



v1.1 **Projet CY-Memo-RPG**

CLASSE préING1 • 2022-2023

AUTEUR Romuald GRIGNON

E-MAIL romuald.grignon@cyu.fr

DESCRIPTION

Le but de ce projet est de coder un jeu de type Minimalist-Competitive-Local-Multiplayer-Memory-RPG de type GaaS (Game as a Service) très en vogue dans le monde de l'E-Sport.

Le but du jeu pour chacun des aventuriers (entre 2 et 4 par partie) est de parcourir le labyrinthe du donjon à la recherche d'un coffre au trésor et d'une arme antique, tout en terrassant les monstres ennemis qui s'y trouvent.

Il existe 4 aventuriers, un guerrier, un ranger, un magicien et un voleur. Tous ces aventuriers recherche la richesse, mais chacun recherche également une arme antique bien précise :

- le guerrier recherche l'épée de feu
- le ranger recherche le baton de contrôle des familiers
- le magicien recherche le grimoire interdit
- le voleur recherche la dague de sommeil

Le plateau de jeu (le labyrinthe) se compose de 25 cartes mélangées aléatoirement, et disposées faces cachées en carré.

Les cases du labyrinthe cachent les éléments suivant :

- 4 armes antiques décrites précédemment (1 de chaque)
- 2 coffres au trésor
- 16 monstres
- 1 portail magique de téléportation
- 2 totems de transmutation

Les monstres qui se trouvent dans le labyrinthe sont au nombre de 16, classés en 4 catégories :

- les Basilics, qui ont un regard pouvant changer un aventurier en statue de pierre : seul un miroir peut les vaincre
- les Zombies, insensibles à toute arme classique, seul le feu peut en venir à bout.
- les Trolls : de grands ennemis qui ne succomberont qu'aux coups dévastateurs de haches à 2 mains.
- les harpies : monstruosité volante, elles ont un point faible contre les armes de jet. Un arc semble être la solution.

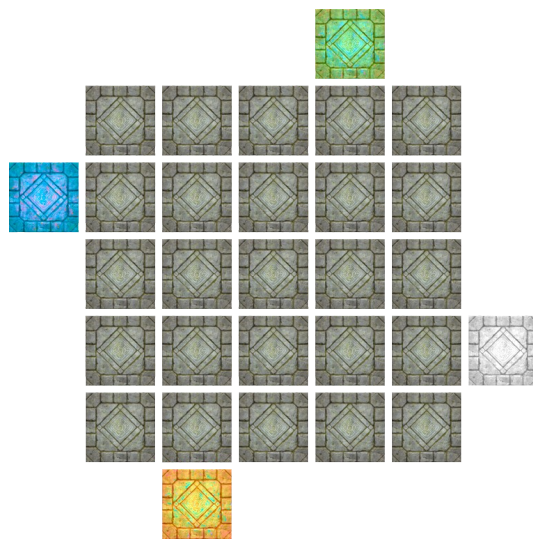
Les joueurs sont positionnés dans les coins du labyrinthe (cf. schéma du plateau de jeu). Puis la partie peut démarrer : chaque joueur effectue un trajet dans le labyrinthe chacun son tour.

Chaque aventurier, avant d'avancer d'une case dans le labyrinthe, doit s'équiper d'une arme parmi 4 : un bouclier réfléchissant, une torche, un arc long ou une hache de pierre. Chacune de ces armes permet de venir à bout d'un seul type de monstre en fonction des descriptions précédentes.

Un aventurier continue de jouer tant qu'il n'est pas vaincu par un monstre, qu'il n'est pas bloqué dans le labyrinthe ou qu'il n'est pas encore vainqueur.

Pour être un vainqueur, un aventurier doit partir de sa position de départ, parcourir le labyrinthe sans jamais être vaincu par un monstre, trouver au moins 1 coffre au trésor et trouver son arme antique.

Les 4 positions de départ sont placées comme sur le schéma ci-dessous (cartes colorées). Seules les cases face cachées autour de l'aventurier peuvent être révélées : ceci impose forcément la première case à retourner lorsque débute le tour d'un joueur.



Un tour de jeu pour un aventurier se déroule de la manière suivante :

#1 - Equipement d'une arme : le joueur choisit une des 4 armes parmi le bouclier, la hache, la torche et l'arc. C'est l'arme active.

```
Veillez choisir une arme :  
1 - Torche  
2 - Bouclier  
3 - Hache  
4 - Arc
```

#2 - Parcours du labyrinthe : le joueur révèle une case cachée du labyrinthe, adjacente à celle sur laquelle il se trouve (jamais en diagonale) :

Maintenant choisissez une direction :
Haut
[8]
Gauche [4] [6] Droite
[2]
Bas

- si la case contient un monstre, et que l'arme active de l'aventurier est la bonne arme pour vaincre le monstre, le joueur actif est placé sur la case du monstre. Le joueur continue d'avancer dans le labyrinthe (il peut donc retourner une autre case cachée adjacente). Par contre si son arme est inefficace, l'aventurier meurt, se retrouve à sa position de départ, et toutes les cases sont retournées face cachée. Les monstres déjà vaincus sont ressuscités à leurs emplacements respectifs. C'est au tour du joueur suivant de parcourir le labyrinthe.

