

Université de Montréal

Développement et implémentation du jeu des vingt questions

par

Chghoura Karim

Département d'informatique et de recherche opérationnelle

Université de Montréal

Rapport de stage présenté à l'Université de Montréal
en vue de l'obtention du grade de Maitrise
en informatique

Septembre, 2014

© Chghoura karim, 2014

Résumé

Ce document est un rapport de stage de maîtrise qui portait sur la réalisation d'un jeu de 20 questions (20Q) consistant à faire deviner par l'ordinateur un objet d'une maison ou un environnement choisi par le joueur. L'ordinateur pose des questions en langue naturelle et analyse les réponses des joueurs.

Ce jeu de 20Q a pour but de faire la publicité gratuite de l'entreprise d'accueil, North Side Inc. 20Q est lié au projet Bot Colony, un jeu vidéo développé par l'entreprise basé sur la capacité de conversation et de compréhension des personnages du jeu. Cette technologie est utilisée dans 20Q sous forme de digressions et aussi indirectement au sein de l'algorithme, des définitions et des liens avec les besoins humains.

20Q est présenté sur une plateforme web qui permet au joueur un accès libre et gratuit à travers le portail de Bot Colony. Pour susciter l'intérêt des joueurs et attirer plus d'internautes, nous avons ajouté un aspect de compétition avec un système de classement et de points.

Nous présentons également le déroulement du stage au sein de l'entreprise.

Mots-clés : 20 questions, digressions, ILF, OALD, sémantique, interactivité, compétition.

Abstract

This document is a master internship report, it involved the implementation of a game of 20 questions (20Q) the computer has to guess an object in the house chosen by the player. The computer asks questions in natural language and parses the player's responses.

20Q aims to provide free advertising for North Side Incorporation. 20Q is related to the Bot Colony project, a video game developed by the company based on the game characters ability to use NLU for conversation. This technology is used in 20Q as digressions, and also indirectly in the algorithm for the definitions and the links to human needs.

20Q is presented on a web platform that allows the player a free access through the portal Bot Colony. In order to attract the interest of players and more users, we have added an aspect of competition with a ranking system and scores.

We also present the progress of our internship within the company.

Keywords: 20 questions, digressions, ILF, OALD, semantic, interactivity, competition.

Table des matières

I.	Introduction	7
II.	Contexte du projet.....	8
	a. Organisme d'accueil	8
	b. Bot Colony	8
	c. 20 Questions	9
	d. Organisation du projet.....	10
III.	Analyse et spécifications fonctionnelles	12
	a. Les motivations	12
	b. Les caractéristiques et objectifs.....	12
	c. Les technologies et techniques.....	13
	d. Conclusion	16
IV.	Mise en œuvre.....	17
	a. L'interface utilisateur.....	17
	b. L'architecture.....	20
	c. L'algorithme	21
	d. Les digressions	23
	e. La base de données.....	24
	f. Le script.....	24
	g. La présentation web	27
	h. Human Need Script	28
V.	Évaluation et nouvelles perspectives	30
	a. Évaluation	30
	b. Perspectives futures.....	32
	c. Conclusion	33
VI.	Déroulement du projet	34
VII.	Conclusion	37

Liste des figures

<i>Figure 1 : Le début du jeu</i>	17
<i>Figure 2 : Le début du jeu</i>	18
<i>Figure 3 : Les requêtes d'explications de mots</i>	18
<i>Figure 4 : Le refus d'exécuter une commande.</i>	18
<i>Figure 5 : La capacité à donner un sens à l'objet et le lier aux besoins humains.</i>	18
<i>Figure 6 : Des questions sur l'objet.</i>	19
<i>Figure 7 : L'apprentissage du robot.</i>	19
<i>Figure 8 : Représentation des informations sur les joueurs</i>	19
<i>Figure 9 : Architecture du jeu 20Q.</i>	20
<i>Figure 10 : script du début du jeu 20Q.</i>	25
<i>Figure 11 : Les différents fichiers des besoins humains.</i>	28
<i>Figure 12 : Le fichier script d'un besoin humain.</i>	29

Liste des tableaux

<i>Tableau 1 : cahier des charges initial.</i>	11
<i>Tableau 2 : script de 20Q</i>	26
<i>Tableau 3 : comparaison entre jeu des vingt questions.</i>	30

Ce travail, et bien au-delà, je le dois à mes très chers parents qui m'ont fourni au quotidien un soutien et une confiance sans faille et de ce fait, je ne saurais exprimer ma gratitude seulement par des mots.

A ma précieuse sœur Btissam, les mots ne peuvent résumer ma reconnaissance et mon amour pour toi.

A mon encadrant Mr. Guy Lapalme pour son aide et son encouragement qu'il n'a cessé de me communiquer.

A Mr. Eugene Joseph pour ses directives, conseils et encouragement qu'il m'a prodigués.

A tous mes amis avec lesquels j'ai partagé mes moments de joie et de bonheur.

Que toute personne m'ayant aidé de près ou de loin, trouve ici l'expression de ma reconnaissance.

Karim Chghoura

Remerciements

Je remercie M. Eugene Joseph, d'avoir bien voulu accepter de me prendre sous son aile et m'avoir poussé à toujours voir plus loin et à dépasser les limites que je m'imposais.

Je remercie M. Lapalme pour m'avoir communiqué son savoir et ses conseils avisés tout au long de mon projet et la rédaction de mon rapport.

Je remercie le corps Enseignant de l'Université de Montréal pour avoir soutenu mes efforts tout au long de ces deux années, pour m'avoir inculqué le sens des responsabilités et m'avoir permis d'acquérir le savoir nécessaire pour réaliser ce projet.

Je remercie l'Université de Montréal, pour avoir été le cocon protecteur de ma personne, nourrissant mes espoirs, apaisant mes mœurs et me préparant au monde du travail.

Je remercie, ceux, qui prendront le temps de me lire et je vous souhaite tous les bonheurs que la vie peut apporter

I. Introduction

Le monde des jeux vidéos ne cesse d'évoluer et d'innover et de ce fait les entreprises de se multiplier, car ce monde est de plus en plus lucratif et peut rapporter énormément. Ceci dit cette concurrence ne laisse pas place aux doublures, aux copies, mais encourage plutôt l'originalité, surtout pour les entreprises qui se lancent dans ce monde et c'est ainsi que l'entreprise North Side Inc. s'inscrit dans cette dynamique d'innovation, en étant la première entreprise de jeux à proposer un produit qui se base sur la langue parlée et écrite et non sur un script prédéfini.

Dans ce cadre d'innovation, j'ai eu à développer un jeu en apparence simple, mais qui permet de promouvoir la portée de la technologie de cette entreprise et de faire une publicité pour le produit de North Side. Ce stage m'a donc permis d'approfondir mes connaissances sur les technologies utilisées par l'entreprise, puis de concevoir et de développer un jeu de 20 Questions,

Nous présentons d'abord le contexte général du projet, l'organisme d'accueil North Side Inc, puis l'organisation de notre projet. Vient la partie d'analyse et de spécifications fonctionnelles qui débute par une étude des différentes caractéristiques et atouts du projet et des technologies pour réaliser ce projet. Puis nous présentons la mise en oeuvre qui introduit, dans un premier temps, les outils technologiques et architecturaux composant notre jeu. Suit une présentation des différentes étapes de réalisation du projet répondant à un scénario particulier. Finalement, nous évaluons le produit réalisé et proposons de nouvelles perspectives pour son évolution. Le dernier chapitre présente notre parcours au sein de l'entreprise.

II. Contexte du projet

a. Organisme d'accueil

North Side est une entreprise de développement de logiciel, qui développe une technologie permettant aux gens d'interagir en anglais avec des applications, en parlant ou en utilisant le clavier. L'interaction avec les ordinateurs est plus efficace (peut-être même agréable) quand on peut dire avec ses propres mots, à un ordinateur, ce qu'on lui ordonne et que ce dernier s'exécute.

L'entreprise a développé un pipeline de dialogue en anglais qui est utile dans des applications interactives qui interagissent avec les utilisateurs, par exemple: assistants intelligents et Informatique mobile, recherche sémantique, e-commerce, enseignement des langues (English as a Second Language) et jeux vidéo.

Le logiciel est déployé sur des serveurs en ligne et prend en charge l'accès linguistique à des faits spécifiés par l'utilisateur ou des connaissances procédurales (en anglais), les ontologies, les objets de base de données et des scènes 3D.

