



Projet d'informatique
Algorithmique et langage C

LABYRINTHE



Antoine HINTZY - 2022-2023

Labyrinthe

Labyrinthe est un jeu de société créé par **Max J. Kobbert** et édité par **Ravensburger** en **1986**.

L'objectif est d'atteindre des trésors dans un labyrinthe dont les couloirs glissent et évoluent à chaque tour de jeu.

Vidéo : [Publicité du jeu \(1988\)](#)

Source : Wikipedia, Crédits image : ludos.brussels



Règles du jeu

Vous trouverez les règles complètes et détaillées du jeu sur la page suivante :

[Règles officielles \(PDF\)](#)

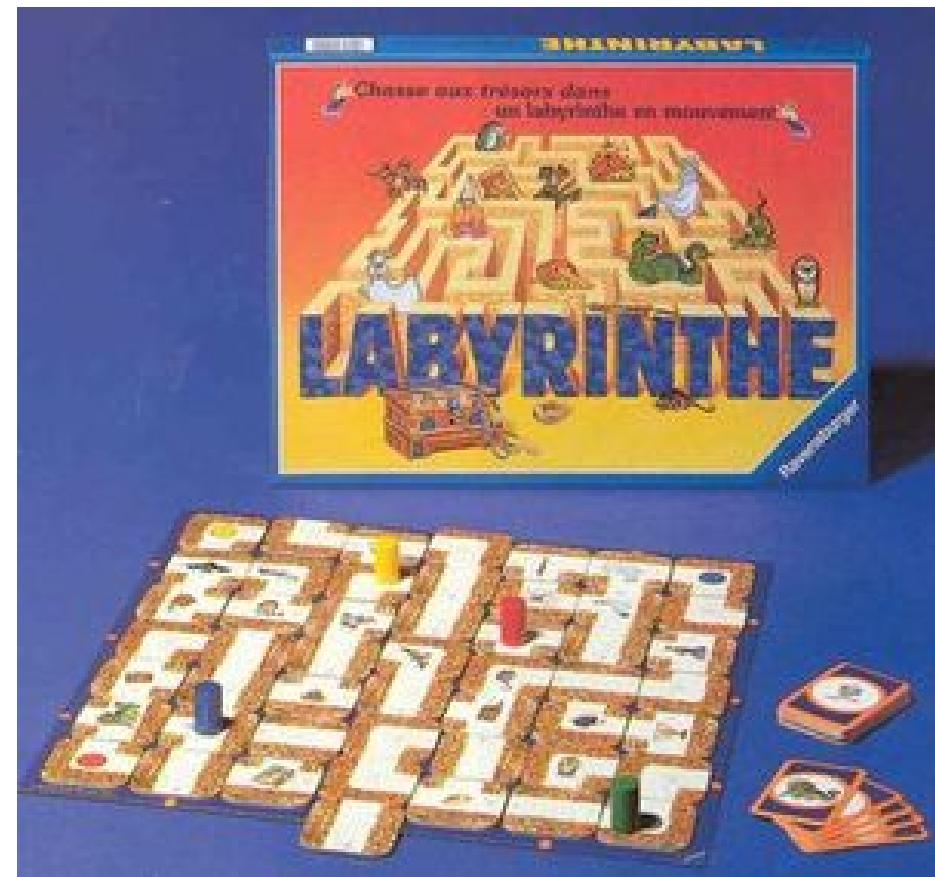
Ainsi que les règles d'une version étendue sur la page suivante :

[Règles avec variantes \(PDF\)](#)

Composition du coffret

Vous allez devoir recopier le plus fidèlement possible le jeu officiel, dont le coffret se compose :

- d'un plateau pouvant accueillir 7x7 tuiles.
- de 16 tuiles "couloir de labyrinthe" à position fixe sur le plateau (*voir slide suivante*).
- de 34 tuiles "couloir de labyrinthe", dont 33 sont à positionner sur le plateau en début de partie, en plus des 16 tuiles déjà présentes. *La tuile restante permettra de pousser les rangées de tuiles.*
- de 24 cartes "trésor" à distribuer.
- de 4 pions.



