Numéro Sciper	PARTNER SCIPER	Style (4pts max)	Violation list: [criteria]lines	style violation comments	Global comment on rendu1	TOTAL POINTS (11pts)	type of problem	Pénalité rendu1
270354	302076	4		I suggest you leave a bit of air in your code	In general, a good code. The main issues are with it's compactness which makes it not very pleasant to read, but what really penalised you were small mistakes such as the constructors lists in the .h	6,5	archive file is damaged	1
271829	296759	3	[P2] readfile, ERROR_COLL_BALL_OBSTACLE, ERROR_COLL_OBST_PLAYER	Les noms des variables doivent être beaucoup plus clairs (c1, c2, c3 ?), le code est très obscure sinon. Les noms de fonction de devraient pas être en all caps, c'est réservé aux macros. Attention aussi à toujours indenter du même nombre d'espaces/tabs dans vos blocks (e.g. Ball.cc) C'est du code de bonne qualité dans l'ensemble.	La fonction errorSearch pourrait être omise de l'interface et appelée directement depuis reafile. Vous avez beaucoup de duplication de code dans simulation.cc pour les vérifications de collisions. Vous pourriez vérifier les collisions entre cercles et/ou rectangles directement, au lieu de player_ball, player_player, ball_ball etc. Attention aussi à variables globales! Par ailleurs, les variables statics sont autorisées, mais attention à ne pas en abuser. Si vous en avez trop (simulation.cc), c'est peut-être qu'il y a une meilleure manière de faire. Pas besoin d'import à la fois dans le .h et le .cc; un seul des deux est suffisant. Vous n'avez pas utilisé le module tools! Par exemple, Point dans map.h serait utile à bien d'autres endroits. Similairement, Ball et Player sont tous les deux un rayon et une position.	8		

273712	301881	4		Dans l'ensemble, le code est propre et le style est bon. Toutefois, j'ai quelques doutes quant à la façon dont vous avez choisi d'implémenter tool : qu'est-ce que Const_data ? Normalement, dans tool, privilégiez de créer des classes Circle, Square etc pour performer des opérations géométriques. Ce n'est également pas le rôle de tool d'effectuer la lecture des données	Dans l'ensemble le projet est bien réalisé et on note des efforts pour le style. Toutefois, faites plus attention aux divers modules et à la façon dont vous les utilisez par rapport à ce pourquoi ils sont sensés être conçus.	11		
275336	289067	4		Good use of symbolic constants! Very clear formatting.	Nothing to declare. ✈	11		
276861	301291	4		Un code propre et agréable à lire ! Attention à bien aligner vos expressions pour le rendre encore plus lisible (tools.cc:44 par exemple) et d'éviter les "poor choice of symbols" (obj_1_id, obj_2_id ce n'est pas très explicite).	Très bon rendu 1 ! Continuez ainsi et vous ne devriez pas avoir de problème pour le rendu 2.	10	execution full manual check	1
279202	288530	4		Parfait et très lisible	Vous avez fourni un excellent code et facilement lisible ! Bravo !	11		

282211	288820	3	[L1]simation.cc, 22 [L1]simation.cc, 30 [L1]map.cc, 41-52 [L1]map.cc, 17-39 [L1]ball.cc, 14-25 [L1]ball.cc, 28-31	Nice work	Please note that we don’t indent the public/private keywords in class declaration. overall very good job. Keep on the high spirit	10		
282287	283419	3	[P2] main, collision_o_j, collision_o_b	N'hésitez pas à faire des noms plus longs mais compréhensibles (e.g. test_obst_duplication, pas O_J_1 ?, a, b, c, d, ...). Duplication de code dans map.cpp. N'oubliez pas les règles de nomage: pas de majuscule, snake_case, camelCase, etc	Ne faites jamais d'include d'un .cpp! L'utilisation d'un getValeurs global est une mauvais idée : complique incroyablement l'extraction des informations, et représente tout en virgule flottante.	4		
282360	289256	4		Pensez à bien aérer le code avec des espaces entre les opérateurs et les virgules pour aider le lecteur. Choisir des noms plus clairs (e.g. int count, counter; getPosition devrait être getRectangle)	Du bon code dans l'ensemble. Attention à ne pas créer de modules suferflus. Attention également à n'utiliser les define que quand nécessaire.	10,5		
282409	284529	4	[L1] sim.cc 237 sim.h 12 - 17 tools.cc 53	The code often looks too dense (e.g. sim 225-250: unbreathable!)	It all seems like good work but you could easily enhance the code legibility. It would all look much nicer.	11		
282416	287609	4		ok	projet assez clair mais ne pas donner un nom d'une seule lettre majuscule à des paramètres (player.cc, ball.cc, tools.cc), mettre les fonctions non-exportées en static (ex: dans simulation.cc). Pas nécessaire de remplacer 0.5 par MOITIE car ça n'est pas un paramètre.	11		

282435	286955	2	[L1] : Map.cc: 13-43, Player.cc: 16-23, Player.cc: 32-65, Player.cc: 80-89 Obstacle.cc: 14-40 [P5] : Simulation.cc: 150-167, ball.h: 35, ball.h: 38, obstacle.h: 42 , player.h: 44	Vous devez faire plus attention à votre indentation. Vous laissez des doubles indentations dans Map, Obstacle, Player, Tools, et un oubli d'indentation dans Player. Par ailleurs de nombreux endroit dans votre code sont difficilement lisible : des espaces sont oubliés, d'autres répétés... Il est difficile de relire votre code tant celui-ci est condensé, par exemple la fonction cerc_obstacle() de simulation est bien trop compacte !	Votre code est bon, mais malheureusement bien trop obscur pour un étranger. A vrai dire on dirait que votre rendu final a été fait dans la précipitation, sans nettoyer votre code. Pour le rendu 2, essayez de donner des noms de fonction et de variable plus explicites, et de reprendre includes et indentation. Que les includes ne se trouvent que dans le .h. Que les accolades et les espaces soient disposés selon les mêmes règles de partout (si vous mettez un espace avant et après ==, cela doit être dans tout le code !) Evitez à tout prix des doubles espaces, ou l'absence totale d'espaces. Et surtout évitez une fonction comme cerc_obstacle() qui est affreusement complexe à lire.	8		
282515	288166	3	[P5] projet.cc:15,17 ; map.cc:NEAR_CORNER()	Le code est propre et bien indenté ! Bravo. Dommage pour l'étourderie sur les Magic Numbers. Un enum ARG_MODE, ARG_FICHIER et un enum TOP_LEFT, TOP_RIGHT etc... remplaceraient parfaitement les magic numbers des arguements de main ou ceux des quadrants.	Dans l'ensemble c'est un très bon code ! Continuez ainsi et corrigez ces petites erreurs et vous aurez une excellente note au rendu 2 !	9		
282565	284284	4		Style parfaitement clair. L'utilisation d'un switch avec un seul cas (projet.cc) n'est pas forcément très judicieux. Attention aux noms de variables peu précis (ex : listeB)	Bravo, les criteres du rendu 1 sont parfaits remplis.	10	Retard 5 jours	1
282657	287800	4	ok	Warnings : You use in many places names of variables that are a bit unclear to a stranger reading the code (comptO, A, B, B1). Also you mix a lot camelCase and snake_case convetion. You should choose one and stick to it. Finally you have sometimes huge amount of conditions in your if-statement. It might be good to	Good architecture and encapsulation ok. Be care with your style : adding some space could make it more agreable to read for external reader and yourself.	11		

282708	288602	4	warning [L1] obstacle.h 17, 20	Avoiding the use of braces on "one-line ifs" (e.g. ball.cc 56) may arguably be better.	Your code is overall very well presented and is pleasing to browse through. That said, the density of ball.cc still leaves room for improvement... Don't get discouraged by the loss of one point, it is definitely not the most meaningful thing regarding your work. Just try to pay more attention to the rules (not funny, I know) for the next steps, and it will be okay!	10		
282756	286734	1	[L1]simulation.cc 256; ball.cc 35,37; player.cc 57 [P2]lecture in simulation.cc [P5]2)player.h 21; simulation.cc 24,26,27; simulation.h 41,44,46	[L1] You should be more coherent with the style of your indentation. In addition that you don't use the same convention for your for-loop and your if-loop, sometimes you mix these two styles. [P2] Your function lecture in simulation.cc is way too long (130 lines). Even if we see that you tried to make it well structured, you should break it or simplify it. [P5]2) In all the lines mentionned, you use too poor name of variables (i, a, b, NB1, NBB, NB2, NBR) which say nothing to an outside reader. You should try to find more expressive name or at least put comments to explain what are they.	We see that you put real efforts in this project. Try to be more rigorous with the style and add some comments in header files and it will be a great improve to your project.	7	execution full manual check	1
282836	302583	3	[P5]2	It is good to define some macro for constant, but here you use too poor names in many place to define them (for example MOITIE, OPPOSE, DOUBLE in tools.h	We see the effort put in the project. Try to fix the mentionned problems and add some comments in the header file if you have time and it will be great !	9		
282844	288453	3	[L1] projet.cc 17 + projet.cc 11,16 ; maps.cc 11,20	Inconsistent indentation of curly brace, mainly on the next line, but sometimes on the same line. Warning : some of your variable names are very limit (see decodage_ligne) and in maps.cc it might be not clear what is the difference when you use i,j or l,c for indexing. If it is the same, use the same names.	Good project in overall. Be more consistent in the style and keep up the good work for the next submit.	10		

282988	284213	4	[L1]ball.cc:53,	Assez bonne présentation. Il y a quelques irrégularités dans la présentation mais presque aucun manquement aux conventions de programmation.	Le code est facile à lire mais pensez à abstraire en boucles et fonctions plutôt que de mettre des longues suites de conditions.	10		
283111	284430	4	ok	style très clair, rien à redire	les modules player, ball et map ont peu de responsabilité. L'essentiel du travail sera dans simulation.	11		
283183	296952	2	[P5]simulation.cc,155 [L1]ball.cc,7-28 [L1]map.cc,6-103 [L1]obstracle.cc,7-22 [L1]player.cc,6-31	- You should give meaningful names to: comp1, comp2, comp3, comp4, comp5, comp6, comp7, comp8, comp9 - You tabbed twice when defining the methods of the classes	You should be careful with the indentation and the naming, but overall very nice job. Keep on the good work!	9		
283196	284316	4		Code très clair, bravo. Les noms des variables dans l'enum de projet.cc pourraient être plus explicites : exemple, MODE ou FICHIER	Votre projet respecte bien les critères fixés pour le premier rendu. Prenez bien en compte la remarque sur les structures de données pour les rendus ultérieurs, ces vecteurs doivent pouvoir être utilisables dans d'autres fonctions de la classe simulation.	10	Retard de 4jours	1
283391	289122	3	[P5]2) simulation.cc	Using an enum which contains TRUE, FALSE and DOUBLE is a bad practice. As you are using bool type, you can already return "true" or "false". Also, you should put clearer name for your variable compare to y2, x2, a, b (balls.h line 30 for example).	Good project. It would be even nicer if you add some comments in the header files when you use some parameters where the name are not clear.	10		

283395	288243	4	[L1]simulation.cc:81	Très bon travail dans l'ensemble ! Dans simulation.cc:185-194, il pourrait être plus clair d'aligner les éléments les uns avec les autres (cf L25). Attention également à rester consistant dans le style d'indentation, c'est à dire garder le même style dans tout le projet, notamment sur les instructions contrôlées d'une seule lignes.	Un travail très sérieux de manière générale ! Je vous recommande toutefois de revor votre architecture car certains modules sont superflus.	9,5		
283419	282287	3	[P2] main, collision_o_j, collision_o_b	N'hésitez pas à faire des noms plus longs mais compréhensibles (e.g. test_obst_duplication, pas O_J_1 ?, a, b, c, d, ...). Duplication de code dans map.cpp. N'oubliez pas les règles de nomage: pas de majuscule, snake_case, camelCase, etc	Ne faites jamais d'include d'un .cpp! L'utilisation d'un getValeurs global est une mauvais idée : complique incroyablement l'extraction des informations, et représente tout en virgule flottante.	4		

283442	287811	4		Dans l'ensemble une très grande attention portée sur le style, c'est très bien. Faites toutefois attention à rester consistants dans l'ensemble de votre code : il y a des différences entre vos switch de simulation et map, ou encore des styles changeants pour les structures de contrôle ne controlant qu'une seule instruction, par exemple entre tools.cc:14 et projet.cc:24	Très bon rendu, on se rend compte que vous avez mis des efforts pour respecter les conventions et garder votre code propre. Continuez comme ça !	9,5		
283470	286317	4	ok	Warnings: quelque erreurs d'indentation dans ball.cc aux lignes 25, 30, 31. Peut-être mettre des noms de méthodes plus complets (e.g collisionCC -> collisionCercleCercle).	Quelques petits points d'architecture et de style à améliorer. Sinon travail impeccable !	11		
283616	301494	3	[L1] moduleLec.cc in function traitement	The use of curly bracket inside case is not the convention and you use not rigourously	Good code. If you find time try to add some short comment in the header file to make it easier to read.	10		

