

Vos composants métier doivent pouvoir être testés unitairement au sein d'un projet de tests.

Vos composants doivent être documentés en recourant aux commentaires XML qui seront ensuite traités pour réaliser une documentation technique. Voir la documentation proposée sur le site.

Vos composants doivent pouvoir être facilement internationalisés. Les littéraux de type chaîne devront donc être définis dans un fichier de ressources en fonction de la culture définie. Aujourd'hui, un développeur se doit de penser la conception de son application dans un contexte international et multiculturel. Prévoyez dès le début d'externaliser vos chaînes et images dans des fichiers de ressources.

Vérifiez la qualité de votre code en utilisant l'outil d'analyse de code intégré à l'environnement de développement. Il permet de mettre en évidence les violations des règles de nommages ou les opérations illicites.

Les paramètres de l'application sont externalisés afin de pouvoir être modifiés sans nécessiter la modification du programme.

Les paramètres de portée utilisateur :

- Nombre de manches

Les paramètres de portée application :

- Nombre de points par seconde
- Nombre de points par essai infructueux
- Nombre d'essais max
- Règles de validité d'un mot : nombres min et max de caractères

Les composants à développer

Les composants métier

- Classe JeuQuinto
- Classe Score
- Classe Pioche

Les composants d'interface

- Fenêtre MDI avec 3 options
 - Nouvelle Partie
 - Alimenter Pioche
 - Options
 - Scores

- Fenêtre Partie

La fenêtre qui permet de jouer.

Le joueur peut visualiser le nombre total de manches et la position de la manche courante

Le joueur a connaissance du nombre d'essais erronés sur le nombre d'essais total, du temps passé, du score...

Le joueur visualise les caractères qui figurent dans le mot à découvrir et ceux d'ores et déjà découverts.

L'interface permet au joueur de ne pas proposer plusieurs fois la même lettre.

Le joueur doit pouvoir souffler entre deux manches... La gestion du chronomètre doit tenir compte de cette règle.

- Fenêtre Alimenter Pioche

Elle permet de choisir un fichier texte à lire en vue d'alimenter la pioche.

Elle affiche la liste des mots extraits du fichier conformes aux paramètres de l'application (longueur min et longueur max)

Elle permet, par opération manuelle, d'éliminer les mots qui ne devront pas être introduits dans la pioche.

- Fenêtre Options

Dialogue modal.

Permet de consulter les paramètres de l'application de portée Application

Permet de modifier ceux de portée utilisateur

- Scores

Affiche la liste des 10 meilleurs scores (max)