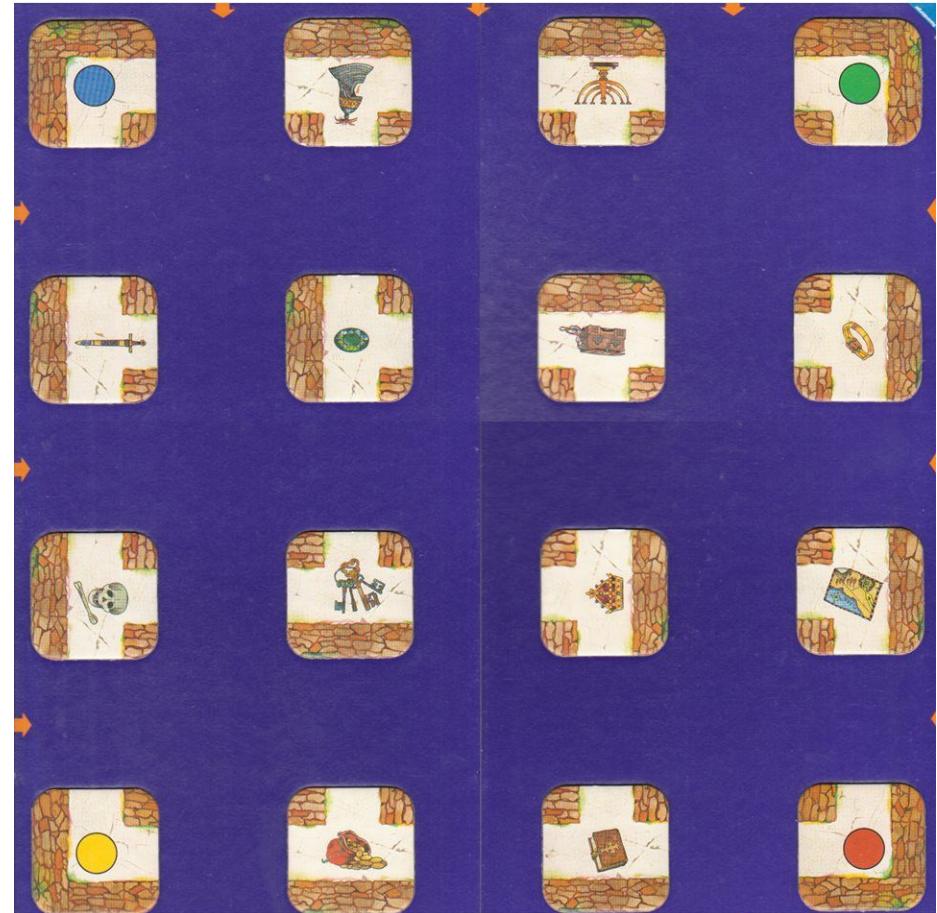
Le plateau

Le plateau, de **7 lignes** et **7 colonnes**, comporte 16 tuiles "couloir de labyrinthe" dont la position est réglementée et fixe (*voir image ci-contre*).

L'orientation de ces 16 tuiles est importante.

Chacune d'entre elles :

- comporte un trésor à trouver  (x12, en **T**)
ou
- indique la position de départ et d'arrivée d'un pion  (x4, en **L**, dans chaque coin)



