Rencontre d’évaluation mi-mandat

# Titre : Astral Rétribution

Nom étudiant 1 : William Caron

Description fonctionnalité : Génération aléatoire de planète avec des chemins permettant le parcours d’un niveau. Dans notre style de jeu rogue like, il doit exister une certaine progression linéaire pour marquer l’avancement du joueur. Cette fonctionnalité garde en mémoire l’emplacement des planètes créer à sa première instance et fait la création de chemin qui permet au joueur de visualiser son déplacement. A chaque retour de l’usager dans le menu principal l’affichage est de nouveau adapté à sa progression.

Plus grand défi : Unity a de la difficulté à garder en mémoire les GameObjects qu’il possède dans une scène. C’est pourquoi, lors d’un aller-retour de scène, il faut utiliser un DontDestroyOnLoad. Dans le cas des planètes il faut donc pour garder en mémoire les emplacements ainsi que leurs rotations. Par la suite, il suffit de le réinstancier de nouveau à leur même position.

Référence code (no ligne, nom fichier) : PlaneteManager.cs : tous le fichier, Planete.cs : #19-22, MenuHub.cs : #25-34, #53-58

Développement important à venir : Pathfinding de chaque membre d’équipage, pour permettre leur déplacement. La gestion des évènements de chaque planète sera aussi une autre implémentation faite.

Plus grand défi évènement à venir : Gestion des membres d’équipage dans le pathfinding fait par l’utilisation de Dijkstra. En cas de manque de temps celui d’unity sera utilisé, mais reste un grand défi d’implémentation dans notre projet.

Nom étudiant 2 : David Demers

Description fonctionnalité : Génération de vaisseau aléatoirement en utilisant une génération procédurale, on utilise des tuiles qui représente des murs ou des sols et on fait une partition binaire de l’espace pour faire des salles de forme rectangulaire.

Plus grand défi : générer aléatoirement des tuiles selon des paternes, des portes au milieu des salles, les limites du vaisseau.

Référence code (no ligne, no fichier) : ShipManager.cs : tout le fichier, BSPTree.cs : tout le fichier

Développement important à venir : Faire le menu combat, c-à-d de faire un vaisseau qui peut jouer des cartes, bouger les membres d’équipage, notion de tour par tour, AI de l’ennemi, gestion d’objets, etc.

Plus grand défi évènement à venir : Faire un FSM qui permet de changer entre chaque état, sans avoir de problème, car il y a beaucoup d’évent à prendre en compte.

Progression globale du projet synthèse :

40%