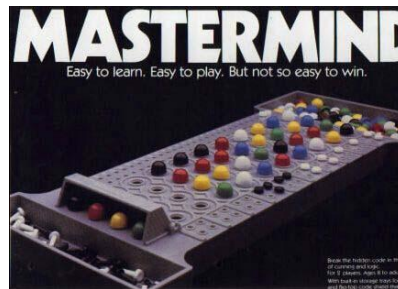


LE PROJET

Création d'un jeu MasterMind

Module de référence :
Apprendre à programmer en langage objet (Java)

Durée estimée de la mission : 18 heures



1. Mission

Avant de le proposer sur smartphones (un graphiste est en train de réaliser la maquette de l'UX), HASBRO vous demande de réaliser en Java son jeu de MasterMind sans UI graphique.

2. Enjeux Objectifs

- Appliquer les règles d'algorithmie en Java
- Penser et concevoir « objet »
- Créer les entités nécessaires (classes, constructeurs, méthodes...)
- Manipuler les tableaux
- Apprendre à déboguer efficacement
- Conceptualiser et manipuler une base de données

3. Règles du jeu

Etape 1 : l'application génère, le joueur cherche.

Votre application génère un nombre aléatoire à 5 chiffres différents (pas de doublons) ne comportant pas de 0. Elle demande ensuite au joueur une combinaison de 5 chiffres. Il indique alors le nombre de chiffres bien placés (BP) et mal placés (MP). La somme des BP et MP ne peut dépasser 5.

Si le joueur trouve la bonne combinaison, l'application affiche "Bravo".

Si après 10 coups, le joueur n'a pas trouvé la solution, l'application affiche "Perdu" et affiche la solution.

Les erreurs de saisie (doublons, pas assez de chiffres...) sont signalées.

Exemple d'une partie (en gras, les saisies) :

A vous de jouer !

1 : **12345** > 3BP/1MP

2 : **12364** > 2BP/1MP

3 : **12547** > 1BP/2MP

4 : **15348** > 2BP/2MP

5 : **18395** > 4BP/0MP

6 : **18325** > 5BP/0MP

BRAVO !

Etape 2 : le joueur génère, l'application cherche.

Améliorez l'application précédente. Maintenant, le joueur note sur papier une combinaison puis laissez l'application proposer des combinaisons. A chaque tentative, le joueur lui donne son score.

Les erreurs de saisie (format BP/MP incorrect) sont signalées.

Si l'application ne trouve pas en 10 coups, elle demande la solution puis recherche d'éventuelles erreurs d'évaluation BP/MP dans les réponses et les signale le cas échéant (honte au joueur !). Elle conserve donc les réponses (10 maxi).

Ne cherchez pas un algorithme "intelligent", la force brute est ici très efficace : écrivez un algorithme qui génère des combinaisons progressives (à partir de 12345 par exemple) puis, pour chacune d'elles, teste si c'est une solution possible en se basant sur les réponses précédentes. Si c'est effectivement une solution possible, l'application la propose sinon elle génère la combinaison suivante et ainsi de suite. De cette façon, elle trouvera facilement avant 10 coups (bien plus efficacement qu'un humain).

Vous ajouterez également un menu d'accueil.

Exemple de partie (en gras, les saisies) :

Les propositions de l'application sont factices, faites mieux !

MENU

- 1 : Je choisi la combinaison, vous jouez.
- 2 : Vous choisissez la combinaison, je joue.
- 3 : Quitter.

Votre choix ? 2

Prêt ? A moi de jouer !

- 1 : 12364 > **2BP/1MP**
- 2 : 12547 > **1BP/2MP**
- 3 : 15348 > **2BP/2MP**
- 4 : 18395 > **4BP/0MP**
- 5 : 18325 > **5BP/0MP**

Je suis heureux d'avoir trouvé !

MENU

- 1 : Je choisi la combinaison, vous jouez.
 - 2 : Vous choisissez la combinaison, je joue.
 - 3 : Quitter.
- Votre choix ?

Etape 3 : avec des doublons.