- si la case retournée est un coffre au trésor : bravo, il reste encore à trouver l'arme antique si ce n'est pas déjà fait.

- si la carte retournée est l'arme antique correspondant à l'aventurier, bravo, il reste encore à trouver un coffre au trésor si ce n'est pas déjà fait.

- si la carte retournée est une arme antique mais pas celle recherchée par l'aventurier, alors il faut simplement continuer (retourner une nouvelle case).

- si la carte retournée est un portail magique de téléportation, la prochaine case choisie par le joueur peut être n'importe quelle case encore cachée (pas forcément une case adjacente). A vous de définir l'interface de choix (numéro de ligne et de colonne de la case sur laquelle se téléporter, ...).

- si la carte retournée est un totem de transmutation, le joueur doit choisir une case face cachée n'importe où sur le plateau de jeu, et le totem sera échangée avec la case choisie (sans révéler la case cachée). Cela met fin au tour du joueur (l'aventurier repasse à sa position de départ, les cartes du labyrinthe sont toutes retournées face cachée). Attention il n'est pas possible d'échanger un totem avec les cases situées en face des positions initiales des joueurs.

- si il n'y a plus aucune case cachée autour de la position de l'aventurier, cela met fin à son tour de jeu également (il est bloqué dans le labyrinthe).

#3 - Conditions de victoire :

- si l'aventurier est vainqueur (il a révélé SON arme antique, ET AU MOINS UN coffre au trésor) la partie s'arrête et les scores sont mis à jour (voir section "fonctionnalités du projet pour les détails des scores").

- si l'aventurier est toujours en vie et que son tour n'est pas terminé (il a besoin de révéler un coffre et son arme antique), mais qu'il n'est pas encore vainqueur, on repasse à l'étape #1 (choix d'une arme, puis choix d'une case face cachée, ...). De fait un monstre terrassé n'apparaît plus

sur la carte, la case où se trouvait le monstre apparaît maintenant complètement vide.

- si l'aventurier est mort ou bloqué, ou est tombé face à un totem de transmutation, il est positionné à sa position initiale, toutes les cartes sont retournées face cachée, les monstres réinitialisés et c'est au tour du joueur suivant.

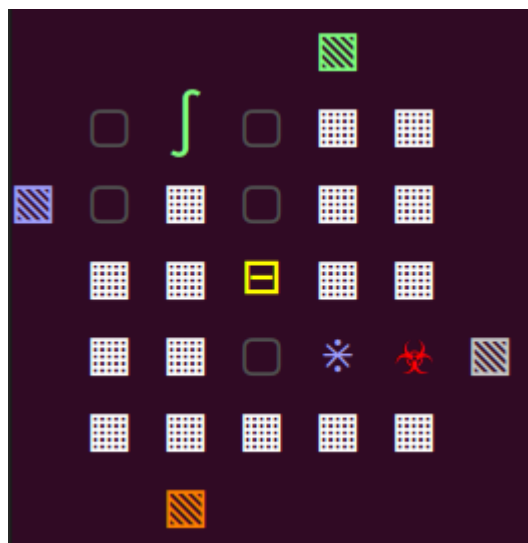
INTERFACE GRAPHIQUE

Voici un exemple d'interface pour ce jeu, avec une légende associée. Vous êtes libres de l'aspect graphique, il s'agit néanmoins que la zone de jeu soit compréhensible pour les joueurs :

POSITIONS DE DEPART		AVENTURIERS	
Ranger [▨]	[▩] Guerrier	Ranger [➔]	[♣] Guerrier
Voleur [▧]	[▪] Magicien	Voleur [♠]	[*] Magicien

TRESORS	
Baton [∫]	[×] Epee
Dague [↗]	[≡] Grimoire
	[☐] Coffre trésor

LABYRINTHE	
Inconnu [■]	[□] Vide
Basilic [⊖]	[*] Zombie
Troll [●]	[♣] Harpie
Portail [D]	[▲] Totem



Sur l'exemple ci-dessus, c'est le joueur qui possède le magicien qui est en train de jouer. Il a découvert une arme antique mais c'est celle pour le ranger : le ranger connaît donc maintenant sa position : c'est une aide pour la suite du jeu. On voit également qu'un coffre au trésor a été

découvert, c'est également une aide pour les autres joueurs. Les cases vides correspondent aux cases où des monstres ont été vaincus. Malheureusement, les autres joueurs ne peuvent plus savoir lesquels : il faut absolument mémoriser l'emplacement des monstres au fur et à mesure qu'ils sont dévoilés sur le plateau.

Là on voit qu'un monstre de type zombie vient d'être découvert : si le joueur actuel est équipé de la torche il terrasse le monstre et avancera sur cette case automatiquement. Sinon c'est la fin de son tour. C'est le monstre qui ressort vainqueur du combat. C'est au tour du joueur suivant de parcourir le labyrinthe (bien sûr tous les monstres ont été réinitialisés, et les cases du labyrinthe remises face cachée).