Le premier produit de North Side Inc est Bot Colony, le premier jeu vidéo utilisant un dialogue intelligent.

b. Bot Colony

Inspiré du Roman *Bot Colony* [1] écrit par Eugene Joseph le président de North Side Inc, est un jeu d'aventure 3D situé sur une île du nom de Agrihan, qui fut durant la deuxième guerre mondiale un bastion japonais.

L'histoire se déroule en l'an 2021, la guerre, la pauvreté et le manque de ressources combinés à la chute des marchés mondiaux, poussent les industriels à utiliser les robots comme main d'œuvre. Nakagawa le leader mondial en nouvelles technologies loue Agrihan et y construit un vaste complexe urbain, industriel et de loisir totalement robotisée, nommé « Bot Colony ».

Le jeu, comme le livre, est composé de plusieurs épisodes : le premier *Intruder* est une affaire d'espionnage, où il faudra commander un robot à distance pour résoudre une énigme et ceci par commande vocale. Cet épisode sert de tutoriel pour les nouveaux joueurs, permettant d'apprendre les rudiments de l'interaction avec le robot nommé Jimmy. Les compétences acquises au cours de la première mission d'intrusion se révéleront cruciales dans les futurs épisodes de la campagne Bot Colony. Dans le deuxième épisode *Arrival*, on arrive sur un aéroport où on peut manipuler plusieurs objets et utiliser une multitude de robots pour répondre à la mission principale de cet épisode. Le dernier épisode

se nomme *Oil rig*, où il faudra affronter des robots agressifs et autonomes qui vous prendront pour cible et dont il faudra se méfier et éviter différentes embuscades.

Le jeu, joué en solo, finit après avoir résolu toutes les énigmes : Il y a aussi des exploits sur Steam (la plateforme de jeux vidéo de la compagnie Valve) à achever ce qui prolonge la durée de vie du jeu. Finir le jeu est le but principal, mais il ne faudra pas négliger la partie apprentissage des robots, l'interaction avec ces derniers et le perfectionnement des commandes à donner au robot.

c. 20 Questions

Twenty Questions (20Q [2]), un jeu de société, basé sur la créativité, la déduction et le raisonnement, a été popularisé aux Etats-Unis à partir du 19^{ème} siècle jusqu'à devenir un programme télévisé.

Pour jouer à ce jeu, il faut au minimum deux joueurs : un répondeur et un questionneur. Le répondeur choisit un objet et le questionneur pose une question dont la réponse est : *oui, non ou peut-être*. Le jeu est gagné par le questionneur s'il arrive à deviner l'objet en moins de vingt questions, sinon la victoire revient au répondeur.

La stratégie la plus efficace pour le questionneur est de poser des questions qui divisent les possibilités approximativement en deux à chaque question. Ce processus est analogue à celui d'un algorithme de recherche binaire en informatique.

Le jeu est souvent utilisé comme exemple pour illustrer la théorie de l'information, en suggérant des descriptions d'objets en au plus 20 bits. Plusieurs développeurs se sont intéressés à ce jeu en développant des algorithmes de plus en plus efficaces pour résoudre le puzzle des vingt questions.

Depuis la révolution d'internet le jeu a connu beaucoup de succès et plusieurs sites proposent de jouer à 20Q tel qu'Akinator (<http://fr.akinator.com/>) ou 20Q.com (<http://20q.com/>).

d. Organisation du projet

i. Présentation du projet

20Q est un jeu qui a une importance majeure pour North Side Inc, notre version consiste à demander au joueur de penser à un objet qui se trouve dans sa maison.

Utilisant un algorithme binaire notre ordinateur représenté par un personnage virtuel Jimmy, essaye de deviner l'objet en posant au maximum 20 questions, basées sur des critères d'espace, de définition d'objet et d'ontologie tout en gardant les questions claires et concises.

Comme ce jeu est relativement simple et peu représentatif de la technologie de l'entreprise, nous avons permis des digressions et des dialogues gérés par le pipeline de dialogue (*Dialogue Manager DM*) de North Side Inc. Nous avons aussi géré la fin du jeu de manière à montrer à l'utilisateur les connaissances sur les objets mentionnés en retournant leurs définitions. Nous avons établi certaines relations avec des besoins humains : le système pose, avec l'accord de l'utilisateur, quelques questions à propos des objets trouvés ou non pour approfondir ses connaissances sur ces objets, ces informations seront reliées aux scripts des besoins humains.

Nous créons une certaine émulation pour intéresser les utilisateurs et construire une communauté, en accumulant les pointages. Par exemple déjouer Jimmy rapporte dix points et répondre aux questions sur l'objet rapporte cinq points par question. Ces points permettent de calculer un classement des meilleurs joueurs et différentes statistiques du jeu.

ii. Cahier des charges

Pour réaliser ce projet, un cahier des charges a été établi en début de projet et qui se présente comme suit :

Tâche à réaliser	Date de début	Date de fin
Amélioration de l'algorithme du jeu.	05/05/2014	12/05/2014
Amélioration des données du jeu.	12/05/2014	19/05/2014
Développement du script du jeu.	19/05/2014	26/05/2014
Développement des questions du jeu.	19/05/2014	26/05/2014
Développement d'une base de données.	26/05/2014	02/06/2014
Développement d'un « End Game ».	02/06/2014	09/06/2014
Intégration du jeu dans une page web.	02/06/2014	16/06/2014
Développement d'une page utilisateur.	16/06/2014	30/06/2014
Modélisation des besoins humains.	30/06/2014	20/07/2014
Intégration des besoins humains.	20/07/2014	31/07/2014
Développement d'un algorithme de brassage d'informations.	31/08/2014	18/08/2014
Amélioration de la base de données.	18/08/2014	29/08/2014

Tableau 1 : cahier des charges initial.

III. Analyse et spécifications fonctionnelles

Cette partie permet d'introduire les différents aspects du projet, son importance et aussi les outils dont il dispose.

a. Les motivations

Les motivations pour la mise en place de ce projet sont publicitaires et financières.

Le but principal est de divertir, intriguer et inciter les gens à acheter le jeu Bot Colony ou le produit de l'ESL (Lecture pour Apprendre). On espère installer une publicité virale pour toucher un maximum d'utilisateurs en un minimum de temps et finalement acquérir des connaissances du monde de la vie quotidienne, à travers le *crowd sourcing*.

Pour ce faire les utilisateurs joueront contre un robot qui se souviendra d'eux, puis ce jeu aura pour but de devenir le canal principal de recrutement pour Bot Colony et les produits « Play to Learn » ; à long terme, il deviendra un assistant intelligent vendu par abonnement.

Un autre but est de permettre au serveur de générer un revenu suffisant pour couvrir ses frais grâce à la publicité si ce jeu devient viral et attire plus d'un millier d'utilisateurs par jour.

20Q est donc un jeu gratuit pour le joueur et peu coûteux pour le développeur : Il donne à l'entreprise un apport publicitaire important avec une source de revenu potentielle.

b. Les caractéristiques et objectifs

Le projet se dote de certaines caractéristiques qui lui permettent de réaliser le cahier de charge et de ce fait atteindre les objectifs.

- Impressionner l'utilisateur avec des capacités de conversation.
- Avoir la capacité de se rappeler du joueur et de toutes les informations qu'il a donné à Jimmy.
- Pouvoir poser des questions pertinentes et claires au joueur, en évitant toute ambiguïté.
- Afficher une connaissance des différents objets de la vie quotidienne.
- Afficher une connaissance des différents besoins humains et poser des questions liées à ces derniers.
- Gérer les questions de clarification.
- Capacité de gérer les digressions lorsque l'utilisateur ne répond pas avec *oui*, *non* ou *peut être*.

- Gérer un dialogue avec l'utilisateur et donc les digressions, mais revenir au jeu après un certain nombre de digressions.
- Poser des questions à l'utilisateur sur son lien avec l'objet deviné.
- Offrir à l'utilisateur de voir Jimmy le robot en action dans le jeu Bot Colony, lorsque l'utilisateur ordonne au robot de faire des actions (pub du jeu).
- Expliquer les mots difficiles dans un anglais correct pour les clients de l'ESL et ainsi faire la pub du jeu Bot Colony en démontrant notre capacité à comprendre les questions et à retourner des explications.
- Favoriser la concurrence entre joueurs, avec une page représentant le classement des joueurs en fonction des points qu'ils ont gagnés au long du jeu.

c. Les technologies et techniques

i. Technologies

Les différentes technologies sont décrites de manière sommaire afin de respecter la confidentialité des informations les concernant, puis leur rôle pour le jeu 20Q est décrit.

