

Création d'un agent conversationnel virtuel pour la conférence WWW'2012 avec Unity3D

Benjamin BARATE, Florian TRAVERSAZ

Janvier 2012

Résumé: Ce projet a pour but de créer un agent conversationnel pour un monde virtuel créé par l'entreprise EXT-Reality à l'aide du moteur de jeux Unity3D. Ce monde virtuel sera utilisé dans le cadre de la session posters de la conférence WWW'2012. Le rôle de cet agent est d'accueillir, orienter et informer les utilisateurs par le biais de leurs avatars.

Mots-clés: agent intelligent, agent conversationnel, monde virtuel, WWW'2012, Unity3D

1. Introduction

Ce projet s'inscrit dans le cadre de la conférence WWW'2012 qui se déroulera à Lyon du 16 au 20 Avril 2012 à la Cité Internationale. L'entreprise GAEANOVA souhaite recréer une session posters virtuelle reprenant les posters qui seront utilisés lors la conférence. Cette société est une Start up fondée en 2011 et qui est spécialisée dans l'édition de plateformes web 3D sociales, communautaires et géo-contextualisées dédiées aux univers virtuels professionnels. Les principales personnes impliquées dans ce projet sont Christian MORAND (fondateur de GAEANOVA), Jérôme POIRAUD (responsable marketing et Technologie) et Henry MORLAYE (co-fondateur de la société A WORLD FOR US). La société A WORLD FOR US est partenaire de GAEANOVA et lui fournit une plate-forme permettant la création de monde virtuel.

Notre encadrant, Lionel MEDINI est, quant à lui, responsable de la session posters pour la conférence WWW'2012.

2. Etat de l'art :

a. Les agents intelligents :

Notre Etat de l'art s'est orienté autour des agents intelligents, leurs comportements et ceux des individus interagissant avec eux. Nous nous sommes focalisés sur un type d'agent intelligent, l'agent conversationnel.

L'agent conversationnel animé est un personnage interactif situé dans un environnement virtuel et qui a pour but la communication avec un utilisateur par différents biais (écrit, oral, gestuel, posture). Il convenait donc avant tout de cerner concrètement la manière de définir un agent, pour le différencier d'un programme.

Dans son article [FG96], Stan FRANKLIN classe les agents selon différentes propriétés (cf. figure 1) : la réactivité, l'autonomie, l'orientation vers un but, la persistance dans le temps, la communication, l'apprentissage, la mobilité, la flexibilité et la personnalité.

Propriété	Autre Nom	Signification
Réactif	Détection et action	Répond dans un court laps de temps à un changement dans l'environnement
Autonome		Exerce un contrôle sur ses propres actions
Orienté vers un but	Proactif décidé	N'agit pas simplement en réponse à l'environnement
Persistant		Processus fonctionnant en permanence
Communicatif	Sociable	Communique avec d'autres agents et éventuellement avec des personnes
Evolutif	Adaptable	Change son comportement en fonction de ses expériences précédentes
Mobile		Capable de se transporter d'une machine à une autre.
Flexible		Les actions ne sont pas scriptées
Doté d'une personnalité		Personnalité réaliste et états émotionnels.

Figure 1 : Tableau de propriétés d'un agent d'après Stan Franklin

Selon lui, un agent intelligent doit répondre aux quatre premiers critères au minimum. Les autres propriétés permettent de développer des agents plus évolués. Les différentes études faites dans ce domaine ont avant tout pour but d'humaniser le plus possible l'agent conversationnel. D'une part, en le rendant le plus « intelligent » possible et, d'autre part, en lui donnant un comportement le plus réaliste possible vis-à-vis des autres individus de l'univers virtuel. Nous allons donc développer par la suite ces deux principales caractéristiques.

i. L'intelligence :

Les premières vraies avancées en matière d'intelligence, de réaction et d'interaction verbale avec un individu sont apparues dans les années 1950 avec la proposition du Test de Turing. Ce test consiste à faire engager deux conversations simultanées à un individu. L'une avec un agent et l'autre avec un autre individu. Si le candidat n'est pas capable de dire lequel de ses interlocuteurs est un programme informatique, on peut considérer que l'agent conversationnel a passé le test avec succès. Il en a découlé la création de deux types d'agents conversationnels : ceux capable d'apprentissage et ceux qui ne le sont pas.

