



Initiation à la Programmation C

Travaux Pratiques - L2

Projet 2024-2025



Darkest C Dungeon

À propos

Ce projet crée une version simplifiée d'un jeu de type *Darkest Dungeon*. Vous êtes équipé d'un ensemble de personnages qui doivent affronter des ennemis lors de combats au tour par tour, et progresser ainsi jusqu'au donjon final. . . Au fur et à mesure de votre avancée, vous gagnez des accessoires que vos personnages pourront équiper pour être plus forts. De nouveaux personnages pourront également rejoindre votre équipe.

Techniquement, on imposera l'utilisation de listes chaînées et de structures pour stocker les personnages, leurs statistiques et les accessoires. Il est à réaliser en **binôme** (ou exceptionnellement seul.e, contactez vos enseignants dans ce cas) et à développer par étapes successives. Le rendu consiste en votre code et le compte-rendu de projet. La date limite de rendu est le mardi 14 janvier. Une soutenance à l'université aura lieu le 17 janvier (détails à la fin de l'énoncé).

1 Description d'une partie

Chaque personnage a un nom, une classe et des statistiques correspondant à sa classe : points d'attaque, points de défense, points de vie maximum, points de restauration. Il a aussi des statistiques qui évolue au cours des combats : points de vie et points de stress. On démarrera avec 2 personnages disponibles, d'autres rejoindront la partie au fur et à mesure des combats. A l'annonce d'un combat, le joueur doit choisir les personnages qui combattront, entre 1 et 3 maximum. Il a la possibilité d'équiper chaque personnage participant au combat d'accessoires qui vont booster certaines statistiques (plus de points d'attaque ou de défense par exemple).

La partie se déroule en 10 combats, et chaque combat se déroule au tour par tour contre un ennemi qui possède un niveau allant de 1 à 10, tel qu'un ennemi de niveau $k \in [1..10]$ est affronté au k -ième combat. Le niveau de l'ennemi correspond à des statistiques de points d'attaque, points de défense et points de vie. Ces statistiques augmentent avec le niveau de l'ennemi. Pour vaincre un ennemi au cours d'un combat, il faut amener ses points de vie à 0.

Au cours d'un tour, le joueur choisit une action par personnage : attaquer l'ennemi ou restaurer des points de vie (à soi-même ou à un camarade participant au combat), ou se mettre en position de défense pour réduire les dommages causés lors de l'attaque ennemie. Lorsqu'on attaque l'ennemi, ses points de vie diminuent (voir le calcul plus bas).

Après le tour des personnages, c'est au tour de l'ennemi. Il peut soit effectuer une attaque physique sur un de vos personnages, diminuant ainsi ses points de vie, soit lancer une attaque de stress. Lorsqu'un personnage tombe à 0 point de vie, on dit qu'il meurt, et il est supprimé du combat en cours et de la partie. Une attaque de stress augmente la barre de stress du personnage. Celui-ci deviendra inutilisable (mais pas mort) lorsqu'il atteindra un niveau de 100% de stress.

Si lors d'un combat tous les personnages du joueur meurent, la partie est perdue et le jeu s'arrête. A la fin d'un combat gagné par les personnages, le joueur remporte un accessoire et de

l'or. Chaque accessoire possède des statistiques : des points bonus d'attaque et de défense, des points de vie bonus, et des points de restauration bonus. Lorsqu'on équipe un accessoire sur un personnage, celui-ci reçoit les points de l'accessoire en addition des points de ses statistiques. Par exemple : en équipant un accessoire à 10 points bonus d'attaque, on ajoute 10 points d'attaque au personnage.

Pour les points de vie, il faudra distinguer les points de vie actuels du personnage et les points de vie maximum du personnage. Un sort de restauration lancé lors d'un combat sur le personnage augmentera ses points de vie actuels dans la limite de ses points de vie maximum. Par exemple, si le personnage est à 10 points de vie sur un maximum de 13 et qu'on le soigne pour 5 points, il sera à 13 points. Un accessoire peut augmenter le nombre de points de vie maximum (voir détails plus bas).

Lors d'un combat, les tours des personnages et de l'ennemi s'enchaînent. La fin de combat a lieu si l'ennemi est défait ou que tous les personnages combattant sont morts. Le joueur reçoit un accessoire en cas de victoire, et un nouveau personnage rejoint le groupe après les niveaux 2, 4, 6 et 8. Il pourra alors participer aux combats et équiper des accessoires.

