**Projet ScIn1B**

Dijon 16/12/2019

Intégrants :

Cartaya Agustin

Moussa Hassan

Enseignant :

Bricq Stephanie

*Version FR*

Índex

[Projet 3](#_Toc27350328)

[Fonctionnalité du projet 3](#_Toc27350329)

[Modularisation du code 3](#_Toc27350330)

[Fichier shell (main.sh) 3](#_Toc27350331)

[Fichiers perl (Ex#.pl) 4](#_Toc27350332)

[Ex1.pl 4](#_Toc27350333)

[Ex2.pl 4](#_Toc27350334)

[Ex3.pl 5](#_Toc27350335)

[Ex4.pl 5](#_Toc27350336)

[Ex5.pl 6](#_Toc27350337)

[Fichiers Gnuplot (g1.gp y g1.gp) 7](#_Toc27350338)

[g1.gp 7](#_Toc27350339)

[g2.gp 7](#_Toc27350340)

[Remarques : 8](#_Toc27350341)

[Étapes à suivre pour l'exécutions 8](#_Toc27350342)

[Resultats 8](#_Toc27350343)

# Projet

## Fonctionnalité du projet

Tout le code écrit dans ce projet vise à lire, classer, interpréter et représenter graphiquement un tableau de données qui contient des informations sur les objets du « Metropolitan Museum of Art », des données dont la signification et la présentation ont été expliquées dans la déclaration proposée.

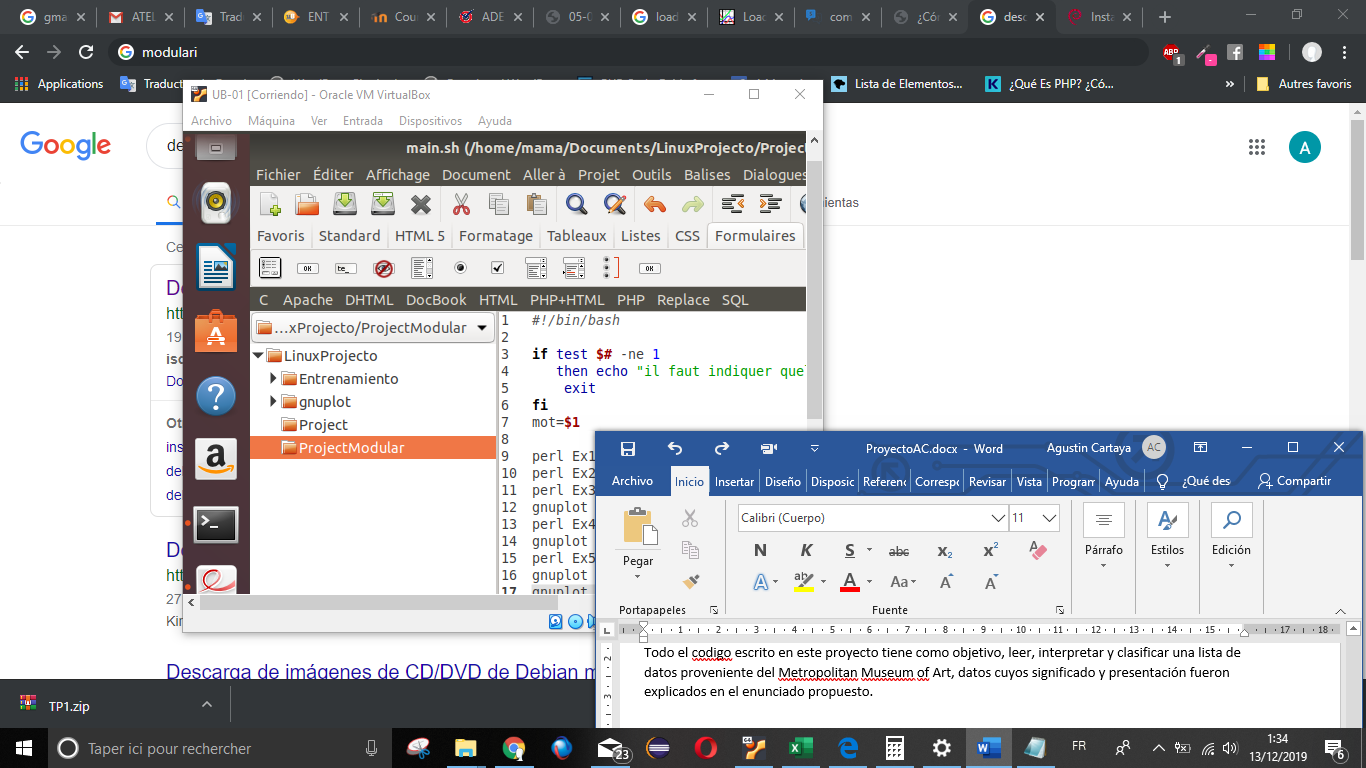
## Modularisation du code

Le code qui sera présenté est divisé en

* 1 fichier Shell (.sh)
* 5 fichiers Perl (.pl)
* 2 fichiers Gnuplot (.gp)

