[Rattrapage] EVIJV Rapport Final

Kévin TEBIB

September 2017

Contents

1 Introduction

Le but de ce projet est de concevoir et prototyper un jeu en environnement virtuel interactif sous Unity 3D. Ce prototype se focalisera sur deux aspects:

- l'interaction du joueur avec son environnement
- l'intelligence artificielle des différents PNJ

Ce prototype se nommera **AgeQuodAgis**. Ça correspond à une expression latine, littéralement traduite par : *fais ce que tu fais*, à comprendre *fais bien ce que tu fais*. AgeQuodAgis invite donc à découvrir par soi-même un environnement vivant peuplé de créature étrange. Comme le souligne justement la traduction du nom du jeu, les actions du joueur influenceront le comportement des PNJ et pourront altérer ses chances de victoire. Il devra donc assumer les conséquences de ces actes.
Le rapport commence tout d'abord par une présentation des différentes hypothèses considérées, puis par une présentation du scénario que le joueur devra suivre. Nous continuerons par une présentation des différentes mécaniques de gameplay développer dans le cadre de ce projet et nous conclurons par les résultats obtenus au terme de celui-ci, ainsi que ses différentes perspectives d'amélioration.

2 Hypothèses

Nous avons déjà évoqué que le jeu serait réalisé à partir d'Unity 3D. Il s'agit d'un moteur de jeu multi-plateforme (smartphone, Mac, PC, consoles de jeux vidéo et web) développé par Unity Technologies. Il est l'un des plus répandus dans l'industrie du jeu vidéo, du fait de sa rapidité aux prototypages pour les très gros studios, aussi pour la sphère du jeu indépendant qui développe directement dessus pour sortir leurs applications sur tout support.
Unity 3D a la particularité d'utiliser un éditeur de script compatible mono (C#), UnityScript (un langage proche du JavaScript et inspiré d'ECMAScript) et Boo au lieu de Lua très utilisé dans les jeux vidéo. Son approche orientée asset, par le biais d'un EDI dédié, le différencie des moteurs comme le Quake engine dont les éléments centraux sont les codes sources. Il est l'équivalent du logiciel de création Director pour la 2D qui utilise Lingo. Il se rapproche plus pour la 3D des logiciels tels que Shiva, Virtools, Cheetah3D. Parmi les logiciels d'animations, il ne permet pas la modélisation mais permet de créer des scènes supportant des éclairages, des terrains, des caméras, des textures, la musique et les vidéos. Il est par ces fonctionnalités un mélange de VRML et de QuickTime. Il a la particularité de proposer une licence gratuite dite *Personal* sans limitation au niveau du moteur.

RAIN est un ensemble de moteurs et d'outils gratuit créés par Rival Theory pour ajouter de l'intelligence artificielle aux jeux et aux simulations construits

dans Unity. Il combine tous les aspects du développement de l'IA dans les jeux et les médias interactifs dans une expérience de conception transparente. RAIN est le moteur d'IA le plus utilisé pour Unity, avec une communauté en pleine croissance proche de 100 000 développeurs.
La fonctionnalité majeure de RAIN est divisée en 3 parties principales:

Des outils pour créer des personnages individuels: Permet de construire des personnalités auto-dirigées qui produisent des mouvements et des actions basées sur leur état et leur raisonnement internes.

Des outils pour modifier l'environnement : Mise à disposition de nombreuses fonctionnalités afin que l'IA puisse comprendre et interagir avec le monde qui l'entoure.

Les extensions de développeur: Ajout possible de script servant à compléter, voire même surpasser les fonctionnalités intégrées.

Ce prototype étant prévu qu'il fonctionne sur pc. L'avatar du joueur en vue à la première personne sera contrôlé à l'aide du clavier et de la sourie du pc de celui. D'autres part le manque de temps ainsi que la notation ne se focalisant ni sur l'aspect visuel et auditif du joueur ni sur la cohérence des animations (particulièrement la gestion des rotations qui sont trop longues à traiter vu le temps qu'il me reste). Il a été décidé que ce prototype se concentre sur les mécaniques de jeu permettant l'interaction avec le joueur et les différents comportements des PNJ. Bien sûr, il sera aussi fait mention des autres scripts permettant le bon fonctionnement du jeu.