A la base, un agent conversationnel n'est capable de répondre qu'aux questions dont il connaît la réponse. Il est donc assez limité dans le maintien d'une conversation avec un être humain. Les agents capables d'apprentissage, quant à eux, tentent d'isoler des mots clés dans les données qu'ils reçoivent de leurs interlocuteurs et d'enrichir leurs panels de connaissances grâce aux utilisateurs. Cependant, cette approche est compliquée à mettre en œuvre et les résultats ne sont pas convaincants et très dépendants de la bonne volonté de l'utilisateur qui converse avec l'agent. Il faut cependant relativiser ce test et les réactions des agents selon le contexte dans lequel ils sont utilisés. La plupart de ces agents conversationnels sont utilisés dans le cadre d'assistance en ligne pour répondre à certaines questions bien spécifiques des utilisateurs. L'agent n'a donc pas besoin d'une grande diversité de vocabulaire et de « culture ».

ii. Le comportement :

La grande majorité des études faites sur les agents conversationnels s'articulent autour du comportement de l'agent vis-à-vis de l'utilisateur, de ses réactions et de ses gestes dans le cadre d'une discussion. Le but étant avant tout de faire en sorte que l'utilisateur soit le plus à l'aise possible durant toute la discussion. Comme l'explique GOLANSKI et CAELEN dans [GC05], le comportement d'un agent conversationnel s'axe principalement autour des animations corporelles, des expressions du visage (si le moteur graphique le permet) et des gestes sémiotique (les gestes de désignations d'objet ou de lieu et

d'indication). Grâce à ces études, il a été déduit plusieurs « règles d'usage » pour la création d'un agent conversationnel. Dans un premier temps, il faut que l'agent soit doté d'un large panel d'animations pour un seul et même contexte (discussion, déplacement, attente...). Par exemple, si l'agent parle à l'utilisateur, il a une gestuelle propre à la discussion (mouvement des bras, de la bouche...) mais cela devient gênant pour l'interlocuteur lorsqu'il s'aperçoit d'une redondance dans le déroulement des animations. Il a tendance à se focaliser dessus et à ne plus être réceptif au message que l'agent tente de faire passer. De plus cela déshumanise l'agent et crée de la distance entre lui et l'utilisateur. C'est sur les gestuelles qui dure dans le temps qu'il est le plus difficile de rendre réaliste l'agent conversationnel (position d'attente, de discussion).

Au niveau de l'échange verbal, il a aussi été montré qu'il n'était pas pertinent d'utiliser un agent conversationnel dans un unique but de question-réponse. Il est important que, malgré les limites rencontrées dans l'échange entre l'utilisateur et l'agent, celui-ci rende la discussion la plus réelle et fluide possible. Pour cela, l'agent doit pouvoir faire des transitions entre deux phrases, marquer des pauses dans la discussion pour simuler une réflexion et surtout essayer de garder une cohérence dans le discours et entre le discours et la gestuelle.

D'après BUISSINE dans [Bui05], il lui semble aussi important que l'utilisateur ait les mêmes moyens de communication. On parle alors de multi-modalité en entrée et sortie avec l'agent. Cela signifie que si l'agent peut répondre par oral ou par écrit à l'utilisateur, il est important que l'utilisateur ait lui aussi ces deux moyens de communications afin d'être sur un pied d'égalité sur l'aspect communicatif avec l'agent.

