

LOG8235

**AGENTS INTELLIGENTS  
POUR JEUX VIDÉO**

/ SEMAINE 1

# TRAVAUX PRATIQUE 1

# INTRODUCTION



# / INTRODUCTION

Le but de ces travaux pratiques est de mettre en pratique les différentes notions abordées lors des différents modules de cours théorique.

Les travaux doivent être réalisés à l'aide du cadriciel fourni basé sur le Unreal Engine. Le fil conducteur est le développement d'agents intelligents pour un jeu de style Pac-Man.

Un certain nombre d'ingrédients vous sont fournis afin de vous concentrer pleinement sur les différentes tâches demandées (cf section *Éléments Fournis*)

L'objectif de ce premier TP est le développement de comportements simples basés sur l'analyse de l'environnement dans lequel évolue les agents intelligents.

# TRAVAIL À ACCOMPLIR



# / TRAVAIL À ACCOMPLIR

## 1. Comportement de déplacement de base

L'agent doit pouvoir se déplacer à l'aide d'un vecteur vitesse déterminé par l'étudiant. Les variables d'entrées pour calculer le vecteur vitesse doivent être une valeur d'accélération et une valeur de vitesse max.

Le pawn doit aussi être orienté vers la direction du déplacement.

Éléments obligatoires	Points
Calcul du vecteur vitesse	1 pt
Déplacement de l'agent en fonction du vecteur vitesse	2 pts
Orientation de l'agent dans la direction de son déplacement	2 pts

# / TRAVAIL À ACCOMPLIR

## 2. Détection de murs et comportement d'évitement

L'agent doit pouvoir identifier qu'un mur l'empêche d'avancer. Il doit alors effectuer une rotation de X degrés par rapport au plan de collision tout en continuant de son déplacement.

Éléments obligatoires	Points
Détection de la proximité d'un mur	2 pts
Ajustement de la vitesse de déplacement	2 pts
Rotation progressive de l'agent vers la nouvelle orientation	1 pt

# / TRAVAIL À ACCOMPLIR

## 3. Détection d'obstacles et comportement d'évitement

De la même manière que l'agent identifie les murs, il doit pouvoir identifier les dalles au sol de type *death* et appliquer la même stratégie d'évitement que pour les murs.

Éléments obligatoires	Points
Détection de la proximité d'une dalle	3 pts
Réutiliser le comportement d'évitement des murs	2pts



# / TRAVAIL À ACCOMPLIR

## 4. Détection et collecte de pickup

L'agent doit pouvoir détecter la présence d'un pickup devant lui et se diriger vers celui-ci s'il n'y a pas d'obstacle sur son chemin.

Éléments obligatoires	Points
Détection d'un pickup devant l'agent	3 pts
Se diriger vers le pickup s'il n'y pas d'obstacle	2 pts

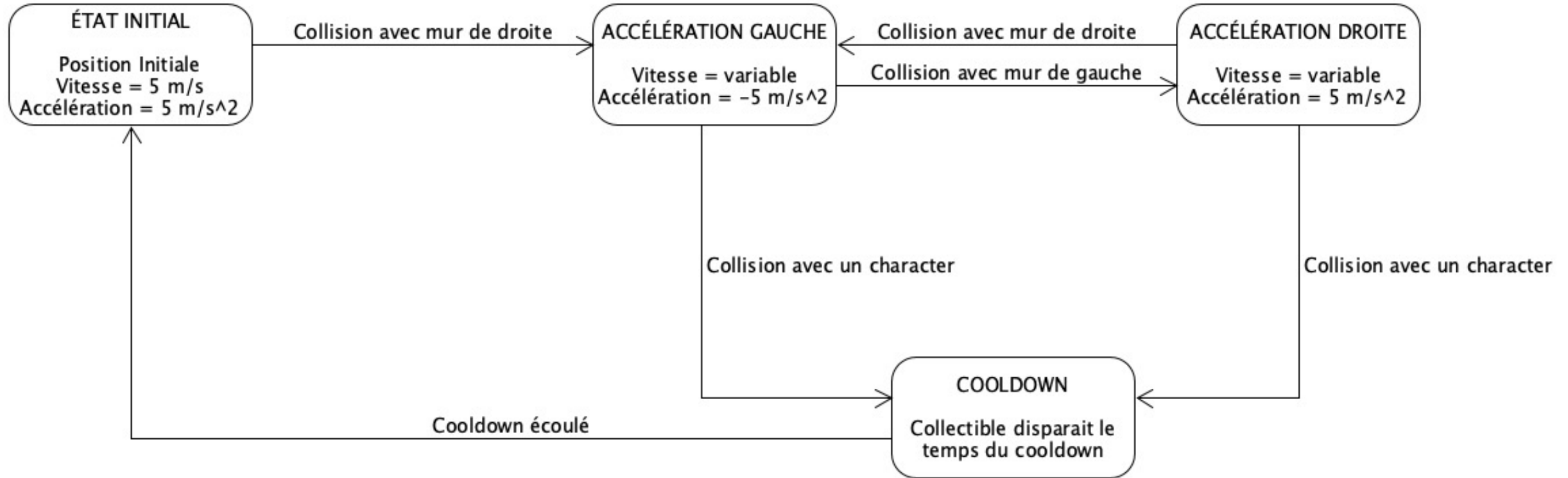
# / TRAVAIL À ACCOMPLIR

## 5. Déplacement d'un collectible

Si la variable `isMoveable` est réglé à "True", le collectible argenté doit se déplacer de gauche à droite dans le couloir du bas. Le collectible doit avoir une vitesse maximale de 5m/s et une accélération de 2.5m/s<sup>2</sup>. Il doit se déplacer selon la méthode d'Euler et doit se comporter selon la machine à états de la prochaine diapositive.