3 Scénario du jeu

Dans le but de divertir et de contrôler le jeu au cours de sa partie. Un scénario a été mis en place pour ce prototype. Il a pour vocation de mettre en avant les différents scripts codés. Le scénario est décomposé en 5 actes. Chaque acte contiendra une quête que le joueur devra réaliser pour progresser.

Acte 1 : Une rencontre inespéré

Le joueur se réveille dans un lieu inconnu sans eau, sans nourriture. S'il se retrouve totalement déshydraté ou affamé alors c'est la mort assurée. Pour échapper à son sort, il devra avancer sur l'unique chemin, se présentant à lui. Le joueur s'engagera donc dans une grotte, au fur à mesure qu'il avance des parois légèrement entrouvertes, se refermeront.Plus loin dans la grotte, le joueur trouvera un passage qui ne refermera pas. Le joueur devra s'y engouffrer pour avancer dans l'histoire.
Il arrivera ainsi devant un champ de fleurs. Au milieu de ces fleurs, il découvrira une créature ressemblant à un tubercule faisant partie du peuple des "Veggie". Le joueur a la possibilité de communiquer avec cette créature à l'aide d'un arbre de dialogue. Malheureusement pour lui, le veggie parle un langage étranger,

rendant ainsi la communication impossible. Pour pouvoir avancer dans le jeu, le joueur devra donc remarquer le cercle de fleur incomplète sur le sol de la caverne. Il devra donc récupérer les fleurs du champ qui lui permet de compléter le cercle. Si le joueur a une des fleurs manquante en main alors le veggie suivra le joueur jusqu'à ce qu'il lui donne la fleur ou qu'il la lâche. Une fois la fleur donné, le veggie s'approchera du cercle de fleur et ajoutera la fleur au cercle. Le joueur devra recommencer pour toutes les fleurs manquantes.
Une fois le cercle complété, le veggie s'approchera du joueur et posera une pomme au sol. Si le joueur la ramasse alors il la consommera automatiquement. En la mangeant, il pourra ainsi parler à tous les veggies du jeu, sans être mis en difficulté par la barrière de la langue. Ensuite le joueur devra discuter avec le veggie. Il apprendra que la pomme était sa récompense pour avoir reconstitué le cercle de fleur. De plus sa reconstruction a permis l'ouverture de la porte du village des "Veggies". Le veggie guidera le joueur au village des "Veggies".

Acte 2 : Une quête pour la survie

Une fois arrivé au village des "Veggies", le joueur pourra observer la présence de 5 veggies dans le village. Les veggies se déplaceront continuellement sur des chemins prédéfinis dans le village. Ces chemins seront différents pour chaque veggie. Le joueur aura la possibilité d'interpeller n'importe quel veggie. Celui-ci s'arrêtera et commencera à discuter avec le joueur à l'aide d'arbre de dialogue. Lors des discussions du joueur avec les veggies. Un sujet est récurrent. Il s'agit du peuple des loups qui leur a volé leur réserve de nourriture entraînant donc une famine pour les veggies. Un des veggies aura dans ces dialogues, la demande d'aide pour récupérer la nourriture chez les loups. Quand le joueur aura accepté, le lieu ouvrir le passage vers la tanière des loups.
Le joueur devra donc aller à la tanière des loups, s'y engouffrer et récupérer la nourriture des veggies. Pour corser légèrement la difficulté de cette quête. La tanière sera en fait un petit labyrinthe dans lequel des loups patrouillent. La patrouille des loups sera évidemment prédéfinie pour que le joueur puisse apprendre leurs déplacements et accomplir la quête. Si un loup voit le joueur, il le poursuivra. S'il arrive à toucher le joueur, alors celui-ci sera téléporter à l'entrée de la tanière et devra tout recommencer. Bien sûr si le joueur est touché par un loup quand il possède la nourriture alors le joueur sera bien ramené à l'entrée et la nourriture sera reposée dans la réserve de la tanière.
Une fois la nourriture récupérée et le joueur sorti de la tanière, il devra simplement la rendre au veggie lui ayant donné la quête.

