

296190	4.00		Le style est très bon mais il faut faire attention à ce que les expressions écrivent sur plusieurs lignes soient correctement alignées et que le code soit facilement lisible. Il ne faut pas non plus hésiter à utiliser des switch afin d'éviter de mettre trop de if ... else.	Le code et le style sont très bon mais il faut faire attention à ne pas trop se compliquer la tâche.
301678	3.00	[L1] simulation.c.124, squarecell.cc.176, fourmillière.h.17, Fourmi.h.18	[L1] Il faut indenter les cases, il y a une tabulation en trop, il ne faut pas indenter les mots-clé de portée	Bon travail, le code et sa structure sont très soignés. Pour gagner en place et en clareté, vous pourriez éviter de sauter une ligne au début de chaque fonction, Très bon travail, continuez ainsi !
302461	4.00		Parfait	Code très agréable à lire, attention à l'encapsulation. L'ensemble est très lisible et clair.
312271	3.00	[P2] deux fonctions plus grandes que 40 lignes : Simulation::lecture dans simulation.cc et lecture_fourmilliere dans fourmillière.cc	Warning: Evitez les noms de variables avec une seule lettre (excepté pour les compteurs explicites et locaux comme dans une boucle for), et essayez de nommer vos variable d'avantage en fonction de ce qu'elles représentent (par exemple appelez un pointeur sur une fourmi ant plutôt que pr)	Bon travail, le code est très bien structuré ! Attention aux nombreux noms de variables qui ne sont pas toujours clairs, ainsi qu'à garder vos styles de noms de fonctions cohérents (par exemple getGeneratrice vs get_collectors) et significatifs : par exemple vos fonctions d'affichage pourraient simplement s'appeler show() car, même si elles ont initialement été créées pour du debug, elles restent pertinentes pour un affichage de manière générale et une convention de nommage comme celle ci est plus perturbante qu'efficace, en témoigne votre commentaire nécessaire pour l'expliquer. Con courage pour la suite !
314458	4.00	good	use of global variables is not permitted	proper tabbing, good file structure, just make sure all variables are private
314564	3.00	[L1] simulation.h 23, simulation.cc 82,113,126, squarecell.cc 44, nourriture.cc 18-27, fourmi.cc 20,27,34,41	Double indentation dans toutes les classes (on n'indente pas les mots-clés "public" et "private"); indenter les "case"	Bon travail d'architecture et de structuration; attention à déclarer les variables locales à un module comme "static" ou dans le namespace non-nommé; beaucoup d'erreurs d'indentation
314664	4.00	Bravo	Rien à dire	Impécable! Très lisible, commentaire pertinents, bonne implémentation des dépendances, rien à dire. Bravo
315034	4.00	L1:nourriture.cc:26-31 L2:fourmilier.h:9		Attention à contrôler le nombre d'argument (argc) dans votre main. Très bon style de code dans tout le projet, à continuer pour la suite!

315144	3.00	[P5] magic number	<p>WARNING: [C2] mauvaise définition de constructeur dans simulation.h (soit on défini dans la class soit dans le .cc) Warning: projet.cc debugage, et dans les sous-type de fourmi Warning: projet.h inutile + vide Warning: définition de fonction dans le header simulation.h Warning: [L1] simulation.h indentation 15-18 et dans private (double indentation) fourmilere.cc, fourmi.cc il manque une tabulation après un retour à la ligne dans les prototypes de fonction Warning: simulation.cc il manque les "(" et fonction l.131 est un bool mais ne retourne rien Warning: il manque ifndef ... define... endif dans tous les module (le .h) (cela risque de ne pas compiler avec des inclusions multiples) Warning: mauvaise déclaration des variables fourmilere.h Warning: projet.o n'est pas effacé du dossier</p>	<p>Votre code manque de cohérence. Il y a à la fois des concept avancé sur l'héritage qui sont bien implémenté et parfois ces mêmes concepts ne le sont pas. Parfois les variables sont bien initialisées parfois pas. Je pense que pour la prochaine fois vous devriez vous concentrer sur des choses plus simple et de bien les comprendre. N'oubliez pas de faire des pause de temps en temps et relisez votre code avant de coder. Par contre j'ai vu que vous avez pris la bonne habitude d'annoter votre code de manière à ce que cela soit visible, continuez comme ça.</p>
315404	3.00	[L1] squarecell.cc 11,59, nourriture.h 21, nourriture.cc 9-10, fourmilere.h 47,54, fourmi.h 24		<p>Dans l'ensemble très bon travail; il faut cependant faire attention à l'indépendance de squarecell, aux variables globales dans un .cc (à déclarer avec "static" ou dans le namespace non-nommé), et à l'indentation</p>
315678	3.00	[L2] modules fourmi.cc fourmi.h fourmillere.cc fourmillere.h	<p>warning: double indentation dans squrcell.cc L 64 et 65</p>	<p>points sur l'externalisation des méthodes. Si vous voulez implémenter le constructeur dans l'interface il faut qu'il soit sur une seule ligne.</p>
315933	4.00		<p>paramètres dans la définition du constructeur de Fourmilere (pour des raisons de documentation). J'évitais de mettre des accolades dans les case d'un switch. A part cela le style est très bon.</p>	<p>Très bon !</p>

316149	4.00	warning: inconsistent brackets	method bodies should be externalized if they are more than 1 line long	followed structure, properly tabbed, had all constants, just make sure to externalize constructors more than 1 line long
316468	4.00	warning: inconsistent brackets	method bodies should be externalized if they are more than 1 line long	followed structure, properly tabbed, had all constants, just make sure to externalize constructors more than 1 line long
316708	3.00	[L1] simulation.h 23, simulation.cc 82,113,126, squarecell.cc 44, nourriture.cc 18-27, fourmi.cc 20,27,34,41	Double indentation dans toutes les classes (on n'indente pas les mots-clés "public" et "private"); indenter les "case"	Bon travail d'architecture et de structuration; attention à déclarer les variables locales à un module comme "static" ou dans le namespace non-nommé; beaucoup d'erreurs d'indentation
324605	4.00	L1:nourriture.cc:26-31 L2:fourmilier.h:9		Attention à contrôler le nombre d'argument (argc) dans votre main. Très bon style de code dans tout le projet, à continuer pour la suite!
324718	4.00		Style excellent !	Le code est le style sont très bons !
324806	3.00	[L1] squarecell.h 35-63, simulation.h 6, simulation.cc 37-99, fourmilier.cc 39-42	Double indentation dans toutes les classes (on n'indente pas les mots-clés "public" et "private"); indenter les "case"; plusieurs styles incohérents d'accolades pour les "if"	Dans l'ensemble très bon travail, il faut juste faire attention à l'indépendance de squarecell et aux conventions d'indentation.
325002	3.00	[L1]sqarecel.cc 108,116,125,81...	warning: revisit your indentation	Great work, just be careful about squarecell being completely independant
325562	2.00	[L1] fourmilier.cc dépassement de ligne 31 squarecell.cc [P5] squarecell.cc fonction remplissage > 40 lignes	Warning: indentation simulation.cc 37-61 Warning: simulation.cc utilisation de LIGNE qui n'est déclaré nul part. Warning: si LIGNE est une variable, elle doit être écrite en minuscule Warning: le makefile n'est pas correctement rempli cela ne peut pas compiler Warning: simulation.cc pas d'utilisation du break dans le switch	le code est malheureusement très incomplet et visiblement il ne peut pas compiler (le makefile n'est pas conçu pour votre programmes et certaines variables ne sont pas initialisées
325743	4.00		Parfait	Code très clair et lisible, agréable à lire.

325856	2.00	[L2]anthill.cc 22-23,43,72,80,90,96,105,anthill.h25,fourmi.cc70,73,83,119,126,132,nourriture.cc42,simulation.cc24,97,squarecell.cc11,[P2]simulation.cc49-172	Attention beaucoup de lignes sont trop longues ! La fonction <code>decodage_ligne</code> dépasse aussi largement la limite de 80 lignes. Les noms des différentes fonctions pourraient être plus descriptifs. Mis à part cela, le style du code est très bon.	Le code est très bien fait et le style est bon. Il faut juste faire attention aux quelques fautes d'inattention qui se sont glissées dans le projet.
325961	3.00	[L1] simulation.cc 70, 81	warning: please avoid ant in your methods names in squarecell since squarecell is considered as a library and should not be related to the rest of the Tchanz project	really good implementation and respect of architecture, just some small indentation mistakes but really good job overall
326003	4.00	[L2] simulation.cc 121	fourmiellere should not depend on simulation, squarecell must be independent, no global variables, warning: line too long	several violations of project architecture, and use of global variables.
326025	3.00	[L1]simulation.cc46-51	warning: aligner public et private à gauche	le code est malheureusement très incomplet et visiblement il ne peut pas compiler. Plusieurs choix sont étranges comme de mémoriser <code>argc</code> et <code>argv</code> dans la classe <code>simulation</code> ; <code>squarecell</code> est plus avancé et clair mais il manque la définition de plusieurs types.

326065	3.00	[P5] magic number	<p>WARNING: [C2] mauvaise définition de constructeur dans simulation.h (soit on défini dans la class soit dans le .cc) Warning: projet.cc debugage, et dans les sous-type de fourni Warning: définition de fonction dans le header simulation.h Warning: [L1] simulation.h indentation 15-18 et dans private (double indentation) fourmiliere.cc, fourni.cc il manque une tabulation après un retour à la ligne dans les prototypes de fonction Warning: simulation.cc il manque les "()" et fonction l.131 est un bool mais ne retourne rien Warning: il manque ifndef ... define... endif dans tous les module (le .h) (cela risque de ne pas compiler avec des inclusions multiples) Warning: mauvaise déclaration des variables fourmiliere.h Warning: projet.o n'est pas effacé du dossier</p>	<p>Votre code manque de cohérence. Il y a à la fois des concept avancé sur l'héritage qui sont bien implémenté et parfois ces mêmes concepts ne le sont pas. Parfois les variables sont bien initialisées parfois pas. Je pense que pour la prochaine fois vous devriez vous concentrer sur des choses plus simple et de bien les comprendre. N'oubliez pas de faire des pause de temps en temps et relisez votre code avant de coder. Par contre j'ai vu que vous avez pris la bonne habitude d'annoter votre code de manière à ce que cela soit visible, continuez comme ça.</p>
326251	4.00	OK	Good style.	<p>Good coding style but be careful with your project conception. Squarecell should not be a class. The process class job could have been done by the simulation one.</p>
326346	1.00	L1:food.h:16-40; fourmiliere.cc :12-83; fourmiliere.cc :11-61; fourmis.cc :14-49;fourmis.h:14-35;... L2:fourmiliere.cc:35;fourmilier.e.h:26;fourmis.h:69,99,112,12 2simulation.cc:46 P2:simulation.cc:decodage_lig ne	warning: aligner public et private à gauche.	<p>Attention à l'indentation dans l'ensemble du projet. Pensez également à ne pas indenter les private, public et protected dans les classes. Il faut également raccourcir la fonction de décodage pour respecter les 80 lignes max par fonction. Dans les prototypes des méthodes de la classe il ne faut pas mettre l'implémentation, cela se fait dans les fichiers .cc. Bonne continuation pour les prochains rendus!</p>

