

Projet POO-IHM1.

Membres de Projet :

- Al Natour Mazen
- Caillaud Tom
- Fadli Mohamed



Rapport Projet

Jeu d'aventure textuel

Inspiration du Jeu

SCP - Containment Breach

L3 informatique , Groupe 7



TABLE DES MATIERES

I.	Documentation Utilisateur	3
II.	Documentation Développeur	5
III.	Répartition des tâches	10

I. Documentation Utilisateur :

Pour jouer à notre jeu, nous vous proposons 2 solutions :

- 1- Une version sous le logiciel Eclipse, où vous allez importer le projet « POO_Project_G.7_eclipse » et par la suite ouvrir le package game, puis ouvrir le fichier SCP_CB et lancer le jeu.
- 2- Une Version sous Linux, où vous allez vous placer dans le même répertoire que le Makefile et vous aurez accès à plusieurs commandes pour jouer :
 - La commande “ make all ” :
 - Permet de compiler tous les fichiers du projet pour pouvoir lancer le jeu.
 - La commande “ make start ” :
 - Permet de lancer le jeu
 - La commande “ make clean ” :
 - Permet de supprimer les fichiers .class générés par la commande make all

Comment jouer à notre jeu maintenant ?

Une fois le jeu installé et lancé, vous êtes perdu dans une salle obscure. Vous entendez des bruits, des hurlements de partout mais vous entendez surtout le bruit d'une alarme. Vous avez peur, votre instinct comprend que vous devez vous échapper d'ici le plus rapidement possible en relevant quelques défis et sans mourir face aux différents obstacles. Vous pourrez utiliser la commande "help" dans le jeu pour connaître toutes les actions et commandes possibles.

Bonne chance aventurier.

Comment jouer à notre jeu, mais d'une manière plus facile (pt.1) ?

(/\ \ ATTENTION SPOIL)

Vous trouverez un dossier nommé « Livrables » contenant différentes choses mais contenant surtout une image nommée “ SCP_CB_Map ”. Dans celle-ci, vous trouverez la carte qui représente notre jeu et qui vous permettra de jouer plus facilement.

Les numéros des salles données sur “ SCP_CB_Map ” correspondent dans le jeu à :

Salle 0: Starting Room

Salle 1: The Lost Room

Salle 2: Luck Or Not Room

Salle 3: What Room

Salle 4: Slimy Room

Salle 5: Darkness Room

Salle 6: Victroy Room

Comment jouer à notre jeu mais d'une manière beaucoup plus facile (pt.2) ?

(/!\ ATTENTION SPOIL SOLUTION DU JEU)

Ici vous trouverez la solution de notre jeu.

Dans un premier temps, vous serez dans la **salle 0**, il n'y a qu'une seule porte pour quitter cette salle donc vous allez saisir la commande **go north**.

Vous êtes dans **la salle 1**, vous voyez 2 portes, une bloquée avec une image de pistolet (donc, pour ouvrir celle-ci, il faut avoir un pistolet sur soi.) et une autre ouverte, il faut donc saisir la commande **go east**.

Vous vous retrouvez maintenant dans **la salle 2**, vous voyez 2 portes, une porte à verrouillage automatique qui peut s'ouvrir seulement avec une carte d'accès, vous trouverez aussi un pistolet par terre, il faut le récupérer avec la commande **take gun**, ensuite il faut revenir dans la salle 1 avec la commande **go west**.

Vous êtes dans **la salle 1** de nouveau, mais maintenant la porte qui était bloquée est maintenant ouverte, il faut donc saisir la commande **go north** pour passer à la suite.

Vous êtes maintenant dans **la salle 3**, il y a le monstre scp-387, il ne faut surtout pas utiliser la commande use gun, ou sinon ce monstre va vous tuer et vous allez donc perdre la partie, il faut, à la place, utiliser la commande **go east** pour passer dans la salle suivante.

Vous êtes maintenant dans **la salle 4**, il y a le monstre scp-4049, c'est un monstre très collant donc il faut saisir la commande **go east** pour passer dans la salle suivante.

