

**Département Electronique et Informatique Industrielle - 3ème année**

**PROJETS DE LANGAGE C – MODALITES**

**Durée / Début**

Le projet de langage C se déroule durant tout le second semestre de 3<sup>ème</sup> année (S6). Ce projet comporte différentes phases qui s'étaleront sur tous le semestre : les phases de conception, de programmation, et de mise au point et test ; phases auxquelles s'ajoutent la rédaction du rapport et la préparation de l'exposé.

Les étudiants constituent librement leur équipe de projet au sein d'un même groupe de TP (4 ou 5 étudiants par projet). Les sujets attribués aux groupes d'étudiants et la liste des enseignants référents seront transmis le 17 janvier 2020. Le démarrage effectif du projet aura lieu entre le 20 et le 24 janvier 2020. Chaque enseignant responsable d'un projet rencontrera l'équipe d'étudiants concernée par le projet afin de discuter du cahier des charges, dans un créneau qui sera déterminé ultérieurement.

**Gantt Prévisionnel**

Un diagramme de Gantt prévisionnel du projet est exigé en début de projet. Celui-ci a pour but d'inciter les étudiants à anticiper les phases de déroulement du projet, ainsi qu'à penser leur répartition dans le temps, et parmi les membres du groupe.

Le diagramme doit être envoyé au format PDF. Le diagramme peut être accompagné d'une courte description textuelle des tâches qui le composent. L'ensemble des descriptions des tâches ne doit pas dépasser 1 page en plus du diagramme lui-même.

Le rapport est à rendre sur Moodle (<http://moodle.insa-rennes.fr/course/view.php?id=553>) **le vendredi 14 février 2020 avant 23h55. Attention, le serveur se bloquera automatiquement à l'heure donnée, et tout retard entraînera une perte immédiate d'un point sur la note finale du projet.** N'attendez pas la dernière minute pour envoyer votre diagramme, et n'oubliez pas qu'il est possible de remplacer un diagramme précédemment envoyé par un nouveau, plus abouti, dans la limite de la deadline de soumission.

**Utilisation de GitLab et git**

L'hébergement et le partage des projets devront obligatoirement se faire en utilisant la plateforme GitLab proposée par la DSI. L'utilisation de système de gestion de versions git fera l'objet d'un cours et d'un TP dans le cadre du module de « Méthodologie de conception logicielle ».

La bonne utilisation de git selon les « guidelines » données en cours sera prise en compte pour la notation du projet. De plus, une part individualisée de la notation du projet sera basée sur une analyse des logs du repository git utilisé. Il est donc indispensable que tous les membres d'un groupe utilisent git pour contribuer au projet commun et partager leur travail.

Le code pris en compte pour la notation du projet sera celui présent sur la branche master de votre dépôt Git le 26/05 à 14h00. Le code du dernier commit effectué avant la deadline sera pris en compte, et tout commit ultérieur à cette date sera ignoré.

## Réunions

Des rendez-vous entre les étudiants d'un projet et l'enseignant responsable de ce projet auront lieu environ tous les 15 jours dans des créneaux disponibles programmés dans l'emploi du temps. A l'issue de chaque réunion il sera convenu de la date de la réunion suivante.

Chaque réunion d'équipe, avec ou sans l'enseignant, devra donner lieu à un compte-rendu de réunion. Ces compte rendus devront également être stockés sur le projet GitLab, soit sous forme de documents texte/markdown stockés sur le repository git du projet, soit en utilisant le système de wiki intégré à la plateforme.

## Doxygen

Le code du projet doit être commenté en utilisant les commentaires au format doxygen partout où cela est possible (prototypes de fonctions, structures, débuts des fichiers, ...). La présence de commentaires, en générale, et l'utilisation du formatage de commentaires doxygen valides partout où cela est possible seront prises en compte dans la notation du travail réalisé.

## Conception logicielle : Modèle MVC

Le pattern de programmation Modèle-Vue-Contrôleur (MVC) précédemment vu en cours devra être appliqué pour la réalisation du projet. Ce pattern implique une séparation du code dédié à la gestion de l'interface graphique, du code dédié à la modélisation de l'état du programme, et du code de contrôle. En particulier, des fichiers de codes et des structures de données distincts devront être utilisés pour chacune des trois parties du programme.

La bonne utilisation du modèle MVC sera prise en compte dans la note de travail.

## Tests unitaires et couverture de code

Des tests unitaires doivent être réalisés sur la partie modèle du projet. La couverture de code de ces tests unitaires sur la partie modèle doit atteindre les 100%. La réalisation de ces tests sera prise en compte dans la note de travail.

Ces tests peuvent être réalisés avec les outils utilisés en TP : CMocka et gcov, ou avec d'autre outils équivalents. Si un autre outil de test unitaire est utilisé, il devra impérativement respecter le framework xUnit.

De manière facultative, l'automatisation de ces tests unitaires en utilisant les fonctionnalités d'intégration continue (CI) de GitLab donnera lieu à un bonus intégré dans la note finale du projet.

L'outil Valgrind vu en TP de MCL devra également être utilisé afin de vérifier l'absence de fuite mémoire dans le programme réalisé.