Pour augmenter les points de vie d'un personnage qui a subi des dommages, on pourra utiliser une action de restauration s'il participe à un combat, ou bien en dehors des combats on pourra l'envoyer au sanitarium récupérer des points de vie pendant que d'autres combattent. De même on pourra réduire le niveau de stress des personnages en les envoyant récupérer à la taverne pendant que d'autres combattent. Entre les combats, on aura aussi accès à la roulotte, un magasin dans lequel on peut dépenser son or pour acheter des accessoires supplémentaires.

La partie est gagnée lorsque le joueur vainc le dernier ennemi, c'est-à-dire au niveau 10.

2 Description des éléments du jeu

2.1 Les classes

On pourra définir les classes de personnages comme des structures constantes ayant les attributs suivants.

- Points d'attaque, notés **att**, qui contrôlent sur la puissance des attaques
- Points de défense, notés **def**, qui peuvent réduire les dommages causés par l'ennemi,
- Points de vie max, notés **HPmax**,
- Points de restauration, notés **rest**, qui permettent de soigner un camarade.

On impose les 4 classes suivantes. Vous pourrez changer les valeurs des attributs pour tester et adapter le niveau de difficulté du jeu aux ennemis que vous aurez créés.

Classes	att	def	HPmax	rest
Furie	13	0	20	0
Vestale	3	0	20	10
Chasseur de primes	7	3	25	3
Maître chien	10	6	17	5

On affichera au démarrage de la partie le tableau des classes et leurs statistiques.

2.2 Les personnages

On partira avec 2 personnages et on déblocquera un nouveau personnage tous les 2 niveaux en cas de victoire (aux niveaux 2, 4, 6, 8), ce qui fait 6 personnages au total. On pourra faire

un type structuré de personnage, incluant leur nom, leur classe (et on pourra rappeler leurs statistiques de classes), leurs points de vie actuels **HP** (différents de **HPmax**), leur points de stress **str**, leurs accessoires numéro 1 et numéro 2 (qui peuvent être vides), et le nombre de combats auxquels chaque personnage a participé **nbcomb**.

Au démarrage de la partie, on affichera les 2 personnages de départ avec leur nom et leur classe. Vous pouvez choisir avec un tirage aléatoire la classe de ces 2 personnages, mais il faudra qu'elles soient différentes l'une de l'autre (par exemple une furie et une vestale mais pas 2 furies). On affichera par exemple sur le terminal

Nom	Classe	att	def	HP/HPmax	rest	str	nbcomb
Boudicca	Furie	13	5	20/20	0	0	0
Junia	Vestale	3	5	20/20	10	0	0

On peut choisir une autre méthode d'affichage que le tableau proposé, par exemple un affichage en ligne tout simple :

Boudicca

Classe Furie, att : 13, def : 5, HP/HPmax : 20/20, rest : 0, str : 0, nbcomb : 0

Junia

Classe Vestale, att : 3, def : 5, HP/HPmax : 20/20, rest : 10, str : 0, nbcomb : 0

Au départ du jeu, les 2 personnages ont tous leurs points de vie : **HP=HPmax**, un niveau de stress nul : **str=0**, et n'ont pas encore effectué de combats : **nbcomb=0**.

Aux niveaux 2, 4, 6 et 8, on introduit un nouveau personnage, lui aussi avec **HP=HPmax**, **str=0**, **nbcomb=0**. Il faudra que ces nouveaux personnages soient des 4 classes différentes. Par exemple, une vestale au niveau 2, un maître chien au niveau 4, un chasseur de primes au niveau 6 et une furie au niveau 8. Lorsqu'on gagne un combat aux niveaux 2, 4, 6, 8, on affichera le nouveau personnage avec son nom et sa classe, comme montré au-dessus pour les 2 personnages de départ.

Les caractéristiques des personnages peuvent être modifiées grâce à des accessoires, ou à cause d'attaques ennemies lors des combats. Les attaques ennemies peuvent être

- soit physiques : elles réduisent les **HP**. Ceux-ci peuvent être restaurés par des sorts de restauration pendant les combats ou au sanitarium. Un personnage réduit à 0 **HP** meurt et disparaît de la partie. On pourra restaurer les **HP** soit par des sorts de restauration au cours de combats, soit en passant du temps au sanitarium.
- soit de stress : elles augmentent le niveau de stress du personnage. Lorsque son niveau atteint les 100 **str**, il devient inutilisable dans le combat en cours et ne pourra participer aux combats suivants que si son niveau de stress est réduit sous les 100 **str** en passant du temps à la taverne.

