Guide des livrables J2

Ce document décrit les livrables attendus pour J2.

Table des matières

1 Préambule

1.1 Définition d'un livrable applicatif

Chacun des livrables applicatifs A0, A1 et A2 comprend deux parties : l'application proprement dite, sous *src*, et les tests unitaires correspondants, sous *test*.

1.2 Choix du langage de programmation

Si vous écrivez votre application en langage C, la note des rubriques d'évaluation directement liées à la production de code est majorée de 20%. Ces rubriques sont identifiées comme telles dans les grilles d'évaluation fournies.

1.3 Exécution de l'application et des tests unitaires

Pour faciliter le travail de l'équipe et de son mentor, vous créez à la racine du projet un script shell permettant d'exécuter l'application. Il s'appelle `application.sh`, est exécutable, s'exécute depuis la racine du projet et n'utilise que des chemins d'accès relatifs. Il ne nécessite aucun paramètre et ne demande aucune information à l'utilisateur. Si votre application est écrite en Python, il lance sa fonction principale. Si elle est écrite en C, il compile les sources avec gcc et les options `-Wall -Wextra -Werror -Wvla -pedantic -std=c99`, et lance l'exécutable généré.

Vous créez de même un script shell permettant d'enchainer automatiquement tous les tests unitaires. Il s'appelle `test.sh` et a toutes les caractéristiques de `application.sh`.

2 A0 – Application version élémentaire

2.1 Application

La version élémentaire de l'application fournit les fonctionnalités suivantes.

Sujet 1 : L'application évalue une suite d'expressions à deux opérandes réels, y compris négatifs, avec les quatre opérateurs de base + - * /. Elle contrôle que la syntaxe est correcte et affiche un message d'erreur si ce n'est pas le cas.

Sujet 2 : L'application affiche l'état courant du plateau et permet aux quatre joueurs de successivement placer une pièce de type ligne de 1 à 5 carrés. Elle contrôle qu'une pièce placée ne chevauche aucune autre et affiche un message d'erreur si ce n'est pas le cas.

Le manuel utilisateur (livrable T4) n'étant pas encore rédigé, l'application affiche à son lancement 2 à 3 lignes d'aide détaillant les entrées attendues de l'utilisateur. En particulier : la syntaxe d'une expression (sujet 1), la désignation d'une case, d'une pièce et de son orientation (sujet 2), comment quitter proprement l'application (sujets 1 et 2).

2.2 Tests unitaires

Vous développez une fonction de test unitaire pour chaque fonction (Python ou C) importante de votre application. Son but est d'enchainer de façon automatique une batterie de tests couvrant le plus largement possible les cas nominaux, particuliers et erreurs.

Vous pouvez utiliser un framework, comme pytest pour une application écrite en Python, et/ou le support de votre IDE pour faciliter la génération et l'exécution des tests unitaires. Ceux-ci restent cependant exécutables en dehors de votre IDE, par le script `test.sh`.

A0 comprend au moins une fonction de test unitaire.

2.3 Qualité du code

Pour le livrable A0, le code respecte les critères de qualité minimaux suivants : chaque fonction est décrite par un entête de 1 à 3 lignes, le nom des fonctions, paramètres et variables est significatif, les fonctions comptent au plus 40 lignes de code.

Des critères qualité supplémentaires seront introduits dans les livrables applicatifs suivants.

3 T3 – Cahier de validation

Le Cahier de validation est le Cahier de tests de validation (livrable T2) complété des informations suivantes, pour chaque test : (i) la version logicielle sur lequel le test est exécuté, (ii) le résultat du test : « OK » si le test est correct, ou la sortie obtenue (erronée) sinon. La version logicielle est A0, A1 ou A2, ou toute numérotation de version plus fine que vous jugez utile.

Pour simplifier la rédaction du Cahier de validation, (i) vous ajoutez les informations ci-dessus au fichier `cahierTestsValidation.md` existant, typiquement en rajoutant deux colonnes « Version » et « Résultat » aux tests déjà décrits, (ii) vous n'exécutez et ne rendez compte que des tests en rapport avec les fonctionnalités de la version A0. Si vous avez omis des tests de validation à J0, vous pouvez les ajouter à J1.