

Projet POO

Colossal_Cave_Adventure

SOMMAIRE :

Présentation du Jeu

Diagramme UML

Règle du Jeu

Comment Jouer

Problème Rencontrer

Conclusion

INTRODUCTION :

Dans le Cadre de ce Projet de POO, nous avons décidé, d'aborder l'histoire de la Mythique quête du Graal comme thème, l'histoire est simple, Arthur et ses compagnons partent à la recherche du Graal. Dans ce périple nous l'accompagnons malgré nous dans divers lieux et combat Epique contre différent Monstre voire même des Personnages suspect. Une fois le Graal trouver, le but étant de l'utiliser afin d'aboutir à la fin du Jeu.

Présentation Du Jeu

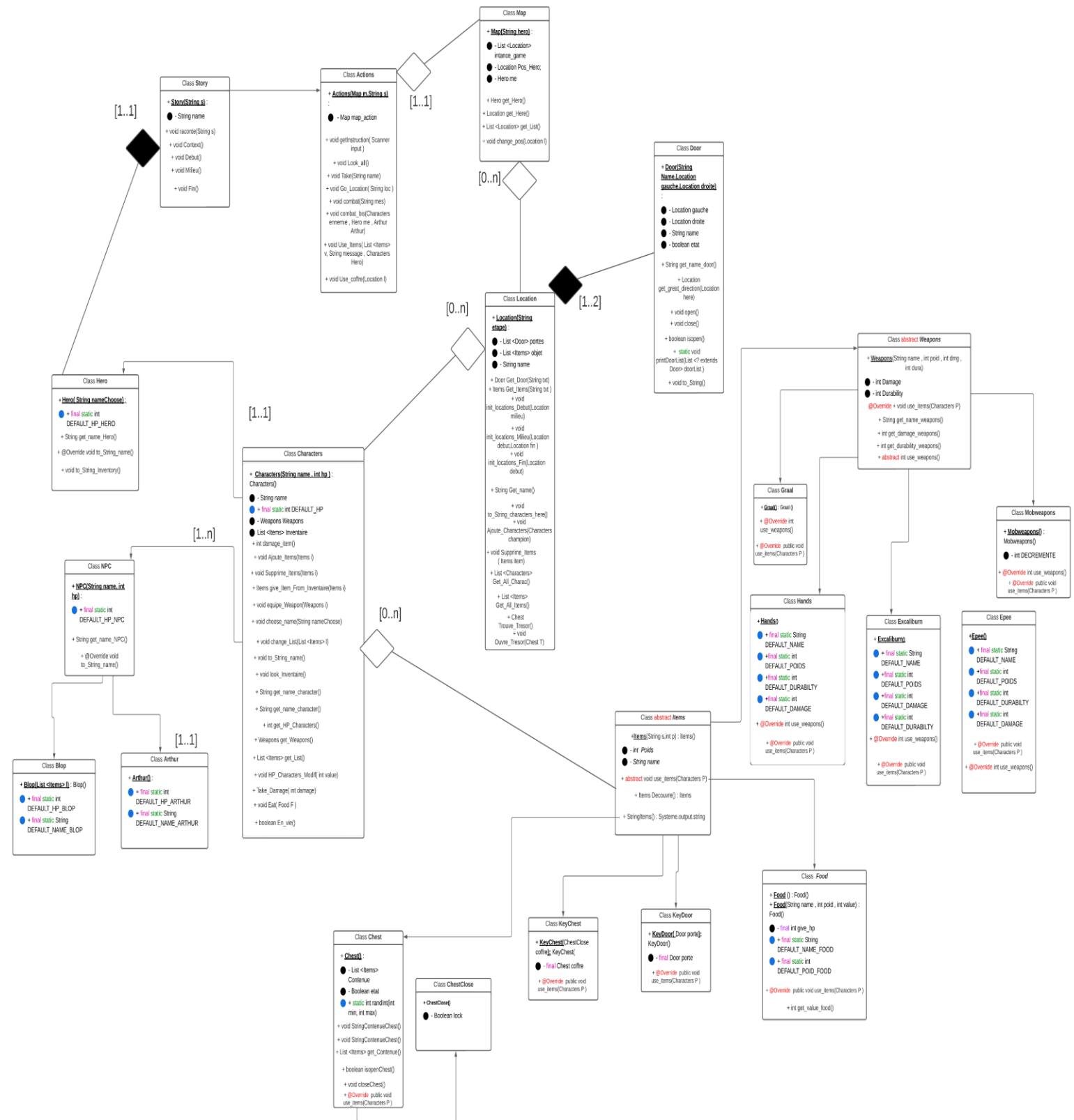
Vous commencez donc l'aventure avec un Compagnon (Arthur), et devrez effectuer diverse "Actions" afin de vous mener tout droit vers votre But ultime, "Trouver le Graal", puis "L'Equiper", pour cela vous aurez besoin de bien observer les alentours de chaque lieu que vous visiterez.

