

ECE CITY 2022



Contenu

Objectifs et modalités du projet	2
Présentation des objectifs du jeu ECE-City 2022	3
Le terrain de jeu - Niveau 0 – Modalités et règles de connexion	4
La boîte à outils	5
Les réseaux (graphes) d'eau et d'électricité: Niveaux -1 et -2	5
Compléments au démarrage du jeu.....	8
Les stades de développement automatique des constructions.....	8
2 modes d'évolution des constructions : mode communiste/mode capitaliste.....	9
Régression	9
Aspects financiers.....	10
Extensions possibles : Casernes et zone d'influence en cas d'incendie	10
Autres extensions possibles : Pause, reprise, sauvegarde et chargement.....	10
Extensions en extra : 2D ½ ou la perspective isométrique	10
Dernières extensions possibles	11
Contraintes matérielles et logicielles	11
Livrable de conception : PowerPoint	11
Livrable du code	12
Critères d'évaluation et barème	12

Objectifs et modalités du projet

Ce projet informatique a pour but de travailler la **théorie des graphes** (pour le moteur de jeu), **intégrée à un environnement console et éventuellement à un environnement graphique** (pour la vue).

Il nécessite, comme tous les projets bien menés, une analyse des problèmes de manière **modulaire** et une réflexion poussée au niveau algorithmique que vous détaillerez dans le rapport qui accompagnera votre travail (éventuellement sous forme d'organigrammes).

Il se déroule de la **semaine du 24 octobre 2022 à celle du 21 novembre incluse** (5 semaines) selon l'approche « apprentissage orientée projet » (Project Based Learning).

Le Cahier Des Charges (CDC) du sujet vous est au début du projet. Deux conséquences pour vous :

- Il vous sera nécessaire d'étudier attentivement et rapidement le CDC pour le décomposer en étapes, identifier celles que vous pouvez traiter dès maintenant et commencer à les réaliser (comme vue en cours avec la méthode Pert).
- Pour les autres tâches, vous aurez également à identifier les besoins et à rechercher les connaissances nécessaires pour y répondre. Pour cela, vous aurez à utiliser tous les outils à votre disposition (web, tutoriaux, livres...) pour identifier, comprendre et vous approprier les techniques utiles au jeu et qui ne vous auraient pas été encore enseignées. Ainsi, à l'occasion des futurs cours, vous aurez les idées claires, un besoin précis, de nombreuses questions, et surtout une capacité accrue de comprendre la théorie enseignée. Il vous sera nécessaire d'étudier attentivement le Cahier Des Charges (CDC) pour le décomposer en étapes, identifier celles que vous pouvez traiter dès maintenant pour en faire la conception et commencer à les réaliser.

Pour chaque tâche, vous aurez à identifier les besoins et à rechercher les connaissances nécessaires pour y répondre. Pour cela, vous aurez à utiliser tous les outils à votre disposition (web, tutoriaux, livres...) pour identifier, comprendre et vous approprier les techniques utiles au jeu en vous servant, entre autres, de la théorie enseignée en cours et de sa mise en pratique en TP.

Cette phase d'étude et d'analyse donnera lieu à **un première documentation PowerPoint à déposer sur BoostCamp au plus tard le dimanche 20 novembre 2022** sur le lien [Documentation à déposer : PowerPoint deadline le dimanche 20 novembre 2022](#). Les consignes de ce livrable sont expliquées dans la page BoostCamp [Cours : Projet Informatique Théorie des graphes, Section : Livrables à déposer \(omneseducation.com\)](#).

Un **second livrable d'implémentation en C et de ses tests** sera à déposer sur BoostCamp au plus tard le dimanche 27 novembre 2022 sur le lien [Livrable du code à déposer : code en C deadline le dimanche 27 novembre 2022](#) de la même page BoostCamp. **Respectez les consignes spécifiées plus loin pour ces 2 livrables et sur cette page BoostCamp.**

Ce projet sera à faire par équipes de 3 ou de préférence 4 issu du même groupe de TD.

Pour finir, vous aurez le plaisir de **soutenir oralement votre projet devant votre groupe de TD : les soutenances se dérouleront la semaine du 28 novembre 2022.**

Présentation des objectifs du jeu ECE-City 2022

Vous serez en charge de développer un jeu inspiré du célèbre simulateur SIM-CITY.

