Université du Québec à Chicoutimi **Session : Été 2020**
Département d'Informatique et de Mathématique **Professeur : A. Bouzouane**

Cours 8IAR125: Intelligence Artificielle pour le Jeu Vidéo

Mini-Projet : logique floue, Apprentissage et comportement dirigé par les buts

L'équipe doit être formée de 4-6 étudiants

À remettre le 23 Juin 2020[1]

1. But

Se familiariser avec les techniques de prise de décision en utilisant :

- La logique floue pour rendre réaliste le contexte de prise de décision;
- L'apprentissage à base d'un réseau de neurones pour améliorer la performance d'un bot;
- Le comportement dirigé par les buts, en vue de s'approcher du raisonnement humain dans la résolution de problèmes;
- Langage C++ (.NET).

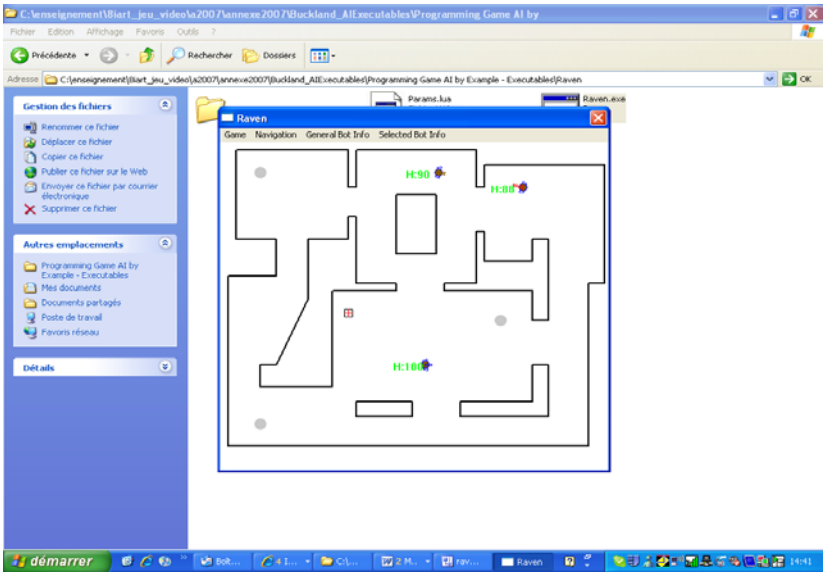
2. Cadre de l'étude

Adapter le code source des chapitres 7-10 du livre de Mat Buckland, afin d'améliorer le jeu « Raven ». Il s'agit de simuler des agents –bots- qui s'entretuent ! Chaque agent est dirigé par un ou plusieurs buts tels que, se **déplacer vers un item** (trousse de soins, munitions), **attaquer une cible**,

[1] *Pour les présentations des projets, un calendrier vous sera proposé, à surveiller. Tous les travaux doivent être remis selon la date prévue.*

explorer l'environnement,…. Ces bots se déplacent en suivant une carte qui consiste, comme le montre la figure qui suit, en un ensemble de chambres, de couloirs, et plusieurs points de génération :

- d'agents ***(spawn points)*** : pour générer un agent éliminé,
- de trousses de soins pour se soigner et prendre des forces,
- d'une variété de munitions….



L'agent-raven est déjà doté des habilitées suivantes :

1- percevoir son environnement afin de repérer les agents opposants,
2- se mouvoir à l'intérieur du monde du jeu en utilisant les techniques de ***steering behavior***,
3- prendre des décisions d'une manière rationnelle, en se basant sur les états buts et une fonction ***d'utilité ou de désirabilité***, lui permettant de faire un choix parmi l'ensemble des buts possibles,
4- choisir une arme d'une manière réaliste, en utilisant ***la logique floue***.

