

Rapport de Projet Zuul

1.A Auteurs

Vadim Sitbon 9c

1.B Thème

SCP-014-B est un jeune Homme de type caucasien [REDACTED], il se trouve dans une installation souterraine de la fondation SCP sont but est de rétablir le courant pour les quatres SCP de classe Keter pouvant causer [REDACTED] et beaucoup d'autres problèmes.

1.C Résumé du scénario

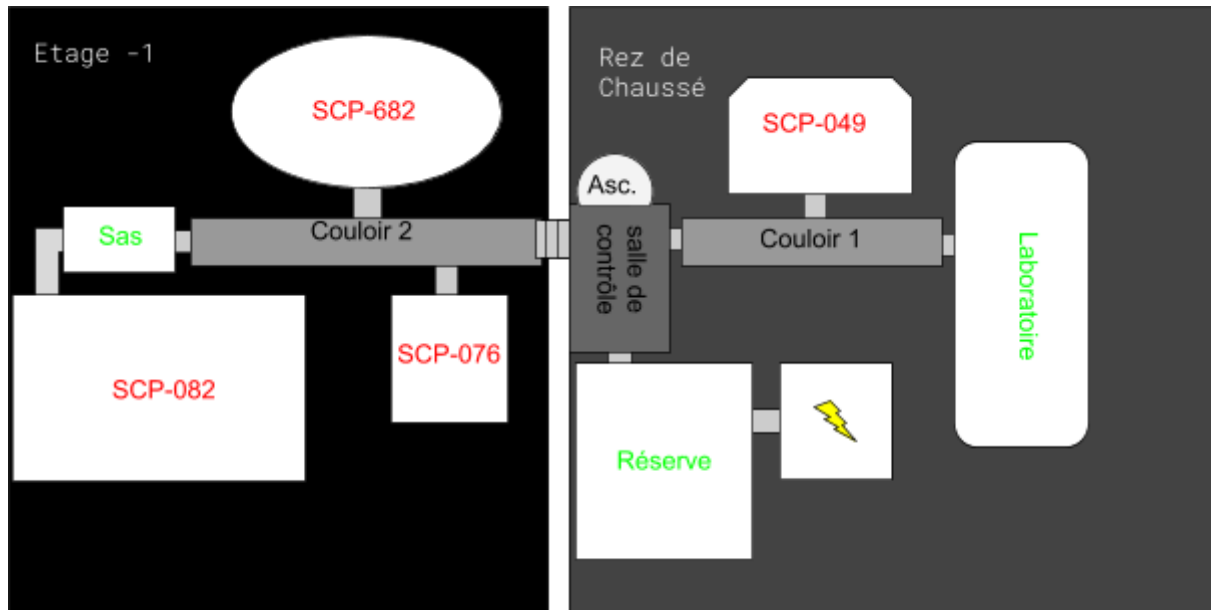
Vous incarnerez lors de l'aventure SCP-014-B un jeune adulte se retrouvant dans le complexe ZX_67 qui se situe en [REDACTED] près du lac de [REDACTED]. Pour des raisons évidentes certains passages de ce rapport seront censurés par la fondation, cette dernière ne voulant pas dévoiler au grand public certaines informations car la fondation Scp a pour mission de Sécuriser Contenir et Protéger des Entités qui peuvent aller de moi (grille pain dont on peut se référer seulement à la 1ère personne) à une entité pouvant rayer votre existence de la réalité, ce Scp est [REDACTED]

[REDACTED], [REDACTED]. J'espère qu'en lisant ce rapport vous ne vous soyez pas enfuis . Lors de ce jeu relatant les faits datant du [REDACTED] à [REDACTED]

vous devrez sécuriser les Scp et réparer les systèmes de survie du complexe pour ceci merida une superviseure des complexes de [REDACTED] vous aidera. Le jeu auquel vous allez jouer relate des faits qui se sont réellement déroulés , les quatres scp se sont échappés de leur salle de confinement vous arriverez par l'ascenseur principal, de la vous aurez accès à l'ensemble du bâtiment mais attention car les scp s'étant échappés ils essayeront de vous [REDACTED] donc de vous tuer. Ne vous inquiétez pas merida sera la pour vous dire tout ce qu'il faudra faire pour mettre les scp dans leurs cellules de confinement , pour éviter les scp et de toute manière votre

personnage est immortel . vous devrez mettre tout en oeuvre pour remettre les scp en confinement et réparer les systèmes de sécurité pour ce faire il y aura 4 salles qui vous serviront à "crafter" et réparer les objets nécessaire à votre réussite.

