

Contact : Agnes Delaborde – agnes.delaborde@gmail.com

Dates
Dimanche 10 avril : envoi du code et du rapport (par mail)
Mercredi 13 avril : envoi de la présentation (par mail)
Vendredi 15 avril : soutenance

Objectifs

Mémorisation, fusion d'information. Quêtes.

Exercice 1 – The road is long.

Nous allons ajouter à la mémoire de Gus l'information ayant trait à la distance qu'il a parcourue.

- Dans gus.lua, dans la méthode `memory.addInfo()`, vous avez déjà une conditionnelle pour le domaine de connaissances « NPCs ». Créez un nouveau cas, pour le domaine « environment ».
- La variable « data » (second paramètre de la fonction) sera un tableau {tag, value} contenant l'étiquette « step_taken » (string) et la distance parcourue par Gus à chaque fois qu'il bouge (un float). Dans le tableau `Gus.memory.environment`, ajoutez une paire clé-valeur nommée « step_taken », et qui s'incrémentera de la distance parcourue. La mémoire de Gus devra donc se présenter sous cette forme :

```
Gus.memory = {  
  npcs = {  
    toutes les infos sur les PNJ...  
  },  
  environment = {  
    step_taken = xxxx,  
  }  
}
```

- Réalisez un appel à la méthode `memory.addInfo` à chaque fois que Gus effectue un mouvement.
- Vous avez vu que la méthode `memory.addInfo` appelle `status_board.UpdateContent()`. Allez modifier cette fonction afin que, lorsque Gus n'est pas en train d'interagir avec un PNJ, le `status_board.message` contienne par exemple « Distance parcourue : 0,8 km » (partons du principe que l'unité de déplacement de Gus est le centimètre).

Note : Au même titre qu'il existe une fonction `status_board.getNPCdata()` pour organiser l'affichage des informations relatives aux PNJ, je vous conseille de créer une fonction `status_board.getEnvironmentData()`.

- Ajoutez l'information « wall_hit » à ce domaine de mémoire. Cette valeur représentera le nombre de fois où Gus s'est cogné contre un des murs de la salle.

Exercice 2 – En quête de pouvoir

Attention : si vous avez déjà codé un système de quêtes, je ne vous demande pas de modifier votre code. Adaptez-le (dans la mesure du possible) en fonction des thématiques abordées dans cet exercice.

- Sur le dossier partagé, récupérez le script `quest.lua` et mettez-le à la racine de votre dossier de jeu.

- b. Récupérez l'image Gus_Crown.png dans le dossier /assets/img, et mettez-la au même emplacement dans votre dossier de jeu.
- c. Dans main.lua, chargez le fichier quest.lua juste après le fichier gus.lua (`love.filesystem.load("quest.lua")()`).
- d. Dans main.lua, dans la fonction `love.load()`, affectez la quête `quest_meet_everyone` au PNJ arrogant, et la quête `quest_get_tired` au PNJ timide.

Syntaxe :

```
quest_meet_everyone:assignQuest(NPC1)
quest_get_tired:assignQuest(NPC4)
```

- e. Dans gus.lua, ajoutez ce code à la fonction `Gus.manageInteraction()`, juste avant l'appel à `status_board.UpdateContent()` :

```
if npc.quest ~= nil then
    if not npc.quest:checkCompletion(npc) then
        npc.quest:startQuest(npc)
    end
end
```

Le code signifie : si ce PNJ a bien une quête à proposer, et que cette quête n'est pas encore réussie, alors le PNJ propose cette quête.

- f. Dans quest.lua, repérez et comprenez les conditions nécessaires pour réussir les deux quêtes.
- g. Si vous le jugez nécessaire, modifiez les phrases de prompt afin que le lexique corresponde mieux aux choix que vous avez faits pour chaque PNJ.
- h. Lancez la simulation, allez voir les deux PNJ donneurs de quête, et dirigez Gus pour l'aider à résoudre les quêtes.

Exercice 3 – Chacun sa quête

- a. Créez une quête pour le PNJ agressif : Gus devra s'être cogné au moins 50 fois contre un mur (`wall_hit`).
- b. Créez une quête pour la PNJ charmeuse : Gus doit connaître le profil de tous les PNJ masculins de la salle, être en pleine forme, et être d'humeur très sociable.
- c. Ajoutez une condition à la quête du PNJ arrogant : Gus devra avoir vu *beaucoup de fois* la PNJ charmeuse.
→ Ajoutez dans la mémoire de Gus le nombre de fois où il a croisé chaque PNJ, et également une indication en termes de « jamais vu / pas beaucoup vu / beaucoup vu » (une valeur chiffrée 1, 2, ou 3).

Projet – Propositions relatives à ce cours

Proposition 1 : L'état de fatigue de Gus joue sur ses capacités de mémorisation.

Proposition 2 : Si Gus est trop fatigué, il ne pourra pas accepter les quêtes des PNJ.

Proposition 3 : Dotez Gus d'une mémoire émotionnelle. S'il était d'humeur sociable (ou d'humeur positive si vous avez cette dimension à votre disposition) pendant la discussion avec un PNJ, alors il aura tendance à se rappeler du moindre mot des conversations qu'il a eues avec lui. Par contre, s'il est d'humeur peu sociable, il n'aura rien mémorisé.

Proposition 4 : Jusque là, nous sommes obligés de diriger Gus à la main pour qu'il aille rencontrer les PNJ, et effectuer les quêtes. Créez une quête que Gus pourra résoudre automatiquement, sans que vous n'ayez à intervenir. (Je ne parle pas nécessairement d'implémenter un algorithme de navigation ; il pourra aussi s'agir d'une quête nécessitant un mécanisme de raisonnement logique que Gus pourra effectuer sur place)