

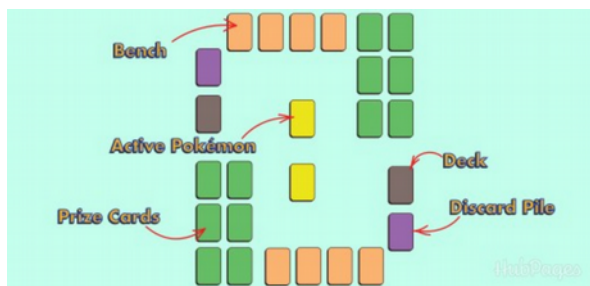
# Ecemon



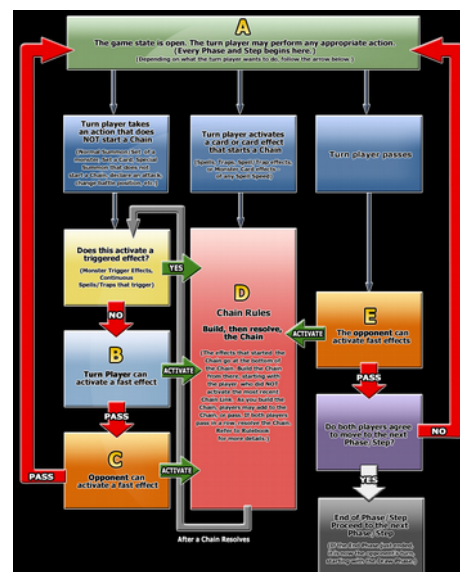
Pokémon



Yu-Gi-Oh!



Terrain de Pokémon



Logigramme Yu-Gi-Oh!

## → Principe du jeu

Le jeu qui vous est demandé s'inspire des jeux de cartes à jouer et collectionner type Pokémon. Outre les « card games » Pokémon et Yu-Gi-Oh! éponymes des mangas bien connus, il existe beaucoup d'autres collections de cartes :

Magic the Gathering, Cardfight!!Vanguard, WeissSchwarz, Dragoborne, Future Card Buddyfight ...

On ne va bien sûr pas vous demander de peupler votre jeu de milliers de créatures ni de développer un système de jeu aussi complexe. Ecemon, votre projet, sera donc une version « allégée » de ce genre de jeu. On imagine que personne avant vous n'avait eu l'idée de ce genre de jeu. Vous êtes une équipe de 3 (ou 2) ingénieurs innovants et vous devez, en 1 mois ouvrable, tout en continuant votre job (n'oubliez pas les autres matières !) produire une **maquette** qui saurait convaincre un éditeur de jeu ou des investisseurs de la viabilité du concept.

La description des règles qui suit est indicative. Il n'est pas évident qu'avec ces simplifications le jeu soit encore passionnant. On pourrait le voir comme une sorte de Bataille (le jeu de carte) améliorée. **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 Pokémon : un bonus d'évaluation viendra récompenser les équipes qui auront fait l'effort de l'originalité. Pensez à des univers différents des mangas ou de l'heroic fantasy, par exemple le monde des Dieux au temps de la Grèce antique (Zeus contre Hades), ou les animaux du désert (mangouste contre cobra), ou les compositeurs de musique (Beethoven contre Wagner) etc...

Pour illustrer la présentation des règles par des exemples précis, je propose un jeu basé sur l'univers du Western, que j'appellerai **Westmon**. Les règles abstraites à respecter sont en noir, les exemples qui viennent illustrer sont en bleu.

<b>Cahier des charges de base : 15 points</b>
---

*Autour des matchs*

Chaque participant au jeu est enregistré avec son nom. Il possède une collection, un ensemble de cartes du jeu, en quantité indéfinie (des dizaines, des centaines...) mais au moins assez pour constituer un « deck ». Un deck est une sélection (sélection faite par le joueur) d'une quantité bien précise de cartes (toujours la même) qui permet de jouer un match (faire une partie quoi) contre un autre joueur. Avant chaque match le joueur peut (re)faire un nouveau deck à partir de sa collection.

Votre programme permet de créer de nouveaux joueurs, et de leur créer une collection de départ en choisissant 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 de successivement sélectionner leur deck à partir des cartes de leur collection qui est sauvee sur fichier (pas à retaper à chaque exécution). Ensuite la partie commence.

