Revue de code du Mars Rover

(à distribuer au bout d’une heure après avoir permuté les binômes)

(Cette feuille sera complétée de manière anonyme)

Jetez un petit coup d’œil rapide au code devant vous, puis répondez aux questions suivantes…

1. ***Premières impressions (M2104-M3105) :***

* Le code a-t-il été écrit ?

Anglais ❑ Français ❑ Franglais ❑

* Y a-t-il des tests qui couvrent le code ?

Non ❑ Oui ❑

Si oui : Combien de tests ont été écrits ? …………………….

Combien de tests passent AU VERT ? …………………….

Quelle est le % de couverture de code ? …………….

Si oui, pensez-vous que les tests ont été écrits :

❑ A priori (via un cycle de développement TDD)

❑ A posteriori (une fois le code de production écrit)

* Le programme est-il fonctionnel ?

Non ❑ Oui ❑

Si oui : Que peut-on faire avec ? ……………………………………….

……………………………………………………………………………………………….

………………………………………………….……………………………………………

* Le code a-t-il été versionné (git) ?

Non ❑ Oui ❑

Si oui : combien de commits ont été réalisés ? ………………….

* De combien de fichiers (classes), le code est-il composé ? : ……………………..
* Combien de lignes de code ont été écrites ? : ……………………..

… Explorez un petit plus précisément le code à revoir

en répondant aux questions suivantes ….

1. ***Zoom sur les règles de simplicité (Rules of simple Design (M2104))***

**2.1. Tous les tests passent**

* Si des tests ont été écrits, est-ce qu’ils passent TOUS AU VERT ?

Oui ❑ Non ❑

**2.2. Révéler l’intention du code**

* Retrouve-t-on la terminologie métier en lisant le code ?

Oui ❑ Non ❑ (Si non, expliquez : ……………………………………………………………..)

* Les classes sont-elles bien nommées ?

Oui ❑ Non ❑ (Si non, expliquez : ……………………………………………………………..)

* Les méthodes sont-elles bien nommées ?

Oui ❑ Non ❑ (Si non, expliquez : ……………………………………………………………..)

* Les variables sont-elles bien nommées ?

Oui ❑ Non ❑ (Si non, expliquez : ……………………………………………………………..)

* Le code est-il consistant ?

*Par exemple : si nommage des méthodes camelCase, le camelCase est utilisé partout, si le nommage utilise des underscores, underscore partout, mais pas de une fois camelCase, une fois underscore 😉*

Oui ❑ Non ❑ (Si non, expliquez : ……………………………………………………………..)

* Le code est-il correctement indenté/formaté ?

Oui ❑ Non ❑ (Si non, expliquez : ……………………………………………………………..)

**2.3. Pas de duplication: DRY (Don’t Repeat Yourself)**

* Y-a-t-il de la duplication dans le code ?

Oui ❑ Non ❑

Si oui, expliquez où se situe la duplication, ce qu’elle concerne dans le code :

…………………………………………………………………………………………………………………………………..

**2.4. Le plus petit nombre d’éléments (réduction de la complexité)**

* Combien de classes ? ……………………………………………………………..

Que pensez-vous de la taille des classes ? (nombre de méthodes par classe)

……………………………………………………………………………..…………………………………………………………….

……………………………………………………………………………..…………………………………………………………….

* Que pensez-vous de la taille des méthodes (nombre de lignes de code) ? ………………………………………………………………………………..…………………………………………………………….

……………………………………………………………………………..…………………………………………………………….

* Y-a-t-il des méthodes de plus de 20 lignes ?

Oui ❑ Non ❑

* Que pensez-vous de la complexité cyclomatique des méthodes (c-a-d nb de niveaux imbriqués)

………………………………………………………………………………..…………………………………………………………….

* Voyez-vous des méthodes avec une trop grande complexité cyclomatique (c-a-d des méthodes où des extract method auraient pu améliorer la lisibilité du code de ces méthodes)

Oui ❑ Non ❑

Si oui, indiquez lesquelles : …………………………..…………………………………………………………….

1. ***Zoom sur la simplicité YAGNI / KISS : Eviter l’over engineering 😉 (M2104)***

* Y-a-t-il des choses dans le code qui ont été implémentées mais qui ne sont pas utilisées ?

(comme un attribut taille pour le robot, des getteurs et setteurs que l’on n’utilise pas dans le code déjà écrit, etc …)

Oui ❑ Non ❑

Si oui, indiquez ce que vous pensez avoir détecté comme over engineering dans l’état actuel

du code : …………………………………………………………………………………………………………………………………..

*……………………………………………………………………………………………………………………………………………………………*

*……………………………………………………………………………………………………………………………………………………………*

*……………………………………………………………………………………………………………………………………………………………*

*……………………………………………………………………………………………………………………………………………………………*

1. ***Zoom sur les cas limites (M2104 – M3301 – M3105)***

* Les cas limites *techniques* ont-ils été implémentés ?

Oui ❑ Non ❑

*Débordement, etc …., déclenchement d’exception,… :*

Si non, expliquez : …………………………………………………………………………………………………………………………..

*……………………………………………………………………………………………………………………………………………………………*

*……………………………………………………………………………………………………………………………………………………………*

*……………………………………………………………………………………………………………………………………………………………*

*……………………………………………………………………………………………………………………………………………………………*

* Les cas limites *fonctionnels* ont-ils été traités ?