Corsez le jeu en permettant les doublons de chiffres dans les combinaisons (ex : 32325). Attention, la somme des BP et MP ne doit toujours pas dépasser 5 donc réfléchissez bien à votre logique de traitement.

Complétez le menu d'accueil pour que l'on puisse choisir entre les versions avec ou sans doublon.

Etape 4 : identification/authentification, profil et statistiques de jeu.

Vous devez mettre un système d'identification/authentification pour accéder au jeu. L'accès au jeu ne peut se faire que si le joueur est inscrit. Le menu du jeu ne s'affichera que si le joueur s'est identifié/authentifié correctement en saisissant un login et un mot de passe valides. Ainsi, au lancement du jeu, l'application affichera un menu proposant à l'utilisateur soit de s'inscrire, soit de s'identifier avant de pouvoir accéder au menu du jeu.

Dans le cas d'une inscription, le joueur devra indiquer son nom, son prénom, son login, son mot de passe, sa date de naissance ainsi qu'une adresse mail. Le login doit être unique et l'application doit demander une nouvelle saisie de login, tant que celui-ci existe déjà.

MENU

- 1 : Je m'inscris
- 2 : Je m'authentifie

INSCRIPTION

Veuillez saisir votre nom ...

Le jeu devra également proposer la modification de ces informations par l'utilisateur, l'affichage de ses informations de profil et l'affichage des statistiques de jeu.

BIENVENUE JOHN

- 1 : Je choisis la combinaison, vous jouez.
- 2 : Vous choisissez la combinaison, je joue.
- 3 : Afficher votre profil
- 4 : Modifier vos informations
- 5 : Afficher les statistiques
- 6 : Quitter

Les statistiques permettent de fournir les informations suivantes : nombre de parties jouées, nombre de parties gagnées/perdus, résultats pour la meilleure partie avec la date et le classement général du joueur. Vous êtes libre de rajouter d'autres informations.

A ce stade de votre apprentissage, vous développerez ce projet en groupe de 4. Avant d'être exigeant au sein d'un groupe, devenez d'abord exigeant avec vous-même.

4. Livrables

Vous livrerez un *zip* du dossier racine de la source de votre application, un exécutable ainsi qu'un rapport. La source devra comprendre une classe Main (programme principal) et un fichier séparé pour chacune des classes utilisées. Vous pouvez répartir vos classes en différents packages si vous le souhaitez. Le rapport devra contenir une introduction, une présentation des acteurs du projet, un cahier des charges, les spécifications de l'application, modélisation de la base de données, diagramme de classes métier, diagramme de cas d'utilisation, diagramme de séquence, algorithmes mis en places (pseudo-code et explications de ces derniers), règles du jeu, mode d'emploi et de déploiement du jeu, patterns de conception mis en œuvres, planification, annexes ainsi qu'un bilan final du projet.

Votre code sera 100% personnel et annoté afin de générer une Javadoc. Vous ne ferez aucun usage d'API externes. Une fois dézippée, l'application devra être immédiatement compilable.

5. Évaluation

Structure et forme du document :		
Logique et efficacité de l'algorithme « l'application génère, le joueur cherche »		/2
Logique et efficacité de l'algorithme « le joueur génère, l'application cherche »		/2
Cohérence et efficacité du modèle objet		/4

Gestion des doublons	/1
Identification/authentification et profil	/2
Statistiques	/2
Astuce, compacité, propreté du code	/1
Qualité et contenu du rapport	/6
NOTE	/20

6. Ressources en lignes pour aller plus loin

Habituez-vous aux docs officielles en anglais. Tout commence ici :

[//docs.oracle.com/javase/8/docs](https://docs.oracle.com/javase/8/docs)

La référence absolue :

[//docs.oracle.com/javase/8/docs/api/](https://docs.oracle.com/javase/8/docs/api/)

7. Ouverture d'esprit

Challenge personnel

Revoyez votre code. Traquez les incohérences, les variables inutiles, les redondances... En un mot, optimisez !

Trucs et astuces :

Préférez la doc officielle de l'API de Java à toutes les autres sources.

Utilisez un IDE efficace pour accélérer votre développement.

Pensez « objet », passez des objets entre méthodes.