

Rapport test JUnit

TestUpdateMapAndSimulation

L'objectif de la méthode updateMapAndSimulation est de mettre à jour l'affichage et la classe Simulation en fonction de ce qu'on lui fait passer en paramètre.

On a décidé de la tester de manière exhaustive avec la technique de boîte noire, car celle-ci est assez compliquée et dépend d'autres fonctions que l'on n'a pas testées, le graphe de flot pourrait devenir assez dense, si on prend en compte les autres fonctions.

Pour simplifier au maximum, on a découpé les tests en différentes catégories.

Tous les tests ont été effectués sur une map initialisée par défaut d'une taille de 5x5.

Test interaction entre entité

Classe Loup :

- Pas positionner
- Positionner [x, y]

Classe Mouton :

- Pas positionner
- Positionner [x, y]

Action:

- Positionner Loup sur case "Vide"
- Positionner Loup sur case "Loup"
- Positionner Loup sur case "Mouton"
- Positionner Mouton sur case "Vide"
- Positionner Mouton sur case "Loup"
- Positionner Mouton sur case "Mouton"

n°	Loup	Mouton	Action	Résultat
1	Pas positionner	Pas positionner	Positionner loup sur case "Vide" Coordonnée 1, 1	Loup dans simulation au coordonné 1, 1 Mouton null dans simulation Image du loup sur la map 1, 1 Reste image bord rocher, herbe intérieur
2	Pas positionner	Pas positionner	Positionner Loup sur case "Loup"	Ce cas n'existe pas
3	Pas positionner	Pas positionner	Positionner Loup sur case "Mouton"	Ce cas n'existe pas
4	Pas positionner	Pas positionner	Positionner Mouton sur case "Vide" Coordonnée 1, 1	Loup null dans simulation Mouton dans simulation au coordonné 1, 1 Image du mouton sur la map 1, 1 Reste image bord rocher, herbe intérieur
5	Pas positionner	Pas positionner	Positionner Mouton sur case "Loup"	Ce cas n'existe pas
6	Pas positionner	Pas positionner	Positionner Mouton sur case "Mouton"	Ce cas n'existe pas

