



Avec l'aide de GRIGNON Romuald

CodYnGame

ABDELMALEK Mehdi, BELLETRE Jules, GUYOT Julien, LOPEZ Mateo, ARMANNO Yannis

Sommaire

Sommaire.....	2
Introduction.....	3
Organisation de l'équipe.....	4
Répartition des tâches.....	4
Réunion entre membre du groupe.....	4
Défis rencontrés.....	5
Jdbc.....	5
JavaFx.....	5
Mode Include.....	5
Amélioration possible.....	6
Documentation.....	7
Diagramme de cas d'utilisation.....	7
Diagramme de classe.....	7
To Do Liste.....	8

Introduction

Notre groupe est composé de 5 personnes. Les 4 premiers sont dans le même groupe et se sont donc réunis naturellement. Le cinquième membre du groupe, Mehdi est dans une autre classe mais son entente avec les autres membres de ce groupe est bonne depuis deux ans où différents projets avaient déjà été effectués en commun. Il a donc été normal pour les 4 autres de l'inviter au sein du groupe. Le niveau moyen en informatique au sein du groupe est assez bon, avec Mehdi et Jules ayant un peu plus d'expérience.

La première décision du groupe fut le choix du sujet. Nous nous sommes mis d'accord pour choisir à la fois un sujet challengeant et intéressant. C'est pour cela que nous avons décidé de ne pas choisir la solution de facilité (CY Books) qui nous semblait plutôt simple et peu attrayante. C'est le cas aussi pour l'arbre généalogique qui nous intéressait peu. Nous avons alors hésité entre ChromatYnk et CodYngame, et finalement, l'aléatoire a décidé CodYngame à notre place.

Le sujet consiste donc à programmer en java un logiciel proposant des exercices de programmation que l'on pourra coder et exécuter au sein du logiciel. Il y aura deux types d'exercice (en partie aléatoire), Include et Stdin/Stdout et l'on pourra suivre notre progression via certaines statistiques disponibles.

Organisation de l'équipe

Répartition des tâches

Nous avons commencé par faire une to do liste pour résumer le projet (cf Documentation).

Nous avons divisé le projet en 5 parties :

- Le stockage des exercices (classe + base de données)
- L'interface,
- Les exercices Include
- Les exercices Stdin/Stdout
- La compilation/exécution des langages.

Mehdi étant à l'aise avec les bases de données, il s'est chargé de cette partie. Mateo et Yannis se sont chargés de l'interface. Julien et Jules on fait ensemble la compilation/exécution et respectivement les exercices Include et Stdin/Stdout. De plus, Julien s'est occupé de générer la JavaDoc.

Réunion entre membre du groupe

Excepté les réunions avec le tuteur, nous n'avons pas fait de réunions régulières. En revanche, nous nous parlions tous les jours sur le discord que nous avons créé et dès que besoin, nous nous réunissions (partiellement ou entièrement) en vocal pour discuter du projet et des problèmes rencontrés. Avant chaque push sur github nous prévenions le groupe des modifications apportées.

Défis rencontrés

Jdbc

Mehdi a eu des difficultés avec l'installation de jdbc pour gérer la bdd dans notre projet ce qui l'a bloqué durant plusieurs jours. Il a finalement surmonté cela grâce à l'ajout d'une structuration maven au sein du projet, ce qui a permis l'installation semi-automatique des librairies voulue dans le projet.

JavaFx

On a rencontré beaucoup de difficultés pour faire fonctionner JavaFX avec le reste de projet, en fonction des versions, soit le JavaFX, soit la version terminal, ne marchait pas. Ce problème a été résolu grâce à Mehdi qui, après de nombreux essais, a trouvé comment le résoudre en lançant l'application JavaFx dans une classe Starters plutôt que de la lancer dans la classe App où elle est créée.

Mode Include

Julien a eu des difficultés avec la vérification du code saisi par l'utilisateur et sa comparaison avec la fonction solution. C'est notamment le fait d'envoyer le résultat des tests réalisés pour chaque exercice via le code retour au reste du code qui a été difficile. Il a finalement pu résoudre ce problème après de nombreuses tentatives dans les différents langages.

Amélioration possible

Il y a différentes choses qui étaient dans le cahier des charges et que nous n'avons pas pu faire pour des raisons de temps et/ou des raisons techniques.

Une des choses que nous n'avons pas pu faire dans les temps est le fait d'afficher les erreurs de compilation et/ou d'exécution dans l'enceinte du terminal intégré à l'appli graphique, les erreurs sont affichés dans le terminal qui tourne en arrière plan mais elle forcent l'utilisateur à faire des alt tab de temps à autre.

Une autre chose qui ne marche que trop peu est le fait que l'appli tourne indéfiniment si l'user rentre une boucle infinie, nous avons mis un cas d'arrêt pour le langage C mais nous n'avons pas eu le temps de le faire pour les autres langages

Il n'y a pas de menu scrollable pour sélectionner le type d'exo, on a en revanche mis un tag dans le nom des exo pour pouvoir rechercher par type. On ne peut cependant pas rechercher par langage.

Dans les choses que nous aurions pu faire pour aller plus loin si nous avions eu plus de temps il y a :

- Une refonte de l'interface graphique pour la rendre plus jolie, avec possiblement plusieurs fenêtres ext.

- Nous aurions pu aussi ajouter un éditeur d'exercices directement à l'appli.

