Technical Design Document (T.D.D)

ITESCIA / CK Mai 2016

*Ci joint un squelette de document technique à remplir. Le TDD s’écrit en général pendant la prototypage et se termine pendant la “pré production”. Les prototypes laissent la possibilité de faire des essais techniques sur les points les moins clairs (pendant que les designers font des tests de gameplay sur un prototype) et ainsi de remplir completement le document.*

# Presentation du jeu

*Resumé rapide du type de jeu et des points importants en lien avec technique.*

# Plateformes et spécifications hardware

*Liste des plateformes cibles et configurations si besoin (sur PC, config mini, et conseillée).*

# Environnement de developpement

*Outils utilisés pour coder, debugger et versionner. (Visual Studio, etc …)*

# Technologie

*Moteur utilisé et sa version (elle pourra évoluer).*

# Pipeline de production

*Liste de toutes les données qui sont produites et la façon dont elles sont traitées pour arriver jusque dans le jeu. Donner des détails sur les formats et sur les réglages pour chaques données (par exemple animations au format FBX, avec nn clés par seconde).*

# Outils spécifiques

*Liste les outils spécifiques qui seront utilisés ou alors programmés pour l’occasion.*

# Libraries Externes (si besoin)

*Si le projet utilise des librairies externes, les lister ici.*

# Architecture Logicielle

*Schema et explication des differents modules qui composent le jeu (et les liens entre eux). Faire l’architecture dans un style “top-down” avec un “game manager” au sommet de votre architecture. (ne pas rentrer dans le détail ici, juste des gros modules).*

# Techniques de rendu

*Techniques mises en oeuvres pour obtenir le rendu désiré par les spécifications du GDD. (si par exemple le style graphique est en ombrage, comment on le réalise. Si le rendu est en cell shading, comment on le fait.).*

# Budgets

*Poser les limitations dans chaque domaine (nombre de polygones maximum, taille jeu en espace de stockage). Pour cela faire des essais sur la plateforme cible (comme crée une scène de test avec 500 000 polygones et mesurer le framerate).*

# Performances

*Parler ici des performances visées. Certains jeux ou certaines machines ont besoin d’un framerate (Nombre d’images par seconde) élévé ( jeux oculus, FPS, courses …). Certains autres ne necessitent pas d’un haut framerate (jeu de plateaux, casuals…).*

# Mecanismes de gameplay (implementation technique)

*Pour les mécanismes de gameplay un peu complexes, lister ici les solution techniques. (par exemple, on veut choisir des unités dans un RTS avec une selection à la souris, expliquer comment techniquement on recuperer les unités dans le monde 3D).*

# HUD

*Techniques utilisées pour créer les interfaces décrites dans le GDD.*

# Menus

*Techniques utilisées pour créer les menus décrits dans le GDD.*

# Audio

*Moyens mis en oeuvre pour la partie audio.*

# Reseau (option)

*Solution techniques pour les besoins de reseau et de partage sur les reseaux sociaux.*

# Convention d’écriture de code

*Liste de toutes les conventions d’écritures que devront respecter les membres de l’equipe (membres de classes, entetes, commentaires avec initiales et date, etc ….). Si il y a plusieurs programmeurs, cette partie est importante.*

# Equipe et organisation

*Lister les methodes mises en place pour travailler en équipe. Partage du code (versionning ?), partage des données, ….*

# Risques techniques

*Liste des risques techniques au moment de l’écriture du TDD ainsi que les solutions possibles. (par exemple, on ne sait pas si le framerate sera de 75 fps pour l’oculus, donc on prevoit d’optimiser le decor avec des niveaux de détails, LOD).*