### Fichier shell (main.sh)

C'est le fichier principal en charge d'exécuter tous les autres et de transmettre automatiquement les données nécessaires. Ce fichier est le seul qui sera appelé par console lors de l'exécution du projet, ce fichier a besoin d’un paramètre qui est le nom du fichier qui contient les données à étudier.

(ligne 1) en-tête il est utilisé pour spécifier le chemin complet et les paramètres de l'interpréteur qui va exécuter le programme.

(lignes 3-7) conditionnel pour vérifier si le nom du fichier contenant la table de données a été passé en paramètre, si ce paramètre n'existe pas, il affiche une erreur dans la console (ligne 4) et quitte le programme (ligne 5). Si ce paramètre existe, il est stocké dans la variable $ table (ligne 7) et le programme continuera son exécution.

(lignes 9-17) exécution de fichiers .pl y .gp en charges de classer les données.

* Les fichiers perl sont exécutés en tapant le mot «perl» puis le nom du fichier à exécuter. Exemple *perl Ex2.pl* si le fichier à exécuter doit passer un paramètre (cas unique Ex1.pl) cet-ci doit être placé après le nom du fichier. Exemple *perl Ex1.pl $ table*
* Les fichiers Gnuplot sont exécutés en mettant d'abord le mot "gnuplot", puis en utilisant le mot clé "-e" on indique qu’on va utiliser des variables et enfin on écrit le nom du fichier. Exemple *gnuplot -e "fn1 = 'filename\_dep' g1.gp*

### Fichiers perl (Ex#.pl)

Il y a 5 fichiers .pl qui sont chargés de lire et de classer le tableau de données selon les exercices, tous les fichiers ont une structure similaire, qui a été divisée en blocs pour faciliter la compréhension.

Dans chaque fichier, il y a 6 blocs énumérés de A à F, sauf indication contraire, chaque bloc a la fonction suivante :

Bloc #A: création de variables communes utilisées pour lire les données

* $fichier: variable où sera enregistré le fichier de lecture
* $ligne: variable où sera enregistré les lignes dans la boucle
* @par: tableau où chaque colonne de la ligne est stockée
* $sorti: variable où sera enregistré le fichier de lecture

Bloc #B: création de variables propre l'exercice

Bloc #C: ouverture de fichiers de lecture et d’écriture

Bloc #D: début de la boucle while

Bloc #D1: obtention sw La ligne et division de colonnes avec la méthode split ()

Bloc #D2: Partie logique de l'exercice (conditionnel, compteurs et écriture de fichiers)

Bloc #E: Écriture des résultats dans des fichiers ou dans la console

Bloc #F: fermeture de fichiers

#### Ex1.pl

Bloc #B (variables)

* $valides: compteur de lignes valides
* $publics: compteur d'objets publics
* $date: compteur d'objets avec date
* $p: compteur d'objets dont le nom commence pour P

Bloc #D2

|  |  |
| --- | --- |
| Condition | Fonction |
| if(@par[2] =~ /^[0-9]+$/) | Vérifier que la colonne 3 ne contient que des chiffres |
| if(@par[1] =~ /true/i) | Vérifier que la colonne 2 contient le mot “true’ (en minuscule ou majuscules) |
| if( @par[7] =~ /[a-zA-z0-9]+/ && @par[7] !~ /n\.d/i) | Vérifiez que la colonne 8 contient au moins un caractère alphanumérique et qui est différente de “n.d” (en minuscule ou majuscules) |
| if( @par[4] =~ /^P/) | Vérifiez que la colonne 8 commence pour P (majuscule) |

#### Ex2.pl

Bloc #B (variables)

* $sd: compteur d'objets sans dimension

Bloc #D2

|  |  |
| --- | --- |
| Condition | Fonction |
| If ( @par[9] !~ /[0-9]+/) | Vérifiez que la colonne 10 ne contient pas de chiffres |