2.3 Les accessoires

On pourra définir un type structuré d'accessoire avec son nom et les bonus fournis,

- **attbonus** : des points qui s'ajouteront au **att** du personnage équipé,
- **defbonus** : des points qui s'ajouteront au **def** du personnage équipé,
- **HPbonus** : des points qui s'ajouteront au **HPmax** et **HP** du personnage équipé,
- **restbonus** : des points qui s'ajouteront au **rest** du personnage équipé.
- **strred** : des points qui diminueront les attaques de stress de l'ennemi.

On partira avec 2 accessoires disponibles, par exemple :

Nom	attbonus	defbonus	HPbonus	restbonus	strred
Pendentif tranchant	+5	+1	+0	+0	0
Calice de jeunesse	+0	+3	+5	+0	5

Lorsqu'un personnage meurt, ses accessoires sont détruits. Chaque accessoire ne peut être équipé que sur un personnage à la fois.

2.4 Les ennemis

On pourra définir un type structuré d'ennemi et définir 10 ennemis au total. Chacun aura son nom, son niveau, ses points d'attaque physique (**attenn**), ses points de défense (**defenn**), ses points de vie actuels (**HPenn**), et ses points d'attaque de stress **attstrenn**. Les ennemis ne se restaurent pas, donc ils n'ont pas de points de restauration et on n'a pas besoin de connaître leurs points de vie max.

Nom	Niveau	attenn	defenn	HPenn	attstrenn
Brigand	1	3	3	9	0
Squelette	2	6	4	13	10
Goule	3	8	8	16	20
Gargouille	4

2.5 Le sanitarium

Entre les combats, on peut envoyer des personnages au sanitarium pour récupérer des HP. Il s'agira d'une liste chaînée de personnages, pouvant accueillir entre 0 et 2 personnages. Un personnage qui est au sanitarium pendant que d'autres combattent récupère 7 HP par combat passé. Les personnages qui ne combattent pas mais ne vont pas au sanitarium ne récupère pas de HP.

2.6 La taverne

Entre les combats, on peut envoyer des personnages à la taverne pour diminuer leur **str**. Il s'agira d'une liste chaînée de personnages, pouvant accueillir entre 0 et 2 personnages. Un personnage qui est à la taverne pendant que d'autres combattent a son niveau de stress diminué de 25 **str** par combat passé. Il s'agit de la seule façon de réduire le niveau de stress des personnages.

2.7 La roulotte

Entre les combats, il est aussi possible d'acheter des accessoires à la roulotte. Il s'agit d'une liste chaînée d'accessoires avec un prix associé à chacun. On mettra des articles à 7 or, d'autres à 16, d'autres 35 et plus si vous le souhaitez. Les caractéristiques devront s'améliorer globalement avec le prix.

3 D roulement d'une partie

3.1 Mise en place d'un combat

Avant un combat, on affichera la liste des personnages disponibles avec leurs donn es. Les personnages disponibles seront stock s dans une liste cha n e. On pourra utiliser un des mod les d'affichage donn  en page 3. Pour l' nonc , on choisit le tableau

Personnages disponibles

	Nom	Classe	att	def	HP/HPmax	rest	str	nbcomb
1	Boudicca	Furie	13	5	13/20	0	70	4
2	Junia	Vestale	3	5	18/20	10	20	4
3	William	Ma�tre chien	10	15	17/17	5	45	2
4	Tardif	Chasseur de primes
5	Alhazred

L'ordre n'a pas d'importance, les num ros sont l  pour permettre au joueur de choisir ses combattants. Les combats 1   5 se font   au plus 2 combattants. Les combats suivants peuvent se faire   3 combattants. On donnera le choix de ne pas emmener le nombre maximal de combattants autoris s, par exemple si le joueur rentre la lettre N. Le combat ne peut se lancer qu'avec au moins 1 combattant. Exemple de prompt :

Choix du premier combattant : 3

Choix du second combattant : 1

Choix du troisi me combattant : N

On passe ainsi les personnages s lectionn s de la liste des personnages disponibles   une nouvelle liste cha n e pour les personnages combattant. Le m me combattant ne peut pas  tre s lectionn  plusieurs fois dans le m me combat.

