FACULTÉ DES SCIENCES ET GÉNIE

Projet RoboCup
Équipe I.A.
2015-2016

Compétition de programmation – hiver 2016

Énoncé de la compétition

29 décembre 2015

Équipe I.A.

Contents

1 Introduction

La cinquième édition du DA IEEE CODE se déroulera le vendredi 12 mars entre 13h30 et 17h30 au local 0103 du pavillon Adrien Poulliot. Cette compétition est organisée par

l'équipe en intelligence artificielle du projet RoboCup ULaval en partenariat avec la branche étudiante de l'IEEE.

Cette compétition de programmation prendra la forme d'une compétition en intelligence artificielle. Les détails du défi et de la tâche à accomplir seront expliqués en détails au début de la compétition à 13h30.

La compétition s'effectue en équipe de deux. Afin d'être déclaré vainqueur, une équipe devra résoudre trois défis et marquer le maximum de points selon les barèmes expliqué plus loin.

Le présent document présente les règles de fonctionnement de la compétition ainsi que son déroulement.

2 Règlements

2.1 Généralités

- Tous les règlements se doivent d'être appliqués dans le but unique que chaque participant ait un maximum de plaisir ainsi qu'une chance égale de gagner. Les juges feront appliquer les règlements dans la plus grande convivialité possible

- Tous les documents de la compétition seront dans la langue française. Il est toutefois permis à un compétiteur unilingue anglophone d'employer l'anglais dans son livrable sur demande.

- Le masculin est utilisé de façon générique pour simplifier le texte ; femmes et hommes sont également acceptés pour la compétition!

2.2 Composition des équipes

- Chaque équipe est constituée d'un étudiant en 1ère ou 2e année de son programme d'études ainsi que d'un second étudiant de son programme d'études, préférablement de 3e ou 4e année. Pour les personnes au cheminement particulier, un 1ère année est quelqu'un qui possède moins de 30 crédits; un 2e année quelqu'un qui a moins de 60 crédits ; un 3e année quelqu'un qui a moins de 90 crédits et un 4e année quelqu'un qui a moins de 120 crédits.

- Les étudiants doivent être inscrits dans un programme de premier cycle de génie électrique, génie informatique, génie logiciel, informatique, ou d'un programme de premier cycle avec suffisamment de similarité avec un de ces derniers.

2.3 Inscription

- Les équipes de deux doivent s'être inscrites pour la compétition via le site ieee.gel.ulaval.ca.

- Les participants acceptent d'être pris en photos durant la compétition. Ces photos pourront être employées à des fins de promotions et publicité pour l'évènement, la branche étudiante de l'IEEE de l'Université Laval ainsi que pour le projet étudiant ROBOCUP ULAVAL.

- Les gagnants seront photographiés lors de la remise des prix. Ces photographies et leur nom, prénom et programme d'étude seront employés à des fins de promotion et publicités pour l'évènement, la branche étudiante de l'IEEE de l'Université Laval et le projet étudiant ROBOCUP ULAVAL.

- Le nombre d'équipes maximales est limité par les dimensions de la salle, veuillez-vous inscrire dès que possible. Premier arrivé (et qualifié) premier servi.

- Des inscriptions individuelles sont acceptées, ces derniers seront mis en équipe sur place si nécessaire. Nous ne garantissons pas qu'il soit possible de former des équipes avec les inscrits individuels et ceux-ci doivent accepter le fait qu'ils risquent de ne pas pouvoir prendre part à la compétition.

- Puisque, pour la compétition, l'équipe ROBOCUP ULAVAL fournit son engin de simulation et ses outils de développement et afin d'être équitable pour tous, toute personne membre du projet ne peut s'inscrire à la compétition.

2.4 Équipement, ressources permises et nécessaires

- On permet aux compétiteurs d'apporter leur propre ordinateur portable. Le programme devrait tout de même pouvoir s'exécuter sur une station de bureau présente au laboratoire.

- Des stations de bureau sont aussis fournies en nombre limité.

- Internet, l'électricité, la lumière, des bureaux et chaises, ainsi que l'oxygène sont fournis sans frais aux participants.

