

Sciper	Archi (2pts)	Architecture comment	Class (2pts)	Class violation comment	Code style (5pts)	Style violation	general comment	Total rendu2 (22pts)
302064	1.75	[A4]	2.00		3.00	[L2] graphic.h 8 9, gui.h63,64, shape.h 32 34 36 [P2] guic:model()	Bien mais problème de lecture de fichier (plantage)	18.00
326025	2.00		2.00		4.00	[P2]robot.cc:54	Bon code, continuer comme ça	19.50
328364	2.00	ok	2.00	ok	5.00	warning magic number: taille des chaines dans main	Globalement très bien structuré et clair. qq remarques: main() est un peu cryptique, le passage de vector par valeur pourrait pénaliser les performances. warning:style parfois trop compact (lecture et constructeur gui).	21.50
328433	2.00	ok	2.00	ok	4.00	Attention aux variations de style d'indentation [L2]simulation.cc:92,131;robot.cc:35; constantes.h:6;gui.cc:199-200,209,211-213,215-216,240	Rapport bien imagé. Le code à l'air assez bon de l'extérieur, c'est dommage qu'il ne fonctionne souvent pas. Il y a beaucoup de choses à revoir avant le rendu 3.	14.00
329207	2.00		2.00		5.00		Très bien	20.50
329309	2.00	ok	2.00	ok	4.00	Attention aux variations de style d'indentation [L2]simulation.cc:92,131;robot.cc:35; constantes.h:6;gui.cc:199-200,209,211-213,215-216,240	Rapport bien imagé. Le code à l'air assez bon de l'extérieur, c'est dommage qu'il ne fonctionne souvent pas. Il y a beaucoup de choses à revoir avant le rendu 3.	14.00
329942	2.00	OK	0.00	[C2] robot.h:dessin,reinitialiser_avec_zeros ; particule.h:dessin ; gui.h:on_button_clicked_exist. Seuls les constructeurs et les getters définis sur la même ligne que le prototype sont autorisés dans l'interface.	4.00	[L2] simulation.cc:23,25,37,42,46,51,58,66,86,97,103,165-167,171,174 ; robot.h:54,66 ; robot.cc:42,51,85,94,159 ; particule.cc:51,55,58,68,71,74,77 ; constantes.h:11 ; graphic.cc:26,37 ; gui.cc:21,24,31,112,121,124,130,139,144,152,155,170,238,248,261,267,304 ; projet.cc:19	Très bon travail, mais attention au lignes trop longues et aux exigences par rapport aux classes et au dessin	18.75

330515	0.00	[A2.1] [A2.3] [A3][A5]	1.00	[C1] namespace: pas le bon endroit pour robot ni particule, en plus public	2.00	[L2] myWindow on button clicked_save[P2] constructeur mywindow trop long , simulation.cc verification d'integrité trop long	Résultat correcte mais ne respecte pas la donnée	11.50
330686	1.75	[A4]	2.00		3.00	[L2] graphic.h 8 9, gui.h63,64, shape.h 32 34 36 [P2] guic:model()	Bien mais problème de lecture de fichier (plantage)	18.00
333256	1.50	[A2.2]	2.00	OK	5.00	OK	Good work, the code is really clean and nice to read. Continue like that.	21.50
339452	2.00		2.00		4.00	[P2]robot.cc:54	Bon code, continuer comme ça	19.50
339655	2.00		2.00		2.00	[L1]gui.cc:268,339[L2]everywhere[P2] simulation.cc:57,gui.cc:138	Code incomplet, à travailler	10.50
339837	2.00	ok	2.00	ok	3.00	Warning: les cases des switch n'ont pas besoin d'être indenté [L1] gui.cc:315-319,408-422,473; particule.cc:51,33-40; robot.cc:26-30,66-74,etc... [L2] robot.h:111,116,134,115, simulation.cc:179,195	Le code globalement est bon, même si par endroit il est assez compact et dur à lire, surtout dans simulation et robot.cc (essayez de plus utiliser le principe d'abstraction). Pensez aussi à corriger les fautes de style pour le prochain rendu.	20.00
339870	1.75	[A4]	2.00		5.00		Bien mais la sauvegarde est à revoir	18.75
339959	1.00	[A3] shape doit être totalement indépendant du modèle: il représente donc des carrés et des cercles, pas des "robots" et des "particules" (fonctions dessin_particules & dessin_robots) [A5] pareil pour graphic (fonction particules_couleur)	0.00	[C2] simulation.h:set_check ; robot.h:set_nb_update,set_r ; particule.h:set_check_2 ; gui.h:change_ferme, change_ouvert Seuls les constructeurs et les getters définis sur la même ligne que le prototype sont autorisés dans l'interface.	2.00	[L1] particule.cc:27,45 ; shape.h:29-37 [L2] constantes.h:11 (warning) [P2] simulation.cc:40-155 ; gui.cc:29-111	Attention aux indentations, soyez consistants avec le saut à la ligne de l'accolade. Attention également aux définitions de fonction dans l'interface et au respect de l'architecture. Votre code ne compilait pas, j'ai dû moi-même apporter des modifications au code pour que la compilation fonctionne. Tâchez de régler tout ceci pour le prochain rendu, bon courage ! :)	14.25

339964	2.00	OK	0.00	[C1] simulation.cc,e you should have declared it as static [C2] robot.h 92, 98	3.00	[L1]graphic.cc, 9, particule.cc, 10, 17, 19, 26, 29, projet.cc 21 [P2] MyWindow(), Simulation::check_Superposition(), Simulation::draw:planet()	Be sure to check for consistency in your bracket style, and for the indentation on control structure. The execution test passed really well, Continue on this way.	18.00
340062	2.00	Top	1.00	[C1]simulation.cc: the method remove, outside the class Simulation, has access to the vector of particles, simply add it as a method of simulation	5.00	Warnings / Suggestions: 1. Don't mix camelCase, snake_case and PascalCase 2. Inconsistent spacing: use a formatting tool	The code is readable and overall more or less easy to understand, goob job! Suggestions: 1. In project.cc, pass the instance of Simulation to gui (same applies for DrawingArea) as either a pointer or const reference, not as instance 2. The method Simulation::reset is pure magic... do not internally change *this... 3. Do not pass around large vectors of classes and continuisly copying them... pass them as const reference and replace the instances with pointers 4. To simplifying the destructuring of the vector of data from the simulation, use structured binding declaration (c++17)	21.00
340435	2.00	ok	2.00	ok	3.00	[P1] simulation.cc:lecture [L1] gui.cc:350-353,367- 369,379,221,227,215,etc... Warning: [L2] simulation.cc 174, 171	Le code est bien écrit. Attention tout de même aux indentations, il y a trop d'incohérence. Pour vos deux grosses fonctions (fenetre et lecture) pensez à user du principe d'abstraction.	20.00
340497	2.00		2.00		5.00	Trop peu d'espaces	Très bien mais code à rendre plus lisible	22.00
340863	1.50	[A3]constante in shape	1.00	[C2]rorbot.h et particule.h pas sur une ligne	5.00	[L1]simulation.cc:81	Bon travail, attention à bien lire les consignes et tester le code	17.75
341002	2.00	ok	2.00	ok	4.00	[L1] simulation.cc 114-116, 131- 136,robot.cc130-132, gui.cc32,33,36, indentation, accolade différentes dans gui.cc ; warning alignement robot.h52 ; les noms de type commencent par une majuscule (particule), un nom d'attribut ne doit pas être 100% en majuscule (P)	Bon travail mais les conventions sont prises un peu à la légère. Améliorez au moins vos indentations et la cohérence du style des accolades.	21.00

341055	2.00	ok	0.00	[C1] variable globales: simulation.cc 42, 43 robot.cc 24 43, 128, 179 particule.cc 7, 8	5.00	ok	C'est très bien, le code est propre, bravo. Attention aux variables globales, vous auriez pu éviter cette erreur en rajoutant le mot-clé "static"	20.00
341138	1.25	[A4] gui inclu constantes.h [A5] graphic inclu constantes.h	2.00	ok	5.00	ok	Bravo, c'est du bon travail	21.25
341237	2.00		2.00		5.00	Trop peu d'espaces	Très bien mais code à rendre plus lisible	22.00
341489	2.00	Pourquoi constantes.cpp ???	2.00		4.00	[L2]graphic.h:11-14,Simulation.cpp:83,graphic.cpp:8,9,50	Bon travail, attention à la forme du code	17.25
341508	1.75	[A4] gui inclu constantes.h	2.00	votre class simulation est inhabituelle, c'est un mix de C et de C++ mais qui cumulent les défauts des 2 langages.	5.00	ok	C'est bien bravo	21.75
341509	1.25	[A4] gui inclu constantes.h [A5] graphic inclu constantes.h	0.00	[C2] definition de constructeurs sur plusieurs lignes dans le .h (robot.h par exemple)	4.00	[P2] simulation.cc decodage_ligne 89 ligne	C'est bien, vous êtes sur une bonne lancée	16.25
341530	2.00		2.00		3.00	[L2]simulation.cc:4,shape.cc:1,53,constantes.h:4,...[P2]gui.cc:308	Bon travail, continuer comme ça	20.00
341668	2.00		2.00		2.00	[L1]gui.cc:268,339[L2]everywhere[P2]simulation.cc:57,gui.cc:138	Code incomplet, à travailler	10.50
341672	2.00	OK	2.00	OK	5.00	Very Good, be careful to keep the same indentation style, even for different control instructions. Small mistakes at line 42 in gui.cc	Very good job it all work really well.	21.25