326349	4.00	[L2] simulation.cc 121	fourmiellere should not depend on simulation, squarcell must be independent, no global variables, warning: line too long	several violations of project architecture, and use of global variables.
326369	4.00		Style excellent !	Le code est le style sont très bons !
326463	3.00	[L1]simulation.cc 63,67,72,78	Style d'accolades inconsistent sur l'ensemble du code, particulièrement pour les if/else (simulation.cc, fourmilere.cc,squarecell.cc). Même chose pour l'indentation des switch (simulation.cc, fourmilere.cc).	Le style général est bon, la structure du code est claire. Bonne utilisation de namespaces, bonne structuration du projet.
326477	1.00	[L1]anthill.h16,28,30 anthill.cc 28-36 [L2] anthill.cc 32,68 simulation.cc 8,124,125 [P1] squarecell.cc superposition_carre (80l), simulation.cc decodage_ligne (51l)	Only write one instruction by line (e.g not like simulation.cc 8). Only one function can be more than 40 lines (and under 80).	Good code but be careful with your style.
326655	3.00	[P2]simulation35:170	max # of lines was 80, this method is 140 lines long	all the structure and formatting rules are respected, just break up your method into smaller traitements
326790	3.00	[P2] deux fonctions plus grandes que 40 lignes : Simulation::lecture dans simulation.cc et lecture_fourmiere dans fourmilliere.cc	Warning: Evitez les noms de variables avec une seule lettre (excepté pour les compteurs explicites et locaux comme dans une boucle for), et essayez de nommer vos variable d'avantage en fonction de ce qu'elles représentent (par exemple appelez un pointeur sur une fourmi ant plutôt que pr)	Bon travail, le code est très bien structuré ! Attention aux nombreux noms de variables qui ne sont pas toujours clairs, ainsi qu'à garder vos styles de noms de fonctions cohérents (par exemple getGeneratrice vs get_collectors) et significatifs : par exemple vos fonctions d'affichage pourraient simplement s'appeler show() car, même si elles ont initialement été créées pour du debug, elles restent pertinentes pour un affichage de manière générale et une convention de nommage comme celle ci est plus perturbante qu'efficace, en témoigne votre commentaire nécessaire pour l'expliquer. Con courage pour la suite !
327096	3.00	[L1] squarecell.h 35-63, simulation.h 6, simulation.cc 37-99, fourmilere.cc 39-42	Double indentation dans toutes les classes (on n'indente pas les mots-clés "public" et "private"); indenter les "case"; plusieurs styles incohérents d'accolades pour les "if"	Dans l'ensemble très bon travail, il faut juste faire attention à l'indépendance de squarecell et aux conventions d'indentation.
327215	4.00	good		good initiative on using heritage and doing so properly, and great work overall

327221	3.00	[P5]poor choice of variable names: v,w in fourmilere.cc 83 84 110 111...	warning in fourmi.h 55 and fourmilere.cc 116 : wrapping lines (more than 87 chars)	Nicely done. Make sure the squarecell is next time fully independent
327283	4.00	OK	Good style.	Good coding style but be careful with your project conception. Squarecell should not be a class. The process class job could have been done by the simulation one.
327476	4.00	[L1] Incohérence dans les fichiers et dans les fonctions de tout le code	Incohérence des indentations dans les fichiers et dans les fonctions d'un même fichier.	Code clair et lisible, attention aux erreurs d'encapsulation et à l'indentation.
327538	4.00	OK		Attention avec les includes dans les différents fichiers, il faut respecter l'architecture du projet. Faire également attention à la fonction de décodage qui atteint la limite autorisée (82 caractères) . Autrement très bon code et style de code, à continuer pour les prochains rendus!
327683	3.00	[L1] inconsistency in the methods names squarecell.h 35, simulation.h 31		really good implementation, indentation is consistent though not always very readable fut to multilines if conditions especially in fourmilere.cc
327708	3.00	[L1] nonconsistent indentation when dealing with 1 execution line after an if statement squarecell.cc 76, simulation.cc 86		really good implementation and architecture, a few inconsistencies in the indentation but really good job overall
327732	3.00	[L1] simulation.h.14, simulation.cc.39, simulation.cc.61/138, squarecell.h.12-42	Il ne faut pas indenter les mots-clé de portée, il faut indenter les cases, il faut rester cohérent quand à l'utilisation ou non des {} (ici lorsqu'il y n'y à qu'une instruction), et il ne doit pas y avoir d'indentation par défaut pour les instructions hors de tout bloc (peut-être manque t-il la déclaration du namespace ?)	Bon travail, bonne structure et bonne division des taches à l'intérieur des modules. Continuez ainsi !
327815	3.00	[L1] anthill.h20-39	indent public/private fields	good structure and respected all style rules, other than one public/private
327821	4.00		Parfait	Code très clair et lisible. Très agréable à lire. Attention à l'architecture ou deux malencontreuses erreurs viennent se glisser. Dans l'ensemble, c'est vraiment excellent.
327964	4.00		Parfait	Code très agréable à lire, attention à l'encapsulation. L'ensemble est très lisible et clair.

328056	4.00	L1:fourmilier.cc:37		Attention à ne pas inclure constant.h dans squarecell, le module doit être indépendant du reste du projet. Sinon très bon code et style de code dans l'ensemble du projet, continuez comme ça pour les prochains rendus!
328162	3.00	[L2] simulation.cc 53,70,78,80,...	warning : a lot of wrapping lines (more than 87 char)	Good overall, just be careful next time in making squarecell independent and respecting the number of chars per line
328194	1.00	[L1]anthill.h16,28,30 anthill.cc 28-36 [L2] anthill.cc 32,68 simulation.cc 8,124,125 [P1] squarecell.cc superposition_carre (80l), simulation.cc decodage_ligne (51l)	Only write one instruction by line (e.g not like simulation.cc 8). Only one function can be more than 40 lines (and under 80).	Good code but be careful with your style.
328239	4.00		Style excellent !	attention à laisser l'implémentation des méthodes aux fichiers .cc.
328245	3.00	[L1]squarecell.cc 19,35,50,81	warnings : les noms de classes devraient commencer par une majuscule (nourriture, simulation); le style d'accolades n'est pas le même pour tous les types d'instructions de contrôle. Style d'accolades inconsistant avec le reste du code pour les fonctions dans squarecell.cc.	Bon travail travail dans l'ensemble, attention aux quelques conventions de style et au respect de l'architecture. Petit warning : on refabrique l'exécutable après le rendu, il n'est pas à inclure dans le .zip.
328331	4.00	Bien	warning: [L2] fourmi.cc 21 et squarcell.h 32	Bravo le code est complet mais vous n'avez pas respecté l'achitecture du projet. Le module simulation supervise tout les modules
328354	3.00	[L1]simulation.cc.55,90,99, 107 une tabulation en trop	Warning: dans le module squarecell, déclarez et initialisez plutôt votre grille dans le .h Warning [L1]: restez cohérents dans vos tyles concernant les accolades après un if/else Warning [L2]: simulation.cc.62 possède plus de 87 caractères	Très bon travail, ainsi que très bonne décomposition en fonctions. Toutefois, essayez de soigner d'avantage votre code (indenter selon la norme, mettre des espaces entre les opérateurs mathématiques, etc..), et faites attentions aux erreurs de programmation comme les header guards oubliés dans vos fichiers .h, ou la gestion d'erreur incomplète dans votre main. Bon courage pour la suite !
328436	3.00	L1:squarecell:18-56,174- 175,190,195,201-210, ... L2:squarecell:219,223	warning: aligner public et private à gauche.	Attention à respecter l'indentation en général et à ne pas indenter les public, private et protected dans les classes. Il faut corriger ces problèmes pour les prochains rendus et continuer avec une bonne base pour la suite.

328686	1.00	caractères au lieu de 4, indentation des mots-clés de portés qui ne devraient pas être indentés [L2] simulation.cc.87.93.99.107.18 0.etc... plus de 87 caractère sur une même ligne (activez sur geany l'affichage d'un marquage pour la limite de 87 caractères) [P2] les fonctions simulation::decodage_ligne (simulation.cc) et superposition (squarecell.cc) font toutes les deux plus de 40 lignes	Warning: squarecell.h déclarez plutôt grille dans squarecell.cc	Bon travail, essayez néanmoins de modulariser d'avantage votre code, notamment en déléguant la lecture des champs dont vous connaissez la nature à l'objet en question (par exemple une ligne permettant de créer une collectrice devrait être interprétée par la class Collector et non par Simulation), ainsi que d'utilisez des noms plus clairs lorsque vous nommez vos variables (par exemple x plutôt que ptx dans squarecell.19, ou collectors plutôt que vc dans Fourmillière.13). Bon courage pour la suite !
328879	3.00	[L1] simulation.cc.33, fourmilliere.cc.48	[L1] Il faut indenter les cases d'un switch	Très bon travail, votre code est clair, lisible et possède une très bonne séparation des tâches à l'intérieur des modules : continuez comme ça !
329038	4.00	warning [L1] simulation.cc 40-96, warning [L2] fourmilliere.cc 43	Indenter les "case"	Bon travail sur la structure et le style du projet. Cependant, il est nécessaire d'utiliser des classes pour les objets qui ne sont pas des types concrets de squarecell, afin d'encapsuler leurs attributs et d'avoir un contrôle sur leur utilisation.
329093	4.00	Bravo	Rien à dire	Impécable! Très lisible, commentaires pertinents, bonne implémentation des dépendances, rien à dire. Bravo
329098	3.00	[L1] inconsistency in the files names squarecell.cc , Simulation.cc		really good work overall, two points loss because of small mistakes but the implementation is fine apart from nourriture
329113	2.00	[L1] simulation.cc 44-45, 50-52, 57-59, fourmilliere.cc 38-73, 84-85, 91-92, [L2] fourmi.cc 23,25,29, simulation.cc 88, fourmilliere.cc 91	Style d'accolades des constructeurs différent du reste du code	Bon travail au niveau de l'architecture du code, mais faites attention à externaliser les constructeurs, et à être plus attentifs aux conventions de style.

