Apprendre l'algorithmique

Rémi HEITZ Youssef IRHBOULA Antoine HUGUET

Vendredi 22/11/19

Sommaire

- Présentation
- 2 Structuration du projet
- Répartition des tâches

- Présentation
 - Quel est ce projet?
 - Quel est son intérêt?
- Structuration du projet
- Répartition des tâches

Le projet

Á quoi sert ce projet?

Ce projet à pour but d'apprendre à l'utilisateur les bases de l'algorithmique à l'aide d'une interface utilisateur graphique, à l'aspect ludique.

Á qui s'adresse ce projet?

Ce projet s'adresse aux personnes ayant de **aucune à très peu d'expérience** en algorithmique et en programmation.

Nos objectifs

Facilité d'utilisation

Ce programme doit être le plus simple et intuitif possible à utiliser : l'utilisateur doit pouvoir accéder à toutes les fonctionnalités du programme dés sa première utilisation.

User friendly

L'interface doit être agréable d'utilisation, afin d'éviter de dissuader des débutants, peu habitués à l'allure austère des IDE.

Utilisation actuelle

L'objectif

- Appendre les bases de l'algorithmique à des débutants
- Découvrir le concept de variables ou de boucles

Les fonctionnalités

L'utilisateur a à sa disposition de nombreux blocs graphiques parmi lesquels :

- Des boucles WHILE
- Des test IF et ELSE
- Des variables (pouvant être de type Int ou String)
- Des blocs pour effectuer des calculs
- Des blocs pour afficher des résultats

Utilisation future possible

Objectifs futur

- Apprendre à des utilisateurs plus expérimentés des algorithmes usuels
- Résoudre des défis en utilisant des structures de données plus complexes
- Afficher en temps réel l'état des structures de données parcourues

Exemples de structures de données

- Des graphes dont les sommets visités sont affichés au fur et à mesure du parcours
- Des listes dont l'état de tri est affiché en temps réel.

- Présentation
- Structuration du projet
 - Découpage du code
 - Modification de la structure
- Répartition des tâches

Découpage du code

Front end / Back end

Notre projet, de par sa nature, est très fortement découpé entre son *Front end* et son *Back end*, avec des fonctions pour communiquer entre.

Exécution

Le lancement du code est assuré par un fichier main.py, qui récupère les fonctions définies dans les autres modules.

Planification et restructuration

Planification

Les dossiers codeAnalysis/models et GUI/ressources ont été prévus dés le début du projet.

Refactoring

La partie GUI à subi 2 grosses restructuration :

- Pour découper le module en différents sous programme
- Pour faciliter l'exécution du programme depuis main.py

Processus de développement

Processus

- Créations des objets
- Incorporation dans les fonctions principales

Créations des objets

- Réflexion commune sur les besoins en attributs et méthodes
- Création des classes correspondantes

Incorporation

- Discussions sur les besoins pour l'interconnexion Front/Back.
- Ajout des objets

- Présentation
- Structuration du projet
- Répartition des tâches

Répartition des tâches

Front End

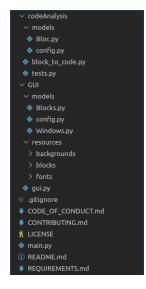
- Création de la fenêtre principale Rémi
- Graphismes Antoine
- Création de la logique d'affichage Rémi
- Ajout de blocs et modules complémentaires Rémi & Antoine
- Refactoring Antoine

Back End

- Lecture et exécution du code Youssef
- Vérification syntaxique et sémantique Youssef
- Tests unitaires Rémi

Annexes

Arborescence



Test

Back end

Un fichier tests.py utilise le module Unittest afin d'effectuer des tests unitaires.

Front end

Les tests unitaires ne sont pas nécessaires car les fonctions graphiques sont *par définition* testable visuellement.