

ISIMA – ZZ2 F2 et F3 - Architectures Logicielles et Qualité

Etude de cas – Réalisation d'un petit logiciel

La mise en œuvre d'un petit logiciel se réalise en binôme ou trinôme il donne lieu à une documentation + un code. Pour la filière 2, il s'agit de réaliser un code de simulation multi-agents (SMA : intelligence artificielle distribuée sur un domaine de leur choix). Pour la filière 3, même si un sujet de multi-agent est possible en l'orientant sur les négociations et les échanges socio-économiques, il vous sera peut-être plus simple de choisir de développer un petit code en lien avec la Business Intelligence (cela peut-être une fonctionnalité Web pour un tableau de bord, un outil pour du CRM – Customer Relationship Management – un outil pour du reporting, ou tout autre outil de votre choix.)

Le choix spécifique du sujet est de votre ressort ainsi que le choix du langage et de l'environnement de développement. C'est un choix qui peut donc être en lien avec vos goûts, votre projet professionnel, vos sujets de stage. Cette autonomie forte fait partie de la difficulté, elle a l'avantage de vous préparer au stage et d'augmenter votre capacité à trouver comment vous dépanner.

Rendu : 70% de la note finale (le 1^{er} projet sur 30% vous aide à obtenir des points plus « facilement »).

Date de rendu : Dimanche 07/01/2024 à 23h, par mail. Les retards seront pris en compte dans la notation.

Un rapport final (10 points). Le rapport peut être composé de deux documents :

« Prévisionnel » :

- Un cahier des charges : *Le cahier des charges est un document réalisé en amont du projet. Il décrit le contexte du projet, les besoins fonctionnels, les objectifs, les paramètres, la sécurité, les normes à respecter, etc. Le cahier des charges peut être général et vague. Il donne l'esprit de ce qui doit être réalisé, sans nécessairement entrer dans les détails. Etant réalisé en amont, le cahier des charges est fourni au prestataire et c'est sur la base de celui-ci que le contrat est réalisé : budget, deadline, contraintes.* Présentation du projet, Objectif, diagramme de cas d'utilisation, spécifications fonctionnelles, ... Il ne faudra pas non plus faire quelque chose d'extrêmement poussé. Des exemples de cahier des charges sont fournis sur l'Ent. Le cahier des charges permet de décrire le projet.
- Une liste de tâches prévisionnelle, associées à une durée. (A mettre dans un tableau, comme pour faire un diagramme de PERT par exemple).

« Réel » :

- Un diagramme de classe d'implémentation/conception UML (représentation de la solution réelle, différent du diagramme de classe d'analyse/modèle du domaine. La conception propose une solution – le COMMENT – l'analyse n'expose que le contexte du domaine et les objectifs de votre application – le QUOI).
- Un diagramme de Gantt réel (Pas prévisionnel). L'objectif est de voir comment les tâches ont été découpées entre les membres du projet, ce qui a pris plus de temps ou moins de temps que prévu. Vous pouvez utiliser l'outil « GanttProject » pour réaliser votre diagramme de Gantt final. Vous commenterez ces différences dans votre rapport.
- Une présentation succincte des outils que vous avez choisi d'utiliser, et la manière dont vous les avez utilisés dans le projet. La liste donnée ci-dessous n'est pas exhaustive, vous pouvez utiliser un outil de développement que vous aimez. (Pour les binômes, Git + Make/Maven sont classique par exemple et ne vous demanderont pas de travail supplémentaire dans la mise en place, puisqu'ils sont de toutes façon presque indispensable. Pour les trinômes, vous devez donc ajouter un outil supplémentaire : Valgrind pour l'étude des fuites mémoires, gdb pour débbugger, Doxygen pour générer de la documentation automatique, Junit dans le cadre d'un Test Driven Development, ...)

Restez modestes sur les objectifs, c'est un TP long mais pas un projet. La réalisation de cette petite étude de cas se fera sur environ 2 mois.

L'application (10 points) :

Un langage à Objet est vivement conseillé, mais pas obligatoire dans un contexte de développement Web. Les contraintes pour la conception et l'implémentation sont les suivantes :

- Utilisation autant que possible de 3 patrons de conception. Voici 3 exemples de patrons candidats pour ceux qui font un SMA : le patron composite (groupe d'agent), le patron stratégie (comportement d'agents), les patrons MVC/Observer, state et singleton peuvent être aussi être considérés ainsi que d'autres pour des développements Web.
- Vous utiliserez au moins 2 autres outils du génie logiciel pour ce TP (1 par personne dans le cas d'un binôme), 3 outils dans le cas d'un trinôme (1 outil par personne). Chaque personne choisit un outil qu'elle a envie de découvrir ou d'approfondir, cet outil est mis en œuvre sur l'étude de cas logicielle.
- Obligation d'utiliser Git, avec un repository Github/GitLab (GitLab est disponible à l'ISIMA).

Si vous avez envie, vous pouvez mettre en place un type de gestion de projet agile : Test Driven Development. Dans ce cas vous pourrez donner un pourcentage de couverture de test du logiciel.

Attention : éviter les copier/coller pour les présentations d'outils, le code et citer toutes vos sources (URL). L'université dispose de l'outil « compilatio » pour l'anti-plagiat, il est très efficace.

Pour les outils à utiliser, voici une liste non exhaustive (à rafraîchir...) – un outil à utiliser par personne du groupe.

Gestion de projets

make
cmake
Ant
Maven – (Apache Software Foundation)
Utilise un paradigme connu sous le nom de Project Object Model (POM) pour décrire un projet logiciel et ses dépendances avec des modules / bibliothèque externes
...

Gestion des versions

Git,
GitLab,
CVS (ancêtre),
Tortoise CVS
SVN, Subversion
Mercury

Etude des performances

Prof / Gprof
Jprofiler
AQtime
Sonar Cube,
Valgrind
...

Etude des tests

CppUnit, Junit,
, Kcachegrind,

...

Logger de traces d'exécution

Log4J...

Production de documentation

Doxygen, Javadoc

...

Etude d'un débogueur

gdb – ddd, dbx – dbxtool

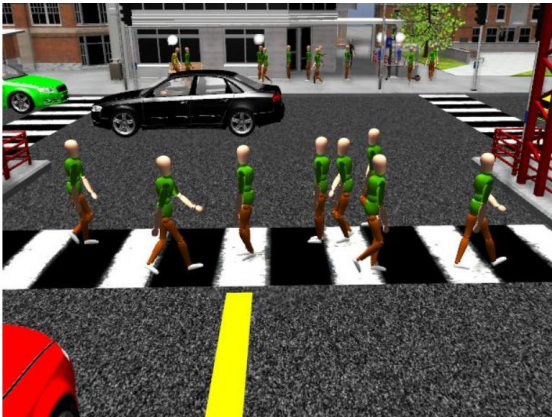
Eclipse débogueur, .Net débogueur

...

Intégration continue :

Jenkins, Bamboo, Cruise Control, CodeShip, TeamCity
Travis CI, GitLab CI, Circle CI

SMA



BI

