



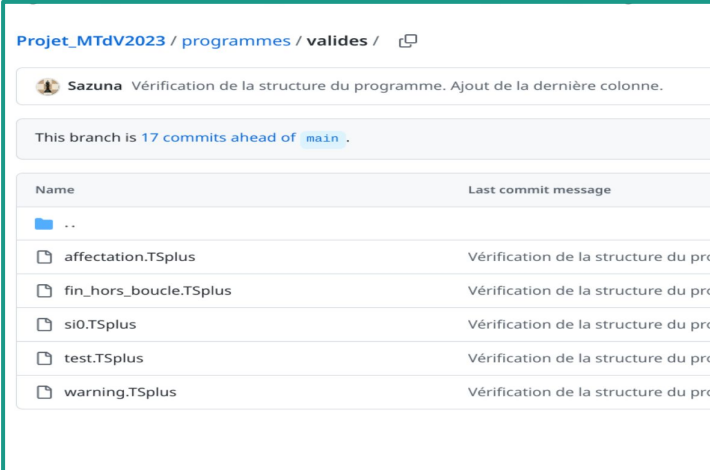
Présentation du Groupe 2

Laura Darenne
Shami Thirion Sen


Gestionnaire des noms de variables

INPUT : le fichier tsv du groupe 1

- grâce aux fichiers tsv correspondant aux programmes MDTV+ du groupe 1, nous obtenons des fichiers tsv avec des informations pertinentes concernant les variables



Name	Last commit message
..	
affectionation.TSplus	Vérification de la structure du pro
fin_hors_boucle.TSplus	Vérification de la structure du pro
si0.TSplus	Vérification de la structure du pro
test.TSplus	Vérification de la structure du pro
warning.TSplus	Vérification de la structure du pro

token_num 	line_n	token	type_token	instruction_numero	type instruction	pos_instruction	scope boucle
1	2	x	variable	1	affectation	G	-
2	2	=	opérateur	1	affectation	M	-
3	2	0	valeur	1	affectation	D	-
4	3	x	variable	2	affectation	G	-
5	3	=	opérateur	2	affectation	M	-
6	3	x	variable	2	affectation	D	-
7	3	+	opérateur	2	affectation	D	-
8	3	1	valeur	2	affectation	D	-
...



OUTPUT : création d'un dictionnaire qui suit les affectations

```
class GestionnaireVariables  
  
{  
    "x" : adresse_de_x  
    "y" : adresse_de_y  
}
```

La classe contient également plusieurs méthodes :

- addVariable : ajoute une variable au gestionnaire
- updateAdresse : met à jour l'adresse de la variable qui existe déjà dans le dictionnaire
- deleteVariable : supprime une variable dans le dictionnaire
- getVariables : affiche toutes les variables connues du gestionnaire sur le terminal
- getAdresse : renvoie l'adresse d'une variable du dictionnaire
- doesVariableExist : vérifie si une variable est présente dans le gestionnaire
- et d'autres au fur à mesure des demandes



OUTPUT : Comment aller chercher ce dictionnaire ?

création d'un module pour qu'il puisse être appelé comme ceci

```
from groupe2_GestionVariables import GestionnaireVariables  
  
dict_variables = GestionnaireVariables()
```

Gestion de la documentation et du reporting



Un template au format word ou/et latex

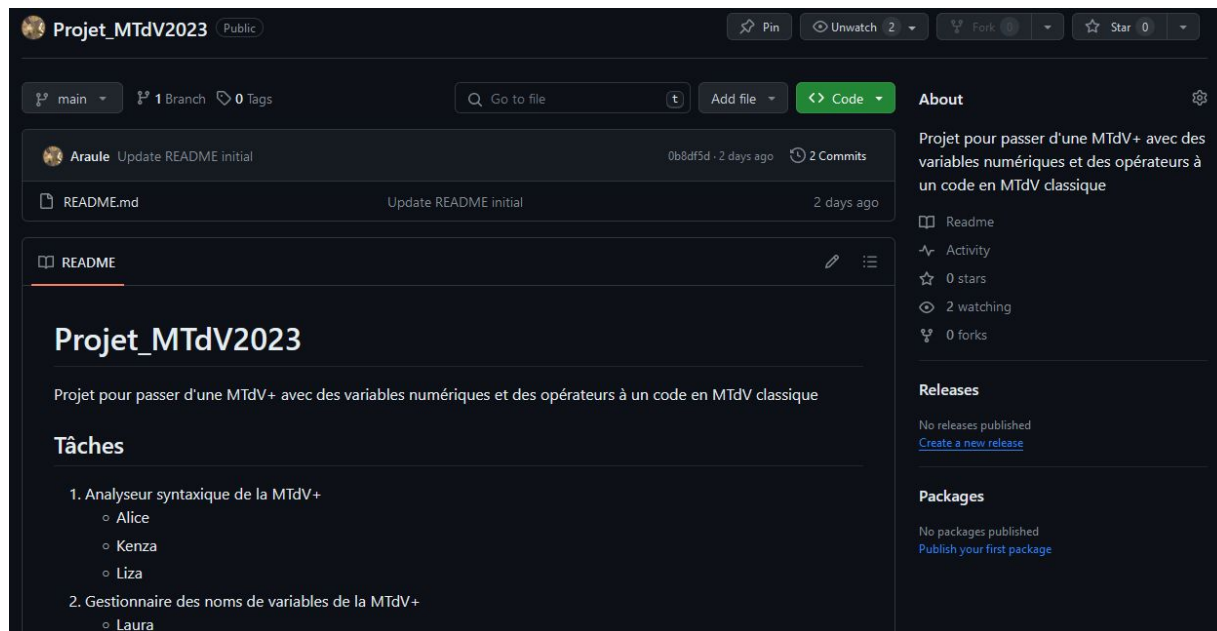
Nous mettons à disposition des templates que les groupes peuvent utiliser pour construire leur rapport.

Quelques idées rapides des parties :

- Réalisation des livrables
 - quoi, comment, quels problèmes rencontrés et comment ils ont été résolus
- Contributions
 - chaque membre de l'équipe explique ce qu'il a fait dans le travail de son groupe

Le Git du projet

Création du Git du projet



The screenshot shows a GitHub repository page for 'Projet_MTdV2023'. The repository is public and has 0 stars, 2 watchers, and 0 forks. The main branch is 'main'. The repository description is 'Projet pour passer d'une MTdV+ avec des variables numériques et des opérateurs à un code en MTdV classique'. The README file is visible, showing the project title and a list of tasks. The right sidebar contains sections for 'About', 'Releases', and 'Packages'.

Projet_MTdV2023 Public

main 1 Branch 0 Tags

Go to file Add file <> Code

Commit history:

- Araule Update README initial 0b8df5d · 2 days ago 2 Commits

Files:

- README.md Update README initial 2 days ago

README

Projet_MTdV2023

Projet pour passer d'une MTdV+ avec des variables numériques et des opérateurs à un code en MTdV classique

Tâches

- Analyseur syntaxique de la MTdV+
 - Alice
 - Kenza
 - Liza
- Gestionnaire des noms de variables de la MTdV+
 - Laura

About

Projet pour passer d'une MTdV+ avec des variables numériques et des opérateurs à un code en MTdV classique

Readme Activity 0 stars 2 watching 0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)



Création du Git du projet

- url : https://github.com/Araule/Projet_MTdV2023
- une branche par groupe et une branche pour les rapports, diaporamas et gantt
- le code final sur main

