

Mettez votre logique à l'épreuve

MISSION COURS ÉVALUATION

🕒 140 heures

Mis à jour le jeudi 12 septembre 2019

Contexte

GAMEPLAY STUDIO, l'entreprise d'édition de logiciels, où vous venez d'être recruté, a signé un nouveau contrat.



GAMEPLAY STUDIO - entreprise d'édition de logiciels

Il s'agit d'une commande pour la réalisation d'un jeu numérique basé sur le concept des Escape Games.

Un escape game se joue en une heure dans une pièce fermée. Le principe est de résoudre, en équipe, une série d'énigmes scénarisées pour pouvoir sortir de la pièce dans le temps imparti.

Vous êtes affecté à l'équipe de développeurs Java qui va travailler sur ce projet. En tant que développeur Java junior, vous êtes sous la direction de Sophie, team lead, et de Steven, l'architecte logiciel.

Vous recevez ce mail de la part de Sophie :

Objet : Projet "Escape Game ONLINE"

Hello,

Comme nous en avons discuté à la réunion de lancement de projet, avec Steven, nous souhaitons que tu travailles sur la brique "mécanisme de recherche d'une combinaison à X chiffres".

Comme c'est ton premier projet, on voudrait t'intégrer en douceur dans nos process de développements. Nous te proposons donc de réaliser cette brique de

façon indépendante au reste du projet.

Je récapitule les spécifications à suivre :

1. L'intelligence artificielle de l'ordinateur joue le rôle de défenseur. Elle définit une combinaison de X chiffres aléatoirement.
2. Le joueur a le rôle d'attaquant et doit faire une proposition d'une combinaison de X chiffres.
3. L'intelligence artificielle de l'ordinateur indique pour chaque chiffre de la combinaison proposée si le chiffre de sa combinaison est plus grand (+), plus petit (-) ou si c'est le bon (=).
4. Il y a un nombre limité d'essais.

Exemple :

```
1 (Combinaison secrète : 1234)
2 Proposition : 4278 -> Réponse : ---
3 Proposition : 2214 -> Réponse : -+=
4 ...
```

text

Cette façon de faire est le mode **“challenger”**.

Voici les spécifications pour l'autre mode **“défenseur”** :

1. Le joueur (cette fois dans le rôle de défenseur) définit une combinaison de X chiffres aléatoirement.
2. L'intelligence artificielle de l'ordinateur doit faire une proposition d'une combinaison de X chiffres (c'est le rôle attaquant).
3. Le joueur indique pour chaque chiffre de la combinaison proposée si le chiffre de sa combinaison est plus grand (+), plus petit (-) ou si c'est le bon (=).
4. L'intelligence artificielle fait une autre proposition en se basant sur la réponse fournit par le joueur.
5. Il y a un nombre limité d'essais.

Enfin, il y a le mode **“duel”** ! Voici les spécifications pour ce mode :

- Le joueur et l'intelligence artificielle de l'ordinateur jouent tour à tour. Le premier à trouver la combinaison secrète de l'autre a gagné ! (Perso, je pense que ce mode sera le plus sympa à jouer 😊).

Pour tester ton développement, tu mettras en place une interface “console” qui permettra les actions suivantes :

- Au démarrage, l'utilisateur choisit un mode (challenger, défenseur, duel). L'application lance le mode sélectionné.
- L'utilisateur joue. S'il perd, l'application affiche la solution.
- À la fin de la partie, l'utilisateur peut choisir :
 - de rejouer au même mode ;
 - de lancer un autre mode (retour à l'écran de choix du début) ;
 - de quitter l'application.

Et par défaut, le nombre de chiffres de la combinaison est 4.

Si tu as des questions, n'hésite pas à me demander.

Sophie

Steven vous a également envoyé un mail afin de vous préciser les contraintes techniques à respecter :

Objet : Escape Game ONLINE - quelques recommandations

Salut,

Sophie t'a normalement déjà fourni les spécifications fonctionnelles. En tant qu'architecte, je souhaite que tu tiennes compte des points suivants dans ton développement :

1. Utilise la modularité de la programmation objet pour que ton code soit ensuite réutilisable dans le projet principal Escape game (donc pas de programmation procédurale !). Il faut qu'on puisse extraire tes packages/classes à terme.
2. Un fichier de configuration doit contenir les paramètres de l'application (nombre de chiffres dans la combinaison, nombre d'essais, mode "développeur" activé ou non).
3. Un mode "développeur" doit pouvoir être activé via un paramètre. Ce mode affiche la solution dès le début du jeu.
4. Un fichier de configuration (`log4j.xml`) permettra de paramétrer les logs de l'application. La gestion des logs se fera avec Apache Log4j.
5. Comme tu es nouveau je préfère le préciser, tu n'oublieras pas de mettre ton code sur un repository github, d'indenter, commenter et formater le nom des variables. 😊

Je reste à ta dispo !

Steven

Livrables attendus

Vous livrerez, sur [GitHub](#) ou [GitLab](#) (dans un seul dépôt Git) :

- le code source de l'application
- le fichier de configuration
- une documentation succincte (un fichier `README.md` suffit) expliquant comment compiler et lancer l'application

Le projet est validé tant sur le fond (application fonctionnelle) que sur la forme (organisation technique, présentation du code...).

Vous donnerez un accès en lecture à votre dépôt Git à votre mentor et au mentor qui vous fera passer la soutenance.

Soutenance

Avant la soutenance

Quelques jours avant la date de la soutenance (en général 3-4 jours avant), vous transmettez tous vos livrables à l'évaluateur qui vous fera passer celle-ci, afin qu'il puisse en prendre connaissance en amont et éventuellement tester votre application.

Déroulement de la soutenance

La soutenance, d'une durée de 20-25 minutes, se déroulera en deux parties :

- Partie 1 – 15 minutes : **Présentation de l'application**
 - [~ 5 minutes] Vous ferez une démonstration rapide de votre application.
 - [~ 10 minutes] Vous présenterez votre solution technique et les points particuliers de votre développement (à vous d'être pertinent sur les points à aborder).
- Partie 2 – 5-10 minutes : **Retour sur la soutenance**
 - L'évaluateur pourra vous demander d'approfondir certains aspects ou vous questionner sur vos livrables.
 - L'évaluateur vous fera un debrief sur votre prestation en soutenance.

Compétences évaluées



Mettre en oeuvre les concepts fondamentaux de la programmation en Java

OpenClassrooms

L'entreprise

Alternance

Forum

Blog

Nous rejoindre

Entreprises

Business

En plus

Devenez mentor

Aide et FAQ

Conditions Générales d'Utilisation

Politique de Protection des Données Personnelles

Nous contacter



Français



Télécharger dans
l'App Store