

SAE_2.01 SAE_2.02: Dossier de tests

RPG	Version: 1.0
Document: Dossier de tests	Date: 12/05/2023
Responsable de la rédaction: DEGAT Teddy	

Dossier de tests

1. Introduction

Ce document comporte tous les tests réalisés dans le but de créer un RPG . L'objectif est la validation partielle du système.

2. Description de la procédure de test

Les tests auront pour but de vérifier différentes méthodes des classes Joueur, Position, Quete et Scenario. Sont utilisés les tests unitaires en boîte noire.

3. Description des informations à enregistrer pour les tests

1. Campagne de test

Produit testé: Joueur, Position, Quete, Scenario	
Configuration logicielle :	
Configuration matérielle:	
Date de début: 19/05/2023	Date de finalisation: 06/06/2023
Tests à appliquer: Joueur: -mouvement Position: -posGap Quete: -lecturePos -getQuestPrecond -questExists -hasPrecond Scenario : -getLastQuest -getFirstQuests	
Responsable de la campagne de test: DEGAT Teddy, CHOISY Alexis	

2. Tests

Identification du test: mouvement()	Version: 1.0
Description du test: Vérifier au bon fonctionnement des déplacements	
Ressources requises:	
Responsable: DEGAT Teddy	

Cas testé	Acteur	Actions	Résultats attendus
P1	int[] {0 , 0}	donne le nombre de déplacements requis pour atteindre {4 , 5)	9
P2	int[] {3 , 2}	donne le nombre de déplacements requis pour atteindre {0 , 0)	5
P3	int[] {2 , 1}	donne le nombre de déplacements requis pour atteindre {5 , 4)	6
P4	int[] {0 , 0}	donne le nombre de déplacements requis pour atteindre {0 , 0)	0

Identification du test: posGap()	Version: 1.0
Description du test: Vérifier au bon fonctionnement de la différence entre deux positions	
Ressources requises:	
Responsable: DEGAT Teddy	

Cas testé	Acteur	Actions	Résultats attendus
P1	int[] {0 , 0} int[] {4 , 5}	Donne les coordonnées des déplacements nécessaires	int[] {4 , 5}
P2	int[] {3 , 2} int[] {0 , ,0}	Donne les coordonnées des déplacements nécessaires	int[] {3 , 2}
P3	int[] {2 , 1} int[] {5 , 4}	Donne les coordonnées des déplacements nécessaires	int[] {3 , 3}
P4	int[] {0 , 0} int[] {0 , 0}	Donne les coordonnées des déplacements nécessaires	int[] {0 , 0}

Identification du test: lecturePos()	Version: 1.0
Description du test: Vérifier au bon fonctionnement de la lecture d'une position	
Ressources requises:	
Responsable: DEGAT Teddy	

Cas testé	Acteur	Actions	Résultats attendus
P1	"1 (0, 0) () 0 0 test"	Donne la position récupérée de la chaîne de caractères	int[] {0 , 0}
P2	"1 (0, 1) () 0 0 test"	Donne la position récupérée de la chaîne de caractères	int[] {0 , 1}
P3	"1 (1, 0) () 0 0 test"	Donne la position récupérée de la chaîne de caractères	int[] {1 , 0}
P4	"1 (1, 1) () 0 0 test"	Donne la position récupérée de la chaîne de caractères	int[] {1 , 1}

Identification du test: getQuestPrecond()	Version: 1.0
Description du test: Vérifie le bon fonctionnement de la création d'un tableau d'entier à partir d'une chaîne de caractères	
Ressources requises:	
Responsable: CHOISY Alexis	

Cas testé	Acteur	Actions	Résultats attendus
P1	"1 (4, 3) () 0 0 test"	Transcrit la chaîne de caractères en préconditions	int[] {0, 0, 0, 0}
P2	"1 (4, 3) ((1,),) 0 0 test"	Transcrit la chaîne de caractères en préconditions	int[] {1, 0, 0, 0}
P3	"1 (4, 3) ((1,), 2,)) 0 0 test"	Transcrit la chaîne de caractères en préconditions	int[] {1, 0, 2, 0}
P4	"1 (4, 3) ((1,), (2, 3)) 0 0 test"	Transcrit la chaîne de caractères en préconditions	int[] {1, 0, 2, 3}
P5	"1 (4, 3) ((1, 2,),) 0 0 test"	Transcrit la chaîne de caractères en préconditions	int[] {1, 2, 0, 0}
P6	"1 (4, 3) ((1, 2), (3,)) 0 0 test"	Transcrit la chaîne de caractères en préconditions	int[] {1, 2, 3, 0}
P7	"1 (4, 3) ((1, 2), (, 3)) 0 0 test"	Transcrit la chaîne de caractères en préconditions	int[] {1, 2, 0, 3}
P8	"1 (4, 3) ((1, 2), (3, 4)) 0 0 test"	Transcrit la chaîne de caractères en préconditions	int[] {1, 2, 3, 4}