Observer (Lire) les actions/réactions des êtres vivant qui vous entoures. Et n'hésitez pas prendre les butins rencontrer en route.

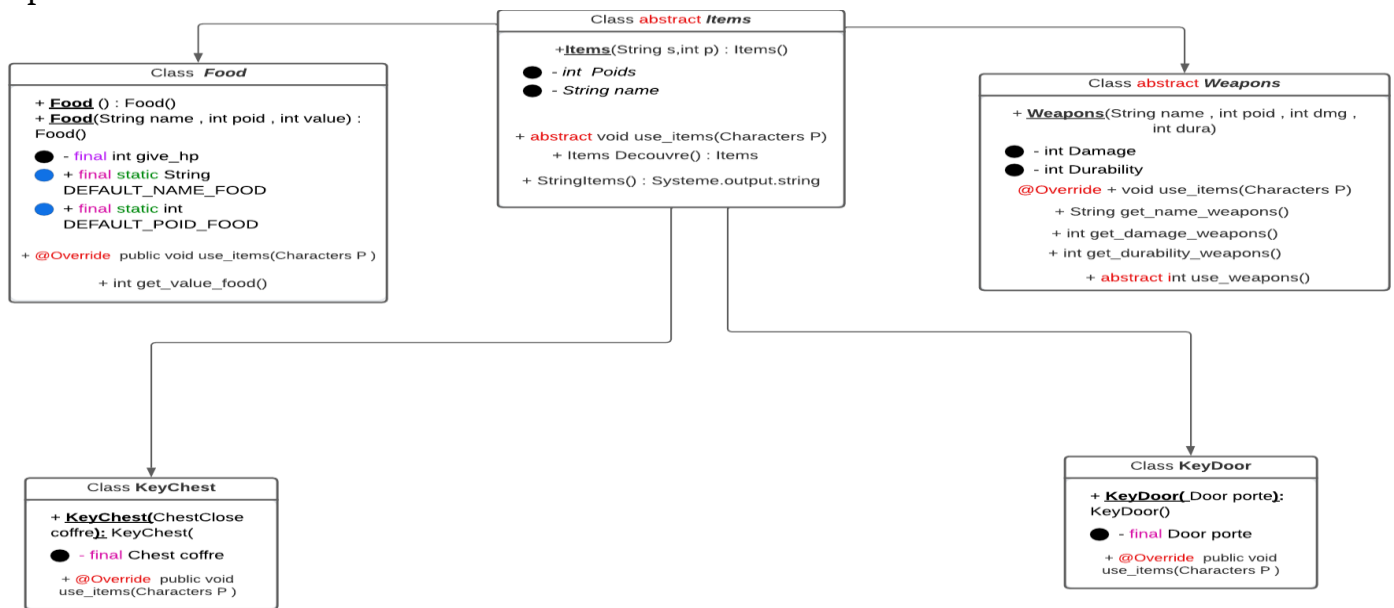
Il n'y a jamais d'impasse. La "clé" se trouve en vous, alors gardez foi dans votre aventure pour retrouver le Graal perdu et acquérir les bénédictions qu'il promet !

