STRATEGIE DE TESTS

Tournament Builder

Stratégie de test

Nom / Code projet	Tournament Builder

Référence	TB-Stratégie 2024

[A quelle stratégie, objectif stratégique, portfolio ou programme, le projet est-il affilié ?]

Service/Organisation	Développement progiciels

Historique

Version	Auteurs	Description	Date
001	Oussama DAHMANI, Maxim Costa	Initialisation document	27/02/2024
002	Oussama DAHMANI, Maxim Costa	Finalisation du document	10/03/2024

Table des matières

Objectifs et Contexte

Pourquoi ce document ?

Dans le cadre de notre stratégie, nous visons à atteindre une fiabilité sans faille, assurant que l'application fonctionne de manière fluide et sans erreurs, pour que rien ne vienne perturber l'expérience utilisateur. Enfin, notre objectif est de garantir une performance exceptionnelle, permettant à l'application de gérer un grand nombre d'utilisateurs et de transactions sans ralentissement, évitant ainsi de se transformer en un dispositif lent et frustrant.

Généralités

Contexte général et déroulement

En développant des cas de test détaillés axés sur les fonctionnalités de l'application, nous allons mettre l'accent sur les scénarios d'utilisation réelle ainsi que sur les parcours moins fréquentés. De plus, l'intégration du Behavior-Driven Development (BDD) et du Test-Driven Development (TDD) dans notre processus de conception jouera un rôle clé. Cette approche nous permettra de mieux comprendre les exigences à travers le prisme des comportements utilisateurs et de garantir que chaque aspect de l'application est rigoureusement testé dès les premières étapes du développement, assurant ainsi une qualité et une fiabilité accrues.

-

Informations détaillées

Définition des test effectués et leur but

Test générateur d'équipe:

- **Test de la méthode generateTeams() :**

But : Ce test vise à s'assurer que la méthode generateTeams() de la classe TeamGenerator génère bien une équipe. Il vérifie que le nombre d'équipes créées est égal à 2 et que chaque équipe contient un nombre de joueurs égal à la propriété playersPerTeam de l'instance de TeamGenerator.

- **Test de la méthode getTeams après la génération des équipes :**

But : Ce test s'assure que la méthode getTeams retourne un tableau (array) d'équipes après que les équipes aient été générées par generateTeams(). Il confirme également que le tableau n'est pas vide, indiquant que la génération d'équipes a été un succès.

- **Test d'ajout d'un joueur à une équipe :**

But : Ce test vérifie la fonctionnalité d'ajout d'un joueur à une équipe existante. Après la génération initiale des équipes, le test ajoute un nouveau joueur à une équipe spécifiée par teamIndex et s'assure que la longueur du tableau de joueurs dans cette équipe a augmenté. De plus, il vérifie que le nouveau joueur est bien inclus dans la liste des joueurs de l'équipe.

Test générateur de tournoi:

- **Test de génération des poules (generatePoules) :**

But : S'assurer que les poules sont correctement générées à partir des équipes fournies, avec le nombre attendu d'équipes par poule.

- **Test de simulation des matches de poules (simulatePoulesMatches) :**

But : Vérifier que la simulation des matches des poules fonctionne comme prévu et sélectionne correctement les équipes qualifiées pour les étapes suivantes du tournoi.

- **Test de génération des phases finales (generateFinalStages) :**

But : Confirmer que les étapes finales du tournoi sont générées correctement après la simulation des matches de poules, avec un nombre approprié d'équipes dans chaque étape, y compris la finale.

- **Test de génération complète du tournoi (generateTournament) :**

But : Valider que le tournoi est entièrement généré, des poules jusqu'à la finale, et que le résultat final n'est pas vide, signifiant qu'une équipe est déclarée gagnante.

- **Test d'élimination d'une équipe (eliminateTeam) :**

But : S'assurer que la fonction d'élimination retire correctement une équipe du tournoi, réduisant ainsi le nombre total d'équipes restantes.

- **Test de qualification des équipes après matches de poules (getQualifiedTeams) :**

But : Confirmer que la méthode renvoie les équipes qualifiées après les matches de poules, garantissant que le processus de qualification fonctionne comme prévu.

Analyse des risques

Modèle
L’analyse est jointe à ce document (format Excel) En voici un extrait :

Système	Sous-système	Composant	Fonction	Modes de défaillance	Causes des défaillance	Effets d'une défaillanc	G	F	D	Criticité	Actions correctives et préventive
Générateur d'équipes	Répartition des joueurs	Algorithme de répartition	Création d'équipes équilibrées	Répartition inégale des compétences	Algorithme de répartition défectueux	Compromission de l'équité du tournoi	4	2	2	16	Réviser et optimiser l'algorithme de répartition
Générateur d'équipes	Ajout de joueurs	Fonctionnalité d'ajout de joueurs	Ajout de nouveaux joueurs aux équipes	Ajout incorrect ou échec d'ajout de nouveaux joueurs	Bugs dans la fonction d'ajout	Déséquilibre des équipes	3	3	3	27	Tester rigoureusement la fonctionnalité, correction des bugs
Générateur de tournois	Génération des poules	Algorithme de génération	Formation des poules de tournoi	Poules déséquilibrées ou incorrectement formées	Logique de génération incorrecte	Impact sur l'équité et la progression du tournoi	4	2	3	24	Améliorer l'algorithme de génération des poules, tests supplémentaires
Générateur de tournois	Simulation des matches	Simulation des matches de poules	Sélection des équipes qualifiées	Résultats de simulation incorrects	Erreurs dans l'algorithme de simulation	Qualification inappropriée des équipes	3	3	4	36	Valider et améliorer la logique de simulation, tester avec divers scénarios
Générateur de tournois	Étapes finales	Génération des étapes finales	Progression correcte vers la finale	Échec à générer les étapes finales correctement	Défaut dans la logique de progression	Interruption ou invalidation du tournoi	5	2	2	20	Vérifier la logique de progression, tests d'intégration
Générateur de tournois	Élimination des équipes	Gestion des équipes	Élimination des équipes du tournoi	Élimination incorrecte des équipes	Erreur dans la gestion des équipes	Perturbation des matches suivants	3	2	4	24	Assurer une gestion correcte des équipes, tests approfondis

Répartition des tests par niveau

Modèle

La matrice de tests est jointe à ce document (format Excel)

En voici un extrait :

niveau de tests	responsable	validateur	consulté	informé	Suivi	KPI
tests exploratoires	test manager	QA Manager	Architecte, Chef de projet	Membres du projet	résultat des campagnes	Bugs trouvés
tests de bout en bout	test manager	QA Manager	Architecte, Chef de projet	Membres du projet	Résultats d'exécution dans la CI	taux de défaillance
tests d'intégration	test manager + architecte	QA Manager	Lead développeur	Membres du projet	Résultats d'exécution dans la CI	taux de défaillance
tests service	test manager + architecte	QA Manager	Lead développeur	Membres du projet	Résultats d'exécution dans la CI	taux de défaillance
tests unitaires	Lead développeur	architecte	QA Manager	Membres du projet	Taux de couverture technique et fonctionnel	taux de défaillance
analyse de code	Lead développeur	architecte	QA Manager	Membres du projet	Résultats d'exécution dans la CI	Dette technique

Approbations

L'approbation de ce document vaut applicabilité immédiate au sein des équipes projet sur la version à laquelle il est lié et jusqu'à la date de réévaluation indiquée

Les intervenants (nom) : Oussama DAHMANI & Maxim COSTA

Commentaires

Commentaires éventuels sur les étapes suivantes

Ajouter la possibilité d'ajouter des équipes depuis un csv.