**Exemple Westmon :** On a lancé le programme, on a retrouvé les joueurs Joe et Ursula enregistrés lors d'une précédente session. On récupère bien leur collection (qu'on peut visualiser). On peut leur acheter de nouvelles cartes (avec les sous de leurs parents, à gérer...) parmi toutes les cartes possible du jeu. On peut créer un nouveau joueur Billy, et on lui fabrique une collection initiale sur le même principe. On peut alors faire un match entre Billy (le petit nouveau) et Ursula (la vétérane), le match commence après qu'ils ont sélectionné chacun un deck de 21 cartes depuis leur collection.

### Début de match et enjeu

En principe le 1<sup>er</sup> joueur ne voit pas le deck du 2<sup>ème</sup> ni l'inverse mais évidemment comme votre programme est monoposte ce n'est qu'une question de convention (chacun se retourne quand l'autre sélectionne son deck). Au début du match les cartes sont battues (le programme randomise l'ordre du deck de chacun des 2 joueurs). Le joueur qui commence est tiré au hasard. Ce sont les seuls aspects aléatoires dans les règles de base. Le deck est posé devant chaque joueur face cachée et constitue sa « pioche ». Le deck fonctionnera toujours comme une file, en prenant la carte du haut (piocher) et en remettant (éventuellement) une carte par le dessous.

La 1<sup>ère</sup> carte du Deck (sur le dessus donc) est prise et placée à part, face cachée : c'est l'enjeu. Le gagnant de la partie gagne cette carte qui intégrera sa collection (ceci pourra être vérifié après la partie, en consultant la collection du gagnant) et le perdant n'aura plus cette carte dans sa collection. Les cartes ne sont évidemment pas uniques, une même collection peut contenir plusieurs exemplaires d'un même modèle de carte. Eventuellement en cours de partie certains coups spéciaux peuvent permettre de voir la ou les carte enjeu (de soi et/ou de l'adversaire), ou bien de changer la carte enjeu (avec une carte à soi, ou de permuter entre la sienne et celle de l'adversaire...).

Exemple Westmon : L'enjeu de Billy était une carte spéciale « Shoot you in the back », il a perdu la partie, il perd donc cette carte de sa collection et Ursula gagne cette carte dans sa collection. Billy avait 3 cartes « Shoot you in the back » dans sa collection, il ne lui en reste plus que 2. Après la partie (de retour au menu général) il décide d'acheter 2 nouveaux exemplaires de « Shoot you in the back », il en a désormais 4 exemplaires (mais il a aussi moins de sous).

### Les cartes

Il y a 3 types de cartes : les cartes **Energies**, les cartes **Créatures**, les cartes **Spéciales**.

#### → Cartes Energies (synonymes Terrains/Lieux/Cosmos/Dominantes/Domaines)

Les éléments du jeu se structurent avec au moins 4 « domaines » distincts. Par exemple on peut proposer les 4 éléments Feu|Air|Eau|Terre, ou bien 4 règnes Animal|Végétal|Mécanique|Imaginaire ou bien 4 investissements Foncier|Actions|Métaux précieux|Art ou juste Rouge|Vert|Bleu|Jaune. Si possible ces 4 types d'énergies s'intégreront harmonieusement avec la logique de votre univers de jeu, par exemple des créatures Taupinou et Ver-de-terreur seront associées au domaine Terre...

Exemple Westmon : On aura les domaines

- Saloon : urbain, vision, couteaux, jeu, arnaques, attaques de banques...
- Desert : grands espaces, vitesse, fusils, attaques de trains, de diligences, de banques...
- Sierra : montagnes, endurance, pièges, trafiques, alcools frelatés, gangs, attaques de banques...
- Corral : bétail et chevaux (vol de), force, bagarres, règlements de compte, attaques de banques...

Il y aura au moins un modèle de carte énergie par domaine. Chaque carte énergie apporte 1 point d'énergie qui se cumule avec les autres du même domaine. Les cartes énergies disposent d'un nom et d'une description, mais ces détails ne sont que de l'habillage, 2 modèles d'un même domaine sont équivalents en terme de mécanique de jeu. A un moment du jeu chaque joueur a sur la table un certain nombre de cartes énergies de chaque domaine.

Exemple Westmon :

- Saloon : cartes « Puccino's barber » « Ace in sleeve » « Ringo's brothel »
- Desert : cartes « Vulture » « Last drop of booze » « Snake river »
- Mountain : cartes « Sierra Nevada » « Rockies » « Gold !!!! » « Short fuse dynamite »
- Corral : cartes « Tombstone, Arizona » « A horse, a hat, a colt »

A un moment de la partie Ursula a sur la table une carte Rockies, une carte Gold !!!!, et trois cartes Vulture. L'interface affiche le résumé suivant concernant les points d'énergie d'Ursula:

Saloon 0 Desert 3 Mountain 2 Corral 0

On peut demander un affichage du détail et obtenir la liste des cartes énergies posées, avec leur descriptions, mais pour la mécanique de jeu seule la somme des énergies dans chaque domaine compte.