329309	3.00	[P2]simulation35:170	max # of lines was 80, this method is 140 lines long	all the structure and formatting rules are respected, just break up your method into smaller traitements
329311	4.00			
329505	0.00			
329648	1.00	[L1] squarecells.cc 129,134,141,144,147 [L2] squarecells.cc 43,48,60,68,85,103 [P2]simulation.cc decodage_ligne(99 lines)	The indentation is inconsistent, sometimes 2 spaces (e.g fourmi) sometimes 4 (e.g simulation). Use same brace style everywhere.	Be careful with your coding style, it should be consistent everywhere.
329756	3.00	[L1]simulation.cc.55,90,99, 107 une tabulation en trop	Warning: dans le module squarecell, déclarez et initialisez plutôt votre grille dans le .h Warning [L1]: restez cohérents dans vos tyles concernant les accolades après un if/else Warning [L2]: simulation.cc.62 possède plus de 87 caractères	Très bon travail, ainsi que très bonne décomposition en fonctions. Toutefois, essayez de soigner d'avantage votre code (indenter selon la norme, mettre des espaces entre les opérateurs mathématiques, etc..), et faites attentions aux erreurs de programmation comme les header guards oubliés dans vos fichiers .h, ou la gestion d'erreur incomplète dans votre main. Bon courage pour la suite !
329782	2.00	[L1] squarcell.cc L25-L35 et fourmi.cc L57-L77 [L2] fourmi.cc fourmi.h fourmiellere.cc ...		Bon travail! Dommage que vous avez perdu des points sur le style du code. Il faut toujours respecter les principes de programmation. Il faut aussi respecter les indentations et le nombre de caractères par lignes pour les prochains rendus
329910	4.00		Le style est très propre et cohérent sur l'ensemble du code.	Attention à bien respecter l'architecture pour les modules projet et squarecell, le reste est très bien structuré. Bonne utilisation de tableaux de pointeurs pour stocker les fourmis. Le code est bien indenté et agréable à lire. Petit warning : on refabrique l'exécutable après le rendu, il n'est pas à inclure dans le .zip.
329920	3.00	[L1] nonconsistent indentation when dealing with 1 execution line after an if statement squarecell.cc 76, simulation.cc 86		really good implementation and architecture, a few inconsistencies in the indentation but really good job overall
330043	4.00		Attention, la structure square devrait commencer avec une majuscule (Square).	Très bon !

330070	4.00	OK		Attention avec les includes dans les différents fichiers, il faut respecter l'architecture du projet. Faire également attention à la fonction de décodage qui atteint la limite autorisée (82 caractères) . Autrement très bon code et style de code, à continuer pour les prochains rendus!
330101	4.00	[L2]fourmilier.cc40	Attention au line wrapping	Attention à l'encapsulation et à l'architecture. Code sinon clair et lisible.
330186	2.00	[L2]simulation.cc122,124,139,87,76 [L2]fourmilier.h35,37,39 [L1]simulation.cc87,92,97 [L1]simulation.h23,25	Attention aux erreurs de style	Code peu documenté, difficile à lire. Favoriser un code plus aérer. Attention aux erreurs de style.
330640	2.00	[L1] inconsistency in the files names Food.cc, simulation.cc [L1] indentation inconsistency fourmilier.h 25, 38	warning: please match file name and class file	good implementation an good respect of architecture, please make sure to clean your indentation before you hand in your work
330667	4.00	[L2]Warning fourmi.h 38	Good style.	Good code. You should not use a class for squarecell.
330833	1.00	[L1] squarecells.cc 129,134,141,144,147 [L2]squarecells.cc 43,48,60,68,85,103 [P2]simulation.cc decodage_ligne(99 lines)	The indentation is inconsistent, sometimes 2 spaces (e.g fourmi) sometimes 4 (e.g simulation). Use same brace style everywhere.	Be careful with your coding style, it should be consistent everywhere.
331471	4.00		Le style est très bon mais il faut faire attention à ce que les expressions écrivent sur plusieurs lignes soient correctement alignées et que le code soit facilement lisible. Il ne faut pas non plus hésiter à utiliser des switch afin d'éviter de mettre trop de if ... else.	Le code et le style sont très bon mais il faut faire attention à ne pas trop se compliquer la tâche.
332230	3.00	[L1] squarecell.cc 30-31, simulation.cc 60-90, nourriture.cc 16-22, fourmilier.cc 19-22, fourmi.cc 24-32,36-43,47-53,57-64	Indenter chaque "case"; plusieurs styles incohérents d'accolades pour les "if"; on n'indente pas les mots-clés "public" et "private"	Très bon travail de manière générale, il faut juste faire attention aux conventions d'indentation.

339389	4.00	ok	<p>WARNING: [L1] indetation constructeur fourmilier</p> <p>Warning: aligner public, private à gauche</p> <p>Warning: aucun nom dans le prototype du constructeur</p> <p>Warning: dans fourmi.cc double indentation 193-199</p> <p>Warning: std::vector<Fourmilier> anthill_data dans le .h</p>	<p>Votre code est bon! Bravo!</p> <p>Par contre attention il fortement déconseillé d'initialiser un static std::vector<Fourmilier> anthill_data; dans le .h car alors vous risquez de perdre le contrôle de votre vector à chaque fois que vous incluez votre header. Pour que cela soit juste, vous devez l'initialiser dans le .cc ansi votre tableau sera protégé.</p> <p>Vous avez la mauvaise habitude de ne pas donner de nom aux attributs dans les constructeurs dans le .h mais d'en mettre dans le .cc. Je comprends que vous vous êtes dit que le compilateur ne verra pas la différence mais ce n'est pas comme ça qu'il faut réfléchir. Il faut vous dire que n'importe qui va lire votre code et ne vas jamais regarder le .cc mais seulement le .h et la personne doit comprendre au premier coup d'oeil à quoi sert la fonction et ce qu'il faut mettre dedans</p> <p>Bonne utilisation du #ifdef!</p> <p>Bonne utilisation des commentaires!</p>
339407	4.00	OK	Good style.	Excellent code. Do a make clean before sending your code.
339421	4.00	OK	Good style.	Excellent code. Do a make clean before sending your code.
339442	3.00	L1:simulation.h:25,34 L2:simulation.cc:77,176,177,182,185,...	<p>warning: aligner public et private à gauche.</p> <p>warning: ne pas dépasser les 87 caractères par ligne (les commentaires comptent aussi)</p>	<p>Attention à votre simulation qui est créée dans la fonction lecture, mais qui sera détruite dès que la fonction se terminera. Peut-être créer l'object simulation dans le main et appeler une méthode de la classe pour la lecture. Les fourmilières et la nourriture ne sont pas ajoutés à la simulation non plus.</p> <p>Sans compter les dépassements de caractères par ligne à corriger, très bon style! A garder pour les prochains rendus.</p>
339450	4.00	OK	Good style.	Very good code. You could use a struct instead of the Square class. When you have if(condition)return true ; else return false ; just make return condition ;
339458	3.00	[L1] nonconsistent indentation when dealing with 1 execution line after an if statement nourriture.cc 30, 48		really good implementation and architecture, a few inconsistencies in the indentation but really good job overall

339621	1.00	[L2]: simulation.cc 120, 160, 173, 186, 200, 260 error.h [P2] lecture::separation 98 lignes [P5]: simulation.cc 139-1489, 160, 173, 186 magic number squarecell.ccc	WARNING: [L1] nourriture.cc indentation squarcell.cc 158-162 Warning: debuggage dans simulation.cc 58-61, 236, dans fourmilere.cc 18 Warning: indentation fourmilere.h 56 Warning: depassement ligne commentaire squarecell.h, squarecell.cc Warning: Squarecell.h inconsistance entre la definition du prototype et commentaire de la fonction 35-36	Le code a du potentiel. Le problème vient du fait qu'il est un peu "mal rangé", ce qui est dommage parce que l'on sent que vous avez beaucoup codé et que vous vous êtes investit. Ranger le code est une étape importante, elle vous permettra d'une part de faire une pause dans votre travail et aussi vous aurez les idées plus claire pour la suite. J'ai vu que vous avez pris l'habitude de commenter vos lignes de code, ce qui est une très bonne habitude. C'est mieux, si vous avez de long commentaire de faire un saut à la ligne que d'écrire à la suite de la ligne de code (comme vous avez fait dans Squarecell). Je pense qu'au prochain rendu, en rangeant bien, vous ne perderez aucun point de style. Courage!
339646	4.00	good	project should only connect to the rest of the code through simulation	good tabbing, followed file structure except project.cc
339649	4.00	good		Great work !!
339712	3.00	[L2] simulation.cc 53,70,78,80,...	warning : a lot of wrapping lines (more than 87 char)	Good overall, just be careful next time in making squarecell independent and respecting the number of chars per line
339870	4.00	OK		Excellent travail, code très bien structuré et style irréprochable. Seule remarque: de manière générale, les headers standards (iostream, vector, cmath, etc.) devraient être inclus entre <> et pas entre "".
339943	4.00		warning in sim.cc L34,68 wrapping lines (more than 87 char), and allign private and public left	well done. I have however a remark on a specific choice of implementation that you should maybe revisit which is choosing Square as a superclass for Anthill
340423	3.00	[L2] squarecell.cc 78,81,84,87 [L2]fourmis.h26,44,54,64	4 line wrappings	Excellent code, attention au line wrapping. Du reste, tout est parfait !
340645	4.00	warning [L1] squarecell.cc 27, 70, warning [L2] squarecell.cc 23,109-110		Très bon travail sur la structure et l'architecture, il faut cependant faire attention à l'externalisation des constructeurs et aux variables "static" dans les .h
340769	3.00	[P2] check_ant_overlaps is too long all_objects.cc 109	warning: please avoid ant in your methods names in squarecell since squarecell is considered as a library and should not be related to the rest of the Tchanz project	architecture should be reconsidered but the code is good otherwise, perfect implementation makes it really enjoyable to read

340803	3.00	[L1] fourmilliere.cc.58-61, simulation.cc.90-117	[L1] il faut indenter les cases des switch ainsi que les instructions dans les cases	Très bon travail! Vous avez très bien structuré vos modules et réparti leurs taches. Vous pourriez toutefois grandement améliorer la lisibilité de votre code en fusionnant les conditions qui le peuvent avec des opérations booléennes (par exemple dans squarecell.cc, dans la fonction is_at_distance_of_limit vous pourriez tout synthétiser en un seul return avec des opérations booléennes, ou dans is_perimeter_of_big_square_occupied les boucles for et leurs expressions if peuvent également être fusionnés), et en ajoutant des espaces pour aérer vos expressions (typiquement en ajoutant des espaces entre les opérateurs arithmétiques). Continuez comme ça !
340814	3.00	[L1] simulation.cc.80, anthill.cc.84, ant.cc.42 Warning [P5] simulation.cc.48, squarecell.cc.110	[L1] Les instructions case doivent être indentées Warning [P5]: créez une constante pour le nombre maximale d'aguments (2), ainsi que pour la taille maximale (128)	Bon travail et bonne structure du code en général. Quelques petites remarques : vous pouvez également return EXIT_SUCCESS dans votre main, plutôt que 0, vous avez plusieurs break inutiles car placés après des return (voir par exemple decode_line dans simulation.cc) et std::__throw_invalid_argument n'est pas une fonction standard, remplacez là par une exception de <exception>. Bonne continuation !
340840	4.00	good		excellent
340864	4.00		Parfait	Excellent code, félicitations toutes particulières pour la documentation qui est vraiment maîtrisée. Code très très agréable à lire.
340932	4.00		Parfait	Attention à l'encapsulation. Attention également à l'usage abusif de variables "static". Code clair et lisible.
340933	3.00	[L2]squarecell.cc78,81,84,87 [L2]fourmis.h26,44,54,64	4 line wrappings	Excellent code, attention au line wrapping. Du reste, tout est parfait !
340944	4.00		BoolX in squarecell is good but please avoid it in situations like these since it makes for an unconventional indentation	excellent implementation and very enjoyable code to read due to a well though out indentation