- Les programmes devront être implémentés en python 3.4 ou dans une version supérieure.

- Les participants sont libres d'utiliser l'environnement de développement de leur choix. Toutefois, ils doivent être capable d'exécuter leur programme en ligne de commande afin de s'assurer que la correction se fasse de manière appropriée.

2.5 Déroulement de la compétition

- La durée de la compétition est limitée à quatre heures, débutant à 13h30 le 12 mars, et se terminant à 17h30 sans faute.

- Au tout début de la compétition, les équipes reçoivent le présent document contenant les règles. Ils reçoivent également le lien vers le dossier comprenant les défis. Un fichier par défi permet d'avoir les détails du programme à effectuer.

- Les présences seront prises à plusieurs reprises durant l'évènement. Des ordinateurs de bureau seront mis à la disposition des compétiteurs, ainsi qu'Internet, des prises électriques, chaises, bureaux, etc.

- Des breuvages et collations seront fournis.

2.6 Remise et évaluation

- Les livrables seront évalués par un comité composé d'un ou plusieurs juges. La validité de la réponse pour chaque problème sera évaluée ainsi que la qualité du code(documentation, efficacité, etc). Pour plus de détails veuillez vous référer aux barèmes propres à chaque défi.

- Le livrable final est un fichier ZIP contenant un dossier par défi. Les dossiers doivent être clairement identifiés par leur numéro et tous les noms de fichiers doivent contenir des caractères ASCII exclusivement. Chaque dossier doit contenir le code source du programme réalisé.

- Il vous est possible de faire évaluer chaque épreuve séparément durant la durée de la compétition. Si votre algorithme accomplit sa tâche avec succès et de façon satisfaisante l'épreuve sera considérée comme réussie. Pour plus d'informations, référez-vous aux barèmes propres à chaque épreuve.

- Dans le cas d'une épreuve échouée, vous devrez attendre un minimum de 10 minutes avant de pouvoir demander une recorrection.

2.7 Triche, plagiat et disqualification

- Il est strictement interdit de copier du code trouvé sur internet.

- L'application stricte du règlement 2.7.1 résulterait en une compétition de programmation à toutes fins pratiques impossible. Voici donc des lignes directrices en ce qui concerne sont application :

    1. Copie d'appels de fonctions et d'exemples provenant de sites de référence sur les langages de programmation : permis (et encouragé).
    2. Copie de petits bouts de code (d'un nombre de lignes se comptant sur les doigts de la main) trouvés sur StackOverflow ou un site similaire : toléré, sujet à interprétation.
    3. Copie de paragraphes entiers de code trouvés sur GitHub, Google Code, SourceForge ou un site similaire : interdit.
    4. Copie de paragraphes entiers de code, en changeant les noms de variables et en déplaçant un peu le code afin de camoufler sa provenance : interdit.
    5. « S'inspirer » de code déjà existant pour produire son programme, au lieu de développer ses propres algorithmes : interdit.

    Le jury a le dernier mot quant à l'application de cette règle, et ses décisions sont finales et sans appel. Des règles similaires à celle du plagiat dans les travaux universitaires, ainsi que le gros bon sens seront appliquées.

En contrepartie, l'honnêteté de tous les participants est demandée. Il s'agit d'une compétition de programmation, pas une compétition de copier-coller. Ce n'est pas un cours, vous ne risquez pas l'échec. Des points seront quand même donnés si votre code ne s'exécute pas parfaitement: voir les barèmes propre à chaque défi.

- Un participant violant un ou plusieurs des présents règlements verra automatiquement son équipe disqualifiée. Les décisions du jury à cet égard sont finales et sans appel.

2.8 Remise des prix

Les prix et bénéfices de la compétition sont les suivants :

1. L'équipe en première position recevra un prix de 250$
2. L'équipe en seconde position recevra un prix de 150$
3. L'équipe en troisième position recevra un prix de 100$

Quelques règles concernant les prix :

1. La liste des gagnants sera mise sur le site de l'IEEE `http://ieee.gel.ulaval.ca`.
2. Les prix des équipes gagnates ne seront remis qu'au Vins et Fromages 2016 de l'IEEE
3. S'il s'avère impossible à un gagnant d'être présent, il aura 14 jours pour réclarmer sont prix à compter de la date du Vins et Fromages 2016 de l'IEEE. Après quoi, son prix est abandonné et ne sera pas remis à quiconque.

