

	Ce qu'on évalue	Indications	Pts	Note
F1	Diagramme de cas d'utilisation		1	1
F2	Format du diagramme de classes		1	1
C1	Classe [[Jeu]] qui permet de représenter l'état du jeu			X
		labyrinthe	1	1
		position du monstre	1	1
		sortie	1	1
		tour de jeu	1	1
C2	Classes [[Monstre]] et [[Chasseur]] ont accès aux informations sur l'état du jeu permettant d'afficher les informations utiles pour chaque joueur			X
	[[Monstre]]	position, sortie, murs	2	2
	[[Chasseur]]	cases connues et l'information associée (mur/pas mur, tour de passage du monstre)	2	1
C3	Pas de perte ou fuite d'information	<i>l'évaluation prend en compte le diagramme de classes et la description. Note sur 4, décomposée comme suit:</i>	sur 4	2
	cloisonnées, mais monstre et chasseur accèdent directement aux cases du maze $\Rightarrow (-1)$	les représentations des plateaux sont cloisonnées entre les trois classes et il n'y a pas de fuite d'information	4	X
		représentations cloisonnées avec fuite d'information	-1	X
		[[Jeu]] et [[Monstre]] partagent le même plateau	2	X
		le plateau partagé entre [[Jeu]], [[Monstre]], [[Chasseur]]	1	X
		perte d'information	-1	X
C4	Représentation des données efficace et peu propice aux erreurs			X
		la position du monstre est représentée de manière explicite (et non, par ex., en tant que ICellEvent dans un tableau)	1	0
		les plateaux sont représentés dans des tableaux, éventuellement Map, mais pas sous forme de liste	1	0
D	Déroulement d'un tour de jeu			X
		[[Jeu]] récupère la coordonnée où le joueur veut jouer	1	1
		vérification que la coordonnée est autorisée (monstre)	1	1
		[[Jeu]] met à jour son état (test de victoire, maj position du monstre)	2	2

Équipe : 61

Total sur 20 : 15

C'est bien, mais attention ! le getCell() du maze ne doit pas être public \Rightarrow le chasseur peut accéder directement à la posⁿ du monstre.