Vous êtes maintenant dans **la salle 5**, il y a le monstre scp-501, mais aussi une carte d'accès par terre, il faut donc saisir la commande **take keycard**, et traverser le passage secret via la commande **go south**, MAIS ATTENTION, il faut faire tout cela très rapidement car **le monstre se réveillera dans 25 secondes et vous tuera**.

Vous êtes de nouveau (si vous n'êtes pas mort) dans **la salle 2**, vous avez la carte d'accès, donc la porte se trouvant au sud va s'ouvrir en y allant en faisant un **go south**.

Finalement, vous êtes dans **la salle 6**, et vous avez réussi à vous échapper de ce laboratoire...

Mais est-ce vraiment la fin ?

À suivre.

II. Documentation Développeur :

L'objectif de notre conception de jeu était de faire un jeu facilement extensible. De ce fait nous avons conçu plusieurs classes abstraites avec des méthodes nécessaires à la construction d'un jeu textuel.

Nous pouvons citer par exemple quelques classes importantes :

- *La classe Location* : c'est une classe abstraite contenant un type de lieu comme attribut, mais aussi des sorties sous la forme d'une hashmap, ce sont les choses minimales à avoir pour construire une location dans n'importe quel jeu. Ainsi, avec notre conception, nous pouvons faire des châteaux, des salles, des laboratoires avec des attributs différents sans modifier les autres classes héritant de location.
- *La classe Item* : c'est une classe abstraite représentant un objet, sans préciser la catégorie de cet objet, ainsi nous avons conçu 2 catégories d'objets, les armes (la classe Weapon) et les utilitaires (la classe utilities). Nous pourrions donc ajouter d'autres objets dans ces catégories sans impacter les objets déjà présents dans le jeu, mais aussi créer d'autres catégories avec d'autres types d'objets.
- *La classe Character* : c'est une classe abstraite représentant un personnage avec des attributs classique comme un nom, un nombre de points de vie, mais aussi une location car un personnage dans n'importe quel jeu se trouve dans une location.

Revenons à notre univers, dans notre jeu, il y a un personnage qui est enfermé dans un laboratoire. Dans ce laboratoire, il y a différentes salles où il se passe des expériences scientifiques très étrange avec des créatures encore plus étranges. Ces créatures semblent être immortelles, il ne faut pas les sous-estimer. Notre personnage principal devra s'échapper de ce laboratoire en utilisant certains objets qui ne se trouvent qu'à certains endroits dans le laboratoire, ainsi, nous pouvons construire différentes classes et méthodes :