341744	1.00	[A3] shape ne doit pas inclure message [A5] graphic ne doit pas inclure constantes	2.00	ok	5.00	Quelques erreurs d'indentation.	Beaucoup de fonctionnalités qui ne fonctionnent pas ou pas comme attendu, il faut que vous modifiez votre code en conséquence avant le rendu 3. Bon courage.	14.75
341898	2.00	OK	1.00	[C1] gui.cc:info_text (variable globale, à corriger) GROS WARNING: pas de variables static dans l'interface (gui.h:17)	2.00	[L1] simulation.cc:33;77,79,82,... ; robot.cc:89 ; particule.cc:38 ; ... [L2] robot.cc:32-33 ; graphic.h:32-33 ; graphic.cc:27,43 [P2] simulation.cc:57-139	Bon travail. Attention au style, notamment aux indentations, surtout aux accolades : soyez consistants dans votre style d'indentation, en particulier dans les blocs if-else. Pour ceux-ci, si l'instruction tient sur la même ligne, ne mettez pas d'accolades. Attention aussi à ne pas utiliser de variables globales. Tâchez de régler le problème d'ouverture de fichiers depuis la ligne de commande d'ici le prochain rendu. Bon courage :)	17.50
341903	2.00	OK	2.00	ok	3.00	[L1] graphic.cc:53-54, robot.cc:45-46,15-18,33-36; shape.cc:67-68, simulation.h:33-71 [P1] simulation.cc:decode Warning: Pas besoin d'indenter les cases dans un switch block. [L2] simulation.cc:234	Globalement le code est bien structuré. Faites tout de même attention à vos indentations. Pour le prochain rendu n'oubliez pas de décomposer la fonction decode de simulation en plus petits bouts	14.50
341942	2.00	ok	2.00	ok	5.00	[L1] gui.cc l40-42	Bon travail!	21.50
341949	1.50	[A3]"constantes.h" inclut dans shape.cc	2.00	ok	4.00	[P2]simulation.cc decodage_ligne trop longue,	Bonne exécution, faire attention à l'architecture de projet.	19.50
341970	2.00	OK	0.00	[C2] robot.h:dessin,reinitialiser_avec_zeros ; particule.h:dessin ; gui.h:on_button_clicked_exist. Seuls les constructeurs et les getters définis sur la même ligne que le prototype sont autorisés dans l'interface.	4.00	[L2] simulation.cc:23,25,37,42,46,51,58,66,86,97,103,165-167,171,174 ; robot.h:54,66 ; robot.cc:42,51,85,94,159 ; particule.cc:51,55,58,68,71,74,77 ; constantes.h:11 ; graphic.cc:26,37 ; gui.cc:21,24,31,112,121,124,130,139,144,152,155,170,238,248,261,267,304 ; projet.cc:19	Très bon travail, mais attention aux lignes trop longues et aux exigences par rapport aux classes et au dessin	18.75

342101	2.00		2.00		5.00	[L2] graphic.h: 25	Bien mais il y a un problème de superposition	21.00
342224	2.00	Warning: include inutile de constants.h dans gui.cc; Warning: Il faut utiliser des constructeurs au lieu de créer des instances et de les setter avec des setters	2.00	ok	4.00	[L1] inconsistent brace style: simulation.cc l71-->73, l74-->76, l77-->79, l82-->93; indentation error dans gui.cc l42-->58	Eviter d'utiliser des setters sans vérification, les constructeurs sont là pour créer les instances. A part, votre code n'est pas mal.	20.25
342283	2.00	ok	2.00	ok	5.00	top	Impeccable. Excellent travail!	22.00
342391	1.50	[A3] shape ne peut que include graphic. Mauvais includes: GTKmm	1.00	[C1] Attributs publics: gui.h: static std::string filename. À corriger GROS WARNING: ne pas mettre de variables static dans les interfaces (graphic.h, simulation.h)	3.00	[L1] fichiers .h: ne pas indenter private/public/protected Ne pas mélanger les styles d'indentation / de placement des accolades gui.cc: rester consistant pour l'espacement des indentations (ex: on_timeout) [L2] simulation.cc:52,71,172-173,177-178,295 ; robot.h:31 ; gui.h:68 ; gui.cc:90,119,142,223,237,287,303,321	Bon travail, mais tâchez de régler votre bug pour la sauvegarde des fichiers pour le prochain rendu et faites attention au style et aux différentes exigences par rapport à l'architecture et aux attributs publics.	16.00
342575	1.50	[A3] shape doit être totalement indépendant du modèle: un argument "type_robot" est donc inapproprié (fonction call_draw_circle)	2.00	OK	4.00	[L1] simulation.cc:126 ; graphic.h:15-23 ; gui.h:38-80 ;	Très beau travail, bravo ! Attention à être consistant sur l'espacement de vos indentations dans votre code, et attention à l'indépendance du module shape	20.50
342906	1.25	[A4] gui inclu constantes.h [A5] graphic inclu constantes.h	0.00	[C2] definition de constructeurs sur plusieurs lignes dans le .h (robot.h par exemple)	4.00	[P2] simulation.cc decodage_ligne 89 ligne	C'est bien, vous êtes sur une bonne lancée	16.25

342907	2.00	ok	2.00	ok	3.00	Warning: les cases des switch n'ont pas besoin d'être indenté [L1] gui.cc:315-319,408-422,473; particule.cc:51,33-40; robot.cc:26-30,66-74,etc... [L2] robot.h:111,116,134,115, simulation.cc:179,195	Le code globalement est bon, même si par endroit il est assez compact et dur à lire, surtout dans simulation et robot.cc (essayez de plus utiliser le principe d'abstraction). Pensez aussi à corriger les fautes de style pour le prochain rendu.	20.00
344310	2.00	ok	2.00	ok	5.00	Warning: [L1] robot.cc 70-77, 110-116 Warning: [L2] simulation: 56, 55, 60	Super code! Quand vous avez une fonction trop proche des 87 caractères par ligne, c'est plus lisible de l'écrire de cette façon: fonction(param1, param2,); Sinon Bravo!	22.00
344321	2.00	ok	2.00	ok	4.00	[P1] gui.cc:Gui() Warning: [L1] shape.cc:79,87,95 [L2] gui.cc:57	Super code! Très lisible et clair (sauf pour décodage_ligne et le constructeur de Gui qui mériterait d'être retoucher). Beau travail!	21.00
344325	2.00	ok	2.00	ok	4.00	[L1]simulation.cc 229, (break; }) graphic.h 11-14 surindenté [L2] constantes.h, simulation.h Warning [P2] graphic.cc set_couleur 44 lignes	C'est bien, pour la prochaine fois il faut un peu retravailler l'indentation	20.50
344416	1.50	[A5] graphique.cc inclut constantes.h	2.00	ok	3.00	[P1] gui.cc:Gui(), simulation.cc:decodage_ligne Warning: [L1] shape.cc:38-45; simulation.cc:66,77-80; gui.cc:170 [L2] graphique.cc:93 Les cases n'ont pas besoin d'être indentés Essayez d'être plus propre dans votre manière de wrapper les lignes	Globalement le code est bon. Essayez de remplir un peu la classe particule qui pour l'instant ressemble plus à une struct. Sinon bon travail!	18.75
344529	1.50	[A3]constante in shape	2.00		4.00	[P2]simulation.cc:165	Noms de header guard trompeurs. Bon travail, continuez comme ça	20.50

344936	2.00	ok	2.00	ok	3.00	[P1] simulation.cc:lecture [L1] gui.cc:350-353,367-369,379,221,227,215,etc... Warning: [L2] simulation.cc 174, 171	Le code est bien écrit. Attention tout de même aux indentations, il y a trop d'incohérence. Pour vos deux grosses fonctions (fenetre et lecture) pensez à user du principe d'abstraction.	20.00
345007	2.00	ok	2.00	ok	5.00		Très bon projet, fonctionnalité complète, beau travail!	22.00
345130	0.00	[A2.1] [A2.3] [A3][A5]	1.00	[C1] namespace: pas le bon endroit pour robot ni particule, en plus public	2.00	[L2] myWindow on button clicked_save[P2] constructeur mywindow trop long , simulation.cc verification d'integrité trop long	Résultat correcte mais ne respecte pas la donnée	11.50
345259	1.50	[A5]: graphic.hh ne doit pas inclure shape.h	2.00	ok	2.00	[L2] shape.cc : l40-61-62-74..., simulation.cc : l49,55..., robot.cc, robot.h.. ; [L1]: robot.h: l16-19-30-38 , you must not indent private and public ; [P2]: simulation.cc: decodage_ligne l44-->l194	Votre code est très difficile à lire car il n'y a presque aucun respect des conventions. Faites attention à cet aspect pour le prochain rendu. C'est dommage car vous avez la majorité des tests qui marchent bien	18.00
345308	1.75	[A4] gui inclu constantes.h	2.00	votre class simulation est inhabituelle, c'est un mix de C et de C++ mais qui cumulent les défauts des 2 langages.	5.00	ok	C'est bien bravo	21.75
345310	2.00	Pas de magic number	2.00		5.00	[L1]graphic.h,gui.cc:30	Bon travail, le rendu demandait que le projet se lance avec un fichier de test (./projet test.txt)	19.00
345473	2.00	OK	2.00	ok	3.00	[L1] graphic.cc:53-54, robot.cc:45-46,15-18,33-36; shape.cc:67-68, simulation.h:33-71 [P1] simulation.cc:decode Warning: Pas besoin d'indenter les cases dans un switch block. [L2] simulation.cc:234	Globalement le code est bien structuré. Faites tout de même attention à vos indentation. Pour le prochain rendu n'oubliez pas de décomposer la fonction decode de simulation en plus petit bouts	14.50
345683	2.00	Pas de magic number	2.00		5.00	[L1]graphic.h,gui.cc:30	Bon travail, le rendu demandait que le projet se lance avec un fichier de test (./projet test.txt)	19.00