1. *Intermediate Logic Form (ILF)*

Le but principal de cette technologie est de coder les connaissances et elle dérive de la *Stanford Logic Form*. Les axiomes peuvent être rédigés en anglais et convertis par le programme en ILF, ainsi d'une phrase en anglais, l'ILF [3] doit être progressivement affinée jusqu'à ce qu'elle soit la plus claire possible, sans ambiguïté et la plus proche de la phrase en anglais.

Le squelette d'ILF est un arbre de dépendance d'une phrase. Les mots sont les nœuds de l'arbre et le lien entre les mots sont étiquetés. Les mots qui apparaissent dans un arbre ou un sous-arbre correspondent généralement à une phrase, et le début de la phrase est la racine de l'arbre.

Cette technologie étant propre à l'entreprise les détails seront omis, mais le principe d'ILF est approximativement le même que le Stanford Logic Form, avec des modifications et surtout de nouveaux tags qui le rendent plus précis et couvrant plusieurs domaines surtout en ce qui concerne les événements reliés aux objets de la vie courante.

Concernant la langue de la logique, elle a sa propre grammaire qui comprend un ensemble de lettres formant l'alphabet et qui permettent ainsi de construire des expressions grammaticalement correctes. Toute suite finie d'éléments de l'alphabet représente une expression, mais toute expression n'est pas grammaticale.

Les expressions grammaticales sont ainsi réparties en différentes catégories:

- les termes, correspondant à la morphologie d'une langue naturelle;
- les énoncés, correspondant à la syntaxe d'une langue naturelle.

La langue de la logique évolue en ajoutant de nouveaux mots et en se débarrassant d'expressions désuètes.

Dans notre projet, cette technologie est utilisée pour la gestion des entrées et la décomposition des phrases des joueurs, en de simples parties gérables par le pipeline utilisé par le jeu 20Q.

2. Oxford Advanced Learner's Dictionary (OALD)

L'*Oxford Advanced Learner's Dictionary* (OALD) dont la publication la plus récente est en 2010 par l'*Oxford University Press* est un dictionnaire monolingue de la langue anglaise, ce dictionnaire hiérarchise les objets sous des définitions communes et utilise des définitions claires et des exemples pour chaque mot.

L'entreprise utilise la version numérisée et a développé des outils pour interroger cette version numérisée et la lier à l'ILF, cette technologie est utilisée lors des digressions et pour toutes les définitions des objets dans le jeu 20Q. L'algorithme du jeu s'appuie sur la sémantique qui est un filtre de poids important pour l'algorithme.

3. Base de données

- VERSANT

C'est une base de données orientée objet [4], persistante, rapide et facile à manipuler une fois créée. Ce type de base de données a une grande problématique en cas de changement de structure, dans certains cas il faut recréer toute la base de données.

Cette base de données est utilisée dans le pipeline de l'entreprise et dans le jeu 20Q pour stocker les informations sur les joueurs et les phrases écrites par les joueurs après transformation en ILF.

- Structured Query Language (SQL)

Un langage informatique servant à gérer des bases de données relationnelles, il permet de gérer et de modifier l'organisation de données, de manipuler les données ce qui nous permet de rechercher de supprimer, d'ajouter ou de modifier des données et enfin de contrôler les données, ce qui nous permet l'accès aux données.

Ce type de base de données est utilisé pour enregistrer les informations concernant le jeu 20Q telles que les statistiques du joueur, les objets et leurs définitions.

4. AJAX

Une architecture informatique qui permet la construction de sites web dynamiques et interactifs. Cette technologie se base sur les différents langages de programmation tels que CSS, JS et XML. Le besoin d'interactivité avec l'utilisateur est primordial pour le fonctionnement du jeu 20Q.

ii. Techniques et langages

Les différentes techniques et langages qui suivront ne sont pas détaillés car elles sont bien connues, mais nous décrivons plutôt leur utilité et leur rôle dans le jeu 20Q.

1. Data Access Object (DAO)

Cette technique regroupe les accès à la base données en représentant chaque table par une classe, cette technique minimise les accès à la base et facilite les manipulations de données.

L'utilisation du DAO dans 20Q se limite aux informations reçues par l'utilisateur concernant les objets et des informations de jeu tels que les points gagnés, le nombre de parties ou le nombre de triches ; DAO a été développé pour gérer une base de données de type SQL. Toutes les étapes de connexions et les requêtes sont de ce type.

2. JAVA, JS, HTML, CSS, XML

Ces différents langages sont la base de la programmation de ce jeu, le langage JAVA est le langage de tout le code cœur du jeu, puis JS, HTML et CSS sont les langages utilisés pour la partie développement web, enfin des fichiers XML pour le langage du script.

3. Besoins humains

La fonction essentielle d'un besoin humain est d'indiquer les satisfactions et le but ultime que l'être humain veut atteindre. Sans satisfaction possible, le besoin est inutile et l'émotion est superflue car elle ne peut jouer son rôle. Il est donc important de comprendre les différents besoins ainsi que les mécanismes de leur satisfaction. Ces besoins peuvent être définis de plusieurs façons par exemple la pyramide de Maslow [5], mais concernant notre jeu, une hiérarchie a été prédéfinie, elle est présentée en annexe.

Il m'a été demandé de modéliser ces besoins humains selon trois branches, les buts ultimes, les besoins de base et se reproduire. Ces trois branches sont subdivisés en plusieurs sous-branches par exemple pour les besoins ultimes nous parlons de besoins physiologiques tels que respirer, manger, dormir et boire, puis le but ultime par exemple la sécurité, la prospérité, le matériel et la socialisation.

Tous ces aspects nous permettent de lier les objets du quotidien aux besoins humains et de remonter dans la hiérarchie des besoins pour atteindre l'une des trois branches principales.

d. Conclusion

Nous pouvons conclure que le jeu a de nombreuses raisons d'être réalisé, il comporte des caractéristiques qui représentent un défi. Les différentes technologies, techniques et langages à maîtriser rendent le projet multitâche et englobe toutes les facettes du développement.

IV. Mise en œuvre

Cette partie démontre la complexité du projet et ses différentes composantes.

a. L'interface utilisateur

Ceci est une illustration des fonctions présentes sur la page web qui répondent au cahier de charge, nous avons pris comme objet une cuillère en bois *wooden spoon*, ceci dit 20Q est maintenant disponible (en version navigateur), sous les liens suivant :

- <http://www.botcolony.com/>
- http://www.botcolony.com/ppSD2/custom/get_started/register-trial.php