340949	2.00	[L2]fourmiliere.cc15-46 squarcecell.cc 55 [P5]simulation.cc 52-57 magic numbers	WARNING: fourmiliere.h externaliser les methodes. [C2] Warning: fourmi.h public private indenter à gauche	Le code est bon, bravo! Le code est agréable à lire (noms cohérents etc) Attention vos lignes sont souvent trop longues. N'hésitez pas à faire des sauts à la ligne (après un "and" par exemple ou à faire plusieurs petites fonctions (le prototype de ces fonctions ne doit pas nécessairement apparaître dans le .h cela risquerait de polluer inutilement votre header mais ce ne serait pas une faute.) Attention aussi aux magic numbers comme "6" ou "9". Vous pouvez déclarer un #define INFO_ANTHILL 9 par exemple). (commentaire supplémentaire: c'est mieux de n'inclure les header seulement là ou vous en avez besoin (par exemple si vous n'utilisez constante.h que dans le .cc c'est mieux de l'inclure que dans le .cc. Inclure un header dans le .h (par exemple nourriture.h) implique que quand vous appeler ce .h là, cela inclu aussi le .h (constante.h). Et donc il y a des moment où votre code marche "par hasard" "grâce"/à cause de ces dépendances (par exemple vous n'incluez pas constante.h dans fourmi alors qu'elle en a besoin mais comme nourriture.h est inclu dans fourmi, fourmi récupère aussi constante.h. De même avec using namespace std (d'ailleurs en utilisant using namespace std vous perdez le contrôl sur certain nom par exemple vous ne pouvez plus appeler une variable "size"). Cependant ceci n'est pas pénalisé et ce n'est pas une faute. C'est simplement un commentaire pour la suite de vos études)
340952	3.00	[L2]fourmiliere.cc15,28,110,11 5,143	Plusieurs lignes de fourmiliere.cc dépassent les 87 colonnes et attention à ne pas non plus dépasser la limite dans la documentation en haut du code.	Mis à part quelques fautes d'inattention le code est très bien fait !
340991	4.00		Parfait	Code très clair et lisible, agréable à lire.
340992	3.00	[L1]simulation.cc32-87	la méthode de lecture est pile à la limite de longueur autorisé "grâce" à un style trop compact pour les accolades fermantes	Beau travail : code très cohérent et rigoureux. Style clair mais parfois trop dense comme la méthode de lecture. val_food étant une constante il n'est pas nécessaire d'en faire un attribut d'un élément de nourriture ; pourquoi l'age est-il dans toute les classes dérivées au lieu d'être dans la superclasse ?

340997	2.00	[L2]anthill.cc 22-23,43,72,80,90,96,105,anthill.h25,fourmi.cc70,73,83,119,126,132,nourriture.cc42,simulation.cc24,97,squarecell.cc11,[P2]simulation.cc49-172	Attention beaucoup de lignes sont trop longues ! La fonction decodage_ligne dépasse aussi largement la limite de 80 lignes. Les noms des différentes fonctions pourraient être plus descriptifs. Mis à part cela, le style du code est très bon.	Le code est très bien fait et le style est bon. Il faut juste faire attention aux quelques fautes d'inattention qui se sont glissées dans le projet.
341008	3.00	[L1]simulation.cc 63,67,72,78	Style d'accolades inconsistent sur l'ensemble du code, particulièrement pour les if/else (simulation.cc, fourmilere.cc,squarecell.cc). Même chose pour l'indentation des switch (simulation.cc, fourmilere.cc).	Le style général est bon, la structure du code est claire. Bonne utilisation de namespaces, bonne structuration du projet.
341015	4.00	L1:fourmilere.cc:37		Attention à ne pas inclure constant.h dans squarecell, le module doit être indépendant du reste du projet. Sinon très bon code et style de code dans l'ensemble du projet, continuez comme ça pour les prochains rendus!
341021	4.00	good		excellent
341042	4.00	good	main should only call one or two functions; squarecell shouldn't depend on other functions, global variables in simulation.cc	Overall good tabbing and file structure, just main should only be calling one function and squarecell should be independent of the project
341069	4.00		Le style est très propre et cohérent sur l'ensemble du code.	Attention à bien respecter l'architecture pour les modules projet et squarecell, le reste est très bien structuré. Bonne utilisation de tableaux de pointeurs pour stocker les fourmis. Le code est bien indenté et agréable à lire. Petit warning : on refabrique l'exécutable après le rendu, il n'est pas à inclure dans le .zip.

341115	1.00	<p>caractères au lieu de 4, indentation des mots-clés de portés qui ne devraient pas être indentés</p> <p>[L2] simulation.cc.87.93.99.107.180.etc... plus de 87 caractère sur une même ligne (activez sur geany l'affichage d'un marquage pour la limite de 87 caractères)</p> <p>[P2] les fonctions simulation::decodage_ligne (simulation.cc) et superposition (squarecell.cc) font toutes les deux plus de 40 lignes</p>	<p>Warning: squarecell.h déclarez plutôt grille dans squarecell.cc</p>	<p>Bon travail, essayez néanmoins de modulariser d'avantage votre code, notamment en déléguant la lecture des champs dont vous connaissez la nature à l'objet en question (par exemple une ligne permettant de créer une collectrice devrait être interprétée par la class Collector et non par Simulation), ainsi que d'utilisez des noms plus clairs lorsque vous nommez vos variables (par exemple x plutôt que ptx dans squarecell.19, ou collectors plutôt que vc dans Fourmillière.13).</p> <p>Bon courage pour la suite !</p>
341127	3.00	<p>[L1] simulation.cc.33, fourmilliere.cc.48</p>	<p>[L1] Il faut indenter les cases d'un switch</p>	<p>Très bon travail, votre code est clair, lisible et possède une très bonne séparation des tâches à l'intérieur des modules : continuez comme ça !</p>
341148	3.00	<p>L1:simulation.h:25,34 L2:simulation.cc:77,176,177,182,185,...</p>	<p>warning: aligner public et private à gauche. warning: ne pas dépasser les 87 caractères par ligne (les commentaires comptent aussi)</p>	<p>Attention à votre simulation qui est créée dans la fonction lecture, mais qui sera détruite dès que la fonction se terminera. Peut-être créer l'objet simulation dans le main et appeler une méthode de la classe pour la lecture. Les fourmilières et la nourriture ne sont pas ajoutés à la simulation non plus. Sans compter les dépassements de caractères par ligne à corriger, très bon style! A garder pour les prochains rendus.</p>
341178	3.00	<p>[L1] fourmi.cc 14,17,21,24,27, simulation.cc 77-90, 222-225, squarecell.cc 59, fourmilliere.cc 17,27-29,65</p>	<p>Warning: plusieurs styles d'accolades en fonction des instructions de contrôle (switch, if, etc.)</p>	<p>Très bon travail de manière générale, il faut juste faire attention aux conventions d'indentation.</p>
341181	4.00	<p>good</p>	<p>squarecell must be independent!</p>	<p>the rules set out are followed, including no global variables, good constant names, consistent tabbing. Just need to follow architecture of project in squarecell</p>
341189	4.00	<p>Bien</p>	<p>warning: [L2] fourmi.cc 21 et squarecell.h 32</p>	<p>Bravo le code est complet mais vous n'avez pas respecté l'architecture du projet. Le module simulation supervise tout les modules</p>

341215	2.00	[P2] fourmiliere.cc 118-164 method exceeds 40 lines[L1] simulation.cc 86,87,89 fourmiliere.cc 128,148,...	warning: revisit your indentation	Well done. Some indentations that you did with space rather than tab that should be rectified, then a method of more than 40 lines that unfortunately cost you 1 point. I would also suggest making a separate module for fourmi and not having it in the same one as fourmiliere. Otherwise, good work and good use of the Separation of concerns principle.
341270	3.00	[L2] squarcell.h L31/32/37 , squarcell.cc L26/34 , Fourmillere.cc L22	Warning: double indentation dans fourmillere.cc l22	Bon travail! Il faut faire attention aux conventions de programmation. Il faut aussi que vous ne dépasser pas le 87 caractères par ligne pour les prochains rendus
341273	4.00	OK	Good style.	Very good code. You could use a struct instead of the Square class. When you have if(condition)return true ; else return false ; just make return condition ;
341284	4.00	good	squarecell must be independent!	the rules set out are followed, including no global variables, good constant names, consistent tabbing. Just need to follow architecture of project in squarecell
341354	4.00	warning [L1] squarecell.cc 27, 70, warning [L2] squarecell.cc 23,109-110		Très bon travail sur la structure et l'architecture, il faut cependant faire attention à l'externalisation des constructeurs et aux variables "static" dans les .h
341363	3.00	[L1] squarecell.cc 81,82,94, nourriture.cc 19, fourmi.h 35- 37, fourmi.cc 28,32,51,74	Différents styles d'accolades pour les "if" et "for"	Bon travail, de manière générale bien structuré. Faites attention aux inclusions inutiles qui cassent l'architecture (fourmillière dans fourmi, message dans squarecell) et aux conventions d'indentation.

341372	2.00	<p>[L2]fourmiliere.cc15-46 suarcecell.cc 55 [P5]simulation.cc 52-57 magic numbers</p>	<p>WARNING: fourmiliere.h externaliser les methodes. [C2] Warning: fourmi.h public private indenter à gauche</p>	<p>Le code est bon, bravo! Le code est agréable à lire (noms cohérents etc) Attention vos lignes sont souvent trop longues. N'hésitez pas à faire des sauts à la ligne (après un "and" par exemple ou à faire plusieurs petites fonctions (le prototype de ces fonctions ne doit pas nécessairement apparaître dans le .h cela risquerait de polluer inutilement votre header mais ce ne serait pas une faute.) Attention aussi aux magic numbers comme "6" ou "9". Vous pouvez déclarer un #define INFO_ANTHILL 9 par exemple). (commentaire supplémentaire: c'est mieux de n'inclure les header seulement là ou vous en avez besoin (par exemple si vous n'utilisez constante.h que dans le .cc c'est mieux de l'inclure que dans le .cc. Inclure un header dans le .h (par exemple nourriture.h) implique que quand vous appeler ce .h là, cela inclu aussi le .h (constante.h). Et donc il y a des moment où votre code marche "par hasard" "grâce"/à cause de ces dépendances (par exemple vous n'incluez pas constante.h dans fourmi alors qu'elle en a besoin mais comme nourriture.h est inclu dans fourmi, fourmi récupère aussi constante.h. De même avec using namespace std (d'ailleurs en utilisant using namespace std vous perdez le contrôl sur certain nom par exemple vous ne pouvez plus appeler une variable "size"). Cependant ceci n'est pas pénalisé et ce n'est pas une faute. C'est simplement un commentaire pour la suite de vos études)</p>
341400	2.00	<p>[P2] simulation.cc fonction de lecture < 80 lignes [P5] magic number dans fourmi.cc et squarecell.cc "1000"</p>	<p>Warning: dans la fonction de lecture il y a des accolades inutiles</p>	<p>Le code est bon! Bravo pour avoir bien respecté les 87 caractères, d'avoir fait des sauts à la lignes de manière élégante. Le code est très cohérent, lisible. Quel dommage que vous ailliez dépassé les 80 lignes de la fonction de lecture, mais je pense que aviez fait du debuggage et vous n'avez pas enlevé les accolades. Sans cela vous n'auriez pas dépassé les lignes. Attention aux magic number car à la lecture du code on doit comprendre du premier coup d'oeil sans avoir besoin de lire le code pour comprendre ce que c'est.</p>
341439	2.00	<p>[L1] squarcell.cc L25-L35 et fourmi.cc L57-L77 [L2] fourmi.cc fourmi.h fourmiellere.cc ...</p>		<p>Bon travail ! Dommage que vous avez perdu des points sur le style du code. Il faut toujours respecter les principes de programmation. Il faut aussi respecter les indentations et le nombre de caractères par lignes pour les prochains rendus</p>