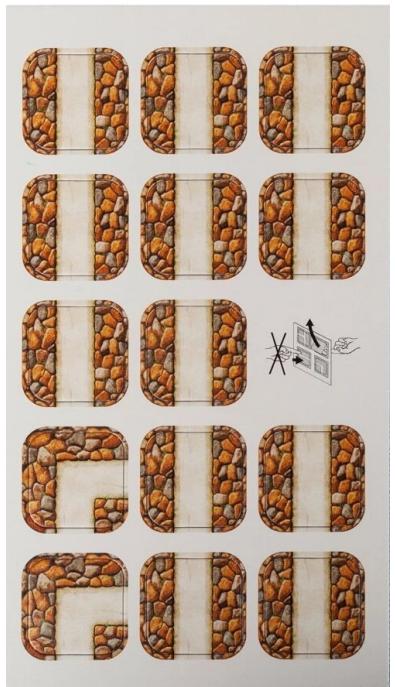
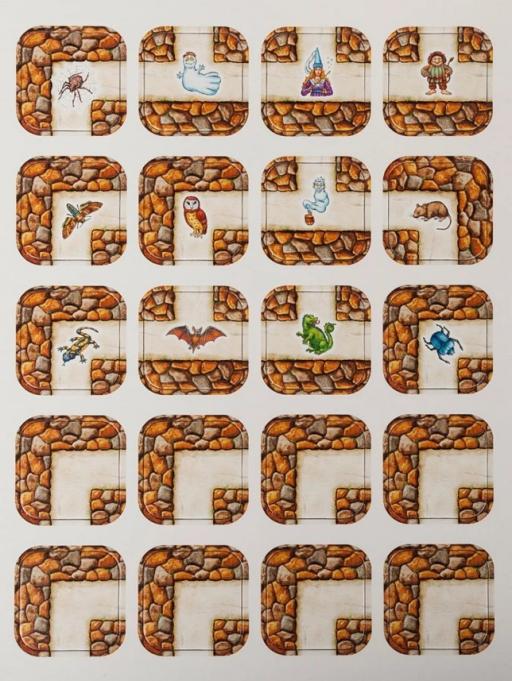
Tuiles "couloir de labyrinthe"

Le coffret comprend, en plus des 16 tuiles fixes, **34 tuiles** "couloir de labyrinthe" à placer en début de partie :

- 6x avec trésor (en **T**)
- 6x avec trésor (en **L**)
- 10x (en **L**)
- 12x (en **I**)

Chacune de ces 34 tuiles est disponible dans toutes les orientations possibles :

- Tuiles en **L** : , , , .
- Tuiles en **I** : , .



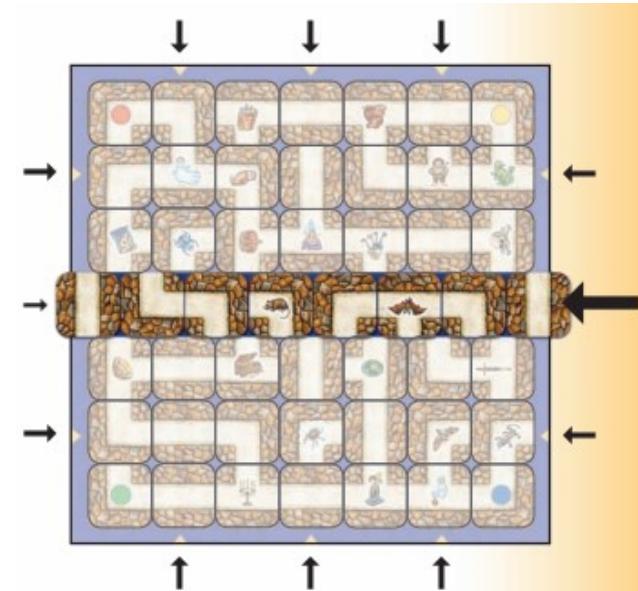
Tuiles "couloir de labyrinthe"

Il y a donc au total **50 tuiles** :

- **16** à position et orientation définies.
- **34** à position et orientation libres, placées aléatoirement au début de chaque partie.

Or, le plateau ne pouvant accueillir que 49 tuiles (7x7), la tuile supplémentaire sert au joueur qui joue à faire coulisser une rangée du plateau.

⚠ Seules les rangées ne contenant aucune tuile à position fixe peuvent être coulissées. Elles sont identifiées par des flèches sur l'image ci-contre.



Cartes Trésor

Le jeu comporte **24 cartes "trésor"** (*autant qu'il y a de trésors sur le plateau*) à distribuer équitablement entre les joueurs, face cachée, au début de chaque partie.

Chaque joueur ne peut voir qu'une seule de ses cartes "trésor" à la fois. Celle-ci lui indique le prochain trésor qu'il doit atteindre. Une fois ce trésor atteint, la prochaine carte "trésor" du joueur lui est dévoilée, lui indiquant le prochain trésor à atteindre.

 *Vous représenterez les trésors comme vous le souhaitez, avec un caractère spécial et une couleur par exemple.*



Pions

Il est possible de jouer au Labyrinthe à 2, 3 ou 4 joueurs.

