



Again and again

Rapport EVIJV

Bargaoui Emna, Farjas Emilie

Sommaire

I.Les outils manipulés.....	3
II.Le Gameplay.....	3
III.Les réalisations finales	4
IV.Les améliorations	11

I. Les outils manipulés

- Unity
- Unity asset store
- mixamo
- Mono develop

II. Le Gameplay

Contexte du jeu :

Notre joueur principal est coincé dans une boucle temporelle, dans un pays imaginaire où tout est possible, qui l'amène à revivre la même journée de manière infinie. Les personnages de ce monde le guideront dans sa quête pour s'en sortir et lui expliqueront ce qu'il doit faire .

Type :

Exploration /résolution d'énigme

Music :

Intégration d'une bande son pour accompagner le joueur tout au long du jeu .

Challenges :

Sortir de cette boucle temporelle donc résoudre des énigmes, lui permettant d'obtenir des morceaux d'une carte. Une fois la carte entièrement reconstituée avant la fin de la journée, une porte secrète s'activera, lui permettant de retourner en 2018

Récompense :

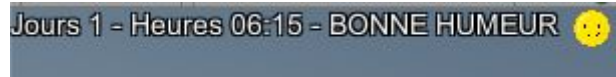
Lorsque le joueur résout des énigmes, ou accomplit certaines interactions(combattre menace ...) il obtient un bout de carte .

Système points :

Chaque énigme apporte un bout de carte.

- doit collecter tous les bouts de carte pour sortir de la loupe temporelle.
- A chaque nouvelle partie (journée) il perd les morceaux de cartes acquises et devra tout recommencer la journée suivante.

Tout au long du jeu le joueur aura accès à l'heure et au jour dans lequel il se trouve, mais également son humeur du jour :



Il peut également visualiser l'inventaire des cartes collectées durant une même journée :



III. Les réalisations finales

A. Interface graphique

Notre monde se compose d'un plateau principal où le joueur devra résoudre ses énigmes et interagir avec l'environnement ainsi qu'avec d'autres personnages.

Voici la partie principale du plateau :



Le deuxième plateau représente la scène de fin, lorsque le joueur aura réussi à sortir de sa boucle temporelle et sera revenu dans son monde actuelle.



B. Les interactions 3D

Voici les différents scripts que nous avons développés :

- Ouverture des portes des maisons par le joueur (il peut les traverser) afin de sortir et d'accéder au monde extérieur pour commencer sa quête .
- Interactions basiques avec la page du menu contenant un lien vers le jeu, l'aide et quit
- Éteindre le feu d'une maison qui brûle



- Déplacement du joueur principal grâce aux flèches du clavier
- Le joueur peut prendre certains objets de la scène avec sa main , les déplacer et les poser au sol .
- Le joueur peut monter une échelle et redescendre
- Le joueur peut tirer des boules de feu sur les menaces



- Le joueur peut couper certains arbres pour l'aider à traverser une rivière
- Le joueur peut traverser un portail lui donnant accès à son monde



- Si le joueur récolte toutes les cartes (cf systeme points) il détruit indirectement une pierre qui scelle un endroit du jeu .
- Interaction avec certains personnages du jeu : affichage de texte lorsque le joueur se situe à proximité:



Nous avons utilisé des animations mixamo pour certaines interactions dans le but de rendre le jeu plus réaliste et amusant .