C'est une version simplifiée de ce jeu magique que nous vous proposons de réaliser.

Dans ce jeu, vous êtes le maire de la ville ECE-city, en charge du bien-être de ses citoyens. Vous avez le pouvoir de construire des infrastructures (routes, lignes électriques, canalisation d'eau potable) et des bâtiments (châteaux d'eau, centrales électriques, caserne de pompiers,...).

Toutes ces tâches seront à réaliser en gérant un budget serré qui doit équilibrer les dépenses liées à ces constructions et les recettes liées aux impôts et taxes.

Vous devrez mettre en œuvre les exigences du CDC que constitue ce document et permettre à votre jury de vérifier facilement que c'est le cas en **respectant un scénario qui vous sera soumis le jour de la soutenance**. Vous pourrez évidemment laisser aller votre imagination et apporter toutes les améliorations susceptibles de rendre le jeu plus attractif, notamment au niveau de la vue.

Les illustrations de la **figure 1** ci-dessous présentent la ville en début de jeu sous forme de grille, que vous pourrez construire en version 2D ou en vue perspective 2D ½, avec sa barre d'outils. Elles ne sont pas contractuelles.

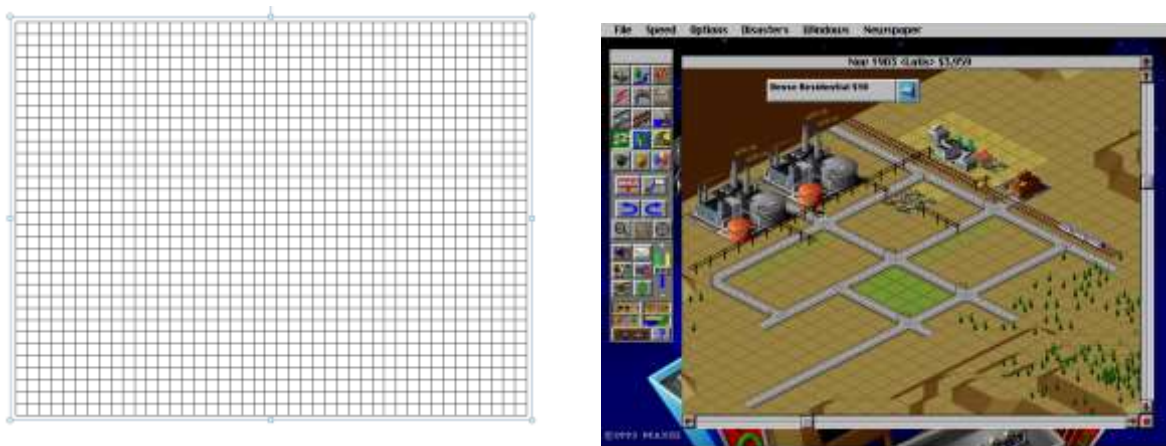


Image extraite du fichier **sim-city-001.bmp**

Figure 1 : Exemples de représentation du terrain de jeu

Dans notre version simplifiée de SimCity, ECE-City, l'objectif pour le joueur est de **favoriser le développement automatique de sa population**.

- Pour que la population se développe, il faut donner aux gens une habitation.
- Pour qu'une habitation se construise, il faut la connecter au réseau routier (qu'il faut donc construire au préalable) et l'alimenter en eau et en électricité.
- Pour distribuer de l'électricité et de l'eau aux maisons, il faut construire des usines électriques et des châteaux d'eau.

- La banque centrale de la ville dispose d'une certaine somme pour construire les infrastructures (routes, lignes électriques, canalisation d'eau potable), les usines électriques, les châteaux d'eau, les maisons, les casernes de pompiers...
- Les habitants payent automatiquement des impôts à la banque centrale.
- Les constructions évoluent individuellement au rythme de « cycles » de 15 secondes représentant 1 mois de la vie réelle.

Le terrain de jeu - Niveau 0 – Modalités et règles de connexion

Le terrain est une **grille de 45 colonnes sur 35 lignes**. Pour les étudiants les plus en difficulté, vous commencerez par dessiner cette grille en mode console. Pour les étudiants qui dessineront ce terrain en mode graphique, ils choisiront une résolution fixe d'écran de **1024x768**. Chaque case de la grille a donc une résolution de **20x20 pixels**.

