

# Qualité du dev – TP

## Gestion Versionning dans un contexte d'entreprise

---

*Le Cas usurpation DAS Compagny*

---

# Table des matières

## Table des matières

<b>1. CONTEXTE .....</b>	<b>4</b>
1.1. OBJECTIF .....	4
1.2. CONSIGNES : .....	5
1.3. BILAN DES CHAPITRES .....	6
1.4. SCREENSHOT GIT.....	7
<b>2. ONCE UPON A TIME.....</b>	<b>10</b>
<b>3. 1ERE MISE EN PRODUCTION .....</b>	<b>11</b>
3.1. ACTIONS À RÉALISER.....	11
3.2. BILAN.....	11
<b>4. 1<sup>ER</sup> LOT D'EVOLUTION.....</b>	<b>12</b>
4.1. ACTION À RÉALISER .....	12
4.2. LE HÉROS.....	13
4.3. LA MARQUE ENTREPRISE .....	13
4.4. L'HISTOIRE.....	14
4.5. PASSER EN FREEMIUM.....	14
4.6. BILAN.....	15
<b>5. JOUR DE LIVRAISON.....</b>	<b>16</b>
5.1. ACTION À RÉALISER .....	16
5.2. BILAN.....	16
<b>6. PANIC EN PROD .....</b>	<b>17</b>
6.1. ACTION À RÉALISER .....	17
6.2. BILAN.....	17
<b>7. RETOUR DE QUALIF .....</b>	<b>18</b>
7.1. ACTION À RÉALISER .....	18
7.2. BILAN.....	18
<b>8. NOUVELLE MISE EN PRODUCTION .....</b>	<b>19</b>

8.1.	ACTION À RÉALISER .....	19
8.2.	BILAN.....	19
<b>9.</b>	<b>ALERTE ! THE FRANGLAIS IS NOT VAIRY GOUDE. ....</b>	<b>20</b>
9.1.	ACTION À RÉALISER .....	20
9.2.	BILAN.....	20
<b>10.</b>	<b>NEXT LEVEL.....</b>	<b>21</b>
10.1.	ACTION À RÉALISER .....	21
10.2.	SCREENSHOT GIT.....	21

# 1. Contexte

---

## 1.1. Objectif

L'objectif de ce TP est de comprendre les différents impacts lors de la mise en place d'évolutions ou de correctifs, dans le cadre d'une application en production, soumis à une recette et co-réalisé par plusieurs équipes de dev.

Ainsi l'objectif de ce TP est d'utiliser les concepts de gitflow (sans utiliser d'éventuels plugin) pour pouvoir réaliser en // différentes évolutions en fonction de la/les cibles.

Seront abordés les concepts suivants :

- Les merges
- Les tags
- La gestion des versions selon les environnements
- Les flow feature / release / hotfix / support

Le présent exemple ne tient pas compte :

- D'une architecture logicielle structurée.
- Des tests.
- De la qualité du code modifié.

## 1.2. Consignes :

---

**IMPORTANT : Pour chaque chapitre, ne réaliser QUE les consignes mentionnées ! Les étapes sont progressives, même si vous savez quoi faire après, ne le faites pas.**

A chaque étape, réalisez bien les push qui permettent de voir vos branches dans votre github, avec des commentaires **clairs**.

L'objectif est surtout de gérer les versions, pas nécessairement développer ce qui est demandé (sauf pour le dernier chapitre bonus, avec un vrai trésor à la clé).

Ainsi, pour chaque chapitre, vous devez à minima modifier les classes, mais uniquement en ajoutant des commentaires pour expliquer que c'est ici que vous rajouterez le code. Attention à commenter toutes les classes concernés par l'évolution.

Vous pouvez bien évidemment réaliser les évolutions en vrai, mais cela demande un peu plus de temps...

Pour chaque chapitre complétez la fiche réponse.

Les ressources du projets sont disponible sur le git suivant :

**[https://github.com/JeremyM-Sensei/BUT3\\_TP\\_GestionVersion.git](https://github.com/JeremyM-Sensei/BUT3_TP_GestionVersion.git)**

**NE CLONER PAS LE PROJET, NE FAITES PAS DE FORK.**

**TELECHARGEZ LE CONTENU, dézippez le dossier talesOfABlindWorldGOTY\_24.0.7z**

Le contenu du dossier **talesOfABlindWorldGOTY** sera le dossier racine sur lequel vous initialiserez votre git. Le dossier rambo.zip servira plus tard...

A l'ouverture du projet avec IntelliJ, précisez qu'il s'agit d'un projet maven.