283652	303039	3	[L1] simulation.cc 265, 271, in player.h, in map.h	Many places where the indentation are neither respected nor coherent with the rest of the code	Good code in overall. In some place your code could be more succint, for example "if (...) return true, else return false" could be simplified in "return (...)". However after correcting the few comments it will already be great.	8,5		
283670	286570	4	[L2] joueur.h 17 [P5] sim 12 (seriously...) warning [P5] proj 17 sim 56	simulation_lecture_fichier is too long!	Your code seems well organized, regarding both architecture and appearance :)	9	delayed	1
283792	287054	4		Merci pour les commentaires/documentation des fonctions!! :) Le code est parfaitement présenté, il n'y a absolument rien à redire.	La représentation des "execution_parameters" est très ingénieuse! Dans le module simulation, le reader ne devrait pas être dans l'interface (.h) puisqu'il n'est utilisé que par Simulation. Reader est en fait assez inutile, tout ce qu'il fait pourrait être dans Simulation directement. La lecture du fichier est extremement complète, c'est impressionnant! Le module Tools est incroyablement complet aussi, bravo!	11		

283796	286386	3	[P5] simulation.cc: 23, player.h: 13, map.h: 20, ball.h: 20, map.cc: 82	Bon respect de l'implémentation, attention toutefois aux noms de vos entités. La classe représentant un Obstacle s'appelle DonneesO, alors que "Obstacle" est bien plus parlant. La variable représentant l'état de votre automate de lecture s'appelle "remplir", "etat" serait plus parlant. Pour finir, évitez les variables d'une lettre, sauf lorsqu'elles sont utilisées ailleurs que dans des index de boucle.	Bon travail, code clair et bien écrit à quelques détails prêt. Attention à bien relire votre code afin d'éviter des erreurs comme l'include de player.h oublié dans projet.cc. Nommez vos variables avec attention en essayant de représenter ce qu'elle contienne. Autre petits conseils afin de rendre un code vraiment parfait : traiter tous les cas de la ligne de commande afin d'éviter que votre programme plante si aucun argument n'est donné ou si trop d'arguments sont données. Ne mettez pas d'include dans les .cc, les includes doivent être dans le .h. Dans le .cc, on ne met que l'include du .h correspondant. Pour finir, évitez de positionner vos #define n'importe où dans le code : vous avez le droit de modifier define.h	9		
284024	289159	3	[L1] ball.cc 31, 49 player.cc 31, 33, 60, ...	Overall, the style is good, but you did a few distraction mistakes on the indentation	Overall, a good work, but unfortunately some errors in the encapsulation heavily penalised you...	6		
284194	287056	3	[L1] simulation.cc 123, 126, 164, 184, ...	Very good variables and functions naming, but consider commenting function prototypes. Look at the coding conventions for how to properly align the call to a fonction on several lines.	Overall, a very good job for now.	10		

284196	299245	4		Soyez peut-être un peu plus précis dans vos choix de noms d'arguments de fonctions ou de variables (param1, param2). À part ça, le code est très bien présenté!	Un projet.h n'est pas nécessaire puisque ce n'est pas un module utilisé ailleurs. Vos collisions ball-ball ou player-player sont identiques, vous pourriez modulariser ça en vérifiant la collision entre deux cercles directement. Enfin, vous pourriez peut-être déléguer un peu de travail à vos classes Ball, Player, Map, qui sont pour l'instant presque inutiles. Ça permettrait d'alléger un peu Simulation qui est pour l'instant très chargé.	11		
284213	282988	4	[L1]ball.cc:53,	Assez bonne présentation. Il y a quelques irrégularités dans la présentation mais presque aucun manquement aux conventions de programmation.	Le code est facile à lire mais pensez à abstraire en boucles et fonctions plutôt que de mettre des longues suites de conditions.	10		
284284	282565	4		Style parfaitement clair. L'utilisation d'un switch avec un seul cas (projet.cc) n'est pas forcément très judicieux. Attention aux noms de variables peu précis (ex : listeB)	Bravo, les criteres du rendu 1 sont parfaits remplis.	10	Retard 5 jours	1
284316	283196	4		Code très clair, bravo. Les noms des variables dans l'enum de projet.cc pourraient être plus explicites : exemple, MODE ou FICHIER	Votre projet respecte bien les critères fixés pour le premier rendu. Prenez bien en compte la remarque sur les structures de données pour les rendus ultérieurs, ces vecteurs doivent pouvoir être utilisables dans d'autres fonctions de la classe simulation.	10	Retard de 4 jours	1

284402	289234	3	[L2] sim 27 tools 25, 66, 102 warning [P5] sim 29 proj 16	INIT_NULLE is not necessary. It does not clarify anything.	Note: default arguments are usually indicated in a prototype (cf. tools) The code is well designed and very clear.	10		
284430	283111	4	ok	style très clair, rien à redire	les modules player, ball et map ont peu de responsabilité. L'essentiel du travail sera dans simulation.	11		
284529	282409	4				11		
284564	289285	4		Perfect	Overall, good job.	11		

284867	296833	3	[L1]ball.cc:11-20,player.cc:9-17,tools.cc:7-57	Bien mais soyez plus consistant et ajoutez plus de retour à la ligne	Votre code est correct mais "aérez-le" avec des retours à la ligne.	9	files doesn't compile	1
284888	286950	4	[L1] tools.cc 41, 45 [L2] player.cc 29	The L1 and L2 are just warnings	Overall, a good work, but unfortunately some errors in the encapsulation heavily penalised you...	6		
284896	301437	3	[L1] test.cc: 77, tool.c: 13, 14, 39, 42, 45, 48, lecture.cc: 69	Beaucoup de petites incohérences dans l'indentation. Pas de retour à la ligne pour une accolade fermante, des instructions doublement indentées, etc...	Très bon code, beau travail. Pourrait être parfait si vous aviez fait un petit clean avant de rendre votre code, pour corriger notamment l'indentation parfois imprécise. Entre autre : ça serait une bonne chose de tester le cas où le nombre d'argument n'est pas bon afin d'afficher un message à l'utilisateur pour corriger son erreur. Autre défaut : les includes doivent aller dans le .h et non le .cc ! Le seul include qui peut rester dans un .cc est l'include du .h correspondant.	10		

285416	289276	2	[L1] simulation.cc [P5]2)	Many errors of indentation (curly bracket placed after a tab, no tab after a new indentation). Also use of many magic numbers for example in lecture.cc (the number 4 that you use many time should have a meaning).	Overall good project, try to correct the few mistakes and let some space between the lines you write and it will be good !	7		
286317	283470	4	Ok.	Warnings: quelque erreurs d'indentation dans ball.cc aux lignes 25, 30, 31. Peut-être mettre des noms de méthodes plus complets (e.g collisionCC -> collisionCercleCercle).	Quelques petits points d'architecture et de style à améliorer. Sinon travail impeccable !	11		
286386	283796	3	[P5] simulation.cc: 23, player.h: 13, map.h: 20, ball.h: 20, map.cc: 82	Bon respect de l'implémentation, attention toutefois aux noms de vos entités. La classe représentant un Obstacle s'appelle DonneesO, alors que "Obstacle" est bien plus parlant. La variable représentant l'état de votre automate de lecture s'appelle "remplir", "etat" serait plus parlant. Pour finir, évitez les variables d'une lettre, sauf lorsqu'elles sont utilisées ailleurs que dans des index de boucle.	Bon travail, code clair et bien écrit à quelques détails prêt. Attention à bien relire votre code afin d'éviter des erreurs comme l'include de player.h oublié dans projet.cc. Nommez vos variables avec attention en essayant de représenter ce qu'elle contienne. Autre petits conseils afin de rendre un code vraiment parfait : traiter tous les cas de la ligne de commande afin d'éviter que votre programme plante si aucun argument n'est donné ou si trop d'arguments sont données. Ne mettez pas d'include dans les .cc, les includes doivent être dans le .h. Dans le .cc, on ne met que l'include du .h correspondant. Pour finir, évitez de positionner vos #define n'importe où dans le code : vous avez le droit de modifier define.h	9		

286570	283670	4	[L2] joueur.h 17 [P5] sim 12 (seriously...) warning [P5] proj 17 sim 56	simulation_lecture_fichier is too long!	Your code seems well organized, regarding both architecture and appearance :)	9	delayed	1
286641	286721	4		Style correct et travaillé tout du long, c'est bien ! Dans simulation.cc:137-145 ou 175-183, je vous recommande de remplacer cette longue expression logique par une fonction renvoyant une valeur booléenne avec un nom clair, pour permettre de plus facilement comprendre le sens du if, et également pour respecter le principe de réutilisation.	Du très bon travail dans l'ensemble ! Continuez ainsi	11		

286721	286641	4		Style correct et travaillé tout du long, c'est bien ! Dans simulation.cc:137-145 ou 175-183, je vous recommande de remplacer cette longue expression logique par une fonction renvoyant une valeur booléenne avec un nom clair, pour permettre de plus facilement comprendre le sens du if, et également pour respecter le principe de réutilisation.	Du très bon travail dans l'ensemble ! Continuez ainsi	11		

286734	282756	1	[L1]simulation.cc 256; ball.cc 35,37; player.cc 57 [P2]lecture in simulation.cc [P5]2)player.h 21; simulation.cc 24,26,27; simulation.h 41,44,46	[L1] You should be more coherent with the style of your indentation. In addition that you don't use the same convention for your for-loop and your if-loop, sometimes you mix these two styles. [P2] Your function lecture in simulation.cc is way too long (130 lines). Even if we see that you tried to make it well structured, you should break it or simplify it. [P5]2) In all the lines mentionned, you use too poor name of variables (i, a, b, NB1, NBB, NB2, NBR) which say nothing to an outside reader. You should try to find more expressive name or at least put comments to explain what are they. Also as a warning, try to use only one convention between snake_case and camelCase notation.	We see that you put real efforts in this project. Try to be more rigorous with the style and add some comments in header files and it will be a great improve to your project.	8		
286830	289281	4		Très bien, code parfaitement lisible.	Projet très bien réalisé. Attention, votre programme peut planter si un nombre incorrect de paramètres est passé au démarrage, vous pourriez gérer ce problème depuis projet.cc. Ajoutez une description des modules en en-tête.	11		
286950	284888	4	[L1] tools.cc 41, 45 [L2] player.cc 29	The L1 and L2 are just warnings	Overall, a good work, but unfortunately some errors in the encapsulation heavily penalised you...	6		

286955	282435	2	[L1] : Map.cc: 13-43, Player.cc: 16-23, Player.cc: 32-65, Player.cc: 80-89 Obstacle.cc: 14-40 [P5] : Simulation.cc: 150-167, ball.h: 35, ball.h: 38, obstacle.h: 42 , player.h: 44	Vous devez faire plus attention à votre indentation. Vous laissez des doubles indentations dans Map, Obstacle, Player, Tools, et un oubli d'indentation dans Player. Par ailleurs de nombreux endroit dans votre code sont difficilement lisible : des espaces sont oubliés, d'autres répétés... Il est difficile de relire votre code tant celui-ci est condensé, par exemple la fonction cerc_obstacle() de simulation est bien trop compacte !	Votre code est bon, mais malheureusement bien trop obscur pour un étranger. A vrai dire on dirait que votre rendu final a été fait dans la précipitation, sans nettoyer votre code. Pour le rendu 2, essayez de donner des noms de fonction et de variable plus explicites, et de reprendre votre indentation. Que les accolades et les espaces soient disposés selon les mêmes règles de partout (si vous mettez un espace avant et après ==, cela doit être dans tout le code !) Evitez à tout prix des doubles espaces, ou l'absence totale d'espaces. Et surtout évitez une fonction comme cerc_obstacle() qui est affreusement complexe à lire.	8		
286962	301464	4		Bien. Bravo !	Votre code est correct et il est bien présenté mais vous devriez plus correctement structurer la manière dont vous déléguer vos tâches.	10		
287015	296048	2	[L1]player.cc:19-24 ; [L2]projet.cc:38, simulation.cc:131, 191, 200, 221, 229, ... ; [P5]simulation.cc:27	Un style dans l'ensemble très correct et très propre, attention toutefois aux lignes trop longues. Egalement, pour votre automate de lecture, utilisez une enum plutôt que des valeurs numériques	Dans l'ensemble du bon travail, entâché par quelques petites erreurs. Ne vous découragez pas, corrigez-les et allez de l'avant pour le rendu 2 !	8		

