

Campus ADN 2021-2022 – Session 2

FINAL PROTOTYPE DESIGN

WENDIGO AI

Auteur du document : Sacha Benamou

Scripteurs : Frédérique Leclerc, Richard Loiselle, Sacha Benamou

25/03/2022

Préface

Le Wendigo fonctionne grâce à trois systèmes principaux : sa **Finite State Machine**, son **Director AI**, et son **Job System**. Ces trois systèmes seront expliqués et détaillés dans le reste de ce document.

Table des matières

Préface.....	1
Job System.....	2
Général	2
Types de Jobs et quand ils sont assignés.....	2
Director AI	3
Général	3
Menace Gauge.....	3
Utilité	3
Facteurs	4
Équation	4
Utilisation du Job System	5
Finite State Machine – Wendigo	6
Général	6
Explication des States.....	6
Idle.....	6
Patrol	6
Investigate	7
Chase	7

Job System

Général

Le **Job System** est essentiellement un vecteur de communication entre le **Wendigo** et son **Director**.

Ainsi, le **Director** l'utilisera pour transmettre des Jobs/tâches à compléter au **Wendigo**, alors que ce dernier l'utilisera pour surveiller ce qu'il a à faire.

Pour cela, une **Classe Job** a été créée, ainsi qu'un **ScriptableObject** contenant toutes les données relatives au **Job System** – notamment la liste de Jobs que le **Director** et son **Wendigo** utiliseront pour communiquer.

Types de Jobs et quand ils sont assignés

Il existe deux types de **Jobs** :

- "Idle" : Ordonne au Wendigo de se remettre dans son State "Idle", dans le but de laisser respirer le joueur*.
- "Search" : Ordonne au Wendigo de sortir de son State "Idle" pour aller chercher le joueur*.

* : Ces conditions sont déterminées par la **MENACE GAUGE**, expliquée dans la section [DirectorAI](#) de ce document.

Director AI

Général

Le **Director** est une partie extrêmement importante du fonctionnement général de l'antagoniste que nous prototypons. En effet, celui-ci est le seul à toujours connaître les positions du joueur ET du **Wendigo**, et facilite ainsi la tâche de recherche de ce dernier.

Cela dit, il ne donne **JAMAIS** au **Wendigo** la position exacte du joueur. L'antagoniste doit "mériter" sa récompense. Ainsi, il va plutôt lui dire dans quel **Chemin** se trouve le joueur au moment où il juge qu'il faut aller le chasser.

Avec cette information, le **Wendigo** s'assurera de *spawner* aussi loin du joueur que possible dans le **Chemin** concerné, et se mettra ensuite à effectuer une patrouille dans l'objectif de le trouver.

Menace Gauge

La **MENACE GAUGE** est une autre partie intégrante du **Director**. Fondamentalement, cette jauge prend la forme d'un nombre décimal oscillant entre 0 et 1, représentant le niveau de pression mise sur le joueur à tout moment.

Utilité

Cette information est nécessaire afin de savoir quand donner ses **Jobs** au **Wendigo**. Ainsi, lorsque la valeur de la **MENACE GAUGE** se trouve en dessous de 0.2, le **Director** envoie le **Wendigo** chasser le joueur. Lorsqu'elle est au-dessus de 0.8, le **Director** demande au **Wendigo** de se remettre dans son State "Idle" à la fin de son **Chase** ou **Investigate** actuel (s'il était en train d'en effectuer).

Facteurs

Pour l'instant, trois facteurs sont utilisés pour calculer la valeur de la **MENACE GAUGE** :

- Le temps passé depuis la dernière fois où le **Wendigo** était dans son State "Idle".
- Le temps passé par le **Wendigo** dans le *Line of sight* du joueur*.
- Un **facteur d'amortissement** servant à mieux maîtriser la vitesse du changement de la valeur générale de la **MENACE GAUGE**.

* : Depuis la dernière fois qu'il était dans son State "Idle".

NOTE : Le nombre de facteurs pris en compte dans le calcul a été volontairement réduit pour les besoins du prototype. Dans le jeu final, nous comptons ajouter deux facteurs (encore indéterminés) au calcul.