#### Ex3.pl

Bloc #B (variables)

* @dCtt: compteurs d'objets par département
  + dCtt[0] : compteurs d'objets de American Decorative Arts
  + dCtt[1] : compteurs d'objets de Drawings and Prints
  + dCtt[2] : compteurs d'objets de Medieval Art
  + dCtt[3] : compteurs d'objets de Egyptian Art

Bloc #D2

|  |  |
| --- | --- |
| Condition | Fonction |
| if (@par[3] =~ /(American Decorative Arts)/i) | Vérifiez que la colonne 4 contient “American Decorative Arts” (en minuscule ou majuscules) |
| elsif (@par[3] =~ /(Drawings and Prints)/i) | Vérifiez que la colonne 4 contient “Drawings and Prints” (en minuscule ou majuscules) |
| elsif (@par[3] =~ /(Medieval Art)/i) | Vérifiez que la colonne 4 contient “Medieval Art” (en minuscule ou majuscules) |
| elsif (@par[3] =~ /(Egyptian Art)/i) | Vérifiez que la colonne 4 contient “Egyptian Art” (en minuscule ou majuscules) |

#### Ex4.pl

Bloc #B (variables)

@mCtt: compteurs d'objets par classification

* + dCtt[0] : Compteur d'objets avec classification metal
  + dCtt[1] : Compteur d'objets avec classification silver
  + dCtt[2] : Compteur d'objets avec classification prints
  + dCtt[3] : Compteur d'objets avec classification drawings
  + dCtt[4] : Compteur d'objets avec classification ceramics
  + dCtt[5] : Compteur d'objets avec classification ephemera
  + dCtt[6] : Compteur d'objets avec une autre classification
  + dCtt[7] : Compteur d'objets sans classification

Bloc #D2

|  |  |
| --- | --- |
| Condition | Fonction |
| if (@par[11] =~ /metal/i) | Vérifiez que la colonne 12 contient “metal” (sin distinción de mayúsculas) |
| if (@par[11] =~ / silver /i) | Vérifiez que la colonne 12 contient “silver” (sin distinción de mayúsculas) |
| if (@par[11] =~ / prints /i) | Vérifiez que la colonne 12 contient “prints” (sin distinción de mayúsculas) |
| if (@par[11] =~ / drawings /i) | Vérifiez que la colonne 12 contient “drawings” (sin distinción de mayúsculas) |
| if (@par[11] =~ / ceramics /i) | Vérifiez que la colonne 12 contient “ceramics” (sin distinción de mayúsculas) |
| if (@par[11] =~ / ephemera /i) | Vérifiez que la colonne 12 contient “ephemera” (sin distinción de mayúsculas) |
| if (@par[11] !~ /[a-z0-9]/i) | Vérifiez que la colonne 12 ne contient pas de caractères alphanumériques |
| elsif (@par[11] !~ /(metal|silver|prints|Drawings|ceramics|Ephemera) | Vérifiez que la colonne 12 ne contient aucune des ces mots |

#### Ex5.pl

Bloc #B (variables)

* $h: variable pour enregistrer la hauteur
* $w: variable pour enregistrer la longueur
* $ar: variable pour enregistrer l’aire

Bloc #D2

|  |  |
| --- | --- |
| Condition | Fonction |
| if(@par[4] =~ /Print/ && @par[9] =~ /[0-9]+/) | Vérifiez que la colonne 5 contient “Print” et que la colonne 10 contient au moins une chiffre |
| if( @par[9] =~ /\(\D\*(\d+[.,]?\d\*)\D\*\).\*\(\D\*(\d+[.,]?\d\*)\D\*\)/) | Vérifiez que la colonne 10 a une structure pareil à “(#.#) (#.#)” les # indiquent une chiffre et aussi cette expression enregistre ces deux valeur numériques dans $1 y $2 (apres ces valeurs seront transférées aux variables $h y $w) |
| elsif( @par[9] =~ /\(\D\*(\d+[.,]?\d\*)\D\*(\d+[.,]?\d\*)\D\*\)/) | Verificar que la columna 10 tenga una estructura parecida a “(#.# x #.#)” |
| if(@par[4] =~ /Drawing/ && @par[9] =~ /[0-9]+/) | Vérifiez que la colonne 5 contient “ Drawing ” et que la colonne 10 contient au moins une chiffre |

Note

Les variables $h et $w ont été utilisées pour convertir les "," en "." Puisque le séparateur décimal n'a pas été établi dans l’énonce

