



UNIVERSITÉ DE
SHERBROOKE

Projet IFT-729

Conception temps réel

Julien LEVARLET
Marius PALLARD
Thomas ROUX-CROTEAU
Thouria BEN HADDI

Encadrant :
M.Patrice ROY

12 avril 2022

1 Livrable 00

1.1 Mise en contexte

L'objectif principal du projet consiste à créer une intelligence artificielle de jeu-vidéo capable de réagir aux actions du joueur sous la forme d'un ou plusieurs ennemis. Dans n'importe quel jeu-vidéo impliquant de l'action, le joueur doit se confronter à des ennemis et ceux-ci doivent apporter du challenge et de l'aléa afin que l'expérience de jeu soit unique et stimulante pour chaque partie. Mais donner un comportement aléatoire aux ennemis n'est ni très intéressant, ni très juste en termes de difficulté, il faut donc trouver un bon équilibre dans les actions des adversaires, avoir une intelligence artificielle qui a les bonnes réactions face aux actes du joueur semble être une excellente solution.

Il faudra concevoir une palette d'actions pour le joueur et une palette de d'actions et de réactions correspondantes pour l'ennemi. L'intelligence artificielle s'assure également du bon fonctionnement et de la cohérence du jeu, en effet un opposant ne devrait pas réagir de la même façon selon l'environnement, s'il voit le joueur ou non, ou encore s'il y a d'autres ennemis à proximité. Par exemple, si l'ennemi détecte le joueur mais que ce dernier est caché derrière un mur, l'ennemi devrait être capable de rejoindre le joueur pour l'attaquer sans rester bloqué dans le mur. Le défi sera de gérer toutes ces réactions en temps réel puisque c'est un jeu d'action et non de tour par tour.

L'intérêt des composantes Temps Réel à notre produit est d'améliorer et d'optimiser le comportement de notre IA et surtout d'avoir une meilleure qualité de jeu vidéo final.

Car oui, si l'IA réagit dans des délais plus serrés avec une meilleure qualité de réponse, on pourra offrir une difficulté plus importante au joueur, et même imaginer avoir plusieurs niveaux de difficulté.

Niveau outils, rien d'imposé, mais une volonté générale d'apprendre Unity d'où l'usage du logiciel qui se révèle pratique pour notre projet et nous permettra de nous concentrer principalement sur le côté fonctionnel.

Ce projet est basé entièrement sur de la programmation et sur le moteur Unity (impliquant donc des scripts en C#), il n'y a donc aucun matériel impliqué autre que nos ordinateurs. Le moteur Unity offre des outils versatiles pour gérer des éléments et un environnement, mêmes si ceux-ci sont simples cela n'est pas très grave puisque l'aspect visuel n'étant pas une priorité de notre projet, notre contribution principale sera sur le comportement de l'ennemi (IA), son optimisation et ses possibilités.

1.2 Description des contraintes

Dans le cadre de ce projet, ce sera principalement l'aspect temporel qui nous intéresse.

Le programme d'IA que l'on affronte dans le jeu devra être capable de prendre en compte nos mouvements et adapter ses décisions dans un temps raisonnable.

Les contraintes que nous avons retenues sont les suivants :

- Brièveté : L'IA doit être capable de repérer (dans son rayon de perception), le joueur dans un délai de temps inférieur à 273 ms (temps moyen de réaction d'un humain¹) et d'agir en conséquence. Il faut donc s'assurer d'une réactivité de l'IA face au joueur ainsi qu'une réponse appropriée à l'opération initiée par le joueur. Ainsi nous pourrions assurer l'aspect de spontanéité du jeu.
- Constance : Le programme doit s'inscrire dans le rendu d'une image toutes les 60èmes de seconde du jeu (rendu d'images géré par Unity). L'actualisation de la scène doit se faire de façon régulière et stable dans le cycle d'affichage.
- Immédiateté : Les décisions de programme suite à la détection d'un stimulus provoqué par l'utilisation d'un périphérique externe (clavier, souris) doivent être réalisées en moins de 1 s (entre la détection des changements de l'environnement et l'action) pour garantir une fluidité dans le jeu.

En cas de non-respect des contraintes, le jeu pourra forcer le programme à se stopper, dans ce cas la décision retenue sera aléatoire.

1.3 Stratégies de test des contraintes

Les trois contraintes énoncées précédemment seront soumises à des tests afin d'en contrôler le respect et les corriger le cas échéant.

Il s'agira de prendre les mesures temporelles liées à chaque contrainte de façon précise à l'aide d'outils logiciel ou à même le moteur Unity, et de vérifier la pertinence des résultats obtenus.

Concernant la contrainte de brièveté, la stratégie sera de mesurer le délai entre l'opération initiée par le joueur et la détection par l'IA de cette opération. Ce délai devrait être idéalement inférieur à 273 ms comme annoncé précédemment.

Afin de vérifier le respect de la contrainte de constance, la mesure temporelle sera réalisée à un endroit prédéfini dans le cycle d'actualisation de la scène. Ainsi cette mesure permettra de rendre compte d'une régularité et stabilité dans le processus d'actualisation et génération de cette dernière.

Enfin la contrainte d'immédiateté est difficile à vérifier, elle sera évaluée sur un critère similaire à celui de la contrainte de brièveté. La stratégie consistera à mesurer le délai entre la détection du stimulus provoqué par le joueur et la réponse à ce stimulus en jeu et donc le réveil de la tâche impliquée.

1. <https://humanbenchmark.com/tests/reactiontime>

1.4 Pr vision pour le livrable L01

Le livrable L01 devra montrer la faisabilit  du projet en impl mentant au moins une des contraintes (bri vet , constance ou imm diat t ) ainsi que son test associ . Pour simplifier le probl me dans un premier temps, la contrainte devra s'appliquer sur une IA uniquement. Dans ce livrable, les d cisions de l'IA seront simplifi es, et n'auront pas besoin de prendre en compte tout l'environnement, mais uniquement les mouvements du joueur.

1.5 Pr vision pour le livrable final (L02)

Le rendu final devra comprendre tous les aspects discut s pr c demment, et donc permettre de jouer au jeu en respectant toutes les contraintes  nonc es dans ce document. Les d cisions de l'IA ici devront  tre plus complexes que dans le livrable L01, avec par exemple du pathfinding pour r ussir   rejoindre le joueur m me dans un labyrinthe.

2 Informations sur l'équipe du projet

2.1 Membres de l'équipe

PALLARD Marius (MP) - palm2802 - palm2802@usherbrooke.ca

LEVARLET Julien (JL) - levj2718 - julien.levarlet@usherbrooke.ca

BEN HADDI Thouria (TB) - bent2904 - thouria.ben.haddi@usherbrooke.ca

ROUX-CROTEAU Thomas (TR) - rout2203 - thomas.roux-croteau@usherbrooke.ca

2.2 Attributions des rôles

- Chef de projet général : Julien
- Communication externe (avec le professeur) : Thomas
- Communication interne (rappel des réunions, échéance etc. . .) : Marius
- Gestion de l'équipe (suivi général de l'avancement du travail des membres) : Thouria
- Gestion du projet (vérification quant à l'avancement du projet selon les conditions prévues (délais, décisions, diagrammes)) : Marius

Les rôles et leur distribution ont été validés par tous les membres de l'équipe.