



2019-2020

# SANND : LA PLANETE DESERTIQUE

INITIATION A LA PROGRAMMATION OBJET

PITOT PIERRE-YVES

ESIEE PARIS

## Table des matières

I.	Partie 1 .....	3
A.	Auteur .....	3
B.	Thème (phrase-thème validée) .....	3
C.	Résumé du scénario (complet) .....	3
D.	Plan (complet, avec indication de la partie "réduit" si exercice 7.3.3) .....	3
E.	Scénario détaillé (complet, avec indication de la partie "réduit" si exercice 7.3.3) .....	5
F.	Détail des lieux, items, personnages .....	5
G.	Situations gagnantes et perdantes .....	6
H.	Eventuellement énigmes, mini-jeux, combats, etc. ....	6
I.	Commentaires (ce qui manque, reste à faire, ...) .....	6
II.	Réponses aux exercices .....	7
III.	Mode d'emploi (si nécessaire, instructions d'installation ou pour démarrer le jeu) .....	15
IV.	Déclaration obligatoire anti-plagiat .....	16

# I. Partie 1

## 1. Auteur

PITOT Pierre-Yves

## 2. Thème (phrase-thème validée)

Après s'être écrasée sur une planète désertique, une jeune pilote cherche à rentrer sur Terre.

## 3. Résumé du scénario (complet)

Après avoir récupéré ses affaires sur le vaisseau, [NOM] elle repère une ville proche sur la carte. Elle pourra effectuer des missions pour prendre :

- Vaisseau de contrebande appartenant à [NOM\_CRIMINEL]
- Vaisseau de transport appartenant à l'Union

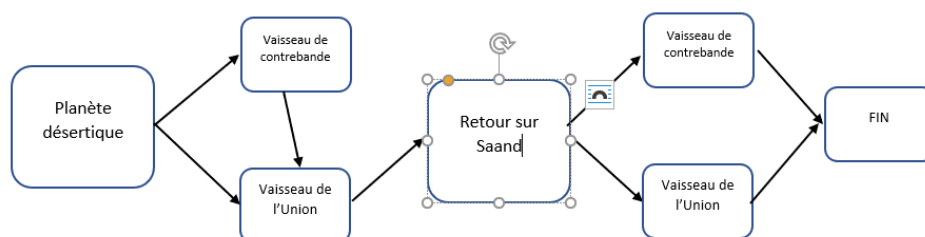
Mais le vaisseau est capturé par les rebelles, et [NOM] est emprisonnée. Elle doit donc faire un choix : rejoindre les rebelles, ou s'échapper pour rejoindre l'union.

Dans les deux cas, elle est envoyée fouiller des ruines pour récupérer un objet permettant de gagner la guerre.