Notre problématique ajoutant à tout cela la notion de monde virtuel, il faut donc aussi prendre en compte les notions de territoire, d'interactions dans l'espace et d'utilisation de l'environnement. Il y a dans ce domaine ce que l'on appelle la théorie des faces. Dans [CHI98], CHICOISNE parle d'acte socialement correct selon huit critères, quatre pour l'ego et la face de l'agent et quatre pour la notion de territoire. L'aspect territorial est cependant très dépendant de la nationalité et de la culture de l'utilisateur. En effet, selon la nationalité, la distance physique avec un autre individu peut être ressentie comme de l'agression ou de la froideur (notions de distance intime, personnelle et sociale).

[BP98] met en exergue l'importance des tours de paroles, cela structure la discussion et évite à un des intervenants d'interrompre l'autre au cours de l'échange. Il est donc de coutume de poser une question, puis d'attendre la réponse et vice versa.

b. Les mondes virtuels :

D'après [Bel08], un monde virtuel (ou métavers) se définit comme un programme informatique permettant à une multitude d'utilisateurs de partager diverses activités. Ce qui différencie un monde synthétique d'un jeu est sa persistance. Lorsqu'un joueur se déconnecte de l'espace virtuel, ce dernier continue à fonctionner et à accueillir d'autres utilisateurs. A l'instar d'un réseau social, on peut considérer qu'un monde virtuel est entièrement basé sur les utilisateurs et leurs interactions. Ce qu'apporte un métavers par rapport à un réseau social est la notion d'espace et d'avatar. Par exemple, on peut différencier une personne proche de nous physiquement d'une autre se trouvant dans une autre pièce.

Les mondes virtuels peuvent être de deux types. Soit le monde permet d'atteindre des buts précis afin de gagner une mission, soit il est essentiellement basé sur la sociabilisation des joueurs entre eux. Le premier type de monde est ce que l'on appelle communément des jeux de rôle massivement multi-joueurs où le principe est d'interagir avec d'autres utilisateurs afin d'accomplir des missions. Des jeux comme World Of Warcraft ou Anarchy Online se classent dans cette catégorie. Le deuxième type de monde virtuel est ce l'on appelle des jeux sociaux massivement multi-joueurs. Dans ces univers, le joueur n'a pas de mission à accomplir. Le but de ces mondes est de partir à la rencontre des autres utilisateurs. Ils peuvent partager des discussions via chat ou toute autre activité praticable dans le monde réel comme danser ou partager un repas. Second Life et OpenSim représentent cette catégorie de logiciel. Dans ce papier, nous nous intéresserons essentiellement à cette deuxième catégorie de mondes virtuels.

Les jeux sociaux massivement multi-joueurs implémentent en général un moteur physique réaliste tout en laissant la possibilité à l'utilisateur de prendre des libertés avec celui-ci (faire voler son avatar dans Second Life). Ces jeux ont comme particularité de donner une grande liberté aux utilisateurs en leur permettant de changer d'apparence, de créer des objets ou bien d'acheter une maison. Réaliser ces différentes actions nécessitent de maîtriser le logiciel ce qui induit en général de devoir s'intégrer dans la communauté des utilisateurs du métavers afin d'obtenir de l'aide. Comme l'explique [MR07], la dernière composante importante de ces univers est leurs systèmes économiques. En général, ces univers permettent d'échanger des unités monétaires réelles en unités monétaires virtuelles. Dans Second Life, on peut échanger 1 US\$ pour environ 250 Linden\$ qui permettent d'acheter des objets ou même des îles virtuelles.

3. Outils utilisés :

Nous avons effectué notre projet de recherche en collaboration avec l'entreprise GAEANOVA. Cette start-up crée des mondes virtuels pour les entreprises et les collectivités afin de dématérialiser des conférences et abolir la barrière de la distance entre les intervenants. Pour cela, GAEANOVA utilise une solution développée par la société A WORLD FOR US qui permet de générer des mondes virtuels. GAEANOVA souhaite fournir une version virtuelle de la session posters de la conférence WWW'2012 qui se déroulera à Lyon en avril 2012.



