

Nom du projet: SAE 2.01	Version: 1.0
Document: Dossier de tests	Date: 22/05/2023
Responsable de la rédaction: Ouvrard Maxence	

Dossier de tests

1. Introduction

Le document suivant à pour but de tester les fonctionnalités principales du niveau 1 de la SAE RPG. Il va se centrer sur les méthodes de la classe Aventure, c'est-à-dire la classe la plus importante du projet.

2. Description de la procédure de test

La classe Java doit être capable de réaliser des opérations et de savoir faire des comparaisons afin de trouver la quête que le joueur doit ensuite réaliser. Pour ce faire, il doit analyser des préconditions, savoir faire des calculs de distance, savoir si la solution que nous souhaitons avoir est efficace ou exhaustive.

3. Description des informations à enregistrer pour les tests

Campagne de test

Produit testé: La classe java Aventure et Quete	
Date de début: 22/05/23	Date de finalisation: 07/06/23
Tests à appliquer: calculDeplacement(), niveau1(), preconditionRemplies(), et testPrecon()	
Responsable de la campagne de test: Ouvrard Maxence	

Tests

Identification du test: calculDeplacement()	Version: 1.0
Description du test: Commençons en testant la fonction calculDeplacement(), en vérifiant tous les cas possibles.	
Ressources requises: Le logiciel IntelliJ et la bibliothèque junit.	

Responsable: Gravier Kylian		
Cas testé		Résultats attendus
P1	avec posFin plus petit que posDebut	un entier positif
P2	avec posFin plus grand que posDebut	un entier positif
P3	avec posFin égal à PosDebut	0

Résultat du test

Référence du test appliqué: calculDeplacement()			
Responsable: Gravier Kylian			
Date de l'application du test: 27/05			
Cas testé		Résultat attendu	Résultat du test
P1	avec posFin plus petit que posDebut	int positif	OK
P2	avec posFin plus grand que posDebut	int positif	OK
P3	avec posFin égal à PosDebut	0	OK

Tests

Identification du test: preconditionsRemplies()		Version: 1.0
Description du test: Testons la méthode preconditionsRemplies() qui doit permettre de retourner un boolean si toutes les préconditions sont remplies pour une quete donnée.		
Ressources requises: Le logiciel IntelliJ et la bibliothèque junit.		
Responsable: Gravier Kylian		
Cas testé		Résultats attendus
P1	pas de précondition	true

P2	preconditions non faite par le joueur	false
----	---------------------------------------	-------

Résultats de tests

Référence du test appliqué: preconditionsRemplies()			
Responsable: Gravier Kylian			
Date de l'application du test: 27/05			
Cas testé		Résultat attendu	Résultat du test
P1	pas de précondition	true	OK
P2	preconditions non faite par le joueur	false	OK

Tests

Identification du test: niveau1()		Version: 1.0
Description du test: Testons la méthode niveau1() qui doit permettre de retourner une solution efficace ou exhaustive.		
Ressources requises: Le logiciel IntelliJ et la bibliothèque junit.		
Responsable: Gravier Kylian		
Cas testé		Résultats attendus
P1	parChoix = 1	solution efficace(pas toutes les quêtes)
P2	parChoix = 2	solution exhaustive(toutes les quêtes)

Résultats de tests

Référence du test appliqué: niveau1()			
Responsable: Gravier Kylian			
Date de l'application du test: 27/05			
Cas testé		Résultat attendu	Résultat du test
P1	1	pas toutes les	OK

		quêtes	
P2	2	toutes les quêtes	OK

Tests

Identification du test: testPrecon()		Version: 1.0
Description du test: Testons la méthode testPrecon() qui se situe dans la classe Quete et qui vérifie si une quête nécessite des préconditions ou non.		
Ressources requises: Le logiciel IntelliJ et la bibliothèque junit.		
Responsable: Ouvrard Maxence		
Cas testé		Résultats attendus
P1	pas de preconditions	true
P2	nécessite des préconditions	false

Résultats de tests

Référence du test appliqué: testPrecon()			
Responsable: Ouvrard Maxence			
Date de l'application du test: 28/05			
Cas testé		Résultat attendu	Résultat du test
P1	pas de preconditions	true	OK
P2	nécessite des préconditions	false	OK

4. Conclusion:

Les fonctions pouvant être testées fonctionnent, pour les autres, elles nécessitent certaines données qui ne se trouvent que dans certaines méthodes. Il est donc impossible de les tester en tant que telle. Pour cela, il faut créer ces même fonctions avec des paramètres différents.