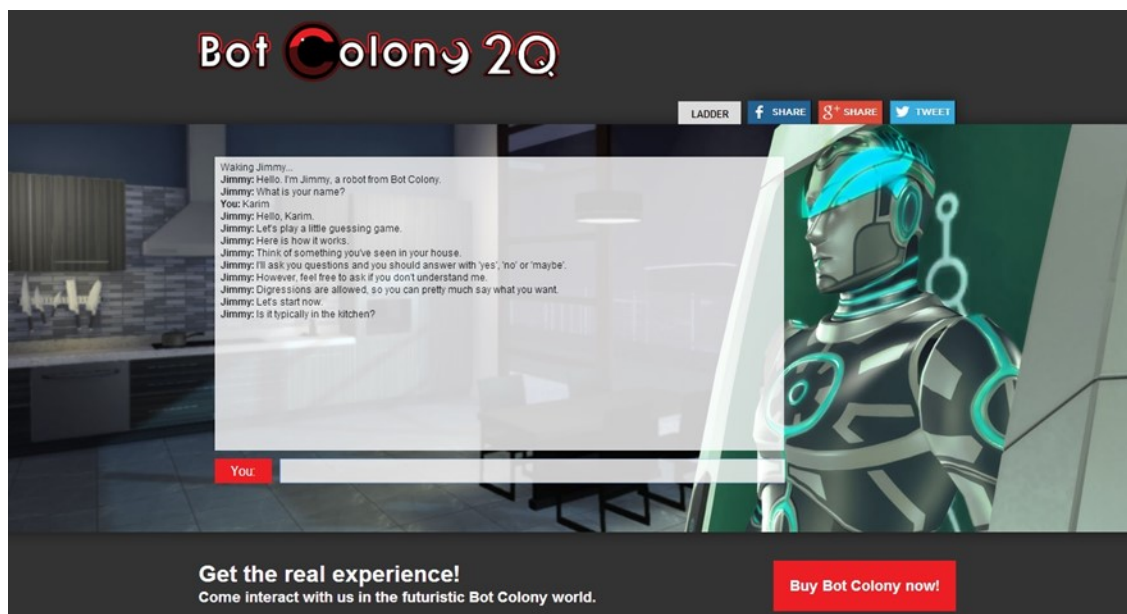


Figure 1: Le début du jeu

Une fois le jeu lancé, nous avons le déroulement suivant

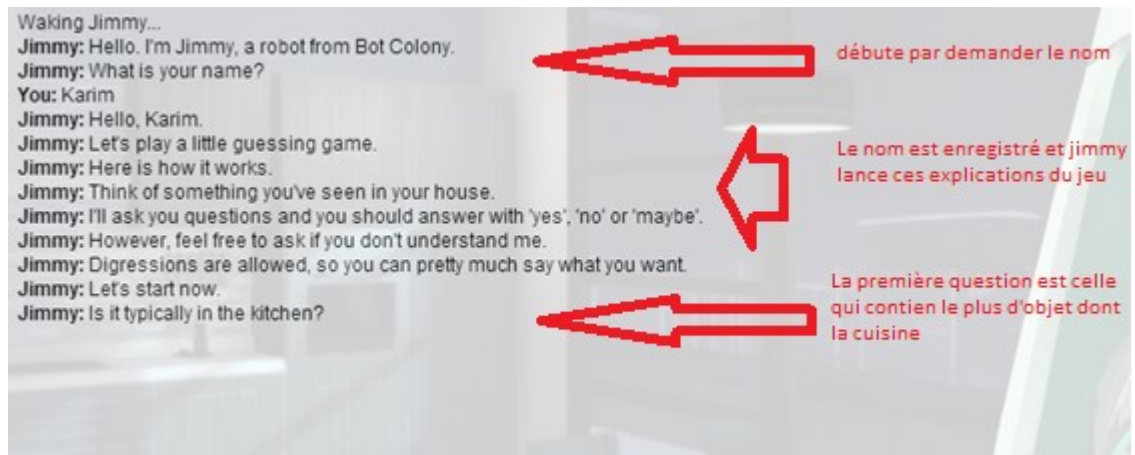


Figure 2 : Le début du jeu

Les digressions et les explications sont prises en charge dans 20Q comme suit

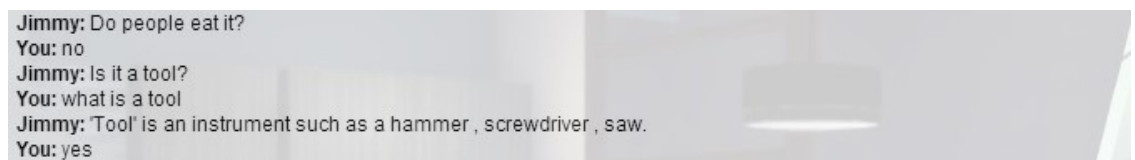


Figure 3 : Les requêtes d'explications de mots

Le joueur peut également poser des questions à propos de Jimmy, son environnement de jeu et tout ce qui lui est relatif dans Bot Colony. Il répondra ainsi à toutes les questions bien formulées, mais après un bout de temps il vous ramènera au jeu et s'attendra à ce que vous répondiez à sa dernière question.

Le joueur ne peut pas donner de commandes à Jimmy comme dans le niveau *Intruder* car c'est un jeu sur navigateur en texte seul, sans 3D. Cependant, 20Q utilise le pipeline de dialogue de Bot Colony.



Figure 4 : Le refus d'exécuter une commande.

Jimmy avertit en fin de jeu si le joueur a triché et qu'il arrive à retracer toutes ses réponses pour donner une preuve. Le robot, que le joueur ait gagné ou perdu démontre une connaissance sur l'objet en question et arrive même à le lier aux besoins humains.

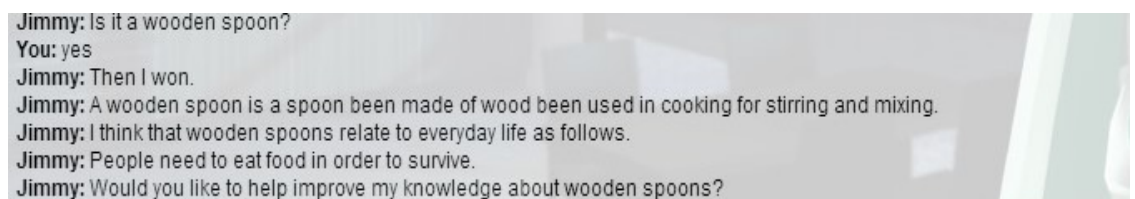


Figure 5 : La capacité à donner un sens à l'objet et le lier aux besoins humains.

Jimmy va essayer d'apprendre des joueurs en posant quelques questions. Il est prévu d'utiliser ce contenu pour améliorer la connaissance de Jimmy sur les gens et la vie de tous les jours.

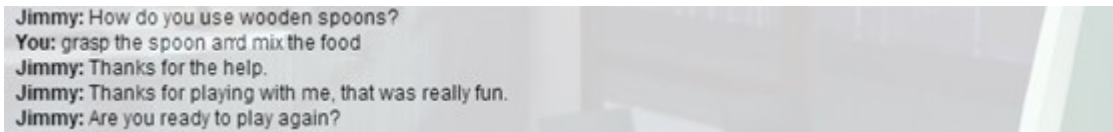


Figure 6 : Des questions sur l'objet.

Jimmy se souvient déjà de tout ce que lui est dit, en lui disant quelques faits, il répondra à la question liée à ce fait, cela fonctionne dans une seule session.



Figure 7 : L'apprentissage du robot.

Une fonctionnalité de la communauté est un classement des joueurs selon les points gagnés durant le jeu. Les nombres de : parties jouées, gagnées, perdues, trichées, le nom et l'ID.

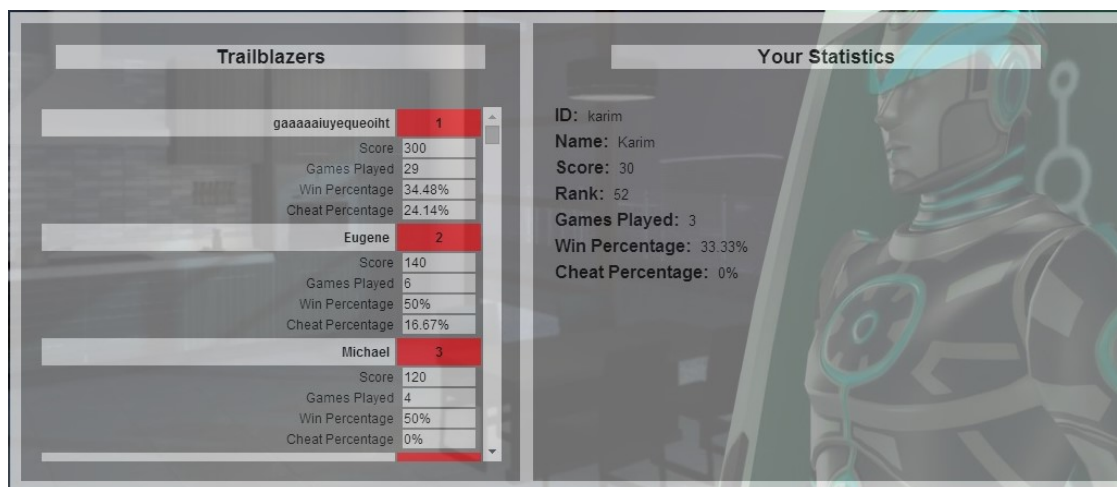


Figure 8 : Représentation des informations sur les joueurs

b. L'architecture

La figure 9 ci-dessous représente l'architecture globale de notre travail au sein de North Side, cette architecture est composée de plusieurs parties interdépendantes et complémentaires:

- Algorithme qui se trouve essentiellement dans les modules 20Q (core, comm et questions).
- Digression qui est, comme son nom l'indique, gérée par le module de digressions et interfacée par le DM (Dialog Manager).
- Base de données (20Q-DB).
- Présentation web avec le module Application server.

