2019-2020

SANND : LA PLANETE DESERTIQUE

INITIATION A LA PROGRAMMATION OBJET

PITIOT PIERRE-YVES
E1 - B3 - ESIEE PARIS

Table des matières

I. Partie 1

1. Informations principales :

Auteur : PITIOT Pierre-Yves
Date de la version finale du jeu : 19/05/2020
Date de la version finale du rapport : 19/05/2020

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

Résumé du scénario :

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

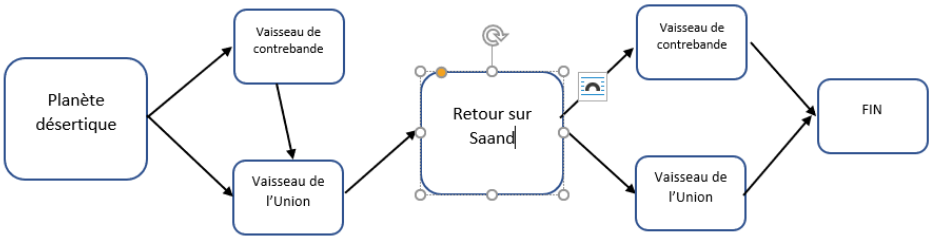
- Vaisseau cargo appartenant aux rebelles.
- Vaisseau apparentant à l'Union

Le vaisseau rebelle est capturé par l'Union. Dans les deux cas, les forces de l'Union vous demanderont de vous racheter en récupérant une arme dans une vieux temple, ce qui permet à l'Union ou aux rebelles, selon les choix de la pilote, de gagner la guerre.

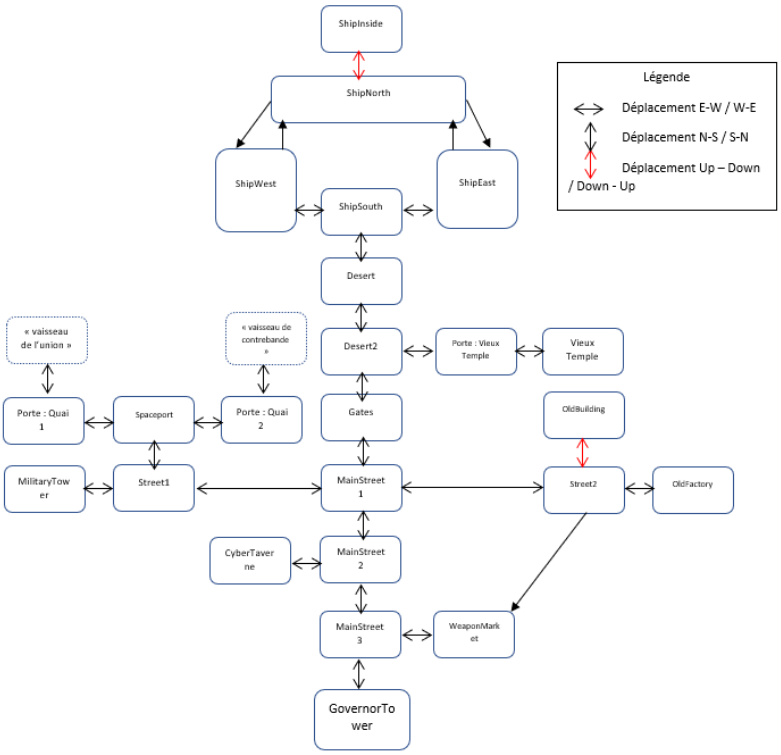
Une fois l'arme rapportée. La pilote est ramenée sur Terre.