341450	3.00	[L2] nourriture.cc 27,39,62 squarecell.cc 16		Clean code. Be careful to only include the necessary.
341451	3.00	[L1] squarecell.cc 11,59, nourriture.h 21, nourriture.cc 9-10, fourmilier.h 47,54, fourmi.h 24		Dans l'ensemble très bon travail; il faut cependant faire attention à l'indépendance de squarecell, aux variables globales dans un .cc (à déclarer avec "static" ou dans le namespace non-nommé), et à l'indentation
341551	4.00	L1:fourmi.h:11,15,22,29,32,38, ...		Attention à l'alignement à gauche des public, private et protected dans les classes. Autrement, très bon code et style dans le projet, à continuer pour la suite des rendus.
341622	2.00	[L1]simulation.cc 29, fourmilier.h 16, 24, fourmi.cc 47, [L2] fourmilier.h 14, fourmi.cc 22, 37, squarecell.cc 40, 101	Double indentation de plusieurs blocs. Plusieurs lignes dépassent la longueur max de 87 caractères. warning : dans simulation.cc, oublie d'indenter le corps des case (34).	Le projet est bien structuré et les concepts sont bien implémentés. Attention au style!
341670	4.00	good		good initiative on using heritage and doing so properly, and great work overall
341687	3.00	[L1]squarecell.cc 19,35,50,81	warnings : les noms de classes devraient commencer par une majuscule (nourriture, simulation); le style d'accolades n'est pas le même pour tous les types d'instructions de contrôle. Style d'accolades inconsistant avec le reste du code pour les fonctions dans squarecell.cc.	Bon travail travail dans l'ensemble, attention aux quelques conventions de style et au respect de l'architecture. Petit warning : on refabrique l'exécutable après le rendu, il n'est pas à inclure dans le .zip.
341739	4.00		Le style est bon et cohérent sur l'ensemble du code. Le nom de type Renvoie n'est pas très explicite.	Bonne structure en général. Les commentaires sont bien placés et clairs. Bonne implémentation du principe de réutilisation.
341899	3.00	[L1] fourmillier.h.11,23, fourmi.h.10,20,23,28,..., fourmillier.cc.67/80 Warning [L2] fourmilier.cc et squarecell.cc.153 ont plus de 87 caractères	[L1] Il ne faut pas indenter les mots-clé de portée et il faut rester cohérent lorsque vous utilisez les {} (ici un cas où vous allez à la ligne, et un cas où vous restez sur la même ligne)	Bon travail ! Voici quelques pistes pour l'améliorer : passez les objets par références constantes (typiquement les vectors), ou références si vous avez besoin de les modifier, à moins de vouloir expressément les copier (c'est rarement le cas dans votre code). (if (expr) return true; else return false;) peut être synthétisé pour simplement donner (return expr;) ce qui rends votre code plus clair. Parfois vous pourriez aérer vos blocs de code avec des sauts de ligne et des espaces dans vos expressions pour les rendre plus agréables à lire. Bon courage pour la suite !

341931	3.00	[L1]private/public keywords [L2]Warning fourmilere.cc 53,62		Good code. Your function <code>int_grid</code> is useless, you can simply initiate your grid as <code>grid(g_max, (std::vector<bool>(g_max, false))</code>). When you have <code>if(condition)</code> return <code>true</code> ; else return <code>false</code> ; simply do <code>return condition</code> ; (e.g <code>simulation.cc 35</code>).
341951	2.00	[L1] Simulation.h, Fourmillière.h, etc... - les mots clés de portée de doivent pas être indentés (il doit donc y avoir une indentation de moins pour le reste du code affecté par ces portées) [L2]: squarecell.h.28, squarecell.cc.101, fourmillière.h.25 et fourmillière.cc.9,35,37, fourmi.cc... contiennent plus de 87 caractères	Warning [P5] 1) : Il manque un symbole pour la vie initiale de la fourmie génératrice Warning: Dans <code>nourriture.cc</code> et <code>fourmillière.cc</code> , le nom de variable <code>o</code> , et dans <code>simulation::décodage</code> , les noms de variables <code>i</code> et <code>j</code> manquent de clareté Warning: Il faut rester cohérent vis-à-vis du retour à la ligne pour un instruction unique suivant un <code>if</code> ou un <code>else</code>	Bon travail ! Essayez d'ahérer un peu plus votre code en ajoutant des espaces dans vos expressions (par exemple après chaque virgule dans la liste d'arguments passés à une fonction), de supprimer les retours à la ligne en trop à divers endroits dans votre code, et de rendre certains de vos noms de variable plus explicites (par exemple <code>val[0/1/2/.../5]</code> dans <code>simulation::decodage_ligne</code> pourrait être plus explicite). Continuez comme ça!
341953	3.00	[L1]fourmi.cc 22,38,54,70	Attention à avoir un style d'accolades cohérent et à éviter la double indentation. warnings : le nom des struct devraient commencer par une majuscule (<code>square</code>), on n'indente pas les mots-clefs " <code>public</code> " et " <code>private</code> " dans la déclaration des classes, et les noms des paramètres des fonctions devraient être en minuscules (<code>fourmilere.h</code>).	Bonne structure du projet, attention aux conventions de style. Dans <code>squarecell</code> , les fonction calculant la valeur absolue et le max redéfinissent des fonctions existantes. Bon usage des constructeurs.
341975	4.00	good	squarecell must be independent!	respected style rules, but <code>squarecell</code> must be independent of the rest of the program (it seems that it even is if the includes are removed)
342052	4.00	OK	Good style.	Excellent code. Maybe use a struct to represent squares, it will make your code lighter.

342054	3.00	[P2] simulation.cc fonction lecture > 80 lignes	<p>WARNING: [L1] mauvaise indentation, mais consistante, dans projet.cc, fourmilere.cc (c'est une tabulation = 4 espace, ici il y en a que 2) Warning simulation.cc il manque le suffixe "static" à vos variables globales static (vos vector) (Warning simulation.cc double ";" 128) Warning [L2] fourmilere.h dépacement 19</p> <p>WARNING: [C1] squarecel.h dans la class Carre definition n'attribut public l53, 54, 55, 56, 57, 58, ok pour squarcell uniquement, sinon construire une structure</p>	Le code est plutôt bon. Faîtes attention à la cohérence de votre code. La plupart du temps c'est bien indenté puis subitement non. Je pense que pour Squarecell, au départ vous vouliez faire une structure et si cela avait été le cas, votre implémentation aurait été correcte.
342063	3.00	[L1]private/public keywords [L2]Warning fourmilere.cc 53,62		Good code. Your function int_grid is useless, you can simply initiate your grid as grid(g_max, (std::vector<bool>(g_max, false))). When you have if(conditon) return true ; else return false ; simply do return condition ; (e.g simulation.cc 35).
342154	3.00	[L1]fourmi.cc 22,38,54,70	Attention à avoir un style d'accolades cohérent et à éviter la double indentation. warnings : le nom des struct devraient commencer par une majuscule (square), on n'indente pas les mots-clefs "public" et "private" dans la déclaration des classes, et les noms des paramètres des fonctions devraient être en minuscules (fourmilere.h).	Bonne structure du projet, attention aux conventions de style. Dans squarecell, les fonction calculant la valeur absolue et le max redéfinissent des fonctions existantes. Bon usage des constructeurs.

342200	3.00	[L1] fourmilliere.h.11,23, fourmi.h.10,20,23,28,..., fourmilliere.cc.67/80 Warning [L2] fourmilliere.cc et squarecell.cc.153 ont plus de 87 caractères	[L1] Il ne faut pas indenter les mots-clé de portée et il faut rester cohérent lorsque vous utilisez les {} (ici un cas où vous allez à la ligne, et un cas où vous restez sur la même ligne)	Bon travail ! Voici quelques pistes pour l'améliorer : passez les objets par références constantes (typiquement les vectors), ou références si vous avez besoin de les modifier, à moins de vouloir expressement les copier (c'est rarement le cas dans votre code). (if (expr) return true; else return false;) peut être synthétisé pour simplement donner (return expr;) ce qui rends votre code plus clair. Parfois vous pourriez aérer vos blocs de code avec des sauts de ligne et des espaces dans vos expressions pour les rendre plus agréables à lire. Bon courage pour la suite !
342204	4.00		Parfait	Code très clair et lisible, agréable à lire. Attention à l'encapsulation.
342215	1.00	[L1] fourmilliere.cc L40-101 et L106-129 [L2] squarecell.h L34-67 et squarecell.cc et simulation.cc .. [P2] simulation.cc L51-148 et fourmillière.cc L38-101		Il faut faire attention aux conventions de programmation. Vous avez raté beaucoup de points sur le style de code.
342216	4.00	L1:fourmi.h:11,15,22,29,32,38, ...	warning: aligner public et private à gauche.	Attention à l'alignement à gauche des public, private et protected dans les classes. Autrement, très bon code et style dans le projet, à continuer pour la suite des rendus.
342745	3.00	[L1] squarecell.cc 30-31, simulation.cc 60-90, nourriture.cc 16-22, fourmilliere.cc 19-22, fourmi.cc 24-32,36-43,47-53,57-64	Indenter chaque "case"; plusieurs styles incohérents d'accolades pour les "if"; on n'indente pas les mots-clés "public" et "private"	Très bon travail de manière générale, il faut juste faire attention aux conventions d'indentation.
342800	4.00	OK		Excellent travail, code très bien structuré et style irréprochable. Seule remarque: de manière générale, les headers standards (iostream, vector, cmath, etc.) devraient être inclus entre <> et pas entre "".
343266	2.00	[L1]simulation.cc 29, fourmilliere.h 16, 24, fourmi.cc 47, [L2] fourmilliere.h 14, fourmi.cc 22, 37, squarecell.cc 40, 101	Double indentation de plusieurs blocs. Plusieurs lignes dépassent la longueur max de 87 caractères. warning : dans simulation.cc, oubli d'indenter le corps des case (34).	Le projet est bien structuré et les concepts sont bien implémentés. Attention au style!