Acte 3 : Un problème de nuisible

Les veggies remercieront le joueur pour avoir ramené la nourriture au sein du village. Ensuite ils évoqueront un nouveau problème qui les préoccupe. Il s'agit de la présence d'un prédateur qui se cache au sein des cavernes. Il représente une menace constante pour les veggies. Un veggie demandera au joueur d'exterminer ce prédateur. En acceptant cette quête, le lieu ouvrira un nouveau chemin dans

la caverne menant directement à ce prédateur.
En empruntant ce nouveau chemin, le joueur est presque directement confronté au prédateur décrit par les veggies. Il s'agit un grand et fier dragon. Le joueur ne possédant pas d'arme. Il est se retrouve dans l'incapacité apparente de tuer le dragon par la force. Le joueur devra observer son environnement et repérer que le dragon se trouve sur une plateforme suspendue au-dessus du vide. Il devra aussi repérer que le dragon essaie de protéger une dalle bizarre. Il s'agit en fait d'un capteur de pression que le joueur devra actionner pour faire tomber dans le vide la plateforme sur laquelle se trouve le dragon. Dans le but de protéger sa vie, le dragon aura la possibilité de cracher du feu et d'envoyer des boules de feu. Il crachera du feu quand le joueur est assez proche de lui alors qu'il enverra ces boules de feu quand le joueur n'est pas à porter. Pour atteindre la plaque de pression, le joueur devra donc esquiver les attaques du dragon.
Une fois le dragon tué, il devra retourner valider la quête auprès d'un veggie.

Acte 4 : La négociation

Maintenant que les problèmes menaçant la vie des veggies ont été résolu. Le joueur va se retrouver sans quête. Il devrait reparler naturellement aux différents veggies du village. Il constatera de nouveaux dialogues sont disponibles chez 4 des 5 veggies présents dans le village. Quant au 5éme veggie, le joueur constatera que son interface de discussion a changé. En effet, il s'agira une boîte de dialogue permettant d'écrire du texte au clavier.
Dans cet acte, le but du joueur sera de négocier avec le veggie ayant la boîte de dialogue pour qu'il ouvre un chemin vers la sortie de caverne. Pour mener à bien cette négociation, il devra employer les bons mots-clés dans le bon ordre. Les mots-clés seront cachés dans les dialogues des 4 autres veggies. Une fois qu'ils sont donnés au veggie alors il ouvrira un nouveau chemin pour le joueur.

Acte 5 : Épilogue

A la fin de l'acte 4, le veggie ouvrira un chemin que devra parcourir le joueur pour sortir de la caverne et ainsi finir le jeu. Cependant il ouvrira en fait un des deux chemins possibles qui correspond aux deux fins du jeu. A partir de l'acte 2 en plus des dialogues nécessaires pour résoudre les quêtes, chaque veggie posera des questions aux joueurs. Il lui sera donné à chaque question exactement 3 réponses possibles. L'une correspondra à que le veggie considéra comme une réponse correcte, une autre comme une réponse fausse et la dernière sera considéré comme une réponse neutre.
L'utilité de ces questions peut être comprise si le joueur regarde les yeux des veggies au cours du jeu. En effet leurs yeux indiquent à quel point, il apprécie le joueur. Les yeux de chaque veggie peuvent prendre les couleurs suivantes: rouge, orange, jaune, vert et bleu. Le changement de couleur ne peut être effectué uniquement pour une couleur voisine de sa couleur actuelle. Par exemple si le veggie a les yeux jaunes, alors il ne pourra que passer en orange ou en vert. Symboliquement, plus les yeux d'un veggie sont proches du bleu, plus il

appréciera le joueur. De la même manière, plus ils sont proches du rouge, plus il détestera le joueur.
Le choix du chemin qui sera ouvert par le veggie se fera tout simplement en regardant l'appréciation générale du joueur auprès des veggies. Si elle est suffisante alors le joueur aura le droit à la bonne fin. Sinon il aura le droit à la mauvaise. La différence entre la bonne fin et la mauvaise fin consiste au type de chemin que devra le joueur pour sortir de la caverne. S'il a accès à la bonne fin, alors il devra parcourir un chemin verdoyant et sans la moindre difficulté. Sinon le chemin sera rempli de piège mortel que le joueur devra éviter pour terminer le jeu.