Il a 3 niveaux de profondeur, **le joueur commençant par construire le niveau 0** :

- Le **niveau 0** correspond au niveau du sol. Ce niveau **doit être chargé à partir d'un fichier texte** contenant les éléments suivants à connecter :
 - Élément de route, canalisation, ligne électrique : **1 case**
 - Élément de construction d'habitation : **3x3 cases**
 - Élément de bâtiments prédéfinis (usines, châteaux d'eau, casernes...) : **4x6 cases**

Pour chaque élément, pensez à inclure dans le fichier les informations nécessaires pour le chargement puis l'affichage en mode console, puis éventuellement en mode graphique, afin d'y connecter les différents éléments.

Ce niveau est le plus élaboré et également graphiquement le plus joli car composé d'un assemblage d'images. C'est là que le joueur pourra construire **avec la souris ou le clavier** le réseau de routes, les habitations, les bâtiments particuliers comme les centrales électriques, les châteaux d'eau et les casernes de pompiers...

Chaque élément construit par le joueur devra être sauvegardé dans un/des fichiers texte, avec toutes les informations nécessaires pour un futur chargement.

❖ Pour ceux qui élaborent graphiquement assemblage d'images, il est plus joli mais pas nécessairement indispensable, car le mode console suffit.

Les cases du terrain où vous déposerez ces éléments pourront être **sélectionnés au clavier** en mode console **ou à la souris** en mode graphique, en fonction de vos compétences de programmeur. Il y aura un seul élément par case, **certains éléments s'étendant sur plusieurs cases**.

Pour qu'une construction soit « viable », elle doit être obligatoirement connectée au réseau routier, alimentée en eau **et** en électricité : voir **Figure 2** ci-dessous.

Pour qu'un réseau de canalisation ou d'électricité soit alimenté, il doit être connecté à un château d'eau ou une centrale électrique.

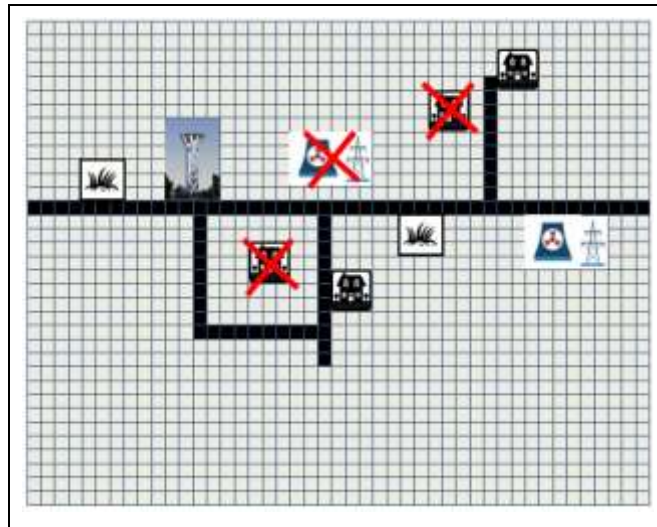


Figure 2 : Constructions viables et non viables

- ❖ Remarque : pour détecter les constructions « viables » et « non viables », on pourra se servir des algorithmes de connexité et de coloration vus en cours et TP 😊

La boîte à outils

Une zone spécifique du niveau 0, dite « **boîte à outils** » permet au joueur de sélectionner avec **le clavier** en mode console ou **la souris** en mode graphique plusieurs types d'objets, chacun étant représenté par une icône (image) : route, terrain vague et tous les types de bâtiments (caserne, château d'eau et centrale électrique...).

Cette zone outils permet aussi de changer le niveau de visualisation.

Les réseaux (graphes) d'eau et d'électricité: Niveaux -1 et -2

Via la boîte à outils présente à côté de la grille, le joueur peut basculer sur les niveaux -1 et -2 : voir la **Figure 3** de la représentation simplifiée des 3 niveaux 0, -1 et -2 page suivante.

- Le niveau -1 correspond au niveau des canalisations d'eau douce. Le réseau de canalisations de ce niveau est construit automatiquement : il correspond en totalité au réseau routier défini au niveau 0, construit à l'étape précédente.
- Le niveau -2 correspond au réseau des lignes électriques sous-terraines. Comme précédemment, ce réseau est construit automatiquement à la création du réseau routier du niveau 0 et lui correspond en totalité.

