

1 Introduction





Vous êtes un grand fan de Cataclysm: Dark Days Ahead <https://cataclysmdda.org/>. Un jeu au tour par tour de survie dans un monde post-apocalyptique. Ce jeu vous plaît particulièrement car il est open-source et qu'il utilise le terminal comme interface graphique. Vous avez très envie de contribuer au développement de ce projet, mais vous n'êtes pas sûr d'avoir le niveau. Pour vous rassurer, vous décidez d'essayer de faire une version simplifiée du jeu dans votre langage préféré : JAVA.





Figure 1: Impression écran de Cataclysm: Dark Days Ahead












2 Travail à réaliser en binôme

La zone de jeu est représenté 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. L'ensemble minimum des différents types de cases que vous souhaitez faire est :

-  : le personnage jouable,
-  : un écureuil (la couleur peut varier en fonction du statut de l'écureuil),
-  : un arbre,
-  : un buisson,

-  : un gland,
-  : un champignon,
-  : un chat (qui fait peur aux écureuils),
-  : une zone vide.

Voici l'ensemble des fonctionnalités que vous devez mettre en place :

- Chargement d'une carte depuis un fichier :
Dans un premier temps, vous voulez utiliser un format de fichier très basique qui contient uniquement les caractères correspondants à la carte. Cependant, vous voulez dans un second temps¹, mettre en place un format plus abouti qui enregistre des informations supplémentaires comme les statistiques de chaque personnage, la croissance des plantes ou encore le niveau de charge de batterie des équipements.
- Une intelligence artificielle pour les personnages non-joueurs. Vous pensez commencer par travailler sur les écureuils, vous êtes un fan de Scrat. Vos écureuils auront différents comportements qu'ils effectueront dans l'ordre de priorité ci-dessous :
 1. Un écureuil junkie se déplace aléatoirement.
 2. S'il voit un gland : peu importe le danger (Scrat spirit inside), il va le chercher.
 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,
 - (d) Sinon : il court dans le sens opposé au danger.
 4. S'il voit des champignons : il va sur les champignons.
- Les états physiologico-psychologique des écureuils. Vous voulez permettre au joueur d'appivoiser un écureuil pour en faire son acolyte. Vous vous décidez donc de mettre en place un statut physiologico-psychologique avancer pour ce dernier.
 - Un écureuil affamé () deviendra rassasié () s'il est nourri.
 - Un écureuil rassasié () sera de nouveau affamé () au bout de 5 tours.
 - Un écureuil affamé () qui est nourri sur une case adjacente à la votre deviendra ami ()
 - Si vous tapez un écureuil ami () , il ne sera plus ami ()
 - Si un écureuil mange un mauvais champignon, il devient junkie () pendant 5 tours.

3 Travail à remettre pour le 10 janvier

1. Vous devez réaliser un diagramme de classes en indiquant précisément les patterns utilisés. Vous écrirez également un document justifiant les choix faits. Vous découperez votre diagramme en plusieurs parties pour plus de lisibilité (un diagramme simplifié général et des diagrammes précis pour chaque partie).
2. Vous devez implémenter l'IA de l'écureuil et la validation de cet IA avec des tests unitaires.

¹c'est une évolution à prévoir mais ce n'est pas demandé dans le cadre de ce devoir.

3. Enfin vous devez implémenter le jeu complet avec une interface textuelle pour définir les déplacements du joueur et l’affichage de la carte du jeu.
4. Les instructions pour le rendu seront données par mail mais vous devrez utiliser un dépôt git qui vous sera fourni ultérieurement. 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 l’écureuil
 - un dossier contenant l’implémentation du jeu complet avec également un README expliquant le déroulement du jeu.

4 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 Hello world !!!.

```
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