345862	2.00	OK	2.00	OK	5.00	Very Good, be careful to keep the same indentation style, even for different control instructions. Small mistakes at line 42 in gui.cc	Very good job it all work really well.	21.25
346154	2.00	OK	1.00	[C1] gui.cc:info_text (variable globale, à corriger) GROS WARNING: pas de variables static dans l'interface (gui.h:17)	2.00	[L1] simulation.cc:33;77,79,82,... ; robot.cc:89 ; particule.cc:38 ; ... [L2] robot.cc:32-33 ; graphic.h:32-33 ; graphic.cc:27,43 [P2] simulation.cc:57-139	Bon travail. Attention au style, notamment aux indentations, surtout aux accolades : soyez consistants dans votre style d'indentation, en particulier dans les blocs if-else. Pour ceux-ci, si l'instruction tient sur la même ligne, ne mettez pas d'accolades. Attention aussi à ne pas utiliser de variables globales. Tâchez de régler le problème d'ouverture de fichiers depuis la ligne de commande d'ici le prochain rendu. Bon courage :)	17.50
346228	2.00		2.00		5.00		Très bien	22.00
346230	2.00		2.00		3.00	[L2]simulation.cc:4,shape.cc:1,53, constantes.h:4,...[P2]gui.cc:308	Bon travail, continuer comme ça	20.00
346255	1.50	[A2.2]	2.00	OK	5.00	OK	Good work, the code is really clean and nice to read. Continue like that.	21.50
346389	2.00	Pourquoi constantes.cpp ???	2.00		4.00	[L2]graphic.h:11-14,Simulation.cpp:83,graphic.cpp:8,9,50	Bon travail, attention à la forme du code	17.25
346483	2.00	ok	2.00	ok	5.00	parfait	Code très bien structuré. Dommage de vous être trompées en lisant la donnée.	21.50

					Top			
					<p>Suggestions / Warnings:</p> <ol style="list-style-type: none"> 1. Don't mix PascalCase, snake_case and camelCase 2. Inconsistent spacing, use a formatting tool 3. Try to find better names (check_si_dieses...) 4. Do not overload operators << 5. You don't need this, when accessing internally members of an instance 6. In the member initializer list, you don't have to change the name of the passed parameters even though they have the same name as the attributes of the class 7. Do not write using std in header file 	<p>The code is readable but sometimes not understandable. Try to simplify it and organize it.</p> <p>Suggestions:</p> <ol style="list-style-type: none"> 1. In gui.cc, MyWindow:open_clicked, you can remove the filter for the c++ files 2. In graphic, even though the constant are marked as const, they should be in a namespace 3. Work on your code style and try to find better names 4. Fix the encapsulation errors, by at least passing them as const reference using getters that return const reference and not directly the vector or even better do the for-loops in the Simulation 		
346515	2.00	Top	1.00	[C1]simulation.cc: Simulation::getspaceship returns the pointer of a member of the class Simulation thereby breaking the encapsulation [C1]robots.cc: erreur_n_r... have reference access to member of the simulation	5.00		21.00	
346518	1.50	[A5]	2.00		5.00		Bien mais problème de lecture de fichier (particule)	21.50
		[A3] shape doit être totalement indépendant du modèle: il représente donc des carrés et des cercles, pas des "robots" et des "particules" (fonctions dessin_particules & dessin_robots) [A5] pareil pour graphic (fonction particules_couleur)		[C2] simulation.h:set_check ; robot.h:set_nb_update,set_r ; particule.h:set_check_2 ; gui.h:change_ferme, change_ouvert Seuls les constructeurs et les getters définis sur la même ligne que le prototype sont autorisés dans l'interface.		[L1] particule.cc:27,45 ; shape.h:29-37 [L2] constantes.h:11 (warning) [P2] simulation.cc:40-155 ; gui.cc:26-111	Attention aux indentations, soyez consistants avec le saut à la ligne de l'accolade. Attention également aux définitions de fonction dans l'interface et au respect de l'architecture. Votre code ne compilait pas, j'ai dû moi-même apporter des modifications au code pour que la compilation fonctionne. Tâchez de régler tout ceci pour le prochain rendu, bon courage ! :)	
346596	1.00		0.00		2.00			14.25
346973	2.00		2.00		5.00		Très bien	20.50

346992	2.00		2.00		2.00	[L1]gui not consistant with other fils[l2]simulation.cc[P2]gui.cc:14	Bon code mais manque de fonctionnalité et de cohérence de style	15.00
347058	2.00		2.00		2.00	[L1]gui not consistant with other fils[l2]simulation.cc[P2]gui.cc:14	Bon code mais manque de fonctionnalité et de cohérence de style	15.00
347115	2.00	ok	2.00	ok	5.00	ok	Bonne exécution en général, code lisible et bon respect des conventions.	21.50
347209	1.50	[A2.3]	2.00		3.00	[P2] 2 fonctions lecture dans simulation	Bien mais hiérarchie de robot à mettre en place	18.50
347288	2.00		2.00		5.00	[L2] graphic.h: 25	Bien mais il y a un problème de superposition	21.00
347341	2.00	ok	2.00	ok	5.00	warning: attention aux lignes trop longues dans simulation.cc et à vos retours à la ligne lors des lignes longues: on ne commence pas un retour à la ligne par une virgule	Bon projet, fonctionnalité complète malgré deux petites erreurs	21.25
347363	1.50	[A3] shape.cc: Mentioning of robots, functions named after robots	2.00	Top	5.00	Warnings / Suggestions: 1. Don't mix camelCase, snake_case	The code is readable and overall easy to understand, goob job! Suggestions: 1. In project.cc, strange declaration of a pointer of class Simulation, simply do Simulation* simu 2. In simulation.cc, the method draw_call should be a method of Simulation, so you don't have to do getters for the vectors of particles, robots... 3. Same applies for robots.cc and particle.cc 4. In simulation.cc, the decoding functions for the robots and particles should be delegated to those modules as it leads to a verbose method 5. Do not pass around large vectors of classes and continuisly copying them... pass them as const reference and replace the instances with pointers 6. The random engine should be resetted after each reading	20.50
347394	2.00	OK	2.00	OK	5.00	OK	Rendu absolument impeccable, excellent travail !! Visez un sans-fautes pour le rendu3 aussi ;)	22.00
347518	1.50	[A2.3]	2.00		3.00	[P2] 2 fonctions lecture dans simulation	Bien mais hiérarchie de robot à mettre en place	18.50

348128	2.00	ok	2.00	ok	4.00	[L1] simulation.cc 114-116, 131-136,robot.cc130-132, gui.cc32,33,36, indentation, accolade différentes dans gui.cc ; warning alignement robot.h52 ; les noms de type commencent par une majuscule (particule), un nom d'attribut ne doit pas être 100% en majuscule (P)	Bon travail mais les conventions sont prises un peu à la légère. Améliorez au moins vos indentations et la cohérence du style des accolades.	21.00
351248	2.00	ok	2.00	ok	5.00	warning magic number: taille des chaines dans main	Globalement très bien structuré et clair. qq remarques: main() est un peu cryptique, le passage de vector par valeur pourrait pénaliser les performances. warning:style parfois trop compact (lecture et constructeur gui).	21.50
355524	2.00	ok	2.00	ok	5.00	top	Impeccable. Excellent travail!	22.00
355529	1.25	[A4] gui ne doit inclure que simulation.h; [A2]: particule.h nr foi pas inclure robot.h	2.00	ok	5.00	parfait	Code vraiment très agréable à lire, très compréhensible. Malheureusement, vous vous êtes aventurés dans une mauvaise direction qui vous a forcés à violer l'architecture fixée dans la figure 9b. Corrigez cet aspect pour le rendu 2	21.25
355583	1.50	[A5] graphic ne doit pas inclure constantes	2.00	ok	4.00	[L1] simulation.cc:19-42;robot.cc:13	Quelques points à revoir avant le rendu 3, encore quelques efforts !	18.50
355638	1.50	[A3]constante in shape	2.00		5.00		Bon travail, continuez comme ça	20.75
355665	2.00	ok	2.00	ok	5.00	ok	Bonne exécution en général, code lisible et bon respect des conventions.	21.50
355726	1.00	[A3] constantes.h dans shape [A5] pareildans graphic	2.00	ok	5.00	Attention à quelques doubles indentations	L'écriture du code est très bon et l'exécution comme attendu, continuer ainsi.	20.00

355728	2.00	[A3] graphic_gui.h dans shape au lieu de graphic.h Vous avez inclu graphic_gui.h dans graphic.h, ce qui importe implicitement gtkmm dans graphic.h. L'idée de faire ces 2 headers et de garder les modules qui importent graphic.h d'être indépendant de gtkmm	2.00	ok	5.00	[L1] gui.h 28 surindenté	C'est bien , quelques petits problèmes à régler mais vous êtes sur la bonne voix	21.75
355732	1.25	[A4] gui ne doit inclure que simulation.h; [A2]: particule.h ne doit pas inclure robot.h	2.00	ok	5.00	parfait	Code vraiment très agréable à lire, très compréhensible. Malheureusement, vous vous êtes aventurés dans une mauvaise direction qui vous a forcés à violer l'architecture fixée dans la figure 9b. Corrigez cet aspect pour le rendu 2	21.25
355779	2.00	ok	2.00	ok	5.00	warning: attention aux lignes trop longues dans simulation.cc et à vos retours à la ligne lors des lignes longues: on ne commence pas un retour à la ligne par une virgule	Bon projet, fonctionnalité complète malgré deux petites erreurs	21.25
355797	1.50	[A3]	0.00	[C1]graphic.cc noir, gris_clair, ... You could have declared them as static, or constexpr	3.00	[L1]simul.h 36, simul.cc 38,64, shape.cc 15, ... [L2]shape.cc 63,67,75, robot.h 54,55	Be careful with the respect of convention. You must really implement the disintegration fonctionnality for the next Rendu. Otherwise, your GUI was good, the opening/saving fonctionnality worked completely, good job for that.	14.00
355807	1.50	[A5] graphic ne doit pas include constantes	2.00	ok	5.00	ok	Très bon projet, les concepts sont bien compris et utilisés, continuez ainsi.	21.00