Apr s la s lection des personnages, on passera   la s lection des accessoires, eux aussi stock s dans une liste cha n e. On affichera la liste des accessoires disponibles et on demandera au joueur de faire ses choix. Exemple d'affichage sous forme de tableau (on pourra faire   la place un affichage par ligne comme pour les personnages en page 3)

Accessoires disponibles

	Nom	attbonus	defbonus	HPbonus	restbonus	strred
1	Pendentif tranchant	+5	+1	+0	+0	0
2	Calice de jeunesse	+0	+3	+5	+0	5
3	Anneau de concentration
4	Manteau solaire

Accessoire 1 pour le combattant 1 : Pendentif tranchant

Accessoire 2 pour le combattant 1 : Calice de jeunesse

Accessoire 1 pour le combattant 2 : Anneau de concentration

Accessoire 2 pour le combattant 2 : Manteau solaire

On propose que les accessoires ajoutant des **HPbonus** se comportent de la mani re suivante. Ils s'ajoutent   **HPmax** et   **HP** lorsqu'ils sont  quip s. A la fin d'un combat, lorsqu'ils sont d s-

équipés, il faudra soustraire **HPbonus** à **HPmax** et à **HP**, sauf si la soustraction à **HP** amène le personnage à 0 **HP**. Dans ce cas, on imposera que le personnage soit réduit à 1 **HP**.

3.2 Déroulement d'un combat

Les combats se déroulent au tour par tour. On pourra faire commencer les personnages ou l'ennemi au hasard. Lors d'un tour du joueur, le jeu commence par rappeler les **HP** et **str** des personnages. On prendra dans cette section l'exemple de Boudicca en personnage 1, William en personnage 2 et Tardif en personnage 3.

Boudicca : 15 **HP**, 90 **str**

William : 12 **HP**, 35 **str**

Tardif : 9 **HP**, 55 **str**

3.2.1 Actions des personnages

Le jeu demande au joueur de choisir les actions de ses combattants l'un après l'autre. Chaque combattant bénéficie d'une action à choisir entre : attaquer (**A**), défendre (**D**), et restaurer (**R**).

- Attaque : on calcule le nombre de points total du combattant de la manière suivante. C'est le **att** du personnage plus les points bonus **attbonus** dûs aux accessoires. Notons **atttotal** ce nombre et **defenn** le nombre de points de défense de l'ennemi. Les dommages causés par l'attaque se calculent alors comme $(atttotal - defenn) * roll$ où **roll** est un nombre à deux décimales tiré aléatoirement entre 0.8 et 1.2, si $atttotal - defenn > 0$. Sinon, l'attaque cause simplement 1 point de dommage. Le nombre de points de vie de l'ennemi est alors amputé des points de dommage calculés.
- Restauration : Dans ce cas, le jeu demande quel joueur restaurer. Le nombre de points de vie **HP** gagnés par le joueur visé est le **rest** du joueur qui lance le sort plus les **restbonus** de ses accessoires. Le nombre de points de vie ne peut pas dépasser le **HPmax** du personnage.

Exemple de prompt avec un ennemi ayant 45 **HP** :

Action de Boudicca (**A**, **D**, **R**) : **A**

Dommages à l'ennemi : 15 **HP**, 30 **HP** restants

Action de William (**A**, **D**, **R**) : **D**

Action de Tardif (**A**, **D**, **R**) : **R**

Quel personnage voulez-vous restaurer (1, 2, 3) ? 2

5 **HP** restaurés à William, 17 **HP** sur 25 max

3.2.2 Action de l'ennemi

L'ennemi n'a qu'une action par tour, qui est de lancer une attaque physique ou une attaque de stress. On fera ce choix au hasard à chaque tour. L'attaque vise un joueur au hasard parmi les combattants. Pour calculer les dommages causés, on utilisera les règles suivantes :

- Dans le cas d'une attaque physique, on prend le nombre de points d'attaque de l'ennemi **attenn** et le nombre de points de défense du joueur visé **def** et on calcule $(attenn - def) * roll$ où **roll** est un nombre au hasard entre 0.8 et 1.2. Le nombre obtenu est alors soustrait aux **HP** du joueur visé. Par exemple,

L'ennemi attaque William pour 6 HP. William descend à 11 HP.

Attention, si le joueur visé par l'attaque a sélectionné l'action de défendre (D) lors de son action précédente, son nombre de points de défense est augmenté de 10% lors du calcul des dommages causés par l'ennemi.