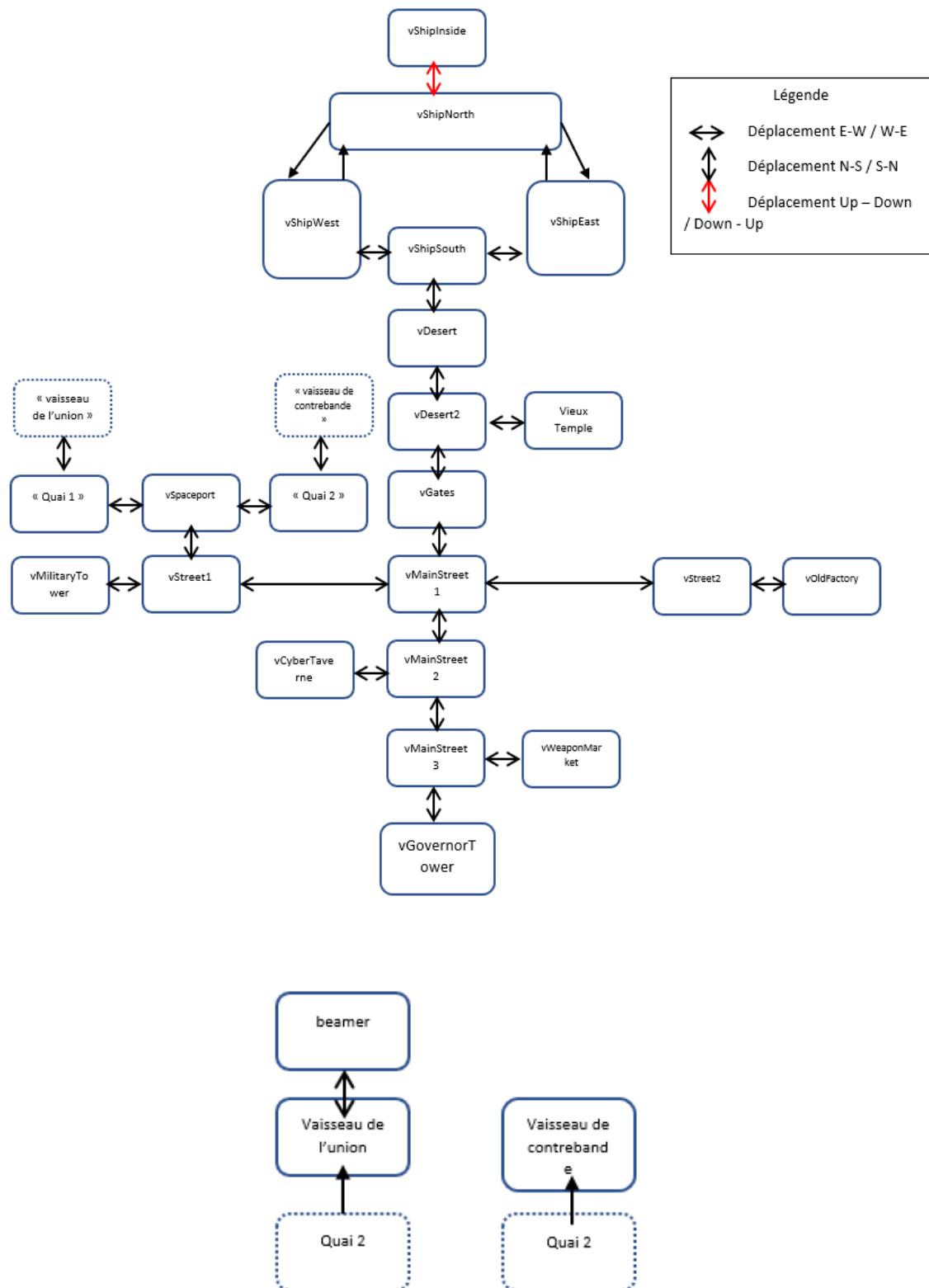
Une fois cet objet en sa possession, elle doit rejoindre une bataille qui a lieu au-dessus de la Terre. Elle a alors le choix entre combattre avec ou contre les rebelles. Une fois la bataille gagnée, [NOM] rentre sur Terre victorieuse.

## 4. Plan (complet, avec indication de la partie "réduit" si exercice 7.3.3)

Plan du jeu :



## Plan du Chapitre 1 :



## 5. Scénario détaillé (complet, avec indication de la partie "réduit" si exercice 7.3.3)

### Chapitre 1

Après s'être écrasée, [NOM] voit de la fumée au nord. Elle y découvre son vaisseau. De nombreux débris jonchent le sol.

En explorant le vaisseau, elle peut récupérer ses affaires et entrer dans le vaisseau. En parlant à l'ordinateur du vaisseau, seul le scanner fonctionne, et repère une ville proche au sud du vaisseau.

En traversant le désert, [NOM] tombe nez à nez avec un scorpion géant. Après avoir tué le scorpion, elle peut continuer.

En continuant sa route, elle voit une cité entourée de murailles. La porte est gardée par un garde. Il ne veut pas lui ouvrir car des rebelles se cacheraient dans la cité. Elle doit réussir à le convaincre de la laisser rentrer (« Je suis pilote de l'empire » ou « Je peux te payer »).

Le joueur entre donc dans la ville.

S'il entre dans la taverne, il pourra parler au barman, qui lui indiquera comment partir de cette planète.