1.D Plan du jeu



1.E Scénario détaillé

Le joueur sera placé en début de partie dans l'ascenseur une salle qui ne lui servira que pour 3 choses arriver dans le jeu se cacher et terminer le jeu , a son arrivée dans la salle de contrôle il est briefé pour savoir quel est son objectif principal (détaillé dans le scénario résumé) ensuite il devra aller directement dans la réserve récupérer son équipement de base détaillé dans les objets, ensuite il devra sécuriser le premier scp, scp-049 le médecin de la peste je vous laisse chercher sur internet toute la documentation sur les scp y sont présente. Pour le sécuriser il devra répondre à trois simple questions à choix multiples(c'est le premier monstre donc autant le rendre simple) si les réponses ne conviennent pas au scp paf vous décédez , dans le jeu bien évidemment et reprendrez au dernier checkpoint comme pour chaque monstre , et comme on peut se cacher dans l'ascenseur vous comprendrez que l'un des scp c'est enfuit et c'est suite à la réussite du confinement du premier monstre que vous devrez vous cachez toute les 2 min dans l'ascenseur pour éviter le scp quand il est dans les couloirs , ce timer sera indiqué dans une salle. le joueur devra crafter certains objet pour permettre le

confinement des scp avec les instructions de merida pour lui donner des indices .

I.F) Détail des lieux, items, personnages

Les objets seront craftables dans le laboratoire et la réserve. La salle du courant elle, servira pour rétablir les fonctions du complexes qui ont été désactivées , la salle de contrôle elle servira pour le timer du monstre rodant dans les couloirs , chaque salle de scp sera utile même suite au re-confinement de ces derniers car certains objets sont présents dans ces salles .

Le sas lui jouera un rôle dans le re-confinement du scp le plus dangereux du complexe.

Liste des item utilisables (qui servent à avancer dans le jeu):

- lampe torche (perd de la batterie puis s'éteint après 4 min pour le recharger il faut aller dans la réserve)
- *fusée éclairante
- détecteur de chaleur
- carte B
- carte Z
- fusible (*1,2,*3,4)
- *Radar
- *C4
- *couverture de dissimulation
- *Chalumeau
- *module de piratage
- Tournevis

*:item créable , réparable ou rechargeable

Liste des item servant à créer, construire ou recharger:

- Schéma Module de piratage
- Schéma C4
- Schéma couverture de dissimulation
- Schéma Fusible (utilisable pour créer 1 fusible)
- *Câbles
- *bouteille d'eau
- *Césium
- *Flacon d'éthanol
- *Circuit Imprimé

*:item servant d'élément de base à la création

I.G) Situations gagnantes et perdantes (voulues)

On perd le jeu si l'on :

- ne surveille pas ont timer du monstre errant (si l'on c'est pas caché à la fin du timer).
- si l'on échoue aux épreuves des scp (ex: les questions du 1er scp).
- si l'on repart dans l'ascenseur

On gagne si l'ont (il faut repartir dans l'ascenseur puis partir pour valider la victoire):

- réussit au moins le re-confinement du premier scp, mais ont est quand même considéré comme lâche ...
- si l'on re-confine au moins 2 scp vous avez votre job.
- si l'on re-confine au moins 3 scp vous êtes quelqu'un de bien.
- si l'on re-confine les 4 scp vous êtes un héros.

I.H) énigmes, mini-jeux, combats, etc.(voulues)

Chaque craft d'objet est une mini énigme simple par exemple pour crafter une lampe , on aura merida qui nous dira: "que la lumière soit " quand nous aurons récupéré tous les objets permettant la création de la lampe ici des piles et une lampe torche (simple mais les craft se compliquent selon l'avancement du jeu) .

Chaque combat avec les scp sera spécial et désigné selon le scp.

I.I) Commentaires (ce qui manque, reste à faire, ...)

j'ai été confronté à beaucoup de soucis dans la création de mon jeu à partir des commandes take et drop (quand l'on a créé l'inventaire) ce qui m'a empêché de créer mon magic cookie car sans take pas d'objet dans l'inventaire .

Pour ce qui est de la quête principale, du système de craft , et du timer (temporel) ils sont inexistant (on ne va pas se le cacher) plutôt par manque de temps que de volonté.

Les commandes take et drop ne marchant pas pour mon zuul j'ai essayé à de nombreuses reprises de changer totalement mon code et ma manière de faire les faire mais sans succès c'est pourquoi je les ai laissés en commentaire. Et en essayant de créer une situation gagnante pour mon jeu j'ai essayé de créer une énigme simple comme dit précédemment mais sans succès la aussi.

La classe door étant aussi inutile je l'ai laissé au même titre que toutes les portions de code en commentaire tel quels pour montrer que j'ai essayé de continuer mon jeu malgré mon échec avec le take.

II. Réponses aux exercices (à partir de l'exercice 7.5 inclus)

7.5 La procédure `printLocationInfo()` est créée dans `game` et remplace `createRoom()` et `goRoom()` qui faisait de la duplication de code.

7.6 L'accesseur `Get Exit()` est pour réduire le couplage ajouté à la classe `Room`.

7.7 la méthode `getExitString()` est créé ans `Room`

7.8 Les hashmap sont créés pour simplifier le code

7.8.1 ajout du déplacement horizontal

7.9 la méthode `KeySet()` permet d'associer une valeur arbitraire a la clé

7.10 La méthode `getExitString()` renvoie toutes les sorties de la `current room`.

7.11 `getLongDescription()` est ajoutée dans la classe `Room`.