### Fichiers Gnuplot (g1.gp y g1.gp)

Il y a 2 fichiers gnuplot qui sont chargés de faire les graphiques correspondant aux résultats obtenus dans certains exercices

#### g1.gp

Ce fichier est divisé en 5 blocs énumérés de A-E

* Bloc #A: Il est responsable de la réinitialisation de Gnuplot afin de ne pas avoir de conflits avec les paramètres précédemment établis, il établit également un titre et active la grille
* Bloc #B : Il est enchargé de définir les paramètres pour créer un histogramme
* Bloc #C: Définir la façon d'afficher les étiquettes sur l'axe des x
* Bloc #D: Définissez la sortie sous forme de fichier png avec le nom passé en argument enregistré dans la variable nf
* Bloc #E: créer le graphique en utilisant la deuxième colonne comme données et la première comme étiquettes pour l'axe des x

#### g2.gp

Ce fichier est divisé en 5 blocs énumérés de A-E

* Bloc #A: Il est responsable de la réinitialisation de Gnuplot afin de ne pas avoir de conflits avec les paramètres précédemment établis, il établit également un titre et active la grille
* Bloc #B : Il est enchargé de définir les paramètres pour créer un Boxplot
* Bloc #C: Définir la façon d'afficher les étiquettes sur l'axe des x
* Bloc #D: Définissez la sortie sous forme de fichier png avec le nom passé en argument enregistré dans la variable nf
* Bloc #E: créer le graphique en utilisant la deuxième colonne comme données

Guide d'utilisation

### Remarques :

* Perl et Gnuplot doivent être installés pour l'exécution du projet
* Il faut une liste de données dont le nom sera transmis par paramètre au fichier main.sh

### Étapes à suivre pour l'exécutions

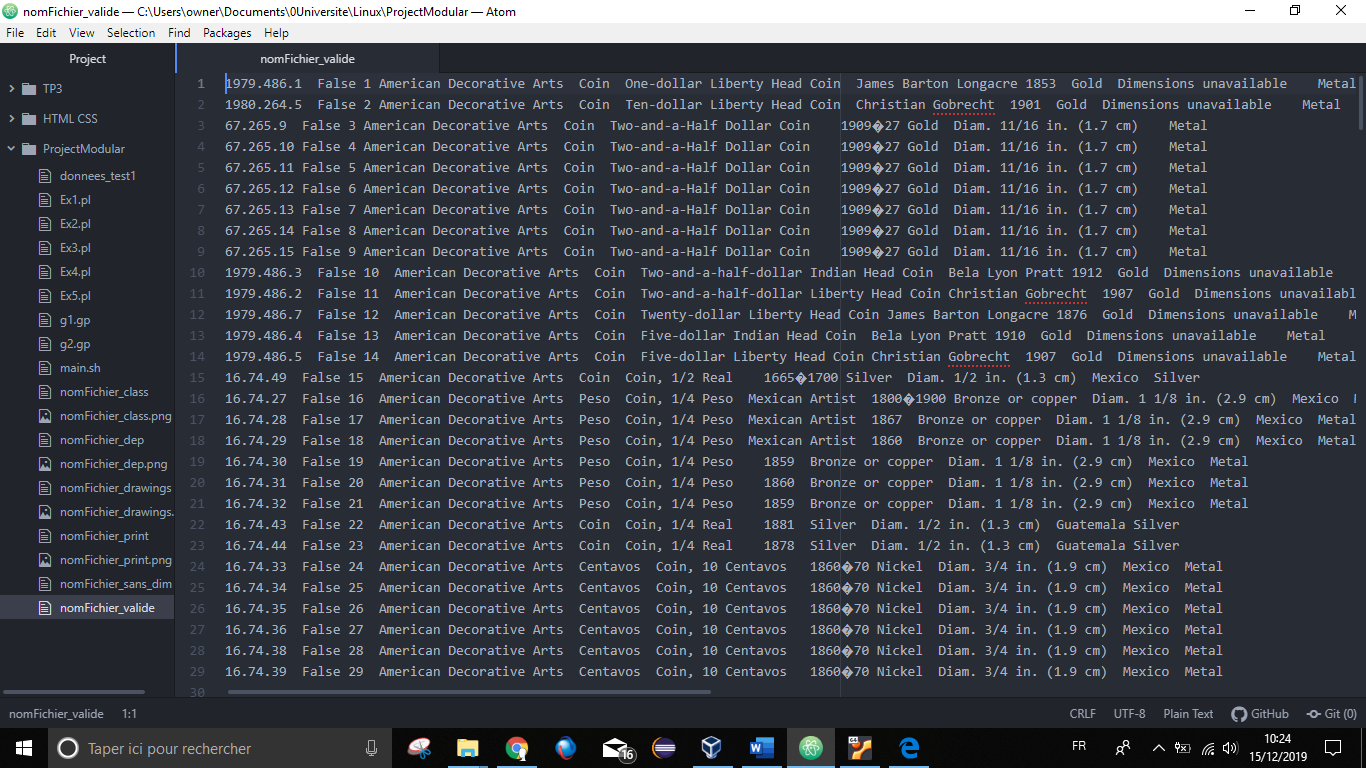
* Décompresser le fichier projet\_ScIn1B\_Cartaya\_Moussa.zip
* Entrez dans le dossier projet\_ScIn1B\_Cartaya\_Moussa
* Ouvrez un terminal à cet endroit
* Donnez des autorisations d'exécution au fichier main.sh avec la commande chmod u + x dans le terminal
* Exécuter le programme, pour cela il faut écrire dans le terminal *./main.sh* suivi du nom du fichier qui contient la liste de données Exemple *./main.sh donnees\_test1*

