

Bureau d'étude : Rapport Semaine 4

Groupe : K-A-C-A-Team

Membres :

- Kellian BALLENTINE
- Alexandre LOUVEAU
- Corentin MADRE
- Anthony BORDEAU

I) Rapport d'activité

Cette semaine nous avons continué à tester des améliorations possibles au niveau de “l'aim” de notre bot, stratégies différentes en fonction de l'arme sélectionnée, par exemple lorsque le bot à le “rocket launcher” en main il essaie d'anticiper le déplacement de son adversaire afin de pouvoir tirer vers l'endroit où il se déplace.

Nous avons ensuite regardé comment optimiser les déplacements de notre bot afin de modifier ces choix de déplacements actuels, pour le moment il fait un inventaire des objets disponibles pour la collecte et lance un jet aléatoire pour définir l'objet à aller récupérer.

Nous allons donc maintenant essayer d'implémenter une version utilisant le navmesh et optimiser les choix du bot pour aller chercher les items (en fonction de leur localisation, de leur distance par rapport au bot ou encore de leur utilité au moment de la décision).

Nous essayons toujours d'implémenter une fonction afin que le bot recule en faisant face à son adversaire mais elle n'est toujours pas au point, nous avons cependant décelé quelques problèmes que nous avons pris en compte pour essayer de la faire fonctionner (nous avons trouvé que l'on ne peut pas effectuer une action de rotation et une action de déplacement type “move.moveContinuos();” lors d'un même passage dans la boucle logic du programme), toutefois nous n'arrivons toujours pas à faire “reculer” le bot.

Pour le moment nous n'utilisons pas d'algorithmes pour définir les choix fait par notre bot, tout est basé sur des jets aléatoires, la prochaine étape est donc de passer à des décisions plus “humaine” plutôt que de tout baser sur de l'aléatoire.

II) Répartition des tâches

Pour la semaine à venir, voir les 2 prochaines semaines nous avons choisi de nous répartir les tâches de la manière suivante :

- Optimisation de l'aim en fonction des armes et adaptation du bot en combat : Kellian

- Optimisation des choix du bot (engager un combat, fuite, collecte) : Anthony

- Optimisation de la navigation (utilisation du navmesh, ...) : Alexandre

- Déplacement hors navmesh (fuite, ...) : Corentin

III) REMARQUE

Nous avons trouvé une solution pour faire reculer le bot en utilisant les localisations du bot et de son adversaire.

Lorsque le bot se fait tirer dessus nous calculons le vecteur entre l'adversaire et le bot et nous ajoutons le vecteur à la position du bot pour définir la nouvelle localisation où le bot doit se rendre tout en gardant le focus sur son adversaire.

Pour le moment le bot recule en ligne droite sans tenir compte du décors (murs, trous, ...), nous avons donc dans l'idée d'effectuer un raycasting dans le dos du bot pour détecter les variations de l'environnement lorsque le bot recule afin d'adapter sa fuite en fonction des circonstances.