2024

Franck Barland

[Nom de la société]

15/01/2024



JYST

Table des matières

[1/ Introduction, approche du projet 2](#_Toc158803654)

[1.1 Approche macro 2](#_Toc158803655)

[*1.2 Contexte du projet et limitation du remapping* 3](#_Toc158803656)

[*2/ Choix des outils du projet* 4](#_Toc158803657)

[2.1 Choix des outils de modélisation et de conception 4](#_Toc158803658)

[2.2 Choix de l’architecture 4](#_Toc158803659)

[*2.3 Choix du langage de programmation* 5](#_Toc158803660)

[*2.3.1 Langage de base* 5](#_Toc158803661)

[*2.3.2 frameworks* 5](#_Toc158803662)

[*2.3.3 Interface graphique utilisateur (GUI)* 5](#_Toc158803663)

[*3/ Gestion du projet* 6](#_Toc158803664)

[*3.1 Présentation* 6](#_Toc158803665)

[3.1.1 Domain-Driven Design 6](#_Toc158803666)

[3.1.2 UML 7](#_Toc158803667)

[3.1.3 MVC et SOLID 7](#_Toc158803668)

[3.1.4 SCRUM 7](#_Toc158803669)

[*3.1.5 Gantt ?* 7](#_Toc158803670)

[*3.1.6 Spécifications fonctionnelles et techniques* 8](#_Toc158803671)

[*3.2 Développement de l’application* 10](#_Toc158803672)

[*3.2.1 Java* 10](#_Toc158803673)

[*3.2.2 Front-End* 10](#_Toc158803674)

[*3.2.3 Back-end* 10](#_Toc158803675)

[*3.3 Tests* 10](#_Toc158803676)

[*4/ Conclusion* 11](#_Toc158803677)

[*4.1 Problèmes rencontrés et contournements* 11](#_Toc158803678)

[*5/ Divers* 12](#_Toc158803679)

[5.1/ Lexique 12](#_Toc158803680)

[5.2/ Références et supports extérieurs 13](#_Toc158803681)

[5.3/ Table des illustrations 15](#_Toc158803682)

# 1/ Introduction, approche du projet

* 1. Approche macro

<https://openclassrooms.com/fr/courses/5647281-appliquez-le-principe-du-domain-driven-design-a-votre-application>

DDD-> vue d’ensemble générale pouvant être comprise par n’importe qui… (joueus et développeurs)

*(Prévoir le « versionning »… ?)*

DDD

UML

## *1.2 Contexte du projet et limitation du remapping*

# *2/ Choix des outils du projet*

* Justification des choix de concepts d’approche pour la conception (justification des choix requis => DDD, UML… ) …

## 2.1 Choix des outils de modélisation et de conception

2.1.1 Domain-Driven Design (approche globale)

L’approche Domain-Driven Design (DDD\*), conception pilotée parle domaine, sera donc choisie. En effet, dans un premier temps, il s’agira de s’adresser à une population de personnes néophytes, hermétiques aux concepts purs du développement informatique.

*Pour qui suis-je en train de concevoir la solution ?*

*Que veulent-ils en faire ?*

* *Représentation conceptuelle des éléments clés*
* *Compris par tous, parties prenantes (clients) et développeurs*

*Eléments clés : Elément important*

[*https://openclassrooms.com/fr/courses/5647281-appliquez-le-principe-du-domain-driven-design-a-votre-application*](https://openclassrooms.com/fr/courses/5647281-appliquez-le-principe-du-domain-driven-design-a-votre-application)

2.1.2 U.M.L.

*Unified Modeling Language (U.M.L.)*

* Choix des diagrammes retenus

## 2.2 Choix de l’architecture

2.2.1 MVC et SOLID

[*https://openclassrooms.com/fr/courses/6810956-ecrivez-du-code-java-maintenable-avec-mvc-et-solid/6881176-tirez-le-maximum-de-ce-cours*](https://openclassrooms.com/fr/courses/6810956-ecrivez-du-code-java-maintenable-avec-mvc-et-solid/6881176-tirez-le-maximum-de-ce-cours)

*A développer….*

2.2.2 SCRUM

*SCRUM et non en « V » (waterfall) «  en cascade »*

SCRUM

<https://openclassrooms.com/fr/courses/4511226-gerez-votre-projet-avec-une-equipe-scrum>

*2.2.3 Gantt ?*

## *2.3 Choix du langage de programmation*

### *2.3.1 Langage de base*

### *2.3.2 frameworks*

* *Spring*

*Injection*

* *Struts*
* *Hibernate*

### *2.3.3 Interface graphique utilisateur (GUI)*

*Swing*

*Jpanel*

[*https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/23108-creez-votre-premiere-fenetre#/id/r-22932*](https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/23108-creez-votre-premiere-fenetre#/id/r-22932)

*AWT*

Abstract Window Toolkit

*2.4 Choix du langage et Framework de programmation*

* Java

<https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/20304-installez-les-outils-de-developpement>

# *3/ Gestion du projet*

## *3.1 Présentation*

### 3.1.1 Domain-Driven Design

L’approche Domain-Driven Design (DDD\*), conception pilotée parle domaine, sera donc choisie. En effet, dans un premier temps, il s’agira de s’adresser à une population de personnes néophytes, hermétiques aux concepts purs du développement informatique.