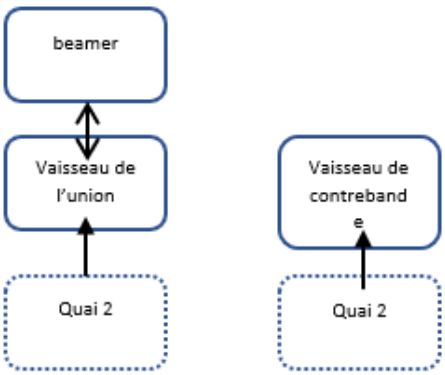
2. Plans

Plan du scénario :



Carte du jeu :





3. Détails des lieux, items, personnages

Lieux :

- Saand : Planète désertique proche de la Terre.
- Saand City : Ville principale de l'action du jeu. Des rebelles s'y cachent.
- Désert de Saand : Désert où commence le joueur.

Personnages importants :

- La jeune pilote : Jeune pilote des forces de l'Union. Elle a grandi sur Terre.
- Le Commandant : Chef des forces de l'Union sur Saand
- Ragnar : Chef des rebelles sur Saand

Items :

- Badge : Badge de l'union appartenant à la pilote.
- Bombe : engin explosif.
- Nanites : Nanorobots qui peuvent prendre n'importe quelle forme.
- Clé : ancienne clé permettant d'ouvrir le Temple.
- Le super cookie : permet d'augmenter la taille de l'inventaire.

4. Scénario détaillé

Après s'être écrasée, une jeune pilote voit de la fumée au Nord. Elle y découvre son vaisseau. De nombreux débits jonchent le sol.
En explorant le vaisseau, elle peut récupérer ses affaires dont son badge de pilote.
En traversant le désert, la pilote 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 avoir son badge pour passer
La pilote entre dans Saand City. Une citoyenne lui indique d'aller à la CyberTaverne.

Si elle entre dans la taverne, elle 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 une usine abandonnée. Elle doit donc aller chercher des bombes normalement illégales au marchant d'arme, puis les déposer dans l'entrepôt. Dans l'entrepôt se cachent en réalité des femmes, des enfants et des malades rebelles. Après avoir tué les rebelles, elle doit se rendre au Quai 1 du spatioport.
- Elle peut se rendre au repère d'un chef rebelle. Celui-ci lui promet de la transporter sur un cargo de marchandise si elle détruit la tour du gouverneur local
  Pour cela, elle doit utiliser la bombe dans la tour. Une fois la tour détruite, elle peut se rendre au Quai 2.

Si elle s'est rendue au vaisseau rebelle, elle peut parler avec le conducteur mais le vaisseau est pris par les forces de l'Union. Le Commandant lui demande de se racheter en allant chercher une arme cachée sur Saand.
Si elle s'est rendue au vaisseau de l'Union, elle peut directement parler au Commandant.

Après avoir ramassé la clé et pris le Téléporteur du vaisseau, elle doit aller au Vieux Temple dans le désert. Elle y récupère des Nanites. Des nanorobots capables de prendre n'importe quelle forme.

Elle peut ramener les Nanites aux rebelles ou à l'Union. Dans les deux cas, la partie est gagnée

5. 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. Le joueur perd si la commande GO est entrée plus de 100 fois.

II. Réponses aux exercices

7.4)

Transfert des classes dans le dossier IPO_Project.

Game :

- Modifiction 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)

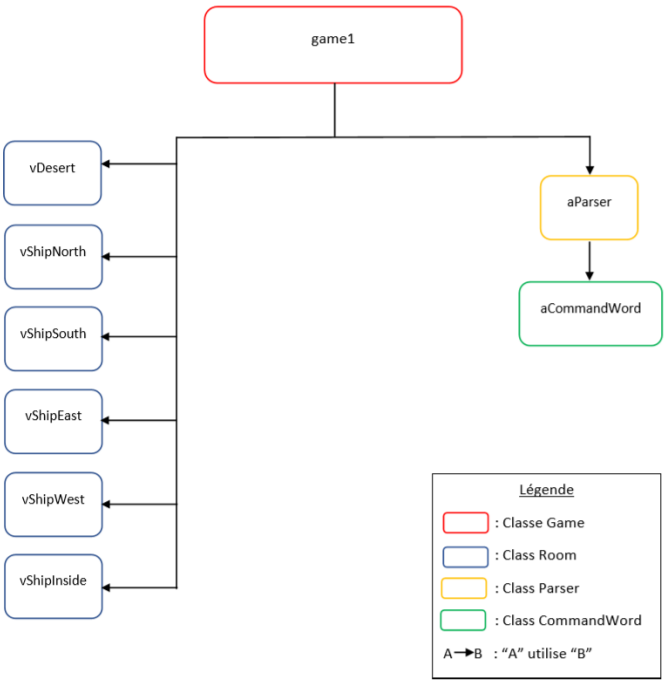
Room :