287038	297846	3	[L1]ball.h, 13-17 [L1]circle.h, 14-25 [L1]game.h, 17-43 [L1]obstacle.h, 12-20 [L1]player.h, 12-18	Be careful with the indentation of the classes (Please note that we don’t indent the public/private keywords in class declaration)	Overall, good job.	9	6 days 13 hours late	1
287054	283792	4		Merci pour les commentaires/documentation des fonctions!! :) Le code est parfaitement présenté, il n'y a absolument rien à redire.	La représentation des "execution_parameters" est très ingénieuse! Dans le module simulation, le reader ne devrait pas être dans l'interface (.h) puisqu'il n'est utilisé que par Simulation. Reader est en fait assez inutile, tout ce qu'il fait pourrait être dans Simulation directement. La lecture du fichier est extrememement complète, c'est impressionnant! Le module Tools est incroyablement complet aussi, bravo!	11		
287056	284194	3	[L1] simulation.cc 123, 126, 164, 184, ...	Very good variables and functions naming, but consider commenting function prototypes. Look at the coding conventions for how to properly align the call to a fonction on several lines.	Overall, a very good job for now.	10		

287173	301560	4	[L1]simulation.cc:55	On note un très grand soin apporté sur le style qui reste clair et cohérent sur l'ensemble du code : de l'excellent travail !	Un rendu d'une grande qualité, bravo ! Pour le suivant, je vous recommande de vous repencher sur la façon dont vous utilisez le module tool.	11		
287327	287388	4		Perfect!	Well organized, keep on the good job!	11		
287388	287327	4		Perfect!	Well organized, keep on the good job!	11		

287433	300540	3	[L1]player.h,8:18 [L1]map.h,9:13 [L1]ball.h,8:17	Please note that we don’t indent the public/private keywords in class declaration.	Overall, very good job.	10		
287459	302017	4		Parfait ! Le style est bien maitrisé et reste constant tout au long du code !! Warning : dans le main, les nombres dans argv[xx] sont aussi des Magic Numbers. N'hésitez pas à faire un enum ou des define pour ces derniers. Remarque : map.cc:12 est assez confuse comme ligne, ce serait cool de mieux l'aligner ou de l'initialiser dans le corps du constructeur pour que cela soit plus propre, mais c'est un détail.	Une excellente documentation du code ! Continuez ainsi	11		
287480								
287609	282416	4		ok	projet assez clair mais ne pas donner un nom d'une seule lettre majuscule à des paramètres (player.cc, ball.cc, tools.cc), mettre les fonctions non-exportées en static (ex: dans simulation.cc). Pas nécessaire de remplacer 0.5 par MOITIE car ça n'est pas un paramètre.	11		
287725	288407	4	[L1]project.cc, 12-14 [L1]simulation.cc, 32	Looks good	Be careful with the indention. Overall, very good job!	11		

287759	295954	2	[L1]ball.cc:19, map.cc:42, player.cc:40, projet.cc:27, tools.cc:12, tools.cc:32, tools.cc:34-35, ... ; [L2]player.h:28, map.cc:59, map.cc:60, player.cc:11, projet.cc:18, projet.cc:27, tools.cc:48, ... ;	Le style est trop brouillon et pas assez relu, de nombreuses erreurs de conventions subsistent	Le projet, plutôt bon dans l'ensemble, donne l'impression d'avoir toutefois été bâclé au niveau du style. Aussi, faites plus attention à la manière dont vous utilisez tools : qu'est-ce que cette classe Tools ? Il semblerait plus approprié de créer de classes correspondant aux objets géométriques afin de pouvoir les manipuler lors d'opérations mathématiques dans le plan	9		
287800	282657	4	ok	Warnings : You use in many places names of variables that are a bit unclear to a stranger reading the code (comptO, A, B, B1). Also you mix a lot camelCase and snake_case convetion. You should choose one and stick to it. Finally you have sometimes huge amount of conditions in your if-statement. It might be good to create boolean variables with clear names to decompose these conditions.	Good architecture and encapsulation ok. Be care with your style : adding some space could make it more agreable to read for external reader and yourself.	11		

287811	283442	4		Dans l'ensemble une très grande attention portée sur le style, c'est très bien. Faites toutefois attention à rester consistants dans l'ensemble de votre code : il y a des différences entre vos switch de simulation et map, ou encore des styles changeants pour les structures de contrôle ne controlant qu'une seule instruction, par exemple entre tools.cc:14 et projet.cc:24	Très bon rendu, on se rend compte que vous avez mis des efforts pour respecter les conventions et garder votre code propre. Continuez comme ça !	9,5		

288166	282515	3	[P5] projet.cc:15,17 ; map.cc:NEAR_CORNER()	Le code est propre et bien indenté ! Bravo. Dommage pour l'étourderie sur les Magic Numbers. Un enum ARG_MODE, ARG_FICHIER et un enum TOP_LEFT, TOP_RIGHT etc... remplaceraient parfaitement les magic numbers des arguements de main ou ceux des quadrants. Attention aussi à respecter la convention pour les noms de fonctions : les noms en majuscule sont reservés aux constantes. Les fonctions doivent être définies comme maFonction ou ma_fonction plutôt que MAFONCTION.	Dans l'ensemble c'est un très bon code ! Continuez ainsi et corrigez ces petites erreurs et vous aurez une excellente note au rendu 2 !	9		

288243	283395	4	[L1]simulation.cc:81	Très bon travail dans l'ensemble ! Dans simulation.cc:185-194, il pourrait être plus clair d'aligner les éléments les uns avec les autres (cf L25). Attention également à rester consistant dans le style d'indentation, c'est à dire garder le même style dans tout le projet, notamment sur les instructions contrôlées d'une seule lignes.	Un travail très sérieux de manière générale ! Je vous recommande toutefois de revor votre architecture car certains modules sont superflus.	9,5		

288252	303011	3	[P5] player.cc: 55, ball.cc: 61, simulation.cc: 28, 43, 45, 59, 132, 135...	L'indentation pourrait être améliorée, mais elle n'est pas catastrophique. Il y a beaucoup de petites erreurs dans simulation.cc notamment (lignes 32, 33, 71) Warning : Vous utilisez deux styles pour vos accolades, l'un pour les fonctions, classes et boucles, l'autre pour les conditions et les struct. Il faut uniformiser tout cela. Warning : Nommez correctement vos variables ! Français OU anglais, snake-case OU camel-case, et tenez-y vous. Warning : le define TAILLE_LIGNE est inutilisé. Par ailleurs, quitte à rajouter des defines, ajoutez les dans le fichier define.h, ainsi vous êtes sûrs que tout votre code en profite	L'idée globale semble bien comprise, l'architecture est bonne et le tout est cohérent. Il faut toutefois faire plus attention à la propreté du code : les espaces et les accolades sont irréguliers, vous nommez vos variables tantôt en snake-case, tantôt en camel-case, certaines variables commencent par des majuscules... Sans être grave, cela rend votre code bien moins compréhensible. Autre conseil : évitez la déclaration de fonction à côté d'une classe, car dans tous les cas ces fonctions auraient du être des méthodes de classe. Corrigez tout ça pour le prochain rendu et il sera d'une excellente qualité ! Bravo pour le bon travail déjà accompli.	10		

288287	299554	4		Assez bon style, attention toute fois à de légères erreurs de style : vos espaces sont parfois aléatoires, et les directives #include sont censées se trouver dans les .h et non les .cc. On n'inclut dans un .cc que le .h correspondant.	Beau projet, un code clair et lisible qui serait parfait si de très légères erreurs n'étaient pas présentes. Belle inititative sur la classe Entite, bien expliquée, et bonne gestion de l'héritage. Continuez ainsi pour le prochain rendu !	11		
288312	296165	4		Style lisible, à quelqeus détails près : retour à la ligne inutile dans player.cc:22, erreurs d'alignement dans tools.cc:75 par exemple.	Projet très bien réalisé. Vous pourriez ajouter un en-tête à chaque module, décrivant sa fonction.	11		

288407	287725	4	Looks good	[L1]project.cc, 12-14 [L1]simulation.cc, 32	Be careful with the indention. Overall, very good job!	11		
288453	282844	3	[L1] projet.cc 17 + projet.cc 11,16 ; maps.cc 11,20	Inconsistent indentation of curly brace, mainly on the next line, but sometimes on the same line. Warning : some of your variable names are very limit (see decodage_ligne) and in maps.cc it might be not clear what is the difference when you use i,j or l,c for indexing. If it is the same, use the same names.	Good project in overall. Be more consistent in the style and keep up the good work for the next submit.	10		
288530	279202	4		Parfait et très lisible	Vous avez fourni un excellent code et facilement lisible ! Bravo !	11		
288580	288825	4	Ok.	éviter: if(condition) return true else return false, plutôt: return condition. Obstacle : public et private sont inversé par rapport au reste du projet. Pas besoin de redéfinir le destructeur, il ne fait rien par défaut. Si vous avez besoin des indices des élèments dans les boucles faites un "for" normal.	Code simple et concis. Continuez comme ça !	11		

288602	282708	4	warning [L1] obstacle.h 17, 20	Avoiding the use of braces on "one-line ifs" (e.g. ball.cc 56) may arguably be better.	Your code is overall very well presented and is pleasing to browse through. That said, the density of ball.cc still leaves room for improvement... Don't get discouraged by the loss of one point, it is definitely not the most meaningful thing regarding your work. Just try to pay more attention to the rules (not funny, I know) for the next steps, and it will be okay!	10		
288820	282211	3	[L1]simation.cc, 22 [L1]simation.cc, 30 [L1]map.cc, 41-52 [L1]map.cc, 17-39 [L1]ball.cc, 14-25	Nice job!	Please note that we don’t indent the public/private keywords in class declaration. Overall very good job. Keep on the high spirit	10		
288825	288580	4	Ok.	éviter: if(condition) return true else return false, plutôt: return condition. Obstacle : public et private sont inversé par rapport au reste du projet. Pas besoin de redéfinir le destructeur, il ne fait rien par défaut. Si vous avez besoin des indices des élèments dans les boucles faites un "for" normal.	Code simple et concis. Continuez comme ça !	11		
288838	289028	3	[P2] carte.cc : 2 méthodes de plus de 80 lignes	Eviter if(condition == true) utilisez plutôt if(condition). Ne pas indenter les mot clés private, public, protected on comprend moins bien ce qui est accessible par l'utilisateur. Définissez plus de méthodes auxiliaires cela rendra votre code plus lisible.	Quelques points à améliorer quant à la répartition des tâches à travers les différentes classes. La notion d'encapsulation est bien comprise.	6	delayed+ manual checkout	2

289028	288838	3	[P2] carte.cc : 2 méthodes de plus de 80 lignes	Eviter if(condition == true) utilisez plutôt if(condition). Ne pas indenter les mot clés private, public, protected on comprend moins bien ce qui est accessible par l'utilisateur. Définissez plus de méthodes auxiliaires cela rendra votre code plus lisible.	Quelques points à améliorer quant à la répartition des tâches à travers les différentes classes. La notion d'encapsulation est bien comprise.	7	delayed	1
289067	275336	4				11		

289121	296142	3	[L1]simulation.cc:34 ,42,47,50,53 map.cc:41	Le style est bon dans l'ensemble. Attention tout de même à bien respecter les règles d'indentations et ne pas faire de sauts aléatoires entre 2 espaces et 4 espaces. Warning : argv[1], argv[2] etc.. sont des Magic numbers et doivent être définis par des symboles.	Dans l'ensemble c'est un très bon code ! Continuez ainsi et corrigez ces petites erreurs et vous aurez une excellente note au rendu 2 !	10		
289122	283391	3	[P5]2) simulation.cc	Using an enum which contains TRUE, FALSE and DOUBLE is a bad practice. As you are using bool type, you can already return "true" or "false". Also, you should put clearer name for your variable compare to y2, x2, a, b (balls.h line 30 for example).	Good project. It would be even nicer if you add some comments in the header files when you use some parameters where the name are not clear.	10		
289159	284024	3	[L1] ball.cc 31, 49 player.cc 31, 33, 60, ...	Overall, the style is good, but you did a few distraction mistakes on the indentation	Overall, a good work, but unfortunately some errors in the encapsulation heavily penalised you...	6		

289188	295855	4	map.cc 14,15	Warning : using ZERO and TROIS as a define is a very bad practise. I advise you to read the convention file provided on moodle on this subject. Also putting no names for parameters in a header is a bad practice and it is even worse if the code is not commented.	Despite the comments written before, the code is elegant. Keep up the good work !	10,5		
289234	284402	3	[L2] sim 27 tools 25, 66, 102 warning [P5] sim 29 proj 16	INIT_NULLE is not necessary. It does not clarify anything.	Note: default arguments are usually indicated in a prototype (cf. tools) The code is well designed and very clear.	10		
289256	282360	4		Pensez à bien aérer le code avec des espaces entre les opérateurs et les virgules pour aider le lecteur. Choisir des noms plus clairs (e.g. int count, counter; getPosition devrait être getRectangle)	Du bon code dans l'ensemble. Attention à ne pas créer de modules suferflus. Attention également à n'utiliser les define que quand nécessaire.	10,5		