Pour rappel, l'exécution du projet se fait via la commande : « mvn clean javafx :run »

### 1.3. Bilan des chapitres

Reportez les versions installées dans les différents environnements et les branches actives de votre git dans les tableaux suivants

**Version du produit sur les environnements :**

	Chapitre 3	Chapitre 4	Chapitre 5	Chapitre 6	Chapitre 7	Chapitre 8	Chapitre 9
Dev :							
Dev Ref :							
Recette :							
Production :							

**Branches existantes :**

	Chapitre 3	Chapitre 4	Chapitre 5	Chapitre 6	Chapitre 7	Chapitre 8	Chapitre 9
Liste des branches							

## 1.4. Screenshot Git

---

<b>Screenshot Git Chapitre 3</b>	
<b>Screenshot Git Chapitre 4</b>	
<b>Screenshot Git Chapitre 5</b>	

<b>Screenshot Git Chapitre 6</b>	
<b>Screenshot Git Chapitre 7</b>	
<b>Screenshot Git Chapitre 8</b>	
<b>Screenshot Git Chapitre 9</b>	



**Chapitre X**

*Déposer ici le  
screenshot de la  
discussion finale.*

## 2. Once upon a time...

Réaliser en 2021 par la DAS Company, ce jeu a été laissé libre de droit et peut ainsi profiter à d'autres entreprises bien moins bienveillantes qui voient en ce jeu un potentiel économique intéressant. Heureusement un courageux professeur avait caché le code sur une disquette 3.5".

Malgré toutes ses précautions, un jour de cours en qualité de dev, vous voyez la disquette dépasser de la veste du professeur et décidez donc de la subtiliser afin de l'utiliser au profit de votre entreprise.

Votre entreprise florissante constituée de :

- 2 équipes de dev.
- 1 équipe de recette.
- 1 équipe opérateur.
- 1 département juridique.
- 57 commerciaux avec de magnifiques costumes.

Le courageux professeur avait toutefois modifié le programme pour le brider et empêcher tout étudiants mal intentionnés et pas suffisamment bon en programmation, de profiter de ce jeu.

Très rapidement vous déposez le code de gitlab de l'IUT :

[https://github.com/JeremyM-Sensei/BUT3\\_TP\\_GestionVersion.git](https://github.com/JeremyM-Sensei/BUT3_TP_GestionVersion.git)

## 3. 1ere Mise en production

---

L'entreprise décide donc de reprendre les travaux initiés en 2021, de réaliser une rapide recette pour s'assurer du bon fonctionnement et de mettre en production la version.

### 3.1. Actions à réaliser

---

Votre tech leader vous demande de :

- Mettre en place un dépôt Git
- Associer les sources au dépôt git.
- Définir les branches de bases
- Réaliser le tag.

### 3.2. Bilan

---

Reportez les versions installées dans les différents environnements et les branches actives de votre git dans le tableau « 1.3 Bilan des chapitres »

Faites une copie écran des branches git et le déposer dans le tableau des screenshot git.

## 4. 1<sup>er</sup> lot d'évolution

L'entreprise DAS, à la suite d'une brillante réunion de brainstorming, constate plusieurs faiblesses sur le produit :

- Le héros n'est pas suffisamment viril, il faut remplacer par le mythique Rambo.
- La marque entreprise doit être revue pour coller avec votre entreprise et faire disparaître l'ancienne.
- Il faut revoir l'histoire.
- Et surtout...Le jeu ne rapporte pas d'argent.

### 4.1. Action à réaliser

A vous de réaliser, **en parallèle**, chaque fonctionnalité. Prenez soin de ne pas mélanger vos évolutions.

## Fonctionnalité 1 : Le héros

---

En utilisant les ressources disponibles sur moodle, changez le visuel du personnage.



## Fonctionnalité 2 : La marque entreprise

---

Revoir dans le jeu les différents endroits où sont mentionnés :

- Le nom de l'entreprise.
- Le nom des auteurs

Mettez votre marque, et pensez à ajouter une Licence GNU, ils avaient pas y penser avant !

### Fonctionnalité 3 : L'histoire.

Les histoires de livre perdu, ce n'est pas vendeur, modifiez l'histoire du jeu ainsi que son nom beaucoup trop ringard. Revoyez donc le résumé du chapitre et le nom du jeu.



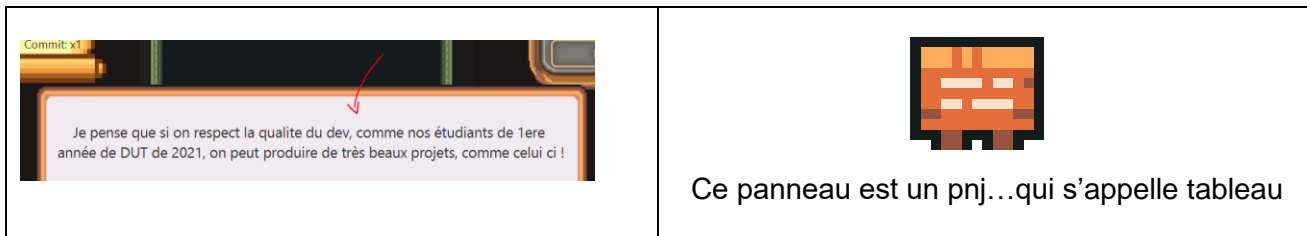
### Fonctionnalité 4 : Passer en Freemium

Il est temps d'implémenter l'élément le plus important faire de l'argent.