- Création de getLongDescription()

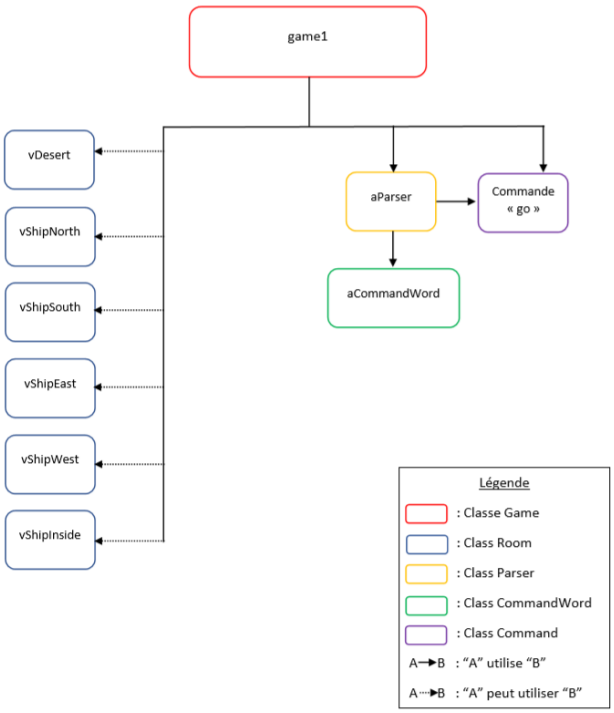
Game :

- 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 élements 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.

7.21.1)
Exercice fait après la Résa 9
Room :
- Ajout de getItems()

GameEngine :
- Modification de look() : Si il y a un second mot, appelle aPlayer.lookItem(), sinon appelle aPlayer.lookRoom()

Player :
- look() devient lookRoom() et renvoie la description de la Room
- Création de lookItem() et renvoie, si il existe dans l'inventaire ou la Room, la description de l'item

7.22)

Item :
- Ajout d'un attribut « aNom » et modification du constructeur
- Ajout de la méthode toSring()
- 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 interpertCommand()
- 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 permettra 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 ».
Supperssion 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 :
- aItems 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.

7.35)
Création de CommandWord

Command :
- Le premier mot est maintenant un CommandWord

CommandWords :
- aValidCommand est maintenant une HashMap<String,CommandWord>
- Ajout de getCommandWord
- Modification de isCommand() et de getCommandList() pour tenir compte des modifications

Parser :
- Modification de getCommand() pour supporter les modifications de Command

GameEngine :
- Modification de InterpretCommand() pour supporter les modifications ci-dessus

7.35.1)
GameEngine :
- Modification de InterpretCommand() : Le Else-if devient un switch

7.41.1)
Implémentation des modifications de zuul-with-enums-v2 :
CommandWord :
- Modification de l'énumération : de [COMMANDE] à [COMMANDE](« [commande] »)
- Ajout de aCommandString et d'un constructeur
- Ajout de toString()

CommandWords :
- Modification du constructeur : la liste de put devient un for

7.42)
GameEngine :
- Ajout de aTurnsLeft (valeur fixée à 100 au début du jeu)
- Décrémentation de aTurnsLeft dans goTo()

Attention : Exercice modifié lors de l'exercice 7.47 : aTurnsLeft est maintenant un attribut de Player, et la décrémentation a lieu dans le GoCommand. testTurnsLeft() et decreaseTurnsLeft() ont été ajoutés dans Player.