355809	1.50	[A3] shape doit etre totalement independant du modele: un argument "type_robot" est donc inapproprié (fonction call_draw_circle)	2.00	OK	4.00	[L1] simulation.cc:126 ; graphic.h:15-23 ; gui.h:38-80 ;	Très beau travail, bravo ! Attention à être consistant sur l'espacement de vos indentations dans votre code, et attention à l'indépendance du module shape	20.50
355873	2.00	ok	2.00	ok	5.00	[L1] gui.cc 40-42	Bon travail!	21.50
355884	1.50	[A5] graphic ne doit pas include constantes	2.00	ok	4.00	[L1] simulation.cc:19-42;robot.cc:13	Quelques points à revoir avant le rendu 3, encore quelques efforts !	18.50
355888	2.00		2.00		4.00	[L1] particule.cc:96; robot.cc:192; shape.cc:134-137; simulation.cc:144-154,158-166,128-131,... Warning: Pas besoin d'indeter vos cases, cela rend le code moins lisible	Super code! On voit que vous avez fait attention à rendre le code lisible. Faites tout de même attention à vos indentions.	20.50
355924	2.00	OK	0.00	[C1]graphic.cc noir, gris, ... You could have declared them as static, or constexpr	4.00	[L1]gui.cc22-28,360,444,447,particule.cc18,26-30,simulation.cc	Good job in general. Be sure to be consistent with your bracket style, even for control instructions.	18.00
355943	2.00	ok	0.00	[C2]particule.h,robot.h, nombreuses méthodes non-externalisées	4.00	[L1] gui.cc335-335-336,339-340,348-349etc...	beau code, clair et structuré. Cependant, on demandait d'externaliser les méthodes + petit souci d'indentation dans mise_a_jour(). Les return de vector risquent de pénaliser l'exécution (à voir).	19.00
355944	2.00	ok	2.00	ok	5.00		Excellent projet, code très agréable à lire! Beau travail!	22.00
355958	2.00	ok	2.00	ok	5.00		Excellent travail! N'oubliez pas de faire un make clean avant d'envoyer vos fichiers, le fichier compilé ne tourne pas forcément de la même manière sur tous les ordinateurs	22.00

355973	2.00	ok	2.00	ok	4.50	[L22] mauvais réalignement: gui.cc: l388-->390+l394+l401-402; Warning: shape.h: l65 - l73: commentaires dépassent 87 caractères	Bon code, tous les tests passent, bon travail!	21.50
356107	1.25	[A4] gui inclu constantes.h [A5] gtkmm dans graphic.h (il manque graphic_gui.h)	2.00	ok	5.00	[L1] robot.cc 276	C'est bien bravo	21.25
356113	1.75	[A4]	2.00		5.00		Bien mais la sauvegarde est à revoir	18.75
356118	2.00	OK	2.00	OK	2.00	[L1] simulation.h 20, gui.h 32-43. 46-90, ... [P2] on_file_dialog_response, Window() , lecture(string line), Robot_N()	Be careful with the respect of convention, be sure to fix the open functionality for the Rendu3. In general however, it works well, and don't need much to become very good.	16.25
356263	2.00	ok	2.00	ok	4.00	[L1]robot.cc164,167,170; gui.cc:la liste d'initialisation de fenetre doit être alignée/indentée ; à quoi servent les 4 variables numérotées ptr1 etc... dans new_particles()? warning:shape.cc:un nom de variable ne doit pas être 100% en majuscule	bon projet bien organisé mais parfois obscur: ptr1 à ptr4 dans particules.cc; le style est généralement très clair.	21.00
356278	1.50	[A3]"constantes.h" inclut dans shape.cc	2.00	ok	4.00	[P2]simulation.cc decodage_ligne trop longue,	Bonne exécution, faire attention à l'architecture de projet.	19.50

356321	2.00	[A3] graphic_gui.h dans shape au lieu de graphic.h Vous avez inclu graphic_gui.h dans graphic.h, ce qui importe implicitement gtkmm dans graphic.h. L'idée de faire ces 2 headers et de garder les modules qui importent graphic.h d'être indépendant de gtkmm	2.00	ok	5.00	[L1] gui.h 28 surindenté	C'est bien , quelques petits problèmes à régler mais vous êtes sur la bonne voie	21.75
356342	2.00	ok	2.00	ok	4.00	[L1]: particule.cc 22--> 25 ; inconsistent brace style particule.cc 18--> 26 + robot.cc 98--> 103	Bon travail. Globalement votre code est bien écrit et bien compréhensible	20.50
356345	2.00	ok	2.00	ok	5.00	top	Impeccable. Excellent travail!	22.00
356420	1.25	[A4]connection gui-modèle doit se faire uniquement à travers simulation, ne pas inclure robot.h, particule.h dans gui.cc [A5]constantes.h inclus dans graphic	2.00	ok	5.00	ok	Bonne implémentation en général, certains aspects sont à revoir. Faire attention à l'architecture du projet.	20.25

356471	1.50	[A5] graphic doit être indépendant et ne peut qu'inclure GTKmm. Mauvais includes: constantes.h	0.00	[C1] Attributs publics: gui.h:Simulation simulation ; shape.h: double longueur,S2d centre,double rayon [C2] robot.h: addNbupdate (Seuls les constructeurs et les getters définis sur la même ligne que le prototype sont autorisés dans l'interface.)	4.00	[L1] fichiers .h: ne pas indenter private, public & protected shape.cc:60,68,69-74,76 [L2] gui.cc:20,398,547 (warning)	Très bon travail. Pour les blocs if/else n'ayant qu'une instruction sur la même ligne, évitez l'utilisation des accolades. Attention aux exigences par rapport aux classes et à l'architecture.	18.50
356520	2.00	ok	0.00	[C1] variable globales: simulation.cc 42, 43 robot.cc 24 43, 128, 179 particule.cc 7, 8	5.00	ok	C'est très bien, le code est propre, bravo. Attention aux variables globales, vous auriez pu éviter cette erreur en rajoutant le mot-clé "static"	20.00
356618	1.50	[A3] shape ne peut que include graphic. Mauvais includes: GTKmm	1.00	[C1] Attributs publics: gui.h: static std::string filename. À corriger GROS WARNING: ne pas mettre de variables static dans les interfaces (graphic.h, simulation.h)	3.00	[L1] fichiers .h: ne pas indenter private/public/protected Ne pas mélanger les styles d'indentation / de placement des accolades gui.cc: rester consistant pour l'espacement des indentations (ex: on_timeout) [L2] simulation.cc:52,71,172-173,177-178,295 ; robot.h:31 ; gui.h:68 ; gui.cc:90,119,142,223,237,287,303,321	Bon travail, mais tâchez de régler votre bug pour la sauvegarde des fichiers pour le prochain rendu et faites attention au style et aux différentes exigences par rapport à l'architecture et aux attributs publics.	16.00

356756	2.00	OK	1.00	[C2] : particule.h with setCarre and setpart_correct_file	4.00	[L1] gui.cc 369, graphic.cc 16,33,81, ...	Well done, your project is good. Be sure to look for the issue with the timer, and correct the small mistakes, otherwise good work. Initialize your attribute from the Constructor or with setter is a better practice than just initialize them in the .cc (simulation.cc, robot.cc, particle.cc). CHEkout aginst the indentation rule before Rendu3 ;)	18.25
356765	2.00	ok	2.00		5.00	Une meilleure mise en page du code serait agréable	Un projet qui fonctionne bien mais qui pourrait encore être meilleur avec un peu plus de rigueur.	20.50
356773	2.00	OK	0.00	[C2] robot.h:is_broken ; robot.h:step : Seuls les constructeurs et les getters définis sur la même ligne que le prototype sont autorisés dans l'interface.	3.00	[L1] gui.h: indentation à 2 caractères au lieu de 4 pour le reste du code [L2] constantes.h:4 (warning) [P2] gui.cc:9-101	Très bon travail. Faites néanmoins attention aux définitions de fonctions dans l'interface et au style.	18.00
356775	2.00	ok	2.00	ok	5.00		Bon projet, petite erreur sur le bouton save et un rapport pas très lisible mais le code est bon, beau travail!	21.25
356840	2.00	Warning: include interdit de shape.h dans graphic.h (pas de perte de point car inutile)	2.00	ok	5.00	top	Impeccable. Votre code est très bien écrit, vous respectez bien l'architecture	22.00
356865	2.00	ok	2.00	ok	5.00		Bon projet, petite erreur sur le bouton save et un rapport pas très lisible mais le code est bon, beau travail!	21.25
356869	2.00	ok	2.00	ok	4.00	[L1]simulation.cc 229, (break; }) graphic.h 11-14 surindenté [L2] constantes.h, simulation.h Warning [P2] graphic.cc set_couleur 44 lignes	C'est bien, pour la prochaine fois il faut un peu retravailler l'indentation	20.50
356944	2.00	ok	0.00	[C2]particule.h,robot.h, nombreuses méthodes non-externalisées	4.00	[L1] gui.cc335-335-336,339-340,348-349etc...	beau code, clair et structuré. Cependant, on demandait d'externaliser les méthodes + petit souci d'indentation dans mise_a_jour(). Les return de vector risquent de pénaliser l'exécution (à voir).	19.00

356957	2.00	ok	2.00	ok	4.00	[P1] simulation.cc:lecture Warning: Pas besoin d'indenter les case dans les switch [L1] graphic.cc:72, gui.cc:84	Le code est bien écrit et globalement assez clair. Essayez de décomposer un peu plus la fonction lecture pour la rendre plus lisible. Bon travail!	20.50
357156	1.50	[A3]constante in shape	2.00		5.00		Bon travail, continuez comme ça	20.75
357228	2.00	ok	2.00	ok	5.00	The default attribut values should be placed in the constructors.	You seems on a really good pace but you have to correct the save part. The style of coding is really good, keep it for report 3.	19.00
357239	2.00	ok	2.00	ok	4.50	[L22] mauvais réalignement: gui.cc: l388-->390+l394+l401-402; Warning: shape.h: l65 - l73: commentaires dépassent 87 caractères	Bon code, tous les tests passent, bon travail!	21.50
357419	2.00	ok	2.00	ok	5.00	Attention à l'utilisation d'indentation et de {} abusifs avec quelques case	Bon projet, continuez ainsi.	21.50
357437	1.25	[A4] gui inclu constantes.h [A5] dependance gtkmm dans graphic.h, il manque graphic_gui.h	0.00	[C2] simulation.h 23, 26, robot.h Neutraliseur en particulier	3.00	[L1] simulation.cc 26-63, 115-129, 151, 163, 165 gui.cc , graphic_gui etc [L2] simulation.h 26, simulation.cc 46, 60, 115, 229, 240, shape.cc, gui.cc 46	C'est bien mais votre code manque de rigourosité	17.25
357574	2.00	Warning: include inutile de constants.h dans gui.cc; Warning: Il faut utiliser des constructeurs au lieu de créer des instances et de les setter avec des setters	2.00	ok	4.00	[L1] inconsistent brace style: simulation.cc l71-->73, l74-->76, l77-->79, l82-->93; indentation error dans gui.cc l42-->58	Eviter d'utiliser des setters sans vérification, les constructeurs sont là pour créer les instances. A part, votre code n'est pas mal.	20.25
357595	2.00	ok	2.00	ok	5.00	top	Impeccable. Excellent travail!	22.00
357650	1.50	[A3]"constantes.h" inclut dans shape.cc	2.00	Ok	4.50	[L2]robot.cc 12,19,35,55,66...	Très bonne exécution, projet bien structuré. Bon travail	20.75

