

Projet : Cluedo



Cluedo est un [jeu de société](#)⁽¹⁾ dans lequel les joueurs doivent découvrir parmi eux qui est le meurtrier d'un crime commis dans un manoir anglais, le Manoir Tudor.

But

Le but du jeu est de trouver qui a tué le Docteur Lenoir, dans quelle pièce du manoir et avec quelle arme a été commis le meurtre.

Mise en place

En début de partie on classe les différentes cartes indices par type On distingue entre trois types de cartes : suspects / armes / lieux. Ensuite on en tire une de chaque au hasard que l'on met dans l'enveloppe prévue sans les regarder : il s'agit de la solution de l'énigme de cette partie. Chaque joueur choisit ensuite un pion. Les cartes indices restantes sont alors mélangées et distribuées faces cachées équitablement à tous les joueurs. Ces cartes servent alors d'alibi : si un joueur possède cette carte c'est qu'elle n'est pas parmi les trois cartes mises à part et donc que l'indice correspondant n'est pas la clef de l'énigme. C'est donc une carte qui a son alibi et qui n'est pas suspecte.

Déroulement

Pour découvrir la vérité sur le meurtre chaque joueur peut à tour de rôle se déplacer dans le manoir et émettre une hypothèse sur le crime.

Pour se déplacer dans le manoir, le joueur jette le dé et se déplace du nombre correspondant de cases sur le plateau dans le couloir. Lorsqu'il passe une porte ouverte, le joueur arrive dans une pièce et ne peut plus avancer ce tour-ci (même

s'il n'a pas utilisé tous les déplacements auxquels le dé lui donnait droit), mais au tour prochain il pourra repartir de n'importe quelle porte ouverte de cette pièce. Les deux passages secrets entre les coins opposés du plateau peuvent s'emprunter uniquement lorsque l'on est dans une pièce de coin, on peut alors au lieu de lancer les dés choisir de passer directement dans le passage secret.

Il existe dans la version classique deux passages secrets : un qui relie la cuisine à l'observatoire, et un autre qui relie le spa au pavillon des invités.

Lorsqu'un joueur termine son tour dans une pièce il peut proposer une reconstitution du crime dans cette pièce. Il choisit une arme et un personnage qu'il amène directement dans cette pièce puis énonce la reconstitution : « je suspecte le professeur Violet d'avoir tué le docteur Lenoir avec le poignard dans le salon » (le joueur doit se trouver dans le salon pour faire cette reconstitution.) Il interroge alors le joueur à sa gauche dans le but d'authentifier sa reconstitution : si ce dernier a au moins une carte alibi, il est tenu de lui en montrer une (il peut choisir celle qu'il montre) en secret. S'il n'en possède pas, il ne montre rien et le joueur à sa gauche prend la relève. Lorsqu'une carte a été montrée, la reconstitution est invalidée et le tour passe au joueur suivant. Dans l'exemple si le joueur questionné a la carte du poignard mais pas celle du professeur Violet ni celle du salon, il la montre alors au joueur qui a fait la reconstitution, sauf si le joueur qui soupçonne pratique la même stratégie proposée ci-dessous (soupçonner alors qu'on a l'alibi pour confirmer une idée). Il faut ainsi être prudent.

Fin du jeu

Lorsqu'un des joueurs pense avoir trouvé les trois éléments du meurtre, il doit accuser (et non plus soupçonner) le personnage voulu (joueur ou non) qu'il pense être coupable du meurtre, ainsi que l'arme et la pièce utilisées (le joueur n'est pas nécessairement obligé de se trouver dans la pièce qu'il soupçonne pour faire son accusation). Le joueur vérifie alors secrètement les trois cartes solution, cachées dans l'enveloppe.

- S'il a raison, il annonce les cartes aux autres joueurs et la partie s'achève.
- S'il s'est trompé, il remet les cartes sans les révéler aux autres joueurs, et il s'arrête de jouer. Les autres joueurs continuent de chercher jusqu'à ce que la partie s'achève.

Cahier des charges de base : 15 points

Ce projet consiste à créer un jeu s'appuyant sur les règles du Cluedo. **Vous imaginez vous-même le scénario du jeu**, c'est à dire qu'un meurtre n'est pas nécessairement commis.

Le but reste avant tout de pratiquer la programmation orientée objet en C++.

