

# Projet de Session

Implémentation d'éléments techniques

## Objectif

Le but de ce projet, qui s'étend sur les deux derniers tiers de la session, est d'intégrer et d'appliquer les éléments théoriques vu en classe dans un projet de démonstration technologique.

## Livrables

La réalisation du projet compte 2 livrables répartis à différents moments de la session. Chaque livrable se concentre sur un des modules vus dans le cours, soit l'interactivité et la rétroaction audiovisuelle.

Pour tous les livrables, les instructions suivantes sont de rigueur :

1. L'ensemble des éléments remis doit être compris **dans une archive au format .zip**. Cette archive doit **respecter la nomenclature** suivante : IFT2103A19\_TP#\_EquipeXX, où # est le numéro du livrable et XX est votre numéro d'équipe (par exemple, le livrable 2 pour l'équipe 3 serait dans une archive nommée IFT2103A19\_TP2\_Equipe03).
2. **Tous** les documents à remettre doivent être **dans un même fichier .pdf** situé **à la racine** et portant le même nom que l'archive, sous forme d'un document principal et de ses annexes.
3. La qualité de la langue, la pertinence et la concision sont de rigueur dans tous vos documents.
4. Tout exécutable doit pouvoir être lancé depuis la racine de l'archive (utilisez un raccourci/.bat au besoin) et ne pas demander de configuration particulière. **Il est de votre responsabilité de vous assurer du bon fonctionnement de ceux-ci.**
5. **Ssi** l'ensemble des remises dépasse la capacité de la boîte de dépôt (250 Mo), vous pouvez inclure un lien de téléchargement dans un fichier readme.txt situé à la racine du projet.
6. Les sources de l'exécutable doivent être incluses afin de valider le travail effectué.
7. Les livrables doivent être déposés dans la boîte de dépôt sur le site du cours avant 23h59. Chaque jour de retard vaut 10% de pénalité, et lorsque les corrections sont publiées il n'est plus possible de déposer vos travaux. Vous pouvez faire plusieurs remises, seule la dernière sera corrigée.

**Le non-respect de ces instructions se traduira en pénalité(s) sur la note.**

## Premier Livrable : Interactivité (TP2)

À remettre le 25 novembre 2020 avant 23h59

### Mandat

Livrer un exécutable permettant à un utilisateur d'interagir avec une boucle de jeu dans un environnement simple.

### Objectif

Ce second livrable vise à mettre en pratique les notions d'interactivité liées au jeu vidéo, soit l'intégration du cycle d'interactivité dans la boucle de jeu, la gestion de la saisie des entrées par l'utilisateur, le support d'utilisateurs multiples et l'intelligence artificielle.

### Détail du livrables

L'exécutable réalisé doit présenter les fonctionnalités suivantes :

- Un flot d'application
- Au moins deux agents contrôlés par des joueurs distincts.
- Une interface graphique présentant la progression de la simulation
- Un agent autonome

L'exécutable doit aussi présenter une fonctionnalité réalisée individuellement par chacun des membres de l'équipes parmi les suivantes :

- Le support du jeu en réseau (ou en ligne)
- Une intelligence artificielle de plus haut niveau s'opposant au joueur
- La personnalisation des méthodes d'entrées

L'exécutable doit être accompagné d'un bref document qui explique l'intégration de chaque fonctionnalité. Ce document devrait contenir :

- Le diagramme du flot d'application
- Le schéma de contrôle des agents
- Une description de l'intelligence artificielle de l'agent autonome
- Une description des fonctionnalités supplémentaires

### Évaluation

Sur 100 points

Flot d'application	/15
Contrôle des agents	/15
Multijoueur	
Gestion des entrées	/5
Affichage	/5
Agent autonome	/10
Recherche de chemin	/10
Interface graphique	/20
Fonctionnalité supplémentaire (individuelle)	/20

## Deuxième Livrable : Rétroaction audiovisuelle (TP3)

À remettre le 20 décembre 2020 avant 23h59

### Mandat

Agrémenter l'exécutable remis au livrable précédent avec des éléments de rétroaction audiovisuelle.

### Objectif

Ce dernier livrable vise à mettre en pratique les notions de rétroaction audiovisuelle liées au jeu vidéo, soit l'animation des éléments, la génération procédurale, Les effets visuels et l'audio.