Figure 2 : La cité internationale virtualisée à l'aide de la plate-forme GAEANOVA

Les mondes créés par GAEANOVA permettent à une multitude d'utilisateurs d'interagir avec d'autres personnes dans un environnement 3D (cf. figure 2). Ces interactions peuvent passer par des gestes (se serrer la main, lever le doigt), par un système de chat ou par une discussion via microphone.

La plate-forme permettant de créer les mondes virtuels est basée sur le moteur de jeu Unity3D. Afin de mener ce projet à bien, nous nous sommes donc formés à la création de scène 3D à l'aide de ce moteur. Il est possible de coder les différentes interactions entre les objets d'une scène à l'aide de scripts C#, Javascript ou BOO. GAEANOVA utilisant des scripts C# pour créer leurs métavers, nous avons décidé d'utiliser la même technologie.

4. La problématique :

Pour l'instant, les jeux sociaux massivement multi-joueurs basés sur des univers virtuels ont deux grands défauts. Le premier défaut est une prise en main difficile pour les néophytes. Ce défaut est lié au grand degré de liberté donné aux utilisateurs. Par exemple, on peut se déplacer n'importe où dans le monde mais il n'est pas évident de savoir comment se rendre à une destination précise. Le deuxième défaut est une sensation de monde vide lorsque peu de gens sont connectés. Les mondes virtuels se basant essentiellement sur les interactions entre les utilisateurs, il est difficile d'immerger l'utilisateur dans l'univers lorsque peu de gens sont connectés.

Le but de ce projet de recherche était donc d'implémenter une solution permettant de guider les nouveaux utilisateurs ainsi que rendre le monde plus vivant et donc plus immersif.

La solution adoptée a été de développer des agents intelligents qui auront pour but de guider au maximum les nouveaux utilisateurs lors de la session posters de la conférence WWW'2012.

5. Travail réalisé :

a. Présentation des agents intelligents :

Le premier agent intelligent a pour but d'accueillir chaque nouvel utilisateur. Pour cela, il ira à sa rencontre puis le guidera tout au long de sa visite de la session posters virtuel. Nous avons appelé cet agent, « l'agent d'accueil ».

Deux autres types d'agents ont été également développés afin de rendre l'univers plus vivant et immersif. Le premier type d'agent sera un « agent de renseignement ». Il aura pour fonction de répondre aux questions des utilisateurs portant sur des termes techniques. Cela permettra à l'utilisateur de récupérer de l'information tout en restant immerger dans le monde virtuel.

Le deuxième type de robot sera des « agents de figuration ». Ils auront pour seul but d'animer l'espace en effectuant des rondes, en parlant avec d'autres agents ou en discutant de différents sujets avec l'utilisateur.

i. L'agent d'accueil :

Nous avons donc développé un agent ayant pour but d'accueillir l'utilisateur lorsqu'il se connecte au monde virtuel. Cet agent est proactif car il vient chercher l'utilisateur pour lui proposer d'effectuer une visite personnalisée de la session posters. Pour déduire la liste des posters pouvant intéresser l'utilisateur, l'agent propose de faire une recherche par mots clés ou par thème de recherche sur l'ensemble des posters présents lors de la conférence. Une fois que l'utilisateur a saisi les informations qui l'intéressent, l'agent effectue une requête Ajax/SPARQL sur le serveur du Semantic Web Dog Food (<http://data.semanticweb.org/>). Ce serveur contient l'ensemble des papiers présentés lors des conférences www depuis 2007 et contiendra également ceux de 2012 ainsi que les posters correspondants. Tous les posters étant disposés sur les murs de notre monde virtuel, le robot consolide la liste des posters intéressants l'utilisateur à l'aide d'un fichier XML permettant de faire la correspondance entre le nom d'un poster et sa position dans la scène 3D. Une fois l'itinéraire calculé, l'agent prend en charge l'utilisateur en prenant le contrôle des mouvements de son avatar afin de l'amener au premier poster. Une fois arrivé au poster, il libère l'avatar de l'utilisateur et lui fait un résumé du poster (cf. figure 3). Il répète la même opération jusqu'à ce que la visite soit terminée. Une fois la visite terminée, l'agent quitte le monde virtuel.