289276	285416	2	[L1] simulation.cc [P5]2)	Many errors of indentation (curly bracket placed after a tab, no tab after a new indentation). Also use of many magic numbers for example in lecture.cc (the number 4 that you use many time should have a meaning).	Overall good project, try to correct the few mistakes and let some space between the lines you write and it will be good !	7		
289281	286830	4		Très bien, code parfaitement lisible.	Projet très bien réalisé. Attention, votre programme peut planter si un nombre incorrect de paramètres est passé au démarrage, vous pourriez gérer ce problème depuis projet.cc. Ajoutez une description des modules en en-tête.	11		
289285	284564	4		Perfect!	Overall, good job.	11		
289304	295808	3	[L1] ball.h: 9 ou map.h: 11, ball.cpp : 53, 54, map.h: 23, 24, simulation.cpp: 29-64	Vous faites coexister aléatoirement plusieurs indentations. Certaines utilisent 4 espaces, d'autres 2, lors d'une condition à une instruction vous utilisez un certain style que vous n'utilisez pas pour une boucle à une instruction, et vous retournez à la ligne pour l'accolade de la définition des classes Ball ou Player en violation du style que vous avez dans tout le reste du code. Ceci mis à part, les variables sont très bien nommées et le camel-case respecté.	Code soigné, agréable à lire, et qui serait parfait sans les petites erreurs d'indentations à droite à gauche. Faites également attention à votre gestion des espaces : si vous en mettez avant les opérateurs = ou ==, faites le partout ! Soyez précis, cela aére le code et le rend encore plus agréable. Mention spéciale pour votre project.cpp : excellente idée de tester tous ces cas, belle initiative !	9	wrong exe name, no folder for Object files -> no compilation, extra folders: Classes, Headers	1

289324	289596	4		Attention à bien rester cohérent dans votre choix de convention pour les accolades (dans growBall), et à bien indenter votre code (simulation). Mis à part ça, c'est très lisible et bien présenté.	Le module Constants est inutile et est très similaire à l'utilisation de variables globales; c'est un problème de design et chaque valeur devrait être stockée au plus proche de son utilisation ou du concept y relatif. Vous avez beaucoup de duplication de code avec votre detection des commentaires dans le fichier lu. Also, you're never kicking any balls! :(... (la fonction est vite)	11		
289596	289324	4		Attention à bien rester cohérent dans votre choix de convention pour les accolades (dans growBall), et à bien indenter votre code (simulation). Mis à part ça, c'est très lisible et bien présenté.	Le module Constants est inutile et est très similaire à l'utilisation de variables globales; c'est un problème de design et chaque valeur devrait être stockée au plus proche de son utilisation ou du concept y relatif. Vous avez beaucoup de duplication de code avec votre detection des commentaires dans le fichier lu. Also, you're never kicking any balls! :(... (la fonction est vite)	11		

290067	300669	3	[P5] universe.cc:20.21.22 main.cc:16,17,18,19 ,20 universe.h:14,15,16 input.cc:34,37,39,77 ,91,105 player.h:16	L'indention, le wrapping et la taille des fonctions est très bien gérée. Le code est agréable à lire. Attention cependant aux Magic Numbers ! Il y a partout dans votre code des nombres bruts qui n'ont pas de signification pour des personnes ne connaisaant pas la donnée du projet !! Remplacez les par des symboles via #define ou enum par exemple. Warning : plusieurs variables ont des noms assez limites ("poor choice of symbol"). Par exemple, var1, var2, var3, var4 ne signifient rien et auraient pu soit être remplacées par input_x, input_y etc... ou un tableau input_data de 4 éléments par exempe. Relisez les Conventions !	Dans l'ensemble c'est un très bon code. Demandez néanmoins l'autorisation à Mr Boulic pour les modules additionnels, changez universe en simulation et réglez le problème des Magic Numbers et vous aurez un excellent rendu 2 !	9		
290470	296032	4		Warning : nombreuses et diverses erreurs d'indentation. Tabs : Map.cc: 97-101, snake-case : Map.cc: 131, Player.cc: 9, franglais : Player.cc: 9 Tools.h: 16. Les erreurs sont légères, et leur répétition motiverait la perte d'un point encore. Beaucoup de variables sont en français, d'autres en anglais, certaines en snake_case, d'autres en camelCase... Il faut porter plus d'attention aux conventions de codage !	Les principes du projet semblent très bien compris et le code est plutôt lisible, c'est un bon travail dans l'ensemble. Toutefois pour le prochain rendu, veillez à produire un code respectant toutes les conventions, quitte à bien repasser dessus dans un but de "clean" avant le rendu. Pensez à n'utiliser define que pour définir des constantes, utiliser une variable dans un define est un signe quasi certain que vous ne devriez pas utiliser define.	10		

295758	296442	4	ok	Good style coherence	Good job for the project ! One small remark for a lighter style : you can rewrite if (...) return true else return false, by just returning the condition inside the if statement (mainly in tools.cc).	10	execution full manual check	1

295784	296046	2	[L1] main.cc 8, 21 simulation.h 11, 22, 25, 28, 31, ... [L2] simulation.cc 14, 127, 176, 216	Look at the coding conventions for how to properly align the call to a fonction on several lines. For the wrapping lines, test with the print prewiev in geany	Overall, you only had a couple of serious mistakes: the main.cc doesn't do what it's supposed to and a some conventions issues. However at least your identation was correct and your variables names too throughout the whole code.	5		

295808	289304	3	[L1] ball.h: 9 ou map.h: 11, ball.cpp : 53, 54, map.h: 23, 24, simulation.cpp: 29-64	Vous faites coexister aléatoirement plusieurs indentations. Certaines utilisent 4 espaces, d'autres 2, lors d'une condition à une instruction vous utilisez un certain style que vous n'utilisez pas pour une boucle à une instruction, et vous retournez à la ligne pour l'accolade de la définition des classes Ball ou Player en violation du style que vous avez dans tout le reste du code. Ceci mis à part, les variables sont très bien nommées et le camel-case respecté.	Code soigné, agréable à lire, et qui serait parfait sans les petites erreurs d'indentations à droite à gauche. Faites également attention à votre gestion des espaces : si vous en mettez avant les opérateurs = ou ==, faites le partout ! Soyez précis, cela aére le code et le rend encore plus agréable. Mention spéciale pour votre project.cpp : excellente idée de tester tous ces cas, belle initiative !	9	wrong exe name, no folder for Object files -> no compilation, extra folders: Classes, Headers	1
295821	299895	4		Code très propre, clair et soigné: bon travail ! player.cc:21 : à quoi servent ces accolades sans structure de contrôle ? Beaucoup de fonctions dans simulation se ressemblent beaucoup, appliquez le principe de réutilisation. simulation.cc:241-248, 253-260 : vous pourriez remplacer ces longues conditions par une fonction booléenne pour plus de clarté.	Un projet dans l'ensemble réalisé avec sérieux, particulièrement attentif aux notions apprises en cours. Très bon travail ! Quelques petites choses à corriger pour le rendu 2, mais continuez ainsi	11		

295855	289188	4	map.cc 14,15	Warning : using ZERO and TROIS as a define is a very bad practise. I advise you to read the convention file provided on moodle on this subject. Also putting no names for parameters in a header is a bad practice and it is even worse if the code is not commented.	Despite the comments written before, the code is elegant. Keep up the good work !	10,5		
295890	302360	4	Ok.	Beaucoup de longues lignes de code . Définissez des méthodes auxiliaires, mettre des retour àa ligne fonctionne pour la correction mais votre est très dur à lire et lorsque vous devrez le modifier il faudra changer partout sans faire d'erreurs. Je ne suis pas entièrement convaincu de votre façon de placer les accolades.On ne comprend pas quelle accolade ferme quel bloc. Un retour à la ligne et une indentation bien faite sont plus clairs.	L'architecture tient la route, continuez comme ça !	10,5		
295927	300882	4		Excellent ! Ajoutez peut-être un peu de retour à la lignes pour améliorer la lisibilité et désengorgez la fonction CollisionCeCa qui est presque un gros bloc de texte !	Très bon projet. Rien à redire !	11		

295954	287759	2	[L1]ball.cc:19, map.cc:42, player.cc:40, projet.cc:27, tools.cc:12, tools.cc:32, tools.cc:34-35, ... ; [L2]player.h:28, map.cc:59, map.cc:60, player.cc:11, projet.cc:18, projet.cc:27, tools.cc:48, ... ;	Le style est trop brouillon et pas assez relu, de nombreuses erreurs de conventions subsistent	Le projet, plutôt bon dans l'ensemble, donne l'impression d'avoir toutefois été bâclé au niveau du style. Aussi, faites plus attention à la manière dont vous utilisez tools : qu'est-ce que cette classe Tools ? Il semblerait plus approprié de créer de classes correspondant aux objets géométriques afin de pouvoir les manipuler lors d'opérations mathématiques dans le plan	9		
296023	296815	3	[L1]map.cc:131, map.cc:79, map.cc:88, projet.cc:22, player.cc:20, player.cc:16, player.cc:14	Le style d'utilisation des accolades n'est pas cohérent sur l'ensemble du projet. Il manque également des tabulations.	Projet très bien réalisé, mais la lisibilité pourrait être améliorée en respectant mieux les conventions d'écriture. Les fonctions dans le fichier error.cc sont superflues, elles ne font qu'un cout.	10		

296026	301418	4	[L1] simulation.cc 222 [L2] tools.cc 68 [P5] 1) simulation.cc 60 [P5] 2) simulation.cc 17, 18 (seriously...) warning [P5] projet.cc 27	Especially with Genay's very limited default syntax highlighting erreurs in simulation.cc gets... visually unsettling. You seemingly put some effort into presenting well your code (alignments for instance), but it can still be improved (avoiding dense packs of instructions, or better organizing them (e.g. in simulation.h), leaving blank lines)	Makefile: "DO NOT DELETE THIS LINE" ... hum. Sporadic mistakes concerning the formatting of the code, but the architecture seems properly set and it is overall a good start.	11		

296032	290470	4		Warning : nombreuses et diverses erreurs d'indentation. Tabs : Map.cc: 97-101, snake-case : Map.cc: 131, Player.cc: 9, franglais : Player.cc: 9 Tools.h: 16. Les erreurs sont légères, et leur répétition motiverait la perte d'un point encore. Beaucoup de variables sont en français, d'autres en anglais, certaines en snake_case, d'autres en camelCase... Il faut porter plus d'attention aux conventions de codage !	Les principes du projet semblent très bien compris et le code est plutôt lisible, c'est un bon travail dans l'ensemble. Toutefois pour le prochain rendu, veillez à produire un code respectant toutes les conventions, quitte à bien repasser dessus dans un but de "clean" avant le rendu. Pensez à n'utiliser define que pour définir des constantes, utiliser une variable dans un define est un signe quasi certain que vous ne devriez pas utiliser define.	9	wrong sciper, wrong name Projet	1

296046	295784	2	[L1] main.cc 8, 21 simulation.h 11, 22, 25, 28, 31, ... [L2] simulation.cc 14, 127, 176, 216	Look at the coding conventions for how to properly align the call to a fonction on several lines. For the wrapping lines, test with the print prewiev in geany	Overall, you only had a couple of serious mistakes: the main.cc doesn't do what it's supposed to and a some conventions issues. However at least your identation was correct and your variables names too throughout the whole code.	5		

296048	287015	2	[L1]player.cc:19-24 ; [L2]projet.cc:38, simulation.cc:131, 191, 200, 221, 229, ... ; [P5]simulation.cc:27, 32, 42, 52, 63, 72, 82	Un style dans l'ensemble très correct et très propre, attention toutefois aux lignes trop longues. Egalement, pour votre automate de lecture, utilisez une enum plutôt que des valeurs numériques	Dans l'ensemble du bon travail, entâché par quelques petites erreurs. Ne vous découragez pas, corrigez-les et allez de l'avant pour le rendu 2 !	8		

296050	297097	4		Parfait ! Le style est très bien maitrisé et reste constant entre les modules. Remarque : il manque 2 ou 3 Magic Numbers à définir dans projet.cc (ARG_MODE, ARG_FICHIER par exemple) pour ne pas mettre de argv[2]. Aussi, FACTOR_TWO est inutile comme define, la division par deux est ici comprise comme une contrainte mathématique : ce n'est pas un magic number.	Une excellente documentation des interfaces et un code extrêmement propre, bien aligné et bien indenté. Bravo !!!	11		
296057	301225	4		Code très clair. Attention seulement à ce que la convention pour la place des accolades après les if soit cohérente sur l'ensemble du code (fin de la ligne ou début de la ligne suivante) (cf map.cc:52)	Projet très bien réalisé, l'architecture a été bien préparée pour les rendus suivants. Une brève description de vos modules pourrait être ajoutée à votre en-tête.	10	Retard 4 jours	1