4 Mécanique de gameplay

Les fonctionnalités développées pour pouvoir créer le jeu peuvent être réparties en 3 sous-ensembles.

Interaction : Permet au joueur d'interagir avec son environnement.

Intelligence artificielle : Permet de définir le comportement des différents PNJ.

Autre : Les fonctionnalités ne pouvant être classées dans les deux sous-ensembles précédents.

4.1 Interaction

Les déplacements du personnage

Le joueur a été modéliser à l'aide du prefab standard "FPSController". Ce préfab permet au joueur d'expérimenter une vue à la première personne grâce au placement de la caméra suivant parfaitement les mouvements du modèle 3D qu'il fournit. De plus il permet une gestion complète des mouvements de la caméra, ainsi que des déplacements de l'agent incluant aussi la possibilité de sauter.

La manipulation des objets

Dans le but de manipuler des objets, un script de "Raycast" a été mis en place. Le Raycast permet au joueur de sélectionner un objet se trouvant en face de son curseur. Si la distance entre le joueur et l'objet qu'il souhaite sélectionner est trop importante alors la sélection ne sera pas activé.
Si l'objet sélectionné contient le script "Saisissable" alors l'agent pourra le saisir à l'aide d'un clic gauche. Une fois la saisie effectuée, l'agent aura la possibilité de se balader tout en tenant l'objet. S'il refait un clic gauche sur la souris alors il lâchera l'objet.

Champs visuel des PNJ

Le champ visuel de tous les PNJ sera géré par l'asset de RAIN AI. Il permet aux PNJ d'avoir en permanence un senseur visuel couvrant un angle de 126 horizontalement et un angle de 66 verticalement. Si un objet définis comme intéressants dans l'arbre de comportement du PNJ entre dans le champ vision du PNJ alors il pourra déclencher les comportements associés.

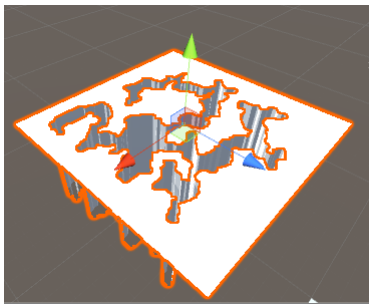
Les arbres de dialogue

Bien que pas intégré dans le jeu par manque temps, l'algorithme de l'arbre de dialogue existe. Il s'agit tout simplement un arbre où chaque noeud contient option de dialogue et dont les arêtes relient des différentes options de dialogue en elle-même.

4.2 Intelligence artificielle

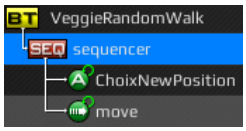
L'automate cellulaire

Dans le but de construire le décor du jeu, nous avons réalisé un script éditeur capable de générer procéduralement des cavernes cohérentes visuellement.
Le script en plus de modifier l'interface d'Unity 3d en ajoutant un menu ainsi qu'une nouvelle fenêtre lorsqu'on clique sur l'option du menu rajouté. Il va aussi générer un cube modifié de la manière suivante : Après avoir agrandi le cube, l'algorithme va créer des régions à l'intérieur même du cube. Évidemment comme tout automate cellulaire, le nôtre suit une règle de propagation qui est la suivante : $voisins > 4$. Une fois la propagation finie, nous éliminons le bruit pour obtenir des régions bien définis. Ensuite nous relierons les différentes régions entre elles. Cela nous permet d'obtenir des cavernes plutôt réalistes sans passer un temps infini à construire les cavernes à la main. Voici un exemple de caverne qu'on peut générer avec l'automate cellulaire :



Déplacement Aléatoire

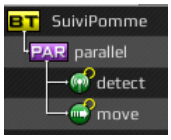
Nous avons utilisé un arbre de comportement de Rain AI pour créer ce comportement.