Diagramme UML :

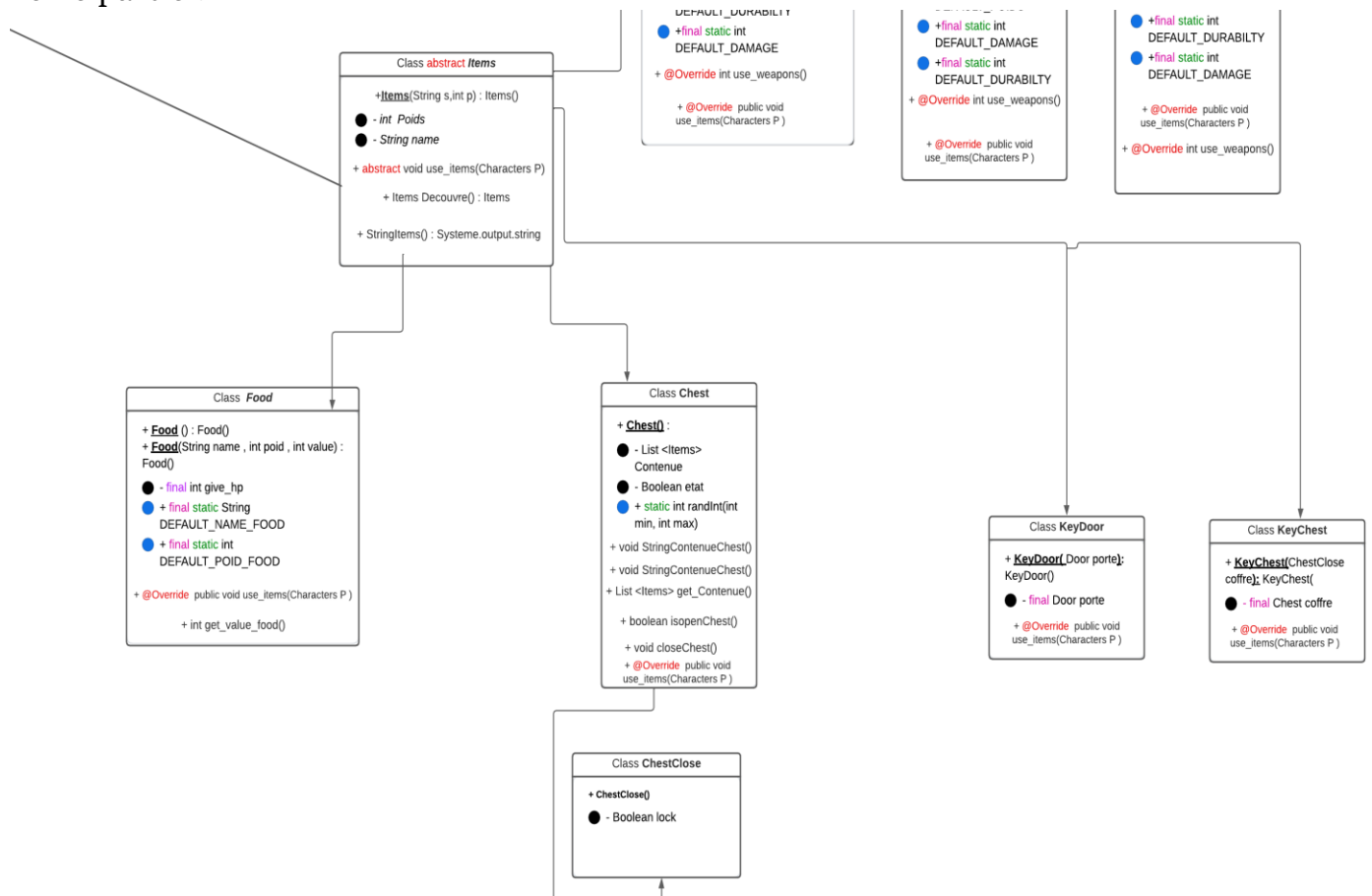
Voici le Diagramme Complet utiliser lors de la conception de notre Jeu