- Dans le cas d'une attaque de stress, on considère **attstrewn** et on calcule le **strred** total comme la somme des **strred** des accessoires portés par le personnage visé. On effectue ensuite $(attstrewn - strred) * roll$ où **roll** est un nombre au hasard entre 0.8 et 1.2. Le nombre obtenu est ajouté au **str** du personnage.

Si un personnage a déjà atteint le niveau total de stress 100 **str**, il ne peut plus être visé par des attaques de stress (mais il peut toujours être visé par des attaques physiques).

Un combattant meurt si son nombre de points de vie devient négatif ou nul, et de même pour l'ennemi. Par exemple si William qui a 11 HP est à nouveau touché par une attaque, cette fois de 13 HP :

L'ennemi attaque William pour 13 HP. William descend à 0 HP.

Il atteint les profondeurs du donjon,
mais nous entendrons ses souffrances pour l'éternité.

Lorsqu'un combattant meurent sous les attaques ennemies, il ne peut plus agir et il est même supprimé de la partie, de même que ces accessoires sont irrémédiablement perdus.

Si un personnage atteint 100 **str**, son tour est automatiquement passé jusqu'à la fin du combat.

3.3 Fin d'un combat

Le combat se termine si tous les participants du joueur sont morts, auquel cas la partie est perdue, ou si l'ennemi est vaincu, c'est-à-dire s'il n'a plus de points de vie.

En cas de victoire, le joueur remporte 10 **or** et un accessoire. On affichera le montant d'or total du joueur et on affichera l'accessoire obtenu avec ses caractéristiques. Aux niveaux 2, 4, 6, 8, on affichera également le nouveau personnage disponible avec ses caractéristiques (comme en page 3). Les participants qui ne sont pas morts rejoignent la liste des personnages disponibles. Les accessoires qui étaient équipés pour le combat sont dés-équipés et rejoignent la liste des accessoires disponibles.

3.4 Avant le combat suivant

On affichera ensuite la liste des personnages qui sont au sanitarium avec leurs caractéristiques. Et on demandera au joueur s'il veut les retirer. Exemple de prompt (sous forme de tableau, mais l'affichage en ligne est autorisé) :

Sanitarium

	Nom	Classe	att	def	HP/HPmax	rest	str
1	Boudicca	Furie	13	5	13/20	0	45
2	Junia	Vestale	3	5	18/20	10	60

Voulez-vous faire sortir Boudicca du sanitarium ? Y

Voulez-vous faire sortir Junia du sanitarium ? N

Un personnage qui sort du sanitarium rejoint la liste des personnages disponibles. On fera de même avec la taverne : Afficher les personnages qui y sont avec leurs statistiques, et

demander au joueur s'il souhaite en faire sortir. Un personnage qui sort de la taverne rejoint la liste des personnages disponibles.

On affichera ensuite la liste des personnages disponibles pour choisir si on souhaite envoyer quelqu'un au sanitarium, à condition qu'il y ait de la place (2 personnages maximum au sanitarium).

Personnages disponibles

	Nom	Classe	att	def	HP/HPmax	rest
2	Junia	Vestale	3	5	18/20	10
3	William	Maître chien	10	15	17/17	5
4

Indiquez les numéros des personnages que vous souhaitez envoyer au sanitarium :

3

On procédera de même avec la taverne.

On proposera ensuite au joueur de passer à la roulotte. Pour chaque accessoire en vente, on affichera son nom, ses caractéristiques et son prix. Comme à chaque fois, on pourra faire un affichage sous forme de tableau

Roulotte

	Nom	attbonus	defbonus	HPbonus	restbonus	strred	prix
5	Anneau de possession	10
6	Cape d'invisibilité	20

ou plus simplement en ligne

Roulotte

5 - Anneau de possession, attbonus : ..., defbonus : ..., ...

6 - Cape d'invisibilité, attbonus : ..., defbonus : ..., ...

On proposera au joueur de faire un achat à la fois, en rappelant la quantité d'or à disposition. On mettra à jour l'or du joueur à chaque achat. On pourra utiliser Q pour quitter la roulotte.

Or : 56

Achat : 5

Or : 46

Achat : 6

Or : 26

Achat : Q

4 Listes chaînées

On impose l'utilisation de listes chaînées pour les personnages disponibles, les personnages combattant, les personnages au sanitarium, les personnages à la taverne, les accessoires disponibles et les accessoires en vente à la roulotte. Il faudra que votre code déplace correctement les personnages et les accessoires d'une liste à l'autre, ajoute les nouveaux personnages recrutés et supprime les personnages morts et les accessoires détruits.