- Elle peut se rendre au poste de l'Union. Le commandant en charge lui promet de l'emmener sur Terre si elle effectue un travail. Elle doit « détruire une base rebelle en ville ». Ils se cacheraient dans un entrepôt abandonné. Elle doit donc aller chercher des bombes normalement illégales au marchand d'arme, puis les déposer dans l'entrepôt. Dans l'entrepôt se cachent en réalité des femmes, des enfants et des malades rebelles. Elle a alors le choix entre faire exploser le bâtiment ou abandonner la quête (Sauf si la quête des rebelles a déjà été refusée). Après avoir tué les rebelles, elle doit se rendre au Quai 1 du spatioport.
- Elle peut se rendre au repère d'un criminel local. Celui-ci lui promet de la transporter sur un cargo de marchandise si elle assassine le gouverneur local. En entrant au palais, elle voit le gouverneur avec un commandant de l'Union. Après un combat, soit elle est blessée et capturée par l'Union, soit elle est victorieuse. Dans ce cas, elle peut se rendre au Quai 2.

Fin chapitre 1

## 6. Détail des lieux, items, personnages

Personnage principal :

[NOM], orpheline, n'a jamais eu d'autres rêve que de partir de sa ville natale, une pauvre petite ville, dans une région pauvre de la Terre. Elle s'engage dans l'armée spatiale de l'Union Galactique à 16 ans, et devient pilote à 18. Peu de temps après, l'armée lui confie sa première mission : détruire des vaisseaux rebelles au-dessus de la planète Saand. Cependant, son vaisseau est touché et s'écrase sur la planète. Elle doit alors trouver un moyen de rentrer sur sa planète.

## 7. Situations gagnantes et perdantes

Une fois que le joueur ramène l'artefact aux rebelles ou aux forces de l'Union, il rentre sur Terre et gagne. S'il meurt, il a perdu.

## 8. Eventuellement énigmes, mini-jeux, combats, etc.

- Combat au-dessus de la Terre (mini-jeu)
- Combats
- Enigmes pour trouver l'artefact

## 9. Commentaires (ce qui manque, reste à faire, ...)

## II. Réponses aux exercices

7.4)

Transfert des classes dans le dossier IPO\_Project.

Game :

- Modification de createRooms() :
  - Remplacement des pièces du Zuul-Bad par les 5 premières pièces du jeu.

7.5)

Game :

- Création de printLocationInfo().
- Remplacement du code récurent dans printWelcome().
- Remplacement du code récurent dans goRoom().

7.6)

Room :

- Création de getExit().
- Privatisation des attributs de Room.

Game :

- Modification de goRoom()
  - Remplacement des itérations de aCurrentRoom.a[DIRECTION]Exit par aCurrentRoom.getExit([DIRECTION])
- Modification de printLocationInfo()
  - Remplacement des itérations de aCurrentRoom.a[DIRECTION]Exit par aCurrentRoom.getExit([DIRECTION])

7.7)

Room :

- Création de getExitString()

Game :

- Modification de printLocationInfo()
  - Remplacement du contenu par  
System.out.println(this.aCurrentRoom.getExitString());

7.8)

Room :

- Renommage de setExits() en setExit()
- Création de HashMap<String, Room> aExits
- Modification de setExits().

Game :

- Modifications de createRooms() : fonctionne désormais avec la nouvelle méthode setExits().

Remarque :

- setExitString() ne fonctionne plus

7.8.1)

Room :

- Modification de setExitString().

7.11)