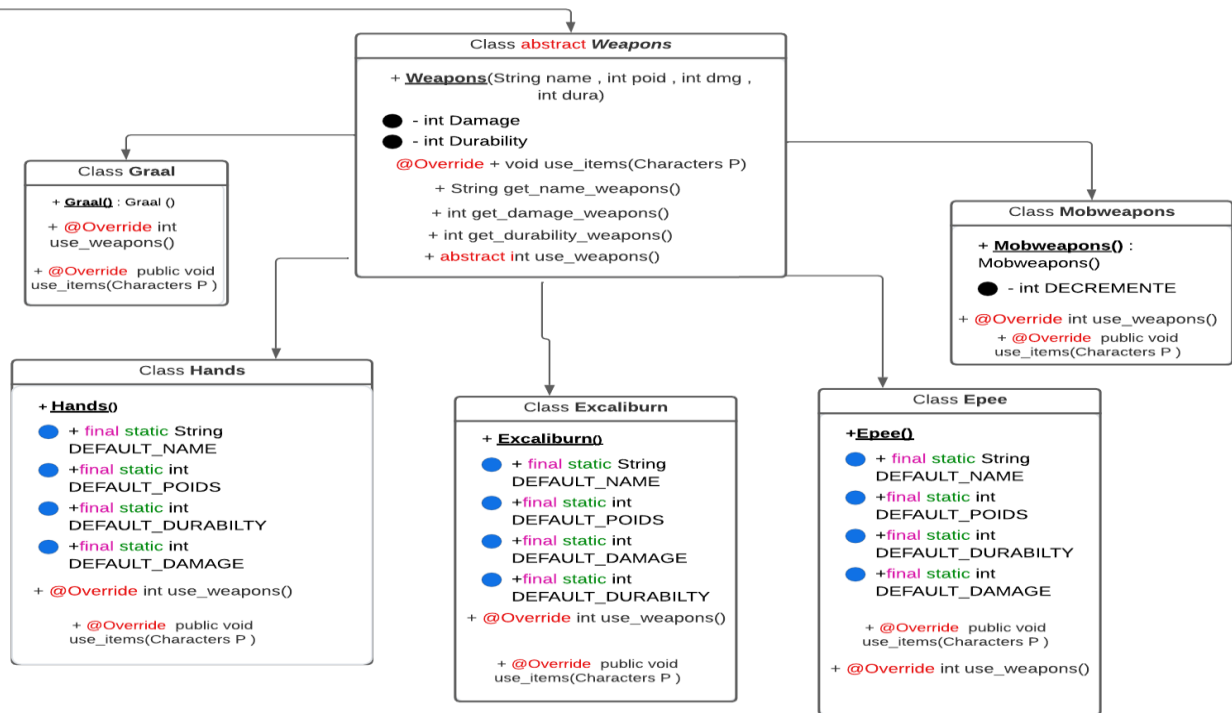
Voici le Diagramme vu part différente part et d'autre :
 En premier lieu voici un Zoom sur l'arborescence de Items



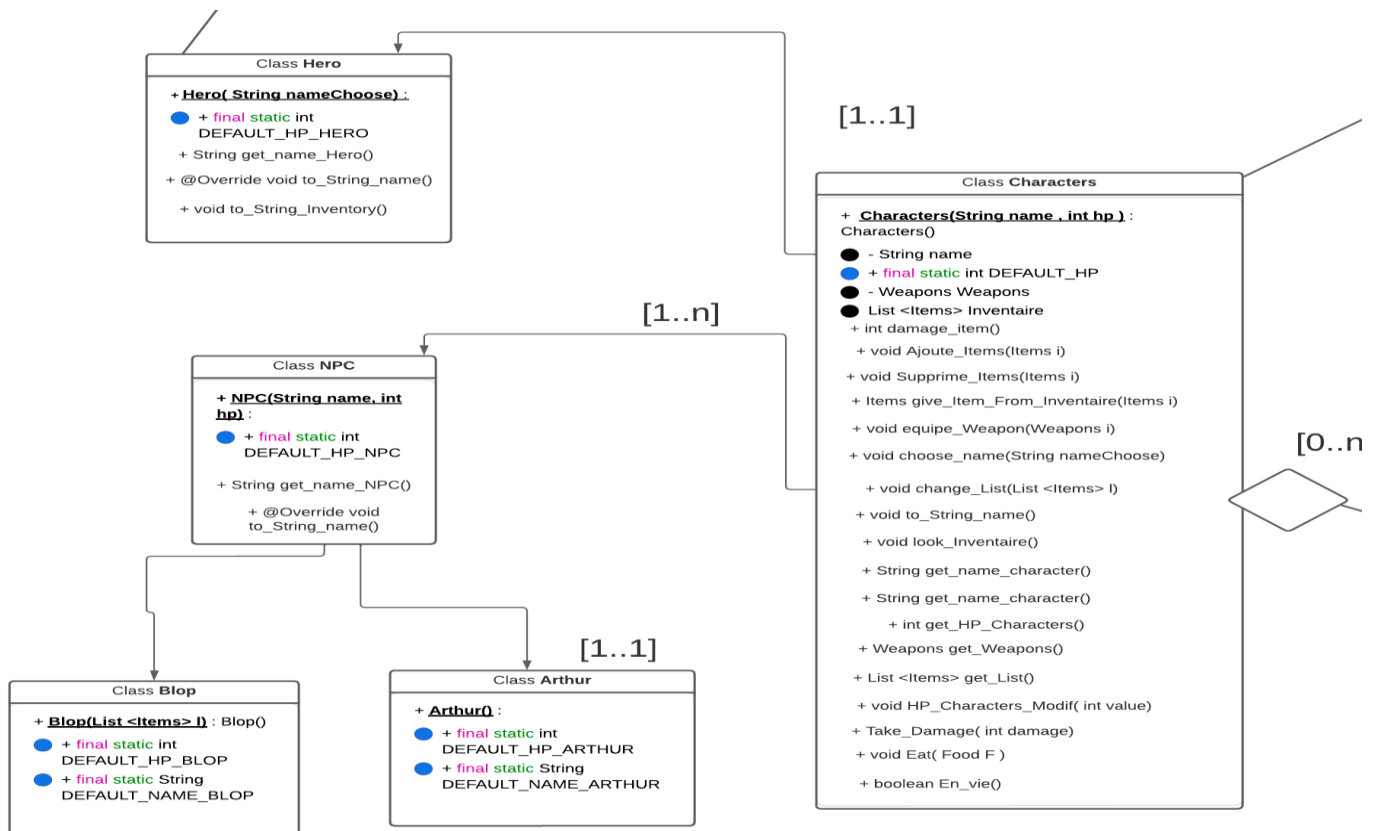
2eme partie :