Après avoir reçu ses cartes "trésor", chaque joueur choisit l'un des 4 pions, qui est ensuite positionné sur la case de départ correspondante (l'un des 4 coins du plateau).

 *Vous représenterez les pions comme vous le souhaitez, avec un caractère spécial et une couleur par exemple.*



Menu principal

Le menu principal apparaît au lancement du jeu, et à tout moment au cours d'une partie (*cela se gérera par des boucles dans des boucles, pas des appels du sous-programme menu dans lui-même*). Le menu proposera par exemple :

- Nouvelle partie
- Sauvegarde (si la partie est en cours)
- Charger une partie
- Afficher les règles / crédits
- Quitter

NB : La sauvegarde et le chargement sont des bonus.

Lancement d'une partie

Au lancement d'une nouvelle partie :

- nous choisissons le nombre de joueurs, leur proposons de choisir leur pion, et leur donnons idéalement un nom
- le plateau (labyrinthe) est généré aléatoirement avec les tuiles "couloir de labyrinthe", et les pions sont placés sur leur tuile de départ
- les cartes "trésor" sont réparties entre les joueurs, face cachée
- les joueurs jouent à tour de rôle (un joueur est choisi aléatoirement pour commencer) :
 - ils déplacent une rangée du labyrinthe en la faisant coulisser avec la tuile supplémentaire (elle ne peut pas être réinsérée là où elle vient de sortir).
 - ils peuvent, s'ils le souhaitent, déplacer leur pion d'autant de tuiles qu'ils veulent, sans traverser un mur.

Génération du labyrinthe

Pour générer le Labyrinthe, vous devez :

- placer les 16 tuiles à position et orientation fixe
- générer les 34 tuiles supplémentaires en respectant leur format et leur quantité, les mélanger (vous pouvez faire autrement), les faire tourner éventuellement, puis les placer sur le plateau

La tuile restante permet au joueur dont c'est le tour de faire glisser une rangée de Labyrinthe (⚠ toutes les rangées ne peuvent pas glisser). La tuile supplémentaire (hors plateau) change à chaque tour (c'est la tuile sortie lors du glissement de la rangée), le joueur suivant aura à la placer en faisant glisser une rangée (⚠ la tuile ne peut pas être réinsérée là où elle vient de sortir).

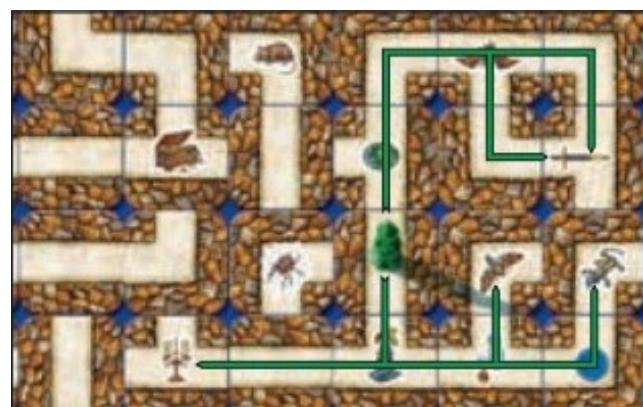
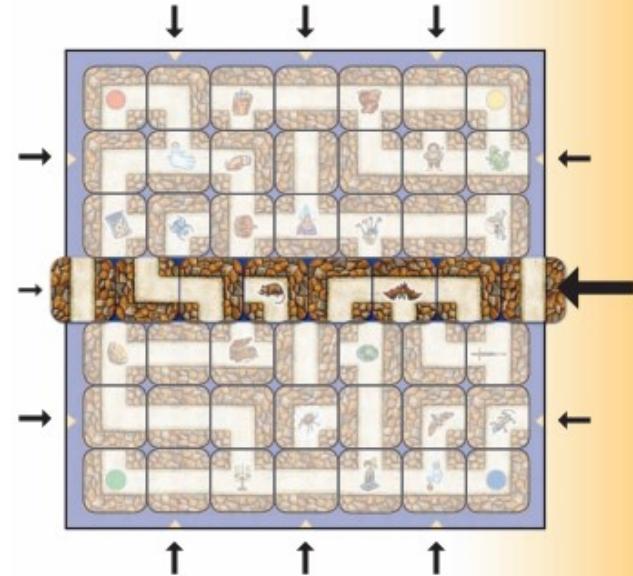
 Les indices des rangées pouvant glisser ont un point commun...

