



Call of Rum

Projet ILU4



L'organisation

NDOYE Assane - Chef de Projet et Responsable Tests Fonctionnels

RICHARD Jérémy - Responsable Développement Noyau

JOSEPH Wilkens Marc Johnley - Responsable Spécifications

IVANOVA Alina - Responsable Modélisation

FRANCES-GENTILLET Solène - Responsable IHM

LAFORGE Matéo - Responsable Technique

Parties du Projet Testées

Actions des joueurs

- Boire des liquides et leurs effets

Déplacement des joueurs sur le plateau de jeu

- Dépassement de la dernière case
- Atteinte de la dernière case
- Déplacement sur un nombre de cases entre 2 et 12
- Combats entre joueurs
- Déclenchement d'événements spéciaux

Techniques de Test Utilisées

Tests Unitaire : Utilisation de JUnit 5.7.1 pour vérifier les comportements des méthodes individuellement.

- Couverture de 66% des methodes du package model
- Couverture de 38% des methodes du package controller

Tests d'Intégration : Vérification de l'interaction entre les différentes parties du système (déplacement des joueurs, déclenchement des événements).

Tests Fonctionnels : Évaluation des exigences fonctionnelles du système, telles que les règles de déplacement et les conditions de victoire.

Test Driven Development (TDD) : Écriture de tests avant le développement de certaines fonctionnalités pour s'assurer qu'elles répondent aux spécifications.

Stubs : Utilisation de stubs pour simuler les composants externes et isoler les tests unitaires, permettant de tester les fonctionnalités spécifiques sans dépendance aux autres composants.

Méthodologie des Tests

- **Initialisation :**
 - Configuration des objets de test (plateau, joueurs, inventaire).
 - Préparation des préconditions spécifiques pour chaque test.
- **Exécution :**
 - Lancement des actions des joueurs (déplacements, consommation de liquides).
 - Utilisation de générateurs de valeurs fixes pour les tests prévisibles.
- **Validation :**
 - Utilisation d'assertions pour comparer les résultats obtenus aux résultats attendus.
 - Enregistrement et comparaison des états avant et après les actions.
- **Scripting :**
 - Écriture de scripts de test pour automatiser les tests et garantir la répétabilité.
- **Stubs :**
 - Implémentation de stubs pour les composants non développés ou externes afin de faciliter les tests unitaires.
 - Exemple : Utilisation de **BoundaryStub** pour simuler les frontières du plateau lors des tests de déplacement,

Bilan des Tests

Analyse des Résultats :

Portée des Tests :

- Couvre toutes les principales fonctionnalités liées au déplacement et à la consommation de liquides.

Couverture :

- Test des scénarios principaux et des cas extrêmes.
- Tests des interactions entre les différents composants du système.
- Tests unitaires couvrant 90% du code du modèle.

Réutilisabilité :

- Les scripts de test sont modulaires et peuvent être réutilisés et adaptés pour les futures phases de maintenance et d'évolution du système.

Succès des Tests :

- Tous les tests ont réussi, confirmant que les fonctionnalités respectent les spécifications et les exigences.

Conclusion

Le travail de test a permis de vérifier avec succès les comportements attendus dans diverses configurations du jeu.

Les résultats montrent que le système est robuste et fonctionnel, avec une couverture complète des cas d'utilisation principaux.

Cette présentation met en lumière non seulement les tests fonctionnels et d'intégration, mais également l'importance des tests unitaires qui garantissent une couverture de 90% du code du modèle, assurant ainsi une base solide et fiable pour le projet.

Lien utile

Le rapport de Test du Usecase Move :

https://github.com/Projet-Pirate-ILU4/call_of_rum/blob/main/documents/rapport_test/rapport_test_Drink.pdf

Le rapport de Test du Usecase Drink :

https://github.com/Projet-Pirate-ILU4/call_of_rum/blob/main/documents/rapport_test/rapport_test_Move.pdf

Le rapport de couverture:

https://github.com/Projet-Pirate-ILU4/call_of_rum/blob/main/htmlReport/index.html