

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 tour par tour.



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

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

Votre code suivra une architecture MVC de manière à répartir le code en trois packages.

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 du jeu et qui interagira avec les classes du modèle et la classe `Ihm`.

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

Le travail sera à réaliser en trinôme selon le calendrier suivant :

1. Présentation et rendu de la première itération du projet le 11/12/24. Dépôt du travail sur CELENE.
2. Présentation et rendu de la seconde itération du projet le 09/01/25. Dépôt du travail sur CELENE.

Le sujet de la seconde itération sera disponible le 28 Novembre.

Pour la première itération, la notation prendra en compte :

1. L'analyse avec le diagramme de cas d'utilisation et rédaction des scénarios et un diagramme de classe décrivant la modélisation du domaine.
2. Un document expliquant les design patterns utilisés et leur justification.
3. Un diagramme de conception intégrant les design patterns.
4. La mise en place du jeu correspondant à la première itération.

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 **20 Novembre 2024**. 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 trinôme avec des étudiants d'un autre groupe que le votre. Inscrivez votre trinôme dans un des groupes. Nous réorganiserons les groupes pour les équilibrer si besoin.

Présentation générale du jeu

Dans le jeu, votre personnage peut se déplacer, se battre, ramasser des objets à l'aide d'un menu textuel. Il peut aussi reposer un objet ou un petit animal apprivoisé. Il peut lancer de la nourriture pour attirer un animal affamé et tenter de l'apprivoiser, la nourriture atterrira dans une case adjacente au personnage dans la direction indiquée via le menu. Le personnage pourra protéger l'animal apprivoisé des prédateurs. L'animal apprivoisé pourra avertir le personnage de certains dangers en se glissant dans sa poche ou en montant sur son épaule.

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.

I-Travail à réaliser pour l'itération 1

La zone de jeu est représentée comme une matrice de caractères. Toute l'interface se fera dans un terminal. **Dans l'itération 1, la forêt est sans danger, il n'y a aucun prédateur.**

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



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
- Petit rocher
- Banane
- Champignon
- 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. 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 écureuils de la forêt.

Les écureuils agiront 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.

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 dans une case à côté de lui: il va le chercher et le mange. L'écureuil prend la place du gland.
2. S'il voit un champignon dans une case à côté de lui: il va le chercher et le mange. L'écureuil prend la place du champignon.
3. Sinon il se déplace dans une case adjacente aléatoirement.

Un écureuil rassasié ne fait pas de stock et ne mange pas. Il se déplace dans une des cases vides la plus proche de lui.

Intelligence artificielle pour le thème Jungle

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

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 nourrit sur une case adjacente au personnage, il commence à l'apprécier. Après deux nourritures consécutives à proximité, le singe devient son ami.
- Si le personnage donne un coup au singe, il ne sera plus son ami.

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

1. S'il voit une banane dans une case à côté de lui, il va la chercher et la mange.
2. S'il voit un champignon dans une case à côté de lui: il va le chercher et le mange.
3. Sinon il se déplace dans une case adjacente aléatoirement.

Un singe rassasié ne fait pas de stocke et ne mange pas. Il se déplace dans une des cases vide la plus proche de lui.

II- 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