## Resultats

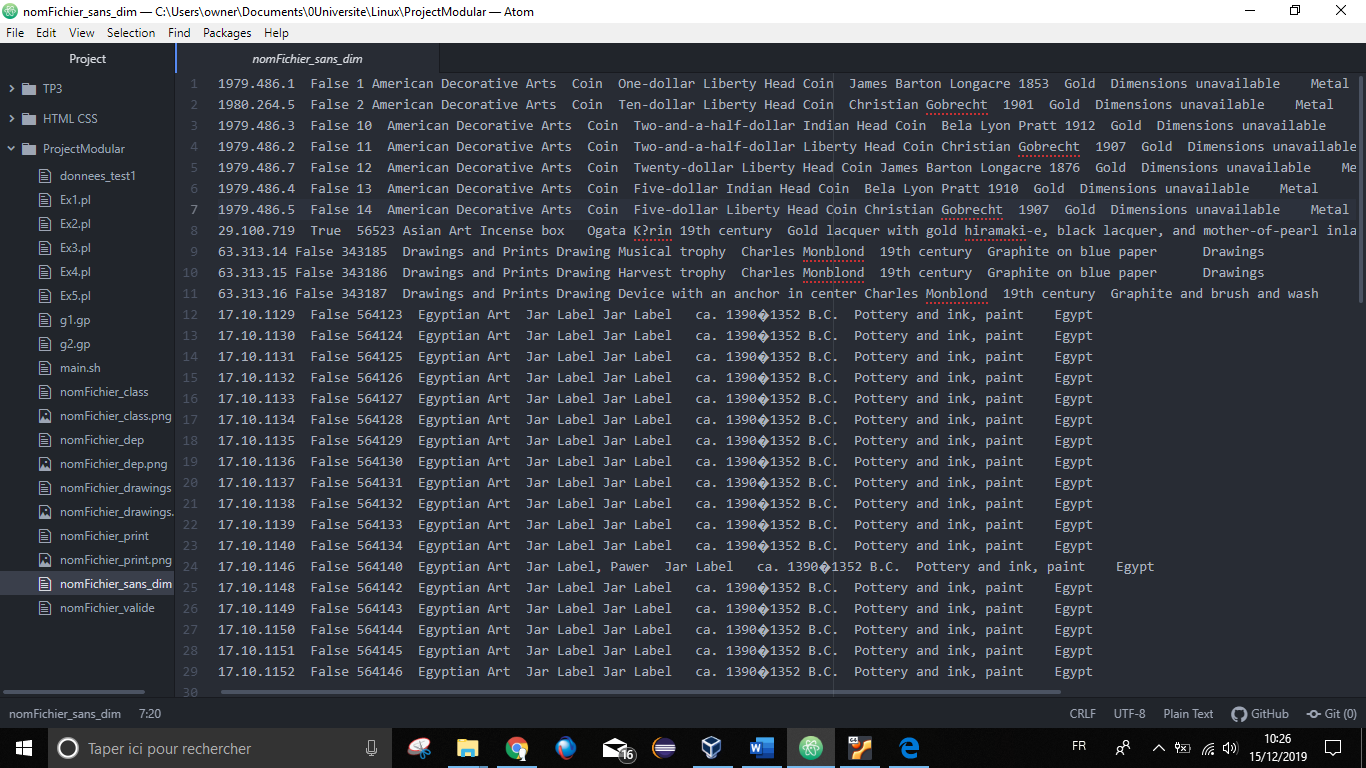
Lors de l'exécution du projet, il sera imprimé par la console