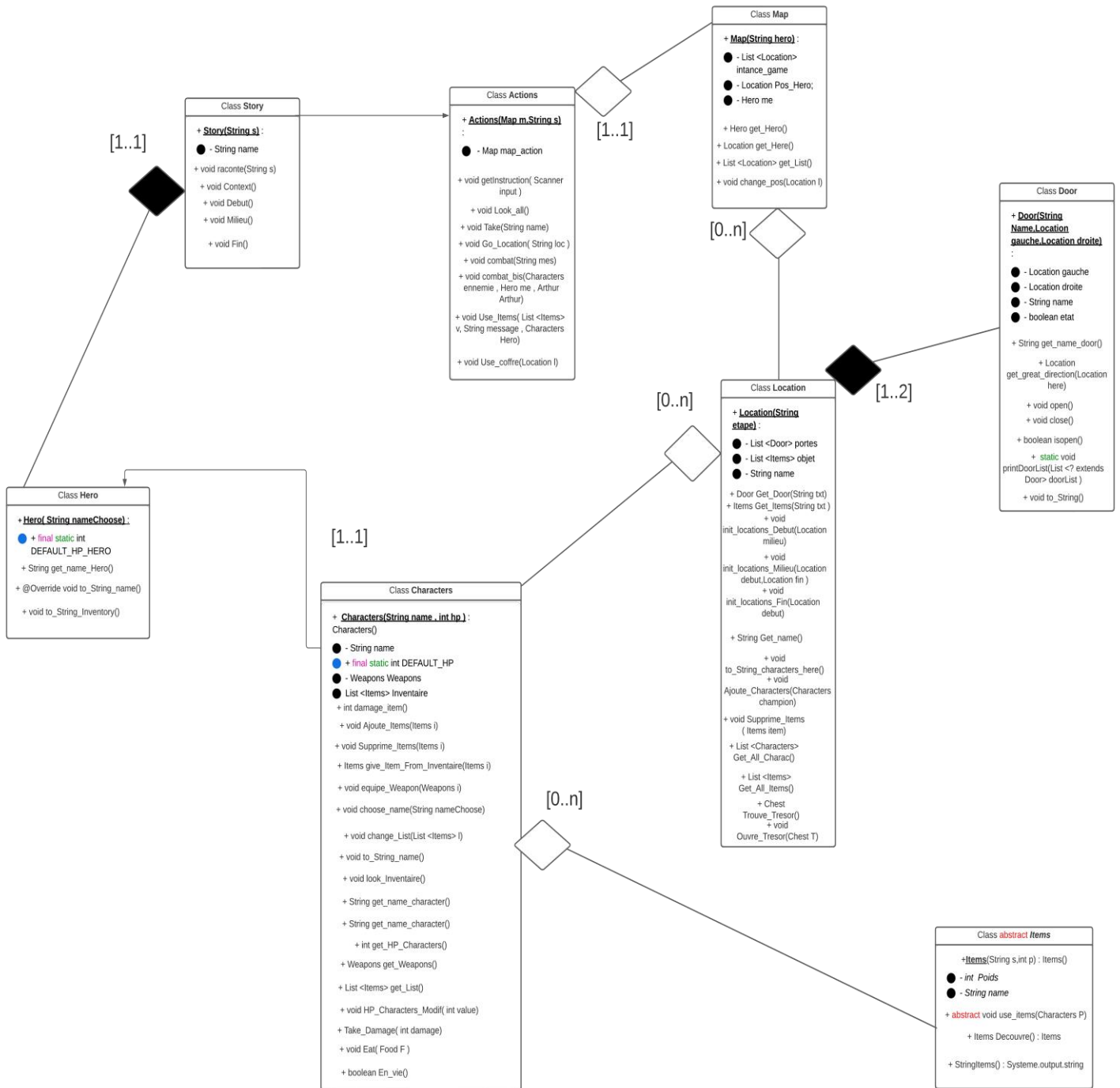
Voici un Zoom sur l'arborescence de Weapons