## (Optionnel) Utilisation d'un CMake

CMake est un outil facilitant la création d'un projet de langage C multi-IDE et multi-OS. A la manière d'un Makefile, CMake permet pour un projet C/C++ de spécifier "simplement", par un fichier de configuration textuel, l'ensemble des règles de compilations du code source, ainsi que les dépendances vers les différentes bibliothèques utilisées dans le projet. A partir de ce fichier de configuration, il est possible de générer automatiquement des projets pour différentes architectures (x86 32bit, x86 64bit, Arm, ...), pour différents OS (Linux, Windows, ...) et pour différents IDE/outils de compilations (Codeblocks, Visual Studio, Makefile, ...).

La réalisation (optionnelle) d'un projet CMake pour le projet de langage C donnera lieu à un bonus intégré dans la note finale du projet.

## Environnement de développement

Libre (Win, Linux, Mac OS)

## Salles

La mise au point des programmes pourra se faire dans les salles suivantes :

- Salles MULTI (Bât. 5) en libre-service.
- Salles TP EII, contactez Samir Keddar à l'avance  
(Samir.Keddar@insa-rennes.fr.)

## Rapport final et exposé

Un rapport succinct (5 pages maximum) sera exigé et donnera lieu à un exposé devant 2 enseignants, dont l'enseignant responsable, et les étudiants intéressés.

Le rapport devra faire ressortir les aspects suivants :

- objectifs
- développement (choix techniques, conception logicielle, ...)
- méthodologie (organisation du projet, répartition du travail, ...)
- résultats (présentation du résultat, adéquation au cahier des charges, perspectives)

Le rapport doit être envoyé au format PDF. Les paragraphes de textes (intro, corps, et conclusion), les illustrations, et les annexes, s'il y en a, sont comptabilisés dans les 5 pages. Les pages de couverture, la page de garde, et le sommaire du rapport ne sont pas comptabilisés dans les 5 pages. Le template à utiliser pour le rapport est le template officiel INSA disponible sur l'intranet :

- Word : [http://intranet.insa-rennes.fr/fileadmin/Communication/Modeles-WORD/2016/Modele-WORD\\_FORMATION.docx](http://intranet.insa-rennes.fr/fileadmin/Communication/Modeles-WORD/2016/Modele-WORD_FORMATION.docx)
- Latex : [http://intranet.insa-rennes.fr/fileadmin/ressources\\_intranet/Communication/Modele\\_PFE/Modele-PFE\\_LaTeX.zip](http://intranet.insa-rennes.fr/fileadmin/ressources_intranet/Communication/Modele_PFE/Modele-PFE_LaTeX.zip)

Ni la taille de police utilisée (11 pt) ni les marges du document ne doivent être modifiées !

Le rapport sera à rendre sur Moodle (<http://coursonline.insa-rennes.fr/course/view.php?id=553>) le **jeudi 21 mai 2020 avant 23h55**. **Attention, le serveur se bloquera automatiquement à l'heure donnée, et tout retard entraînera une perte immédiate d'un point sur la note finale du projet.** N'attendez pas la dernière minute pour envoyer votre rapport, et n'oubliez pas qu'il est possible de remplacer un rapport précédemment envoyé par un nouveau, plus abouti, dans la limite de la deadline de soumission.

L'exposé (20 minutes + questions pour les groupes de 4, 25 minutes + questions pour les groupes de 5) devra présenter une synthèse pertinente du rapport ainsi qu'une démonstration commentée. **Ces exposés auront lieu le 25 et 26 mai 2020.**

## (Optionnel) Vidéo Promotionnelle

De manière facultative, une courte vidéo (~1 à 2 minutes) faisant la « promotion » du logiciel développé durant le projet pourra être rendue. L'idée de cette vidéo est de présenter de manière dynamique et attrayante le résultat de votre travail, pour donner envie à des utilisateurs/joueurs de l'essayer. Cette vidéo devra faire la part-belle aux captures vidéos d'une démonstration de la version finale de votre projet afin d'en apprécier les différentes fonctionnalités. Pour une meilleure compréhension, des textes en surimpression pourront être utilisés. Cette vidéo pourra être utilisée lors d'événements de promotion du département (JPO, ADS), ou à titre d'exemple de projet de langage C pour les promotions suivantes.

La réalisation (optionnelle) de cette vidéo donnera lieu à un bonus intégré dans la note finale du projet. La vidéo devra être uploadée via <https://filesender.renater.fr/>, et le lien transmis par e-mail à [kdesnos@insa-rennes.fr](mailto:kdesnos@insa-rennes.fr) **au plus tard le vendredi 29 mai 2020.**

## **Evaluation**

Le projet sera noté sur 20 (note collective + part individuelle).

L'évaluation prendra en considération :

- Travail, noté sur 10 points (modulation individuelle possible)
  - Travail fourni
  - Gestion de projet
  - MVC
  - Git
  - Doxygen
  - Tests unitaires & couverture de code
  - (Optionnel) Intégration Continue
  - (Optionnel) CMake
- Rapport, noté sur 5,
- Exposé, noté sur 5.