Figure 3 : L'agent d'accueil présente un poster

Cet agent se base totalement sur les informations présentes sur le serveur du Semantic Web Dog Food. Il a donc l'avantage d'être réactif en prenant en compte l'ajout ou la modification de papiers. Pour l'instant, nous n'avons pu tester cet agent qu'avec les papiers présentés lors de la conférence www2011.

ii. Agent de renseignement :

Cet agent est sommaire dans sa conception. Il propose à l'utilisateur de faire une recherche sur un terme. Une fois que l'utilisateur a saisi un terme, l'agent propose une liste d'articles Wikipédia correspondants. L'utilisateur peut ensuite sélectionner l'article qui l'intéresse le plus. Une fois l'article sélectionné, l'agent renvoie le résumé de l'article se trouvant en haut de page (cf. figure 4). Afin de récupérer la liste des articles ainsi que le contenu des articles, l'agent effectue des requêtes Ajax afin de consommer le service web de l'API Wikipédia.

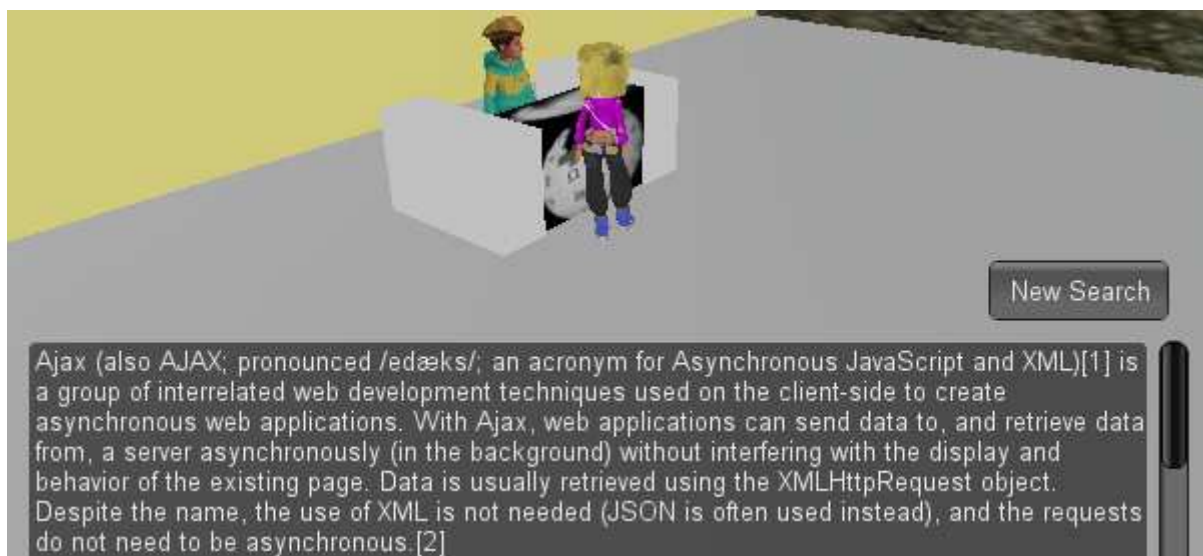


Figure 4 : L'agent de renseignement en action

Cet agent est plus simple que l'agent d'accueil. En effet, il n'est pas proactif (il ne se déplace pas vers l'utilisateur). Pourtant il apporte une plus forte immersion dans l'univers en permettant d'effectuer des recherches sans quitter la session posters. L'utilisateur n'a donc pas besoin d'ouvrir un nouvel onglet de son navigateur afin d'effectuer une recherche. Il serait intéressant d'améliorer ce robot en lui permettant de collecter des informations à partir d'autres sources de données que Wikipédia.

iii. Agents de figuration :

Ce type d'agents permet de renforcer le sentiment d'immersion en animant le monde virtuel même lorsqu'aucun autre utilisateur n'est connecté. Par manque de temps, nous n'avons pu qu'implémenter des versions rudimentaires de cet agent.