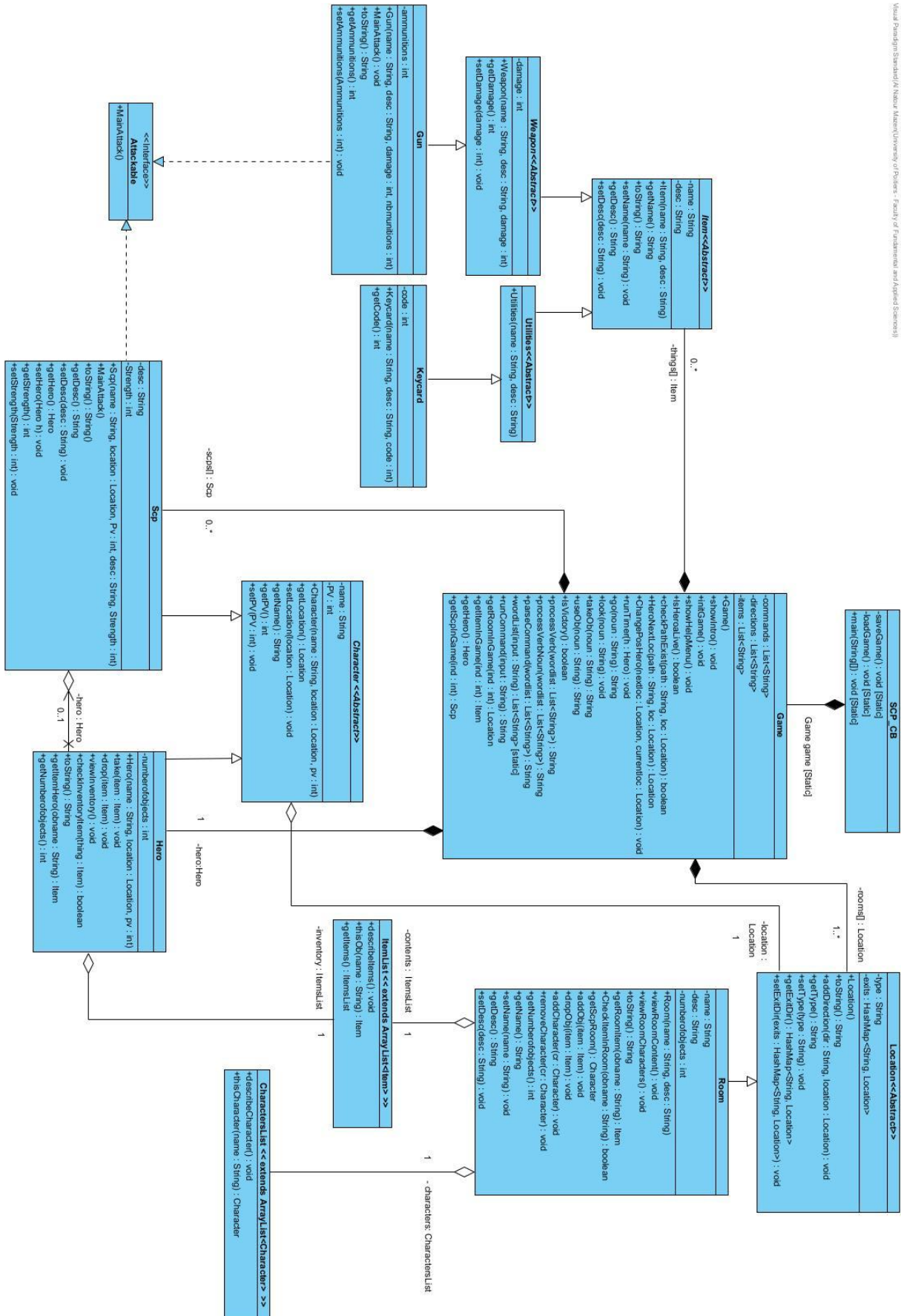
- *La classe Room* : cette classe représente les différentes salles dans le laboratoire, chaque salle a un nom et une description, mais elle peut avoir aussi des objets et des personnages. Nous pouvons récupérer les objets qui se trouvent dans cette salle mais aussi se déplacer dans d'autres salles.
- *La classe Hero* : cette classe représente notre personnage principal, celui qui doit s'échapper du laboratoire, ainsi notre personnage principal a un inventaire où il y placera des objets qu'il récupérera dans une salle.
- *La classe SCP* : cette classe représente les différentes créatures, ainsi on peut construire différentes créatures à partir de cette classe. Un SCP a un nom, une description et une force, cette classe implémente l'interface Attackable, qui permet à un scp d'avoir son attaque principale qui consiste à tuer n'importe qui du premier coup.
- *L'interface Attackable* : c'est une interface contenant une méthode « MainAttack » qui désigne le fait de pouvoir attaquer. Ainsi si une classe l'implémente, cela veut dire que cette classe peut attaquer.

- *La classe Gun* : cette classe représente un pistolet qui fait des dégâts avec des munitions, un pistolet implémente l'interface Attackable. D'autre part, nous avons besoin de cette classe pour pouvoir passer de la salle 1 à la salle 3.
- *La classe Keycard* : cette classe représente une carte d'accès avec un code. Dans notre jeu à l'état actuel, le code ne sert pas vraiment, car nous n'avons qu'une seule porte qui s'ouvre avec une carte d'accès. Cela a été fait de sorte à pouvoir faire plusieurs keycards différentes, une pour une porte.
- *La classe Game* : cette classe est la classe pilote du jeu, dans cette classe, nous allons instancier tous les items, personnages et salles nécessaires à la construction de notre univers, nous allons retrouver aussi toutes les méthodes nécessaires pour la saisie de commande, pour qu'un utilisateur puisse jouer. Nous nous sommes basés sur la liste des commandes proposées dans le sujet.
- *La classe SCP_CB* : c'est à partir de cette classe que le jeu peut démarrer, nous aurons donc la méthode main. D'autre part, nous avons ajouté la sérialisation d'une partie pour qu'un utilisateur puisse sauvegarder et recharger une sauvegarde en jeu.