* Nombre de lignes valides
* Nombre d’objets publiques
* Nombre d’objets avec date
* Nombre d’objets dont le nom commence par P
* Nombre d’objets sans dimention

De plus, le programme créera 6 fichiers texte correspondant à chaque exercice proposé (pour l'exercice n ° 5, deux fichiers ont été créés) et 4 graphiques correspondant aux exercices 3, 4, 5. Les résultats sont présentés ci-dessous::

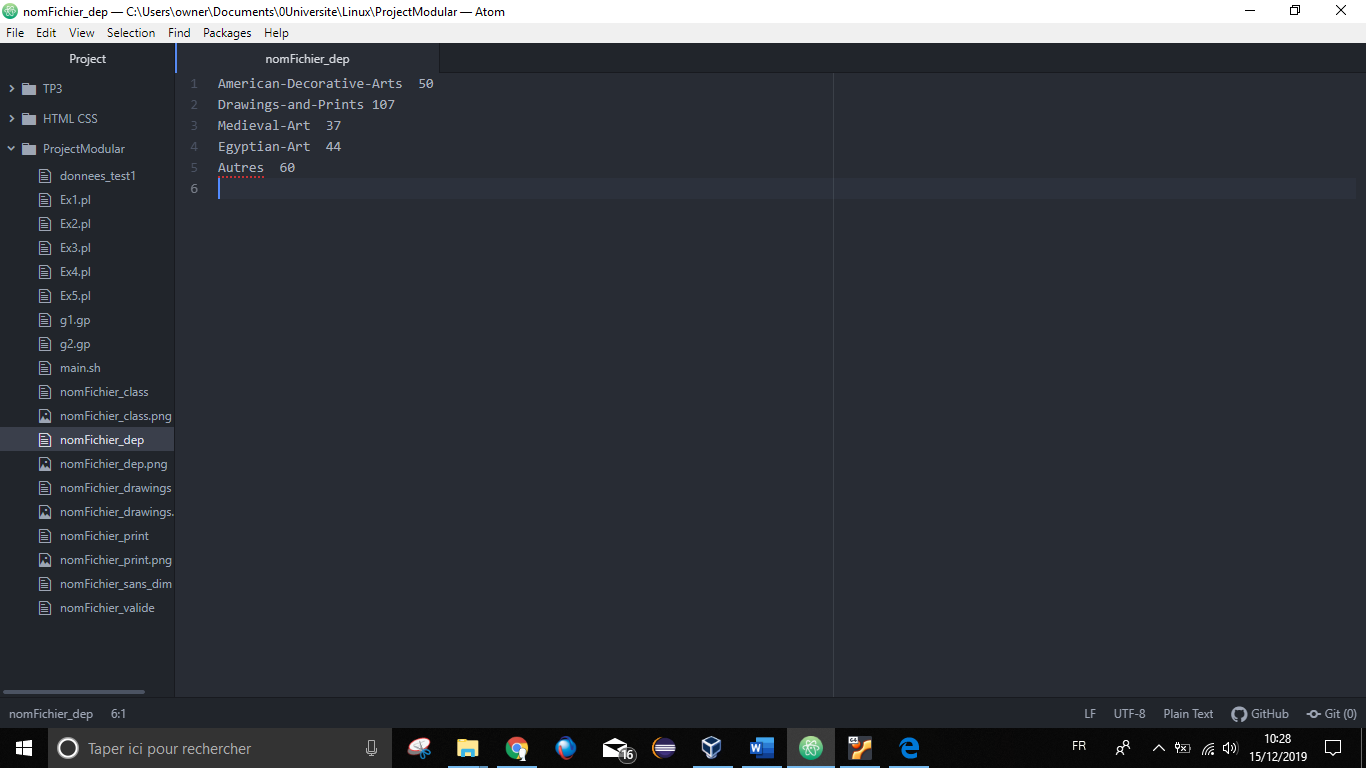
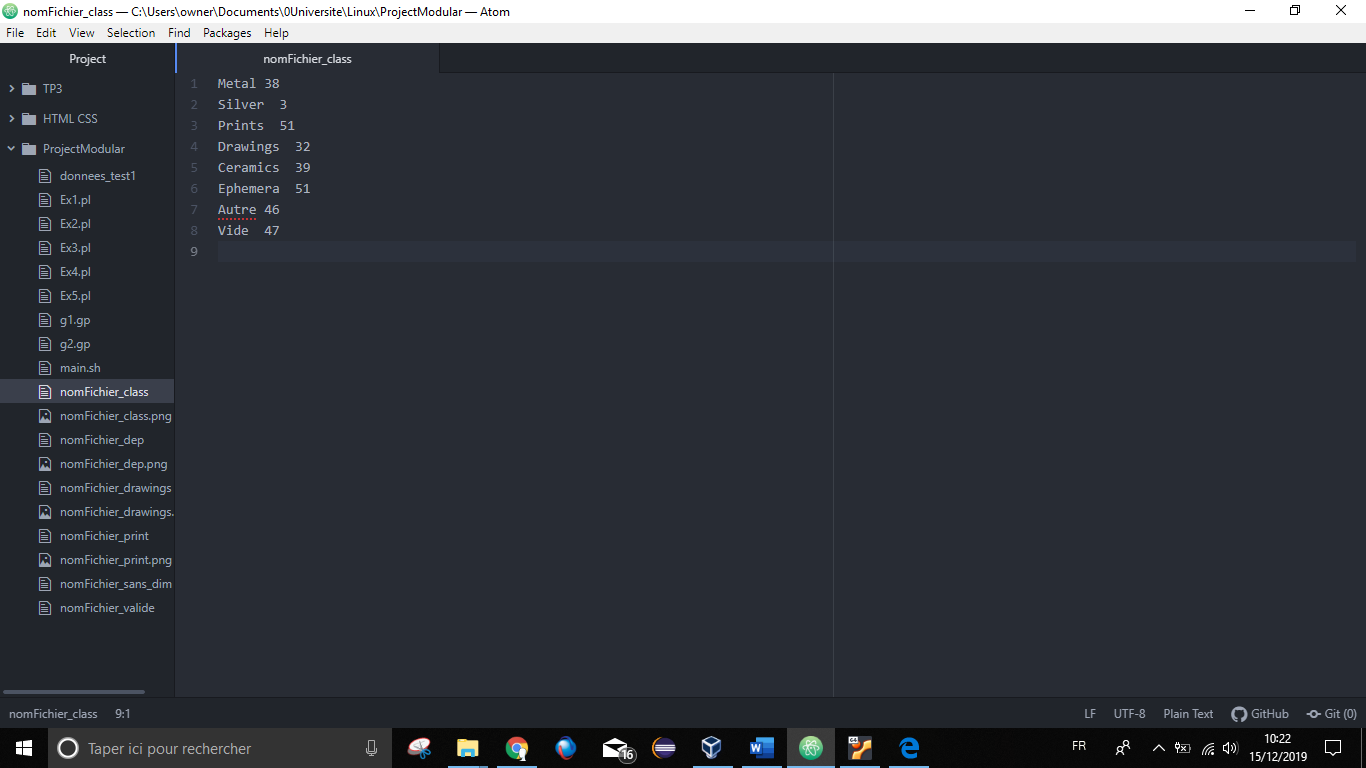
Ex1.pl (nomFichier\_valide)