Identification du test: questExists()	Version: 1.0
Description du test: Vérifie le bon fonctionnement d'un booléen disant si une quête existe ou non	
Ressources requises:	
Responsable: CHOISY Alexis	

Cas testé	Acteur	Actions	Résultats attendus
P1	linkedList (1,2,3,4)	Regarde si une quête existe	True
P2	linkedList (1,2,3,4)	Regarde si une quête existe	False

Identification du test: hasPrecond()	Version: 1.0
Description du test: Vérifie le bon fonctionnement du booléen indiquant si une quête a une précondition ou non	
Ressources requises:	
Responsable: CHOISY Alexis	

Cas testé	Acteur	Actions	Résultats attendus
P1	"1 (1, 1) () 0 0 test"	Transcrit la chaîne de caractère en booléen	False
P2	"1 (1, 1) ((1,2),(3,4)) 0 0 test"	Transcrit la chaîne de caractère en booléen	True

Identification du test: getLastQuest()	Version: 1.0
Description du test: Vérifie le bon fonctionnement de cette méthode renvoyant la quête à finir en dernier	
Ressources requises: scenario_0.txt	
Responsable: CHOISY Alexis	

Cas testé	Acteur	Actions	Résultats attendus
P1	scenario_0.txt	Renvoie la dernière quête à réaliser	0 (1,1) ((3,4),) 4 350 v aincre Araignée lunaire

Identification du test: getFirstQuests()	Version: 1.0
Description du test: Vérifie le bon fonctionnement des méthodes renvoyant les quêtes à réaliser en premier	
Ressources requises: scenario_1.txt	
Responsable: DEGAT Teddy	

Cas testé	Acteur	Actions	Résultats attendus
P1	scenario_1.txt	Renvoie la ou les premières quêtes à réaliser	1 (3, 1) () 2 50 dialoguer avec Alaric le mage noir 4 (4, 0) () 2 100 explorer porte de Ifha Ennore

Contenu de scenario_0.txt:

1|(4, 3)|()|2|100|explorer pic de Bhanborim
2|(3, 1)|((1,),)|1|150|dialoguer avec Kaela la chaman des esprits
3|(0, 4)|((2,),)|3|200|explorer palais de Ahehona
4|(3, 2)|((2,),)|6|100|vaincre Loup Géant
0|(1, 1)|((3, 4,),)|4|350|vaincre Araignée lunaire

Contenu de scenario_1.txt:

2|(2, 2)|((4, 1),)|2|100|explorer tombeau de Reha Thalor
5|(4, 3)|((1, 4), (2,))|1|150|explorer jardin de Syhe Lenora
3|(1, 0)|((4,), (1, 2))|7|100|dialoguer avec Morrigan la déesse de la mort
1|(3, 1)|()|2|50|dialoguer avec Alaric le mage noir
4|(4, 0)|()|2|100|explorer porte de Ifha Ennore
0|(2, 3)|((3,),)|3|400|explorer collines de Kortorhm

3. Résultats de test

Référence du test appliqué : mouvement
Responsable : DEGAT Teddy
Date de l'application du test : 05/06/2023
Résultat du test : OK
Occurrences des résultats : les résultats sont satisfaisants

Référence du test appliqué : posGap
Responsable : DEGAT Teddy
Date de l'application du test : 02/06/2023
Résultat du test : OK
Occurrences des résultats : les résultats sont satisfaisants

Référence du test appliqué : lecturePos
Responsable : DEGAT Teddy
Date de l'application du test : 02/06/2023
Résultat du test : OK
Occurrences des résultats : les résultats sont satisfaisants

Référence du test appliqué : getQuestPrecond
Responsable : CHOISY Alexis
Date de l'application du test : 02/06/2023
Résultat du test : OK
Occurrences des résultats : les résultats sont satisfaisants

Référence du test appliqué : questExists
Responsable : CHOISY Alexis
Date de l'application du test : 02/06/2023
Résultat du test : OK
Occurrences des résultats : les résultats sont satisfaisants

Référence du test appliqué : hasPrecond
Responsable : CHOISY Alexis
Date de l'application du test : 02/06/2023
Résultat du test : OK
Occurrences des résultats : les résultats sont satisfaisants

Référence du test appliqué : getLastQuest
Responsable : CHOISY Alexis
Date de l'application du test : 05/06/2023
Résultat du test : OK
Occurrences des résultats : les résultats sont satisfaisants

Référence du test appliqué : getFirstQuests
Responsable : DEGAT Teddy
Date de l'application du test : 05/06/2023
Résultat du test : OK
Occurrences des résultats : les résultats sont satisfaisants

4. Conclusions

Tous les tests sont concluants, au fur et à mesure de la conception les tests ont été satisfaisants.