7	Pas positionner	Positionner [1, 1]	Positionner Loup sur case "Vide" Coordonnée 1, 2	Loup dans simulation au coordonnée 1, 2 Mouton dans simulation au coordonnée 1, 1 Image du loup sur la map 1, 2 Image du mouton sur la map 1, 1 Reste image bord rocher, herbe intérieur
8	Pas positionner	Positionner [1, 1]	Positionner Loup sur case "Loup"	Ce cas n'existe pas
9	Pas positionner	Positionner [1, 1]	Positionner Loup sur case "Mouton" Coordonnée 1, 1	Loup dans simulation au coordonnée 1, 1 Mouton <u>null</u> dans simulation Image du loup sur la map 1, 1 Reste image bord rocher, herbe intérieur
10	Pas positionner	Positionner [1, 1]	Positionner Mouton sur case "Vide" Coordonnée 1, 2	Loup <u>null</u> dans simulation Mouton dans simulation au coordonnée 1, 2 Image du mouton sur la map 1, 2 Reste image bord rocher, herbe intérieur
11	Pas positionner	Positionner [1, 1]	Positionner Mouton sur case "Loup"	Ce cas n'existe pas
12	Pas positionner	Positionner [1, 1]	Positionner Mouton sur case "Mouton" Coordonnée 1, 1	Aucun changement
13	Positionner [1, 1]	Pas positionner	Positionner Loup sur case "Vide" Coordonnée 1, 2	Loup dans simulation au coordonnée 1, 2 Mouton <u>null</u> dans simulation Image du loup sur la map 1, 2 Reste image bord rocher, herbe intérieur
14	Positionner [1, 1]	Pas positionner	Positionner Loup sur case "Loup" Coordonnée 1, 1	Aucun changement
15	Positionner [1, 1]	Pas positionner	Positionner Loup sur case "Mouton"	Ce cas n'existe pas
16	Positionner [1, 1]	Pas positionner	Positionner Mouton sur case "Vide" Coordonnée 1, 2	Loup dans simulation au coordonnée 1, 1 Mouton dans simulation au coordonnée 1, 2 Image du loup sur la map 1, 1 Image du mouton sur la map 1, 2 Reste image bord rocher, herbe intérieur
17	Positionner [1, 1]	Pas positionner	Positionner Mouton sur case "Loup" Coordonnée 1, 1	Loup <u>null</u> dans la simulation Mouton dans simulation au coordonnée 1, 1 Image du mouton sur la map 1, 1 Reste image bord rocher, herbe intérieur
18	Positionner [1, 1]	Pas positionner	Positionner Mouton sur case "Mouton"	Ce cas n'existe pas
18	Positionner [1, 1]	Pas positionner	Positionner Mouton sur case "Mouton"	Ce cas n'existe pas
19	Positionner [1, 1]	Positionner [1, 2]	Positionner Loup sur case "Vide" Coordonnée 1, 3	Loup dans simulation au coordonnée 1, 3 Mouton dans simulation au coordonnée 1, 2 Image du loup sur la map 1, 3 Image du mouton sur la map 1, 2 Reste image bord rocher, herbe intérieur
20	Positionner [1, 1]	Positionner [1, 2]	Positionner Loup sur case "Loup"	Aucun changement
21	Positionner [1, 1]	Positionner [1, 2]	Positionner Loup sur case "Mouton" Coordonnée 1, 2	Loup dans simulation au coordonnée 1, 2 Mouton <u>null</u> dans la simulation Image du loup sur la map 1, 2 Reste image bord rocher, herbe intérieur
22	Positionner [1, 1]	Positionner [1, 2]	Positionner Mouton sur case "Vide" Coordonnée 1, 3	Loup dans simulation au coordonnée 1, 1 Mouton dans simulation au coordonnée 1, 3 Image du loup sur la map 1, 1 Image du mouton sur la map 1, 3 Reste image bord rocher, herbe intérieur
23	Positionner [1, 1]	Positionner [1, 2]	Positionner Mouton sur case "Loup" Coordonnée 1, 1	Loup <u>null</u> dans la simulation Mouton dans simulation au coordonnée 1, 1 Image du mouton sur la map 1, 1 Reste image bord rocher, herbe intérieur
24	Positionner [1, 1]	Positionner [1, 2]	Positionner Mouton sur case "Mouton" Coordonnée 1, 2	Aucun changement

Test positionnement en fonction de coordonnées.

Classe X :

- x = 0
- x ∈ [1; 3]
- x = 4

Classe Y :

- y = 0
- y ∈ [1; 3]
- y = 4

Classe Décor:

- Herbe
- Cactus
- Fleur
- Rocher
- Exit

n°	X	Y	Décor	Résultat
25	0	0	Herbe	InvalidPositionException
26	0	0	Cactus	InvalidPositionException
27	0	0	Fleur	InvalidPositionException
28	0	0	Rocher	Aucun changement
29	0	0	Exit	InvalidPositionException
30	0	1	Herbe	InvalidPositionException
31	0	1	Cactus	InvalidPositionException
32	0	1	Fleur	InvalidPositionException
33	0	1	Rocher	Aucun changement
34	0	1	Exit	Image exit en 0, 1 Exit dans simulation 0, 1
35	0	4	Herbe	InvalidPositionException
36	0	4	Cactus	InvalidPositionException
37	0	4	Fleur	InvalidPositionException
38	0	4	Rocher	Aucun changement
39	0	4	Exit	InvalidPositionException
40	1	0	Herbe	InvalidPositionException
41	1	0	Cactus	InvalidPositionException
42	1	0	Fleur	InvalidPositionException
43	1	0	Rocher	Aucun changement
44	1	0	Exit	Image exit en 1, 0 Exit dans simulation 1, 0
45	1	1	Herbe	Aucun changement
46	1	1	Cactus	Image Cactus en 1, 1 Cactus dans simulation 1, 1