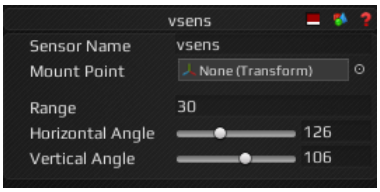
Il est composé d'une action personnalisé *ChoixNewposition* qui génère aléatoirement un vecteur3 correspondant à la nouvelle position souhaitée de l'agent. Ensuite l'action *move* déplacement l'agent vers cette nouvelle position. Ces deux actions seront exécutées de manière séquentielle. La nouvelle position sera stockée dans la mémoire de l'agent à l'aide du module de mémoire de RAIN AI. Cette information restera en mémoire le temps nécessaire à l'agent pour exécuter le déplacement.

Suivi d'un gameObject

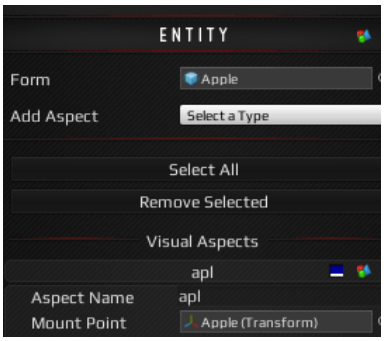
Nous avons utilisé un arbre de comportement de Rain AI pour créer ce comportement.



Il est composé de deux actions qui s'exécutent continuellement et parallèlement. L'action *detect* utilise un capteur visuel défini dans RAIN AI.



Comme vous pouvez le voir, nous pouvons choisir l'angle horizontal et vertical du capteur vision ainsi que distance jusqu'à laquelle les objets peuvent être vus par celui-ci. Le capteur *vsens* peut détecter uniquement les objets ayant un aspect visuel spécifique reconnaissable par le capteur. Donc pour que le capteur puisse reconnaître un objet, il a besoin que celui-ci possède une entité de RAIN AI ayant un aspect visuel.



Grâce à cette mécanique, on peut faire suivre à n'importe quel PNJ, n'importe quel GameObject que ce soit une pomme ou le joueur.

4.3 Autres mécaniques

Des parois mobiles

Nous avons rajouté des murs à la caverne générée par le script de l'automate cellulaire. Le but de ces murs est de délimiter et d'interdire temporairement au joueur certaines zones. Au fur est à mesure que le joueur progressera dans l'histoire, ces murs s'ouvriront au fur et à mesure. Pour renforcer l'impression de réalisme du joueur, nous lui montrons que les murs de la caverne bougent. Ce déplacement est fait à l'aide d'un script qui déplacement un mur vers le bas à une vitesse constante. Pour éviter que le mur descende continuellement, nous l'avons bloqué une fois que la position en y du mur atteint une certaine valeur.

Téléportation d'un gameObject

Un script capable de changer de place instantanément un gameObject. Malgré la simplicité de ce script, il n'en rien reste pas pour le moins pratique. En effet le joueur sera par exemple remis au début des épreuves de l'acte 2 et 3 quand celui-ci subira des dégâts.

5 Résultats obtenus

En plus des scripts décrit dans la section précédente, une caverne qui a été généré à l'aide de l'automate cellulaire. De plus, trois types de PNJ sont présents dans l'environnement. il y a un dragon, un loup et un veggie (une sorte de racine avec un visage. Il ressemble un peu à un poire). Le loup et dragon sont capables se déplacer aléatoirement quant au veggie il peut suivre une pomme.
Les scripts décrit ci-dessus sont tous codés mais qu'un certain nombre sont intégrer à la scène du jeu.

6 Conclusion

Pour conclure, nous avons codé un prototype incomplet de notre jeu. Malgré cet état de fait, la scène contient déjà quelque élément de gameplay, d'interaction et d'IA. Une partie des éléments manquant ne sont pas intégrés au jeu par manque de temps.
Si on souhaite continué ce projet, il suffirait de finir d'intégrer les scripts déjà pré-faits. Et ensuite ajouter les scripts manquants.