**RAPPORTS SEANCES**

SÉANCE DU 24.03.22 :

**Points abordés :**

* Fin du README + organisation générale du projet

**Avancée** :

* Fini Méthologie I
* Début sprint I

**Distribution tâches :**

Olivier : command line

Nicolas : input (parseur, lecture md)

Victoria : output (écriture html)

Requirements :

* Command line :
  + Initialiser / compiler / nettoyer un site statique
* Output html (classe) :
  + Fonction de création de balises suivant h1, p, etc
  + Fonction générant page htm l(écriture)
* Parseur :
  + Parseur yml
  + Parseur md (titres, paragraphes, images, listes à puces)

NB : pour release, après avoir push, faire :

git tag numero\_verison

SÉANCE DU 31.03.22 :

**Progrès réalisés / attendus :**

* Nicolas : fini parsing md 🡪 html , reste à parse yaml
* Olivier  : Correction du readme + début command line
* Victoria : Début parsing Yaml ; recherche d’algorithme + débugs
* Par rapport à ce que nous avions prévu, nous pensions coder entièrement la traduction markdown 🡪 html. Nous avons cependant trouvé des fonctions déjà implémentées en librairie Java ; nous avons donc décidé d’utiliser ces fonctions.

**Sélectionner les stories à mettre en oeuvre ensemble**

* Nous allons continuer sur les story du sprint I

**Diviser les stories en tâches**

**Répartition du travail**

SÉANCE DU 07.04.22 :

**Progrès réalisés / attendus :**

|  |  |  |  |
| --- | --- | --- | --- |
|  | attendu | réalisé | prochain objectif |
| Nicolas | Ajouter lecture fichier pour le parser (string 🡪 fichier) | Ajouté lecture de fichier pour le parser |  |
| Olivier | Cli version init, compile, build et clean | Recherches pour command line, struggle sur fonction « version » |  |
| Victoria | Traitement du YAML | Trouvé implémentation markdown + YAML tout en 1, debug 🡪 à implémenter auj |  |

* Par rapport à ce que nous avions prévu, nous pensions coder entièrement la traduction markdown 🡪 html. Nous avons cependant trouvé des fonctions déjà implémentées en librairie Java ; nous avons donc décidé d’utiliser ces fonctions.

**Sélectionner les stories à mettre en oeuvre ensemble**

Nous allons continuer les stories du Sprint I.

**Diviser les stories en tâches**

* cli version
* cli init
* cli compile
* cli clean
* traitement + tests md et YAML

**Répartition du travail**

* Vic : YAML+md intégrés
* Olivier : commandes
* Nicolas : tests YAML + md intégré

SÉANCE DU 14.04.22 :

**Progrès réalisés / attendus :**

|  |  |  |  |
| --- | --- | --- | --- |
|  | attendu | réalisé | prochain objectif |
| Nicolas | tests YAML + md intégré | Ajouté lecture de fichier pour le parser |  |
| Olivier | commandes | Recherches pour command line, struggle sur fonction « version » |  |
| Victoria | YAML+md intégrés | YAML+md intégrés |  |

* Par rapport à ce que nous avions prévu, nous pensions coder entièrement la traduction markdown 🡪 html. Nous avons cependant trouvé des fonctions déjà implémentées en librairie Java ; nous avons donc décidé d’utiliser ces fonctions.

**Sélectionner les stories à mettre en oeuvre ensemble**

Nous allons continuer les stories du Sprint I.

**Diviser les stories en tâches**

* cli version
* cli init
* cli compile
* cli clean
* traitement + tests md et YAML

**Répartition du travail**

* Vic : YAML+md intégrés
* Olivier : commandes
* Nicolas : tests YAML + md intégré
* Vic : check avec Nicolas la fonction, pour qu’elle crée et écrive direct un fichier html

Dire qu’on a choisi de mettre les noms de fichiers en chemin relatif

Mettre ça dans le portfolio :

On a fait comme ça (avant) dans premier sprint, puis on a commencé comme ça dans le sprint II

🡪 niveau issue : c’est quoi une issue pour nous ?

🡪 Convensions :

* Commit commencent par fonctionel, fini par ce qui reste a faire sur branche
* Quand issue finir, mettre DONE (blabla)
* Nommage de branches : fb-blabla pour une nouvelle fonctionnalité
* Se mettre d’accord sur si on utilise des issues ou des checkbox

Dans portfolio :

* Soucis rencontrés, évolution, collaboration (difficile ou pas), temps pris, etc.