### Détail du livrables

L'exécutable réalisé doit présenter les fonctionnalités suivantes :

- L'animation interactive des agents
- L'animation de l'interface
- Des effets de particules
- Une musique de fond
- Des effets sonores lors des actions

L'exécutable doit aussi présenter une fonctionnalité réalisée individuellement par chacun des membres de l'équipes parmi les suivantes :

- La génération procédurale de l'environnement
- La personnalisation des avatars
- Une musique de fond réactive

L'exécutable doit être accompagné d'un bref document qui explique l'intégration de chaque fonctionnalité. Ce document devrait contenir :

- Les méthodes d'animation employées pour les agents
- Les méthodes d'animation employées pour l'interface
- La description des effets de particules et de leur contexte d'utilisation
- La description de l'ambiance sonore
- La liste des effets sonores et de leur contexte d'utilisation
- La description de la fonctionnalité optionnelle

### Évaluation

Sur 100 points

Animation des agents	/15
Animation de l'interface graphique	/20
Effets visuels	/15
Ambiance sonore	/15
Effets sonores	/15
Fonctionnalité supplémentaire (individuelle)	/20

## TP2 – Interactivité

/100

Commentaires Généraux :

## Flot d'application

/15

L'application a une fin	(2)	
La simulation retourne au menu principal à la fin de la simulation	(5)	
L'application a un écran d'accueil	(2)	
Le flot d'application présente une étape de configuration de la simulation	(4)	
La simulation peut être mise en pause	(2)	
Le flot d'application n'a pas de cul-de-sac	(2)	
L'application peut être interrompue pour de revenir au menu principal	(2)	

Commentaires :

## Contrôle des agents

/15

Un joueur peut déclencher l'action d'un agent.	(3)	
Un joueur peut « configurer » l'action de l'agent.	(6)	
Le joueur contrôle le déplacement de l'agent.	(3)	
Plus d'un type de contrôleur est supporté.	(3)	
Des touches binaires sont utilisées pour simuler un axe.	(3)	

Commentaires :

## Multijoueur – Gestion des entrées

/5

Deux joueurs humains peuvent contrôler des agents distincts	(2)	
Chaque joueur a son contrôleur assigné	(5)	

Commentaires :

## Multijoueur –Affichage /5

Les agents actifs sont toujours visibles	(2)	<input type="text"/>
Le viewport s'adapte en fonction du joueur actif ou chaque joueur a un viewport.	(5)	<input type="text"/>

Commentaires :

## Agent autonomes /10

Un agent autonome a un comportement scripté	(3)	<input type="text"/>
Un agent autonome a un comportement scripté avec plus de deux états.	(6)	<input type="text"/>
Un agent autonome réagit aux actions des autres agents.	(9)	<input type="text"/>
Le comportement de l'agent ne présente pas de bogues.	(10)	<input type="text"/>

Commentaires :

## Recherche de chemin /10

Au moins un agent se déplace.	(3)	<input type="text"/>
Au moins un agent fait de la recherche de chemin programmée par l'équipe.	(6)	<input type="text"/>
La recherche de chemin ne présente pas de bogues.	(9)	<input type="text"/>
La recherche de chemin est faite avec l'algorithme approprié.	(10)	<input type="text"/>

Commentaires :

## Interface graphique /20

L'écran d'accueil est suivi d'un menu principal.	(3)	<input type="text"/>
Les configurations de la simulation sont modifiables par des menus.	(3)	<input type="text"/>
Lors de sa mise en pause, la simulation affiche un menu.	(3)	<input type="text"/>
Des éléments de menus (HUD) affichent les données de la progression de la simulation.	(3)	<input type="text"/>
L'application a un écran de chargement.	(3)	<input type="text"/>
L'écran de chargement affiche la progression du chargement.	(5)	<input type="text"/>
L'implémentation sépare la logique du jeu, l'affichage de l'interface et les données.	(3)	<input type="text"/>

Commentaires :

## Jeu en réseau (par : ) /20

La simulation est synchronisée entre plusieurs instances de l'application.	(3)	
Le jeu peut être joué en réseau sur au moins deux machines.	(6)	
La synchronisation du jeu se fait adéquatement.	(9)	
La performance de la synchronisation est adéquate.	(12)	
Les actions des joueurs distants sont fluides.	(3)	
Un maître serveur permet la mise en relation facilement.	(3)	
Le maître serveur fait de la mise en relation automatique.	(5)	

Commentaires :

## Personnalisation des méthodes d'entrées (par : ) /20

Le schéma de contrôle passe par un gestionnaire de contrôles.	(3)	
Le gestionnaire de contrôles a été codé par l'équipe.	(6)	
Les touches peuvent être réassignées.	(9)	
Le comportement des axes de contrôles peut être personnalisé.	(12)	
Chaque joueur peut choisir son contrôleur.	(3)	
Chaque joueur peut ajuster ses contrôles.	(3)	
Les propriétés de comportement du contrôleur peuvent être enregistrées et récupérées.	(3)	

Commentaires :