Déroulement d'un tour de jeu

Les joueurs jouent les uns après les autres, en boucle, jusqu'à ce que l'un des joueurs ait atteint tous ses trésors (désignés par ses cartes "trésor") et soit revenu à sa position de départ.

Lorsque c'est son tour, un joueur :

- fait glisser une rangée ( pas n'importe laquelle) en insérant la tuile qui n'est pas sur le plateau sur l'une des 9 flèches du plateau (la tuile sortante devient la tuile que le joueur suivant devra insérer).
- déplace son pion, s'il le souhaite, d'autant de cases qu'il veut, sans traverser les murs.



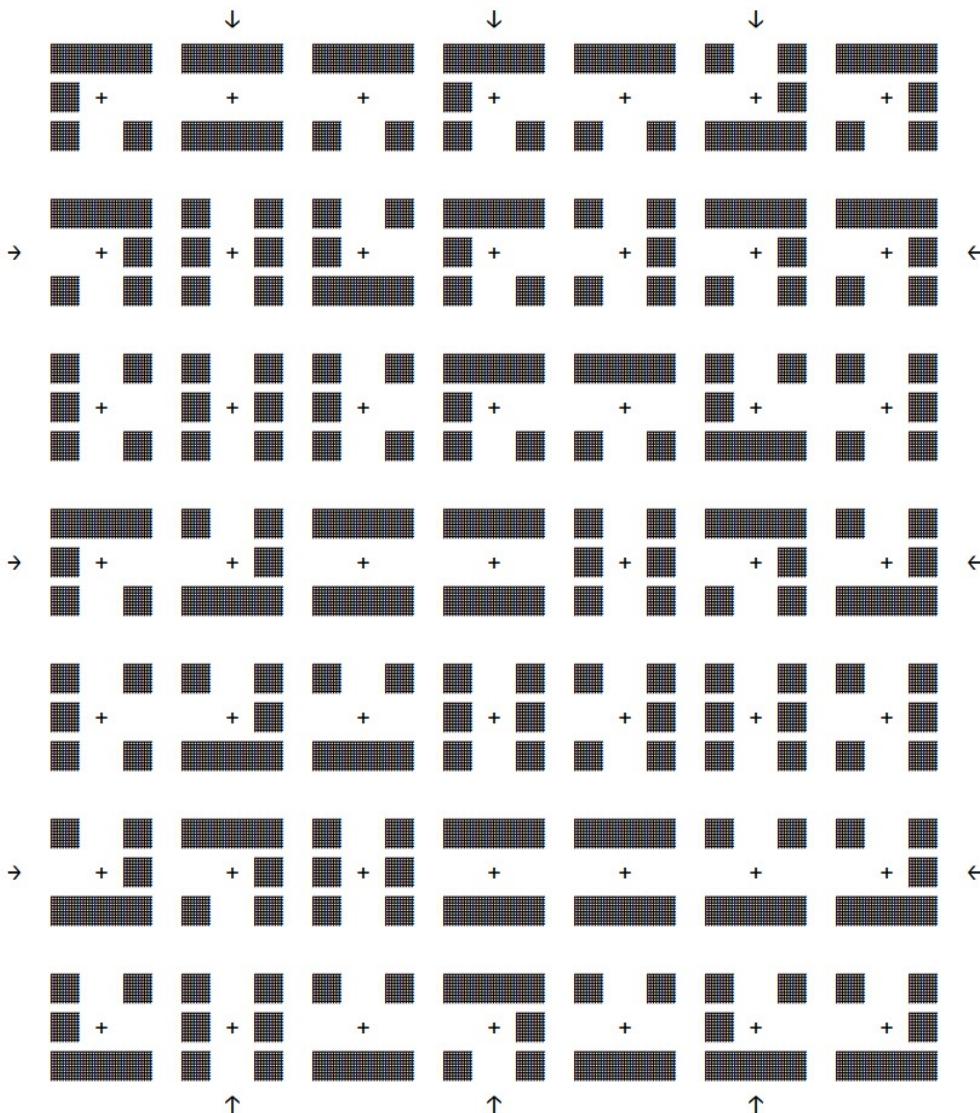
Affichage

L'affichage se fera en mode console, à l'aide de la table ASCII, que vous pouvez afficher ainsi :

```
for(int i = 0; i < 256; i++) {  
    printf("%d : %c\n", i);  
}
```

L'image ci-contre vous est donnée à titre d'exemple, elle ne respecte pas les règles concernant les tuiles à placer (quantité).