296093	300411	3	[L1]simulation.cc:53 et simulation.cc:102, simulation.cc:29 et simulation.cc:62, tools.cc:41->58, simulation.h:18, tools.cc:10	Attention à l'indentation : mauvais alignement, conventions pour les accolades sur les if d'une seule ligne et de plusieurs lignes non cohérentes, sauts de ligne inutiles.	Projet très bien réalisé en ce qui concerne l'architecture, et l'encapsulation. Votre code gagnerait en lisibilité si vous corrigiez les erreurs d'indentation signalées. Attention, votre programme peut planter si un nombre incorrect de paramètres est passé au démarrage, vous pourriez gérer ce problème depuis projet.cc. Ajoutez une description des modules en en-tête.	10		
296142	289121	3	[L1]simulation.cc:34 ,42,47,50,53 map.cc:41	Le style est bon dans l'ensemble. Attention tout de même à bien respecter les règles d'indentations et ne pas faire de sauts aléatoires entre 2 espaces et 4 espaces. Warning : argv[1], argv[2] etc.. sont des Magic numbers et doivent être définis par des symboles.	Dans l'ensemble c'est un très bon code ! Continuez ainsi et corrigez ces petites erreurs et vous aurez une excellente note au rendu 2 !	10		
296165	288312	4		Style lisible, à quelqeus détails près : retour à la ligne inutile dans player.cc:22, erreurs d'alignement dans tools.cc:75 par exemple.	Projet très bien réalisé. Vous pourriez ajouter un en-tête à chaque module, décrivant sa fonction.	11		

296169	302659	2	[L1] simulation.cc: 123,132,162,171,197, map.cc : 29, ball.cc : 50 [L2] ball.h:18 , map.h:25, map.cc:17, simulation.cc: dans son ensemble	ATTENTION A VOTRE STYLE !! En effet, le code est souvent illisible car les lignes sont trop longues, le style d'indentation n'est pas constant au cours du code (accolade ouvrante sur la même ligne ou sur la ligne d'après ? il faut se décider) et les expression évaluées sont souvent sur 3 lignes et non-alignées. Etant donné que le projet se fait à deux et va vite devenir conséquent en lignes de code aux prochains rendus, je vous encourage vivement à faire attention à l'indentation et essayer d'alléger votre code pour le rendre plus lisible et donc plus facile à débugger en cas d'erreur. Warning : Pensez à remplacer les magic numbers dans projet.cc : 13,14 par des define ou enum adaptés.	Le code dans l'ensemble est très bon, vous semblez avoir bien cerné les enjeux de la modularisation et de l'architecture du projet. Il manquerait juste un peu de rigueur sur le style de programmation et votre code serait parfait : n'oubliez pas qu'un style propre aide à la résolution de problèmes et évite les erreurs.	9		
296235	297075	3	L1]simulation.h:16-23 [L1]player.h:15-29 [L1]player.cc:22,23 [L1]map.cc:14-26 [L1]ball.h:15-29 [L1]ball.cc:15-29	Be careful with the indentation!	Overall, good job	10		

296372	303028	4		Code très clair. Attention à ce que la convention utilisée pour la place des accolades reste cohérente sur tout le code.(cf simulation.cc:385)	Bravo pour votre projet : encapsulation et modularisationparfaitement mis en oeuvre, soin accordé aux détails. Vous pourriez rajouter des en-tête à vos fichiers décrivant rapidement chaque module.	10	Retard 4 jours	1
296399	296487	2	[L1] projet.cc 15 warning [L1] map.cc 77 [P2] game.cc 33 warning [P5] projet.cc 14	Even if your getters only contain simple code, I would not advise you to try and get rid of their implementation the way you do in ball.cc It will not take much time to write something that is legible (line breaks/blank lines/alignment) If you don't have room to indent your code inside your loop anymore, the solution is not simply not to do it.	Don't neglect the overall appearance of your code and it will all be fine. Some parts are already properly written, just try to be more consistent.	9		

296442	295758	4	ok	Good style coherence	Good job for the project ! One small remark for a lighter style : you can rewrite if (...) return true else return false, by just returning the condition inside the if statement (mainly in tools.cc).	11		
296487	296399	2	[L1] projet.cc 15 warning [L1] map.cc 77 [P2] game.cc 33 warning [P5] projet.cc 14	Even if your getters only contain simple code, I would not advise you to try and get rid of their implementation the way you do in ball.cc It will not take much time to write something that is legible (line breaks/blank lines/alignment) If you don't have room to indent your code inside your loop anymore, the solution is not simply not to do it.	Don't neglect the overall appearance of your code and it will all be fine. Some parts are already properly written, just try to be more consistent.	9		
296643	299689	4		Bravo pour le soin accordé au style d'écriture, code très lisible.	Très bon projet. Ajoutez simplement une description de vos modules en en-tête.	11		

296644	298386	4		Code très clair et structuré ! Les conventions de style sont bien respectées. J'aurai peut-être défini une constante (ou un enum) MODE_LANCEMENT pour le nombre dans projet.cc:22.	Excellent rendu 1. Continuez ainsi ! Pensez bien à revoir votre classe Case du module tools et de maintenir vos efforts	10		

296759	271829	3	[P2] readfile, ERROR_COLL_BALL_OBSTACLE, ERROR_COLL_OBST_PLAYER	Les noms des variables doivent être beaucoup plus clairs (c1, c2, c3 ?), le code est très obscure sinon. Les noms de fonction de devraient pas être en all caps, c'est réservé aux macros. Attention aussi à toujours indenter du même nombre d'espaces/tabs dans vos blocks (e.g. Ball.cc) C'est du code de bonne qualité dans l'ensemble.	La fonction errorSearch pourrait être omise de l'interface et appelée directement depuis reafile. Vous avez beaucoup de duplication de code dans simulation.cc pour les vérifications de collisions. Vous pourriez vérifier les collisions entre cercles et/ou rectangles directement, au lieu de player_ball, player_player, ball_ball etc. Attention aussi à variables globales! Par ailleurs, les variables statics sont autorisées, mais attention à ne pas en abuser. Si vous en avez trop (simulation.cc), c'est peut-être qu'il y a une meilleure manière de faire.	8		
296815	296023	3	[L1]map.cc:131, map.cc:79, map.cc:88, projet.cc:22, player.cc:20, player.cc:16, player.cc:14	Le style d'utilisation des accolades n'est pas cohérent sur l'ensemble du projet. Il manque également des tabulations.	Projet très bien réalisé, mais la lisibilité pourrait être améliorée en respectant mieux les conventions d'écriture. Les fonctions dans le fichier error.cc sont superflues, elles ne font qu'un cout.	10		
296816	301033	4		Vous pourriez utiliser des noms plus explicites que "cd_" pour vos varibles ou noms de fonction, pour la clarté du code. Un nom de 15 caractères n'est pas un mauvais nom (au contraire). Pensez aussi à rester cohérent dans votre choix de nomage: camelCase, snake_case	L'utilisation d'une classe Game est interessante! Vous pourriez tester les collisions entre cercles directement, ça vous éviterait la duplication de code dans vos methodes coll_* Le fait que out_of_map incrémente i est un effet de bord extrèmement surprenant et potentiellement risqué. Une méthode publique de Ball ne devrait pas avoir conscience de la signification de ce i pour l'appelant.	11		

296827	299894	4		Pensez à mettre plus d'espaces et à aérer votre code, attention aussi à l'indentation (simulation.h). C'est du code très bien présenté.	update_game_scale serait mieux en privé, il n'y a pas de raison de l'avoir dans l'interface de simulation. Vous n'avez pas besoin d'import dans .cc si c'est déjà fait dans le .h correspondant. La hitbox pourrait être calculée une seule fois et au lieu de le faire à chaque appel. Sinon, l'utilisation des classes Players et Balls est interessante!	11		
296833	284867	3	[L1]ball.cc:11-20,player.cc:9-17,tools.cc:7-57	Bien mais soyez plus consistant et ajoutez plus de retour à la ligne	Votre code est correct mais "aérez-le" avec des retours à la ligne.	10		
296867	300131	4		Nice work	Well organized, keep on the good job!	11		

296935	301358	2	[L1] player.cc 21 ball.h 16, 39 simulation.cc 23, 24, ... [L2] player.cc 64 simulation.cc 110, 140, 126, 127, ...	I strongly recomend using 4 spaces to indent (or a tab) as this leaves a bit more air in the code and it's easier to detect indentation errors. Look at the conventions on moodle for how to indent functions calls that take several lines	#ifndef and #endif should cover the whole .h in order to avoid any kind of problem, meaning #ifndef is on the first line (after the comments) and #endif is the very last line. And the #define TOOLS_H should be just after the #ifndef, always in the .h . Overall good, except the simulation.cc that presented a lot of style errors	9		
296952	283183	2	[P5]simulation.cc,155 [L1]ball.cc,7-28 [L1]map.cc,6-103 [L1]obstracle.cc,7-22 [L1]player.cc,6-31	- You should give meaningful names to: comp1, comp2, comp3, comp4, comp5, comp6, comp7, comp8, comp9 - You tabbed twice when defining the methods of the classes	You should be careful with the indentation and the naming, but overall very nice job. Keep on the good work!	9		
297075	296235	3	L1]simulation.h:16-23 [L1]player.h:15-29 [L1]player.cc:22,23 [L1]map.cc:14-26 [L1]ball.h:15-29 [L1]ball.cc:15-29	Be careful with the indentation!	Overall, good job	10		

297097	296050	4		Parfait ! Le style est très bien maitrisé et reste constant entre les modules. Remarque : il manque 2 ou 3 Magic Numbers à définir dans projet.cc (ARG_MODE, ARG_FICHIER par exemple) pour ne pas mettre de argv[2]. Aussi, FACTOR_TWO est inutile comme define, la division par deux est ici comprise comme une contrainte mathématique : ce n'est pas un magic number.	Une excellente documentation des interfaces et un code extrêmement propre, bien aligné et bien indenté. Bravo !!!	10	execution full manual check	1

297151	301452	3	[P5] tools.h : 6,7, simulation.cc : 17	A la place d'utiliser une suite de DEFINE faites une enum. Cela donne un nom à l'ensemble de ces constantes et on comprend mieux à quoi elles servent. Le fait de créer des constantes doit aider à la lecture du code : pour chaque CASE spécifier ce que fait le case (e.g CASE_TROIS -> LECTURE_NB_OBSTACLES). Le point P5 des conventions explque l'erreur sur les constantes DEUX et ZERO.	Bonne encapsulation et définissions des getters setters. Architecture simple et efficace. On peut peut-être encore rajouter des méthodes d'aides comme dans simulation par exemple.	10		

297176	301998	4		Warning : ne pas indenter ce qu'il y a sous les mots-clés public et private. Warning : si vous devez faire appel à une fonction dans un define, c'est que votre define n'est pas une constante ! Préférez dans ce cas un appel classique à cette fonction. Enfin, quitte à rajouter des define, faite le dans define.h, ainsi vous êtes sûr que tout votre code profite de la constante.	Excellent travail. Le code est clair, bien nommé, propre, c'est un plaisir de vous lire. J'ai notamment trouvé la machine à état de lecture très bien faite. Il y a des petites erreurs sans gravité à droite à gauche, veillez simplement à bien corriger les warnings. Ajoutez des en-têtes à vos fichiers avec vos noms et la description du module en une ligne. Enfin, vous devriez gérer dans projet.cc les cas où un mauvais nombre d'argument est passé pour retourner une erreur, et pas simplement EXIT_SUCCESS	11		

297846	287038	3	[L1]ball.h, 13-17 [L1]circle.h, 14-25 [L1]game.h, 17-43 [L1]obstacle.h, 12-20 [L1]player.h, 12-18	Be careful with the indentation of the classes (Please note that we don't indent the public/private keywords in class declaration)	Overall, good job.	9	6 days 13 hours late	1
298011	302747	4	[P5] map 34 sim 38 tools.h 15 warning [P5] proj 13 sim 188	Be careful with magic numbers (raw numbers in code). You did try to reduce their number, but there were still some... it was a close call!	Your efforts towards clarity can be seen. There can sometimes remain room for improvement (read_line)... There is in the end not so much to say. Good job!	11		