*Que veulent-ils en faire ?*

* *Représentation conceptuelle des éléments clés*
* *Compris par tous, parties prenantes (clients) et développeurs*

*Eléments clés : Elément important*

[*https://openclassrooms.com/fr/courses/5647281-appliquez-le-principe-du-domain-driven-design-a-votre-application*](https://openclassrooms.com/fr/courses/5647281-appliquez-le-principe-du-domain-driven-design-a-votre-application)

*Pour qui suis-je en train de concevoir la solution ?*

Pour des joueurs d’un Space Opéra (pilotage des vaisseaux de science-fiction) qui préfère joueur avec des joysticks, en évitant les allers retours clavier/joystick.

*Que veulent-ils en faire ?*

Aider à apprendre par cœur, les boutons affectés du joystick pour une meilleure réactivité pendant le jeu.

Intégration des besoins clients :

* Déportation d’un maximum de touches ‘clavier’ sur celles des joysticks
* Remapping : Multiplication des profils des joystick pour un maximum de touches affectées au clavier.
* Affichage sur second écran de l’affectation des fonctionnalités du jeu (keybinding) du jeu par rapport aux joysticks choisisdessinés

Limitation physique

Limitation ergonomique

Logiciels déjà existant

Vjoy

Joytokey

### 3.1.2 UML

*Unified Modeling Language (UML)*

* Choix des diarammes

### 3.1.3 MVC et SOLID

[*https://openclassrooms.com/fr/courses/6810956-ecrivez-du-code-java-maintenable-avec-mvc-et-solid/6881176-tirez-le-maximum-de-ce-cours*](https://openclassrooms.com/fr/courses/6810956-ecrivez-du-code-java-maintenable-avec-mvc-et-solid/6881176-tirez-le-maximum-de-ce-cours)

*A developer….*

### 3.1.4 SCRUM

*SCRUM et non en « V » (waterfall) «  en cascade »*

*Possibilité de plusieurs tâches en même temps*

SCRUM

<https://openclassrooms.com/fr/courses/4511226-gerez-votre-projet-avec-une-equipe-scrum>

### *3.1.5 Gantt ?*

### *3.1.6 Spécifications fonctionnelles et techniques*

*Ordre de priorisation des tâches de développement à faire*

## *3.2 Développement de l’application*

### *3.2.1 Java*

*awt*

*Interfaçage hardware / Java*

### *3.2.2 Front-End*

*Design system*

*Site map (application vue de haut, schéma cf projet Aicha)*

*Wire frame (sans les couleurs, maquettage grossier (/ergonomie)*

*Charte graphique (mood board , police, couleurs)*

*Maquette (Figma, Adobe XD….)*

*,*

### *3.2.3 Back-end*

*Extractions json*

## *3.3 Tests*

# *4/ Conclusion*

## *4.1 Problèmes rencontrés et contournements*

# *5/ Divers*

## 5.1/ Lexique

## 5.2/ Références et supports extérieurs

AWT

DDD

* <https://openclassrooms.com/fr/courses/5647281-appliquez-le-principe-du-domain-driven-design-a-votre-application>

Java

* <https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/20304-installez-les-outils-de-developpement>
* <https://koor.fr/Java/Index.wp>

Java joystick

* [*https://stackoverflow.com/questions/40391960/joystick-control-java*](https://stackoverflow.com/questions/40391960/joystick-control-java)
* [*https://docs.wpilib.org/fr/latest/docs/software/basic-programming/joystick.html*](https://docs.wpilib.org/fr/latest/docs/software/basic-programming/joystick.html)
* [*https://docs.wpilib.org/fr/latest/docs/software/commandbased/binding-commands-to-triggers.html#binding-commands-to-triggers*](https://docs.wpilib.org/fr/latest/docs/software/commandbased/binding-commands-to-triggers.html#binding-commands-to-triggers)
* [*https://blog.bozjatorium.com/2012/09/02/joystick-in-java-with-jinput/*](https://blog.bozjatorium.com/2012/09/02/joystick-in-java-with-jinput/)

Jira

* <https://openclassrooms.com/fr/courses/7365096-initiez-vous-au-test-et-a-la-qualite-logiciel/8233335-reportez-les-anomalies-aux-developpeurs-avec-jira>

MVC et SOLID

* [*https://openclassrooms.com/fr/courses/6810956-ecrivez-du-code-java-maintenable-avec-mvc-et-solid/6881176-tirez-le-maximum-de-ce-cours*](https://openclassrooms.com/fr/courses/6810956-ecrivez-du-code-java-maintenable-avec-mvc-et-solid/6881176-tirez-le-maximum-de-ce-cours)

Notion

* <https://www.notion.so/fr-fr>

Trello

* <https://openclassrooms.com/fr/courses/5870131-communiquez-en-utilisant-les-outils-numeriques/6160131-organisez-un-travail-collaboratif#/id/r-6167517>

SCRUM

* <https://openclassrooms.com/fr/courses/4511226-gerez-votre-projet-avec-une-equipe-scrum>

Swing (Jpanel)

* [*https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/23108-creez-votre-premiere-fenetre#/id/r-22932*](https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/23108-creez-votre-premiere-fenetre#/id/r-22932)
* <https://waytolearnx.com/2020/05/creation-interface-graphique-avec-swing-les-bases.html>

UML

* [*https://gitmind.com/fr/types-diagrammes-uml.html*](https://gitmind.com/fr/types-diagrammes-uml.html)
* [*https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml?a=1*](https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml?a=1)
* [*https://koor.fr/Other/SupportUML/slide1.wp*](https://koor.fr/Other/SupportUML/slide1.wp)

## 5.3/ Table des illustrations