### → Cartes Créatures (synonymes Entités combattantes/Pièces)

Ce sont ces cartes qui vont s'affronter et en général déterminer l'issue du match. A un moment donné du match, chaque joueur a au plus une créature en jeu. Un match propose donc une succession de duels entre créatures à mesure que des cartes créatures sont tirées de la pioche. Vous proposerez au moins 5 modèles de cartes créatures. Chaque créature a un nom et une description.

Exemple Westmon : « Calamity Jane » « Butch Cassidy » « Bob Dalton » « Belle Starr » « Jesse James »

Chaque carte créature possède un certain nombre de points de vie initial (Life Points : **LP**), valeur qui s'érode à mesure qu'elle reçoit des attaques de la créature ennemie. Arrivé à 0 la carte est « détruite » et rejoint le cimetière (graveyard), face visible. Elle possède également une liste d'au moins 2 attaques de niveaux de puissance croissant nécessitant des points d'énergie (cartes énergies posées). Chaque attaque est caractérisée par un nom, par le nombre et la nature des points d'énergies nécessaires pour la lancer et sa puissance en points de dégât (Hit Points : **HP**). Certaines attaques peuvent accepter des points d'énergie de domaine quelconques (Any).

Exemple Westmon : La carte Calamity Jane a 5 points de vie initiaux quand elle arrive sur le terrain. Elle a pour attaques :

- |                     |     |     |  |
|---------------------|-----|-----|--|
| - « Horse kick »    | C   | 1HP | nécessite 1 point Corral                     |
| - « Knife stab »    | SA  | 2HP | nécessite 1 point Saloon et 1 point Any      |
| - « Sharp shoot »   | SM  | 3HP | nécessite 1 point Saloon et 1 point Mountain |
| - « Seduce & kill » | SSA | 4HP | nécessite 2 points Saloon et 1 point Any     |

Si Billy a une carte Calamity Jane sur le terrain et qu'il a 1 point Saloon et 1 point Corral (par exemple une carte Ringo's brothel et une carte Tombstone Arizona) alors il peut lancer une attaque Horse kick (1C) ou Knife stab (1S et 1A, le point Corral allant pour le Any), mais pas une attaque Sharp shoot (il manque un point Mountain) ni Seduce & kill (il manque un 2ème point Saloon)

L'interface précisera les attaques possibles sur la base des points disponibles. C'est ensuite au joueur de préciser l'attaque qu'il souhaite faire, ou aucune attaque (certaines cartes spéciales pourraient retourner les points de dégât contre l'attaquant...). Dans la version de base, la créature attaquée prend directement les points de dégât occasionnés par la créature attaquante. Arrivée à 0 point de vie elle part au cimetière. Les points de dégât excédentaires impactent les points de vie du joueur.



→ Cartes Spéciales, synonymes Trainer cards(pokémon)



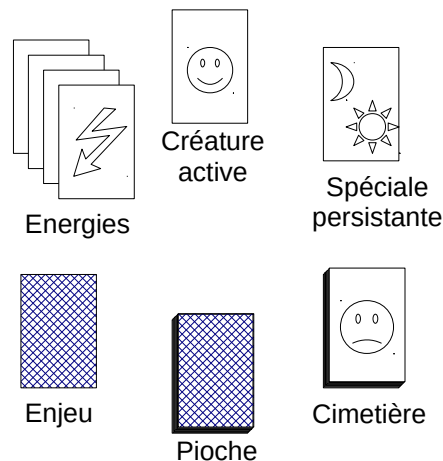
Ce sont des cartes qui permettent d'altérer la mécanique normale du jeu (décrite ci après) en autorisant le joueur à voir/choisir dans sa pioche, à jouer 2 coups d'affilée, à reprendre des cartes du cimetière, à modifier les caractéristiques des créatures (multiplicateurs, pénalités...), à voir/permuter/jouer/changer les cartes enjeux etc... Chaque carte spéciale a un nom et une description de son effet. Elle précise également son cycle de vie : usage unique (valable que sur le tour de jeu) ou persistant (elle reste sur la table et fait effet tant qu'elle y est) si elle peut être remise sous la pioche après usage (recyclable) ou dans le cas où on veut la passer (Refus, par exemple pour ne pas remplacer la carte spéciale persistante actuellement posée). Pour l'intérêt du jeu il est préférable que les cartes spéciales puissantes soient « à double tranchant » et puissent pénaliser l'utilisateur de l'effet dans certaines circonstances... Certaines cartes spéciales peuvent nécessiter des sacrifices pour être activées. Il vous est demandé au moins 6 modèles de cartes spéciales, montrant différentes manipulations de cartes sur le terrain de jeu.