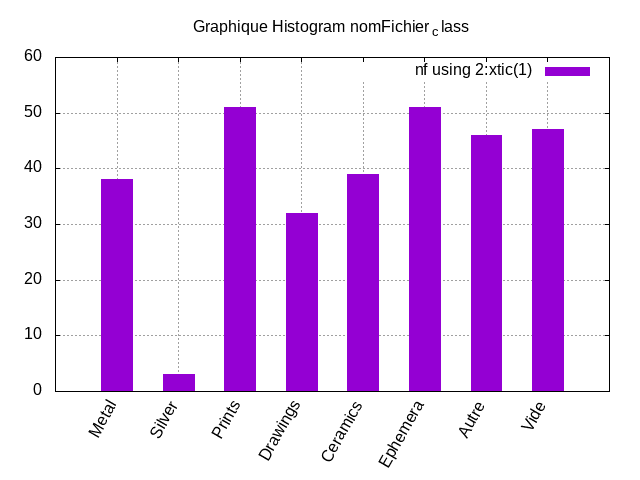
Ex2 (nomFichier\_sans\_dim)

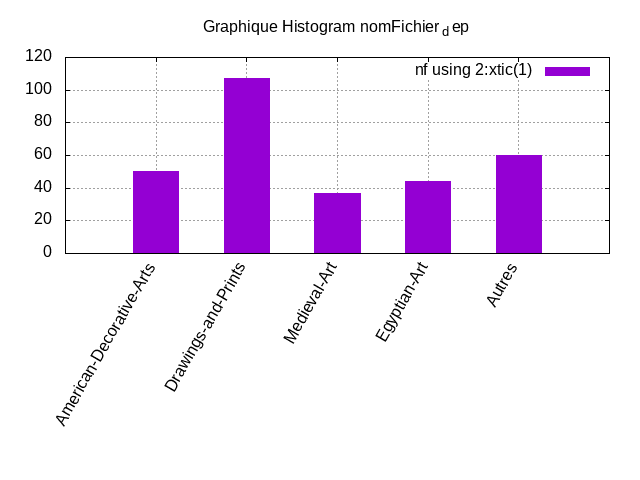


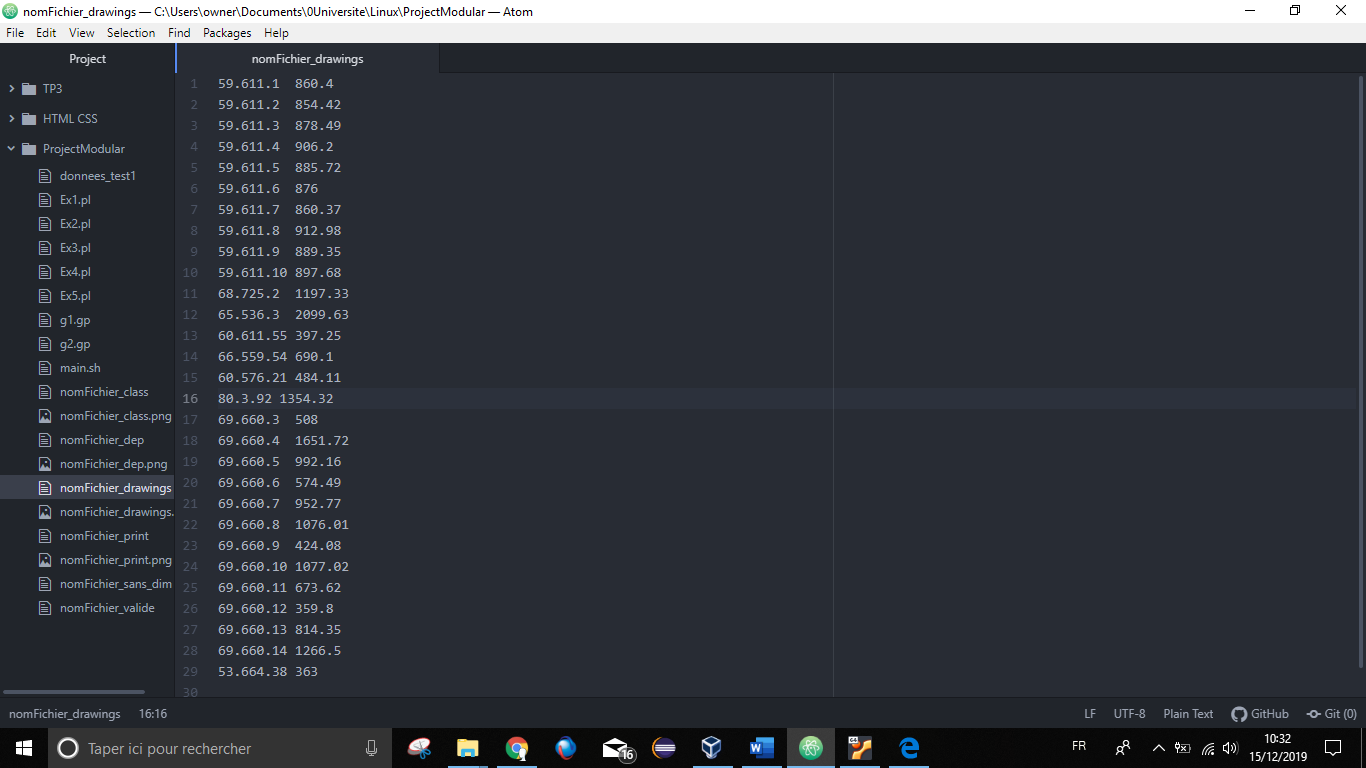
Ex4 (nomFichier\_class)

Ex3 (nomFichier\_dep)







Ex5 (nomFichier\_print, nomFichier\_drarings)