Vous trouverez ci-dessous notre diagramme UML.

Ps : Vous trouverez dans le dossier « Livrables » ce diagramme sous forme d'une image où vous pourrez zoomer pour mieux voir.

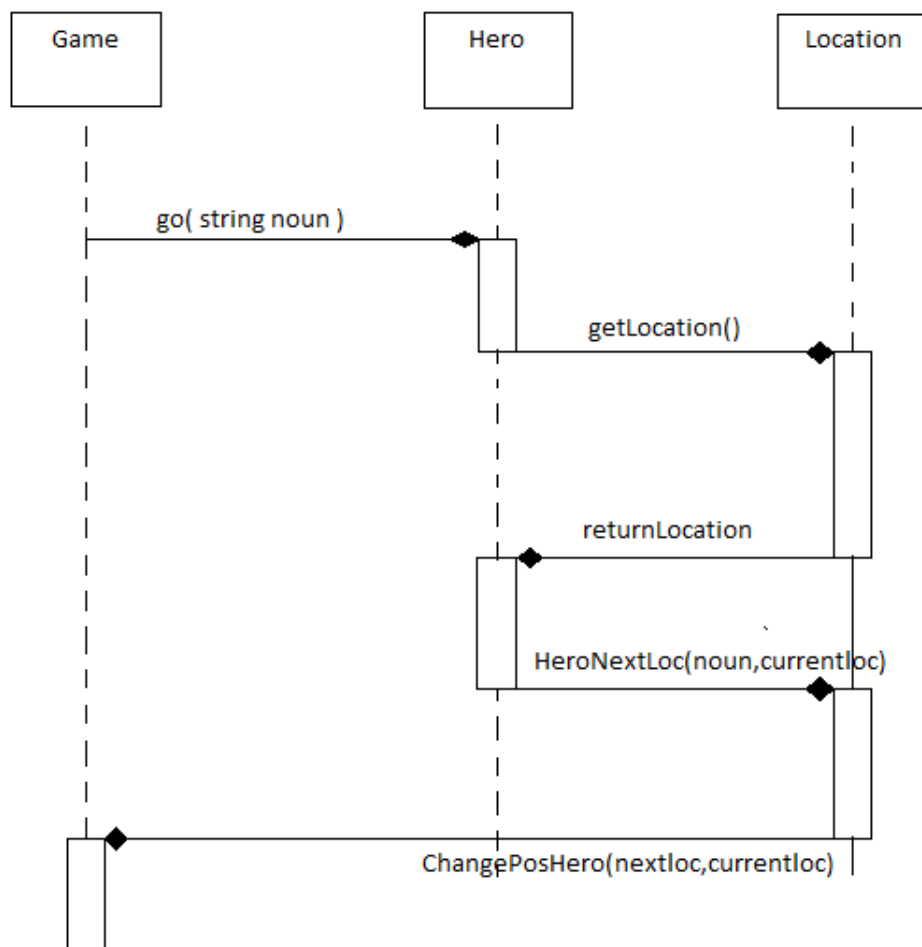
(Visual Paradigm Standard) (Natuurlijk Universiteit) of Physics - Faculty of Fundamentals and Applied Sciences)



Vous trouverez ci-dessous notre diagramme De Séquence.

