

## Rapport

Titre : [pas encore de titre]

I)

A)

Auteur : Jolan Roustant E1Grp10

B)

Ma phrase thème est : « Un soldat dans les tranchées lors de la première guerre mondial doit transmettre un message ».

C)

Résumé du scénario :

Un soldat français se réveille dans les tranchées durant la première guerre mondiale, un camarade lui annonce que son supérieur à quelque chose d'important à lui dire. Il découvre qu'il doit transmettre à temps un message de la plus haute importance à un autre bataillon, en effet ce bataillon va se faire attaquer, il doit donc prévenir ce bataillon à temps. On doit donc suivre les indications des soldats sur notre chemin et éviter les pièges (mines, éboulements...) afin de trouver ce bataillon.

D)



Plan pas fini (il manque des objets et des personnages), les lieux sont en jaune, les personnages en bleu et les objets en rouge.

E)

À son réveil, un camarade lui dit de le suivre pour entendre ce que son supérieur a à dire. Le supérieur donne au soldat une lettre à remettre pour sauver un bataillon en danger. Le soldat doit alors choisir d'accepter ou de refuser la mission. S'il accepte, il doit se rendre à l'endroit indiqué par le supérieur mais se heurte à un glissement de terrain qui bloque le chemin. Il doit trouver une pelle pour dégager le chemin. Plus tard il va rencontrer un aviateur qui est prêt à l'aider mais dont l'avion est endommagé et doit être réparé. Le joueur doit alors l'aider à réparer l'avion. Pendant le vol il peut choisir de parler de sa mission ce qui aura comme conséquence de faire échouer la mission (mais le joueur ne le sais qu'à la fin). Arrivé sur la terre ferme il va continuer son chemin en suivant les indications des soldats, mais s'il se trompe il peut tomber sur des mines. Enfin après avoir beaucoup marché il va arriver au bataillon allié, s'il s'est confié au pilote, le bataillon a malheureusement subi une attaque car le pilote était un traître. Dans le cas contraire, il réussit sa mission, sauve le bataillon, qui va se préparer à lutter et est considéré comme un héros.

Il y a le scénario sous la forme d'un fichier Twine.

F)

G)

Situation gagnante :

- Arrivé au bataillon sans avoir parlé de sa mission au pilote.

Situation perdante :

- Arriver au bataillon en ayant parlé de sa mission au pilote.
- Tomber sur des mines.
- Refuser la mission
- Revenir au QG avant d'avoir donné la lettre car on est considéré comme un traître.

H)

I)

II)

7.5)

L'intérieur de la fonction `printLocationInfo` ressemble à ça :

```

        this.aCurrentRoom = this.aRoom;
        System.out.println("You are " + this.aCurrentRoom.getDescription());
        System.out.print("Exits: ");
        if(this.aCurrentRoom.aNorthExit!=null)
        {
            System.out.print("North ");
        }if(this.aCurrentRoom.aEastExit!=null)
        {
            System.out.print("East ");
        }if(this.aCurrentRoom.aSouthExit!=null)
        {
            System.out.print("South ");
        }if(this.aCurrentRoom.aWestExit!=null)
        {
            System.out.print("West ");
        }
        System.out.println(" ");
    }
}

```

On va l'appeler dans les méthodes printWelcome et goRoom de la classe game.

7.6)

L'accesseur ressemble à ça avec "pDirection" au lieu de "direction", "aNorthExit" au lieu de "northExit" (pareil pour les autres attributs). (Cette partie du rapport a été faite après avoir modifier le code):

```

// méthodes existantes non modifiées

public Room getExit(String direction)
{
    if(direction.equals("nord")) {
        return northExit;
    }
    if(direction.equals("est")) {
        return eastExit;
    }
    if(direction.equals("sud")) {
        return southExit;
    }
    if(direction.equals("ouest")) {
        return westExit;
    }
    return null;
}

```

On appelle cette fonction dans goRoom()

7.7)

Le code ressemble à ça et on appelle cette fonction dans printLocationInfo() :

```

public String getExitString()
{
    String vExitsList = "Les sorties visibles ici sont : ";
    if(this.getExit("nord")!=null)
    {
        vExitsList += "nord ";
    }
    if(this.getExit("est")!=null)
    {
        vExitsList += "est ";
    }
    if(this.getExit("sud")!=null)
    {
        vExitsList += "sud ";
    }
    if(this.getExit("ouest")!=null)
    {
        vExitsList += "ouest ";
    }
    return vExitsList;
}

```

Comprenez-vous pourquoi il est logique de demander à Room de produire les informations sur ses sorties (et ne pas lui demander de les afficher), et pourquoi il est logique de demander à Game d'afficher ces informations (et ne pas lui demander de les produire) ?

- C'est logique car Room possède les attributs sur les sorties alors que grâce à l'accesseur, Game peut les afficher mais pas les modifier donc c'est pour ça qu'elle ne les produits pas mais les affiche justes.

7.11)

Voilà la nouvelle fonction.

```

public String getLongDescription()
{
    return this.aDescription + "\n" + this.getExitString();
}

```

On peut donc modifier printLocationInfo() pour juste afficher ce que nous retourne cette nouvelle fonction.

```

1 import java.util.HashMap;
2 /**
3  * Classe Room - un lieu du jeu d'aventure Zuul.
4  *
5  * @author ROUSTANT Jolan
6  */
7 public class Room
8 {
9     /// ATRIBUTS ///
10    public String aDescription;
11    private HashMap<String, Room> exits;
12
13    /// CONSTRUCTEUR ///
14    /**
15     CONSTRUCTEUR DE LA CLASSE Room
16     */
17    public Room(final String pDescriptionLieu)
18    {
19        this.aDescription = pDescriptionLieu;
20        exits = new HashMap<String, Room>();
21    }
22
23    /// ACCESSEUR ///
24    public String getDescription(){ return this.aDescription; }
25
26    /// MODIFICATEUR ///
27    public void setExits(final String pDirection, final Room pNeighbor)
28    {
29        this.exits.put(pDirection, pNeighbor);
30    }

```

Voilà les modifications pour la classe Room, on a plus que 2 attributs car on utilise une hashmap.

```

// vsalle.setExits(nord, est, sud, ouest);
vDodo0.setExits("est", vT1);
vT1.setExits("est", vT2);
vT1.setExits("sud", vCDS1);
vT1.setExits("ouest", vDodo0);
vT2.setExits("est", vT3);
vT2.setExits("ouest", vT1);
vT3.setExits("nord", vCDS2);
vT3.setExits("est", vT4);
vT3.setExits("sud", vQG);
vT3.setExits("ouest", vT2);
vT4.setExits("est", vT5);

```

Dans la classe room, on ajoute les sorties comme montré dessus.