298386	296644	4		Code très clair et structuré ! Les conventions de style sont bien respectées. J'aurai peut-être défini une constante (ou un enum) MODE_LANCEMENT pour le nombre dans projet.cc:22.	Excellent rendu 1. Continuez ainsi ! Pensez bien à revoir votre classe Case du module tools et de maintenir vos efforts	10		
299245		4		Soyez peut-être un peu plus précis dans vos choix de noms d'arguments de fonctions ou de variables (param1, param2). À part ça, le code est très bien présenté!	Un projet.h n'est pas nécessaire puisque ce n'est pas un module utilisé ailleurs. Vos collisions ball-ball ou player-player sont identiques, vous pourriez modulariser ça en vérifiant la collision entre deux cercles directement. Enfin, vous pourriez peut-être déléguer un peu de travail à vos classes Ball, Player, Map, qui	11		

299479	299496	4	warning [L1] simulation.cc 60 warning [P5] sim 60, 270 [proj 17] [sim 23]	You crammed so many lines into the lecture function that it does not really look okay anymore... Bear in mind that if trying to get away with styles penalties lead you to write something that is not legible... there must be something wrong! Think of using indentation to aligh similar statements (getters in tools.h for instance)	Overall it seems like a good start, just don't lose sight of the fact the code is meant to be read by humans, and it'll be alright.	10	execution full manual check	1
299496	299479	4	warning [L1] simulation.cc 60 warning [P5] sim 60, 270 [proj 17] [sim 23]	You crammed so many lines into the lecture function that it does not really look okay anymore... Bear in mind that if trying to get away with styles penalties lead you to write something that is not legible... there must be something wrong! Think of using indentation to aligh similar statements (getters in tools.h for instance)	Overall it seems like a good start, just don't lose sight of the fact the code is meant to be read by humans, and it'll be alright.	10	execution full manual check	1
299554	288287	4		Assez bon style, attention toute fois à de légères erreurs de style : vos espaces sont parfois aléatoires, et les directives #include sont censées se trouver dans les .h et non les .cc. On n'inclut dans un .cc que le .h correspondant.	Beau projet, un code clair et lisible qui serait parfait si de très légères erreurs n'étaient pas présentes. Belle inititative sur la classe Entite, bien expliquée, et bonne gestion de l'héritage. Continuez ainsi pour le prochain rendu !	11		
299561	300845	4		Aucune faute mais à la limite des conventions de programmation. Pensez à adopter une tabulation plus lisible pour vos accolades	Vous avez très bien programmer donc bravo mais pensez à plus soigner votre présentation même si vous ne violez pas les conventions de programmation	11		

299689	296643	4		Bravo pour le soin accordé au style d'écriture, code très lisible.	Très bon projet. Ajoutez simplement une description de vos modules en en-tête.	11		
299724	301959	4	[L1] simulation.cc 45 simulation.h 26	Good spacing of most of the code. Pay attention occasional identation errors: for now it's only a warning	Overall, a very good job for now.	10		
299882	301272	4	ok	Coherent style in overall. Warning : care just with some space or tab that are put in place where they should not (like in tools.cc)	Good architecture and style, juste be careful with how you decompose your code. Also in some places you should lighter your code (see intersection_cercle_carre for example where one if statement could suffice).	9,5		

299892	300594	3	[P2]Simulation::decodage_ligne	Dans l'ensemble, le style du code est très bon et agréable à la lecture. Cependant beaucoup d'étourderies non pénalisées auraient pu être évitées : - Essaye de garder un style d'indentation constant et éviter les excpetions comme dans simulation.h:47-57 ou ball.h:30 - Attention aux Magic numbers (projet.cc:9) : un define ARG_FICHIER aurait très bien fait l'affaire. - Attention aux étourderies de lignes trop longues : map.h:58. En règle générale, essayez de mieux aligner votre code (par exemple aligner les "and" dans simulation.cc:575) pour le rendre plus lisible. Le point perdu est due à une fonction trop longue (+100 lignes), essayez de la décomposer comme vous l'avez fait ailleurs.	Excellent rendu 1. Continuez ainsi sans oublier de corriger les petites erreurs pointées !	9		
299894	296827	4		Pensez à mettre plus d'espaces et à aérer votre code, attention aussi à l'indentation (simulation.h). C'est du code très bien présenté.	update_game_scale serait mieux en privé, il n'y a pas de raison de l'avoir dans l'interface de simulation. Vous n'avez pas besoin d'import dans .cc si c'est déjà fait dans le .h correspondant. La hitbox pourrait être calculée une seule fois et au lieu de le faire à chaque	11		

299895	295821	4		Code très propre, clair et soigné: bon travail ! player.cc:21 : à quoi servent ces accolades sans structure de contrôle ? Beaucoup de fonctions dans simulation se ressemblent beaucoup, appliquez le principe de réutilisation. simulation.cc:241-248, 253-260 : vous pourriez remplacer ces longues conditions par une fonction booléenne pour plus de clarté.	Un projet dans l'ensemble réalisé avec sérieux, particulièrement attentif aux notions apprises en cours. Très bon travail ! Quelques petites choses à corriger pour le rendu 2, mais continuez ainsi	11		
300131	296867	4		Nice work	Well organized, keep on the good job!	11		
300253	302703	3	[L1]simulation.cc:163-240	Be careful with the indentation	Overall, good job. Keep on the hard work!	10		
300411	296093	3	[L1]simulation.cc:53 et simulation.cc:102, simulation.cc:29 et simulation.cc:62, tools.cc:41->58, simulation.h:18, tools.cc:10	Attention à l'indentation : mauvais alignement, conventions pour les accolades sur les if d'une seule ligne et de plusieurs lignes non cohérentes, sauts de ligne inutiles.	Projet très bien réalisé en ce qui concerne l'architecture, et l'encapsulation. Votre code gagnerait en lisibilité si vous corrigiez les erreurs d'indentation signalées. Attention, votre programme peut planter si un nombre incorrect de paramètres est passé au démarrage, vous pourriez gérer ce problème depuis projet.cc. Ajoutez une description des modules en en-tête.	10		

300443	301030	4	[L1]simulation.cpp:97-145, 132	Très bon travail sur le style et la propreté du code, bravo ! Attention à rester bien cohérents dans le style d'indentation des instructions contrôlées sur une seule ligne et des switch.		10		
300456	301829	3	[L2] Simulation.cc: 185, 198, 204, 211, Tool.cc: 12, 17, Tool.h: 13, 18	Dommage de mettre deux longues instructions sur une seule ligne, ou un long calcul en une fois là ou plusieurs étapes rendraient le code plus lisible	Excellent code, travail de qualité, clair et agréable à lire. Beau respect du camel-case, avec des noms de variable parlants. Une petite erreur sans gravité à corriger pour obtenir un code propre. Pour être vraiment parfait : ne mettez aucun include dans un .cc, tous les includes devraient être dans le .h, et l'unique include d'un .cc est le .h correspondant. ball.cc n'a donc que #include "ball.h"	10		
300540	287433	3	[L1]player.h,8:18 [L1]map.h,9:13 [L1]ball.h,8:17	Please note that we don’t indent the public/private keywords in class declaration.	Overall, very good job.	10		

300558	302896	0	[L1] : tout le code. [L2] : simulation.cc: 27, 29, 31, 33, 35, 37, 39, 64, 68, 73, 75, ... [P2] : simulation.cc, 47, 249, 320 [P5] : simulation.cc: 18-22, tools.h: 8, projet.cc: 11, 12, player.cc: 14	Toute l'indentation est a revoir. Aucune convention n'est respectée, le code est rendu affreusement complexe à lire par les accolades qui se baladent, les espaces aléatoires, les multiples styles qui s'affrontent, les #define abusifs... Par exemple il est inutile d'écrire #define ZERO 0, écrivez simplement un 0 là ou c'est nécessaire ! Et vos noms doivent être plus parlants, renommé char en type1 par un typedef ne sert à rien. Pour finir, lorsque vous choisissez une langue et un style de majuscule pour nommer vos variables, conservez là tout le long du code !	Les principes du projet semblent très bien compris, mais toute cette bonne performance est gachée par le style baclé du code. Ca a été vraiment compliqué de comprendre votre code, car la moindre ligne demande 2 minutes pour trouver à quel bloc elle appartient, et son role. Pour le prochain rendu il est impératif que vous repassiez sur tout votre code en supprimant les define inutiles, en renommant toutes vos variables et en reprennant l'indentation. Un code propre vous octroirea tous les points, en plus de rendre le debugging bien plus simple pour vous !	7		
300559	302674	2	[L1] setup.cc 32, map.cc 9, 11, 20, 23, 27, 56 [P5.2] tools.h and .cc often use variables named co and ce that is hard to understand the meaning of.	Warning: two styles of brackets used	Attention: main.cc doesn't work as you think: argc is always >=1 because argv[0] is always the name of the program even without additional parameters. The code is generally very well indented and spaced, with the exception of a few lines. Overall, a good job.	7		

300594	299892	3	[P2]Simulation::decodage_ligne	Dans l'ensemble, le style du code est très bon et agréable à la lecture. Cependant beaucoup d'étourderies non pénalisées auraient pu être évitées : - Essaye de garder un style d'indentation constant et éviter les excpetions comme dans simulation.h:47-57 ou ball.h:30 - Attention aux Magic numbers (projet.cc:9) : un define ARG_FICHIER aurait très bien fait l'affaire. - Attention aux étourderies de lignes trop longues : map.h:58. En règle générale, essayez de mieux aligner votre code (par exemple aligner les "and" dans simulation.cc:575) pour le rendre plus lisible. Le point perdu est due à une fonction trop longue (+100 lignes), essayez de la décomposer comme vous l'avez fait ailleurs.	Excellent rendu 1. Continuez ainsi sans oublier de corriger les petites erreurs pointées !	9		
300615	301581	2	[P2] TestsError.cpp : TestError, plus de 80 lignes [L1] TestError : 7,14,... les for devraient être indenté un cran plus loin,	Le nom des méthodes doit commencer par une minuscule. éviter les if(condition) return true, plutôt return condition. warning : vous utilisez deux styles d'indentation différents, essayez d'utiliser le même partout !	Le style reste à améliorer mais rien à redire sur le reste !	9		

300620	301641	3	[P5]simulation.cc, 99-104	Tabdonnee devrait être une classe ou une struct au lieu d'un tableau avec des magic number comme indices. warning: utiliser true et false pour bool (obstacle.cc).	mettre un en-tête avec auteurs etc aux fichiers source. attention aux performances (passage par valeur de vector). les tests balle-balle et player-player devraient être en dehors des constructeurs. Droite devrait utiliser Point. A CORRIGER: un nom de classe commence par une Majuscule (-> Ball).	10		
300669	290067	3	[P5] universe.cc:20.21.22 main.cc:16,17,18,19,20 universe.h:14,15,16 input.cc:34,37,39,77,91,105 player.h:16	L'indention, le wrapping et la taille des fonctions est très bien gérée. Le code est agréable à lire. Attention cependant aux Magic Numbers ! Il y a partout dans votre code des nombres bruts qui n'ont pas de signification pour des personnes ne connaisaant pas la donnée du projet !! Remplacez les par des symboles via #define ou enum par exemple. Warning : plusieurs variables ont des noms assez limites ("poor choice of symbol"). Par exemple, var1, var2, var3, var4 ne signifient rien et auraient pu soit être remplacées par input_x, input_y etc... ou un tableau input_data de 4 éléments par exempe. Relisez les Conventions !	Dans l'ensemble c'est un très bon code. Demandez néanmoins l'autorisation à Mr Boulic pour les modules additionnels, changez universe en simulation et réglez le problème des Magic Numbers et vous aurez un excellent rendu 2 !	9		

300803	301708	3	[L2]game_pieces.h:23,40;tools.h:8;verifiy.cc:17;verifiy.h:16	Bonnes conventions mais avec quelques lignes trop longues. Utilisez la ligne verticale à 87 char sur Geany !	Vous avez un beau code avec quelques soucis d'organisation, bravo !	9		
300845	299561	4		Aucune faute mais à la limite des conventions de programmation. Pensez à adopter une tabulation plus lisible pour vos accolades	Vous avez très bien programmer donc bravo mais pensez à plus soigner votre présentation même si vous ne violez pas les conventions de programmation	11		
300882	295927	4		Excellent ! Ajoutez peut-être un peu de retour à la lignes pour améliorer la lisibilité et désengorgez la fonction CollisionCeCa qui est presque un gros bloc de texte !	Très bon projet. Rien à redire !	11		
301001	302551	4		Parfait	Bon projet mais pensez plus à correctement déléguer vos tâches. Il s'agit de faire que chaque module fasse ce qu'on lui demande. Essayez de	10		