Équation

L'équation qui nous permet d'obtenir la valeur de la **MENACE GAUGE** s'écrit comme suit :

$$\frac{Nb\ frames\ dans\ LOS + Nb\ frames\ search}{(NbFramesLOSTMax + NbFramesSearchMax)^{amortissement}}$$

- Nb frames dans LOS : Le temps passé (en frames) par le **Wendigo** dans le *Line of sight* du joueur.
 - NbFramesLOSTMax : "intervalle haute" pour ce facteur.
- Nb frames search : Le temps passé (en frames) depuis la dernière fois où le **Wendigo** était dans son State "Idle".
 - NbFramesSearchMax : "intervalle haute" pour ce facteur.
- Amortissement : facteur d'amortissement servant à mieux maîtriser la vitesse du changement de la valeur générale de la **MENACE GAUGE**.

Utilisation du Job System

Comme précisé [plus haut](#), le **Director** utilise principalement le **Job System** afin de demander au **Wendigo** de chercher le joueur (ou d'arrêter de le faire) en se basant sur la valeur de sa **MENACE GAUGE**.

Ainsi, lorsque cette valeur se situe en dessous de 0.2, le **Director** va créer un **Job** de "Search" pour son **Wendigo** et l'ajouter à la liste disponible dans le **ScriptableObject** lié au **Job System**. Ce **Job** va envoyer le **Wendigo** au **Waypoint** le plus loin du joueur, à l'intérieur de **Chemin** dans lequel se trouve ce dernier. Le **Wendigo** commencera alors le comportement dicté dans son State "Patrol" pour tenter de trouver le joueur.

Dans le cas contraire, lorsque la valeur de la **MENACE GAUGE** est au-dessus de 0.8, le **Director** créera un **Job** de "Idle" pour son **Wendigo**. Ce dernier attendra alors de terminer toute activité actuellement entreprise (Investigate ou Chase) avant de se remettre dans son State "Idle" et d'arrêter de chercher le joueur.

Finite State Machine – Wendigo

Général

La **Finite State Machine** du **Wendigo** est responsable du comportement adopté par ce dernier dans une situation donnée.

Pour cela, 4 States y ont été implantés : “Idle”, “Patrol”, “Investigate” et “Chase”. Un *breakdown* de chacun de ces States est lisible ci-dessous.

NOTE : Un 5^e State, “Ambush”, est actuellement envisagé pour le jeu final. Sa production n’ayant pas pu être complétée à temps pour la remise de ce prototype, son design est repoussé au début de la production, au mois d’avril 2022.

Explication des States

Idle

Ce State est utilisé pour indiquer au **Wendigo** d’arrêter de chasser le joueur. Ainsi, le passage en Idle rend l’antagoniste invisible et complètement inactif.

Le **Wendigo** sera dans ce State au début de chaque partie/niveau, et dès que son **Director** décidera que le niveau de pression qu’il exerce sur le joueur est trop haut.

STATE SWITCHES POSSIBLES : Idle -> Patrol.

Patrol

Ce State est utilisé pour indiquer au **Wendigo** d’effectuer sa routine de *roaming* dans un chemin donné, et dans l’objectif de trouver le joueur.

Ainsi, dans ce State, le **Wendigo** utilisera les **Waypoints** et **Chemins** placés par les LDs pour évoluer dans le niveau. Pendant son *pathfinding*, il se fiera à ses capteurs auditifs et visuels afin de trouver le joueur.

Si son capteur auditif est stimulé, il passera en State “Investigate” et se rendra à la source du bruit qu’il a entendu dans le but de trouver le joueur.

Si son capteur visuel est stimulé par le joueur, il passera en State "Chase" et se mettra à poursuivre le joueur jusqu'à ce qu'il perde sa trace visuelle.

STATE SWITCHES POSSIBLES : Patrol -> Investigate, Patrol -> Chase, Patrol -> Idle.

Investigate

Comme précisé plus haut, dans ce State, le **Wendigo** va enquêter sur un bruit qu'il a entendu.

Ainsi, il va se rendre à la source du bruit, avant d'évaluer la situation :

- S'il voit le joueur, il passe en State "Chase";
- S'il ne le voit pas, il retourne à son State "Patrol" au bout de quelques secondes et après avoir inspecté les environs (rotation sur lui-même).

STATE SWITCHES POSSIBLES : Investigate -> Chase, Investigate -> Patrol, Investigate -> Idle.

Chase

Dans ce State, le **Wendigo** a vu le joueur et se met à le poursuivre. Ainsi, il continuera de le suivre jusqu'à ce qu'il le perde de vue.

De plus, s'il est assez proche du joueur, le **Wendigo** peut lancer une attaque dans ce State.

STATE SWITCHES POSSIBLES : Chase -> Patrol, Chase -> Idle