

Projet : Un jeu d'aventure sur terminal

On vous demande de développer un jeu qui s'inspire du jeu Cataclysm: Dark Days Ahead <https://cataclysmdda.org/>. C'est un jeu de survie dans un monde post-apocalyptique. Il est open-source et se joue au tout par tour.

Pour réaliser ce projet, vous devez utiliser le langage Java et faire appel à vos connaissances des designs pattern pour la phase de conception.

Ce projet est à réaliser en binôme ou en trinôme. On vous demande d'enregistrer les noms des étudiants de votre groupe dans le wiki nommé **Constitution groupes projet** correspondant à votre groupe avant le **21 Novembre 2022** afin de pouvoir créer les dépôts Git rapidement. Pour modifier le tableau du wiki, il faut cliquer sur le wiki puis ouvrir la liste déroulante indiquant « Afficher » et sélectionner « Modifier ».

Vous pouvez constituer votre binôme ou trinôme avec des étudiants d'un autre groupe que le votre. Inscrivez votre binôme ou trinôme dans un des groupes. Nous réorganiserons les groupes pour les équilibrer si besoin.

Figure 1: Ecran du jeu Cataclysm : Dark Days Ahead - Coolirisme, CC BY-SA 4.0, via Wikimedia Commons



I-Travail à réaliser

La zone de jeu est représentée comme une matrice de caractères. Toute l'interface se fera dans un terminal. Votre personnage pourra se déplacer, se battre, ramasser des objets à l'aide d'un menu textuel. Il peut aussi reposer un objet ou un écureuil qui était dans sa poche.

La zone de jeu pourra prendre plusieurs apparences selon le thème choisi. On envisagera deux thèmes : Forêt et Jungle mais on pourrait en ajouter d'autres ultérieurement.

Pour le thème forêt, la zone de jeu comportera les différents types de cases suivantes :



Personnage



Écureuil (la couleur de la police du caractère E varie en fonction de l'état de l'écureuil : affamé : noir, rassasié : bleu, ami : violet, junkie : rouge, perché dans un arbre : vert, caché dans un buisson : jaune).



Arbre



Buisson



Gland



Champignon



Champignon venimeux



Renard



Zone vide.

Pour le thème jungle, la zone de jeu comportera les différents types de cases suivantes :

- Personnage
- Singe (la couleur peut varier en fonction de l'état du singe)
- Cocotier
- Buisson

- Banane
- Champignon
- Champignon hallucinogène
- Serpent
- Zone vide

Vous choisirez la représentation de chaque type de case du thème Jungle.

Au démarrage du jeu, l'application proposera de créer une nouvelle carte ou de charger une carte depuis un fichier. On vous fournit un exemple de fichier « carte.txt » correspondant à une carte de 35 lignes et 100 colonnes pour le thème Forêt (les champignons venimeux sont notés M dans la carte). Vous pouvez apporter des modifications à cette carte. Vous pourrez créer une autre carte pour le thème Jungle.

Intelligence artificielle pour le thème Forêt

Vous devez mettre en place une intelligence artificielle pour les animaux de la forêt.

Vous voulez permettre au personnage d'apprivoiser un écureuil pour en faire son acolyte.

Les écureuils se déplaceront différemment selon leur état et suivant ce qu'ils rencontrent à proximité.

Un écureuil affamé deviendra rassasié s'il se nourrit.

Un écureuil rassasié sera de nouveau affamé au bout de 5 tours.

Un écureuil affamé qui se nourrit sur une case adjacente au personnage deviendra son ami.

Si le personnage donne un coup à l'écureuil, il ne sera plus son ami.

Si l'écureuil mange un champignon venimeux, il devient junkie pendant 5 tours puis ensuite sera de nouveau affamé.

Un écureuil junkie se déplace aléatoirement.

Le déplacement d'un écureuil affamé dépend de ce qu'il voit autour de lui en suivant l'ordre de priorité ci-dessous :

1. S'il voit un gland : peu importe le danger, il va le chercher et le mange. L'écureuil prend la place du gland.
2. S'il voit un champignon: il va le chercher et le mange. L'écureuil prend la place du champignon.
3. Si à moins de 4 cases, il voit un danger et :
 - (a) Si vous êtes amis et à proximité : il court vers votre poche,
 - (b) S'il y a un arbre : il court vers l'arbre pour s'y réfugier,

(c) S'il y a un buisson : il court vers le buisson pour s'y réfugier,

(d) Sinon : il court dans le sens oppose au danger pour se placer dans la première case vide.

Un écureuil rassasié ne se préoccupe que du danger (point 3 ci-dessus) puisqu'il ne cherche pas à manger. Il ne fait pas de stocke et ne mange pas.

S'il ne voit aucun danger, il se déplace dans une des cases vide la plus proche de lui.

Les renards attaquent les écureuils qui se déplacent dans une case adjacente à leur case et les tuent s'il n'y a pas d'arbre dans une case adjacente à l'écureuil et sinon les font fuir dans l'arbre. Les renards se déplacent aléatoirement d'une case à chaque tour et uniquement dans les cases vides.

Intelligence artificielle pour le thème Jungle

Vous devez mettre en place une intelligence artificielle pour les animaux de la jungle.

Vous voulez permettre au personnage d'appriivoiser un singe pour en faire son acolyte.