C. Intelligence Artificielle

Le Behavior Tree

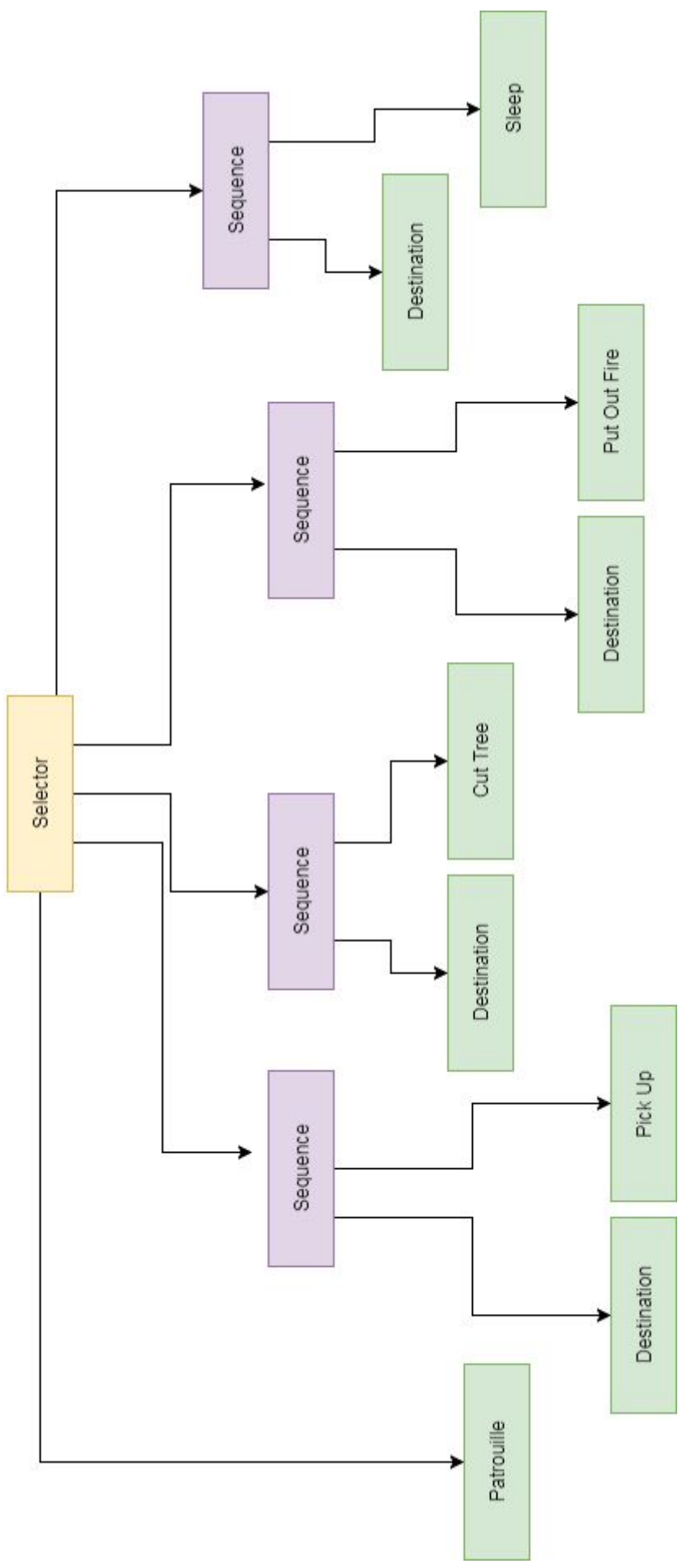
Le comportement des PNJs repose sur un Behaviours tree. Nous avons utilisé un behavior tree plutôt qu'une machine à états. Cela nous a permis un meilleur contrôle du flux de décision effectué par les PNJs.

Nous avons développé notre propre architecture de behavior tree. Cela nous a demandé une phase importante d'analyse pour la création du squelette de l'arbre. Nous avons développé parallèlement les feuilles (les comportements basiques des IAs), dont nous avons besoin. Cela nous a permis de voir les avantages du behavior tree car chaque nouveau noeud de comportement s'y intègre très facilement.

Nous avons implémenté une fonction de priorité, déclenchant les actions des IAs en fonction de certains éléments du contexte et de ceux qu'ils perçoivent de l'environnement .

L'utilisation d'un arbre permet de paralléliser les comportements activés. Mais notre scénario ne nécessite pas ce genre de parallélisme. Un seul comportement s'exécute à la fois.

Behavior Tree



Algorithme Flocking des papillons

Nous avons implémenté un algorithme IA : “Flocking Algorithm”. Nous obtenons un comportement émergent : chacun des agents interagissent entre eux respectant les trois règles de “séparation”, “cohésion” et “alignement” du Flocking. Ceux-ci tournent autour de la tête du joueur et le suivent, simulant une attaque.

Suivi des papillons par les PNJs

Lorsque le joueur se fait attaquer par les papillons, les IAs aident le joueur principal (si celui-ci est de bonne humeur). Les IAs suivent alors les papillons et tentent de les tuer avec une boule de feu.

Coopération des PNJs dans la prise des objets

Lors du comportement de patrouille, les PNJs enregistrent dans leur “sac à dos” les objets qu’ils découvrent au fur et à mesure de leur exploration, dont ils auront besoin plus tard dans le jeu. Lorsqu’ un de leur comportement nécessite l’utilisation d’un objet précis, ceux-ci le cherchent dans leur sac à dos. S’ils le trouvent, ils le réservent et vont le ramasser. Dans le cas contraire, ils se remettent en mode patrouille pour le chercher. Lorsqu’ils le trouvent, ils terminent l’action interrompue précédemment. Un agent central coordonne la disponibilité des objets. Le noeud déplacement gère la communication entre l’agent central et les PNJs.

D. Ebauche du système d’humeur

Le personnage principal a une humeur qui peut varier d’ une journée à l’autre. Celui ci est soit de bonne humeur, soit de mauvaise humeur. En fonction de son humeur, les PNJs auront une attitude différente.

Si le joueur est de bonne humeur, ils vont

- l’aider à tuer les papillons
- ramasser une hache et l’aider à couper l’arbre

Si le joueur est de mauvaise humeur, une fois arrivé sur site, les PNJs vont simuler un comportement agité.

IV. Améliorations

Voici les améliorations intéressantes à implémenter :

- implémenter un système de communication entre les IAs et le joueur principal pour venir en aide à ce dernier (au lieu de la gestion par un agent central)
- implémenter un système de communication local entre les IAs (et non centralisé) : lorsque ceux-ci se rencontrent, ils peuvent s'échanger le contenu de leur sac à dos ou se prévenir si le joueur principal est de mauvaise humeur ou si il a besoin d'aide.
- implémenter un dialogue dynamique entre le joueur et les personnages donnant des indications aux joueur