Rapport de projet reg7 — itération 1

Gestion d'équipe

Nous avons choisi l'utiliser un outil de gestion de projet standard dans l'industrie, Gitlab. Il inclut les notions d'*issues*, qui sont des tâches à réaliser (bug à corriger, fonctionnalité à implémenter, etc.) assignables à des membres de l'équipe (des *assignees*) et de *milestones*, qui sont des jalons avec une date limite et des *issues* associées; entre autres.

L'avantage de l'outil est qu'il est également une plateforme d'hébergément de dépôts Git, ce qui permet une intégration très pratique entre le code source et les *issues*: par exemple, une branche Git peut être liée à une *merge request*, qui est une demande de fusion du code de cette branche dans la branche principale (`main` dans notre projet). L'acceptation d'une *merge request* ferme automatiquement sont *issue* associée, ce qui la déplace automatiquement dans la colonne "terminée" du *milestone* associé.

Cela permet de travailler efficacement en équipe en utilisant la puissance des algorithmes de fusion de code de Git, tout en faisant de la gestion de projet quasi-automatique, sans se perde en multipliant les outils tiers (Trello, Slack, etc.)

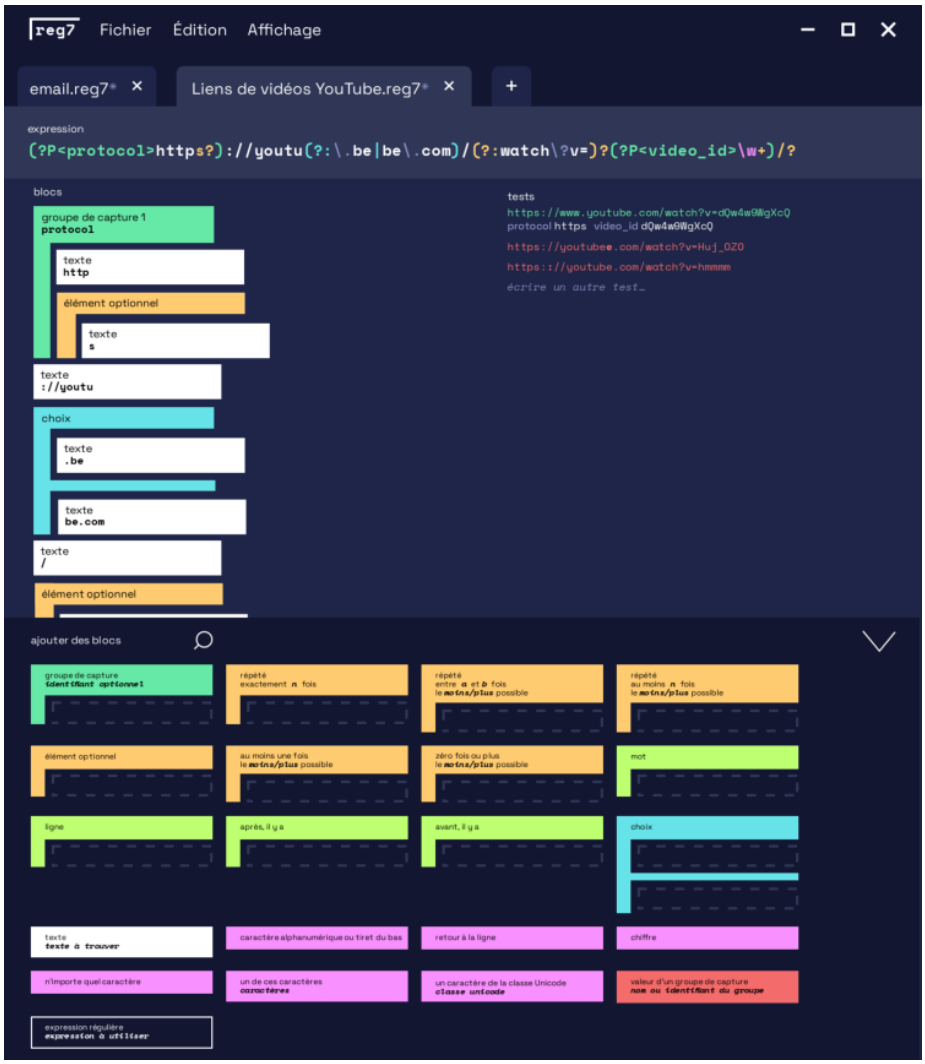
La communication plus générale se fait via un groupe Messenger.

Travail réalisé

Prototype d'interface

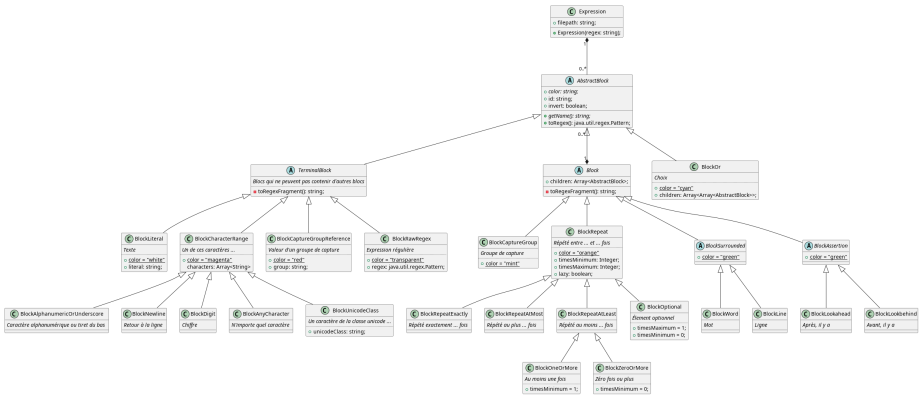
Avant de se lancer, et afin de préciser naturellement les fonctionnalités de l'application, nous avons réalisé un prototype d'interface avec Figma.

Le voici:



Construction d'expressions

L'idée est de représenter l'imbrication des divers blocs comme un arbre de classes. Il a donc fallu écrire un diagramme de classes des expressions.



L'interface permettra ensuite, à travers des boutons et du glisser-déposer, de construire cet arbre. La fonctionnalité d'ouverture d'expression permettra aussi d'analyser une expression sous forme de texte en un arbre.

Nous avons décidé que la construction d'expressions à travers l'interface était prioritaire. L'analyse viendra plus tard.

Le glisser-déposer étant un peu complexe à implémenter correctement avec Swing, nous avons opté pour une première version où les blocs sont ajoutés à la fin de l'expression en cliquant sur des boutons.

Test d'expressions

Nous avons aussi implémenté une première version des tests: on peut donner du texte sur lequel tester l'expression, et observer quelles parties de celui-ci sont *matchés* par l'expression.

Interface

L'interface de cette première version comprend les boutons d'ajouts de blocs en bas, un menu (non fonctionnel) d'ouverture et de sauvegarde d'expressions dans des fichiers, et un panneau de test des expressions.