Exemple Westmon (on a ajouté une 7<sup>ème</sup> et une 8<sup>ème</sup> spéciales pour le plaisir) :

- « Snakepit ! »                                  Persistante, Recyclable, Refus → repart en pioche  
*Diminue de 1 les points de vie des 2 créatures en jeu*
- « Not quite dead »                              Usage unique, Non recyclable, Refus → part en cimetière  
*Si une créature est au sommet de votre cimetière elle lance une attaque comme si elle était active*
- « Marshall law »                                Usage unique, Recyclable, Refus → repart en pioche  
*Les 2 créatures en jeu sont pendues et partent au cimetière, la pioche du joueur est rebattue*
- « Saloon's mirror »                             Usage unique, Recyclable, Refus → repart en pioche  
*Permet de consulter (voir) l'intégralité de la pioche de l'adversaire*
- « Ghost of the lost gold mine »            Usage unique, Non recyclable, Refus → part en cimetière  
*Sacrifiez 1 carte énergie Desert ou Mountain et récupérez 1 créature du cimetière vers la pioche*
- « Indian trouble »                              Persistante, Recyclable, Refus → repart en pioche  
*Les 2 joueurs perdent chacun une carte énergie de leur choix (qui part au cimetière)*
- « Settling scores »                             Usage unique, Non Recyclable, Refus → part en cimetière  
*Sacrifiez votre créature active, en échange vous envoyez une carte ennemie au choix au cimetière*
- « Shoot you in the back »                   Usage unique, Recyclable, Refus → repart en pioche  
*Augmentez l'attaque de votre créature de 1 point de dégât (à énergie égale)*

### Le jeu

Les cartes créatures font des attaques directes, le jeu est une succession de duels entre créatures, les cartes énergies rendent possible les attaques (plus ou moins puissantes) des créatures, les cartes spéciales permettent au joueur d'altérer la mécanique par défaut du jeu, de « changer les règles », de créer des surprises, des retournements de situation... A chaque tour de jeu une nouvelle carte de la pioche est tirée. Le joueur peut décider de la jouer ou pas. Si la carte est jouée elle peut (dans le cas d'une carte créature ou spéciale persistante) remplacer la carte active correspondante (remplacer la créature en jeu ou la spéciale persistante). Les cartes refusées (non jouées) ainsi que les cartes remplacées peuvent repartir sous la pioche ou au sommet du cimetière, selon des règles explicites qui assurent un minimum d'intérêt au jeu mais garantissent des parties courtes (si possible 2 parties faisables en 12 minutes). Les cartes créatures tuées (points de vie à 0) vont au sommet du cimetière. Les points de dégât excédentaires vont se soustraire aux points de vie du joueur. Un joueur dont les points de vie arrivent à 0 a perdu. Quand le joueur dont c'est le tour de jeu n'a pas de créature posée (parce qu'elle vient de se faire tuer) il doit dérouler sa pioche (prendre au sommet, remettre dessous, sans jouer) jusqu'à en trouver une qu'il doit alors obligatoirement poser. Un joueur qui n'a plus de créature dans sa pioche a perdu. Au final il y a 6 « zones » devant chaque joueur :



### Résumé du CDC de base

- 4 domaines (types d'énergies...)
- 4 modèles de cartes énergie (une par domaine au moins)
- 5 modèles de cartes créatures avec chacune au moins 2 attaques
- 6 modèles de cartes spéciales permettant d'altérer les mécanismes par défaut du jeu
- Nombre indéfini de joueurs, pouvoir ajouter des joueurs...
- Pouvoir acheter des cartes pour faire évoluer les collections des joueurs
- Les données des joueurs et leur collections sont sauveées sur fichier (on les retrouve)
- Organiser des matchs entre joueurs au choix
- Phase de sélection des decks (à partir des collections de chacun)
- Battre le deck. Tirage de l'enjeu.
- Partie proprement dite jusqu'à perte d'un des 2 joueurs (ou match nul, exceptionnel)
- Le programme impose le respect des règles, détecte la fin de partie, ne plante pas
- Les cartes disent ce qu'elles font et font ce qu'elles disent (descriptions consultables)
- Attribution de la carte enjeu du perdant à la collection du gagnant
- Interface mode console pure, std::cout et std::cin, présentations séquentielles des infos

<b>Extensions : 4 points</b>
------------------------------

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 du tout 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 (livrable 13/11) 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 : simplement en choisissant au hasard dans les coups légaux, ou en essayant de donner un peu d'intelligence en appliquant des heuristiques.