⚠ Nous afficherons l'intégralité du plateau à chaque tour de jeu. Ce dernier sera généré à l'aide de boucles `for`, dans une procédure dédiée à l'affichage.



Process finished with exit code 0

Thème

Vous avez le choix du thème de votre jeu. Le jeu officiel ayant lui-même été décliné dans différents thèmes.

Puisque nous développons le jeu en mode console, vous ne pouvez réellement exploiter votre thème que dans la narration. Ne négligez pas les interfaces qui doivent être claires et donner envie de jouer !



Adaptez le jeu à votre niveau

Vous pouvez au besoin alléger les règles ou les adapter à votre niveau afin de pouvoir rendre et présenter un jeu fonctionnel. Vous pouvez par exemple faire la variante destinée à de jeunes joueurs :

Variante allégée

Distribuer les cartes Trésor comme d'habitude. Chacun les retourne toutes devant lui (les trésors sont donc visibles). Quand vient son tour, le joueur essaye d'atteindre l'un de ses trésors, au choix. Dès qu'il y parvient, il retourne la carte. Lorsque toutes les cartes ont été retournées, le joueur doit encore rejoindre son point de départ, comme d'habitude.

Bonus

Si vous avez terminé les fonctionnalités de base, et que vous souhaitez aller plus loin et épater le jury, vous pouvez :

- Afficher le plateau en couleurs.
- Réaliser une interface graphique (Allegro ou autre).
- Enrichir les règles, avec des règles de votre invention, ou telles que détaillées pages 6 et 7 de [ce PDF](#).
- Sauvegarder/charger une partie.

NB : Aucun bonus ne rapportera de point si les fonctionnalités de base ne sont pas fonctionnelles.

Travail à faire / livrables

Vous serez évalués lors de votre soutenance sur la qualité de votre **conception** (anticipation), votre **organisation**, la qualité de votre **implémentation**, la **jouabilité** de votre jeu, ainsi que sur votre **présentation orale**.

Voici les différents livrables à rendre sur Boostcamp (*les deadlines y sont listées*):

- Preuves de conception (.pdf)

Aucune rédaction n'est demandée, ce document regroupe toutes vos recherches de conception (DTI, ACD, Algorithmes, schémas...) effectuées avant votre implémentation.

- Code source (.zip)
- Slides de soutenances (.pdf)
- *Bonus : vidéo et/ou infographie (bonus pouvant aller jusqu'à 1 point)*

Barème indicatif de la soutenance 1/2

Tâche	Détails	Points
Conception	<ul style="list-style-type: none">- <i>DTI / ACD</i>,- <i>Algorithmes</i> : ayant nécessité une certaine réflexion : comment déplacer une rangée, générer le labyrinthe...,- <i>Interfaces</i> : Affichage soigné et intuitif,- <i>Architecture de votre programme / stockage des données en mémoire</i> : comment vous représentez le plateau ? les tuiles ? les cartes ? etc. (tableaux de ..., structures, pointeurs ?).	7
Organisation	<i>Répartition des tâches, gestion des deadlines, outils.</i>	1
Bilan	<i>Fonctionnalités (non-)implémentées, des problèmes rencontrés, bonus réalisés.</i>	2

Barème indicatif de la soutenance 2/2

Tâche	Détails	Points
Respect du CDC	<i>Toutes les fonctionnalités attendues (slide suivante) ont été implémentées.</i>	15
Jouabilité	<i>Le programme compile et permet de jouer à une partie.</i>	5
Bonus	<i>Seulement si le jeu est fonctionnel à 100%.</i>	(+4)