343725	3.00	[P2] simulation.cc fonction lecture > 80 lignes	<p>WARNING: [L1] mauvaise indentation, mais consistante, dans projet.cc, fourmilere.cc (c'est une tabulation = 4 espace, ici il y en a que 2)</p> <p>Warning simulation.cc il manque le suffixe "static" à vos variables globales static (vos vector)</p> <p>Warning simulation.cc double ";" 128</p> <p>Warning: [L2] fourmilere.h dépacement 19</p> <p>WARNING: [C1] squarecel.h dans la class Carre definition n'attribut public l53, 54, 55, 56, 57, 58, ok pour squarcell uniquement, sinon construire une structure</p>	Le code est plutôt bon. Faîtes attention à la cohérence de votre code. La plupart du temps c'est bien indenté puis subitement non. Je pense que pour Squarecell, au départ vous vouliez faire une structure et si cela avait été le cas, votre implémentation aurait été correcte.
343736	3.00	L1:squarecell:18-56,174-175,190,195,201-210,... L2:squarecell:219,223	warning: aligner public et private à gauche.	Attention à respecter l'indentation en général et à ne pas indenter les public, private et protected dans les classes. Il faut corriger ces problèmes pour les prochains rendus et continuer avec une bonne base pour la suite.
343876	3.00	[L2] fourmi.h 13,23,44, fourmi.cc 17,27, squarcell.cc 49,50,149,150,etc.	Les lignes ne doivent pas dépasser 87 caractères.	L'implémentation est très incomplète. Le style est majoritairement bon. Petit warning : les exécutable ne devraient pas être inclus dans le rendu.
344139	3.00	[L1]simulation.cc32-87	la méthode de lecture est pile à la limite de longueur autorisé "grâce" à un style trop compact pour les accolades fermantes	Beau travail : code très cohérent et rigoureux. Style clair mais parfois trop dense comme la méthode de lecture . val_food étant une constante il n'est pas nécessaire d'en faire un attribut d'un élément de nourriture ; pourquoi l'age est-il dans toute les classes dérivées au lieu d'être dans la superclasse ?
344193	4.00	warning [L1] simulation.cc 40-96, warning [L2] fourmilere.cc 43	Indenter les "case"	Bon travail sur la structure et le style du projet. Cependant, il est nécessaire d'utiliser des classes pour les objets qui ne sont pas des types concrets de squarecell, afin d'encapsuler leurs attributs et d'avoir un contrôle sur leur utilisation.
344276	3.00	[L2] fourmi.h 13,23,44, fourmi.cc 17,27, squarcell.cc 49,50,149,150,etc.	Les lignes ne doivent pas dépasser 87 caractères.	L'implémentation est très incomplète. Le style est majoritairement bon. Petit warning : les exécutable ne devraient pas être inclus dans le rendu.
344312	3.00	[L1] nonconsistent indentation when dealing with 1 execution line after an if statement nourriture.cc 30, 48		really good implementation and architecture, a few inconsistencies in the indentation but really good job overall

344385	4.00		Le style est bon et cohérent sur l'ensemble du code. Le nom de type Renvoi n'est pas très explicite.	Bonne structure en général. Les commentaires sont bien placés et clairs. Bonne implémentation du principe de réutilisation.
344399	2.00	[L1] il faut aligner public et private à gauche dans tous les modules [L2] squarcell.cc L30/34/38/42 et simulation.cc L38/48 ..		Malheureusement vous avez perdu tous les points concernant l'encapsulation et la modularisation du projet. Pour les prochains rendu, si vous ne pouvez pas implémenter le constructeur sur une seule ligne dans l'interface alors il faut l'implémenter dans le module .cc
344405	4.00		Attention dans anthill.cc , à plusieurs endroits l'indentation est incorrecte. Attention aussi au style choisi pour les accolades. Plusieurs lignes de squarecell.cc dépassent la limite de 87 colonnes.	Le code est très bien fait et le style est excellent. Il faut juste faire attention aux quelques fautes d'inattention qui se sont glissées dans le projet.
344415	4.00	good		Nice work! I'll give you however a warning for using a lot of static variables, Also you must check in the beginning (in the main) that you receive an argument (name of file). But good modularisation, and use of enums.
344463	4.00	good		Excellent! Very well documented, with good commentaries for methods and within the code. Good modularization and use of concepts treated in the course
344514	3.00	[L1] inconsistency in the methods names squarecell.h 35, simulation.h 31		really good implementation, indentation is consistent though not always very readable fut to multilines if conditions especially in fourmiliere.cc
344516	3.00	[L2]fourmiliere.cc15,28,110,115,143	Plusieurs lignes de fourmiliere.cc dépassent les 87 colonnes et attention à ne pas non plus dépasser la limite dans la documentation en haut du code.	Mis à part quelques fautes d'inattention le code est très bien fait !
344526	2.00	[L1] mauvaise indentation presque partout (c'est une tabulation = 4 espace, ici il y en a que 2) et de manière inconsistante [L2] fourmiliere.h 28, fourmiliere.cc 40, 48, 57, 66 fourmi.cc 21, 23, 27	Warning: mettre une tabulation lors du saut à la ligne dans la définition d'un constructeur dans le .cc Warning: indentation public, private à gauche	Le code est bon! Le code est lisible, il y a quelques commentaire (ce qui est une bonne habitude), il y a une bonne utilisation du polymorphisme. C'est important de prendre l'habitude de bien indenter (4 espaces ou une tabulation), non seulement pour la lisibilité mais aussi car pour la suite de vos études certains langages de programmation ne compile pas si l'indentation n'est pas respectée (Python)
344528	4.00		Style très bon !	Très bon !

344538	4.00		Attention, la structure square devrait commencer avec une majuscule (Square).	Très bon !
344700	3.00	[L2] nourriture.cc 27,39,62 squarecell.cc 16		Clean code. Be careful to only include the necessary.
344732	3.00	[L2] foumi.cc fourmillere.cc foumillere.h squarecell.c ..		Le code est complet. Dommage que vous avez perdu tous les points concernant l'externalisation des méthodes. Si vous ne pouvez pas implémenter le constructeur sur une seule ligne dans l'interface alors il faut l'implémenter dans le module .cc
344736	3.00	[L1] simulation.c.124, squarecell.cc.176, fourmilliere.h.17, Fourmi.h.18	[L1] Il faut indenter les cases, il y a une tabulation en trop, il ne faut pas indenter les mots-clé de portée	Bon travail, le code et sa structure sont très soignés. Pour gagner en place et en clareté, vous pourriez éviter de sauter une ligne au début de chaque fonction, Très bon travail, continuez ainsi !
344793	4.00		Pas d'erreur	Attention à l'encapsulation. Attention également à l'usage abusif de variables "static". Code clair et lisible.
344804	4.00	good	project should only connect to the rest of the code through simulation	good tabbing, followed file structure except project.cc
344822	4.00	good		Nice work! I'll give you however a warning for using a lot of static variables, Also you must check in the beginning (in the main) that you receive an argument (name of file). But good modularisation, and use of enums.
345020	1.00	[L2]: simulation.cc 120, 160, 173, 186, 200, 260 error.h [P2] lecture::separation 98 lignes [P5]: simulation.cc 139-1489, 160, 173, 186 magic number squarecell.cc	indentation squarcell.cc 158-162 Warning: indentation public, private à gauche Warning: debbugage dans simulation.cc 58-61, 236, dans fourmiliere.cc 18 Warning: indentation fourmilliere.h 56 Warning: depassement ligne commentaire squarecell.h, squarecell.cc Warning: Squarecell.h inconsistance entre la definition du prototype et commentaire de la fonction 35-36	Le code a du potentiel. Le problème vient du fait qu'il est un peu "mal rangé", ce qui est dommage parce que l'on sent que vous avez beaucoup codé et que vous vous êtes investit. Ranger le code est une étape importante, elle vous permettra d'une part de faire une pause dans votre travail et aussi vous aurez les idées plus claire pour la suite. J'ai vu que vous avez pris l'habitude de commenter vos lignes de code, ce qui est une très bonne habitude. C'est mieux, si vous avez de long commentaire de faire un saut à la ligne que d'écrire à la suite de la ligne de code (comme vous avez fait dans Squarecell). Je pense qu'au prochain rendu, en rangeant bien, vous ne perderez aucun point de style. Courage!
345219	4.00	good		Excellent! Very well documented, with good commentaries for methods and within the code. Good modularization and use of concepts treated in the course

345259	4.00		warning in sim.cc L34,68 wrapping lines (more than 87 char), and align private and public left	well done. I have however a remark on a specific choice of implementation that you should maybe revisit which is choosing Square as a superclass for Anthill
345261	4.00	[L2]fourmiliere.cc40	Attention au line wrapping	Attention à l'encapsulation et à l'architecture. Code sinon clair et lisible.
345286	4.00	[L1]sqaurecel.cc 108,116,125,81...	warning: revisit your indentation	Great work, just be careful about squarecell being completely independant
345298	3.00	[P5]poor choice of variable names: v,w in fourmiliere.cc 83 84 110 111...	warning in fourmi.h 55 and fourmiliere.cc 116 : wrapping lines (more than 87 chars)	Nicely done. Make sure the squarecell is next time fully independent
345337	3.00	[L1]simulation.cc46-51	warning: aligner public et private à gauche	le code est malheureusement très incomplet et visiblement il ne peut pas compiler. Plusieurs choix sont étranges comme de mémoriser argc et argv dans la classe simulation; squarecell est plus avancé et clair mais il manque la définition de plusieurs types.
345346	2.00	[L2]simulation.cc122,124,139,87,76 [L2]fourmiliere.h35,37,39 [L1]simulation.cc87,92,97 [L1]simulation.h23,25	Attention aux erreurs de style	Code peu documenté, difficile à lire. Favoriser un code plus aérer. Attention aux erreurs de style.
345352	1.00	[L1] fourmillere.cc L40-101 et L106-129 [L2] squarecell.h L34-67 et squarecell.cc et simulation.cc .. [P2] simulation.cc L51-148 et fourmillère.cc L38-101		Il faut faire attention aux conventions de programmation. Vous avez raté beaucoup de points sur le style de code.
345393	3.00	[L1] fourmis.cc 117, 139	warning: please avoid fourmiliere in your methods names in squarecell since squarecell is considered as a lbrary and should not be related to the rest of the Tchanz project	really good architecture and implementation, your code would benefit from a little cleaning (avoid commented out cout)