357744	2.00	Top	2.00	Top	4.00	<p>[L2] robot.cc66 [L1]simulation.h: do not indent private/public [P2]simulation.cc: Simulation::decodage_ligne (95+)</p> <p>Suggestions / Warnings 1. Classes should always start with an uppercase letter (neutralizer...)</p>	<p>Although, the project was not completed, the code was very well written and understandable. Continue like this!</p> <p>Suggestions:</p> <ol style="list-style-type: none"> 1. In gui.cc, MyWindow:open_clicked, you can remove the filter for the c++ files 2. In order to have smaller and more performant vectors (robots, particules), use pointers and not instances 3. Do not pass directly the vectors of robots/particules to the Simulation, but only the informations needed through Simulation 4. To simplify the destructuring of the vector of data from the simulation, use structured binding declaration (c++17) 5. You could simplify your Simulation::decodage_ligne by creating for each a robot, a function that analyzes the line and return a pointer to the new created instance 	12.75
357752	2.00	ok	2.00	ok	5.00		<p>Excellent projet, code et rapport très agréables à lire! Beau travail!</p>	22.00
357754	2.00	ok	2.00	ok	5.00		<p>Excellent projet, code très agréable à lire! Beau travail!</p>	22.00

357841	1.50	[A3] shape.cc: inclusion of constantes.h	2.00	[C1]simulation.cc, newRow externally modifies the members of simulation, same for dist_neutra_particule	3.00	[L1]shape.cc81-104, [P2]gui.cc99 (79 lines) ,shape.cc57 (46 lines),simulation.cc22 (76 lines) Suggestions / Warnings: 1. Do not mix PascalCase, snake_case and camelCase together 2. Do not mix english and french	The code is not very readable nor understandable. Try to simplify it and organize it. Suggestions: 1. Remove .o and the executable from the project before uploading it 2 In gui.cc, MyWindow:open_clicked, you can remove the filter for the c++ files 3. Work on your code style and try to find better names 4. Fix the encapsulation errors, by at least passing them as const reference using getters that return const reference and not directly the vector or even better do the for-loops in the Simulation	19.25
357849	2.00	OK	2.00	OK	3.00	[P2]couleur(), gui(), switch_lecture() [L1] gui.cc 39,150,154,219	Really good code, very efficient and nice to read. Be sure to check for consistency for your method brackets. Very good job.	20.00
357857	2.00		2.00		4.00	[P2]simulation.cc superposition_rob_part, decodage_ligne gui.cc: Gui()	Bien mais compteur de mise à jour à corriger	20.00
358190	2.00		2.00		4.00	[P2]simulation.cc superposition_rob_part, decodage_ligne gui.cc: Gui()	Bien mais compteur de mise à jour à corriger	20.00
358257	1.50	[A3]constante in shape	2.00		4.00	[P2]simulation.cc:165	Noms de header guard trompeurs. Bon travail, continuez comme ça	20.50
358300	2.00	OK	0.00	[C1] simulation.cc,e you should have declared it as static [C2] robot.h 92, 98	3.00	[L1]graphic.cc, 9, particule.cc, 10, 17, 19, 26, 29, projet.cc 21 [P2] MyWindow(), Simulation::check_Superposition(), Simulation::draw:planet()	Be sure to check for consistency in your bracket style, and for the indentation on control structure. The execution test passed really well, Continue on this way.	18.00
358305	2.00	OK	2.00	OK	2.00	[L1] simulation.h 20, gui.h 32-43. 46-90, ... [P2] on_file_dialog_response, Window() , lecture(string line), Robot_N()	Be careful with the respect of convention, be sure to fix the open functionality for the Rendu3. In general however, it works well, and don't need much to become very good.	16.25

358312	1.25	[A4] gui inclu constantes.h [A5] il manque graphic_gui et vous graphic inclu gtkmm	2.00	ok	4.00	[L1] simulation.cc surindentation l 111, 113 fct: error_check2 missing indentation 239, 248, shape.cc , robot.h, gui.h (2 espaces) etc...	C'est bien mais il faudrait retravailler l'indentation	19.75
358313	1.50	[A3]shape.cc: mentioning of robot in graphic_cercle [A4]graphic.cc: mentioning of robot in dessin_cercle	1.00	[C1] simulation.cc: desintegration_particules has access to the vector of Simulation and modify it outside it. Simply add it as a method of Simulation Warnings: 1. Do not use too many statics attributes	4.00	[L1] No consistency in bracing, indentation public/private in robot.h Suggestions / Warnings: 1. Inconsistent spacing, use formatting tool 2. Do not mix PascalCase, snake_case and camelCase	The code is not very readable nor understandable. Try to simplify it and organize it. Suggestions: 1. Remove .o, .gch and the executable from the project before uploading it 2. You don't need a project.h 3. In simulation.cc, all methods have access to static vectors, you don't need to pass them as parameters of member functions 4. In simulation.cc and gui.cc, instead having static members (particles and robots), you could simply have a unique instance of simulation in gui (e.g a pointer as attribute of both MyArea and Gui) 5. In robots.cc, you can transform the draw functions into static functions, as such you don't have to have an instance of the class to access them or even better define them for only one particle / robot and do the for loop before in Simulation 6. In simulation.cc, the drawing functions could be members of the class Simulation, so you don't have to create getters to access the information	19.50
358420	2.00	ok	2.00	ok	5.00	ok	Bonne architecture, certains aspects de l'exécution à revoir	20.75

358442	2.00	Top	2.00	Top	5.00	<p>Top</p> <p>Suggestions:</p> <ol style="list-style-type: none"> 1. In the member initializer list, you don't have to change the name of the passed parameters even though they have the same name as the attributes of the class 2. Instead of using always using std, you could use instead using std::what_you_need 	<p>The code is well written and easy to understand, good job! Suggestions: 1. Try to not pass const reference of vectors of the class simulation, even though it doesn't break the encapsulation, it is very easy to do a mistake... at least use getters that return const reference and not directly the vector</p> <p>2. In order to have smaller and more performant vectors (robots, particules), use pointers and not instances</p>	22.00
358456	1.25	[A4] gui inclu constantes.h [A5] graphic inclu constantes.h	2.00	ok	5.00	ok	Bravo, c'est du bon travail	21.25
358476	2.00	OK	1.00	[C2] : particule.h with setCarre and setpart_correct_file	4.00	[L1] gui.cc 369, graphic.cc 16,33,81,	Well done, your project is good. Be sure to look for the issue with the timer, and correct the small mistakes, otherwise good work. Initialize your attribute from the Constructor or with setter is a better practice than just initialize them in the .cc (simulation.cc, robot.cc, particle.cc). Chekout aginst the indentation rule before Rendu3 ;)	18.25
358500	1.25	[A4] gui inclu constantes.h [A5] dependance gtkmm dans graphic.h, il manque graphic_gui.h	0.00	[C2] simulation.h 23, 26, robot.h Neutraliseur en particulier	3.00	[L1] simulation.cc 26-63, 115-129, 151, 163, 165 gui.cc , graphic_gui etc [L2] simulation.h 26, simulation.cc 46, 60, 115, 229, 240, shape.cc, gui.cc 46	C'est bien mais votre code manque de rigourosité	17.25
359179	2.00	ok	2.00	ok	4.00	[P1] simulation.cc:lecture Warning: Pas besoin d'indenter les case dans les switch [L1] graphic.cc:72, gui.cc:84	Le code est bien écrit et globalement assez clair. Essayez de décomposer un peu plus la fonction lecture pour la rendre plus lisible. Bon travail!	20.50
359187	2.00	OK	2.00	OK	5.00	[L1] gui.cc:125,410 (warning) [L2] particule.cc:105 (warning)	Excellent rendu, très bon travail !! Attention aux warnings pour le style, sinon tout est nickel !	22.00
359285	2.00	ok	2.00	ok	5.00	warning - indent for graphic.h l14	Bon travail, bonne architecture.	21.50

359444	0.50	[A2.2] simulation.cc, robots.cc: inclusion of gtkmm and caiomm refptr in the function draw [A3] shape.cc: Carre::draw, Cercle::draw depends on caiomm [A5] graphic.cc: Mentioning of robots	2.00	Top	3.00	[L1] simulation.h, particle.h...: do not indent public, private, protected [L2]shape.cc48, [L2]shape.h21, [P2]simulation.cc41 (97 lines...) Warnings / Suggestions : 1. Don't mix PascalCase and snake_case in the name of the classes 2. Inconsistent spacing: use a formatting tool 3. Inherited virtual methods don't need the keyword virtual before, only override	The code is readable and overall easy to understand, goob job! Suggestions: 1. Do not reference gtkmm/caiomm in the model (simulation, robots..) files, use the functions defined in shape.cc which themselves call other functions in graphics.cc 2. Try to not pass const reference of vectors of the class simulation, even though it doesn't break the encapsulation, it is very easy to do a mistake... at least use getters that return const reference and not directly the vector 3. In order to have smaller and more performant vectors (robots, particules), use pointers and not instances 4. Instead of having a constructor with isstringstream, create a static method that accepts a string / isstringstream and return a pointer on an instance of the class 5. It is prohibited to add methods to structs (in shape.cc, Carre::draw)	18.50
360558	2.00	ok	2.00	ok	4.00	[L2] shape.cc, gui.cc	Très bon projet, attention à vos lignes trop longues, c'est dommage car les lignes que vous avez pensé à ramener à la ligne sont correctement indentées. Beau travail!	21.00
360837	2.00		2.00		5.00			22.00
360931	1.50	[A3]constante in shape	1.00	[C2]rorbot.h et particule.h pas sur une ligne	5.00	[L1]simulation.cc:81	Bon travail, attention à bien lire les consignes et tester le code	17.75
360970	2.00	ok	2.00	ok	5.00	warning [L2] 2 lines trop longues dans robot.cc	Très bon projet malgré quelques petites erreurs d'inattention! Beau travail!	21.25
361004	2.00	ok	2.00	ok	5.00	Quelques lignes trop longues (gui) Attention de même pour la longueur de fonction souvent limitées.	Une très bonne implémentation, vous avez néanmoins la sauvegarde de fichier à revoir. Continuez vos efforts.	21.00