5 Étapes d'implémentation du projet

L'interface graphique sera mise en place sur le terminal. On pourra d'abord implémenter le jeu par étapes :

1. Le jeu avec les personnages et les ennemis, sans accessoires ni sanitarium ni taverne ni roulotte. On pourra mettre tous les personnages comme disponibles dès le début.
2. On ajoute les accessoires et le démarrage du jeu avec seulement 2 personnages, les autres étant ajoutés aux niveaux 2, 4, 6, 8. À ce stade, toujours pas de sanitarium ni taverne ni roulotte.
3. On ajoute le sanitarium, la taverne et la roulotte.
4. On ajoute un système de sauvegarde (décrit ci-dessous).

On pourra augmenter les caractéristiques des ennemis aux cours des étapes d'implémentation. En effet, le jeu peut être plus difficile sans sanitarium ni taverne ni roulotte puisqu'on ne peut alors pas soigner les personnages hors des combats, ni soigner le stress ni obtenir de nouveaux accessoires en dehors des récompenses des combats. En l'absence de taverne, on pourra même empêcher les ennemis de lancer des attaques de stress.

Le système de sauvegarde se fera de la manière suivante. On écrira dans un fichier toutes les informations nécessaires à reprendre la partie : le niveau où on s'est arrêté (niveau k si on s'est arrêté après le combat $k - 1$), la liste des personnages disponibles (ne pas oublier leur **HP** et leur **str**), la liste des personnages au sanitarium, la liste des personnages à la taverne, la liste des accessoires disponibles, la liste des accessoires en vente à la roulotte, le montant d'or du joueur. Le schéma précis d'écriture du fichier est laissé à votre convenance.

Pour charger une sauvegarde, il faudra demander au démarrage du jeu si le joueur veut charger une sauvegarde et lui demander le nom du fichier de sauvegarde. Il faudra alors que le jeu lise les informations du fichier et ramène ainsi le joueur dans sa partie. On demandera de fournir un fichier de sauvegarde (généré par le jeu ou écrit à la main) pour tester la mise en place du système de sauvegarde.

6 Améliorations possibles

Les améliorations proposées sont à faire une fois le jeu réalisé tel que demandé.

- Faire un système de queue au sanitarium : on peut placer des personnages en attente d'entrer au sanitarium. Les personnages qui ont recouvré leur **HPmax** sortent automatiquement du sanitarium et laissent leur place aux premiers dans la queue,
- Même système de queue à la taverne.
- On pourra proposer une interface graphique avec la librairie MLV. Attention, si l'interface n'est pas utilisable, il est essentiel d'avoir une version fonctionnelle sur terminal pour faire une démonstration à vos enseignants.
- On pourra ajouter des niveaux au delà de 10,
- mettre plusieurs ennemis dans un combat,
- ajouter des classes avec différentes spécificités. Par exemple, une classe capable de réduire le stress pendant les combats.

7 Compte-rendu

Vous devrez impérativement accompagner votre code d'un compte-rendu. Celui-ci devra mentionner :

- une présentation du projet tel que vous l'avez implémenté,
- les structures mises en place,
- les fonctions créées,
- les étapes importantes du `main()`,
- les difficultés auxquelles vous avez été confronté,
- une réflexion sur le thème de l'utilisation de listes chaînées (qu'on vous a imposée) : est-ce qu'il y aurait une solution plus simple ? Est-ce que les listes chaînées offrent tout de même des avantages ici ?

Le but est d'une part d'avoir un vrai rapport de projet, et d'autre part de permettre à un programmeur (ou en pratique à un enseignant) de comprendre votre code si on l'ouvre à côté du compte-rendu, et même de reprendre votre code pour le faire évoluer.

8 Soutenance

Une soutenance aura lieu le 17 janvier à l'université. Il faudra compiler sans erreur ni warnings, puis exécuter votre code sur une machine de l'université. Vous ferez alors une démonstration de votre jeu.

Cette soutenance implique les deux membres du binôme. Des questions vous seront posées sur votre travail.

9 Derniers conseils

Les programmes `C` devront être clairs et suffisamment commentés. Ne commencez pas à développer une étape avant que la précédente ne soit pas totalement fonctionnelle.

Sauvegardez étape par étape des versions fonctionnelles pour pouvoir repartir sur de bonnes bases lorsque de nouvelles fonctionnalités se révèlent trop difficiles à implanter. Chacune des versions pourra être testée le jour de la soutenance.