DO AMARAL Daniel

EL MOKHTAR Ahmed

**Module IAS – S9**

Projet - Conception et développement d'un PNJ avec Pogamut

Dans le cadre du module Intelligence Artificielle et Simulation à l’ENIB, c’est proposé aux élevés comme projet, le développement d’un personnage non joueur (bot) en utilisant le framework *Pogamut* et aussi la langage Java. Dans ce fichier nous présenterons notre projet de façon concise, avec les exigences minimales implémentés et les techniques supplémentaires utilisés dans le développement. Le projet était fondamentalement divisé en deux parties, l’automate à états pour définir le comportement du bot dans chaque état et la choix d’une des techniques de IA vu en cours pour améliorer le bot, en laissant le personnage plus proche d’un personnage réel.

* **Étapes Réalisés**

**Partie 1 – Automate à états**

La création de notre automate était basée sur la création de x états. (Compléter). Chaque état est représenté par l’état actuel du bot et par son prochaine état (état suivant). Cette définition laisse le bot toujours avec un état à suivre et évite des boucle indésirés.

[Compléter avec description de la machine - Ahmed]

**Partie 2– Choix de la technique de l’IA étudie dans le cadre du projet**

Nous avons utilisé la programmation logique comme technique dans le projet. Cette type de programmation est liée à déclaration de sentences que iront exprimer des faits et des règles par rapport à un problème.

Pour la création de l’automate à états, nous avons proposé l’utilisation de la langage *Prolog* pour l’implémenter, qu’est une langage de programmation logique bien utilisé dans le contexte de l’IA. La fonction “logic”, responsable pour le comportement autonome du bot, réalise une appel au fichier Prolog avec tous les définitions des états d’automate. Cette choix a été fait parce que, en première, les règles de syntaxe et la sémantique de la langage sont vraiment simples et considérées comme claires. Les codes Prolog sont surtout basées sur des faits et des règles et c’est exactement ça que nous avons besoin pour décider des prises de décision du bot . Le deuxième point c’est que le code de la machine à états a été mis dans une fichier prolog séparé (fichier ES.pl), que a donnée plus lisibilité au code du fichier principale que a déjà des autres définitions.

Nous avons aussi essayé l’utilisation d’une deuxième technique, l’apprentissage par renforcement lors de la confrontation entre ennemie et bot, en plus de la programmation logique déjà utilisé avec l’automate, pour améliorer notre PNJ . L’idée serait d’utiliser le renforcement pour aider avec des choix des armes du bot, en utilisant un système de pénalisations et récompenses. Nous n'avons pas réussi à implémenter cette technique à cause du temps nécessaire pour créer une base de données avec toujours des meilleures actions pour le bot.

***OBS .:*** Nous avons testé notre projet dans notre machines personnelles en raison de l’utilisation du logiciel *SWI Prolog. Le même* ne peut être pas installé dans les machines de la salle B005 en raison des restrictions imposés sur les PC’s de l’ENIB.