La **vue d'un niveau particulier** met en évidence les informations relatives au fonctionnement de ce niveau et enlève les informations superflues des autres niveaux.

Aucune action n'est possible sur ces 2 sous-niveaux, réservés à l'observation.

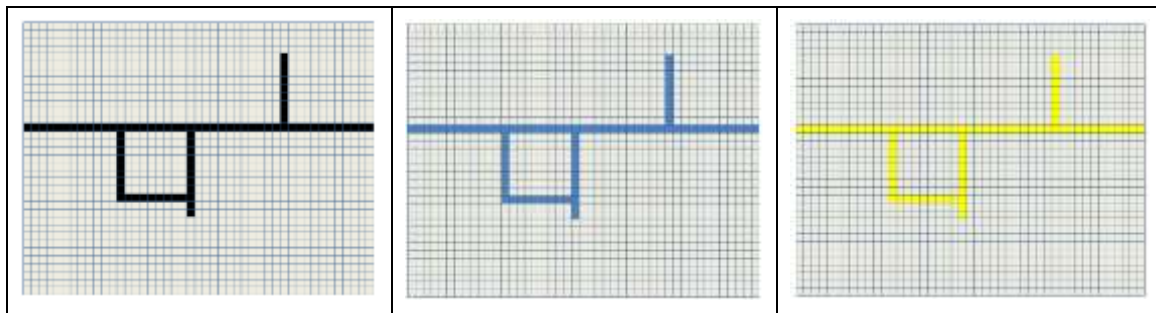


Figure 3 : Niv. 0 : réseau routier (à gauche en noir), niv. -1 eau (en bleu), niv. -2 : électricité (en jaune)

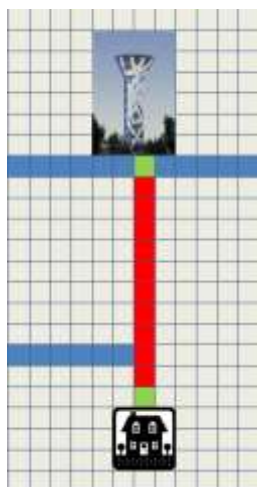
Alimentation en eau – Niveau -1

Ce niveau met en évidence la distribution de l'eau des châteaux d'eau vers les habitants. Cette distribution se fait sur le mode « **au plus proche** » et respecte les règles suivantes :

- La capacité totale d'un château d'eau permet d'alimenter en permanence **5000** habitants.
- Les habitants les plus « proches » du château d'eau sont servis les premiers. La proximité est calculée en comptant le nombre de cases de la grille correspondant à la route **entre** le château d'eau et la construction, avec +1 pour le château d'eau et +1 pour la construction (voir illustrations suivantes).



La distance calculée entre le château d'eau et la construction est ici de 9 (cases rouges) +1+1 (cases vertes), soit 11.



Dans ce cas de figure, la distance est $10+1+1 = 12$.

- Si le nombre d'habitants total sur le jeu est inférieur à la capacité d'un château d'eau, l'eau en surplus n'est tout simplement pas distribuée.

- Si le nombre d'habitants est supérieur, une construction peut être partiellement alimentée.
- Une construction peut être alimentée par plusieurs châteaux d'eau si le plus proche n'a pas la capacité suffisante.
- Lors de l'évolution de la ville, si un château d'eau ne peut plus alimenter toutes les constructions ou si un château d'eau est déplacé ou ajouté, la règle de re-répartition de l'eau doit **minimiser la perte d'habitants**.

Le niveau -1 permet de visualiser le réseau de canalisation, les châteaux d'eau, les constructions alimentées avec leur taux d'approvisionnement en eau ainsi que la(les) source(s) d'approvisionnement (par exemple, pour une habitation de 1000 habitants - château 1 : 100/1000, château 2 : 800/1000 (avec un code couleur si possible)).

Il met en évidence les constructions en carence d'eau.