Vous pouvez demander à votre chargé de TP des modifications si vous pensez à de meilleures règles, mais prévoyez de justifier votre demande par un intérêt « programmatique », par exemple avec un diagramme de classe... et aussi souvenez-vous que la soutenance doit tenir dans 12 minutes (pas trop compliqué !)

Sinon vous avez toute liberté quant à l'univers, son histoire, au style, aux réglages fins de « game-play ». Il n'est pas demandé de copier CLUEDO : un bonus d'évaluation viendra récompenser les équipes qui auront fait l'effort de l'originalité. Pensez à des univers différents des Manoir Tudor, par exemple une musée, ou Disney land, etc...

Pour illustrer la présentation des règles par des exemples précis, je propose un jeu cd CLEDEO à l'ECE de Paris, que j'appellerai **ECEPARIS**. Les règles abstraites à respecter sont en noir, les exemples qui viennent illustrer sont en bleu.

Votre programme permet de créer tout d'abord un plateau de jeu qui se compose d'endroits dont certains sont absorbants (Un endroit absorbant est un endroit qui absorbe un joueur dès qu'il passe par cet endroit).

Exemple ECEPARIS : Comme plateau de Jeu, on peut choisir l'étage 3 et l'étage 4 du bâtiment EIFFEIL 1. Les endroits à considérer sont les salles de cours, les amphis, les bureaux de scolarité, l'ascenseur et la salle des professeurs, les bureaux des enseignants et la cuisine. Comme endroits absorbants, nous pouvons définir 2 salles de cours, 3 bureaux d'enseignant, la cuisine et un amphi.

Le programme permet de créer des cartes. Il existe au moins trois types de carte. Il faut créer pour chaque type de carte différents exemples de carte.

Exemple ECEPARIS : Comme types de cartes, on peut définir des cartes suspect, armes et lieux. Pour les cartes suspect, on peut choisir des personnels de l'ECE tels que M. Segado, M. Rebai, M. diedler, Mme Rendler etc. Pour les cartes arme, on peut définir des armes blanches (couteau, poignard, matraque, etc) ou autre type d'armes. Pour les cartes lieu, on peut définir pour notre cas les endroits cités en haut de l'étage 3 et l'étage 4 du bâtiment EIFFEIL 1.

Le programme permet de construire une enveloppe qui contient 3 cartes (1 de chaque type)

Exemple ECEPARIS :

carte suspect : M. Chaari

carte arme : couteau

carte lieu : amphi 445.

Le programme permet aussi de créer et d'enregistrer de nouveaux joueurs, chaque joueur doit posséder un compte. Un joueur peut être débutant/novice, amateur, intermédiaire ou expert. Il doit permettre, pour chaque joueur, la création d'une collection de départ parmi toutes les cartes du jeu (ces opérations se font dans l'interface du programme et ne nécessitent pas d'éditer du code ou un fichier). Votre programme organise une succession de matchs entre joueurs, l'interface permettant de choisir quels sont les joueurs qui vont s'affronter (faire le prochain match) et permettant à chacun de ces joueurs d'avoir ses cartes sauvées sur fichier (pas à retaper à chaque exécution). Ensuite la partie commence par l'affectation d'un pion pour chaque joueur. Votre programme doit être capable d'enregistrer un match qui n'est pas terminée et de la reprendre dans le futur.

Exemple ECEPARIS : Alexis Dupont ne possède pas de compte. On lui crée un compte. Il considéré comme débutant.

Jean Durant a joué 13 matchs et il a gagné 5 fois. Il est expert.

Maria Larousse a joué 3 fois et n'a jamais gagné. Elle est amatrice.

Etc.

On a lancé le programme, on demande le nombre de joueurs. On demande à chaque joueur de se connecter. On crée un compte pour chaque nouveau joueur. On mélange chaque type de carte. On choisit de chaque type de carte une carte et on la met dans l'enveloppe. On distribue le reste des cartes sur les joueurs. Supposons que Alexis Dupont et Maria Larousse. Chaque joueur, à tour de rôle, lance un dé et se déplace sur le plateau. Alexis Dupont a commencé, il lance le dé. Il a eu comme valeur 5. Il se déplace sur le plateau de 3 cases et rentre dans un endroit absorbant. Il formule son hypothèse. Maria Larousse prend la main et fait les mêmes étapes, etc.