301030	300443	4	[L1]simulation.cpp:97-145, 132	Très bon travail sur le style et la propreté du code, bravo ! Attention à rester bien cohérents dans le style d'indentation des instructions contrôlées sur une seule ligne et des switch.		10		
301033	296816	4		Vous pourriez utiliser des noms plus explicites que "cd_" pour vos varibles ou noms de fonction, pour la clarté du code. Un nom de 15 caractères n'est pas un mauvais nom (au contraire). Pensez aussi à rester cohérent dans votre choix de nomage: camelCase, snake_case	L'utilisation d'une classe Game est interessante! Vous pourriez tester les collisions entre cercles directement, ça vous éviterait la duplication de code dans vos methodes coll_* Le fait que out_of_map incrémente i est un effet de bord extrèmement surprenant et potentiellement risqué. Une méthode publique de Ball ne devrait pas avoir conscience de la signification de ce i pour l'appelant.	11		
301225	296057	4		Code très clair. Attention seulement à ce que la convention pour la place des accolades après les if soit cohérente sur l'ensemble du code (fin de la ligne ou début de la ligne suivante) (cf map.cc:52)	Projet très bien réalisé, l'architecture a été bien préparée pour les rendus suivants. Une brève description de vos modules pourrait être ajoutée à votre en-tête.	10	Retard 4 jours	1

301265	301403	4		Les commentaires décrivant chaque fonction sont très utiles, mais se mettent normalement avant la fonction, et pas après. Pensez à choisir une unique convention pour les accolades : à la ligne ou sur la même ligne (ça s'applique aussi aux boucles et aux ifs).	La classe simulation ne devrait avoir que la méthode openSimulation dans son interface publique, puisque le reste n'est (et ne devrait pas) être utilisé par projet.cc Vos modules playerClass etc sont superflus et redondantes; votre méthode createPlayer par exemple est simplement un constructeur, en fait. La classe Data casse l'encapsulation puisqu'elle est incluse dans simulation et dans player. Vos fonctions de vérification des collisions contiennent beaucoup de duplication de code; vous pourriez modulariser en créant des fonction de test de collision entre cercles directement	10		
301272	299882	4	ok	Coherent style in overall. Warning : care just with some space or tab that are put in place where they should not (like in tools.cc)	Good architecture and style, juste be careful with how you decompose your code . Also in some places you should lighter your code (see intersection_cercle_carre for example where one if statement could suffice).	9,5		

301291	276861	4		Un code propre et agréable à lire ! Attention à bien aligner vos expressions pour le rendre encore plus lisible (tools.cc:44 par exemple) et d'éviter les "poor choice of symbols" (obj_1_id, obj_2_id ce n'est pas très explicite).	Très bon rendu 1 ! Continuez ainsi et vous ne devriez pas avoir de problème pour le rendu 2.	11		
301358	296935	2	[L1] player.cc 21 ball.h 16, 39 simulation.cc 23, 24, ... [L2] player.cc 64 simulation.cc 110, 140, 126, 127	I strongly recomend using 4 spaces to indent (or a tab) as this leaves a bit more air in the code and it's easier to detect indentation errors. Look at the conventions on moodle for how to indent functions calls that take several lines	#ifndef and #endif should cover the whole .h in order to avoid any kind of problem, meaning #ifndef is on the first line (after the comments) and #endif is the very last line. And the #define TOOLS_H should be just after the #ifndef, always in the .h. Overall good, except the	9		

301403	301265	4		Les commentaires décrivant chaque fonction sont très utiles, mais se mettent normalement avant la fonction, et pas après. Pensez à choisir une unique convention pour les accolades : à la ligne ou sur la même ligne (ça s'applique aussi aux boucles et aux ifs).	La classe simulation ne devrait avoir que la méthode openSimulation dans son interface publique, puisque le reste n'est (et ne devrait pas) être utilisé par projet.cc Vos modules playerClass etc sont superflus et redondantes; votre méthode createPlayer par exemple est simplement un constructeur, en fait. La classe Data casse l'encapsulation puisqu'elle est incluse dans simulation et dans player. Vos fonctions de vérification des collisions contiennent beaucoup de duplication de code; vous pourriez modulariser en créant des fonction de test de collision entre cercles directement	10		
301411	302276	4	warning [P5] proj 14 17	decodage_ligne is too long... It seems legibility gradually deteriorates as you browse simulation.cc Think of aligning similar statements (e.g. declarations in multi_obstacle) and leaving blank lines Warning: you use different indentation styles within the code (braces of if statements for instance)	The project seem rather well designed, but you may want to try to make the code look more "airy". It oftens looks like a pack of obscure statements just because of the density... It should still be a good starting point for the next steps!	11		
301418	296026	4	[L1] simulation.cc 222 [L2] tools.cc 68 [P5] 1) simulation.cc 60	Especially with Genay's very limited default syntax highlighting erreurs in simulation.cc gets... visually unsettling. You seemingly put some effort into presenting well your code (alignments for instance), but it can still be improved (avoiding dense packs of instructions, or better organizing them (e.g. in simulation.h), leaving	Makefile: "DO NOT DELETE THIS LINE" ... hum. Sporadic mistakes concerning the formatting of the code, but the architecture seems properly set and it is overall a good start.	11		

301437	284896	3	[L1] test.cc: 77, tool.c: 13, 14, 39, 42, 45, 48, lecture.cc: 69	Beaucoup de petites incohérences dans l'indentation. Pas de retour à la ligne pour une accolade fermante, des instructions doublement indentées, etc...	Très bon code, beau travail. Pourrait être parfait si vous aviez fait un petit clean avant de rendre votre code, pour corriger notamment l'indentation parfois imprécise. Entre autre : ça serait une bonne chose de tester le cas où le nombre d'argument n'est pas bon afin d'afficher un message à l'utilisateur pour corriger son erreur. Autre défaut : les includes doivent aller dans le .h et non le .cc ! Le seul include qui peut rester dans un .cc est l'include du .h correspondant.	10		
301452	297151	3	[P5] tools.h : 6,7, simulation.cc : 17	A la place d'utiliser une suite de DEFINE faites une enum. Cela donne un nom à l'ensemble de ces constantes et on comprend mieux à quoi elles servent. Le fait de créer des constantes doit aider à la lecture du code : pour chaque CASE spécifier ce que fait le case (e.g CASE_TROIS -> LECTURE_NB_OBSTACLES). Le point P5 des conventions explque l'erreur sur les constantes DEUX et ZERO.	Bonne encapsulation et définissions des getters setters. Architecture simple et efficace. On peut peut-être encore rajouter des méthodes d'aides comme dans simulation par exemple.	10		
301464	286962	4		Bien. Bravo !	Votre code est correct et il est bien présenté mais vous devriez plus correctement structurer la manière dont vous déléguer vos tâches.	10		

301494	283616	3	[L1] moduleLec.cc in function traitement	The use of curly bracket inside case is not the convention and you use not rigourously	Good code. If you find time try to add some short comment in the header file to make it easier to read.	10		

301560	287173	4	[L1]simulation.cc:55	On note un très grand soin apporté sur le style qui reste clair et cohérent sur l'ensemble du code : de l'excellent travail !	Un rendu d'une grande qualité, bravo ! Pour le suivant, je vous recommande de vous repencher sur la façon dont vous utilisez le module tool.	11		
301581	300615	2	[P2] TestsError.cpp : TestError, plus de 80 lignes [L1] TestError : 7,14,... les for devraient être indenté un cran plus loin, TestBallOut : 123	Le nom des méthodes doit commencer par une minuscule. éviter les if(condition) return true, plutôt return condition. warning : vous utilisez deux styles d'indentation différents, essayez d'utiliser le même partout !	Le style reste à améliorer mais rien à redire sur le reste !	8	execution full manual check	1

301595	302759	2	[L2]tools.cc:19,27,simulation.cc:131,132,134,etc.[L1]simulation.cc:199-207,249-249,tools.cc:19:23,27:31	Vous avez quelques manquements aux conventions de programmation.Pratiquez la relecture !	Vous avez des problèmes au sein de votre Makefile et vous n'utilisez pas vraiment le principe de classe. Sinon le code est lisible ce qui est une bonne chose.	4,5	damaged	1
301633	302909	3	[L1] tool.cc : 8,9,19,12,13, simulation.cc : 15, projet.cc : 10,12, player.cc : 9, map.cc : 17,18, simulation.cc : 43	Dans map.cc à la place de faire if(condition){} else {//quelque chose} faites if(!condition){//quelque chose}. La constante RAYON_NUL a un nom qui évoque peu à quoi elle sert. Reliser le P5 des conventions. Dans simulation .h les lignes sont trop longues. peut-être une erreur de symbole ou l'interprétation du caractère de reotur à la ligne. Bien vérifier le code dans l'environnement de test, la VM.	Globalement bon code. Continuez comme ça !	10		
301641	300620	3	[P5]simulation.cc, 99-104	Tabdonnee devrait être une classe ou une struct au lieu d'un tableau avec des magic number comme indices. warning: utiliser true et false pour bool (obstacle.cc).	mettre un en-tête avec auteurs etc aux fichiers source. attention aux performances (passage par valeur de vector). les tests balle-balle et player-player devraient être en dehors des constructeurs. Droite devrait utiliser Point. A CORRIGER: un nom de classe commence par une Majuscule (-> Ball).	10		
301708	300803	3	[L2]game_pieces.h:23,40;tools.h:8;verifiy.cc:17;verifiy.h:16	Bonnes conventions mais avec quelques lignes trop longues. Utilisez la ligne verticale à 87 char sur Geany !	Vous avez un beau code avec quelques soucis d'organisation, bravo !	9		

301750	302031	3	[L2]simulation.h:17, simulation.cc:23,108 ,112	Bonne présentation dans l'ensemble.	Excellent projet: il n'y a rien à redire sur le code !	10		
301766	302289	2	[L1] ball.cc:9, lecture.cc:16, tests.cc:11, tests.cc:151, ... ; [P5]functions.cc:35, 40,45,50, tests.cc:133-136, 147-150, ...	Dans l'ensemble l'attention portée au style est assez insuffisante, mais celui-ci reste lisible. Attention à l'alignement des instructions sur plusieurs lignes. Attention également à rester constants dans le choix de style appliqué aux instructions controlées d'une seule ligne. Egalement, pensez à utiliser des define ou des enum pour désigner les coins entre autres (magic numbers).	Travail dans l'ensemble bon, mais le style laisse un peu à désirer : retravaillez ça ainsi que votre architecture pour le prochain rendu. Pour les tests, utilisez mieux le principe de réutilisation.	9		
301829	300456	3	[L2] Simulation.cc: 185, 198, 204, 211, Tool.cc: 12, 17, Tool.h: 13, 18	Dommage de mettre deux longues instructions sur une seule ligne, ou un long calcul en une fois là ou plusieurs étapes rendraient le code plus lisible	Excellent code, travail de qualité, clair et agréable à lire. Beau respect du camel-case, avec des noms de variable parlants. Une petite erreur sans gravité à corriger pour obtenir un code propre. Pour être vraiment parfait : ne mettez aucun include dans un .cc, tous les includes devraient être dans le .h, et l'unique include d'un .cc est le .h correspondant. ball.cc n'a donc que #include "ball.h"	10		
301836	303153	3	[L2] simulation.cc : 32,45,155,193,260	Ok.	Le rendu global est bon. Le code est assez concis. Continuez comme ça !	10		

301881	273712	4		Dans l'ensemble, le code est propre et le style est bon. Toutefois, j'ai quelques doutes quant à la façon dont vous avez choisi d'implémenter tool : qu'est-ce que Const_data ? Normalement, dans tool, privilégiez de créer des classes Circle, Square etc pour performer des opérations géométriques. Ce n'est également pas le rôle de tool d'effectuer la lecture des données	Dans l'ensemble le projet est bien réalisé et on note des efforts pour le style. Toutefois, faites plus attention aux divers modules et à la façon dont vous les utilisez par rapport à ce pourquoi ils sont sensés être conçus.	11		
301949	302565	4	Ok.	Les noms des constantes peuvent être séparés par des underscores (e.g MACONSTANTE -> MA_CONSTANTE). Evitez "if(condition) { return true} else {return false}" utilisez plutôt "return condition".	Bonne subdivision du travail entre chaque classe et la création d'interfaces supplémentaires rend votre code très maniable.	11		
301956	301994	3	[L1] main.cc 15, 16 simulation.cc 152, 208	Your identation is perfect, with the exception of small mistakes here and there	Overall, a very good job for now, it's a shame for the identation distraction errors here and there...	10		
301959	299724	4	[L1] simulation.cc 45 simulation.h 26	Good spacing of most of the code. Pay attention occasional identation errors: for now it's only a warning	Overall, a very good job for now.	10		
301994	301956	3	[L1] main.cc 15, 16 simulation.cc 152, 208	Your identation is perfect, with the exception of small mistakes here and there	Overall, a very good job for now, it's a shame for the identation distraction errors here and there...	10		