361007	0.50	[A2.1]simulation doit être une classe indépendante, violation de l'architecture en ce qui concerne l'indépendance du modèle[A3]"constant es.h" inclut dans shape.cc [A5]"constantes.h" inclut dans graphic.cc	2.00		4.50	[L1]plusieurs styles d'accolades utilisés de façon aléatoire, particule.cc l17,29... gui.cc l56,75... warning indentation simulation.cc l160	Bon travail, bonne exécution	19.50
361030	2.00	ok	2.00	ok	4.00	[L1]: particule.cc l22-->l25 ; inconsistent brace style particule.cc l18-->l26 + robot.cc l98-->l103	Bon travail. Globalement votre code est bien écrit et bien compréhensible	20.50
361046	2.00	warning - simulation.h inclut 2 fois dans gui.cc	2.00	ok	4.00	[L1]plusieurs styles d'accolades utilisés de façon aléatoire, particule.cc l20,l25... gui.cc l28,46,54...; mauvaise indentation gui.cc l101-127	Bonne exécution, architecture bien implémentée	20.50
361051	1.50	[A5] graphic doit être indépendant et ne peut qu'inclure GTKmm. Mauvais includes: constantes.h	0.00	[C1] Attributs publics: gui.h:Simulation simulation ; shape.h: double longueur,S2d centre,double rayon [C2] robot.h: addNbupdate (Seuls les constructeurs et les getters définis sur la même ligne que le prototype sont autorisés dans l'interface.)	4.00	[L1] fichiers .h: ne pas indenter private, public & protected shape.cc:60,68,69-74,76 [L2] gui.cc:20,398,547 (warning)	Très bon travail. Pour les blocs if/else n'ayant qu'une instruction sur la même ligne, évitez l'utilisation des accolades. Attention aux exigences par rapport aux classes et à l'architecture.	18.50

361066	1.00	[A2.2] simulation importe gtkmm / [A5] graphic dépend de robot / warning [A3] shape importe gtkmm	2.00	ok	5.00		Excellent rapport! Plusieurs erreurs de dépendences, je ne vous ai pas compté celle de shape puisque vous ne faites qu'importer gtk sans l'utiliser, c'est dommage car le code est bon et agréable à lire! Beau travail!	20.50
361116	0.25	[A2.Z]robot.h, particule.h inclut <gtkmm.h>; [A3] shape.cc inclut "robot.h", "particule.h"; [A4]gui.cc inclut "robot.h", "particule.h"; [A5]graphic.cc inclut "constantes.h"	1.00	[C2]interface "robot.h" n'est pas correctement externalisée, il faut définir les constructeurs dans "robot.cc"	3.00	[L1]les mots-clés "public-private" ne sont pas à indenter; [P2]gui.cc l291 maj_myevent dépasse les 80 lignes	Bonne exécution mais architecture et conventions à revoir.	16.75
361124	1.00	[A2.2] simulation importe gtkmm / [A5] graphic dépend de robot / warning [A3] shape importe gtkmm	2.00	ok	5.00		Excellent rapport! Plusieurs erreurs de dépendences, je ne vous ai pas compté celle de shape puisque vous ne faites qu'importer gtk sans l'utiliser, c'est dommage car le code est bon et agréable à lire! Beau travail!	20.50
361133	1.25	[A4] gui inclu constantes.h [A5] il manque graphic_gui et vous graphic inclu gtkmm	2.00	ok	4.00	[L1] simulation.cc 26-63, 115-129, 151, 163, 165 gui.cc , graphic_gui etc [L2] simulation.h 26, simulation.cc 46, 60, 115, 229, 240, shape.cc, gui.cc 46	C'est bien mais il faudrait retravailler l'indentation	19.75
361150	2.00	ok	2.00	ok	4.00	[L1]robot.cc164,167,170; gui.cc:la liste d'initialisation de fenetre doit être alignée/indentée ; à quoi servent les 4 variables numérotées ptr1 etc... dans new_particles()? warning:shape.cc:un nom de variable ne doit pas être 100% en majuscule	bon projet bien organisé mais parfois obscur: ptr1 à ptr4 dans particules.cc; le style est généralement très clair.	21.00
361164	2.00	ok	2.00	ok	5.00		Très bon projet, fonctionnalité complète, beau travail!	22.00

361174	2.00	Top Suggestion: 1. No need to include the entirety of gtkmm in graphics.cc	2.00	Top	4.00	[L2]graphic.cc, robot.cc.... Suggestions / Warnings: 1. Don't mix PascalCase, snake_case and camelCase 2. Inconsistent spacing, use a formatting tool 3. Try to find better names (erreur_superneutralisateur, erreur_superrobotparticule...)	The code is not very readable nor understandable. Try to simplify it and organize it. Suggestions 1. In gui.cc, MyWindow:open_clicked, you can remove the filter for the c++ files 2. In gui.cc, there are some methods (cercle_n, cercle_r...) that are not even used and are already present in graphic.cc 3. In shape.cc, instead of declaring the variable on and off, you could simply use a boolean :) 4. Try to not pass const reference of vectors of the class simulation, even though it doesn't break the encapsulation, it is very easy to do a mistake... at least use getters that return const reference and not directly the vector 5. In order to have smaller and more performant vectors (robots, particules), use pointers and not instances	16.00
361235	2.00	OK	1.00	[C1] e in simulation.h	4.00	[L1] shape.cc 59-67, 74-88, simulation.cc 165-175, GUI.cc 493,500	Good work, clean code, just a few small mistakes.	19.50
361276	2.00	WARNING: gtkmm inclut dans le header de graphic, gtkmm est donc indirectement inclus dans shape	2.00	ok	4.00	[P1] simulation.cc:ecriture Warning: [L2] robot.cc:93	Le code est agréable à lire et bien pensé. Super travail!	21.00

361316	2.00	Top	1.00	[C1]simulation.cc: the method remove, outside the class Simulation, has access to the vector of particles, simply add it as a method of simulation	5.00	Warnings / Suggestions: 1. Don't mix camelCase, snake_case and PascalCase 2. Inconsistent spacing: use a formatting tool	The code is readable and overall more or less easy to understand, goob job! Suggestions: 1. In project.cc, pass the instance of Simulation to gui (same applies for DrawingArea) as either a pointer or const reference, not as instance 2. The method Simulation::reset is pure magic... do not internally change *this... 3. Do not pass around large vectors of classes and continuisly copying them... pass them as const reference and replace the instances with pointers 4. To simplifying the destructuring of the vector of data from the simulation, use structured binding declaration (c++17)	21.00
361362	2.00	ok	2.00	ok	5.00	warning [L2] 2 lines trop longues dans robot.cc	Très bon projet malgré quelques petites erreurs d'inattention! Beau travail!	21.25
361380	2.00	ok	2.00	ok	3.00	[L1] gui.cc:295,350-353,367-369,376; robot.cc:88; simulation.cc:81, ... [P1] simulation.cc:lecture Warning: [L2] robot.cc 94,58; simulation..cc: 171	Globalement le code est bien écrit et lisible. Malheureusement les conventions d'indentations ne sont pas toujours respectées. Aussi vos fonction gui et lecture sont assez grosses, il serait judicieux des les décomposer en sous fonctions (ou sous widgets pour gui).	20.00
361405	1.50	[A5] graphique.cc inclut constantes.h	2.00	ok	3.00	[P1] gui.cc:Gui(), simulation.cc:decodage_ligne Warning: [L1] shape.cc:38-45; simulation.cc:66,77-80; gui.cc:170 [L2] graphique.cc:93 Les cases n'ont pas besoin d'être indentés Essayez d'être plus propre dans votre manière de wrapper les lignes	Globalement le code est bon. Essayez de remplir un peu la classe particule qui pour l'instant ressemble plus à une struct. Sinon bon travail!	18.75
361537	2.00	ok	2.00	ok	4.00	[L2] shape.cc, gui.cc	Très bon projet, attention à vos lignes trop longues, c'est dommage car les lignes que vous avez pensé à ramener à la ligne sont correctement indentées. Beau travail!	21.00

361567	2.00	ok	2.00	ok	3.00	[L1] simulation.cc 273, 328-332, robot.h surindenté, robot.cc 173-174 etc... gui 2 espaces [L2]robot.cc 270 [P2] simulation.cc decodage_ligne >80 lignes	Il y a quelques bug à corriger, courage	14.00
361757	2.00	WARNING: gtkmm inclut dans le header de graphic, gtkmm est donc indirectement inclus dans shape	2.00	ok	4.00	[P1] simulation.cc:ecriture Warning: [L2] robot.cc:93	Le code est agréable à lire et bien pensé. Super travail!	21.00
362005	1.50	[A5] graphic ne doit pas include constantes	2.00	ok	5.00	ok	Attention à votre module graphic qui devrait être indépendant de votre modèle (même dans la nomenclature). Sinon le projet est très bien réalisé.	21.50
362019	2.00	ok	2.00	ok	5.00	Attention à l'utilisation d'indentation et de {} abusifs avec quelques case	Bon projet, continuez ainsi.	21.50
362057	2.00	ok	2.00	ok	5.00	The default attribut values should be placed in the constructors.	You seems on a really good pace but you have to correct the save part. The style of coding is really good, keep it for report 3.	19.00
362176	0.25	[A2.z]robot.h, particule.h inclut <gtkmm.h>; [A3] shape.cc inclut "robot.h", "particule.h"; [A4]gui.cc inclut "robot.h", "particule.h"; [A5]graphic.cc inclut "constantes.h"	1.00	[C2]interface "robot.h" n'est pas correctement externalisée, il faut définir les constructeurs dans "robot.cc"	3.00	[L1]les mots-clés "public-private" ne sont pas à indenter; [P2]gui.cc l291 maj_myevent dépasse les 80 lignes	Bonne exécution mais architecture et conventions à revoir.	16.75
362199	2.00	ok	2.00	ok	4.00	[P1] gui.cc:Gui() Warning: [L1] shape.cc:79,87,95 [L2] gui.cc:57	Super code! Très lisible et clair (sauf pour decodage_ligne et le constructeur de Gui qui mériterait d'être retoucher). Beau travail!	21.00

