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 le technique supplémentaire et les exigences minimales implémentés dans le développement. Le dernière a été basiquement 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 machine a été basé sur la création de x états. (En attente machine ahmed). L’automate a dans ta présentation l’état actuel du bot et son prochaine état (état suivant), cette définition laisse le bot toujours avec une action à réaliser.

Pour la création de l’automate, nous avons proposé l’utilisation de la langage *Prolog*, que est une langage de programmation logique. Cette choix c’est fait parce que, en première, les règles de syntaxe et la sémantique de la langage sont vraiment simples et considérées comme claires. 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. Cette choix a aidé également à l’optimisation du temps, nous avons évité toujours les conditions if/else, ça que nous a donné plus de temps pour nous concentrer dans les autres aspects du projet.

[Completer avec description de la machine - Ahmed]

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

La technique choisi pour améliorer le comportement de notre bot a été l’apprentissage par renforcement lors de la choix de les armes pour le bot . Cette technique c’est basé sur la observation des effets des actions du bot, après, le bot déduira le niveau de qualité des observations faites, et ensuite il ira améliorer ses actions futures. Le bot ira apprendre à apprendre avec des pénalisations et récompenses que seront données à lui par rapport à ses actions.

En regardant les actions du bot, nous pouvons voir que il ne prend pas en compte sa santé ni la distance au ennemie pour choisir l’arme lors d’une confrontation. Il prendra la première arme disponible, un facteur que laisse le bot en des mauvaises situations en fonction de le scenario. L’idée c’est alors ajouter l’analyse de ses facteurs pour améliorer cette décision spécifique du bot.

**Utilisation de l’apprentissage par renforcement**: Cette technique a été utilisé en basant sur les concepts vu en cours / TP. Le coefficient de exploration (epsilon) a été utilisé pour réaliser la comparaison et contrôler si le bot doit changer ou nom la choix de l’arme. Des fonctions de punition et récompense ont été créés et chaque une représente les deux différents chemins que le bot peut suivre. Si il est à une distance vraiment longue il ira maintenant prendre une arme propre à cette distance ( une rifle par exemple). Le même se passera par des distances courtes, maintenant il peut choisir une arme propre pour confronter son ennemi que est vraiment proche de lui et n’exige pas une arme très puissante.

**OBS1.:** Si il y avait plus de temps, nous pensons en faire aussi l’application des autres techniques de l’IA, comme la logique floue, comme ça, nous pourrions aussi obtenir un bot que ira arrêter de attaquer l’ennemie quand ta santé est faible ( notre bot encore attaque même avec la santé < 10)

***OBS2 .:*** 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.