7.14 nouvelles commande `look()` ajoutées a `game`

7.15 nouvelles commande `eat()` ajoutées a `game`

7.16 la procédure `showAll()` est incluse dans la classe `CommandWords`

7.18 la méthode `showAll()` présente dans la classe `CommandWords()` devient `get CommandList()` et est Différente

7.18.1 Modification/comparaison du jeu en fonction de `zuul-better`.

7.18.5 On utilise `put()`.

7.18.6 Modification/comparaison du jeu en fonction de `zuul-with-images`.

7.18.8 Dans `UserInterface`, on crée notre bouton grâce à `createGUI()`. On crée un deuxième panel pour les boutons. Puis dans `actionPerformed()`, on indique les cas où les boutons sont appuyés.

7.20 Création d'une classe nommée Item avec pour attributs son poids et sa description. Puis on crée des items dans `creation()` qu'on associera à des pièces.

7.21 Dans `getLongDescription()`, nous rajoutons une string qui affiche nos Items.

7.22 Nous utilisons une `HashMap` pour Items maintenant pour qu'elle puisse contenir autant d'Items que l'on veut. `setItem()` devient un `addItem`, où on utilise un `put`.

7.23 On rajoute un attribut `aPreviousRoom`, dans `creation()` elle va être stockée la pièce on était avant de se déplacer.

7.24 Le `back()` fonctionne même avec un deuxième mot/commande.

7.25 Le `back()` ne fonctionne qu'une seule fois.

7.26 Je me sers des fonctions de `Stack` dans `GameEngine`. dans `back()` on teste si la pièce précédente est vide si elle l'est, sinon on utilise `pop()` qui remplace la pièce courante par la précédente. Et dans `goRoom()`, à chaque fois que le joueur va dans une pièce on définit la pièce courante comme la précédente grâce à un `push()`.

7.27 Tester l'ouverture correcte du jeu ne suffit pas puisque certaines erreurs sont détectées que lorsque la commande associée est tapée.

Exécuter une suite de commandes permettrait de s'assurer du bon fonctionnement du jeu.

7.28 On pourrait mettre cette série de commandes dans un fichier qui sera un scanner, une méthode test qui permettrait de lire le fichier et ainsi exécuter les commandes inscrites.

7.28.1 On crée donc cette méthode test, avec des blocs `try` et `catch`. Je mets dans `try` la lecture du fichier, si jamais il y a une erreur `catch` va venir l'attraper.

7.29 On crée une classe `Player` qui a pour attributs la pièce courante, le poids que porte le joueur et le poids qu'il peut porter

au maximum, le nom du joueur, et un inventaire. On utilise donc plus la pièce courante dans gameEngine directement, mais on l'utilise indirectement grâce au nouvel attribut aPlayer.

7.30 On crée une méthode take() qui permettra de prendre un item et une autre drop() qui permettra de lâcher un item.

7.31 On crée un attribut type HashMap dans Player qui sera l'inventaire où l'on va mettre tous les items que le joueur va porter

7.32 Ont créé un attribut de type double pour paramétrer un poids max supportable par le joueur, le poids max sera pris en compte lors de la création de l'inventaire

7.33 ont créé une hashmap correspondant à l'inventaire de notre joueur le joueur ayant une capacité max l'on va modifier la méthode take pour qu'elle obéisse aux contrainte de poids max , ont fait donc en sorte qu'on ne puisse pas prendre plus que notre force (powa: power).

7.34 non fait a cause du take ne marchant pas suite à l'exercice précédent .

7.42 tentative de création d'une Time limit: on crée une nouvelle classe permettant de créer une donnée sur l'interface ont créé donc des attributs et l'ont fait en sorte d'avoir un timer temporel donc ont utilise le run pour faire une boucle et perdre a la fin c'est à dire quand arrivé à 0.

7.43 tentative de création d'une trap door: on crée une nouvelle classe permettant de créer les portes situés entre chaque salle une porte étant un piège se refermant sur le joueur ont créé un attribut pour fermer l'accès à une autre room.

7.44 le beamer nécessite que l'on crée une classe pour lui , c'est un item à part entière car il permet de performer une téléportation dans le jeu donc lui donne la possibilité de nous faire changer de room après avoir été chargé.

7.46 on crée une classe transporterRoom et roomRandomizer cette dernière sélectionnera une salle aléatoire et le premier désignera une première salle depuis laquelle on se téléporte aléatoirement les deux classes sont donc liées au tableau des différentes salles.

III. Mode d'emploi (si nécessaire, instructions d'installation ou pour démarrer le jeu)

pas d'instruction particulière .

IV. Déclaration obligatoire anti-plagiat

je me suis inspiré du travail de Davina Luximon pour les derniers exercices et je ne vais pas vous mentir j'ai demandé beaucoup de fois de l'aide à mes camarades pour ce fameux take mais sans succès personne n'a réussi à m'aider.