## Intelligence artificielle (par : ) /20

Un agent autonome planifie ses actions.	(3)	
La planification ne présente pas de bogues majeurs	(6)	
L'IA présente au moins deux niveaux de difficultés.	(8)	
Le système d'IA est analysé dans le rapport.	(3)	
L'analyse est complète et juste.	(6)	
L'implémentation présente un bon niveau de performance (PEAS).	(9)	
L'architecture prévue pour l'implémentation est pertinente.	(3)	

Commentaires :

## TP3 – Rétroaction audiovisuelle

/100

Commentaires Généraux :

## Animation des agents

/15

Les agents sont animés.	(3)	<input type="text"/>
Les agents ont plus d'une animation.	(6)	<input type="text"/>
Le flot d'animation ne présente pas de bogues majeurs.	(9)	<input type="text"/>
L'animation des agents est coordonnée aux événements de la simulation.	(3)	<input type="text"/>
Les animations sont structurées de façon à être fluides et réactives.	(6)	<input type="text"/>

Commentaires :

## Animation de l'interface graphique

/20

L'interface est animée.	(3)	<input type="text"/>
L'animation est effectuée par du code écrit par l'équipe.	(6)	<input type="text"/>
Les animations utilisent du easing.	(9)	<input type="text"/>
Le système d'animation ne présente pas de bogues majeurs.	(12)	<input type="text"/>
L'interface comporte plusieurs types d'animations.	(15)	<input type="text"/>
Les animations sont structurées de façon à être fluides et réactives.	(18)	<input type="text"/>
L'animation de l'interface est agrémentée de VFX et de SFX.	(2)	<input type="text"/>

Commentaires :

## Effets visuels

/15

L'application présente un système de particules.	(3)	<input type="text"/>
L'application présente plusieurs systèmes de particules variés.	(6)	<input type="text"/>
Les effets de particules sont accompagnés d'effets lumineux.	(3)	<input type="text"/>
Les effets de particules sont « poolés » et mis en cache.	(2)	<input type="text"/>
Le système de pooling est codé par l'équipe.	(4)	<input type="text"/>
Le système de pooling ne présente pas de bogues	(6)	<input type="text"/>

Commentaires :

Ambiance sonore /15

L'application fait jouer une musique d'arrière-plan.	(3)	
L'application fait jouer plusieurs musiques d'arrière-plan.	(6)	
Des effets sonores environnementaux (foleys) complètent l'ambiance sonore.	(3)	
Les effets sonores sont spatialisés.	(6)	
Le volume de la musique et des foleys peut être ajusté séparément.	(3)	

Commentaires :

Effets sonores /15

Les actions de la simulation sont accompagnés d'effets sonores.	(3)	
Il y a plus d'un effet sonore.	(6)	
Les effets sonores sont bien synchronisés avec la simulation.	(9)	
Les effets sonores sont spatialisés.	(3)	
Le volume des SFX peut être ajusté séparément des autres sons.	(3)	

Commentaires :



## Génération procédurale de l'environnement (par : ) /20

L'environnement est recomposé à l'exécution en disposant des « blocs » selon un plan.	(3)	
La structure du niveau est générée procéduralement en permutant quelques gros blocs.	(6)	
La structure du niveau est générée procéduralement.	(9)	
La structure générée est toujours jouable.	(3)	
La génération est paramétrable et reproductible par germe.	(3)	
Une variation dans le visuel est générée lors de la reconstitution.	(3)	
L'environnement généré est varié et animé	(5)	

Commentaires :

## Personnalisation de l'avatar (par : ) /20

L'utilisateur peut choisir entre une sélection d'avatar préconstruits.	(3)	
L'utilisateur peut changer la couleur de l'avatar.	(3)	
L'utilisateur peut choisir plusieurs couleurs de son avatar séparément.	(6)	
Le joueur peut modifier la transformation de base de son modèle.	(3)	
L'avatar est construit par composition de maillages/textures.	(6)	
L'avatar est construit par combinaisons de maillages/textures.	(9)	
La combinaison de maillages utilise un système de masque.	(11)	

Commentaires :

## Musique dynamique (par : ) /20

La musique des menus est différente de la musique du gameplay.	(3)	
Des « jingles » sont jouées lors d'évènements clés de la simulation.	(3)	
La musique du gameplay change en fonction des événements de la simulation.	(3)	
La transition des musiques se fait par fondu croisé.	(6)	
Le système de musique dynamique ne présente pas de bogues.	(9)	
Les pistes sont synchronisées pour modifier l'ambiance musicale de façon fluide.	(12)	
Un système de mixage dynamique modifie l'ambiance musique en utilisant des canaux.	(14)	

Commentaires :