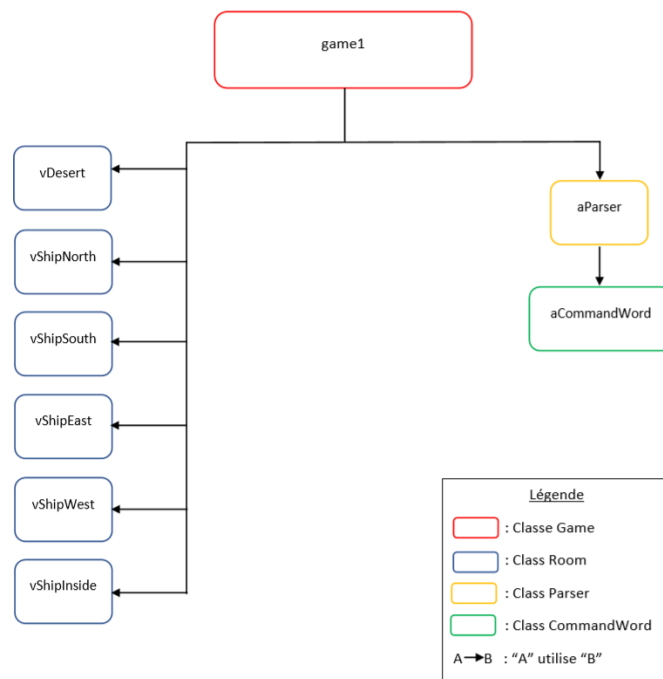
Room :

- Création de getLongDescription()

Game :

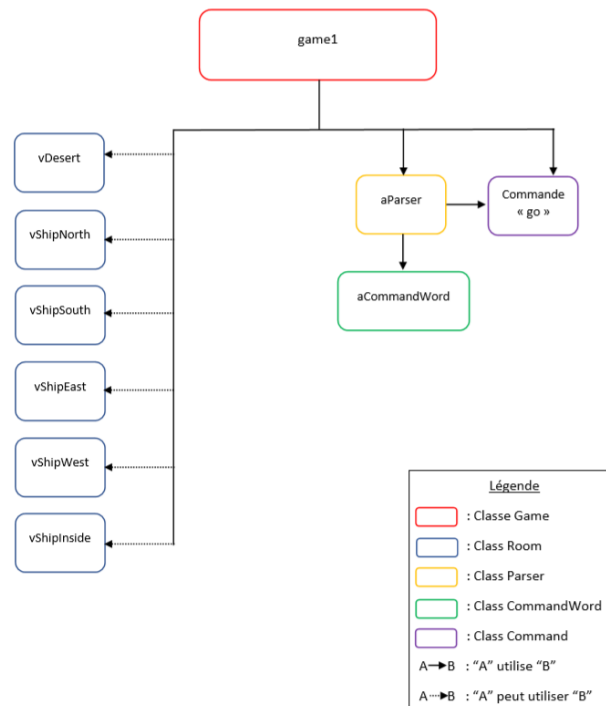
- Modification de printInfoLocation() : affiche désormais la description d'une Room et ses sorties.

7.12)



7.13)





7.14)

#### CommandWords :

Ajout d'une nouvelle commande (look) :

- Ajout de « look » dans aValidCommands.

#### Game :

- processCommand() : teste désormais si le second mot correspond à "look".
- Création de la méthode look() qui affiche la description de la pièce courante.

7.15)

#### CommandWords :

Ajout d'une nouvelle commande (eat) :

- Ajout de « eat » dans aValidCommands.

#### Game :

- processCommand() : teste désormais si le second mot correspond à "eat".
- Création de la méthode eat() qui affiche un String.

7.16)

#### CommandWord :

- Création de showAll() qui affiche les commandes

#### Parser :

- Création de showCommands() qui appelle showAll()

#### Game :

- Modification de help() : Affiche désormais toutes les commandes possibles

7.17)

If you now add another new command, do you still need to change the Game class ? Why ?

Il n'est plus nécessaire de modifier la classe Game car l'ajout d'une commande n'est plus lié à la classe Game.

7.18)

CommandWord :

- Renommage de showAll() en getCommandList(), qui retourne maintenant un String contenant les commandes.

Parser :

- Renommage de showAll() en showCommands(), qui retourne maintenant le résultat de getCommandList()

Game :

- Modification de printHelp() qui affiche le résultat de showCommands()

7.18.2)

Room :

- Modification de getExitString() : On utilise à présent un StringBuilder pour construire le String de sortie.

7.18.5)

Room :

- Ajout d'une Hash Map contenant les pièces.

7.18.6)

Comparaison avec Zuul-With-Images et implémentation des éléments manquants .