Extensions : 5 points

Le jury évaluera le CDC de base sur tous les points sus-mentionnés, tout manquement ou réalisation trop approximative sera pénalisé au barème de 15 points prévu. D'autre part il est certainement possible de réaliser ce CDC avec une approche procédurale et sans utiliser les mécanismes objets du C++ et ses bibliothèques. Vous devrez convaincre votre jury lors de la soutenance que votre architecture est bien « orientée objet », par exemple en présentant un diagramme de classe lisible et cohérent avec les objectifs du projet et effectivement implémenté dans le code source C++. Le barème des extensions sur 4 points est séparé : vous ne pouvez pas compenser des manquements au CDC de base en collectionnant les extensions au-delà de 4 points. Il y a une note sur 15 d'une part et une note sur 4 d'autre part. Le jury se réserve le droit de ne pas considérer les extensions si le CDC de base est trop déficient.

La liste suivante n'est pas exhaustive, il n'est pas nécessaire de réaliser toutes ces idées pour obtenir les 4 points d'extension, discutez avec votre chargé de TP des autres idées que vous pourriez avoir ! Les points (ou ½ points) accordés par le jury dépendront de la difficulté et de la qualité de réalisation. On peut valoriser les efforts « architecture logicielle » pertinents... à vous de les faire valoir.

Console améliorée : en utilisant « gotoligcol » ou une bibliothèque plus évoluée comme PDcurses...

Graphismes : en utilisant la bibliothèque Allegro, SDL (sous réserve...) ou en attendant un code complémentaire qui permettra de faire de l'affichage graphique simple. En attendant, développez une version console parce que c'est ce qui est demandé au CDC de base.

Moteur de jeu paramétrable par fichier : Les différents aspects des réglages du code sont accessibles de façon centralisée et accessible à un game-designer (non programmeur), par exemple le nombre de cartes des decks en début de partie, le nombre de points de vie des joueur etc... toutes ces valeurs peuvent être réglées sans recompiler. De façon comparable, pouvoir ajouter de nouveaux modèles de cartes créatures « standard » sans recompiler. A défaut de paramétrage par fichier vous pouvez réfléchir aux approches permettant de centraliser vos valeurs constantes plutôt que de les disperser dans le code.

Opposant joué par l'ordinateur : On peut choisir de jouer seul contre 2 joueurs virtuels. On doit pouvoir visualiser leurs hypothèses, répondre si on possède une des cartes demandées, gérer leurs déplacements. L'intelligence artificielle de chaque joueur se limitera à émettre des hypothèses aléatoires, sans redemander un lieu (ou autre) déjà demandé.

Ajout de difficultés : des cases spécifiques (cases « mystères ») sont disposées sur le plateau et permettent d'obtenir des « bonus » (aller où on le souhaite, rejouer, faire une hypothèse, etc) ou des « malus » (passer un tour, montrer une carte à tout le monde, etc).

Annexe et Consignes générales

- En section [Annexes : ressources externes](#) de la page campus [Cours : Projet d'informatique 3 \(campusonline.me\)](#), vous trouverez les annexes suivantes de code en mode console :

- Table ANSI des caractères semi-graphiques
 - La manipulation du curseur sous Windows avec la fonction *gotoligcol*
 - La gestion de la couleur sous Windows avec la fonction *color*
- Dans le cas de programme graphique, il est autorisé d'utiliser des graphismes tiers trouvés sur Internet, si possible libres de droits (*Creative commons...*) ou avec *Copyright*, dans ce dernier cas votre logiciel doit rester strictement dans le cadre scolaire (citation dans un travail d'étude) et ne doit pas être publié. Dans tous les cas les sources tierces doivent être rigoureusement citées dans un fichier **licence.txt** dans votre archive.
- Il est autorisé d'utiliser/adapter des codes sources du cours, ainsi que des **extraits** de code source libre de droit (*public domain*, GPL, LGPL, exemples sur [cplusplus.com](#) ...) en précisant ces emprunts. Les commentaires initiaux de ces codes sources empruntés doivent être mis à jour par vos soins pour refléter leur utilisation dans votre contexte (appropriation correcte des codes tiers utilisés). En soutenance vous devez être capable d'expliquer chaque ligne de code de votre projet déposé.
- Il n'est pas autorisé d'utiliser un « *framework* » haut niveau de type moteur de jeu.

Il n'est pas autorisé de copier massivement un code tiers.