7.42.2)
Changements déjà réalisés :

7.43)
Room :

- Ajout de isExit() qui teste si la Room entrée est une sortie de la Room actuelle

Player :

- Modification de goTo(), qui vide le aPreviousRooms si la précédente CurrentRoom n'est pas une sortie de la nouvelle.

7.44)
Ajout de Beamer

- Ajout des methodes use() et load()

CommandWord :

- Ajout des commandes use et load

GameEngine :

- Ajout d'un item Beamer dans le WeaponMarket
- Ajout des commandes use() et load()

Player :

- Ajout des méthodes use() et load()
- Ajout de la méthode clearPreviousRooms()

7.45) Looked Door
Ajout de Door (hérite de Room)
Une Door est une pièce qui sera traversée par le joueur si il en possède la clé, ou si elle est ouverte.

- Ajout de setExit(), qui override Room.setExit(). Appelle Room.setExit() et stocke la pièce ajoutée dans une ArrayList.
- Ajout de isOpen() / setOpen()
- Ajout de getKey(), qui retourne la clé nécessaire à l'ouverture de la porte.
- Ajout de getNextRoom

GameEngine :

- Ajout de la Room vGatesFront, Gates devient la porte entre vGateFront et vMainStreet1
- Ajout de l'Item « badge », clé de vGates

Player :

- Ajout de getKey()
- Modification de goTo() pour qu'elle teste si la prochaine Room est une Door. Si oui, teste si elle est ouverte ou si le joueur possède la clé pour passer à travers.

7.46) Transporter Room

Ajout de TransporterRoom

- Création de getExit(), qui retourne appelle findRandomRoom() si la direction donnée est « beam ». Sinon appelle Room.getExit().
- Création de findRandomRoom(), qui appelle RoomRandomizer.getRoom()

Ajout de RoomRandomizer :

- Création de getRoom() qui renvoie une pièce aléatoire

Room :

- Ajout de isTransporterRoom()

GameEngine :

- Ajout de aRoomRandomizer
- Ajout des pièces vUnionShip et vTransporterRoom

- Modification de goTo() : Teste si la pièce est une TransporterRoom. Si oui, appelle getExit() dessus

7.46.1) Alea
GameEngine :
- Ajout de aTestMode
- Modification de test() : aTestMode = true au démarage du test et = false à la fin.
- Ajout de aAleaRoom

Ajout de la commande alea :
- Utilisable qu'en mode test
- Si n'a pas de second mot, aAleaRoom = null
- Si le second mot n'est pas une clé de aRooms, l'information est affichée.
- Sinon, la pièce est stockée dans aAleaRoom.

7.46.2) Héritage
Ces ajouts ont déjà été effectués dès la première approche des exercices.

7.46.3)
Ajout de commentaires Javadoc manquants.

7.47)
Ajout d'une nouvelle classe par commande. Migration des methodes « commandes » de Player et de GameEngine vers ces commandes

GameEngine :
- Remplacement du switch par vCommand.execute(this.aPlayer, this, this.aGui);
- Ajout de getTestMode()
- Ajout de getParser()
- Ajout de getAleaRoom()
- Ajout de getRooms()
- Ajout de setTestMode()
- Ajout de setAleaRoom()

Command :
- Devient une classe abstraite
- Suppression de aCommandWord, de son getter et de isUnknown()
- Ajout de setSecondWord()
- Ajout de la methode abstraite execute()

Parser :
- Ajout de la HashMap aCommands
- Modification de getCommand() pour accepter les nouvelles modifications.

7.47.1) Packages
Création des packages et migration des classes. Ecriture de tous les imports necessaires.

7.48) Characters
Créations des PNJ (classe Character)

Tout PNJ peut dire deux choses : une lors de la première rencontre et l'autre le reste du temps.

Room :

- Ajout de aPNJ, une Hashmap contenant les PNJ de la pièce.
- Ajout de méthodes pour gérer les PNJ, comme pour les Items.
- La liste des Pnj apparait lorsque l'on entre dans la pièce.