Voici un Zoom sur l'arborescence de Characters



Voici un Zoom sur l'arborescence des Lieux compris avec les actions et l'histoire :



Règle du Jeu :

Les Règles du jeu, consiste à envoyer, successivement des Commande jusqu'à atteindre, l'objectif donné. Dans l'histoire, pour cela, vous devez utiliser la liste des Commande donnée.

Si vous les oubliez, il vous suffit d'utiliser la Commande "Help", ou dans le cas d'une erreur l'ont Vous les rappelleras,

-Pour Observer autour de vous, vous disposez de la Commande LOOK ALL, Cependant LOOK avec un arguments différent vous permettra de vérifier votre inventaire, Ou l'arme que vous portez actuellement.

Attention, il n'est pas possible de regarder le contenu d'un coffre avant de l'avoir ouvert.

-Si vous souhaitez Equiper une Arme trouvée dans lieu, il vous faudra l'ajouter à votre Inventaire Avant même de penser à l'équiper, c'est à dire utiliser "TAKE" avant "USE"

-Si vous souhaitez combattre une ennemie précise, vous devez décrire son nom après la Commande COMBAT.

-Attention si vous souhaitez ouvrir un Coffre, il vous faut effectuer la Commande USE et non TAKE , il est possible d'ouvrir un Coffre sans l'ajouter dans l'inventaire, à conditions qu'il ne nécessite Pas de clé évidemment.

-Le Système de "Key" est très simple, chaque clé est directement reliée à son verrou, pour Déverrouiller une porte ou un Coffre, il suffit de ramasser la Clé et de l'utiliser une fois dans L'inventaire pour ouvrir directement l'objet fermé.

-Effectuer la Commande QUIT revient à quitter le Jeu de manière définitive, elle provoque également La mort définitive de votre personnage.

-Si votre Arme consomme ces points de DURABILITE, votre arme infligera moins de dégâts.

-Votre Compagnon Arthur, Subit moins de dégât que vous lors des combats, eh oui, il est bien plus Fort que vous.

-La Commande GO, permet également de revenir en arrière.

-Une fois un COMBAT lancé, il s'effectue jusqu'à la mort d'un personnage, on ne déserte pas Le champ de bataille.

Comment Jouer :

Installations :

Le programme provient de l'interpréteur NetBeans (JDK17), pour l'installer plusieurs méthodes, Mais relié à java :

Netbeans compile les fichiers exécutables au format JAR, repérez

Donc le fichier Colossal_Cave_Adventure.jar

que vous trouvez en ouvrant le fichier /dist

Fichiers exécutables Java s'exécutent sur n'importe quel ordinateur Windows ou Linux, de sorte que vous n'avez pas besoin de créer un fichier exécutable

pour chaque plate-forme de système d'exploitation.