En dehors des exemples du cours et des codes officiellement fournis, toute détection de copie massive de code tiers sera considéré comme plagiat et très

sévèrement sanctionné, avec circonstances aggravantes si

- l'emprunt n'est pas cité
- l'emprunt vient du travail d'une autre équipe d'étudiants de l'ECE pour ce même projet
- il y a tentative de dissimulation (maquillage de code...)

- Le code source doit être présentable : bien indenté, raisonnablement aéré et commenté.

Pour rappel, un bon commentaire ne paraphrase pas le code (// nc prend la valeur 10) mais explique à quoi sert le code (// Initialisation du nombre de créatures). L'ensemble doit

être structuré : classes regroupées dans des .cpp et .h etc... Exceptionnellement si votre approche implique de dériver une classe par modèle de carte on peut accepter que plusieurs classes cohabitent dans un même .h (par exemple toutes les classes créatures dans creature.h). Il n'est pas évident que de dériver une classe par modèle de carte soit une bonne approche (les classes ont des attributs...).

Le projet est évolutif, c'est à dire que différentes versions peuvent être proposées, allant de la plus simple à la plus compliquée, de façon à ce que tous puissent, à la fin du semestre, proposer un logiciel en état de fonctionnement.

Toute autre fonctionnalité que vous pourriez ajoutée sera la bienvenue et sera prise en compte lors de la notation. L'utilisation des notions d'héritage, de conteneurs, de classe abstraite, d'interface ou d'ensemble seraient nécessaires. L'aspect graphique du logiciel peut être réalisé en utilisant Allegro.

Personnage du jeu

Les personnages du jeu qui peuvent disparaître sont de trois types : Enseignants (vacataires ou permanent), enseignant chercheur et personnel du service scolarité.

Lieux accessibles

Les lieux dans lesquels le jeu se déroule sont multiples. Ça peut être par exemple, une salle de cours, un amphi, un bureau, l'ascenseur ou la cuisine.

Cartes du jeu

Les différentes cartes indices sont de trois types : cartes suspects , cartes armes et cartes lieux.

Version 1

Un plateau de jeu simplifié apparaît, c'est à dire comprenant 9 lieux accessibles par n'importe quel joueur. La gestion des déplacements (dés, cases) n'est pas

demandée. On se limite à 3 joueurs qui joueront chacun à leur tour. La solution de l'énigme aura été faite de manière aléatoire par le logiciel parmi 9 lieux possibles et deux listes de 6 personnages/armes/sujets/outils/matières suivant votre scénario. Chacun des joueurs aura également 6 cartes distribuées de manière aléatoire (prévoir un scénario pour que les joueurs ne voient pas les cartes de leurs adversaires en écrivant dans un fichier par exemple au lieu d'afficher sur la console).

Le joueur qui émet une hypothèse le fait oralement et les autres lui répondent (si possible).

Le joueur qui pense avoir la solution le signale (clic sur au bouton ou autre) et saisit son hypothèse. Le logiciel valide l'hypothèse (le joueur gagne) ou non (il est éliminé).

Version 2

Comme la version 1, mais :

- on peut choisir le nombre de joueurs.
- on gère les cases entre les lieux ainsi que la génération de nombres (lancement des dés). On ne peut donc pas forcément atteindre le lieu désiré.
- l'hypothèse formulée par un joueur doit être faite de manière graphique (on clique sur des images par exemple).
- Le lieu de départ est celui où le joueur doit se trouver s'il pense connaître la solution.

Version 3

Comme la version 2, mais des cases spécifiques (cases « mystères » à ***
744

) sont disposées sur le plateau et permettent d'obtenir des « bonus » (aller où on le souhaite, rejouer, faire une hypothèse, etc) ou des « malus » (passer un tour, montrer une carte à tout le monde, etc).

Version 4

Comme la version 3, mais on peut choisir de jouer seul contre 2 joueurs virtuels. On doit pouvoir visualiser leurs hypothèses, répondre si on possède une des cartes demandées, gérer leurs déplacements. L'intelligence artificielle de chaque joueur se limitera à émettre des hypothèses aléatoires, sans redemander un lieu (ou autre) déjà demandé.

Version 5

Comme la version 4, mais des animations sont ajoutées : les personnages se déplacent sur le plateau, les dés sont lancés, des objets apparaissent (par exemple le personnage ou l'arme après une hypothèse).