362201	2.00	OK	0.00	[C1]graphic.cc noir, gris, ... You could have declared them as static, or constexpr	4.00	[L1]gui.cc22-28,360,444,447,particule.cc18,26-30,simulation.cc	Good job in general. Be sure to be consistent with your bracket style, even for control instructions.	18.00
362204	2.00	ok	2.00	ok	5.00	warning - indent for graphic.h l14	Bon travail, bonne architecture.	21.50
362233	1.50	[A5] graphic ne doit pas include constantes	2.00	ok	5.00	ok	Très bon projet, les concepts sont bien compris et utilisés, continuez ainsi.	21.00
362253	2.00	ok	2.00	ok	5.00		Très bon projet, fonctionnalité complète, beau travail!	22.00
362308	2.00	OK	0.00	[C2] robot.h:is_broken ; robot.h:step : Seuls les constructeurs et les getters définis sur la même ligne que le prototype sont autorisés dans l'interface.	3.00	[L1] gui.h: indentation à 2 caractères au lieu de 4 pour le reste du code [L2] constantes.h:4 (warning) [P2] gui.cc:9-101	Très bon travail. Faites néanmoins attention aux définitions de fonctions dans l'interface et au style.	18.00
362320	2.00	ok	2.00	ok	5.00		Excellent travail! N'oubliez pas de faire un make clean avant d'envoyer vos fichiers, le fichier compilé ne tourne pas forcément de la même manière sur tous les ordinateurs	22.00
362358	0.50	[A2.1]simulation doit etre une classe indépendante, violation de l'architecture en ce qui concerne l'indépendance du modèle[A3]"constantes.h" inclut dans shape.cc [A5]"constantes.h" inclut dans graphic.cc	2.00		4.50	[L1]plusieurs styles d'accolades utilisés de façon aléatoire, particule.cc l17,29... gui.cc l56,75... warning indentation simulation.cc l160	Bon travail, bonne exécution	19.50

362379	0.50	[A2.2] simulation.cc, robots.cc: inclusion of gtkmm and caiomm refptr in the function draw [A3] shape.cc: Carre::draw, Cercle::draw depends on caiomm [A5] graphic.cc: Mentioning of robots	2.00	Top	3.00	[L1] simulation.h, particle.h...: do not indent public, private, protected [L2]shape.cc48, [L2]shape.h21, [P2]simulation.cc41 (97 lines...) Warnings / Suggestions : 1. Don't mix PascalCase and snake_case in the name of the classes 2. Inconsistent spacing: use a formatting tool 3. Inherited virtual methods don't need the keyword virtual before, only override	The code is readable and overall easy to understand, goob job! Suggestions: 1. Do not reference gtkmm/caiomm in the model (simulation, robots..) files, use the functions defined in shape.cc which themselves call other functions in graphics.cc 2. Try to not pass const reference of vectors of the class simulation, even though it doesn't break the encapsulation, it is very easy to do a mistake... at least use getters that return const reference and not directly the vector 3. In order to have smaller and more performant vectors (robots, particules), use pointers and not instances 4. Instead of having a constructor with isstringstream, create a static method that accepts a string / isstringstream and return a pointer on an instance of the class 5. It is prohibited to add methods to structs (in shape.cc, Carre::draw)	18.50
362425	1.50	[A3]	0.00	[C1]graphic.cc noir, gris_clair, ... You could have declared them as static, or constexpr	3.00	[L1]simul.h 36, simul.cc 38,64, shape.cc 15, ... [L2]shape.cc 63,67,75, robot.h 54,55	Be careful with the respect of convention. You must really implement the disintegration functionality for the next Rendu. Otherwise, your GUI was good, the opening/saving functionality worked completely, good job for that.	14.00
362429	2.00	ok	2.00	ok	5.00	parfait	Code très bien structuré. Dommage de vous être trompées en lisant la donnée.	21.50
362434	2.00	ok	2.00	ok	3.00	[L1] gui.cc:295,350-353,367-369,376; robot.cc:88; simulation.cc:81, ... [P1] simulation.cc:lecture Warning: [L2] robot.cc 94,58; simulation.cc: 171	Globalement le code est bien écrit et lisible. Malheureusement les conventions d'indentations ne sont pas toujours respectées. Aussi vos fonctions gui et lecture sont assez grosses, il serait judicieux de les décomposer en sous-fonctions (ou sous-widgets pour gui).	20.00

362471	1.50	[A3] shape.cc: inclusion of constantes.h	2.00	[C1]simulation.cc, newRow externally modifies the members of simulation, same for dist_neutra_particule	3.00	[L1]shape.cc81-104, [P2]gui.cc99 (79 lines) ,shape.cc57 (46 lines),simulation.cc22 (76 lines) Suggestions / Warnings: 1. Do not mix PascalCase, snake_case and camelCase together 2. Do not mix english and french	The code is not very readable nor understandable. Try to simplify it and organize it. Suggestions: 1. Remove .o and the executable from the project before uploading it 2 In gui.cc, MyWindow:open_clicked, you can remove the filter for the c++ files 3. Work on your code style and try to find better names 4. Fix the encapsulation errors, by at least passing them as const reference using getters that return const reference and not directly the vector or even better do the for-loops in the Simulation	19.25
362474	2.00	Top Suggestion: 1. No need to include the entirety of gtkmm in graphics.cc	2.00	Top	4.00	[L2]graphic.cc, robot.cc.... Suggestions / Warnings: 1. Don't mix PascalCase, snake_case and camelCase 2. Inconsistent spacing, use a formatting tool 3. Try to find better names (erreur_superneutralisateur, erreur_superrobotparticule...)	The code is not very readable nor understandable. Try to simplify it and organize it. Suggestions 1. In gui.cc, MyWindow:open_clicked, you can remove the filter for the c++ files 2. In gui.cc, there are some methods (cercle_n, cercle_r...) that are not even used and are already present in graphic.cc 3. In shape.cc, instead of declaring the variable on and off, you could simply use a boolean :) 4. Try to not pass const reference of vectors of the class simulation, even though it doesn't break the encapsulation, it is very easy to do a mistake... at least use getters that return const reference and not directly the vector 5. In order to have smaller and more performant vectors (robots, particules), use pointers and not instances	16.00
362542	2.00	ok	2.00	ok	5.00	ok	Bonne architecture et style, éxécution ok. Bon travail	19.50

362549	1.50	[A5]	2.00		5.00		Bien mais problème de lecture de fichier (particule)	21.50
							<p>Although, the project was not completed, the code was very well written and understandable. Continue like this!</p> <p>Suggestions:</p> <ol style="list-style-type: none"> In gui.cc, MyWindow:open_clicked, you can remove the filter for the c++ files In order to have smaller and more performant vectors (robots, particules), use pointers and not instances Do not pass directly the vectors of robots/particles to the Simulation, but only the informations needed through Simulation To simplify the destructuring of the vector of data from the simulation, use structured binding declaration (c++17) You could simplify your Simulation::decodage_ligne by creating for each a robot, a function that analyzes the line and return a pointer to the new created instance 	
362582	2.00	Top	2.00	Top	4.00	<p>[L2] robot.cc66 [L1]simulation.h: do not indent private/public [P2]simulation.cc: Simulation::decodage_ligne (95+)</p> <p>Suggestions / Warnings</p> <ol style="list-style-type: none"> Classes should always start with an uppercase letter (neutralizer...) 	12.75	
362633	2.00	ok	2.00	ok	5.00	<p>Quelsques lignes trop longues (gui) Attention de même pour la longueur de fonction souvent limites.</p>	<p>Une très bonne implémentation, vous avez néanmoins la sauvegarde de fichier à revoir. Contnuez vos efforts.</p>	21.00
362672	2.00	OK	1.00	[C1] e in simulation.h	4.00	<p>[L1] shape.cc 59-67, 74-88, simulation.cc 165-175, GUI.cc 493,500</p>	<p>Good work, clean code, just a few small mistakes.</p>	19.50
362713	2.00	ok	2.00	ok	3.00	<p>[L1] simulation.cc 273, 328-332, robot.h surindenté, robot.cc 173-174 etc... gui 2 espaces [L2]robot.cc 270 [P2] simulation.cc decodage_ligne >80 lignes</p>	<p>Il y a quelques bug à corriger, courage</p>	14.00

362819	2.00	ok	2.00	ok	5.00	Warning: [L1] robot.cc 70-77, 110-116 Warning: [L2] simulation: 56, 55, 60	Super code! Quand vous avez une fonction trop proche des 87 caractères par ligne, c'est plus lisible de l'écrire de cette façon: <pre> fonction(param1, param2,); Sinon Bravo!</pre>	22.00
362863	1.50	[A5] graphic ne doit pas include constantes	2.00	ok	5.00	ok	Attention à votre module graphic qui devrait être indépendant de votre modèle (même dans la nomenclature). Sinon le projet est très bien réalisé.	21.50
362883	1.50	[A3]"constantes.h" inclut dans shape.cc	2.00	Ok	4.50	[L2]robot.cc 12,19,35,55,66...	Très bonne exécution, projet bien structuré. Bon travail	20.75
362932	2.00	ok	2.00	ok	5.00		Excellent projet, code et rapport très agréables à lire! Beau travail!	22.00
362936	1.75	[A4] gui inclu constantes.h [A5] gtkmm dans graphic.h (il manque graphic_gui.h)	2.00	ok	5.00	[L1] robot.cc 276	C'est bien bravo	21.75
362937	2.00	ok	2.00	ok	5.00	warning [L2]gui.cc120	style exemplaire à part un magic-number dans shape.cc 53, .h43. Excellent travail	22.00
363073	2.00		2.00		5.00		Très bien	22.00
363089	1.25	[A4]connection gui-modèle doit se faire uniquement à travers simulation, ne pas inclure robot.h, particule.h dans gui.cc [A5]constantes.h inclus dans graphic	2.00	ok	5.00	ok	Bonne implémentation en général, certains aspects sont à revoir. Faire attention à l'architecture du projet.	20.25