❖ Remarque : pour calculer la distribution de l'eau en fonction de la capacité donnée d'un château d'eau, on pourra se servir de certains algorithmes vus en cours et TP permettant de donner la priorité aux habitants les plus proches mais le plus équitablement possible pour tous les habitants. Tout cela, en tenant compte de toutes les règles (points) mentionnées ci-dessus. 😊

Réseau électrique – Niveau -2

Ce niveau met en évidence la distribution de l'électricité vers les habitants. Cette distribution se fait sur le mode « **en tout ou rien** » et respecte les règles suivantes :

- La capacité totale d'une centrale permet d'alimenter en permanence **5000** habitants.
- Si le nombre d'habitants total sur le jeu est inférieur à la capacité d'une centrale, l'électricité en surplus n'est tout simplement pas distribuée.
- Une centrale alimente une construction si sa capacité lui permet d'alimenter « **tous** » les habitants de la construction. Dans le cas contraire, la centrale n'alimente **pas du tout** la construction.
- Les capacités des différentes centrales présentent sur le jeu sont additionnées et alimentent indifféremment une construction ou une autre.

Le niveau -2 permet de visualiser le réseau électrique, les centrales, les constructions alimentées et **met en évidence les constructions en carence d'électricité**.

❖ Remarque : comme pour la remarque du **niveau -1 (alimentation en eau)**, vous pouvez appliquer des algorithmes vus en cours et TP en tenant compte de toutes les règles (points) mentionnées ci-dessus pour le **niveau -2 (réseau électrique)**. 😊

NB : Pour ces 2 niveaux -1 et -2, vous afficherez pour chaque château d'eau (au niveau -1) et chaque centrale électrique (niveau -2) leur quantité distribuée vs leur capacité : par exemple au niveau -1, château 1 : 4000/5000, château 2 : 3000/5000 etc.; au niveau -2, centrale 1 : 2000/5000, centrale 2 : 5000/5000 etc.

Compléments au démarrage du jeu

Quel que soit le niveau affiché à l'écran, le joueur dispose d'informations complémentaires permanentes :

- Un **compteur temporel** temps réel à la seconde apparaît à l'écran et tourne jusqu'à la fin de la partie.
- Un compteur « **monnaie** » affiche la somme initiale en jeu : **500 000 ECEflouz**. Cette somme sera décrémentée par les achats et incrémentée par les impôts.
- Un compteur « **habitants** » indique la somme totale d'habitant dans la ville.
- Un compteur « **capacité électrique** » et « **capacité en eau** » indiquent la somme des capacités de chaque ressource.

Un objet posé sur la grille et qui respecte les conditions de viabilité (connexion au réseau routier, eau, électricité pour ses habitants) est immédiatement opérationnel :

- Les châteaux d'eau et les centrales alimentent non-stop les réseaux.
- Les terrains vagues commencent leurs cycles de vie de **15 secondes par état** (voir ci-dessous)

Les stades de développement automatique des constructions

Le terme de **construction** représente 2 types de bâtiments dans le jeu :

- Des bâtiments « clef en main » comme les usines, les centrales, les casernes...
- Des terrains vagues dont le joueur définit l'emplacement, et qui vont ensuite se construire tout seul en changeant d'état (on parle de « stade » de construction) **toutes les 15 secondes** (1 cycle de vie) si les conditions idéales sont réunies.

Le joueur qui sélectionne via la boîte à outils la construction d'un terrain vague et qui le positionne sur la grille, déclenche immédiatement le « compteur temporel de vie » de la construction qui va la faire évoluer automatiquement à chaque cycle de 15 secondes de la vie de la construction.

Si les conditions le permettent, une construction va évoluer, à la fin de chacun de ses cycles, selon les stades suivants, **les données suivantes devant être chargées d'un fichier** :

- **Terrain vague** (0 habitant) ou **Ruine** en cas de régression (0 habitant)
- **Cabane** (10 habitants)
- **Maison** (50 habitants)
- **Immeuble** (100 habitants)
- **Gratte-ciel** (1000 habitants)

Les usines, centrales, casernes et autres bâtiment que vous êtes susceptibles d'intégrer au jeu sont des lieux de travail et ne comptent aucun habitant. Ils sont opérationnels dès leur positionnement sur le terrain (sous réserve des règles d'activation) et n'évoluent pas durant toute la partie.

En 1 minute une construction a donc atteint son stade ultime, **si tout va bien**.

Un terrain vague dont les conditions d'évolution (connexion au réseau routier, eau, électricité pour ses habitants) ne sont pas remplies ne démarre pas ses cycles d'évolution.