301998	297176	4		Warning : ne pas indenter ce qu'il y a sous les mots-clés public et private. Warning : si vous devez faire appel à une fonction dans un define, c'est que votre define n'est pas une constante ! Préférez dans ce cas un appel classique à cette fonction. Enfin, quitte à rajouter des define, faite le dans define.h, ainsi vous êtes sûr que tout votre code profite de la constante.	Excellent travail. Le code est clair, bien nommé, propre, c'est un plaisir de vous lire. J'ai notamment trouvé la machine à état de lecture très bien faite. Il y a des petites erreurs sans gravité à droite à gauche, veillez simplement à bien corriger les warnings. Ajoutez des en-têtes à vos fichiers avec vos noms et la description du module en une ligne. Enfin, vous devriez gérer dans projet.cc les cas où un mauvais nombre d'argument est passé pour retourner une erreur, et pas simplement EXIT_SUCCESS	11		
302017	287459	4		Parfait ! Le style est bien maitrisé et reste constant tout au long du code !! Warning : dans le main, les nombres dans argv[xx] sont aussi des Magic Numbers. N'hésitez pas à faire un enum ou des define pour ces derniers. Remarque : map.cc:12 est assez confuse comme ligne, ce serait cool de mieux l'aligner ou de l'initialiser dans le corps du constructeur pour que cela soit plus propre, mais c'est un détail.	Une excellente documentation du code ! Continuez ainsi	11		
302031	301750	3	[L2]simulation.h:17, simulation.cc:23,108,112	Bonne présentation dans l'ensemble.	Excellent projet: il n'y a rien à redire sur le code !	10		

302076	270354	4		I suggest you leave a bit of air in your code	In general, a good code. The main issues are with it's compactness which makes it not very pleasant to read, but what really penalised you were small mistakes such as the constructors lists in the .h	6,5	archive file is damaged	1
302276	301411	4	warning [P5] proj 14 17	decodage_ligne is too long... It seems legibility gradually deteriorates as you browse simulation.cc Think of aligning similar statements (e.g. declarations in multi_obstacle) and leaving blank lines Warning: you use different indentation styles within the code (braces of if statements for instance)	The project seem rather well designed, but you may want to try to make the code look more "airy". It oftens looks like a pack of obscure statements just because of the density... It should still be a good starting point for the next steps!	11		

302289	301766	2	[L1] ball.cc:9, lecture.cc:16, tests.cc:11, tests.cc:151, ... ; [P5]functions.cc:35, 40,45,50, tests.cc:133-136, 147-150, ...	Dans l'ensemble l'attention portée au style est assez insuffisante, mais celui-ci reste lisible. Attention à l'alignement des instructions sur plusieurs lignes. Attention également à rester constants dans le choix de style appliqué aux instructions controlées d'une seule ligne. Egalement, pensez à utiliser des define ou des enum pour désigner les coins entre autres (magic numbers).	Travail dans l'ensemble bon, mais le style laisse un peu à désirer : retravaillez ça ainsi que votre architecture pour le prochain rendu. Pour les tests, utilisez mieux le principe de réutilisation.	9		
302327	302584	4		Le code est propre, bien indenté et très bien aligné ! Bravo. Warnings : - Deux lignes dépassent les 87 caractères autorisés [L2]simulation.cc : 198, 216 - argv[1], argv[2] etc.. sont des Magic numbers ! N'hésitez pas à définir des symboles ou utiliser un enum pour les remplacer. - corner1, corner2 etc.. sont des "poor choice of symbol" ! Favorisez des noms plus explicites ou au moins documentez ces noms car un lecteur ne peut pas savoir quel corner fait référence au coin en haut à droite par exemple.	Votre code est agréable à la lecture et très bien documenté ! Continuez ainsi sans oublier de corriger les petites erreurs pour avoir un excellent Rendu 2.	10		
302360	295890	4	Ok.	Beaucoup de longues lignes de code . Définissez des méthodes auxiliaires, mettre des retour àa ligne fonctionne pour la correction mais votre est très dur à lire et lorsque vous devrez le modifier il faudra changer partout sans faire d'erreurs. Je ne suis pas entièrement convaincu de votre façon de placer les accolades.On ne comprend pas quelle accolade ferme quel bloc. Un retour à la ligne et une indentation bien faite sont plus clairs.	L'architecture tient la route, continuez comme ça !	10,5		

302551	301001	4		Parfait	Bon projet mais pensez plus à correctement déléguer vos tâches. Il s'agit de faire que chaque module fasse ce qu'on lui demande. Essayez de restructurer mais le fond est correct et fonctionnel.	10		
302565	301949	4	Ok.	Les noms des constantes peuvent être séparés par des underscores (e.g MACONSTANTE -> MA_CONSTANTE). Evitez "if(condition) { return true} else {return false}" utilisez plutôt "return condition".	Bonne subdivision du travail entre chaque classe et la création d'interfaces supplémentaires rend votre code très maniable.	11		
302583	282836	3	[P5]2	It is good to define some macro for constant, but here you use too poor names in many place to define them (for example MOITIE, OPPOSE, DOUBLE in tools.h	We see the effort put in the project. Try to fix the mentionned problems and add some comments in the header file if you have time and it will be great !	9		

302584	302327	4		Le code est propre, bien indenté et très bien aligné ! Bravo. Warnings : - Deux lignes dépassent les 87 caractères autorisés [L2]simulation.cc : 198, 216 - argv[1], argv[2] etc.. sont des Magic numbers ! N'hésitez pas à définir des symboles ou utiliser un enum pour les remplacer. - corner1, corner2 etc.. sont des "poor choice of symbol" ! Favorisez des noms plus explicites ou au moins documentez ces noms car un lecteur ne peut pas savoir quel corner fait référence au coin en haut à droite par exemple.	Votre code est agréable à la lecture et très bien documenté ! Continuez ainsi sans oublier de corriger les petites erreurs pour avoir un excellent Rendu 2.	10		
302659	296169	2	[L1] simulation.cc: 123,132,162,171,197, map.cc : 29, ball.cc : 50 [L2] ball.h:18 , map.h:25, map.cc:17, simulation.cc: dans son ensemble	ATTENTION A VOTRE STYLE !! En effet, le code est souvent illisible car les lignes sont trop longues, le style d'indentation n'est pas constant au cours du code (accolade ouvrante sur la même ligne ou sur la ligne d'après ? il faut se décider) et les expression évaluées sont souvent sur 3 lignes et non-alignées. Etant donné que le projet se fait à deux et va vite devenir conséquent en lignes de code aux prochains rendus, je vous encourage vivement à faire attention à l'indentation et essayer d'alléger votre code pour le rendre plus lisible et donc plus facile à débugger en cas d'erreur. Warning : Pensez à remplacer les magic numbers dans projet.cc : 13,14 par des define ou enum adaptés.	Le code dans l'ensemble est très bon, vous semblez avoir bien cerné les enjeux de la modularisation et de l'architecture du projet. Il manquerait juste un peu de rigueur sur le style de programmation et votre code serait parfait : n'oubliez pas qu'un style propre aide à la résolution de problèmes et évite les erreurs.	9		

302674	300559	2	[L1] setup.cc 32, map.cc 9, 11, 20, 23, 27, 56 [P5.2] tools.h and .cc often use variables named co and ce that is hard to understand the meaning of.	Warning: two styles of brackets used	Attention: main.cc doesn't work as you think: argc is always >=1 because argv[0] is always the name of the program even without additional parameters. The code is generally very well indented and spaced, with the exception of a few lines. Overall, a good job.	7		
302703	300253	3	[L1]simulation.cc:163-240	Be careful with the indentation	Overall, good job. Keep on the hard work!	10		
302747	298011	4	[P5] map 34 sim 38 tools.h 15 warning [P5] proj 13 sim 188	Be careful with magic numbers (raw numbers in code). You did try to reduce their number, but there were still some... it was a close call!	Your efforts towards clarity can be seen. There can sometimes remain room for improvement (read_line)... There is in the end not so much to say. Good job!	11		

302759	301595	2	[L2]tools.cc:19,27,simulation.cc:131,132,134,etc.[L1]simulation.cc:199-207,249-249,tools.cc:19:23,27:31	Vous avez quelques manquements aux conventions de programmation.Pratiquez la relecture !	Vous avez des problèmes au sein de votre Makefile et vous n'utilisez pas vraiment le principe de classe. Sinon le code est lisible ce qui est une bonne chose.	4,5	damaged	1

302896	300558	0	[L1] : tout le code. [L2] : simulation.cc: 27, 29, 31, 33, 35, 37, 39, 64, 68, 73, 75, ... [P2] : simulation.cc, 47, 249, 320 [P5] : simulation.cc: 18-22, tools.h: 8, projet.cc: 11, 12, player.cc: 14	Toute l'indentation est a revoir. Aucune convention n'est respectée, le code est rendu affreusement complexe à lire par les accolades qui se baladent, les espaces aléatoires, les multiples styles qui s'affrontent, les #define abusifs... Par exemple il est inutile d'écrire #define ZERO 0, écrivez simplement un 0 là ou c'est nécessaire. Et vos noms doivent être plus parlants, renommer char en type1 par un typedef est inutile. Pour finir, lorsque vous choisissez une langue et un style de majuscule pour nommer vos variables, conservez là tout le long du code !	Les principes du projet semblent très bien compris, mais toute cette bonne performance est gachée par le style baclé du code. Ca a été vraiment compliqué de comprendre votre code, car la moindre ligne demande 2 minutes pour trouver à quel bloc elle appartient, et son role. Pour le prochain rendu il est impératif que vous repassiez sur tout votre code en supprimant les define inutiles, en renommant vos variables et en reprennant l'indentation. Un code propre vous octroirea tous les points, en plus de rendre le debugging bien plus simple pour vous !	7		
302909	301633	3	[L1] tool.cc : 8,9,19,12,13, simulation.cc : 15, projet.cc : 10,12, player.cc : 9, map.cc : 17,18, simulation.cc : 43	Dans map.cc à la place de faire if(condition){} else {//quelque chose} faites if(!condition){//quelque chose}. La constante RAYON_NUL a un nom qui évoque peu à quoi elle sert. Reliser le P5 des conventions. Dans simulation .h les lignes sont trop longues. peut-être une erreur de symbole ou l'interprétation du caractère de reotur à la ligne. Bien vérifier le code dans l'environnement de test, la VM.	Globalement bon code. Continuez comme ça !	9	execution full manual check	1

303011	288252	3	[P5] player.cc: 55, ball.cc: 61, simulation.cc: 28, 43, 45, 59, 132, 135...	L'indentation pourrait être améliorée, mais elle n'est pas catastrophique. Il y a beaucoup de petites erreurs dans simulation.cc notamment (lignes 32, 33, 71) Warning : Vous utilisez deux styles pour vos accolades, l'un pour les fonctions, classes et boucles, l'autre pour les conditions et les struct. Il faut uniformiser tout cela. Warning : Nommez correctement vos variables ! Français OU anglais, snake-case OU camel-case, et tenez-y vous. Warning : le define TAILLE_LIGNE est inutilisé. Par ailleurs, quitte à rajouter des defines, ajoutez les dans le fichier define.h, ainsi vous êtes sûrs que tout votre code en profite	L'idée globale semble bien comprise, l'architecture est bonne et le tout est cohérent. Il faut toutefois faire plus attention à la propreté du code : les espaces et les accolades sont irréguliers, vous nommez vos variables tantôt en snake-case, tantôt en camel-case, certaines variables commencent par des majuscules... Sans être grave, cela rend votre code bien moins compréhensible. Autre conseil : évitez la déclaration de fonction à côté d'une classe, car dans tous les cas ces fonctions auraient du être des méthodes de classe. Corrigez tout ça pour le prochain rendu et il sera d'une excellente qualité ! Bravo pour le bon travail déjà accompli.	10		
303028	296372	4		Code très clair. Attention à ce que la convention utilisée pour la place des accolades reste cohérente sur tout le code.(cf simulation.cc:385)	Bravo pour votre projet : encapsulation et modularisationparfaitement mis en oeuvre, soin accordé aux détails. Vous pourriez rajouter des en-tête à vos fichiers décrivant rapidement chaque module.	10	Retard 4 jours	1
303039	283652	3	[L1] simulation.cc 265, 271, in player.h, in map.h	Many places where the indentation are neither respected nor coherent with the rest of the code	Good code in overall. In some place your code could be more succint, for example "if (...) return true, else return false" could be simplified in "return (...)". However after correcting the few comments it will already be great.	8,5		

303153	301836	3	[L2] simulation.cc : 32,45,155,193,260	Ok.	Le rendu global est bon. Le code est assez concis. Continuez comme ça !	10		
305538	270354	4		I suggest you leave a bit of air in your code	In general, a good code. The main issues are with it's compactness which makes it not very pleasant to read, but what really penalised you were small mistakes such as the constructors lists in the .h	6,5	archive file is damaged	1