

# SYLLABUS PROJET

Nom du projet :	Projet de Synthèse : « Ce n'est qu'une question de performances ... »
Matière(s) liée(s) au projet	Programmation Avancée de Jeu Vidéo à l'aide d'Unity
Année et spécialisation(s) :	4A-DJV

Enseignant :	Nicolas Vidal	Mail de l'enseignant :	<a href="mailto:pro@nicolasvidal.fr">pro@nicolasvidal.fr</a>
--------------	---------------	------------------------	--

Type de projet :	Projet annuel	Projet de cours	Semaine projet
<i>Cocher la case</i>		X	

Type de sujet :	Sujet imposé	Sujet à faire valider	Sujet fourni par entreprise
<i>Cocher la case</i>	X		

Nombre d'étudiants par groupe :	3-4	Règles pour la constitution des groupes	Libre	Charge de travail estimée de chaque étudiant :	40h
---------------------------------	-----	---	-------	--	-----

## Objectif du projet (à la fin du projet, les étudiants sauront réaliser un...)

Réaliser un Jeu Vidéo complet, en se plaçant dans un contexte de ressources matérielles limitées et+ en exploitant des algorithmes d'exploration et de planification.

## Objectifs pédagogiques complémentaires

Réaliser un jeu vidéo complet avec Unity, savoir correctement utiliser les ressources de la machine.

## Ouvrages de Référence (livres, articles, revues, sites Web)

<http://unity3d.com/support/documentation/> ; <http://fr.openclassrooms.com/informatique/cours/realisez-votre-premier-jeu-video-avec-unity> ;  
<http://aigamedev.com/open/interviews/mario-ai/>

## Outils informatiques à installer

Unity(développement) <http://unity3d.com/unity/download> , Suite Office (présentation et document de suivi de projet), Git SCM (source control and versioning) : <http://git-scm.com/>

## Sujet détaillé (à fournir éventuellement en pièce jointe)

### Cadre général

Conception d'un Jeu Vidéo comportant un personnage contrôlé par l'ordinateur utilisant des techniques de planification pour atteindre un objectif, le joueur humain devra pouvoir agir sur l'environnement soit dans un but coopératif, soit compétitif.

Le GameDesign est à la charge de l'équipe projet et devra être proposé dans les plus brefs délais pour validation.

### Contraintes

#### Type de Jeu

Le type de jeu est **libre**, mais il est cependant recommandé de vous inspirer d'un classique existant, en remplaçant le joueur par une IA :

- Mario : <https://www.youtube.com/watch?v=DlkMs4ZHHr8>

D'autres classiques peuvent être reproduits, ils devront cependant posséder les caractéristiques suivantes :

- Environnement continu (ou simili continu)
- Explosion combinatoire du nombre d'états possibles
- Moteur de jeu déterministe et simple à reproduire

L'idée est de laisser une liberté conséquente à l'équipe de projet pour permettre de valoriser une création dans le cadre de leur scolarité, tout en ayant un socle commun permettant une évaluation sensée des différents projets les uns par rapport aux autres.

#### Innovation

Il sera attendu de la part du groupe d'étudiant de pouvoir justifier **la totalité des choix d'implémentations et d'architecture**. Il sera important de réfléchir longuement avant de se lancer dans l'écriture du code.

#### Gameplay

Le gameplay devra être centré autour du comportement de l'entité non joueur gérée par un algorithme de planification. Le joueur humain devra pouvoir modifier l'environnement de manière imprévue pour aider ou mettre en défaut l'IA.

#### Features obligatoires

- Menu, options, etc.
- Robustesse
  - Mise en pause in-game
  - Résolution d'écran
  - Format d'écran
  - etc.
- Performances

- 60fps constant sur la plateforme cible (Intel Core i7-4710MQ, 16 Go RAM DDR3, NVIDIA GeForce GTX 880M)

## Features spécialisées

Chaque groupe « non débutant » devra choisir **un minimum de 1 feature spécialisée** au choix dans la liste suivante :

- Multijoueur en réseau (planification sur le serveur) (60fps avec 2 clients + 1 serveur sur la même machine)
- Version mobile Android (60fps sur un Oneplus One)
- Version mobile iOS (60fps sur un iPhone 4s)
- Planification multi niveaux
- Apprentissage artificiel des heuristiques

## Durée d'une partie

Pour permettre au jury de pouvoir évaluer correctement le jeu produit il sera nécessaire de montrer en peu de temps le déroulé d'une partie. Trois options s'offrent à vous :

- Vous baser sur un format court de partie (3-5 minutes)
- Proposer un mécanisme pour avancer dans la partie plus rapidement lors de la soutenance (sauvegarde/chargement in-game)
- Cheat codes

## Plateformes Cibles

La plateforme cible : **PC/Windows (sauf feature spécialisée)**

## Programmation d'outils

En dehors du rendu du jeu proprement dit, il sera demandé pour les groupes « non débutant » le développement **d'un minimum de 1 outil** (pouvant prendre la forme de scripts/plugins pour Unity ou de programmes complètement indépendants) ayant pour but de faciliter la conception du jeu.

