

# Rapport de conception du projet de programmation objet 2024- 2025

Licence d'informatique -- 2<sup>ème</sup> année  
Faculté des sciences et techniques de Nantes

## Fight Legendary Alien Nest (FLAN)



(images libre de droit)

Présenté par :

- Groupe TP : 385L :
- Lien vers le projet GitLab : <https://gitlab.univ-nantes.fr/E232257D/poo.24.25.20>
- Composition de l'équipe :

Nom	Prénom	Adresse mail
Bernini	Daniele	E233136J@etu.univ-nantes.fr
Roger	Gwenolé	E232257D@etu.univ-nantes.fr
Boucherau	Noé	E233136J@eu.univ-nantes.fr
Bon	Romain	E234774P@eu.univ-nantes.fr
Michon	Solène	E232121F@etu.univ-nantes.fr

- Encadré par : LINO Christophe

### *1.Cahier des charges*

## Contexte

Le jeu est destiné aux personnes voulant se divertir. Mais peut aussi être un très bon divertissement pour développer ses capacités cognitives car son Gameplay repose sur la stratégie. Si vous souhaitez jouer à un jeu qui mêle réflexion et divertissement, ce jeu est fait pour vous.

## Fonction

Ce jeu se joue en contrôlant un nombre limité de soldats se déplaçant sur une carte, le tout affiché sur un terminal de commande. Votre but est de tuer des ennemis aliens métaphores de la situation américaine. La fin étant liée à cette situation.

On doit pouvoir : déplacer un personnage, attaquer l'ennemi, se soigner, utiliser des objets, se défendre.

Par exemple on choisit un soldat en cliquant dessus puis on choisit l'ennemi que l'on souhaite attaquer en cliquant aussi dessus, il faut cependant être à proximité de ce dernier pour pouvoir l'atteindre. Ou alors on peut choisir de se déplacer pour par exemple se rapprocher de l'ennemi, l'éviter, ou trouver/utiliser un objet. Chaque interaction se base sur l'endroit où clique le joueur. la partie se termine quand on a terminé le nombre de tour prédéfinis ou tuer tous les ennemis les aliens n'attaquent que le jour car la nuit ils dorment.

## 2. Architecture

### Description générale

*Les classes/Interface sont écrites en **Gras***

La classe **Jeu** est celle qui sera appelée pour lancer le jeu, elle fera appel à :