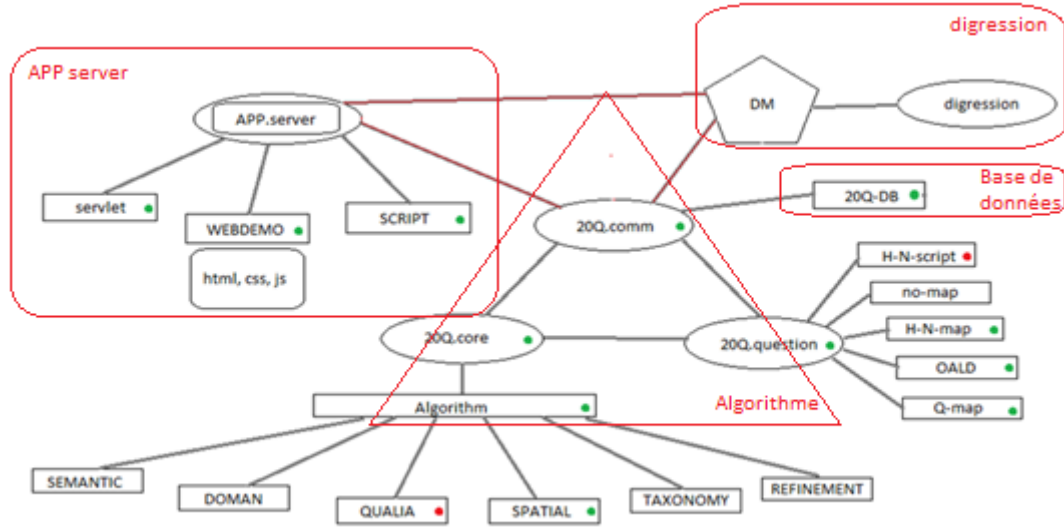


Figure 9 : Architecture du jeu 20Q.

c. L'algorithme

Le jeu se base sur un algorithme de recherche binaire qui départage les différents objets selon différents critères; Il est divisé en trois modules, 20Q.core, 20Q.questions et 20Q.comm.

i. Core

L'algorithme utilise des filtres avec des poids différents:

- Sémantique, qui est la définition OALD de l'objet.
- Domaine, la branche de la hiérarchie auquel appartient l'objet.
- Qualité, l'aspect de l'objet.
- Espaces, la localisation spatiale par défaut de l'objet.
- Taxonomie, classification de l'objet.
- Raffinement, les différentes formes d'un même objet.

Chaque filtre a un coefficient et un poids dépendamment des objets qu'il contient une fois un filtre choisi, ce module génère des questions sous forme de tags, représentant la question à poser sous forme d'un mot en se basant sur l'algorithme. Les tags sont différents selon les filtres : Pour le cas d'une pomme la sémantique sera un tag sous forme de « #isorganic » a une étape puis l'espace sera « \$inthekitchen ». Ce module enregistre le cheminement des questions/réponses ainsi en cas de triche du joueur ou de mauvais cheminement, il le compare aux définitions puis confronte le joueur avec une justification.

ii. Questions

Ce module génère les questions posées au joueur. Il récupère la question sous forme de tags à partir du module (core) puis formule la question et la redirige vers le module (comm). Pour formuler les questions, ce module utilise un fichier qui liste les questions ambiguës (Q-map), ambiguës car leurs définitions sont trop vagues ou globales, puis pour les questions non listées il utilise la technologie ILF pour formuler une question, ayant la définition de l'objet, ce module reformule cette définition sous forme de question.

Il formule les étapes où le joueur a triché, puis relate les parties où ce dernier ne fut pas honnête, cette détection est possible en comparant à chaque étape les définitions disponibles dans notre base de

données et la réponse proposée par le joueur. Par exemple, la pomme est, selon notre définition, organique alors que le joueur répond « non » à la question « est-ce que c'est organique ? »;

Une fois la partie terminée, ce module formule la description de l'objet finale, grâce à l'OALD.

Finalement le module génère les besoins humains adaptés à l'objet choisi par le joueur. Ces besoins sont codifiés dans un fichier Excel où chaque objet satisfait un besoin humain. Le module formule des questions sur l'objet avec l'accord du joueur. Ces questions nous permettent d'avoir plus de détails sur l'objet, par exemple l'endroit où est généralement gardé l'objet, à quoi sert l'objet et comment on l'utilise. Le but est de rassembler un maximum d'informations sur les objets.

iii. Comm

Ce module filtre les messages du joueur et permet des digressions, qui sont l'essence du jeu puisque le joueur peut parler librement avec le robot. Ce module interagit aussi avec le script puisqu'il fait appel au script à chaque fois qu'une différente étape du jeu est atteinte par exemple, le début du jeu la partie des questions puis la fin du jeu.

L'End Game est géré avec les fonctionnalités suivantes :

- Résultat de la partie, le joueur a gagné, perdu ou triché.
- Description de l'objet choisi par le joueur et les besoins humains qui y sont liés.
- Questions sur l'objet choisi.

Ce module enregistre aussi les informations fournies par le joueur dans une base de données.

d. Les digressions

La partie la plus vendeuse du jeu, selon l'entreprise, est la possibilité de faire des digressions, de gérer une communication pseudo humaine avec le robot, cette digression est principalement tirée du pipeline du jeu Bot Colony.

Une fois les digressions permises : le Dialog Manager(DM) qui fait partie du pipeline de l'entreprise, redirige tous les inputs du joueur vers ce module qui utilise différentes technologies :

- Processing.
- Disambiguation.
- Co-reference resolution.
- Reaction module and natural language generation.
 - Script Engine
 - EM (External module)
 - QA (Question Answering)
 - Clarification

Pour pouvoir gérer une conversation avec le joueur et plus spécifiquement se rappeler des faits déclarés par le joueur, la partie digression est dotée d'une base de données qui stocke des informations sur le joueur et qu'on peut interroger ultérieurement.

Toutes ces technologies sont fournies par le pipeline de l'entreprise et elles sont utilisées au sein de 20Q lors de digressions du joueur, mais le jeu reprend après un cycle de trois questions-réponses avec la réplique « let's get back to the game » et puis la dernière question relative au jeu posée de nouveau au joueur.

e. La base de données

La base de données de 20Q est composée de deux tables du nom de Client et Object. Pour y accéder nous avons utilisé la technique des classes DAO, le rôle principal de cette base de données est de pouvoir stocker les informations sur le joueur dont le nom, l'id et le nombre de points accumulés, puis les informations qu'il nous fournit sur les objets qu'il a choisis pour jouer.

En contrepartie des informations fournies sur les objets à chaque fin de partie, le joueur gagnera cinq points par question à laquelle il a répondu; Toutes les informations rassemblées sur les objets auront pour but à grande échelle de donner des définitions communes chez les joueurs, des définitions déduites des statistiques.

f. Le script

Le script principal est sous forme d'une structure XML. Après avoir récupéré le nom du joueur, il introduit le jeu avec une brève présentation du jeu de la procédure à suivre et de la possibilité de poser des questions et de faire des digressions.

A part la fonction de présentation, le script du jeu nous permet de faire des formes de contournements en forçant le DM (*Dialogue Manager*) avec la commande *step redirect* pour alléger le code; Par exemple, pour que le jeu accepte tous les synonymes de *oui*, donc au lieu que le code contient tous les synonymes de ce *oui* le *step redirect* nous permet de rediriger les différentes réponses possibles sous forme d'un *oui* au DM.

Le script aussi nous permet d'expliquer des mots qui n'existent pas dans l'OALD incomplet surtout en ce qui concerne les mots composés. Le script nous permet d'automatiser les définitions en cas de questionnement sur ce type de mots. Le script couvre tous les mots composés produits par Jimmy.

Comme tous les scripts de jeu, il nous permet de suivre un cheminement dans le jeu, des étapes bien prédéfinies; Enfin il nous permet de réagir à quelque phrase spécifique, telles que les commandes ordonnées à Jimmy; Nous pouvons voir ci-dessous une portion du script du jeu qui récupère le nom du joueur (notez que quelques parties sont remplacées par *xxx* pour des raisons de confidentialités):

```

1 <GOAL "xxx">
2 <PLANSTEP>
3 <IF>
4 <AND>
5 <OR>
6 <!-- I am X, My name is X -->
7 <EVENT_DIALOG "xxx"/>
8 <EVENT_DIALOG "xxx"/>
9 <EVENT_DIALOG "xxx"/>
10 </OR>
11 <CONDITION VALUE="Asked the player for name"/>
12 </AND>
13 <SEQUENTIAL>
14 <STEP_SETFACT>~Asked the player for name</ STEP_SETFACT >
15 <STEP_REDIRECT>My name is $FOCUS_X</ STEP_REDIRECT >
16 <STEP_SUBGOAL>Ready to start 20 Questions Minigame</ STEP_SUBGOAL >
17 </SEQUENTIAL>
18 </IF>
19 </PLANSTEP>
20 </GOAL>

```

Figure 10 : script du début du jeu 20Q.

