

Réalisation d'application(2020)

iteration 4

Groupe 2

24 AVRIL 2020

1 7.42 TIME OUT

On appelle la fonction dans la classe GameView dans la méthode update on code la méthode dans GameModel. Au lieu d'utiliser le temps on compte le nombre de rooms

```
public void timeOut(){
    if(pastRooms.size()==20) {
        gameView.show("Time ouuuuuuuutt\n");
        interpretCommandString("quit");
    }
}
```

2 7.43.45 looked door/trap door

ayant fait un fichier csv pour la description des rooms on a juste rajouté une méthode pour dire c'est quoi l'état de la porte en lisant le fichier csv, état 1 pour trap door, 2 pour looked room 0 si y'a rien

```
if (currentRoom.getStateExit(nextRoom) == 0 || currentRoom.getStateExit(nextRoom) == 1) {
    goRoom(nextRoom);
    if (beam1()) {
        cpt++;
        if (cpt == 1) {
            checkpoint = nextRoom;
            gameView.show("beamer charged you can use it in the next room");
        } else if (cpt >= 2) {
            gameView.show("beamer can be used\n");
            used = true;
        }
    }
}

else {
    if (!key()) {
        gameView.show("\nloocked rooom right here find a key to open it\n");
    } else {
        goRoom(nextRoom);
    }
}
```

```

        if(pastRoom.getStateExit(currentRoom)==0) {
            goBack(pastRoom);
            if(beam1()){
                cpt++;
                if(cpt==1) {
                    checkpoint=pastRoom;
                    gameView.show("beamer charged you can use it in the next room");
                }else if(cpt>=2){
                    gameView.show("beamer can be used\n");
                    used=true;
                }
            }
        }
    }
    else {
        if(!key()) {
            gameView.show("\nyou don't have a key to back to this room use look to
        }else{
            gameView.show("\nyou opened the door \n");
            goBack(pastRoom);
        }
    }
}

```

3 7.44 beamer

pour cette fonctionnalité on a utiliser a compteur qui reviens a zero quand on utilise le beamer,pour utiliser le beamer faut le prendre comme un item en utilisant look et ensuite quand il est prêt a être utiliser faut taper la commande beam

```

private boolean beam1(){
    for(int i=0;i<p1.getItems().size();i++){
        if(p1.getItems().get(i).getName().equals("beamer")){
            return true;
        }
    }
    return false;
}

```

case BEAM:

```

        if(used) {
            goBack(checkpoint);
            gameView.show("beamer used\n");
            used=false;
            cpt=0;
        }
        else
            gameView.show("beamer uncharged\n");
        break;

```

4 7.46 Transporter room

on a créé deux classe `TrasporterRoom` et `RoomRandomizer` ; `TrasporterRoom` hérite de `Room` comme stipulé dans le chapitre 9 et utilise `RoomRandomizer` qui utilise elle même la classe `Random` .

```
import java.util.HashMap;
import java.util.Random;
public class TrasporterRoom extends Room
{
    private RoomRandomizer randomRoom;
    private HashMap <String, Room> myRooms;

    public TrasporterRoom (final String description, final HashMap <String, Room> Rooms)
    {
        super(description);
        this.myRooms = Rooms;
    }

    @Override
    public Room getExit (final String direction)
    {
        randomRoom = new RoomRandomizer(myRooms);
        return randomRoom.randomizeRooms();
    }
}
```

```
import java.util.HashMap;
import java.util.Random;
public class RoomRandomizer
{
    private HashMap <String, Room> myRooms;
    private Random randomize;

    public RoomRandomizer (final HashMap <String, Room> myRooms)
    {
        this.myRooms = myRooms;
        this.randomize = new Random();
    }

    public Room randomizeRooms()
    {
        Room[] roomTab = myRooms.values().toArray (new Room[0]);
        return roomTab[randomize.nextInt (myRooms.size())];
    }
}
```

```
}  
}
```

5 7.47.0 abstract Command