Ajout de la commande Talk qui déclenche le dialogue

7.49) MovingCharacters
Création de la classe MovingCharacter

Room :

- Ajout de getAleaExit(), qui renvoie une sortie aléatoire de la room
- Ajout de removePNJ()

GameEngine :

- Ajout de moveCharacters() qui bouge les MC.

GoCommand :

- Appel de la méthode moveCharacters()

7.49.1)
Toutes les sortes d'item héritent d'Item. Il en est de même pour les Characters (AbstractCharacter), les Rooms (Room), et les commandes (Command)

7.49.2)
Ajout du scénario dans le jeu

7.50 – 7.51)
La méthode qui renvoie le maximum de deux entiers est « max() »
Sa signature est « public static int max(int a, int b) »

Ces méthodes sont toujours en Static afin qu'elles puissent être appelées sans instancier une classe. De plus, ce serait inutile car elles n'ont besoin que d'informations extérieures pour fonctionner.
Instancier une classe serait possible mais inutile et couteux en ressources.

7.53 – 7.34)
Création de la méthode main dans Game.

III. Déclaration obligatoire anti-plagiat et sources des images

Toutes élément du jeu, excepté les fichiers zuul-*.jar fournis dans le cadre de notre formation et les éléments suivants, ont été créés par moi, PITIOT Pierre-Yves, élève à ESIEE Paris.

UserIterface : 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 17/03/2020]

Nuage de Fumée grise PNG transparent – Importée sur StickPng - https://www.stickpng.com/fr/img/nature/fumee/nuage-de-fumee-grise [Consulté le 17/03/2020]

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 17/03/2020]

Aeronewstv - Quel est le poids d'un siège éjectable d'un avion de chasse ? - https://www.aeronewstv.com/fr/lifestyle/a-votre-avis/3099-quel-est-le-poids-dun-siege-ejectable-dun-avion-de-chasse.html [Consulté le 17/03/2020]

PlaceDesVacances – Désert du Sahara - https://www.place-des-vacances.com/dossier-thematique-desert-du-sahara-59.html [Consulté le 17/03/2020]

Pixabay – Importée le 23/01/2016 sur Pexels - https://www.pexels.com/fr-fr/photo/aride-ciel-colline-desert-37544/ [Consulté le 31/03/2020]

ArtStation – Simon Verstraete - https://www.artstation.com/artwork/5NBeP [Consulté le 17/05/2020]
EuropaPark - https://www.europapark.de/fr/restauration/o-mackays-cafe-and-pub [Consulté le 17/05/2020]

Le Progrès – Johan BOZON - https://www.leprogres.fr/encadres/2016/09/20/le-pont-levis-du-chateau-en-quete-de-mecenes-sur-internet [Consulté le 17/05/2020]

TurboSquid – Mark Mons - https://www.turbosquid.com/3d-models/3d-sci-fi-street-1258640 [Consulté le 17/05/2020]

GoodFon – Kipish_fön - https://www.goodfon.com/wallpaper/damian-bonczyk-by-damian-bonczyk-the-hunt-sci-fi-environment.html [Consulté le 17/05/2020]

Pantone Canevas – Marc Mons - http://canvas.pantone.com/gallery/90561219/Scifi-Street-City [Consulté le 18/05/2020]

Renderosity – SIGMAWORLD - https://www.renderosity.com/mod/gallery/?image_id=2275410 [Consulté le 18/05/2020]

Pinterest – Bridge - Elijah McNeal - https://www.pinterest.fr/pin/517139969696850853/ [Consulté le 18/05/2020]

Polycount – Spaceship interior - Rutkovsky - https://polycount.com/discussion/170497/ue4-spaceship-interior [Consulté le 18/05/2020]

Paradox Interactive – Stellaris

Notre Loft - https://www.notreloft.com/bradford-lofts-tourcoing/ [Consulté le 18/05/2020]

MarketValue - http://www.marketvalue.fr/ [Consulté le 18/05/2020]

AncientHistory - https://etc.ancient.eu/interviews/petra-wonder-in-the-desert/ [Consulté le 18/05/2020]