345420	4.00	[L1] fourmilieres.cc dépassement de ligne 31 squarecell.cc [P5] squarecell.cc fonction remplissage > 40 lignes	Warning: indentation simulation.cc 37-61 Warning: simulation.cc utilisation de LIGNE qui n'est déclaré nul part. Warning: si LIGNE est une variable, elle doit être écrite en minuscule Warning: le makefile n'est pas correctement rempli cela ne peut pas compiler Warning: simulation.cc pas d'utilisation du break dans le switch	le code est malheureusement très incomplet et visiblement il ne peut pas compiler (le makefile n'est pas conçu pour votre programmes et certaines variables ne sont pas initialisées
345431	4.00		Attention dans anthill.cc , à plusieurs endroits l'indentation est incorrecte. Attention aussi au style choisi pour les accolades. Plusieurs lignes de squarecell.cc dépassent la limite de 87 colonnes.	Le code est très bien fait et le style est excellent. Il faut juste faire attention aux quelques fautes d'inattention qui se sont glissées dans le projet.
345442	3.00	[L1] mauvaise indentation presque partout (c'est une tabulation = 4 espace, ici il y en a que 2) et de manière inconsistante [L2] fourmilieres.h 28, fourmilieres.cc 40, 48, 57, 66 fourmi.cc 21, 23, 27	Warning: mettre une tabulation lors du saut à la ligne dans la définition d'un constructeur dans le .cc Warning: indentation public, private à gauche	Le code est bon! Le code est lisible, il y a quelques commentaire (ce qui est une bonne habitude), il y a une bonne utilisation du polymorphisme. Attention à votre indentation! C'est important de prendre l'habitude de bien indenter (4 espaces ou une tabulation), non seulement pour la lisibilité mais aussi car pour la suite de vos études certains langages de programmation ne compile pas si l'indentation n'est pas respectée (Python)
345448	4.00	L1:fourmi.h:17-36,41-51,56-69, 74-86,91-103,...	warning: aligner public et private à gauche.	Attention à l'indentation dans les classes, ne pas indenter les public, private, protected et ne pas double indenter le reste du bloc. Très bon code et style dans le projet, à garder pour la suite!
345459	3.00	[L1] Incohérence dans les fichiers et dans les fonctions de tout le code	Incohérence des indentations dans les fichiers et dans les fonctions d'un même fichier.	Code clair et lisible, attention aux erreurs d'encapsulation et à l'indentation.
345489	3.00	[L1] fourmis.cc 117, 139	warning: please avoid fourmilieres in your methods names in squarecell since squarecell is considered as a library and should not be related to the rest of the Tchanz project	really good architecture and implementation, your code would benefit from a little cleaning (avoid commented out cout)

345491	3.00	[L1] simulation.h.14, simulation.cc.39, simulation.cc.61/138, squarecell.h.12-42	Il ne faut pas indenter les mots-clé de portée, il faut indenter les cases, il faut rester cohérent quand à l'utilisation ou non des {} (ici lorsqu'il y n'y à qu'une instruction), et il ne doit pas y avoir d'indentation par défaut pour les instructions hors de tout bloc (peut-être manque t-il la déclaration du namespace ?)	Bon travail, bonne structure et bonne division des taches à l'interieur des modules. Continuez ainsi !
345522	4.00	OK		Excellent travail, code très bien structuré et style irréprochable. Seule remarque: de manière générale, les headers standards (iostream, vector, cmath, etc.) devraient être inclus entre <> et pas entre "".
345577	3.00	[L2] foumi.cc fourmillere.cc foumillere.h squarecell.c ..		Le code est complet. Dommage que vous avez perdu tous les points concernant l'externalisation des méthodes. Si vous ne pouvez pas implémenter le constructeur sur une seule ligne dans l'interface alors il faut l'implémenter dans le module .cc
345618	4.00		BoolX in squarecell is good but please avoid it in situations like these since it makes for an unconventional indentation	excellent implementation and very enjoyable code to read due to a well though out indentation
345671	4.00		Le style est très bon mais il faut faire attention à ne pas indenter les mots clé "public", "private" et "protected" dans les classes.	Le code est le style sont très bons !
345712	2.00	[L1]private/public keywords, nourriture.h 14-19 [L2] squarecell.cc 101,102 simulation.cc 110, fourmi.h 28,35	The public/private keywords should not be indented, be careful to always use the same size of indentation.	Very good code. Be careful with your style for next time.
345716	3.00	[L1] anthill.h20-39	indent public/private fields	good structure and respected all style rules, other than one public/private
345765	3.00	[P2] check_ant_overlaps is too long all_objects.cc 109	warning: please avoid ant in your methods names in squarecell since squarecell is considered as a library and should not be related to the rest of the Tchanz project	architecture should be reconsidered but the code is good otherwise, perfect implementation makes it really enjoyable to read

345770	4.00	L1:projet.cc:19;fourmis.h:55;fo od.cc:27 ,31-39		Attention à ne pas inclure constant.h dans squarecell, le module doit être indépendant du reste du projet. Sinon très bon style de code dans le projet, attention à quelques petits problèmes d'indentation. Continuez comme ça pour la suite du projet!
345772	2.00	[L1]squarecell.cc 6,7,11,15,21, [P2]simulation.cc (decodage_ligne, m_mess_erreur) 217-411	Le style d'accolades n'est pas cohérent d'un module à l'autre. Les fonctions ne doivent pas dépasser la limite de 40 lignes, sauf une à max 80 lignes. warnings : on n'indente pas les mots-clés "public" et "private" dans la déclaration des classes; les valeurs que peut prendre l'argument fourni_type passé à generate_fourmi devraient être définies soit dans un enum, soit par des constantes nommées	Bonne structure et début d'implémentation de fonctionnalités utiles pour les rendus suivants. Attention à l'encapsulation et à la longueur de vos fonctions. Le style est propre mais incohérent entre les modules.
345785	4.00		Le style est très bon mais il faut faire attention à ne pas indenter les mots clés "public", "private" et "protected" dans les classes.	Le code et le style sont très bons !
345863	4.00	L1:fourmi.h:17-36,41-51,56-69, 74-86,91-103,...	warning: aligner public et private à gauche.	Attention à l'indentation dans les classes, ne pas indenter les public, private, protected et ne pas double indenter le reste du bloc. Très bon code et style dans le projet, à garder pour la suite!
345866	4.00		paramètres dans la définition du constructeur de Fourmiere (pour des raisons de documentation). J'évitais de mettre des accolades dans les cases d'un switch. A part cela le style est très bon.	Très bon !
345936	4.00	good		Good work. Unfortunately you lost a point for including constants in squarecell. So be aware that squarecell must be totally independent. Also, the use of global variables is forbidden. Some modules had commentaries, that's an excellent practice to get used to!
346039	4.00	good	main should only call one or two functions; squarecell shouldn't depend on other functions, global variables in simulation.cc	Overall good tabbing and file structure, just main should only be calling one function and squarecell should be independent of the project

346080	3.00	[L1] squarecell.cc 81,82,94, nourriture.cc 19, fourmi.h 35-37, fourmi.cc 28,32,51,74	Différents styles d'accolades pour les "if" et "for"	Bon travail, de manière générale bien structuré. Faites attention aux inclusions inutiles qui cassent l'architecture (fourmillière dans fourmi, message dans squarecell) et aux conventions d'indentation.
346151	4.00	L1:food.h:16-40; fourmiliere.cc:12-83 ; fourmiliere.cc:11-61 ; fourmis.cc:14-49 ;fourmis.h:14-35;... L2:fourmiliere.cc:35;fourmilier e.h:26;fourmis.h:69,99,112,12 2simulation.cc:46 P2:simulation.cc:decodage_lig ne	warning: aligner public et private à gauche.	Attention à l'indentation dans l'ensemble du projet. Pensez également à ne pas indenter les private, public et protected dans les classes. Il faut également raccourcir la fonction de décodage pour respecter les 80 lignes max par fonction. Dans les prototypes des méthodes de la classe il ne faut pas mettre l'implémentation, cela se fait dans les fichiers .cc. Bonne continuation pour les prochains rendus!
346183	3.00	[L1] fourmi.cc 14,17,21,24,27, simulation.cc 77-90, 222-225, squarecell.cc 59, fourmiliere.cc 17,27-29,65	Warning: plusieurs styles d'accolades en fonction des instructions de contrôle (switch, if, etc.)	Très bon travail de manière générale, il faut juste faire attention aux conventions d'indentation.
346220	2.00	[L1] simulation.cc 44-45, 50-52, 57-59, fourmiliere.cc 38-73, 84-85, 91-92, [L2] fourmi.cc 23,25,29, simulation.cc 88, fourmiliere.cc 91	Style d'accolades des constructeurs différent du reste du code	Bon travail au niveau de l'architecture du code, mais faites attention à externaliser les constructeurs, et à être plus attentifs aux conventions de style.
346235	2.00	[P2] fourmiliere.cc 118-164 method exceeds 40 lines[L1]simulation.cc 86,87,89 fourmiliere.cc 128,148,...	warning: revisit your indentation	Well done. Some indentations that you did with space rather than tab that should be rectified, then a method of more than 40 lines that unfortunately cost you 1 point. I would also suggest making a separate module for fourmi and not having it in the same one as fourmilliere. Otherwise, good work and good use of the Separation of concerns principle.
346247	4.00		Parfait	Code très clair et lisible, agréable à lire. Attention à l'encapsulation.
346301	3.00	L[2] squarecell.cc simulation.cc		Bon travail!. Votre code est complet, mais il faut faire attention au style de code. Pour les prochains rendus il ne faut plus implémenter le constructeur dans l'interface .h si son implémentation se fait sur plusieurs lignes.

346370	3.00	[L1] fourmilliere.cc.58-61, simulation.cc.90-117	[L1] il faut indenter les cases des switch ainsi que les instructions dans les cases	Très bon travail! Vous avez très bien structuré vos modules et réparti leurs tâches. Vous pourriez toutefois grandement améliorer la lisibilité de votre code en fusionnant les conditions qui le peuvent avec des opérations booléennes (par exemple dans squarecell.cc, dans la fonction is_at_distance_of_limit vous pourriez tout synthétiser en un seul return avec des opérations booléennes, ou dans is_perimeter_of_big_square_occupied les boucles for et leurs expressions if peuvent également être fusionnées), et en ajoutant des espaces pour aérer vos expressions (typiquement en ajoutant des espaces entre les opérateurs arithmétiques). Continuez comme ça !
346450	2.00	[L1] Indentation de public et private [L2] fourmi.cc L31 , fourmi.h L15/28/39 , fourmiellere.cc L12/19		Il faut faire attention aux conventions de programmation. Vous avez raté beaucoup de points sur le style de code. Pour les prochains rendus il faut faire attention à l'indentation de public et private et au nombre de caractères par ligne
346471	3.00	[L2] squarecell.cc simulation.cc		Bon travail!. Votre code est complet, mais il faut faire attention au style de code. Pour les prochains rendus il ne faut plus implémenter le constructeur dans l'interface .h si son implémentation se fait sur plusieurs lignes.
346514	3.00	[L2] squarcell.h L31/32/37 , squarcell.cc L26/34 , Fourmillere.cc L22	Warning: double indentation dans fourmillere.cc l22	Bon travail! Il faut faire attention aux conventions de programmation. Il faut aussi que vous ne dépasser pas le 87 caractères par ligne pour les prochains rendus