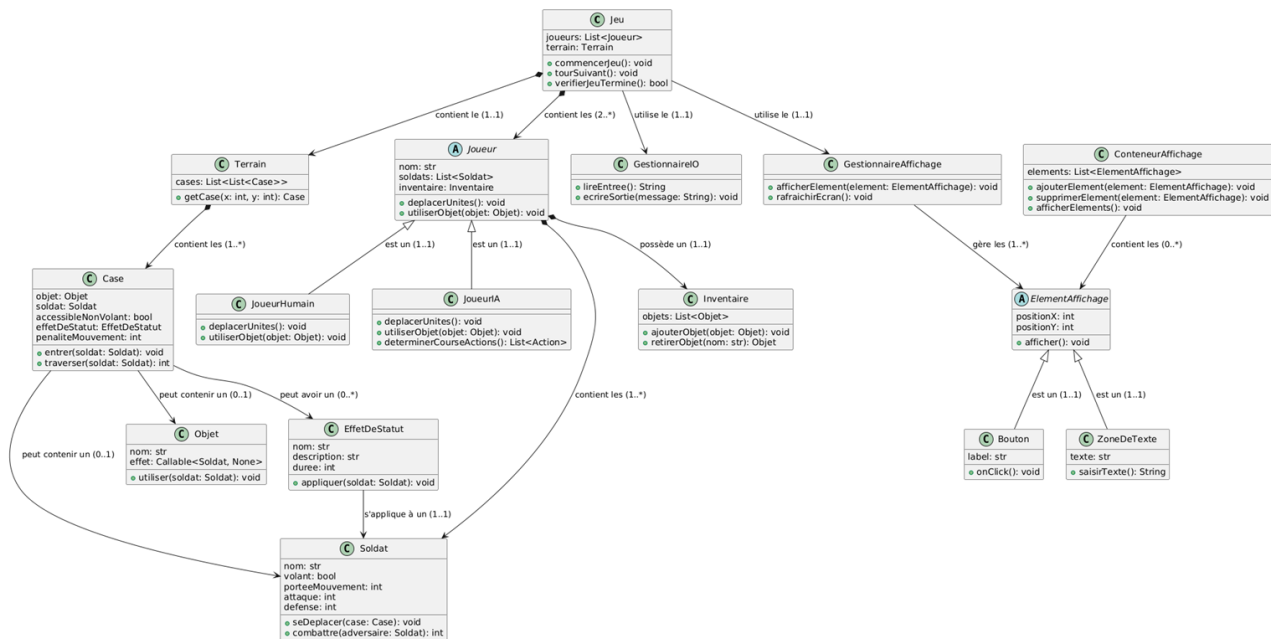
- La classe **Terrain**, qui sera elle-même composée :
  - o D'une liste de **Case** une classe représentant les cases du plateau de jeu. Ces dernières peuvent contenir :
    - une particularité spécifique (**EffetDeStatut**) qui s'appliquera à un **Soldat**
    - un **Objet**, une classe abstraite mère des classes :
      - BoostDega , BoostPorte, BoostSante, MalusDega, MalusPorte, MalusSante
    - un **Soldat**, correspondant au personnage que vous contrôlerait et qui pourra lui attaquer, se défendre...
- La classe abstraite **Joueur**, qui est définie par :
  - o Une liste de **Soldat**
  - o Un **Inventaire** contenant la liste d'**Objet** qu'a le Joueur

Elle sera la classe mère des classes :

- o **JoueurHumain** qui choisira de déplacer ou non l'unité (la liste de **Soldat**) ainsi que d'utiliser ou non un **Objet**
- o **JoueurIA** qui sera le bot contre lequel un joueur humain va jouer
- Un gestionnaire d'entrée/sortie (**GestionnaireIO**)
- Un **GestionnaireAffichage** qui s'occupera lui de l'affichage dans le terminal, il est gère :
  - o Les **ElementAffichage**, une classe abstraite mère des classes :
    - **Bouton** qui va récupérer l'endroit cliqué sur le terminal pour le renvoyer au **ConteneurAffichage**
    - **ZoneDeTexte**

Il y a également une classe **ConteneurAffichage** qui garde une liste des **ElementAffichage**, c'est dans cette classe que sont implémenter les différentes méthodes permettant d'ajouter, supprimer et afficher les éléments.

## Diagramme



## Interfaces

L'interface graphique en elle-même est assez simple faute de temps on a un canvas qui affiche la carte et les soldats ; un mini terminal qui affiche les logs un autre optionnel pour entrer des commandes et un dernier menu pour les caractéristiques des soldats on a aussi un bouton pour passer au tour suivant