Ex : génération automatique de niveau, génération de scènes de test, éditeur de niveau respectant des contraintes liées au gameplay, etc.

## Outils

### Moteur de Jeu

L'utilisation d'Unity comme moteur/environnement de développement est requise.

### Scripting

### IDE

L'utilisation de Microsoft Visual Studio ou Monodevelop est vivement conseillée, de même que l'utilisation du langage C# pour le scripting, il sera également conseillé de ne pas hésiter à avoir recours à l'utilisation de plugins natifs dans le but d'obtenir de meilleures performances.

### Coding Style

Tous les scripts développés devront respecter un **coding style** identique. Il est envisageable de reprendre des coding styles classiques et existants tels que :

- [http://www.codeproject.com/KB/cs/c\\_coding\\_standards.aspx](http://www.codeproject.com/KB/cs/c_coding_standards.aspx)
- <http://csharpguidelines.codeplex.com/releases/view/46280>

Dans tous les cas, le code produit devra être propre (notamment exempt de traces de debug), et **commenté**.

### Versionning

L'utilisation de Git SCM est **obligatoire**. De même, il est vivement conseillé de produire régulièrement des archives de l'intégralité du projet en cours et de dupliquer les supports de sauvegardes dans le but d'éviter les traditionnelles pannes de disques durs au moment des soutenances.

### Sources

L'intégralité des sources devront être accessibles au moment du rendu final et à **chaque jalon** dans le but de pouvoir contrôler la bonne application des contraintes présentes dans ce document.

### Rendu

Chaque rendu sera effectué via le pull de la branche master du dépôt Git par l'enseignant. Tout retard et/ou problème sera sanctionné par une perte de points.

### Authoring

Dans le but de pouvoir clairement **identifier le travail de chacun** au sein du groupe de projet, tout fichier produit devra porter soit dans son nom, soit dans son entête, le nom de son auteur ainsi que son numéro de version. Si cela se révèle être problématique pour le nommage de certains assets, il est aussi envisageable de créer un fichier texte récapitulatif, décrivant clairement pour chaque fichier produit la contribution de chacun.

### Réutilisation de code / bibliothèque existant

A priori, **aucune utilisation de bibliothèque existante** en dehors de l'API de Unity et du SDK de Mono **ne sera autorisée**.

La réutilisation de code existant ne sera tolérée que dans les cas suivants :

- Fonctionnalités d'import/export d'assets aux **formats exotiques** et non supportés tels quels dans Unity.
- Utilisation de **shaders** particuliers pour un effet artistique **nécessaire** au jeu proposé.
- Utilisation de **camera effects** particuliers pour un effet artistique **nécessaire** au jeu proposé.

Une exception est cependant faite pour l'utilisation des plugins Unity3DRider, UniRx et de l'AssetBundleManager.

### Références

<http://aigamedev.com/open/interviews/mario-ai/> , <http://aigamedev.com/open/review/planning-in-games/>, ...

Description de l'étape de suivi (et éventuellement date)	Livable(s) à valider
1 – Alpha	<p>Version comprenant la majorité des éléments de Gameplay du produit multimédia.</p> <p>Dates proposées :</p> <ul style="list-style-type: none"> <li>4A : Mercredi 26 Octobre 2015</li> </ul> <p>Livrables :</p> <ul style="list-style-type: none"> <li>Exécutable PC/Windows, Présentation Démo</li> <li>Sources (répertoire du projet Unity)</li> <li>Bref rapport présentant la TODO list</li> </ul>
2 – Soutenance Publique	<p>Présentation <b>technique</b> du jeu devant la classe.</p> <p>Dates proposées :</p> <ul style="list-style-type: none"> <li>4A : Vendredi 28 Octobre 2015</li> </ul> <p>Livrables :</p> <ul style="list-style-type: none"> <li>Présentation du projet (Powerpoint)</li> <li>Démonstration pertinente et ciblée</li> <li>Exécutable PC/Windows</li> <li>Sources (répertoire du projet Unity)</li> <li>Bref rapport postmortem</li> </ul>

### Soutenance

Durée de présentation par groupe (hors questions réponses et débriefing)	Groupe de 3 : 20 minutes Groupe de 4 : 25 minutes		
Type de soutenance :	Présentation / PowerPoint	Démonstration	Autre
<i>Cocher la(les) case(s)</i>	X	X	
Audience :	A huis clos	Devant la promotion	Publique
<i>Cocher la case</i>			X