Ordinateur contre ordinateur & statistiques : il est difficile de régler les jeux de cartes à collectionner, souvent de nouvelles cartes ou combinaisons trop puissantes créent des déséquilibre (tout le monde achète la carte « Super dynamite » et il n'y a plus que ça dans les decks). Compte tenu du temps limité du projet on ne s'attend pas à ce que vous ayez trouvé des réglages très fins, mais tout effort **rationnel** pour améliorer le traitement de ce problème sera apprécié. Jouer des centaines de parties jusqu'à 5 heures du matin ne compte pas comme un effort rationnel. Faire jouer à votre programme des centaines de parties en mode automatique et tirer des statistiques serait au contraire bien reçu.

Défenses : Les créatures ont une liste d'attaques mais aussi une liste de défenses qui viennent limiter les points de dégât reçus. Attention de ne pas faire trop compliqué à expliquer/présenter en démo !

Aptitudes créatures spéciales : certaines créatures peuvent disposer d'attaques ou de défenses spéciales nécessitant des actions particulières (par exemple un bonus d'attaque sur le sacrifice d'une carte énergie d'un domaine particulier).

Types supplémentaires : vous pouvez ajouter plus de types d'énergie, de créatures, de spéciales, que les minimas demandés dans le CDC de base. Discutez avec votre chargé de TP de la valorisation en fonction de ces ajouts (ce n'est pas forcément un point de gagné par créature en plus !) parce que ce qui est intéressant (de notre point de vue de prof d'info) c'est de mettre en place les mécanismes de gestion pour N catégories/types, pas forcément de les remplir. Démo pas trop compliquée SVP.

Des cartes spéciales particulièrement acrobatiques : les cartes spéciales « changent les règles » du jeu, d'un point de vue programmatique il est plus intéressant de voir comment vous intégrez ces cas particuliers harmonieusement (ou pas !) dans la logique de départ.

#### Des cartes énergies spéciales

Exemple Westmon : on décide d'ajouter un nouveau modèle de carte énergie « Joker » qui permet d'alimenter en énergie de n'importe quel domaine. Comme c'est une carte très puissante on décide d'instaurer la règle qu'il ne peut y en avoir qu'une dans un deck initial. Mais comme l'univers de Westmon est composé de bandits et de truands on autorise le joueur à tricher : à la constitution du Deck il est possible de prendre plus d'une carte Joker. Tant que l'adversaire ne voit pas la supercherie durant la partie le joueur peut bénéficier d'une plus grande probabilité de tirer cette carte et d'en finir rapidement. En revanche si durant le jeu 2 cartes Joker apparaissent dans votre pioche, l'adversaire a le droit de vous tirer sur-le-champ une balle entre les 2 yeux (défaite immédiate).

Autres idées ? Discutez en avec votre chargé de TP...

<b>Originalité et cohérence : 1 point (+1 point bonus exceptionnel)</b>
---

Le jury appréciera l'originalité de votre proposition. Essayez de déplacer la mécanique de jeu imposée dans un univers différent, adaptez le récit et le [gameplay](#) à cet univers pour en faire un tout cohérent et harmonieux. Dans le point d'originalité serait aussi compté (sans caractère obligatoire) un travail de design graphique personnel, valorisé par rapport à la simple utilisation de graphismes empruntés (voir ci-dessous).

Très exceptionnellement 1 point supplémentaire de bonus est laissé à l'appréciation du jury (talent prodigieux à la palette, images de synthèse sensationnelles...) Attention ceci ne doit pas vous faire oublier qu'il s'agit avant tout d'un projet de programmation orienté objet C++ : chaque membre de l'équipe doit participer au code, pouvoir expliquer en détail les parties sous sa responsabilité et de façon synthétique le reste du code source, sans se retrouver perdu.

<b>Consignes générales</b>
----------------------------

- 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 à faire en trinôme ou binôme, à l'intérieur d'un même groupe de TP. Les consignes de constitution des équipes de projet seront communiquées très rapidement.

Vous réaliserez ce projet en respectant la méthode de conception objet qui part d'une conception par diagramme de classe avant l'implémentation. Evitez le bricolage. Tenez à jour, dès le départ de la conception, un « cahier de laboratoire ».

La date de soutenance les modalités de rendu vous seront précisées rapidement.

Robin Fercoq et toute l'équipe encadrante,

06/11/2017

Version 1, sujette à ajustements mineurs en cas de besoin.