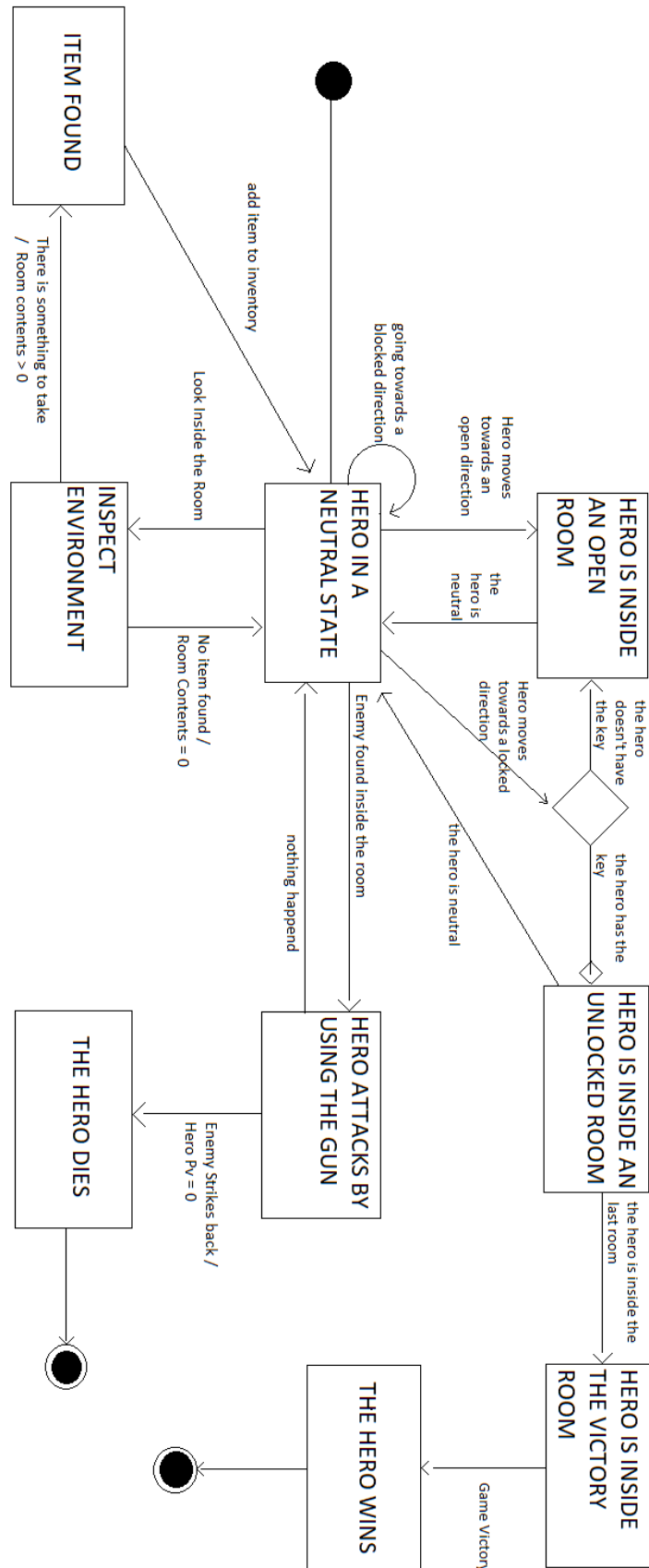
Ps : Vous trouverez dans le dossier « Livrables » ce diagramme sous forme d'une image où vous pourrez zoomer pour mieux voir.

Ps : nous avons développé pour ce diagramme qu'une fonctionnalité du jeu, nous aurions pu en développer d'autres comme par exemple take, look, use, etc...



Vous trouverez ci-dessous notre diagramme D'Etat.

Ps : Vous trouverez dans le dossier « Livrables » ce diagramme sous forme d'une image où vous pourrez zoomer pour mieux voir.



III. Répartition des tâches :

Vous trouverez ci-dessus La répartition des tâches durant le projet.

Al Natour Mazen	<ul style="list-style-type: none">• Rédaction du rapport/readme• Organisation de l'archive du rendu• Développement des classes/interface du jeu : Location, Room, Character, Hero, Scp, Item, Weapon, Gun, Keycard , Attackable• Implémentation de quelques méthodes dans la classe Game• Création du diagramme UML• Création d'un makefile pour la version linux• Conception de l'histoire du jeu
Caillaud Tom	<ul style="list-style-type: none">• Développement des classes/interface du jeu : Utilities, Game, SCP_CB• Implémentation de toutes les méthodes nécessaires à l'utilisations des commandes (le parseur)• La Sérialisation• Développement des tests• Conception de l'univers du jeu• Préparation de la soutenance• Conception de l'histoire du jeu
Fadli Mohamed	<ul style="list-style-type: none">• Développement des classes/interface du jeu : ItemList, CharacterList• Développement des tests• Création des diagrammes de Séquence et d'état