**Parties techniques manquantes pour SSWE au sein de Soganatsu Studios France**

Grossièrement **6 parties manquantes**:

* Gestion du son
* Gestion de la physique (moteur physique)
* Gestion des animations
* Gestion de terrains
* Gestion de l’intelligence Artificielle
* Gestion des systèmes de particules

**Gestion du son :**

* Choix de la librairie permettant la lecture de sons
* Définition des formats de sons (MP3, WAV, etc.)
* Placer les sons dans l’espace puis activer, désactiver, rembobiner la lecture d’un son
* Manager les sons en mémoire
* Définir les dépendances dans un graphe de sons (tout comme les nœuds irrlicht)
  + Un son peut dépendre d’un autre son
  + Un son peut avoir plusieurs sons
  + Un son peut dépendre d’un objet
  + Un son peut donc être un nœud Irrlicht spécial

**Gestion de l’intelligence artificielle :**

* Besoin d’étudier le sujet avec une personne de ce domaine

**Gestion de la physique (moteur physique) :**

* Définir le moteur physique
* Possibilité de définir si un objet a des contraintes physiques ou non (Checkbox)
* Créer onglet « Physic » dans la fenêtre d’édition d’un nœud
* Mapper un terrain (géré, ou statique) pour interagir avec les autres objets de la scène
* Définir et pouvoir paramétrer les données liées aux objets contraints (masses, gravité, etc.)
* Définir et paramétrer les différentes contraintes en fonction des objets édités (personnage, sphère, cube, objet quelconque, etc.)

**Gestion de terrains :**

* Création, management (en 2D pour bonne vue globale du dessus) et rendus avancés de terrains complète
* Méthodes de texturing pour le/les terrain(s), matériau spécial comme le « Terrain Splatting », etc.
* Terrains personnalisés :
  + Terrain Painting : Consiste à modifier les hauteurs du terrain. Partir d’un terrain plat de taille x\*x puis contruire des collines, etc. Application du « triplanar mapping shader algorithm », qui permet de personnaliser les textures à appliquer sur le terrain en fonction de la hauteur d’un point donné.
  + Terrain Caves : Consiste à creuser dans un terrain et ainsi pouvoir par exemple creuser des caves, ou des grottes, très important et indispensable pour un gestionnaire de terrain complet, mais peut arriver après la version 1 de SSWE
  + Fournir des matériaux par défaut pour les terrains
  + Donner possibilité de partager un terrain en plusieurs parties soit plusieurs matériaux, pour un terrain.
  + Définir zones d’eau (fluides) pour exemples : chuts d’eau, lacs, etc.
  + Gestion de la végétation (arbres, herbe, etc.)
    - Définir comment animer les arbres, herbe, etc. et comment les construire.
    - Développer l’outil d’animation
    - Développer outils physiques liés au vent, etc.
  + Définir un randomizer de terrain :
    - Génération de terrain aléatoire
    - Génération aléatoire de végétation en fonction du terrain etc.
  + Définition de chemins sur un terrain, chemin à pieds, ou voiture, ou routes également, tous types.
  + Travailler sur les cascaded shadow maps pour couvrir les ombres de grands terrains.
  + Pouvoir définir zones de blocage.
  + Pouvoir exporter et importer le terrain
    - Trouver format (XML ?)

**Conclusion**: Je pense une personne à plein temps car c’est une partie sans cesse en développement et demande énormément de boulot.

**Gestion des animations :**

* Animation des modèles animés
  + Donner possibilité d’utiliser des plugins qui iront compléter les outils existants.
  + Pouvoir utiliser des outils grand public : Kinect par exemple pour les personnages
  + Pouvoir communiquer avec des clients lourds, exemple avec Blender ou 3DS MAX
  + Animation (assistée ou non) manuelle
  + Format d’exportation et d’importation des animations
* Création et édition des « animators »
  + « Follow Path Camera » : Animer la position de la caméra courante de la scène, peut être utile pour les cinématiques temps réel.
  + Création et édition de différents « animators » (rotation automatique, « A-B paths repeat », etc.
* Animations issues de vidéos
  + Créer lecteur de vidéos (cinématiques), transformateur « image video-texture »
  + Édition de « real time cinematics »
    - Afficher en vidéo via transparence en plein rendu 3D, afficher des titres par exemple, ou des crédits
  + Appliquer une vidéo sur un matériau comme étant une texture animée par exemple
* Jouer un son lors d’une vidéo
* Animer un son

**Gestion des systèmes de particules :**

* Création et édition des systèmes de particules
  + Création assistée et graphique sous forme de liens nœuds à nœuds
  + Prévisualisation temps réel des modifications (viewport)
* Systèmes de particules par défaut
  + Pluie
  + Feu
  + Neige
  + Fumée
  + Feuilles
  + Poussière
  + Brouillard
  + Explosion
  + Tonnerre
  + Tornade
  + Poils
  + Jets d’eau
  + Sable (désert)

**Conclusion :**

En conclusions je dirais que je peux m’occuper du son, des systèmes de particules, un peu de la physique et un peu des animations. Ça fait 3 parties sur 6 que je peux assurer. Le reste me demande d’aller plus loin et ça tout dépend de mon temps disponible. En revanche la gestion de terrains peut-être trop longue pour moi et l’intelligence artificielle est un domaine que je ne connais absolument pas et très compliqué.

Je suis capable d’estimer mes trois parties à environ 6 mois de travail étant donné le fait que je connais déjà les librairies (open-source) que je vais utiliser, comment je vais les utiliser.

En outre, un développeur 3D (domaine du jeu vidéo si possible) pour la gestion de terrains (partie en constante innovation + amélioration) et un autre développeur pour l’intelligence artificielle. C’est le minimum si on veut pouvoir couvrir le maximum de fonctionnalités au sein du logiciel.