363258	2.00	OK	0.00	[C2] simulation.h:setErreur ; robot.h:setnbU ; gui.h:setD Seuls les constructeurs et les getters définis sur la même ligne que le prototype sont autorisés dans l'interface.	5.00	[L1] simulation.cc:107 ; graphic.cc:62 ; gui.cc:53 (warning) [L2] gui.h:13 (warning)	Très beau travail, bravo ! Faites simplement attention au style, et surtout à la définition de fonctions dans l'interface.	20.00
363301	2.00	warning - simulation.h inclut 2 fois dans gui.cc	2.00	ok	4.00	[L1]plusieurs styles d'accolades utilisés de façon aléatoire, particule.cc l20,l25... gui.cc l28,46,54...; mauvaise indentation gui.cc l101-127	Bonne exécution, architecture bien implémentée	20.50
363358	2.00	Top	2.00	Top	5.00	Top Suggestions: 1. In the member initializer list, you don't have to change the name of the passed parameters even though they have the same name as the attributes of the class 2. Instead of using always using std, you could use instead using std::what_you_need	The code is well written and easy to understand, good job! Suggestions: 1. Try to not pass const reference of vectors of the class simulation, even though it doesn't break the encapsulation, it is very easy to do a mistake... at least use getters that return const reference and not directly the vector 2. In order to have smaller and more performant vectors (robots, particules), use pointers and not instances	22.00

363444	1.50	[A3] shape.cc: Mentioning of robots, functions named after robots	2.00	Top	5.00	Warnings / Suggestions: 1. Don't mix camelCase, snake_case	The code is readable and overall easy to understand, goob job! Suggestions: 1. In project.cc, strange declaration of a pointer of class Simulation, simply do Simulation* simu 2. In simulation.cc, the method draw_call should be a method of Simulation, so you don't have to do getters for the vectors of particles, robots... 3. Same applies for robots.cc and particle.cc 4. In simulation.cc, the decoding functions for the robots and particles should be delegated to those modules as it leads to a verbose method 5. Do not pass around large vectors of classes and continuisly copying them... pass them as const reference and replace the instances with pointers 6. The random engine should be resetted after each reading	20.50
363599	2.00	OK	2.00	OK	5.00	[L1] gui.cc:125,410 (warning) [L2] particule.cc:105 (warning)	Excellent rendu, très bon travail !! Attention aux warnings pour le style, sinon tout est nickel !	22.00
363600	2.00	ok	2.00	ok	5.00	warning [L2]gui.cc120	style exemplaire à part un magic-number dans shape.cc 53, .h43. Excellent travail	22.00

363687	1.25	[A4] gui.cc: direct inclusion of graphic.h (without passing by shape.cc) [A5] graphic.cc: mentioning of robots and particles	2.00		4.00	[P2]simulation.cc61 (102 lines), gui.cc69 (40+), gui.cc194 (40+)	The code is readable and easy to understand, good job! 1. In gui.cc, MyWindow:open_clicked, you can remove the filter for the c++ files 2. In gui.cc MyDrawingArea::on_draw, you can simply add a method to Simulation so you don't have to do getters for the vectors particles and robots 3. In order to have smaller and more performant vectors (robots, particles), use pointers and not instances 4. Your shape and graphic module should be as independent as possible, do not mention robots, particles... Besides, Simulation should call methods in Shape who themselves call methods of graphic, not directly GUI to graphic	20.25
363718	2.00	ok	2.00	ok	5.00	ok	Bonne architecture et style, exécution ok. Bon travail	19.50
363765	1.50	[A5]: graphic.hh ne doit pas inclure shape.h	2.00	ok	2.00	[L2] shape.cc : l40-61-62-74..., simulation.cc : l49,55..., robot.cc, robot.h.. ; [L1]: robot.h: l16-19- 30-38 , you must not indent private and public ; [P2]: simulation.cc: decodage_ligne l44-->l194	Votre code est très difficile à lire car il n'y a presque aucun respect des conventions. Faites attention à cet aspect pour le prochain rendu. C'est dommage car vous avez la majorité des tests qui marchent bien	18.00
363799	2.00		2.00		4.00	[L1] particule.cc:96; robot.cc:192; shape.cc:134-137; simulation.cc:144- 154,158-166,128-131,... Warning: Pas besoin d'indenter vos cases, cela rend le code moins lisible	Super code! On voit que vous avez fait attention à rendre le code lisible. Faites tout de même attention à vos indentions.	20.50
363802	2.00	Warning: include interdit de shape.h dans graphic.h (pas de perte de point car inutile)	2.00	ok	5.00	top	Impeccable. Votre code est très bien écrit, vous respectez bien l'architecture	22.00

363823	1.25	[A4] gui.cc: direct inclusion of graphic.h (without passing by shape.cc) [A5] graphic.cc: mentioning of robots and particles	2.00		4.00	[P2]simulation.cc61 (102 lines), gui.cc69 (40+), gui.cc194 (40+)	The code is readable and easy to understand, good job! 1. In gui.cc, MyWindow:open_clicked, you can remove the filter for the c++ files 2. In gui.cc MyDrawingArea::on_draw, you can simply add a method to Simulation so you don't have to do getters for the vectors particles and robots 3. In order to have smaller and more performant vectors (robots, particles), use pointers and not instances 4. Your shape and graphic module should be as independent as possible, do not mention robots, particles... Besides, Simulation should call methods in Shape who themselves call methods of graphic, not directly GUI to graphic	20.25
363889	2.00	Pourquoi constante.cc ??? Pas de magic number	2.00		2.00	[L1]simulation.cc:60,115,133,363,...[L2]gui.cc:138-141,...[P2]Pop_Graphic,	Bon projet, attention à la forme du code	18.50
363966	2.00	ok	2.00	ok	5.00	ok	Bonne architecture, certains aspects de l'exécution à revoir	20.75
363979	2.00	OK	2.00	OK	5.00	OK	Rendu absolument impeccable, excellent travail !! Visez un sans-fautes pour le rendu3 aussi ;)	22.00
364036	2.00	ok	2.00	ok	5.00		Très bon projet, fonctionnalité complète, beau travail!	22.00

364043	1.50	[A3]shape.cc: mentioning of robot in graphic_cercle [A4]graphic.cc: mentioning of robot in dessin_cercle	1.00	[C1] simulation.cc: desintegration_particles has access to the vector of Simulation and modify it outside it. Simply add it as a method of Simulation Warnings: 1. Do not use too many statics attributes	4.00	[L1] No consistency in bracing, indentation public/private in robot.h Suggestions / Warnings: 1. Inconsistent spacing, use formatting tool 2. Do not mix PascalCase, snake_case and camelCase	The code is not very readable nor understandable. Try to simplify it and organize it. Suggestions: 1. Remove .o, .gch and the executable from the project before uploading it 2. You don't need a project.h 3. In simulation.cc, all methods have access to static vectors, you don't need to pass them as parameters of member functions 4. In simulation.cc and gui.cc, instead having static members (particles and robots), you could simply have a unique instance of simulation in gui (e.g a pointer as attribute of both MyArea and Gui) 5. In robots.cc, you can transform the draw functions into static functions, as such you don't have to have an instance of the class to access them or even better define them for only one particle / robot and do the for loop before in Simulation 6. In simulation.cc, the drawing functions could be members of the class Simulation, so you don't have to create getters to access the information	19.50
364060	1.00	[A3] shape ne doit pas inclure message [A5] graphic ne doit pas inclure constantes	2.00	ok	5.00	Quelques erreurs d'indentation.	Beaucoup de fonctionnalités qui ne fonctionnent pas ou pas comme attendu, il faut que vous modifiez votre code en conséquence avant le rendu 3. Bon courage.	14.75
364120	2.00	OK	2.00	OK	3.00	[P2]couleur(), gui(), switch_lecture() [L1] gui.cc 39,150,154,219	Really good code, very efficient and nice to read. Be sure to check for consistency for your method brackets. Very good job.	20.00
364132	2.00	Pourquoi constante.cc ??? Pas de magic number	2.00		2.00	[L1]simulation.cc:60,115,133,363,...[L2]gui.cc:138-141,...[P2]Pop_Graphic,	Bon projet, attention à la forme du code	18.50

364150	2.00	OK	0.00	[C2] simulation.h:setErreur ; robot.h:setnbU ; gui.h:setD Seuls les constructeurs et les getters définis sur la même ligne que le prototype sont autorisés dans l'interface.	5.00	[L1] simulation.cc:107 ; graphic.cc:62 ; gui.cc:53 (warning) [L2] gui.h:13 (warning)	Très beau travail, bravo ! Faites simplement attention au style, et surtout à la définition de fonctions dans l'interface.	20.00
364154	2.00	ok	2.00	ok	5.00	Une meilleure mise en page du code serait agréable	Un projet qui fonctionne bien mais qui pourrait encore être meilleur avec un peu plus de rigueur.	20.50
364156	1.00	[A3] constantes.h dans shape [A5] pareildans graphic	2.00	ok	5.00	Attention à quelques doubles indentations	L'écriture du code est très bon et l'exécution comme attendu, continuer ainsi.	20.00
364183	1.50	[A5] graphic ne doit pas include constantes	2.00	ok	4.00	[L1] simulation.cc:19-42;robot.cc:13	Quelques points à revoir avant le rendu 3, encore quelques efforts !	18.50