Première Méthode :

Etape 1 :

Ouvrez l'application Netbeans dans le menu "Démarrer ". Ouvrez votre application de Netbeans

Etape 2 :

Cliquez sur le menu " Build" et sélectionner " Créer principale " pour compiler l'application dans un fichier Exécutable. Tant qu'il n'y a pas d'erreurs de syntaxe Java, le programme compile en un fichier exécutable

Etape 3 :

Cliquez sur le menu "Projet" . Cliquez sur " Exécuter le projet . "

Cette option vous permet d'exécuter le fichier exécutable et de test pour les erreurs de logique dans la conception.

Deuxième Méthode :

Effectuer la Commande de terminale suivante en vous plaçant dans le répertoire /dist

< java -jar Colossal_Cave_Adventure.jar >

Vous pouvez utiliser l'option --release *version de java utiliser* devant la commande

Commande :

GO – Permet de se déplacer d'une Locations a une autre, il est important d'indiquer Le chemin choisi, il est repérable et mis en évidence dans le texte de l'histoire

LOOK – Nécessite un Deuxième Argument parmi les suivants :

-ALL : Permet d'observer les environs, repérez les monstres, objets du lieu

-INVENTAIRE : Permet de vérifier son inventaire

-ARME : Vérifie votre Arme, sa durabilité et ses caractéristiques

TAKE – Permet de prendre un objet disponible sur le lieu présent.

COMBAT – Nécessite un Deuxième Argument Correspondant au nom d'un Monstre, repérable De pars la commande LOOK ALL.

QUIT – Met fin à la partie du joueur. Son personnage meurt également.

USE – Nécessite un Argument supplémentaire qui est le nom d'un objet se situant dans votre

INVENTAIRE, exception pour les Coffres dont il est possible de faire directement sans avoir

A placer le Coffre dans l'inventaire :

- Coffre : Permet d'ouvrir un coffre si celui-ci, si celui-ci n'est pas fermé à clé.
- WEAPONS : Equipe une Arme à votre Personnage.
- FOOD : Redonne à votre Hero des Points de vie.
- KEY : Déverrouillez, une porte ou un Coffre.

HELP – Permet d'afficher les COMMANDES possible.

Les Informations de COMMANDE sont disponibles dans le fichier README.txt situé dans Le fichier /dist du jeu.

Problème Rencontrer :

Lors de notre Projet Nous avons rencontré différents problèmes Majeurs, à savoir les Tests, ainsi que le manque de temps afin d'apporter un Projet plus Complet et Consistant comme nous l'imaginions.

Vous l'aurez compris, tous les Tests requis du jeu n'ont pas été effectués, l'essentiel des tests ayant été fait dans le Package Item, du fichier test. Et ce pour cause, nous avons eu du mal à utiliser l'interface de Test de NetBeans qui ne marchait pas, car nous n'utilisions pas JUNIT4.

Des Tests ont été faits sur les Boolean renvoyer, les entiers se rapportant au dégât infliger ou encore la durabilité, la vérification de la suppression d'un élément dans une liste.

Par ailleurs également la Commande LOOK, ou ne nous souhaitions pas ajouter l'argument ALL pour observer tout ce qui entoure notre Hero, car le test de lecture via `input.hasNest()`; n'a pas fonctionné comme nous l'imaginions. Nous avons donc évité de nous attarder sur le problème à cause d'autre priorité essentielle à nos yeux "avoir un jeu qui marche", mais la question a été étudiée, et nous aurions procédé via un Tableau de String contenant les **Scanner scan** effectuer.

Une note également, sur le manque de temps, en effet nous souhaitions introduire un système de reconnaissance d'objets ou même de String plus performant en utilisant la fonctionnalité du `HashCode()`, en effet le `hashCode` nous aurait permis, en étant fait dans une classe, de pouvoir retracer son identifiant, et par conséquent sa class, mais surtout un outil terriblement efficace quand il s'agit de distinguer deux mêmes objets d'une même class mais ayant des caractéristiques courantes différentes.

Conclusion

Malgré nos déceptions, vis-à-vis des attentes sur ce Projet et des problèmes rencontrés, nous avons quand bien même atteint notre Objectif sur Projet, à savoir rendre un jeu fonctionnel et répondre au cahier des charges dans son strict minimum, c'est à dire, les

Commandes, les items, les lieux, une condition de fin ou d'arrêt ainsi qu'un personnage interagissant avec le Hero.

Notre schéma de construction nous semble également adapté à l'objectif demander, même si des ajustements reste bien-sûr possible.