Ligne	Balises	Fonctions
1	<GOAL "xxx">	Délimite chaque fonction et détermine quel agent en est concerner dans notre exemple c'est Jimmy, représenter par xxx.
2	<PLANSTEP>	Contient les étapes qu'il faut suivre de condition en input au séquentiel qui sont les outputs.
3, 4, 5	<IF>, <AND>, <OR>	des conditions d'évènements ou de variables différentes.
13	<SEQUENTIAL>	Contient les étapes à suivre une fois les conditions remplis
14	<STEP_SETFACT>	C'est une phrase qu'on va utiliser comme variable par la suite.

15	<STEP_REDIRECT>	C'est une manière de forcer le DM à accepter des phrases bien définies.
16	<STEP_SUBGOAL>	C'est un commentaire qu'on met juste pour connaître le résultat final de cette partie de script
8	<EVENT_DIALOG "xxx"/>	C'est l'évènement qu'on traque dans tout ce que dit le joueur par exemple « <i>my name is Karim</i> » qui est décomposé de manière à pouvoir récupérer le maximum d'informations sur la phrase telles que son type qui dans notre cas est <i>conventional opening</i> ou <i>ellipsis adverb</i> et aussi une variable contenant « Karim ».

Tableau 2 : script de 20Q

Le script que j'ai développé, contient plus de 1500 lignes couvrant la majorité des scénarios du jeu, la partie la plus complexe fut celle d'assimiler toutes les fonctionnalités des balises et de pouvoir jongler avec un langage de script qui n'est pas très flexible pour faire des boucles, des fonctions et des conditions.

g. La présentation web

La présentation web est la partie qui permet d'interagir avec l'utilisateur : elle comprend l'interface web et une partie serveur, le module APP server, qui gère le démarrage du serveur, la création des *servlets* puis le démarrage du DM. Une fois le serveur démarré, il permet l'interaction avec *webdemo*, et lance les pages web du jeu et de l'interface utilisateur.

Puis l'application serveur, une fois les parseurs téléchargés, initialise les scripts pour le DM et ainsi commence le jeu.

Une fois les pages web accessibles il lance le script du jeu décrit. Ce module basé sur du code JS, HTML et AJAX est le plus important pour le jeu 20Q puisqu'il interface avec l'utilisateur et avec toute la partie cœur du jeu et code du jeu.

J'ai développé des *servlets* et modifié la *servlet* principale ainsi que toute la partie développement des pages web en JavaScript, allant même jusqu'à la mise en page de l'interface utilisateur avec CSS.

Tout au long du projet il y a eu des modifications concernant l'aspect graphique et j'étais responsable de l'intégration des nouvelles interfaces.

La partie la plus difficile fut de comprendre le module APP Server, sachant qu'il comprend plus de 2000 lignes et fait appel à une dizaine de fonctions, dont certaines se basent sur AJAX et qu'il fallait maîtriser.

h. Human Need Script

Human Need Script est un script modélisant les besoins humains selon une hiérarchie s'apparentant à la pyramide de Maslow [5] mais qui est plus complexe et recouvre les besoins de base d'un être humain et aussi ses objectifs de vie.

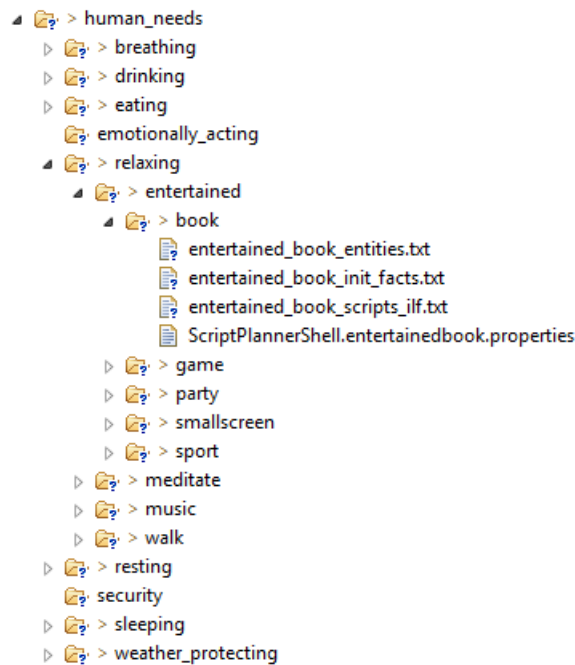


Figure 11 : Les différents fichiers des besoins humains.

Ce scripte se base sur quatre fichiers, prenons l'exemple du besoin de divertissement (*entertained*) nous avons donc les fichiers suivants

- *ScriptPlannerShell.entertainedbook.properties* ou le fichier de propriétés, contient toutes les propriétés du besoin humain en développement et ce qui est le chemin pour atteindre les trois autres fichiers car ce dernier va être lu par le Script Shell qui sera détaillé plus loin.
- *Entertained_book_entities* ou le fichier d'entités : correspondent aux objets du quotidien qui sont liés aux besoins humains.
- *Entertained_book_init_facts* ou le fichier d'initialisation des faits : définit l'existence d'objets prérequis pour que le besoin humain soit satisfait.
- *Entertained_book_scripts_ilm*, le fichier de script contient les enchainements à suivre et la hiérarchie des besoins humains, composé de préconditions qui une fois satisfaites produisent un effet de bord, puis les actions sont appelées les « steps ».

```

1 TYPEDECLARATIONS:
2
3 # a human as an individual
4 person person.n.01
5
6 # a set of paper that we can read
7 book book.n.01, magazine.n.01, newspaper.n.01
8
9
10 PRECONDITIONS:
11 ENGLISH:=The book|exist.
12 STEPS:
13 ENGLISH:=The person open the book
14 ENGLISH:=The person read the book
15 SIDEEFFECTS:
16 ENGLISH:=The person read the book
17
18
19 PRECONDITIONS:
20 ENGLISH:=The person read the book|
21 SIDEEFFECTS:
22 ENGLISH:=The person get entertained.
23
24
25 PRECONDITIONS:
26 ENGLISH:=The person get entertained.
27 SIDEEFFECTS:
28 ENGLISH:=The person relaxes.
29
30
31 PRECONDITIONS:
32 ENGLISH:=The person relaxes.
33 SIDEEFFECTS:
34 ENGLISH:=The person fulfilled one of the need

```

Figure 12 : Le fichier script d'un besoin humain.

La figure 12 représente le développement du besoin humain de divertissement au travers la lecture (lignes 19-22), ainsi avec la présence des objets tels qu'un magazine, un journal ou un bouquin (ligne 11), on peut se divertir en lisant un livre après l'avoir saisi puis ouvert (lignes 13-14). En se divertissant on peut relaxer (lignes 25-28) et donc par la suite on répond à l'un de nos besoins (lignes 31-34).

V. Évaluation et nouvelles perspectives

Cette partie est une évaluation du travail répondant aux différentes requêtes du cahier de charge, puis une liste des perspectives futures pour le jeu.

a. Évaluation

Nous avons comparé notre jeu avec les différentes représentations de ce jeu sur le web, tel que Akinator (<http://fr.akinator.com/>) 20Q.com (<http://20q.com/>). Pour les comparer à notre jeu nous avons utilisé plusieurs critères : l'efficacité de l'algorithme, la polyvalence, l'interactivité, la popularité et la compétitivité, représentée au tableau suivant.

	Efficacité	Polyvalence	Interactivité	Popularité	Compétition
20Q	80%	500 objets	Langue parlé	--	Pointages et classement
Akinator	96%	Milliers d'objets et de personnalités	Boutons	++	Absente
20Q.com	92%	Milliers d'objets	Choix limitées	+	Absente

Tableau 3 : comparaison entre jeu des vingt questions.

L'efficacité de l'algorithme, est définie par le nombre de fois où l'algorithme arrive à trouver l'objet sur un échantillon de 50 objets. Ces objets ont été choisis au hasard à partir de la base de données de notre jeu (la liste des objets est donnée en annexe). L'efficacité de notre système n'est que de 40/50, pour le 20Q.net il est de 46/50 et Akinator de 48/50 (concernant ces deux derniers le critère de présence de l'objet dans la base de données ne peut être vérifié).

Pour la polyvalence, notre jeu est limité à quelques centaines d'objets du quotidien qu'on retrouve dans une maison alors que les deux autres jeux couvrent pour leur version française une plus large palette d'objets avec un avantage pour Akinator qui comprend les personnalités connues. Ces deux jeux traitent plusieurs langues alors que 20Q ne gère que la langue anglaise.

L'interactivité est un point fort de 20Q car ce dernier ne propose pas les réponses aux utilisateurs sous forme de bouton ou de texte à cliquer. L'utilisateur est libre de répondre sous la forme qu'il souhaite et même de s'écarter du jeu pendant un court moment.