Éléments obligatoires	Points
Le collectible se déplace de gauche à droite avec la bonne vitesse.	1 pt
Implémentation de la méthode d'Euler avec la bonne accélération.	2 pts
Le collectible se comporte selon la machine à états.	2 pts

# / MACHINE À ÉTATS DU COLLECTIBLE



Voir le vidéo suivant : [https://youtu.be/\\_ZE52bKavYU](https://youtu.be/_ZE52bKavYU)

# / TRAVAIL À ACCOMPLIR

## 6. Comportement de poursuite

L'agent doit pouvoir avoir un comportement de poursuite du joueur. L'agent se dirige alors vers le joueur en prenant en compte la topologie du niveau.

Ce comportement ne doit être actif que lorsque l'état du joueur n'est pas affecté par un pickup.

Éléments obligatoires	Points
Détection sphérique du joueur	2 pts
Se diriger vers le joueur s'il n'y pas d'obstacle	2 pts
Pris en compte de l'état du joueur	1 pt

# / TRAVAIL À ACCOMPLIR

## 7. Comportement de fuite

L'agent doit pouvoir avoir un comportement de fuite du joueur. L'agent cherche alors à s'éloigner du joueur en prenant en compte la topologie du niveau.

Ce comportement ne doit être actif que lorsque l'état du joueur est affecté par un pickup.

Éléments obligatoires	Points
Comportement de fuite (éloignement du joueur)	2 pts
Pris en compte de l'environnement	3 pts

# / TRAVAIL À ACCOMPLIR

## 8. Identification et exposition des différents paramètres permettant de régler les comportements de l'agent

Les différents paramètres tel que la vitesse maximale de déplacement, l'accélération ou la distance de détection doivent être exposées afin de pouvoir être éditées sans avoir à modifier le code de l'agent. Les paramètres étant destinés aux designers, on notera avec attention l'effort apporté sur la documentation de ces paramètres ainsi que sur la vérification de leur validité.

Éléments obligatoires	Points
Paramètres exposés dans Unreal	1 pt
Description de chaque paramètre	2 pts
Test de la validité de chaque paramètre	2 pts

# / TRAVAIL À ACCOMPLIR

## 9. Test automatique des agents

Afin d'évaluer l'efficacité des agents, le programme devra compter le nombre de pickup ramassé ainsi que le nombre de mort de l'agent pendant un temps donné. Le jeu est alors lancé en mode simulation sans le joueur (Alt-S).

Éléments obligatoires	Points
Compteur de pickup ramassé / nombre de mort de l'agent	2 pts
Compteur de temps	1 pt
Affichage du résultat	2 pts

# ÉLÉMENTS FOURNIS





# / ÉLÉMENTS FOURNIS

## Blueprints Unreal

### *BP\_DeathFloor*

- Ingredient gameplay de type obstacle dont la fonction est de provoquer la téléportation au point de départ de l'entité ayant collisionné avec
- StaticMeshActor avec une collision de type "DeathObject"

### *BP\_SDTCollectible*

- Ingredient gameplay de type pickup dont la fonction est de disparaître lorsqu'une entité rentre en contact avec. Si l'entité est le joueur, le pickup confère alors l'état « PowerUp » pendant un certain temps au joueur. Si la variable isMoveable est réglé à true, le collectible doit se déplacer.

### *BP\_SDTACharacter*

- Classe de base devant être utilisé pour implémenter le code du pawn de l'agent

### *BP\_SDTAController*

- Classe de base devant être utilisé pour implémenter le code du controller de l'agent

### *BP\_SDTMainCharacter*

- Entité joueur qui doit être contrôlé par un humain.
- Utilise une collision de type « Player »

## Document

### [ToolBox-Polytechnique](#)

- Liste de fonctions Unreal documentées à utiliser pour la réalisation du TP.

# ÉVALUATION



# / ÉVALUATION 1/2

Travail à accomplir	Points
1. Comportement de déplacement de base	5 pts
2. Détection de murs et comportement d'évitement	5 pts
3. Détection d'obstacle et comportement d'évitement	5 pts
4. Détection et collecte de pickup	5 pts
5. Déplacement d'un collectible	5 pts
6. Comportement de poursuite	5 pts
7. Comportement de fuite envers le joueur	5 pts
8. Identification et exposition des différents paramètres permettant de régler les comportements de l'agent	5 pts
9. Test automatique des agents	5 pts
Aspect global de la réalisation*	5 pts
Soin apporté au code**	-10 pts

**Total** : 50 points

## **/ ÉVALUATION 2/2**

**\*L'aspect global de la réalisation prendra en compte le soin apporté à la crédibilité des comportements de l'agent. Il sera apprécié entre autres :**

- Qualité des transitions entre les comportements**
- Accélération/décélération des mouvements**

**\*\*Le soin apporté au code appréciera les éléments suivants :**

- Taille des fonctions**
- Factorisation du code**

# REMISE



# / REMISE

Pour la remise, veuillez me remettre un rapport pdf.

Ce rapport devra contenir les informations suivantes :

Vos noms et matricules, ainsi qu'une liste des fichiers et fonctions vous avez modifié.

La solution devra être un fichier .zip contenant les fichiers et répertoires suivants :

Config/...

Content/...

Source/...

SoftDesignTraining.uproject

PS: assurez vous que votre code est bien commenté et lisible. Cela va m'aider à vous aider lors de ma correction.

Bonne remise! 😊

LOG8235

AGENTS INTELLIGENTS  
POUR JEUX VIDÉO