

## Compte rendu du Sprint 1

### 1) Initialisation

-Beaucoup de recherche et de réflexion sur le fonctionnement du tas, de la pile et de la gestion mémoire. Lecture des liens disponibles dans la bibliographie.

### 2) Implémentation de la pile :

-Nous avons commencé par suivre le cours. Un état mémoire est composé d'une entrée (Quad). Un Quad est composé d'un ID (identificateur), d'une VAL (valeurs associées), d'un OBJ (nature de l'objet) et d'un SORTE (type de l'objet. (slide 26/76 du cours). et c'est cet objet qui est ajouté/modifié dans la pile.

-Nous avons donc réalisé une classe "Quad" composé de ces quatres variables avec son constructeurs associés. Avec les données et les exemples du cours, on a compris que OBJ (la nature de l'objet) était soit une variable, une constante, un tableau ou une méthode. Nous avons donc réalisé une classe énumération pour ces objets. Leurs noms dans le cours sont, var, vcst, tab et meth. Nous avons fait le choix de conserver la même syntaxe.

-Nous avons réalisé une grande partie des axiomes de gestion de la pile (diapo 28/76) dans une nouvelle classe "pile". Les fonctions réalisées sont : Empiler, Depiler, Echanger, DeclVar, DeclCst et DeclMeth. IdentVal et DeclTab n'ont pas été réalisées. Pour IdentVal, nous avons pas bien compris son utilisation. Et DeclTab nécessitait le tas.

-Nous nous sommes rendu compte que l'utilisation de la pile nécessitait une Table des symboles (diapo 16/76). Nous avons donc fait le choix de réaliser une classe "TableDesSymboles". Cette table est remplie lors des phases d'analyses (entre l'analyse lexicale et l'analyse syntaxique, voir lien en bibliographie). Lorsque son constructeurs est appelé nous lui affectons une HashMap. La table des symboles est composée d'entrée qui sont elles-mêmes composées d'un nom et d'un attribut. Le nom est une chaîne permettant d'identifier le symbole et l'attribut est un Quad. Cette table recensera donc plusieurs valeurs, nous avons donc créé une map avec clé le nom du symbole et valeur le Quad.

### 3) Création d'un objet pour le tas

-Le tas, nous n'y avons pas touché, mais avec notre réflexion et les échanges que nous avons eu. Nous pensons qu'il faut implémenter un arbre de liste. Cela permettra d'accéder rapidement aux données qui sont stockées sous forme de listes.

### 4) Réflexion à une fonction de hachage

La fonction de hachage qu'on avait eu lors d'un précédent projet (en L2 en algorithmie) :

```
size_t fnv_hash(const char *key) {  
    //constante pour système 64 bits  
    const size_t FNV_offset_basis = 14695981039346656037;  
    const size_t FNV_prime = 1099511628211;  
    size_t hash = FNV_offset_basis;  
    size_t longueur= str_length(key);  
    for (size_t i=0; i<longueur ;++i){  
        hash = hash * FNV_prime;
```

```

        hash = hash ^ (key[i]);
    }
    return hash;
}

```

Ce qu'on en pense : Cette fonction nous avait été donnée sans grande explication. Nous constatons que celle-ci reçoit en entrée un char \* dans notre cas ce sera sûrement un objet. Cette fonction peut-être une base de départ pour celle que nous aurons à réaliser.

#### 5) Problèmes rencontrés & potentiel solution

-L'implémentation de IdentVal : nous n'avons pas bien compris l'utilisation de cette fonction. Dans le cours nous avons remarqué que cette fonction prenait en entrée (ID x SORTE x MEM x NAT) et en sortie (MEM). Nous connaissons tous les paramètres sauf NAT. Au moment où nous écrivons le CR, nous ne connaissons toujours pas l'utilité de ce paramètre. -> pas de solution apportée.

#### 6) Conclusion

Ce qui a été implémenté :

- Pile
- Début de la table des symboles
- Quad

La prochaine étape :

- Réaliser le tas et le faire interagir avec le reste.
- Poursuivre la table des symboles
- Comprendre IdentVal ajouté DeclTab
- Test unitaire

#### 7) Bibliographie et lien utile

Lien générale :

<https://totodu.net/Compilation/Generation>  
<https://cel.archives-ouvertes.fr/inria-00001231/document>  
[https://www.youtube.com/channel/UCz\\_j3CVO1gFM7eSUC6IE39A/videos](https://www.youtube.com/channel/UCz_j3CVO1gFM7eSUC6IE39A/videos)  
<https://beta.hackndo.com/gestion-de-la-memoire/>  
<https://github.com/mlboua/Mini-jaja>

Lien sur la mémoire :

[https://fr.wikipedia.org/wiki/Pile\\_\(informatique\)](https://fr.wikipedia.org/wiki/Pile_(informatique))  
[https://fr.wikipedia.org/wiki/Pile\\_d%27ex%C3%A9cution](https://fr.wikipedia.org/wiki/Pile_d%27ex%C3%A9cution)  
[https://fr.wikipedia.org/wiki/Tas\\_\(informatique\)](https://fr.wikipedia.org/wiki/Tas_(informatique))  
[https://fr.wikipedia.org/wiki/Table\\_des\\_symboles](https://fr.wikipedia.org/wiki/Table_des_symboles)

Lien sur le hachage :

[https://fr.wikipedia.org/wiki/Liste\\_de\\_fonctions\\_de\\_hachage](https://fr.wikipedia.org/wiki/Liste_de_fonctions_de_hachage)  
[https://jipe.developpez.com/articles/algo/table-hachage/?page=page\\_1](https://jipe.developpez.com/articles/algo/table-hachage/?page=page_1)