2.9 Vous avez des questions ?

Pour toutes questions ou clarifications, écrivez à julien.morissette.1@ulaval.ca.

Les présents règlements seront interprétés par le jury-organisateur, leurs décisions et interprétations seront finales et sans appel.

3 Organisation

La compétition aura lieu pendant une durée de 4h en équipe de 2. Une configuration "minimale" sera exigée; c'est-à-dire que l'outil *GrSim* sera fourni au préalable avec une stratégie. Si celle-ci ne s'exécute pas correctement, il sera déconseillé de participer. Il est conseillé aux participants de se présenter une demi-heure à l'avance, soit à 13h00, afin de vérifier que leur environnement est fonctionnel.

Pendant la compétition, chaque participant pourra faire corriger l'épreuve sur laquelle il travaille. Cinq tentatives sont allouées. Parmis celle-ci, l'algorithme devra accomplir sa tâche un minimum de trois fois.

Après un délai de 10 minutes, un participant peut demander une recorrection. Lors d'une recorrection, une équipe ne pourra obtenir, au maximum, plus de la moitié des points de l'épreuve.

4 Barême

La correction d'une épreuve est directe. Lorsque vous êtes prêt, vous pouvez l'indiquer à un des organisateurs et il évaluera votre programme. Voici le barème général :

- Toutes les tentatives fonctionnent:
  L'ensemble des points de l'épreuve sont gagnés.
- Trois tentatives ou plus ont réussies
  La moitié des points sont gagnés. (nombre suffisant de tentative pour se qualifier un crédit supplémentaire)
- Moins de trois tentatives
  Aucun point n'est gagné.

Une tentative est considérée réussie si l'objectif est atteint. Si l'objectif nécessite un déplacement à une coordonnée, la position de *GrSim* est considérée avec une incertitude de ± 90. Si un problème technique est rencontré, *e.g* l'exécution de l'algorithme échoue, la tentative est considérée ratée. Les cinq tentatives doivent s'exécuter en moins de 5 minutes. Tout dépassement de ce temps est considéré comme un échec pour l'épreuve.

Pour plus de détails sur l'évaluation de chaque épreuve veuillez vous référer à son énoncé directement.

5 Épreuves

Chaque épreuve est présentée de la même façon. En premier lieu, une brève description est fournie. Ensuite, la liste des outils utiles et disponibles est fournie.

Dans la section susbséquente, un exemple de code et son explication sont donnés. Cet exemple de code résout un problème similaire ou simplifié de l'épreuve. Il illustre aussi comment utiliser les outils.

Ensuite les objectifs précis sont énoncés. Il s'agit des élements à respecter pour passer l'épreuve.

Finalement, quelques indices et pistes sont suggérés. Chaque épreuve déclare sa pondération et le fonctionnement de la correction.

De plus, des crédits supplémentaires peuvent être accordés.

5.1 Déplacement point à point avec obstacles

Description

Votre premier objectif sera de prendre un robot et de le déplacer jusqu'à son objectif. Des obstacles se trouveront sur votre chemin et vous devrez les évitez.

L'épreuve se divise en deux parties:

- D'abord, les obstacles seront statiques. Vous aurez trois essais différents à réaliser. Cet partie vaut 2 points et vous ferez 0.66 points par essais réussi.
- Par la suite, les obstacles ne seront plus statiques. Vous aurez trois scénarios différents à réussir et 2 essais par scénario. Chaque essai réussi vous donnera 0.5 points pour un total de 3 points.

Outils

- Une fonction qui permet de détecter une collision.
- Une fonction qui permet de récupérer la position d'un obstacle – ou de tous les obstacles.
- Une fonction pour déplacer à un point précis le robot.