7.18.8)

UserInterface :

- Import de javax.swing.JButton
- Création des attributs aButton1 et aButton2
- Modification de createGUI() pour ajouter les boutons
- Modification de actionPerformed() pour lancer l'action

7.19.2)

Création du dossier « Images ».

GameEngine :

- createRooms() : Remplacement de "[NOM\_IMAGE]" par "Images/[NOM\_IMAGE]"

7.20)

Création de « Item »

Room :

- Ajout d'un attribut de type Item.
- Création de setItem()
- Création de getItemDescription()
- Modification de getLongDescription() pour afficher si il y a ou non un objet dans la pièce

7.21)

L'information doit être générée par la classe Item, car les informations nécessaires sont les attributs des objets Item. Cependant, c'est GameEngine qui doit l'afficher car c'est là le but de cette classe.

7.22)

Item :

- Ajout d'un attribut « aNom » et modification du constructeur
- Ajout de la méthode toString()
- Suppression de getLongDescription()

Room :

- Ajout d'une HashMap content les items.
- Modification de getItemDescription() pour que la méthode fonctionne avec plusieurs items.

7.22.1)

On utilise une HashMap car elle permet d'accéder à un objet et de le supprimer facilement.

7.22.2)

GameEngine :

- Ajout de nouveaux items

7.23)

Game :

- Ajout de la commande back dans interpretCommand()
- Ajout de l'attribut aLastRoom qui vaut « null » lors de son initialisation.
- Ajout de la méthode goTo() qui permet de se déplacer dans une salle.
- Modification de la commande goRoom pour remplacer le code redondant par goTo()
- Ajout de la methode back() qui appelle la methode goTo() si aLastRoom ne vaut pas null.

CommandWords :

- Ajout de « back » dans les commandes acceptées.

7.24)

On doit tester s'il y a une salle précédente. Si le joueur n'a pas bougé, la pièce précédente sera nulle.

7.25)

A l'heure actuelle, entre « back » trois fois revient à l'entrer une fois.

7.26)

GameEngine :

- Remplacement de aLastRoom par le Stack aPreviousRooms
- Initialisation de aPreviousRoom dans le constructeur
- createRooms() : Ajout de aCurrentRoom dans aPreviousRooms.
- Modification de back() pour fonctionner avec aPreviousRooms.

7.27 / 7.28)

Une commande qui permettrait de lire un fichier contenant des commandes serait un ajout important. Cela permettrait de facilement vérifier si toutes les fonctionnalités du programme fonctionnent. Pour cela, il faudrait créer une nouvelle commande permettant de lire un fichier.

7.28.1)

Création de la commande test.

GameEngine :

- Ajout de test() qui lit le fichier donné en paramètre
- Modification de interpretCommand() pour ajouter « test »

CommandWord :

- Ajout de test dans la liste des commandes valides

Création de test.txt contenant des commandes

7.28.2)

Création des deux fichiers de commande « court » et « long » et « win ».

Suppression de test.txt

7.29) Player

Création de la classe Player.

Player :

- Création de goTo()
- Création de goBack()
- Création de getCurrentRoomExit()
- Création de getCurrentRoom()
- Création de setCurrentRoom()
- Création de previousRoomIsEmpty()

- Création de pushRoom()

GameEngine :

- Déplacement d'une partie de look() dans Player
- Déplacement d'une partie de goRoom() dans Player
- Déplacement d'une partie de back() dans Player

7.30) Take / Drop

CommandWord

- Ajout des commandes dans la liste des commandes

GameEngine

- Modification de interpretCommand() pour implémenter take et drop
- Ajout de take()
- Ajout de drop()

Player :

- Ajout de take()
- Ajout de drop()
- Ajout de aInventory

Room

- Ajout de getItem()
- Ajout de removeItem()
- Ajout de hasItem()

7.31) porter plusieurs items

Fonctionnalité implémentée dans l'exercice 7.30.