Oui ❑ Non ❑

*Et oui, la planète mars est ronde !!! Attention aux exceptions et bordure. Si on est en (0,0) vers l’ouest et on avance de 1, on doit se trouver sur une case de la grille (laquelle ?) et surtout ne pas avoir un null pointer !!! En effet, on ne peut pas se permettre de perdre notre robot sur Mars, juste parce que les développeurs ont négligé un pointeur null …*

Si non, expliquez : …………………………………………………………………………………………………………………………..

*……………………………………………………………………………………………………………………………………………………………*

*……………………………………………………………………………………………………………………………………………………………*

*……………………………………………………………………………………………………………………………………………………………*

* Voyez-vous un ou plusieurs cas limites qui n’ont pas été implémentés (et auraient dû l’être)

Oui ❑ Non ❑

Si oui, indiquez lesquels : …………………………..…………………………………………………………….

…………………………..………………………………………………………………………………………………………..

…………………………..………………………………………………………………………………………………………..

…………………………..………………………………………………………………………………………………………..

…………………………..………………………………………………………………………………………………………..

1. ***Zoom sur un logiciel opérationnel (minimum) (M3301 : MVP)***

* Le code écrit est-il opérationnel (utilisable, fonctionne-t-il) ?

Oui ❑ Non ❑

* La détection d’obstacle a-t-elle été implémentée ?

Oui ❑ Non ❑

1. ***Zoom sur les principes SOLID (M3105)***

**5.1. Principe SRP** (**S**ingle **R**esponsibility **P**rinciple : Principe de Responsabilité Unique)

* Le principe SRP vous parait-il toujours respecté ?

Oui ❑ Non ❑

Si non, indiquez le(s) endroit(s) dans le code où vous pensez que ce principe n’est pas respecté :

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..……..………………………………………………………………………………………………………………………………………………………..……..………………………………………………………………………………………………………………………………………………………..……..………………………………………………………………………………………………………………………………………………………..

**5.2. Principe OCP** (**O**pen **C**losed **P**rinciple : Principe Ouvert/Fermé)

* Le principe OCP vous parait-il toujours respecté ?

Oui ❑ Non ❑

Si non, indiquez le(s) endroit(s) dans le code où vous pensez que ce principe n’est pas respecté :

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..………………………………………………………………………………………………………………………………………………………..

**5.3. Principe LSP** (**L**iskov **S**ubstitution **P**rinciple : Principe de substitution de Liskov)

* Le principe LSP vous parait-il toujours respecté ?

Oui ❑ Non ❑

Si non, indiquez le(s) endroit(s) dans le code où vous pensez que ce principe n’est pas respecté :

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..………………………………………………………………………………………………………………………………………………………..

**5.4. Principe ISP** (**I**nterface **S**egregation **P**rinciple : Principe de ségrégation des interfaces)

* Le principe ISP vous parait-il toujours respecté ?

Oui ❑ Non ❑

Si non, indiquez le(s) endroit(s) dans le code où vous pensez que ce principe n’est pas respecté :

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..………………………………………………………………………………………………………………………………………………………..

**5.5. Principe DIP** (**D**ependency **I**nversion **P**rinciple : Principe d’inversion de dépendances)

* Le principe DIP vous parait-il toujours respecté ?

Oui ❑ Non ❑

Si non, indiquez le(s) endroit(s) dans le code où vous pensez que ce principe n’est pas respecté :

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..………………………………………………………………………………………………………………………………………………………..

1. ***Zoom sur d’éventuels design patterns (M3105)***

Si vous pensez qu’à certains endroits du code, certains patterns auraient pu être utilisés, n’hésitez

pas à le faire remarquer et indiquez au travers de suggestions/conseils le(s) endroit(s)

dans le code où vous pensez que ce(s) éventuel(s) pattern(s) pourrai(en)t être implémenté(s)

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

1. **Bilan…. Rédigez ci-après un bilan de *votre revue de code*** suite à votre lecture du code et aux remarques que vous a inspiré ce code (remarques venues en remplissant la fiche précédente et/ou en tenant compte de vos propres impressions/réactions face à ce code).

La revue de code devra faire remonter les *bad smells* que vous avez identifiés dans le code au travers de remarques bienveillantes accompagnées de conseils/suggestions d’amélioration pour que le(s) auteur(s) de ce code puisse(nt) l’améliorer :

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

**En fin de séance :** Rubrique à compléter par le(s) auteur(s) du code, pas ceux qui font la revue 😉

Avez-vous utilisé le TDD pour développer ce bout de code ?

Oui ❑ Non ❑

Est-ce que la revue de code qui vous a été faite vous a permis d’améliorer votre code :

Oui ❑ Non ❑

Justifiez en quoi cette revue de code vous a permis d’améliorer ou pas votre code :

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

Indiquez également quelles ont été les premières modifications que vous avez effectué dans votre code suite aux remarques/conseils de cette revue ….

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..……………………………………………………………………………………………………………………………………………………….

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

……..………………………………………………………………………………………………………………………………………………………..

Merci de remettre cette feuille anonyme à votre enseignant de TP en sortant de la salle 😉