346544	4.00	ok	<p>WARNING: [L1] indetation constructeur fourmilere</p> <p>Warning: aligner public, private à gauche</p> <p>Warning: aucun nom dans le prototype du constructeur</p> <p>Warning: dans fourmi.cc double indentation 193-199</p> <p>Warning: std::vector<Fourmilere> anthill_data dans le .h</p>	<p>Votre code est bon! Bravo!</p> <p>Par contre attention il fortement déconseillé d'initialiser un static std::vector<Fourmilere> anthill_data; dans le .h car alors vous risquez de perdre le contrôle de votre vector à chaque fois que vous incluez votre header. Pour que cela soit juste, vous devez l'initialiser dans le .cc ansi votre tableau sera protégé.</p> <p>Vous avez la mauvaise habitude de ne pas donner de nom aux attributs dans les constructeurs dans le .h mais d'en mettre dans le .cc. Je comprends que vous vous êtes dit que le compilateur ne verra pas la différence mais ce n'est pas comme ça qu'il faut réfléchir. Il faut vous dire que n'importe qui va lire votre code et ne vas jamais regarder le .cc mais seulement le .h et la personne doit comprendre au premier coup d'oeil à quoi sert la fonction et ce qu'il faut mettre dedans</p> <p>Bonne utilisation du #ifdef!</p> <p>Bonne utilisation des commentaires!</p>
346561	4.00		Style très bon !	Très bon !
346565	3.00	[L2] modules fourmi.cc fourmi.h fourmillere.cc fourmillere.h	warning: double indentation dans squrcell.cc L 64 et 65	points sur l'externalisation des méthodes. Si vous voulez implémenter le constructeur dans l'interface il faut qu'il soit sur une seule ligne.
346576	4.00	OK	Good style.	Excellent code. Maybe use a struct to represent squares, it will make your code lighter.
346718	4.00	good		Good work. Unfortunately you lost a point for including constants in squarecell. So be aware that squarecell must be totally independent. Also, the use of global variables is forbidden. Some modules had commentaries, that's an excellent practice to get used to!
347035	4.00	good		Great work !!
347036	4.00		Parfait	Excellent code, félicitations toutes particulières pour la documentation qui est vraiment maîtrisée. Code très très agréable à lire.
347038	4.00	[L2]Warning fourmi.h 38	Good style.	Good code. You should not use a class for squarecell.

347056	3.00	[L1] simulation.cc.80, anthill.cc.84, ant.cc.42 Warning [P5] simulation.cc.48, squarecell.cc.110	[L1] Les instructions case doivent être indentées Warning [P5]: créez une constante pour le nombre maximale d'aguments (2), ainsi que pour la taille maximale (128)	Bon travail et bonne structure du code en général. Quelques petites remarques : vous pouvez également return EXIT_SUCCESS dans votre main, plutôt que 0, vous avez plusieurs break inutiles car placés après des return (voir par exemple decode_line dans simulation.cc) et std::_throw_invalid_argument n'est pas une fonction standard, remplacez là par une exception de <exception>. Bonne continuation !
347057	4.00	good	use of global variables is not permitted	proper tabbing, good file structure, just make sure all variables are private
347087	0.00			
347105	4.00		Parfait	Code très clair et lisible. Très agréable à lire. Attention à l'architecture ou deux malencontreuses erreurs viennent se glisser. Dans l'ensemble, c'est vraiment excellent.
347263	2.00	[L1] il faut aligner public et private à gauche dans tous les modules [L2] squarcell.cc L30/34/38/42 et simulation.cc L38/48 ..		Malheureusement vous avez perdu tous les points concernant l'encapsulation et la modularisation du projet. Pour les prochains rendus, si vous ne pouvez pas implémenter le constructeur sur une seule ligne dans l'interface alors il faut l'implémenter dans le module .cc
347321	3.00	[L1] simulation.cc 70, 81	warning: please avoid ant in your methods names in squarecell since squarecell is considered as a library and should not be related to the rest of the Tchanz project	really good implementation and respect of architecture, just some small indention mistakes but really good job overall
347328	2.00	[L1]squarecell.cc 6,7,11,15,21, [P2]simulation.cc (decodage_ligne, m_mess_erreur) 217-411	Le style d'accolades n'est pas cohérent d'un module à l'autre. Les fonctions ne doivent pas dépasser la limite de 40 lignes, sauf une à max 80 lignes. warnings : on n'indente pas les mots-clefs "public" et "private" dans la déclaration des classes; les valeurs que peut prendre l'argument fourni_type passé à generate_fourni devraient être définies soit dans un enum, soit par des constantes nommées	Bonne structure et début d'implémentation de fonctionnalités utiles pour les rendus suivants. Attention à l'encapsulation et à la longueur de vos fonctions. Le style est propre mais incohérent entre les modules.

347346	2.00	[L1] Simulation.h, Fourmillière.h, etc... - les mots clés de portée de doivent pas être indentés (il doit donc y avoir une indentation de moins pour le reste du code affecté par ces portées) [L2]: squarecell.h.28, squarecell.cc.101, fourmillière.h.25 et fourmillière.cc.9,35,37, fourmi.cc... contiennent plus de 87 caractères	Warning [P5] 1) : Il manque un symbole pour la vie initiale de la fourmie génératrice Warning: Dans nourriture.cc et fourmillière.cc, le nom de variable o, et dans simulation::décodage, les noms de variables i et j manquent de clareté Warning: Il faut rester cohérent vis-à-vis du retour à la ligne pour un instruction unique suivant un if ou un else	Bon travail ! Essayez d'adhérer un peu plus votre code en ajoutant des espaces dans vos expressions (par exemple après chaque virgule dans la liste d'arguments passés à une fonction), de supprimer les retours à la ligne en trop à divers endroits dans votre code, et de rendre certains de vos noms de variable plus explicites (par exemple val[0/1/2/.../5] dans simulation::decodage_ligne pourrait être plus explicite). Continuez comme ça!
347369	4.00		Style excellent !	attention à laisser l'implémentation des méthodes aux fichiers .cc.
347381	4.00	good	squarecell must be independent!	respected style rules, but squarecell must be independent of the rest of the program (it seems that it even is if the includes are removed)
347428	4.00	L1:projet.cc:19;fourmis.h:55;fo od.cc:27,31-39		Attention à ne pas inclure constant.h dans squarecell, le module doit être indépendant du reste du projet. Sinon très bon style de code dans le projet, attention à quelques petits problème d'indentation. Continuez comme ça pour la suite du projet!
347445	2.00	[L1]private/public keywords, nourriture.h 14-19 [L2] squarecell.cc 101,102 simulation.cc 110, fourmi.h 28,35	The public/private keywords should not be indented, be careful to always use the same size of indentation.	Very good code. Be careful with your style for next time.
347499	3.00	[L1] inconsistency in the files names squarecell.cc , Simulation.cc		really good work overall, two points loss because of small mistakes but the implementation is fine apart from nourriture
347510	2.00	[L1] Indentation de public et private [L2] fourmi.cc L31 , fourmi.h L15/28/39 , fomiellere.cc L12/19		Il faut faire attention aux conventions de programmation. Vous avez raté beaucoup de points sur le style de code. Pour les prochains rendus il faut faire attention à l'indentation de public et private et au nombre de caractères par ligne

347530	2.00	[P2] simulation.cc fonction de lecture < 80 lignes [P5] magic number dans fourmi.cc et squarecell.cc "1000"	Warning: dans la fonction de lecture il y a des accolades inutiles	Le code est bon! Bravo pour avoir bien respecté les 87 caractères, d'avoir fait des sauts à la lignes de manière élégante. Le code est très cohérent, lisible. Quel dommage que vous ailliez dépassé les 80 lignes de la fonction de lecture, mais je pense que aviez fait du debuggage et vous n'avez pas enlevé les accolades. Sans cela vous n'auriez pas dépassé les lignes. Attention aux magic number car à la lecture du code on doit comprendre du premier coup d'oeil sans avoir besoin de lire le code pour comprendre ce que c'est.
---------------	------	---	--	---

10.00
6.00
6.00
8.00
9.00
8.00
10.00
10.00

6.00
7.00
6.00
10.00

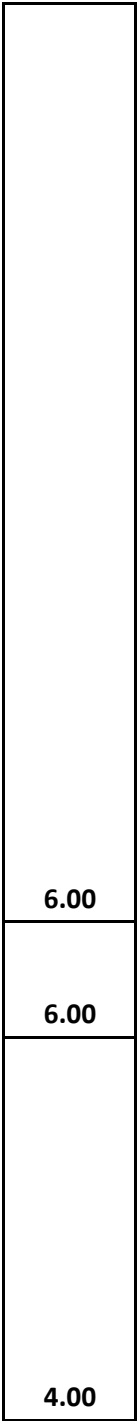
8.00
8.00
8.00
8.00
10.00
10.00
8.00
8.00
5.00
10.00

8.00

9.00

7.00

6.00



7.00
10.00
9.00
5.00
9.00
8.00
8.00
9.00

8.00
6.00
7.00
8.00
<u>9.00</u>
9.00
8.00
9.00
8.00
6.00

9.00
8.00
5.00
9.00
8.00
9.00
8.00
8.00

6.00

9.00

7.00

10.00

8.00

6.00

9.00
10.00
0.00
4.00
8.00
8.00
8.00
9.00
10.00

8.00

4.00

5.00

8.00

7.00

4.00

10.00

9.00

9.00

10.00

10.00

9.00

9.00

9.00

5.00
9.00
10.00
8.00
10.00
9.00
9.00
8.00
7.00

9.00

6.00

10.00

10.00

8.00

9.00

10.00

8.00

9.00

10.00

9.00

8.00

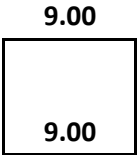
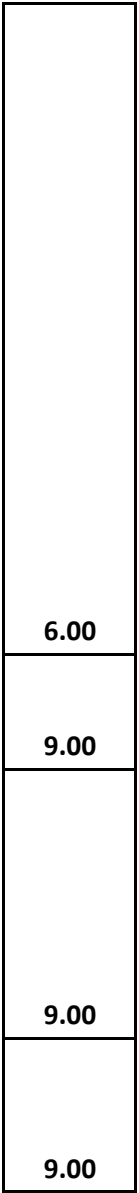
9.00

9.00

10.00

8.00

8.00



8.00
8.00
9.00
9.00
8.00
7.00

8.00
8.00
8.00

6.00
7.00
10.00
6.00
9.00
8.00
9.00
8.00

6.00

7.00

9.00

9.00

10.00

9.00
6.00
9.00

8.00
7.00
6.00
10.00
9.00
10.00
6.00

9.00
8.00
5.00
9.00
7.00
5.00
9.00

9.00
6.00
10.00
9.00
10.00
<u>9.00</u>
9.00
7.00
9.00

10.00
6.00
5.00
6.00
6.00
9.00
9.00
5.00
10.00

9.00
4.00
9.00
8.00
6.00
5.00
6.00
9.00

8.00

10.00

8.00

9.00

6.00

9.00

8.00
10.00
5.00
10.00
10.00
7.00
9.00
7.00

9.00

5.00

10.00

9.00

10.00

8.00

8.00

7.00

10.00

9.00

6.00

8.00

7.00

8.00

9.00
7.00
8.00
8.00

9.00
9.00
6.00
10.00
8.00
10.00
10.00
7.00

6.00
9.00
0.00
8.00
6.00
9.00
5.00

7.00
9.00
9.00
9.00
7.00
8.00
6.00

8.00