Ainsi pour pouvoir changer de niveau, il faut que le joueur :

- Soit doit faire 5.000 pas (bref agacer le joueur)
- Soit consommer une publicité. (Faire rentrer l'argent \$-) )

Faites en sorte que ces consignes soient données par le spectre du début de jeu, ajouter un panneau publicitaire qui quand on clique dessus sort un « slogan mythique » à votre guise.



## 4.2. Bilan

---

Reportez les versions installées dans les différents environnements et les branches actives de votre git dans le tableau « 1.3 Bilan des chapitres »

Faites une copie écran des branches git et le déposer dans le tableau des screenshot git.

## 5. Jour de livraison

Ça y est on est prêt pour la livraison...Mais un mail urgent de la com' nous alerte sur l'une des évolutions demandées : La présence de Rambo...

La licence, achetée 10.000.000\$ à Paramount, est jugée complètement has been, l'équipe prospective et statistiques nous alerte de la perte potentielle d'une partie de la population progressiste, et que les réactionnaires de tout façon, ne jouent pas aux jeux vidéo.

**Ainsi la fonctionnalité héros doit être écartée de la livraison.**

### 5.1. Action à réaliser

---

Débutez la release et livrez le composant pour permettre à la recette de valider vos évolutions.

### 5.2. Bilan

---

Reportez les versions installées dans les différents environnements et les branches actives de votre git dans le tableau « 1.3 Bilan des chapitres »

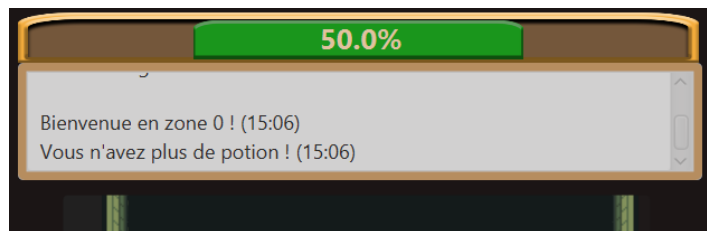
Faites une copie écran des branches git et le déposer dans le tableau des screenshot git.



## 6. Panic en prod

Pendant que l'équipe de recette test les nouvelles fonctionnalités, un twitter (« Xeur ») a mis l'entreprise sans dessus dessous...

Un célèbre Streamer vient de publier un let's play du jeu....Et vient de détecter un bug... En voulant boire une potion, il s'avère qu'il..perd 50% de sa vie !!!



### 6.1. Action à réaliser

---

Menez sur le champ un correctif, et déployez-le en production dès que possible.

### 6.2. Bilan

---

Reportez les versions installées dans les différents environnements et les branches actives de votre git dans le tableau « 1.3 Bilan des chapitres »

Faites une copie écran des branches git et le déposer dans le tableau des screenshot git.

## 7. Retour de Qualif

L'équipe de recette vous fait part de plusieurs retours, toutes mentionnées dans des issues :

- #001 : Le héros se déplace trop vite, le cahier des charges annonce 0,02s par frame.
- #002 : Afin de respecter les joueurs vegans, l'aliment du milieu doit être du pain et non pas de la viande.



- #003 : Le joueur doit commencer avec seulement 5 pièces. Réduire donc l'argent initiale.

### 7.1. Action à réaliser

---

Réalisez les évolutions de retour de recette.

### 7.2. Bilan

Reportez les versions installées dans les différents environnements et les branches actives de votre git dans le tableau « 1.3 Bilan des chapitres »

Faites une copie écran des branches git et le déposer dans le tableau des screenshot git.

## 8. Nouvelle mise en production

---

Maintenant que la version est validée par la recette, mettre officiellement en production

### 8.1. Action à réaliser

---

Procéder à la validation de cette release, réaliser un tag de cette version.

Préparer en dev la prochaine version.

### 8.2. Bilan

Reportez les versions installées dans les différents environnements et les branches actives de votre git dans le tableau « 1.3 Bilan des chapitres »

Faites une copie écran des branches git et le déposer dans le tableau des screenshot git.

## 9. Alerte ! The Franglais is not vairy goude.

Bien que la nouvelle version apporte son lot d'évolution, une nouvelle raillerie sur le net fait la une des réseaux sociaux. En effet les différents menus mélanges maladroitement de l'anglais et du français. N'ayant pas forcé l'installation obligatoire de la dernière version pour jouer, le jeu se trouve en production sur plusieurs version mineur (on considère toutefois qu'un hotfix est obligatoirement porté sur le client)

### 9.1. Action à réaliser

Réaliser les correctifs sur toutes les versions mineures.

### 9.2. Bilan

---

Reportez les versions installées dans les différents environnements et les branches actives de votre git dans le tableau « 1.3 Bilan des chapitres »

Faites une copie écran des branches git et le déposer dans le tableau des screenshot git.

## 10. Next Level.

Comme vous avez pu le constater, le méchant professeur a bridé le jeu...

Corriger le pour pouvoir jouer jusqu'au bout et gagner l'objet bonus.

### 10.1. Action à réaliser

---

Debugger le jeu.

Enjoy. !

---

### 10.2. Screenshot Git

---

*Déposer ici le screenshot de la discussion finale.*