

# Hunt(ed)

## **Projet de Moteur de jeu 3D - alpha**

*GENET Martin*

*GROSSHENNY Guillaume*

*WASMER Audric*

### **Concepts implémentés en version alpha 0.5**

#### Environnement de jeu

- Une map fixe sur laquelle le joueur est libre de se déplacer
- Un décor très sombre; le joueur possède une faible torche éclairant son entourage
- Une interface in game (affichage pour le moment simpliste)
- Un ensemble de menus (main menu & in game menu)

#### Gameplay

- Le but du jeu est de trouver une proie (représentée pour le moment par une capsule) se déplaçant sur la map, et de l'attraper. Si le joueur parvient à toucher la proie, il gagne la partie.
- Le joueur peut se déplacer comme bon lui semble, courir (consommant alors de l'endurance, représentée à l'écran par une barre), sauter, s'accroupir. La façon de se déplacer n'influence pour l'instant pas le comportement des IA.
- Le joueur dispose d'un petit nombre de pièges qu'ils peut placer au sol. Ces derniers immobilisent un court moment quiconque marche dessus (joueur, proie ou traqueur).
- Un certain nombre de traqueurs (représentés pour le moment par des sphères) se déplacent sur la map. Lorsque le joueur entre dans leur périmètre de détection, le joueur est pourchassé. Si un traqueur parvient à toucher le joueur, celui-ci perd la partie.
- Les traqueurs/la proie fonctionnent de la manière suivante:
  - Lorsque le joueur est hors du périmètre de détection, le traqueur/la proie se déplace de façon aléatoire.
  - Lorsque le joueur entre dans le périmètre de détection, le traqueur/la proie pourchasse le joueur/s'enfuit dans la direction opposée.
  - Lorsque le joueur entre dans le second périmètre de détection, plus petit, le traqueur/la proie accélère.
  - Lorsque le joueur sort du périmètre de détection, le traqueur/la proie recommence à se déplacer aléatoirement.

### **Bugs connus dans la version actuelle**

- Les pièges se posent parfois juste sous le joueur et non à la position prévue
- L'IA de la proie est instable

- Les traqueurs & la proie ne semblent pas gagner en vitesse lorsque le joueur s'approche trop

### **Pistes de progression envisagées**

- Génération procédurale d'une map au lancement de la partie
- Diversité de biomes de map (désert, forêt, urbain...)
- Faible probabilité de fin alternative
- Gestion de l'obstruction du champ de vision par les éléments du décor pour les traqueurs et la proie pour la détection du joueur
- Prise en compte du mode de déplacement du joueur (sprint/sneak) dans le comportement des IA.
- Équilibrage du système de piège (temps d'activation, visibilité plus ou moins importante, etc.)
- Environnement sonore
- Système de traces laissées par la proie

-