Au point de vue popularité, Akinator est supérieur à de 20Q.com. Les deux accueillent des milliers de visiteurs par jour, alors que notre jeu ne dépasse pas les quelques dizaines de joueurs par jour. La compétition est un avantage pour notre jeu, car les autres jeux ne le font pas.

Akinator domine en ce qui concerne un jeu pur et simple divertissant et complet. Notre jeu, même s'il souffre de plusieurs faiblesses, possède un atout important : se base sur le dialogue et essaie d'être le plus clair possible dans ses questions, répondant ainsi aux attentes de M Eugene Joseph qui avait donné une grande importance à ce projet en écrivant :

“The Twenty Questions game will get additional resources, because it is strategic

- to get us visibility
- to source players without paying a fortune for ads on Google AdWords
- to crowdsource everyday life knowledge

Twenty Questions is now considered a full-fledged North Side product, and has its own business plan.” Eugene Joseph, *business plan*

b. Perspectives futures

Parmi les perspectives futures, il y a l'utilisation de la modélisation des besoins humains qui est sous forme de script, puis plus important encore c'est de transformer le robot en un assistant personnel et enfin concernant le jeu, de rendre son algorithme plus performant.

i. Assistant personnel

Le but ultime du jeu est, à part divertir, d'être un véritable assistant personnel. Pour ce faire, il doit être capable de se rappeler de toutes les informations qui lui ont été apprises par le joueur et les rappeler au moment opportun.

A premier abord cela semble difficile, mais le jeu y arrive car il est doté déjà d'une mémoire certes éphémère mais qui retient toute information inculquée au robot durant une session, ainsi il faudra développer un module qui pourra filtrer les informations de chaque session et enregistrer les plus pertinentes sous forme d'ILF. Ce module pourra restaurer toutes ces informations, une fois questionné sur ceux-ci ou lors de rendez-vous.

Cette version d'assistant personnel, inédite et gratuite, pourrait intéresser une communauté importante d'internautes.

ii. Performance de l'algorithme

La performance de l'algorithme peut être nettement améliorée si les tâches suivantes étaient réalisées :

La sémantique des objets doit être correctement adaptée à un algorithme binaire, c'est-à-dire qu'un objet doit être atteignable sur plus de trois ou quatre étapes dans la hiérarchie et doit être liée à la racine, ce qui n'est malheureusement pas le cas actuellement. Les objets utilisés ont été testés et corrigés suivant ces critères, mais l'ensemble des objets traités par l'OALD ne le sont pas.

La liste des objets doit être augmentée car pour le moment le jeu se contente d'une liste de 500 objets localisés dans la maison. Ce qui est problématique car le joueur aperçoit une faible performance du robot à deviner un objet, alors qu'il retourne sa définition et même le lien entre cet objet et les besoins humains.

Les questions posées au joueur doivent être plus précises et moins ambiguës, actuellement plusieurs domaines ont été écartés pour éviter ce type d'ambiguïté. Le filtre qualité doit être rajouté à l'algorithme pour pouvoir gérer le raffinement sur les objets.

Quand il ne reste plus qu'un certain nombre d'objets qui font partie de la même famille, la distinction entre ces objets devient difficile. Ainsi les qualités des objets peuvent différencier entre eux et continuer à faire un parcours binaire, nous citons des qualités telles que la composition de l'objet, la couleur, étant électrique ou pas et plein d'autres.

c. Conclusion

Les principaux critères du projet ont été satisfaits sauf pour la modélisation ou plus précisément l'intégration du script aux besoins humains. Le projet aurait pu être amélioré de manière significative si toutes les perspectives futures avaient été réalisées.

VI. Déroulement du projet

Nous décrivons maintenant le déroulement du projet qui a connu des hauts et des bas et fut plein de rebondissements, le stage ne s'étant pas terminé comme prévu.

Le 6 avril 2014, j'avais contacté M Guy Lapalme pour un sujet de mémoire, mais après discussion et mon affinité pour un stage étant habitué à ces derniers, la décision fut le stage comme projet de fin d'études. Ainsi commença ma recherche de stage, le temps passa, j'ai postulé à plusieurs endroits, mais j'ai plutôt reçu des offres d'emploi que de stage.

M Lapalme me signala le 26 avril, une offre de stage au sein d'une entreprise North Side Inc. qui œuvrait dans les jeux vidéo avec une emphase sur le traitement du langage. Le directeur M Eugene Joseph avait accepté un stagiaire provenant de l'Université de Montréal auparavant. L'offre était intéressante et portait sur plusieurs projets, je n'ai pas hésité à répondre positivement à cette offre et à montrer mon enthousiasme.

Après un entretien téléphonique le 28 avril et un test le lendemain au sein de l'entreprise, j'ai été accepté pour un stage de 4 mois qui dès la première impression fut très plaisante. Le directeur engageant et passionné ne fit qu'augmenter ma motivation et mon intérêt pour ce stage.

Vint la signature de l'offre de stage le 1^{er} mai et le problème du permis de travail car étant étudiant étranger je ne peux exercer en dehors de l'université. Les procédures par voie postale et même en ligne prennent du temps, allant de deux à quatre semaines ouvrables pour l'obtention du permis de travail.

Le seul moyen, qui reste aventurier et périlleux est le fameux « tour du poteau », qui consiste à sortir du territoire canadien puis rentrer pour passer par la douane canadienne et ainsi obtenir le permis au centre de douane frontalier.

Je passerai les détails de cette aventure qui pour un marocain, ne fut pas moins stressante; je ne cacherai pas ma surprise quant à la gentillesse des douaniers.

Le papier en mains le samedi 3 mai, je débutais le stage le lundi 5 mai.

Première journée de stage le 5 mai, les logiciels à installer les fichiers à importer, on s'installe. Vint l'après-midi où j'ai été convoqué pour une réunion regroupant l'équipe des développeurs, une réunion dont les termes me dépassaient puisque je n'avais pas connaissance des différentes technologies utilisées et des différents termes dont la réunion parlait. Ceci dit M Joseph nous a demandé de faire quelque tâches telles que le débogage de l'algorithme et de commencer à travailler sur le jeu 20Q en ayant les spécifications et le cahier de charge.

Le lendemain, j'ai été présenté aux différentes équipes responsables du développement du jeu Bot Colony, ce qui fut gratifiant et fort utile.

La première semaine fut consacrée au parcours de l’algorithme et à quelques modifications avec en parallèle la mise en place d’une interface web pour l’affichage du jeu.

La semaine passa vite et je me suis retrouvé le 12 mai à relater tout ce qui avait été accompli durant la semaine; je me suis fixé d’autres tâches pour les semaines à venir dont l’étude de la technique DAO et j’ai commencé à faire des tests sur l’algorithme, son efficacité et les mots qui n’étaient pas atteignables dont la liste fut précisée par un autre test qui a déterminé les différents objets qui n’avaient pas de racine. Cette liste fut fournie au groupe du traitement de langage qui a manuellement mis en place des liens entre ces objets et les racines.

La deuxième semaine, j’ai commencé à entendre parler d’une échéance qui était autour de la mi-juin, ce qui était serré, vu le nombre de tâches à accomplir pour que le jeu soit fonctionnel.

La troisième semaine consista à debugger le jeu, à enlever les ambiguïtés concernant les questions posées. Plus le temps passait, plus je maîtrisais le pipeline et les différentes classes du projet. Pendant cette semaine, j’ai procédé à la création d’une base de données contenant toutes les infos du joueur.

La quatrième semaine, les tests pour le lancement de la bêta du mini jeu et rapport de bugs continue. Puis le travail sur de nouveaux filtres pour l’algorithme pour ajouter de la précision notamment le filtre des espaces et le filtre des qualités.

La cinquième semaine fut très chargée avec une requête du directeur pour faire des heures sup. J’avais atteint les 70 heures cette semaine, mais le résultat fut concluant car au cours de cette dernière, il y a eu beaucoup d’avancement concernant les tests du jeu, puis l’intégration du filtre d’espaces, et la mise en place d’un script pour la

- gestion les définitions des noms composés reliés aux questions du jeu.
- mise en place d’une stratégie de vente du produit.
- mise en place de trois questions reliées aux besoins humains.

Sans oublier l’intégration au jeu des besoins humains décrits dans un fichier Excel.

La dernière semaine avant l’échéance était centrée sur la migration de ma base de données sur le serveur principal puis l’intégration d’une nouvelle page web développée par le groupe d’artistes le 9 juin.

