

## Consignes pour le compte-rendu et la démonstration finale du mini-projet

### 1. Note concernant le plagiat

**Rappel : le plagiat consiste à faire passer pour sien des travaux d'autrui. Le plagiat à l'université est passible de sanctions disciplinaires pouvant aller jusqu'à l'exclusion définitive de tout établissement d'enseignement supérieur.**

Les documents ou travaux en question peuvent être de toutes sortes : compte-rendus, code source, scripts, fichiers de test.

Il vous est demandé de présenter vos travaux, dans le but qu'ils soient évalués. Il est important que cette évaluation porte exactement sur ce que **vous** avez fait (et non ce qu'a fait un autre étudiant ou binôme).

**Il n'y a pas plagiat si l'auteur (ou la source) est citée explicitement pour toute la partie copiée.** Il est acceptable dans le cadre d'un travail complexe de s'appuyer sur les travaux des autres : il faut alors mentionner **précisément** ce qui a été récupéré, et à qui (par exemple, «pour cette partie nous nous sommes basés sur le paquetage X du binôme Y»); et ce **avant** que l'enseignant-e ne pose la question de la provenance du fichier présenté. Dans ce cas la partie «récupérée» n'est simplement pas évaluée.

**NB :** l'université fournit des outils anti-plagiat, qui sont habituellement utilisés par les enseignant-e-s.

### 2. Consignes générales

Le but de cette démonstration est de mettre en valeur votre travail. Par conséquent elle doit être soigneusement préparée. Préparez un synopsis (un plan), préparez les fichiers de données.

Répartissez-vous les rôles : un des membres du binôme est au clavier, l'autre parle.

Faites une ou plusieurs répétitions.

Préparez les documents à rendre.

**Important :** conservez toujours une version de votre travail qui fonctionne, attention aux modifications de dernière minute!

### 3. Document à préparer et à déposer avant la soutenance

**Ce document est à déposer sur Moodle avant la soutenance, au format PDF. Il ne doit pas dépasser 2 pages. Vous devez apporter une version imprimée au moment de la soutenance.**

Ce document doit comporter :

- pour le TP7 : une explication de la démarche suivie pour écrire les tests;
- pour le TP8 : la description des deux programme-robots écrits et évalués aux exercices 4 et 6; le résultat statistique de l'évaluation de ces programmes-robots sur un ensemble de terrains générés de manière aléatoire;
- pour le TP9 : la description de l'observateur défini, sa traduction en automate d'états fini.

Enfin, vous commenterez les résultats obtenus en une brève conclusion de quelques lignes.

**De plus, il vous est demandé de déposer sur Moodle une archive contenant votre code source, ainsi que tous les tests écrits pendant le projet.**

## 4. Déroulement de la démonstration

La démonstration se déroulera ainsi : 5 minutes de présentation suivies de 5 minutes de questions.

Dans votre présentation, vous devrez montrer au minimum les points suivants :

- validation de votre *Makefile* : nettoyage du répertoire (`make clean`) et refabrication des exécutable ;
- présenter les tests écrits, montrer l'exécution du programme `curiosity-test` sur un des tests ;
- présenter la méthode et le programme de génération d'un ensemble de terrains dans un fichier ;
- comparaison de 2 programmes-robots «infinis» :
  - sur des ensembles de terrains générés (plusieurs ensembles de caractéristiques différentes) production des résultats permettant de comparer les performances des 2 programmes-robots, et commentaire des résultats ;
  - test de ces programmes-robots sur des terrains spécifiquement créés pour chacun d'eux : 1 terrain qui valide la stratégie mise en œuvre par ce programme-robot, et 1 terrain qui montre ses limites (duquel il ne sort pas) ;
- présentation de l'observateur défini au TP9.

## 5. Évaluation

L'évaluation du mini-projet portera sur les points suivants :

- Makefile : le fonctionnement est-il correct ? Les programmes compilent-ils sans erreur ?
- Exécution des programmes : les programmes présentés fonctionnent-ils correctement ?
- Méthode de génération de terrains : la méthode est-elle correcte et clairement présentée ?
- Démarche de tests : la démarche est-elle clairement présentée ? Les tests écrits couvrent-ils une bonne partie de l'ensemble des cas possibles ?
- Pour chaque programme-robot présenté :
  - pertinence des explications concernant la construction de ce programme
  - pertinence des terrains construits pour tester ce programme
  - présentation et clarté des résultats
- Observateur :
  - pertinence et clarté des explications concernant l'observateur décrit
  - correction de l'automate par rapport aux explications
  - démonstration de l'observateur sur des exemples
- Démonstration :
  - Préparation
  - Enchaînement
  - Clarté