47	1	1	Fleur	Image Fleur en 1, 1 Fleur dans simulation 1, 1
48	1	1	Rocher	Image Rocher en 1, 1 Rocher dans simulation 1, 1
49	1	1	Exit	InvalidPositionException
50	1	4	Herbe	InvalidPositionException
51	1	4	Cactus	InvalidPositionException
52	1	4	Fleur	InvalidPositionException
53	1	4	Rocher	Aucun changement
54	1	4	Exit	Image Exit en 1, 4 Exit dans simulation 1, 4
55	4	0	Herbe	InvalidPositionException
56	4	0	Cactus	InvalidPositionException
57	4	0	Fleur	InvalidPositionException
58	4	0	Rocher	Aucun changement
59	4	0	Exit	InvalidPositionException
60	4	1	Herbe	InvalidPositionException
61	4	1	Cactus	InvalidPositionException
62	4	1	Fleur	InvalidPositionException
63	4	1	Rocher	Aucun changement
64	4	1	Exit	Image Exit en 4, 1 Exit dans simulation 4, 1
65	4	4	Herbe	InvalidPositionException
66	4	4	Cactus	InvalidPositionException
67	4	4	Fleur	InvalidPositionException
68	4	4	Rocher	Aucun changement
69	4	4	Exit	InvalidPositionException

Test interaction entité sur décor

Classe Element1 :

- [Loup, Mouton]

Classe Element2 :

- [Herbe, Cactus, Fleur]

- [Rocher, Exit]

n°	Element1	Element2	Resultat
70	Loup	Herbe	Image Loup Loup initialisé
71	Loup	Rocher	Image Rocher Loup null

Test interaction décor sur entité

Classe Element1 :

- [Herbe, Cactus, Fleur, Rocher]

Classe Element2 :

- Loup

- Mouton

n°	Element1	Element2	Resultat
72	Herbe	Loup	Image Herbe Loup null
73	Rocher	Mouton	Image Rocher Mouton null

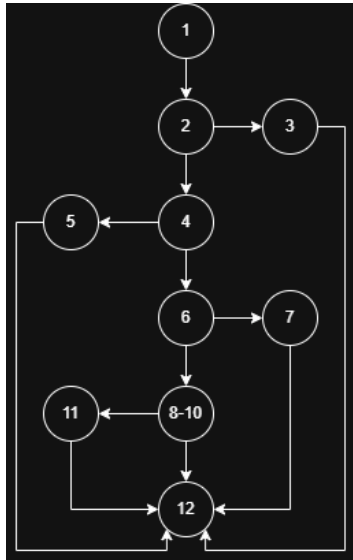
Un dernier test est de vérifier si on positionne une sortie alors qu'un est déjà positionné.

n°	Description	Résultat
74	Positionner Exit quand il y a déjà une Exit	Une seul Exit

TestIsValidMap

L'objectif de cette méthode est de vérifier si le labyrinthe est valide. Elle teste si une sortie, un loup et un mouton sont bien présents et également elle vérifie si le labyrinthe est connexe.

Graphe de flot :



$E = 13$

$N = 10$

$V(G) = 13 - 10 + 2 = 5$

Test n°	Chemin	Etat de la carte	Résultat
1	1 -> 2 -> 3 -> 12	Par défaut	NoExitException
2	1 -> 2 -> 4 -> 5 -> 12	Une sortie	NoWolfException
3	1 -> 2 -> 4 -> 6 -> 7 -> 12	Une sortie, un loup	NoSheepException
4	1 -> 2 -> 4 -> 6 -> 8-10 -> 11 -> 12	Une sortie, un loup, un mouton, une case non accessible	UnconnectedGraphException
5	1 -> 2 -> 4 -> 6 -> 8-10 -> 12	Une sortie, un loup, un mouton	Test validé