Le premier robot est un patrouilleur effectuant une ronde en passant par plusieurs points de passage prédéfinis. Il sert donc à créer une impression de mouvement dans le monde virtuel de la conférence.

Le deuxième robot enchaîne diverses animations de manière cyclique. Lorsque l'on place plusieurs agents de ce type ensemble, ils donnent l'impression d'interagir ensemble. Ce robot a également la possibilité de parler à l'utilisateur. En fait, il lui adresse la parole lorsque ce dernier arrive à proximité (cf. figure 5).



Figure 5 : Agent de figuration adressant la parole à l'utilisateur

Afin de renforcer le sentiment d'immersion, ces agents pourraient être grandement améliorés. Par exemple, les patrouilleurs pourraient s'arrêter à proximité d'autres robots afin d'interagir avec eux durant un temps prédéterminé avant de reprendre leur ronde. On pourrait également configurer un patrouilleur pour qu'il interrompe l'utilisateur lors de sa visite afin de le mettre au courant d'un événement qui pourrait se dérouler dans la conférence réelle WWW'2012 (par exemple, keynote de Tim BERNERS-LEE).

b. Implémentation des agents :

La société A WORLD FOR US a commencé le développement d'un moteur d'intelligence artificiel basé sur des arbres de comportements. Malheureusement, ce moteur n'était pas totalement disponible au commencement de notre projet de recherche. Nous avons donc décidé de créer notre propre scène 3D comportant des agents intelligents implémentés sous forme de machine d'états. Même si les machines d'états nous apportaient moins de souplesse que les arbres de comportements, cela nous a permis de créer rapidement des agents intelligents. En effet, développer intégralement un moteur d'I.A. gérant les arbres de comportement n'était pas réalisable dans le temps imparti. Une fois que la société A WORLD FOR US nous a fourni son moteur d'I.A. nous avons pu transformer nos machines d'états en arbre de comportements.

Dans les chapitres suivants, nous allons donc vous présenter l'implémentation de nos machines d'états ainsi que le principe des arbres de comportements. Nous présenterons également la technique utilisée par les agents d'accueil et les agents de renseignement pour récupérer des données stockées sur d'autres serveurs (Semantic Web Dog Food et Wikipédia).

i. Les machines d'états :

La machine à états finis est le système le plus couramment utilisé pour programmer une intelligence artificielle du fait de sa relative simplicité d'implémentation. Néanmoins, elle permet des comportements cohérents et évolués. Son fonctionnement consiste en un enchaînement d'états dans lesquelles on effectue

une série d'actions. Le point faible de ce système est qu'il ne gère pas l'imprévu ou le semi-imprévu, c'est-à-dire la capacité à s'adapter à un évènement qui n'a pas été prévu par le développeur.

