

SYLLABUS PROJET

Nom du projet :	Projet Annuel : Développement d'un Jeu Vidéo Planification/Simulation Multijoueur
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 :	pro@nicolasvidal.fr
--------------	---------------	------------------------	--------------------------------------------------------------

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 :	25h
---------------------------------	-----	-----------------------------------------	-------	------------------------------------------------	-----

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

Réaliser un premier Jeu Vidéo complet, avec l'objectif de pouvoir réutiliser la majeure partie des assets développés pour une éventuelle production plus sérieuse (pipeline d'intégration générique, code réutilisable, prefabs génériques, ...).

Objectifs pédagogiques complémentaires

Réaliser un jeu vidéo complet sur mobile avec Unity, pouvant être joué sur une durée infinie.

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://en.wikipedia.org/wiki/Turns,_rounds_and_time-keeping_systems_in_games#Simultaneously_executed_and_clock-based_turns

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 Mobile multijoueur doté d'une phase de planification simultanée et d'une phase de simulation. 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 devra être composé d'une phase de planification et d'une phase de résolution. Exemples :

- <http://www.frozensynapse.com/> ou <http://www.frozensynapse.com/>
- <http://www.robozzle.com/> (à imaginer en multijoueur)
- ...

Le mode **multi-joueurs** sera le mode standard de jeu.

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 chercher à innover, de se démarquer des grands classiques. Ainsi le facteur différenciant du projet devra être mis en valeur et ne pas être anecdotique.

Gameplay

Le jeu devra proposer une interface de **planification d'actions**, ainsi qu'une visualisation synchronisée de la **résolution de ces actions**.

Le multijoueur peut être pensé **coopératif ou compétitif**.

Les interactions doivent être compatibles Touch et Clavier Souris

Features obligatoires

- Menu, options, etc.
- Crossplatform : Le jeu devra pouvoir être joué de manière indifférente sur Mobile (Android/iOS) et Desktop (Windows/OSX)
- Espace d'attente, de connexion entre joueurs et paramétrage de partie (Lobby)
- Mode multijoueur 1vs1 ou 1coop1
- Robustesse
 - Mise en pause in-game
 - Reconnexion partie en cours
 - Résolution d'écran
 - Format d'écran

- etc.
- Performances
 - 60fps constant sur la plateforme cible (Core2Duo T9900, NVidia GeForce 260M, 8Go Ram)
 - Pas de lag observable sur une connexion ADSL classique (débit maximum ciblé montant et descendant : 100 ko/s)

Features spécialisées

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

- Mode partie contre/avec l'ordinateur
- Multijoueur avancé et exotique XvsX et XvsXvsX...
- Enregistrement et mode lecture de Replay
- Serveur de jeu global et matchmaking automatique selon des critères explicites
- Réseaux sociaux : mise à jour de statut Facebook, Twitter, utilisation de l'OpenGraph Facebook, etc.
- Level Editor : proposer un éditeur de niveau en dehors de l'éditeur d'Unity avec la possibilité de jouer sur ces niveaux personnalisés
- Génération de niveaux automatique

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 (5-8 minutes)
- Proposer un mécanisme pour avancer dans la partie plus rapidement lors de la soutenance (sauvegarde/chargement in-game)
- Cheat codes

Plateformes Cibles

Les plateformes cible sont multiples : **PC/Windows ou OSX et Android ou iOS**

Programmation d'outils

En dehors du rendu du jeu proprement dit, il sera demandé 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, vérification de la cohérence des sauvegardes, visualisation de l'équilibre des factions si plusieurs, 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.

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://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.

Références

Frozen Synapse, Robozzle, ...

Livrables et étapes de Suivi

Description de l'étape de suivi (et éventuellement date)	Livrable(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"> 4A : Jeudi 18 Décembre 23h59 (3DJV1), Mardi 16 Décembre 23h59 (3DJV2) <p>Livrables :</p> <ul style="list-style-type: none"> Exécutable PC/Windows, apk Android, Présentation Démo Sources (répertoire du projet Unity) permettant de Build sur un téléphone Android et sur Windows Bref rapport présentant la TODO list
2 – Bêta	<p>Version quasi-complète, incluant l'intégralité des assets, pouvant posséder quelques bugs mineurs.</p> <p>Dates proposées :</p> <ul style="list-style-type: none"> 4A : Lundi 09 Février 23h59 (3DJV1), Mercredi 11 Février 23h59 (3DJV2) <p>Livrables :</p> <ul style="list-style-type: none"> Exécutable PC/Windows, apk Android, Présentation Démo Sources (répertoire du projet Unity) permettant de Build sur un téléphone Android et sur Windows Bref rapport présentant la TODO list
3 – Soutenance Publique	<p>Présentation 'commerciale' du jeu devant un public ouvert et hétérogène.</p> <p>Dates proposées :</p> <ul style="list-style-type: none"> 4A : Jeudi 05 Mars 14h00 (3DJV1), Jeudi 05 Mars 10h00 (3DJV2) <p>Livrables :</p> <ul style="list-style-type: none"> Présentation du projet (Powerpoint) Démonstration pertinente et ciblée

Soutenance

Durée de présentation par groupe (hors questions réponses et débriefing)			
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