Les singes se déplaceront différemment selon leur état et suivant ce qu'ils rencontrent à proximité.

Un singe affamé deviendra rassasié s'il se nourrit.

Un singe rassasié sera de nouveau affamé au bout de 3 tours.

Un singe affamé qui se trouve sur une case adjacente au personnage qui détient une banane deviendra son ami car le personnage lui donnera la banane.

Si le singe mange un champignon hallucinogène, il perd la notion du danger pendant 3 tours puis ensuite sera de nouveau affamé.

Le déplacement d'un singe affamé dépend de ce qu'il voit autour de lui en suivant l'ordre de priorité ci-dessous :

1. Si à moins de 4 cases, il voit un danger et :

(a) S'il y a un cocotier : il s'y réfugie,

(c) S'il y a un buisson : il court vers le buisson pour s'y réfugier,

(d) Sinon : il court dans le sens oppose au danger pour se placer dans la première case vide.

2. S'il voit une banane, il va la chercher et la mange.

3. S'il voit un champignon: il va le chercher et le mange.

Un singe rassasié ne se préoccupe que du danger (point 1 ci-dessus) puisqu'il ne cherche pas à manger. Il ne fait pas de stocke et ne mange pas. S'il ne voit aucun danger, il se déplace dans une des cases vide la plus proche de lui.

Quand le singe devient ami avec le personnage, il ne se préoccupe plus du danger pendant 2 tours et suit le personnage, c'est à dire qu'il reste à moins de 2 cases de lui.

Les serpents sont sur le sol ou cachés dans les buissons. Les serpents cachés dans les buissons attaquent les singes qui s'y réfugient. Les serpents se déplacent aléatoirement de deux cases s'ils sont sur le sol. S'ils arrivent dans un buisson, ils restent 5 tours dans le buisson avant de repartir.

II- Rendu du travail

1 - Première partie à rendre avant le 16 Décembre à 19h.

Vous déposerez sur le cours en ligne de CELENE dans le dépôt nommé **Dépôt Conception projet** correspondant à votre groupe une archive au format zip contenant les fichiers suivants au format pdf :

- Un ou plusieurs fichiers contenant les diagrammes de classes de conception. Vous pourrez découper votre diagramme en plusieurs parties pour améliorer la lisibilité.
- Un fichier explicatif justifiant vos choix de conception et plus particulièrement vos choix de designs patterns.

2 - Deuxième partie à rendre avant le 04 Janvier à 19h.

Vous devez implémenter le jeu complet avec une interface textuelle pour définir les déplacements du personnage et l'affichage de la carte du jeu. Votre code suivra une architecture MVC de manière à répartir le code en trois package.

Le package `vue` contiendra une seule classe `Ihm` permettant d'interagir avec l'utilisateur et d'afficher les informations.

Le package `controleur` contiendra la classe `Controleur` qui sera en charge du déroulement des parties et qui interagira avec les classes du modèle et la classe `Imh`.

Vous ajouterez une classe `Cataclysm` qui contiendra le `main`.

Vous utiliserez un dépôt Git qui vous sera fourni une fois les groupes définis. Le dépôt devra contenir

- le document justifiant les patterns
- une image pour chacun de vos diagrammes
- un dossier contenant l'implémentation avec un README donnant des indications sur ce qui a été réalisé pour l'IA de chacun des deux thèmes.
- un dossier contenant l'implémentation du jeu complet avec également un README expliquant le déroulement du jeu.

III - Quelques indications pour la couleur

Pour afficher de la couleur dans la plupart des consoles, voici quelques informations.


Vous pouvez mettre de la couleur dans la sortie standard en utilisant les valeurs ci-dessous :

// Strings for foreground colors

```
public static final String ANSI_RESET = "\u001B[0m";  
public static final String ANSI_BLACK = "\u001B[30m";  
public static final String ANSI_RED = "\u001B[31m";  
public static final String ANSI_GREEN = "\u001B[32m";  
public static final String ANSI_YELLOW = "\u001B[33m";  
public static final String ANSI_BLUE = "\u001B[34m";  
public static final String ANSI_PURPLE = "\u001B[35m";  
public static final String ANSI_CYAN = "\u001B[36m";  
public static final String ANSI_WHITE = "\u001B[37m";
```

// Strings for background colors

```
public static final String ANSI_BLACK_BACKGROUND = "\u001B[40m";  
public static final String ANSI_RED_BACKGROUND = "\u001B[41m";  
public static final String ANSI_GREEN_BACKGROUND = "\u001B[42m";  
public static final String ANSI_YELLOW_BACKGROUND = "\u001B[43m";  
public static final String ANSI_BLUE_BACKGROUND = "\u001B[44m";  
public static final String ANSI_PURPLE_BACKGROUND = "\u001B[45m";  
public static final String ANSI_CYAN_BACKGROUND = "\u001B[46m";  
public static final String ANSI_WHITE_BACKGROUND = "\u001B[47m";
```

Voici un petit exemple qui affiche 

```
System.out.println(ANSI_PURPLE_BACKGROUND+ANSI_WHITE+"Hello  
world !!!"+ANSI_RESET);
```

Pour plus d'informations sur les couleurs : https://en.wikipedia.org/wiki/ANSI_escape_code#Colors