3. Travail à effectuer

On vous demande d'améliorer l'agent-raven en ajoutant les fonctionnalités suivantes :

a) Enrichir les règles concernant le choix d'une arme de type ***RocketLauncher***. Il s'agit d'augmenter jusqu'à **5 le nombre d'ensembles** pour les variables floues : ***Distance à la cible, État des munitions et***

Désirabilité. Cette modification sera effectuée, uniquement pour la sélection d'arme de type ***Rocketlauncher***. Le nombre maximal de règles sera de l'ordre de 25 règles **(5 points)**

b) Modifier la visée et le tir de l'agent en introduisant un **comportement flou**, à traves un **ensemble de nouvelles règles à formuler**. Par exemple, la déviation d'un tir –trajectoire d'une balle- sera en fonction des variables floues suivantes : ***la distance à la cible, la vélocité, et la période pendant laquelle la cible est demeurée visible*** **(5 points).**

c) Introduire un joueur (humain) dans le jeu et lui donner les capacités de créer **une équipe d'agent-raven** qui attaque une même cible –un agent ennemi- désigné par le joueur humain. Quand un des membres de l'équipe meurt, il doit laisser ses armes à un des coéquipiers dans un endroit connu et dédié à l'équipe. Dans le code actuel, on peut prendre possession d'un bot qui représentera le joueur humain, ce dernier peut être le leader de l'équipe. La communication au sein de l'équipe sera basée sur le protocole de communication vu dans le TP1. **(30 points)**.

d) Créer un bot-apprenant capable d'apprendre la fonction de décision de tir sur un autre bot, à partir des observations du joueur humain. Initialement, ce bot dispose de toutes les capacités (déplacement, perception, choix d'armes), comme les autres bots sauf celles de tirer. Il n'a aucune capacité de tir, il ne le sait pas ! Le but est de lui apprendre à tirer sur les autres bots ennemis. Pour cela, vous devez créer des données d'apprentissage à partir des observations de tirs du joueur humain. Pour obtenir un échantillon de données, vous faîtes jouer l'humain pendant un certain temps (quelques minutes-heures) afin de collecter les différentes situations de tir et celles où le joueur n'a pas pu tirer. Chaque situation –observation- sera décrite, par exemple, par les variables suivantes : distance en pixels du bot-ennemi-, visibilité, angle, quantité de munitions, type d'arme utilisé, le niveau de vie du bot-ennemi, et d'autres paramètres que vous jugez utiles qui caractérisent le joueur et/ou le jeu. Par ailleurs, vous pouvez utiliser n'importe quelle bibliothèque (C++) qui implémente un réseau de neurones (RN) de type perceptron ou un perceptron multi-couches. Vous n'avez pas besoin de reprogrammer l'algorithme du RN mais plutôt de l'adapter à la décision de tirer en fonction des variables d'entrées citées ci-dessus. Une fois l'entrainement est terminé, utilisé le modèle appris en intégrant le bot en question dans le jeu et mesurer sa performance en comptant le nombre de bot-ennemis atteints et sa durée de vie (**40 points**)

e) Améliorer le comportement de l'agent-raven, en introduisant un nouveau but qui peut s'inscrire dans une stratégie d'attaque ou un autre contexte d'application **(10 points).**

f) Les fonctionnalités suivantes sont facultatives mais bonifiées. La création n'a pas de limites dans votre secteur, tout est ouvert pour vous !

e.1) Création d'une nouvelle arme : grenade qui peut être utilisée selon une situation spécifique donnée **(10 points)**
e.2) Stratégie de comportement au sein d'une équipe. Vous pouvez introduire un mouvement de groupe de type Offsetpoursuit en protégeant le leader **(10 points)**
e.3) Stratégie d'attaque guidée par le leader de l'équipe **(10 points)**
e.4) Compétition entre deux bots ayant des comportements différents **(10 points)** ;
e.5) Compétition entre deux équipes de bots ayant des comportements différents **(10 points)** ;

NB : Bonus de 10 points sur la qualité: professionnalisme, originalité, réalité, ou toute autre nouvelle fonctionnalité non demandée.