Fonctionnalités attendues

- Menu principal (*On peut retourner au menu principal à tout moment*).
- Nouvelle partie (*On peut recommencer une nouvelle partie à tout moment depuis le menu principal*).
 - Choix du nombre de joueurs, création et placement des joueurs/pions... **(1 point)**
 - Création des cartes Trésor et des tuiles Couloir de Labyrinthe **(3 points)**
 - Création du Labyrinthe à partir des tuiles créées **(2 points)**
 - Distribution des cartes Trésor **(1 point)**
- Affichage du Labyrinthe **(3 points)**
- Pose d'une tuile, *glissement de rangée* **(2 points)**
- Déplacement d'un pion **(2 points)**
- Détection de la victoire **(1 point)**

NB : Les joueurs doivent pouvoir jouer à tour de rôle.

Soutenance

Vous démontrerez lors de votre soutenance que vous avez bien répondu aux différents points précédemment énoncés.

La soutenance se découpe en trois phases :

- Présentation (slides)
- Démonstration
- Questions / réponses

Les bonus peuvent ajouter jusqu'à 4 points.

 Attention à ne pas passer trop de temps sur votre présentation (~5 minutes).

Livrables bonus

Les livrables bonus sont à déposer sur Boostcamp. Ils octroient un bonus pouvant atteindre 1 point, en fonction de leur qualité. Ils pourront être présentés lors d'événements école, en Journées Portes Ouvertes par exemple.

Vidéo

La vidéo explique comment se déroule un projet d'informatique à l'ECE. Vous pouvez par exemple présenter les procédés de conception, l'implémentation, et montrer le résultat obtenu. Evitez les plans de type interview et ajoutez si possible des sous-titres.

Infographie (A3, portrait)

La vidéo sera accompagnée d'une infographie "affiche de film" retraçant de façon synthétique votre avancée lors de ce projet. Vous présenterez votre conception, votre implémentation, et le résultat attendu. Soyez créatifs.

Plagiat

Tous les codes sources seront analysés par un outil spécialisé dans la détection de plagiat de code source.

Le plagiat est une faute lourdement sanctionnée en **Conseil de discipline**.

Méthodes de travail

Vous allez travailler par équipes de 3-4 étudiants de même niveau.

L'idée générale est de faire progresser tout le monde en informatique. C'est pourquoi l'entraide au sein d'un même groupe est primordiale.

Pour cela, attribuez vous les tâches en fonction de vos compétences.

Nous vous conseillons de nommer un(e) chef de projet qui s'assurera du bon avancement du projet, de la bonne répartition des tâches, etc.

N'hésitez pas à utiliser des outils pour organiser votre travail : [Trello](#), [Monday](#), etc.

Méthodes de travail

Conception

L'étape de conception est **TRES IMPORTANTE** : chaque année, des équipes de projet se lancent dans le code sans avoir réfléchi à sa conception et se retrouvent bloqués à 60% du projet car ils ont fait de très mauvais choix algorithmiques (la façon dont sont stockées les données par exemple).

Avant de commencer à coder, vous **devez** tous avoir une vision claire et globale de la façon dont va être organisé/structuré/implémenté votre jeu : comment le plateau va-t-il être stocké dans la mémoire ? Comment les rangées et les pions seront-ils déplacés ? Comment les tuiles vont-elles être générées et placées sur le plateau ? Comment accéder à la tuile qui n'est pas sur le plateau ? Comment interdire de replacer la tuile au même endroit ? Comment retourner au menu principal sans perdre les données de la partie ? etc.

Méthodes de travail

Conseils

Veillez à tout prix à éviter la duplication de code et surtout de données : si un joueur est sur une case, cette information ne doit être stockée qu'à un unique endroit. Si cette donnée est stockée à différents endroits, vous risquez d'avoir des incohérences dans votre jeu si vous oubliez de mettre tous les doublons à jour. Utilisez les pointeurs pour faciliter les accès sans dupliquer les valeurs.

Séparez les données et l'affichage : si le plateau (labyrinthe) change, modifiez les données correspondantes, puis, appelez la fonction d'affichage en lui fournissant le plateau actualisé. Mélanger l'affichage et le traitement des données est une mauvaise idée.