Exemple de code

Ce code appel la fonction pour récupérer la position de tous les obstacles. Ensuite, il tente de déplacer le robot, avant de ce faire, il utilise la fonction pour vérifier si le déplacement cause une collision. Par la suite, le code exécute une boucle jusqu'à ce que la fonction retourne l'événement **succeed**.

Objectif

Le robot commence au point $(0, 0)$ et devra atteindre le point $(900, 842)$. Une incertitude de ± 90 est tolérée.

Indice

Le but de l'épreuve est de vous familiariser avec le système de stratégie.

5.2 Défense avec un gardien et un défenseur contre deux attaquants

Description

Votre deuxième défi est de programmer un gardien de but. Votre gardien devra être en mesure de bloquer des tirs face à diverses stratégies. Votre gardien devra faire face à un total de 5 scénarios différents. Pour chaque scénario vous aurez 2 essais et pour chaque tir bloqué vous obtiendrez 0.5 points.

Valeur: 5 points en tout, soit: 1 point pour une tentative bloquée, 3 points pour 2 tentatives bloquées et tous les points pour les 3 tentatives bloquées.

Outils

- Une fonction pour placer en interception un joueur vis-à-vis un autre joueur.
- Une fonction pour positionner un joueur entre deux autres joueurs.

Exemple de code

$coach.positionner\_entre\_deux\_ennemis(joueur,\ ennemi1,\ ennemi2,\ cible)$

joueur, ennemi1 et ennemi2 sont des index de joueurs. La cicle est optionnelle. C'est la direction vers laquelle le joueur doit regarder.

$coach.positionner\_entre\_ami\_et\_ennemi(joueur,\ ami,\ ennemi,\ cible)$

joueur, ami et ennemi sont des index de joueurs. La cicle est optionnelle. C'est la direction vers laquelle le joueur doit regarder.

Les attaquants vont tenter de se faire une passe et de botter. Une commande est envoyée pour placer le défenseur entre les deux attaquants. Une commande est envoyée pour placer le gardien entre le but et l'attaquant en posession de la balle.

Objectif

Bloquer chaque tentative d'attaque: ne pas allouer de buts. La pondération est dépendante du nombre de buts alloués.

5.3 Attaque sur un gardien avec deux joueurs

Description

Votre troisième défi est de programmer deux robots pour former une attaque contre un gardien de but.Vous disposerez de 5 essaies pour compter un maximum de 5 buts. Chaque buts réussit vous donnera 1 point.

Voici quelques détails supplémentaires :

- Si un de vos robots entre en contact avec le gardien adverse, l'essaie se termine et vous ne pouvez pas obtenir votre point pour celui-ci.
- Si la balle sort du terrain ou va dans votre zone déffensive l'essaie se termine et vous ne pouvez pas obtenir votre point pour celui-ci.
- Aussitôt que la balle entre dans la zone de but, cela constitue un essaie. Que vous marquiez ou non, votre essaie se termine.
- Si le gardien parvient à toucher la balle et la botte en dehors du terrain ou encore dans votre zone deffensive, l'essaie se termine.

Outils

- Une fonction pour passer la balle d'un robot à un autre.
- Une fonction pour botter.
- Une fonction pour prendre possession de la balle.

Exemple de code

$coach.passe(joueur1,\ joueur2,\ force)$

joueur1 et joueur2 sont les index des joueurs sélectionnés. La force détermine la puissance avec laquelle la balle sera envoyée. Il est préférable que la force soit inférieure à 5.

$coach.lancer(joueur,\ cible,\ force)$

joueur est l'index d'un des joueurs de l'équipe. cible est une valeur Position(). La force est optionnelle, par défaut sa valeur est 5.

coach.chercher_balle(joueur)

joueur est l'index d'un joueur de l'équipe.

Le robot commence avec la balle sur son dribbleur. Une commande pour faire avancer le robot en position est envoyée. Lorsque la commande est terminée, une commande pour botter la balle est envoyée. Le gardien est statique dans cet exemple.

Objectif

Marquer un but.

Indice

La fonction pour détecter les collisions pourrait être utile.

Crédits supplémentaires

Effectuer une passe avant de marquer le but. (+1)