- Une piste envisagée mais trop chronophage était aussi déjà de rajouter beaucoup plus d'exo, mais de faire une sorte de progression comme dans un mini rpg, où chaque exo n'aurait pu être débloqué qu'après la validation du précédent, cela aurait inclut de classer arbitrairement les exos selon leur difficulté et de revoir en partie l'aspect graphique du projet.

Documentation

Diagramme de cas d'utilisation

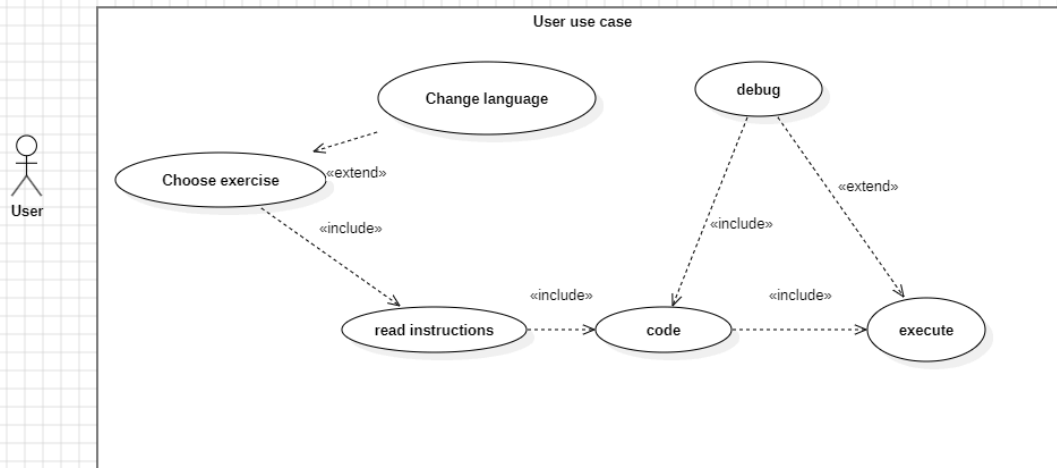
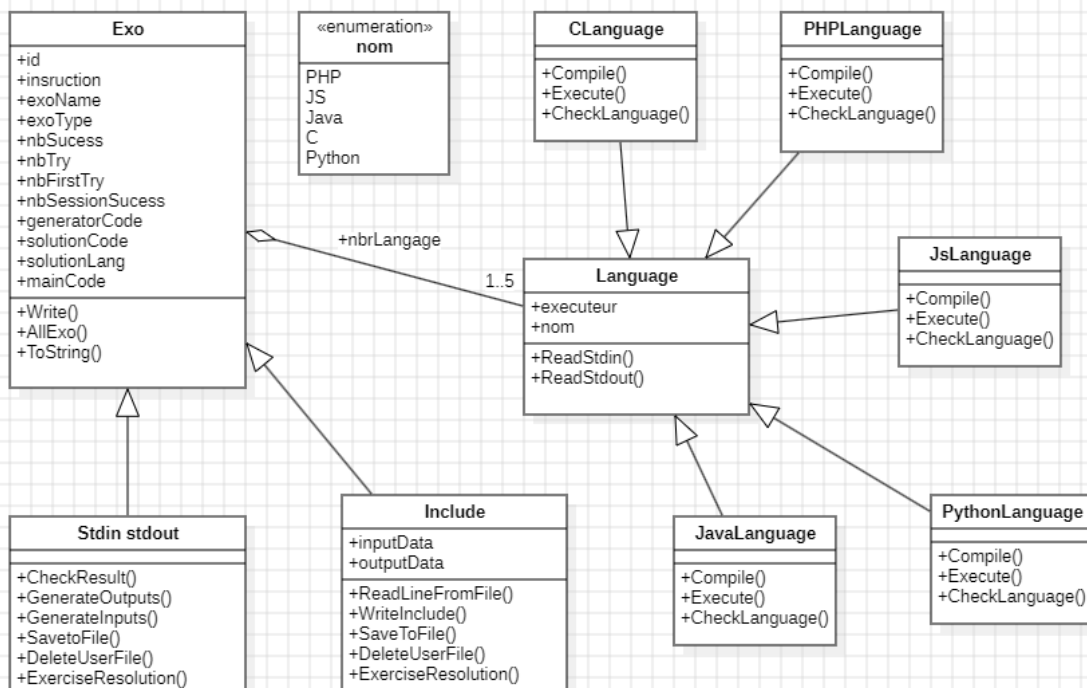


Diagramme de classe



To Do Liste

CodYngame ToDoList

View 1

+ New view

Filter by keyword or by field

Todo 1

This item hasn't been started

Draft

[INTERFACE] Settings

+ Add item

In Progress 0

This is actively being worked on

+ Add item

Done 13

This has been completed

Draft

[BDD] Stockage exercice

Draft

[CERVEAU] Ajouter les compilateur

Draft

[INCLUDE] Format

Draft

[STDIN/STDOUT] Format

Draft

[CERVEAU] Liste exo

Draft

[INTERFACE] Zone Terminale

Draft

[CERVEAU] Give and read data in 5 languages

Draft

[STDIN/STDOUT] Erreur

Draft

...

[\[CMD\] Version console](#)

Draft

[INTERFACE] Zone Principale

Draft

[INTERFACE] Liste exo

Draft

[INTERFACE] Zone Code

Draft

[STDIN/STDOUT] Test

+ Add item

8