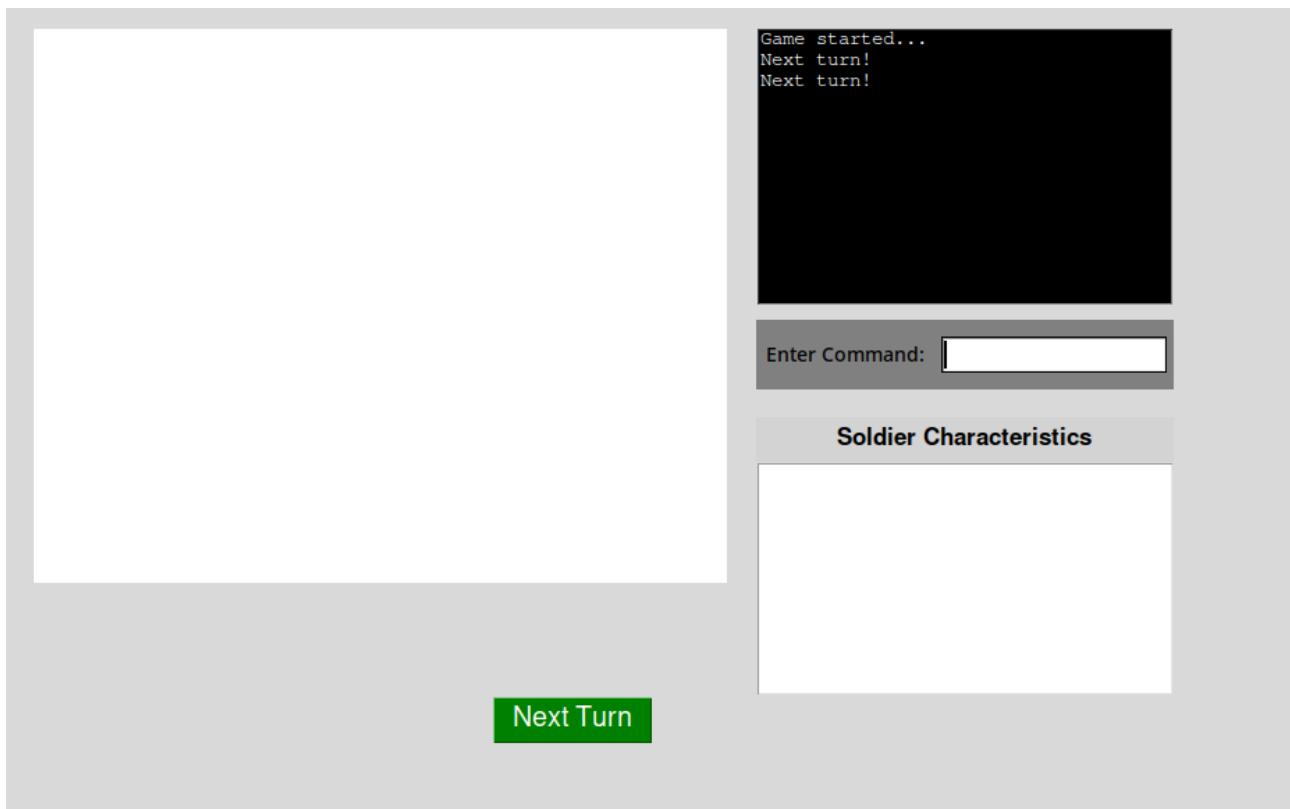


Schéma non contractuel fait en quelques minutes avec tkinter

### *Aspect spécifique*

Nous utilisons une version de l'algorithme wfc pour la génération de terrain.

Et nous avons choisi de ne pas utiliser d'interface graphique mais de nous en rapprocher le plus possible en utilisant les moyens mis à notre disposition.

### *3.Regard critique*

La mise en œuvre des concepts de la POO à été globalement rapide et efficace pour tout les membres du groupe.

Le plus gros problème rencontré à été de comprendre les différentes relations entre toutes les classes.

La quantité de travail fournie semble pour l'instant suffisant pour l'avancement du projet.(malgré quelque difficultés pour le premier rendu)

Cependant notre groupe rencontre des problèmes de communication, sur les différentes tâches à effectuer mais également sur l'avancée personnelle de chaque membre, dans le sens où les avancées faites personnellement ne sont pas partagées.

De plus, certains membres ne semblent pas suivre les distributions des tâches faites.

Bien évidemment le groupe reste motivé à l'idée de donner de son meilleur et de fournir un travail réussi et complet. Car nous sommes globalement satisfaits de la direction qu'a pris le projet et pensons qu'ils utilisent bien chaque notion du cours. Le projet nous permet à tous de développer nos compétences en programmation orientée objet et pensons avoir progresser dans ce domaine.