L'intérêt du jeu consiste donc à mémoriser où se trouvent les monstres pour anticiper le choix de l'arme à chaque déplacement, et bien sûr trouver l'emplacement de son arme antique et d'au moins 1 coffre au trésor. C'est au fur et à mesure que les autres joueurs avancent dans le labyrinthe que vous pourrez mémoriser le chemin vers la victoire.

Le premier joueur à dévoiler son arme antique + 1 coffre au trésor est déclaré vainqueur sur le champ : la partie est terminée.

INFORMATIONS GENERALES

Taille de l'équipe

Ce projet est un travail d'équipe. Il est autorisé de se réunir en groupe de 3 personnes.

Démarrage du projet

Vous obtiendrez de plus amples informations quant aux dates précises de rendu, de soutenance, les critères d'évaluation, le contenu du livrable, ..., quand le projet démarrera officiellement.

Dépôt de code

Vous devrez déposer la totalité des fichiers de votre projet sur un dépôt central Git. Il en existe plusieurs disponibles gratuitement sur des sites comme github ou gitlab.

Démonstration

Le jour de la présentation de votre projet, la version finale sur votre dépôt sera celle utilisée, même si vous avez ajouté des fonctionnalités ou corrigé des bugs entre temps. La démonstration se fera sur une machine de l'enseignant chargé de suivre votre groupe. C'est la date de commit sur votre dépôt qui fera foi.

Vous ferez votre démonstration, en fonction des exigences du cahier des charges de votre projet, et vous aurez à modifier légèrement votre application en direct en fonction de la requête de votre chargé de TD.

De plus des questions supplémentaires pourront être posées afin d'évaluer votre connaissance de l'implémentation de votre projet.

FONCTIONNALITES PRINCIPALES DU PROJET

- Le but principal du projet est de fournir une application **fonctionnelle**.
- Tous les **éléments de votre code** seront écrits en langue **anglaise** (structure, types, fonctions, fichiers, ...).
- Votre code sera généreusement **commenté** (langue française autorisée).
- Votre application ne doit **jamais** s'interrompre de manière intempestive (crash), quelle que soit la raison. Toutes les erreurs

doivent être gérées correctement. Il est préférable de d'avoir une application stable avec moins de fonctionnalités qu'une application contenant toutes les exigences du cahier des charges mais qui plante trop souvent. Une application qui crée des fautes de segmentation par exemple sera très pénalisée.

- Votre application devra être **modulée** afin de ne pas avoir l'ensemble du code dans le même fichier.
- Votre **livrable** sera une **URL** d'un dépôt **Git public** envoyée à votre chargé de TD.
- Votre projet doit pouvoir respecter toutes les règles du jeu et afficher le résultat de la partie quand elle est terminée.
- Votre projet doit pouvoir intégrer une fonctionnalité pour choisir le nombre de joueurs ainsi que leurs noms au début de chaque partie.
- A la fin d'une partie, vous devez donner le choix de recommencer une partie avec les mêmes joueurs, ou bien retourner au menu principal pour redémarrer une toute nouvelle partie.
- A la fin d'une partie, le nom du gagnant est affiché, et l'ensemble du labyrinthe devient visible pour que chacun puisse analyser les emplacements de chaque carte.
- Votre projet doit pouvoir mémoriser à long terme l'ensemble des noms des joueurs qui ont joué au jeu sur cette machine. En plus des noms, vous stockerez différentes informations comme le nombre de parties jouées et le nombre de victoires de chacun de ces joueurs. Vous devez pour chaque joueur stocker le nombre de cases révélées, le nombre de monstres terrassés et le nombre total de trésors découverts. Ce sont ces informations qui doivent être stockées dans le fichier des scores.
- Dans le menu du début du jeu, l'utilisateur peut demander à visualiser les scores de tous les joueurs qui ont participé à ce jeu sur cette machine.
- La durée de chaque manche doit être au moins mesurée et affichée à la fin de la manche. Si possible vous pouvez afficher constamment le temps de la manche en cours.
- Les noms des joueurs doivent être affichés constamment sur l'interface de jeu.
- Le format d'affichage est laissé libre, l'important étant de respecter les règles du cahier des charges

RESSOURCES UTILES

Github

- www.github.com
- <https://docs.github.com/en/get-started/quickstart/hello-world>

Modifier les couleurs du terminal Linux

- https://en.wikipedia.org/wiki/ANSI_escape_code#Colors
- <http://sdz.tdct.org/sdz/des-couleurs-dans-la-console-linux.html>

Contrôler le curseur du terminal

- [https://en.wikipedia.org/wiki/ANSI_escape_code#CSI_\(Control_Sequence_Introducer\)_sequences](https://en.wikipedia.org/wiki/ANSI_escape_code#CSI_(Control_Sequence_Introducer)_sequences)