Figure 4 : Les stades d'évolution des constructions

2 modes d'évolution des constructions : mode communiste/mode capitaliste

Le mode communiste

Dans ce mode, une construction n'évolue pas si les ressources (eau+électricité) nécessaires à sa croissance ne sont pas disponibles. La construction va donc :

- Evoluer à chaque cycle si les conditions eau+électricité du jeu le permettent.
- Conserver son stade actuel tant que les conditions eau+électricité le permettent, et qu'elle n'a pas pu évoluer.
- Régresser au stade précédent si les conditions ne lui permettent plus de se maintenir ou d'évoluer (une cabane régresse au stade de ruine avant d'évoluer de nouveau).

Le mode capitaliste

Dans ce mode, une construction passe systématiquement au stade supérieur, que les ressources (eau+électricité) soient disponibles ou pas. L'évolution de l'ensemble des constructions s'en trouve alors bouleversée, toujours dans le respect des règles de régression qui maximisent le nombre d'habitants total de la ville.

Le mode communiste ou capitaliste sera choisi au démarrage du jeu.

Régression

En cas de simultanéité temporelle, la régression automatique de stade pour des constructions n'ayant plus les conditions pour se maintenir ou évoluer doit se faire de manière à réduire au minimum le nombre global d'habitants dans la ville.

Aspects financiers

La banque dispose au départ de **500 000 ECE-flouz**.

Dépenses

La construction de bâtiment a un coût :

- | | | |
|-------------------|---------|-------------------------------|
| - Centrale : | 100 000 | ECE-flouz |
| - Château d'eau : | 100 000 | ECE-flouz |
| - Route : | 10 | ECE-flouz par unité de grille |
| - Terrain vague : | 1000 | ECE-flouz |

Impôts et taxes

Chaque habitant verse 10 ECE-flouz à chaque fin de cycle de l'habitation qu'il occupe.

Le compteur d'argent indique en permanence le solde à la banque. Une construction n'est pas possible si la banque de la ville ne dispose pas de la somme nécessaire.

Extensions possibles : Casernes et zone d'influence en cas d'incendie

Une caserne coûte 10000 ECE-flouz.

Toutes les constructions qui se trouvent dans une « zone d'influence » de **20 cases maximum** de la route avec une caserne - soit 18 +1 (pour la caserne) +1 (pour la construction) - seront protégées en cas d'incendie.

A chaque cycle d'une construction, un incendie peut se déclencher arbitrairement. Il est alors visualisé.

Si un incendie se déclenche dans une construction en « zone d'influence », on visualise le **chemin le plus proche entre la caserne et la construction**, et l'incendie disparaît.

Si la construction en incendie n'est pas dans une zone d'influence, elle passe directement à l'état de « ruine ».

Autres extensions possibles : Pause, reprise, sauvegarde et chargement

Durant le jeu, l'utilisateur pourra faire une « pause » arrêtant le compteur temporel du jeu et de chaque construction. Une « reprise de la pause » permet de redémarrer les compteurs. Ces 2 actions se font avec un clic souris dans la « zone outil » (mode graphique) ou avec la même touche du clavier (mode console).

De même, le joueur « sauvegardera » son jeu entier (état de la grille, de ses éléments, mode d'évolution, les compteurs etc.) dans un fichier et « chargera » du fichier le dernier jeu sauvé, soit avec la zone outils (mode graphique) soit avec le clavier (mode console).

Extensions en extra : 2D ½ ou la perspective isométrique

Certaines des illustrations de ce document sont en 2D1/2 (encore appelé 3D isométrique), c'est-à-dire sans point de fuite : voir **Figure 5** page suivante. Pour ceux d'entre vous qui le souhaitent, vous pouvez aller dans ce mode graphique MAIS ce n'est pas du tout prioritaire pour ce projet. Vous trouverez facilement sur internet des icônes et illustrations dans ce mode.



Figure 5 : La 2D ½

Dernières extensions possibles

Pour les meilleurs d'entre vous, vous pourrez aussi vous préoccuper de l'éducation des habitants de votre ville et construire écoles, collèges, lycées, universités, écoles d'ingénieurs (évidemment !). Vous pourrez également penser à leurs distractions et construire des bibliothèques, des parcs, des stades... Soyez créatifs dans vos extensions **MAIS qui ne sont pas les tâches prioritaires dans ce projet, par rapport à toutes celles prioritaires qui ne sont pas mentionnées comme des extensions !**

Contraintes matérielles et logicielles

Langage C en mode console en priorité, et éventuellement en mode graphique via Allegro en résolution 1024x678 ou toute librairie graphique (exemples : SFML etc.) compatible avec le langage C, sur CLion ou CodeBlocks.