7.31.1) ItemList

Création de ItemList

ItemList :

- Création de aList
- Ajout de removeItem()
- Ajout de hasItem()
- Ajout de addItem()
- Ajout de getItem()
- Ajout de values()
- Ajout de isEmpty()

Player :

- aInventory est maintenant une ItemList
- Modification des méthodes pour supporter cette modification

Room :

- altems est maintenant une ItemList
- Modification des méthodes pour supporter cette modification

7.32)

Player :

- Ajout de aCurrentWeight
- Ajout de aMaxWeight
- Modification de drop() et take() pour tester si l'objet peut être pris.

7.33)

Ajout de la commande « items » :

ItemList :

- Ajout de toString()

GameEngine :

- Ajout de items() qui affiche le contenu de l'inventaire.
- Modification de interpretCommand() pour ajouter « items ».

CommandWord :

- Ajout de « items » dans la liste des commandes valides

Player :

- Ajout de getMaxWeight()
- Ajout de getCurrentWeigth()
- Ajout de getInventory()

7.34)

GameEngine :

- Modification de eat : appelle aPlayer.eat()
- 

Player :

- Ajout de eat() : Teste si l'item est mangeable (est une instance de EatableItem) et appelle eat sur l'item si il l'est

Création de la classe abstraite EatableItem : Ajout de la méthode abstraite eat()

Création de MaxWeightIncreaserItem : augmente le poids max de l'inventaire quand est mangé

7.34.1 / 7.34.2)

Mise à jour des fichiers tests et de la javadoc.

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

Mode d'emploi non nécessaire pour le moment.

## IV. Déclaration obligatoire anti-plagiat

Toutes élément du jeu, excepté les fichiers zuul-\*.jar fournies dans un cadre éducatif, et les éléments suivants, ont été créés par PITIOT Pierre-Yves, élève à ESIEE Paris.

UserInterface : createGui() :

A été inspiré par : OpenClassrooms - Apprenez à programmer en Java -

<https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/23366-positionnez-des-boutons> [Consulté le 02/03/2020]

Images :

Uncoated – Importée le 21/02/2016 sur Pexels - <https://www.pexels.com/fr-fr/photo/desert-dune-dunes-dunes-de-sable-50628/>

Johndavis9538 - Importée le 24/11/ 2019 sur Pixabay -

<https://pixabay.com/fr/illustrations/%C3%A9trangers-vaisseau-spatial-univers-4648205/>

JAKO5D - - Importée le 07/05/2017 sur Pixabay - <https://pixabay.com/fr/illustrations/ufo-vaisseau-spatial-avion-2289563/>

Schmidsi - Importée le 26/07/2018 sur Pixabay - <https://pixabay.com/fr/illustrations/la-flamme-le-feu-transparent-graver-3561348/>

Schmidsi - Importée le 26/07/2018 sur Pixabay - <https://pixabay.com/fr/illustrations/la-flamme-le-feu-transparent-graver-3561354/>

FUMÉE NOIRE PNG – Importée sur PngImage - <https://pngimage.net/fumee-noire-png/> [Consulté le 17/03/2019]

Nuage de Fumée grise PNG transparent – Importée sur StickPng -

<https://www.stickpng.com/fr/img/nature/fumee/nuage-de-fumee-grise> [Consulté le 17/03/2019]

Mellon3D - Sci Fi corridor destroyed – Importé le 11/06/2011 sur DeviantArt -

<https://www.deviantart.com/mellon3d/art/Sci-Fi-corridor-destroyed-212671804> [Consulté le 17/03/2019]

Aernewstv - Quel est le poids d'un siège éjectable d'un avion de chasse ? -

<https://www.aernewstv.com/fr/lifestyle/a-votre-avis/3099-quel-est-le-poids-dun-siege-ejectable-dun-avion-de-chasse.html> [Consulté le 17/03/2019]

PlaceDesVacances – Désert du Sahara - <https://www.place-des-vacances.com/dossier-thematique-desert-du-sahara-59.html> [Consulté le 17/03/2019]

Pixabay – Importée le 23/01/2016 sur Pexels - <https://www.pexels.com/fr-fr/photo/aride-ciel-colline-desert-37544/> [Consulté le 31/03/2019]