Tout était prêt et disponible pour une version beta pour les joueurs pour l’échéance mais le 18 juin et suite au lancement du jeu principal Bot Colony, le 16 juin, M Joseph nous a réuni pour nous annoncer que l’entreprise ne disposait plus de fonds pour pouvoir continuer à développer le jeu Bot Colony et donc 20Q. Ce fut sans surprise pour certains mais choquant pour d’autres. Personnellement je fus frustré par cette mauvaise nouvelle d’autant plus que le jeu 20Q commençait à susciter de plus en plus d’intérêt chez les internautes passant de quelques joueurs à quelques dizaines de joueurs par jour.

Ceci dit, il nous restait quelques jours pour bien structurer le code et arranger les quelques défauts qui persistaient et surtout grâce aux joueurs découvrir quelques défauts et les corriger.

Le 27 juin, tout le monde devait se dire au revoir sauf pour une poignée de personnes qui avaient pour mission de maintenir le jeu en vie, et je faisais partie de ce groupe de survivants. J'avais pour mission de développer le script des besoins humains et de corriger les quelques bugs qui apparaissent concernant le jeu 20Q. Il fallait expliquer toutes les bases de ce jeu aux autres employés qui allaient rester après mon départ. J'ai fini par modéliser les besoins humains et les objectifs humains le plus près possible de la hiérarchie élaborée par M Joseph.

Ce script fut testé avec le *script shell* qui était malheureusement limité à une simple question « *how can* », qui permettait tout de même de tester le fonctionnement.

Pour finir j'ai discuté avec un autre employé les différentes possibilités d'intégration de ce script et de son utilité pour une éventuelle prochaine version.

J'ai dû quitter l'entreprise le vendredi 1 août 2014 avec regret mais forte d'une expérience sans précédent.

Puis commença la rédaction de ce rapport le mardi 5 août après un entretien avec M Lapalme, durant lequel nous avons déterminé les grandes lignes et la démarche à suivre. Ont suivi des entretiens toutes les deux semaines, entre lesquels je rédigeais le rapport et le corrigeais.

VII. Conclusion

Ce projet de fin d'études, réalisé au sein de North Side Inc, m'a permis de mettre en œuvre mes connaissances théoriques et techniques acquises durant ma formation à l'Université de Montréal, et plus spécifiquement les études en web sémantique. Ce projet m'a permis également de connaître les étapes nécessaires pour la création d'un jeu vidéo.

J'ai pu appréhender les différents métiers de programmeur tel que la réalisation d'algorithme, le développement web, le développement de scripts et la maîtrise de produit nouveaux. J'ai pu également maîtriser les outils software et hardware de développement ce qui m'a permis de réaliser le jeu 20Q.

De plus, sur le plan relationnel, j'ai compris que l'esprit d'initiative, le travail en équipe, le sens de responsabilité et l'aptitude à communiquer sont des éléments capitaux pour la réussite d'un projet. Enfin, sur le plan personnel, j'ai développé mon aptitude d'autonomie, de patience et de persévérance.

Ainsi les objectifs fixés au début de mon stage ont été atteints c'est-à-dire la mise en place d'un jeu qui est 20Q sur une plateforme web.

Ceci dit, ces trois mois ont constitué une expérience extrêmement riche tant sur le plan technique certes, que sur les plans professionnel et relationnel. En effet, j'ai pu parfaire mes connaissances en matière de réseaux et surtout de m'intégrer dans un milieu professionnel où l'erreur était inacceptable.

Certes l'entreprise n'a pas pu déployer ce projet pour des raisons financières, mais nul ne doute du rôle qu'aurait apporté 20Q à la compagnie.

Pour finir, cette expérience professionnelle au sein de North Side Inc m'a fait apprendre les atouts nécessaires pour une intégration d'informaticien débutant sur le marché d'emploi.

Bibliographie

- [1] Eugen Joseph, *Bot Colony: A Novel Set In The Present And Near Future*, Canada: North Side Inc, Transcontinental. juillet 2010.
- [2] David Weinstein, *The Forgotten Network*: DuMont Philadelphia: Temple University Press, 2004.
- [3] Mark Rübans, *Logic as Language and Logic as Calculus*, Syntheses, volume 17: pages 324-30.
- [4] *TechView Product Report: Versant Object Database*, Resource Portal for Big Data and New Data Management Technologies odbms. Retrieved 6 October 2010.
- [5] Abraham Maslow, *A Theory of Human Motivation*, Psychological Review, no 50, 1943.

La majorité de l'ouvrage se base sur le wiki de l'entreprise qu'elle a mise en place pour la compréhension des différentes technologies.

Sites webs utilisés :

- <http://wordnet.princeton.edu/>
 - Wordnet m'a aidé à trouver les liens les objets et les différents besoins humains ainsi qu'à développer mon script et de retourner les définitions non OALD.
- <http://www.oxfordlearnersdictionaries.com/>
 - La base de données de notre OALD.
- <http://www.w3schools.com/ajax/default.ASP>
 - Le w3schools est une importante plateforme de soutien technique pour le développement.
- <http://api.jquery.com/jquery.ajax/>
- <http://www.postgresql.org/>
- <http://dev.mysql.com/doc/>
 - Les trois types principaux de base de données et leurs didacticiels.

Annexe

a) Hiérarchie des besoins humains

- But 1: Besoins basiques.
 - But 1.1: respirer.
 - But 1.2: manger*
 - But 1.3: boire.
 - But 1.4: dormir.
 - But 1.5: se reposer
 - But 1.6: relaxer.
 - But 1.6.1 : faire du sport.
 - But 1.6.2 : faire du yoga.
 - But 1.6.3 : faire une activité créative.
 - But 1.6.3.1 : dessiner.
 - But 1.6.3.2 : jouer de la musique.
 - But 1.6.3.3 : peindre.
 - But 1.6.3.4 : écrire.
 - But 1.6.3.4 : jouer du théâtre.
 - But 1.6.3.5 : faire du cinéma.
 - But 1.6.4 : lire.
 - But 1.6.5 : regarder une émission.
 - But 1.6.6 : regarder une pièce théâtrale.
 - But 1.7: protéger quelqu'un.
 - But 1.8: se protéger.
 - But 1.9: se vêtir.
- But 2: Objectif de vie
 - But 2.1: créativité.
 - But 2.2: argent.
 - But 2.3: éducation.
 - But 2.4: croyance.
 - But 2.5: famille
 - But 2.6: amour.
 - But 2.7: joie de vivre.
 - But 2.8: santé.
 - But 2.9: contribution sociétale.
 - But 2.10: connexions sociales.
- But 3: Procréer.

*A partir du niveau 3 si le besoin ne peut plus être détaillé on développe son script

GOAL 1.2.1

- Person has. ready-to-eat food

PRE-CONDITIONS

- Person has money $>$ price of food
- Person is in retail store
- Retail store has food

PLAN

- Person buys food at the retail store;
- Person or person's helper prepares food.

SIDE-EFFECT

- Person has ready-to-eat food.

ALTERNATE PLAN for Person get ready-to-eat food.

GOAL

- Person get ready-to-eat food.

PRE-CONDITIONS

- Person has money $>$ price of food.
- Person is in restaurant.
- Restaurant has. food.

PLAN

- Cook in restaurant prepare food.
- Waiter in restaurant serve food to person.

SIDE-EFFECT

- Person has ready-to-eat food.

b) Liste d'objet du benchmark

Cette partie représente la liste d'objets qui a été extraite du document qui représente l'ensemble d'objets jouables par notre jeu 20Q du nom de *household*.

automobile.n.O0	dehumidifier.n.O0	keyboard.n.O1	perfume.n.O1	suitcase.n.O0
baggie.n.O0	deckchair.n.O0	knife.n.O0	piano.n.O0	television.n.O1
baking_powder.n.O0	decanter.n.O0	laptop.n.O0	plate.n.O1	thermometer.n.O0
baking_soda.n.O0	desk.n.O1	match.n.O1	radio.n.O2	truck.n.O3
butter.n.O0	dressing_table.n.O0	mirror.n.O1	razor.n.O0	undershirt.n.O0
cable.n.O2	dustpan.n.O0	mustard.n.O1	rug.n.O1	video_game.n.O0
camera.n.O0	egg.n.O2	necklace.n.O0	salt.n.O1	violin.n.O0
cushion.n.O1	flour.n.O0	newspaper.n.O1	shoe.n.O1	wine.n.O1
cutting_board.n.O0	glove.n.O0	oil.n.O2	skirt.n.O1	wooden_spoon.n.O0
deck.n.O3	hat.n.O1	onion.n.O0	sponge.n.O1	xylophone.n.O0