Matériel :

Vos PC personnels ou ceux de l'école. La soutenance se fera sur une machine de l'école après récupération de votre archive déposée sur BoostCamp (**testez-le avant !**).

Livrable de conception : PowerPoint

Votre premier compte-rendu PowerPoint (15 slides maximum) sera à déposer sur la page BoostCamp [Cours : Projet Informatique Théorie des graphes, Section : Livrables à déposer \(omneseducation.com\)](https://omneseducation.com) **au plus tard le dimanche 20 novembre 2022** dans le lien [Documentation à déposer : PowerPoint deadline le dimanche 20 novembre 2022](#).

Il devra respecter le plan détaillé suivant :

1. Page de garde avec titre, groupe de TD, numéro d'équipe et noms (1 slide)
2. Sommaire (1 slide)
3. **Versioning GIT** : screenshot montrant la bonne utilisation et répartition des tâches du code entre coéquipiers. (1 slide)
4. **Méthodologie de votre travail** : (8 slides maximum)
 - tableau de la répartition des tâches par personne et fonctionnalités , avec durées des tâches et tâches antérieures (1 slide)

- Pert : graphe avec dates au plus tôt et au plus tard déduit des tâches au-dessus, marge des tâches et chemin(s) critique(s) déduits de ces marges (1 slide)
- présentation synthétique de vos approches : inspirez-vous des slides du cours avec des algos courts et "en français" (3 slides au max)
- tableau comparatif de la complexité des différents algorithmes (1 slide)
- remarques, schémas, diagrammes, formules mathématiques justifiées, screenshots annotés : tout ce qui peut permettre de rendre votre présentation informative, efficace et dynamique (2 slides max)
- présentation claire, concise et précise du travail réalisé (1 slide)

-> qu'avez vous entrepris -> comment -> quels sont les résultats

5. Bilan individuel et collectif sans blabla de l'état du travail effectué. (1 à 2 slides)

6. Bibliographie précise de toutes vos sources (web, livres etc.). (1 slide). Toute source non citée est considérée de facto comme un plagiat

Livrable du code

Votre second livrable du « code » sera à déposer à déposer sur la page BoostCamp [Cours : Projet Informatique Théorie des graphes, Section : Livrables à déposer \(omneseducation.com\)](https://omneseducation.com) **au plus tard le dimanche 27 novembre 2022** dans le lien [Livrable du code à déposer : code en C deadline le dimanche 27 novembre 2022](#).

Le second livrable du code à implémenter en C devra respecter les consignes suivantes :

Fichier zip ou rar avec tous les dossiers et fichiers du projet développé sur Clion ou CodeBlocks avec :

- Le code source avec tous les fichiers .h et .c, l'exécutable .exe
- Les fichiers texte, éventuellement svg pour le visuel graphique, de vos graphes et des résultats en respectant le format des fichiers texte dans le sujet
- Eventuellement, toute librairie graphique nécessaire que cela soit sur Allegro (dll nécessaires), SFML ou toute autre librairie compatible avec le langage C.

Pour vos sources, citez précisément tout document exploité en commentaires de votre code : auteur(s), liens vers les sites etc.

Critères d'évaluation et barème

Votre travail sera jugé sur les critères suivants :

- Le respect rigoureux des règles du jeu énoncées précédemment (CDC)
- La modularité de votre conception et donc de votre code.
- La bonne répartition des tâches entre les membres de l'équipe.
- L'intérêt, l'originalité, la jouabilité et toutes les caractéristiques que vous prendrez soin de mettre en avant lors de la soutenance.

La note de projet est la moyenne pondérée du **PowerPoint coef 1** (notée sur 20) et de la **soutenance de la démo du code coef 2** (sur 20 aussi), avec au maximum 4 points de bonus pour les extensions mentionnées dans le sujet MAIS pas prioritaires, la note étant plafonnée à 20.

- ❖ Les grilles d'évaluation détaillées du PowerPoint et de la démo arriveront plus tard...
En attendant, bonnes constructions dans ECE CITY 2022 ☺