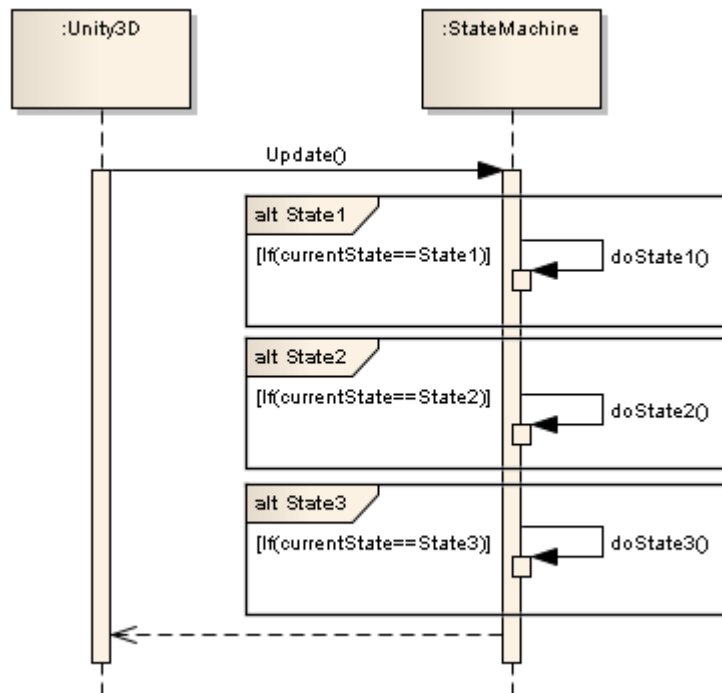


Figure 6 : Fonctionnement des machines d'états contrôlant les agents

L'implémentation de nos machines d'états est assez rudimentaire. Comme le montre la figure 6, lorsque l'on crée une classe C# qui sera utilisée par Unity3D, on peut décider d'implémenter la méthode abstraite Update(). Cette méthode est appelée par Unity3D à chaque rafraîchissement d'image. Nous avons donc décidé de l'utiliser comme une boucle infinie où la machine d'états teste la valeur de son état courant pour aller dans une fonction implémentant le comportement associé. A la fin de chaque fonction implémentant un comportement, on effectue des tests pour savoir si la machine doit changer d'état. Par exemple, si l'état courant de la machine d'états est « marcher » alors l'agent marche pour arriver à une destination précise. Lorsque l'agent arrive à destination, la machine d'état passe à l'état « attendre ».

ii. Arbres de comportements :

Les arbres de comportement ne sont utilisés que depuis très peu de temps dans les systèmes intelligents. Ce système gère mieux les imprévus que les machines d'état.

Etant un arbre (cf. figure 7), cette modélisation permet de prioriser des comportements par rapport à d'autres en effectuant des parcours en largeurs sur certaines branches. Il permet de tester un nœud, en cas d'échec, on teste le frère ; si celui-ci est valide, on explore le sous arbre de ce nœud. Et dans le cas où l'on arrive au nœud le plus à droite de l'arbre (le nœud le moins prioritaire), on explore le sous arbre de celui-ci.

Cela permet d'avoir un comportement pour la quasi-totalité des évènements possibles dans l'environnement présent. L'une des manières les plus simples d'implémenter ces arbres est le XML. Cette modélisation est utilisée dans de nombreux jeux vidéo comme Halo.

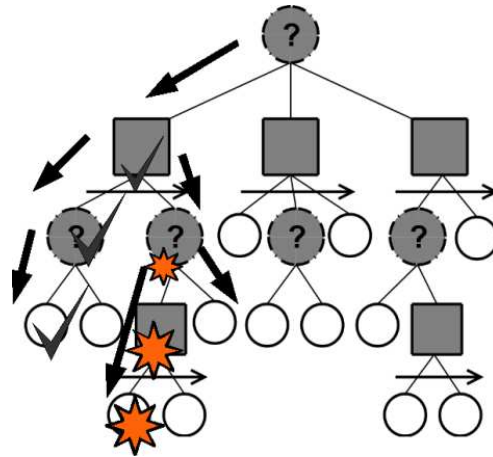


Figure 7 : Exemple d'arbre de comportements

iii. Récupération des données à partir de serveurs distants :

Comme le montre la figure 8, les scripts C# intégrés dans le WebPlayer Unity3D ne peuvent pas directement exécuter des requêtes AJAX. Dans notre cas, nous avons besoin de cette fonctionnalité pour développer l'agent d'accueil et l'agent de renseignement qui vont respectivement récupérer des données sur les serveurs du Semantic Web Dog Food et sur ceux de Wikipédia.

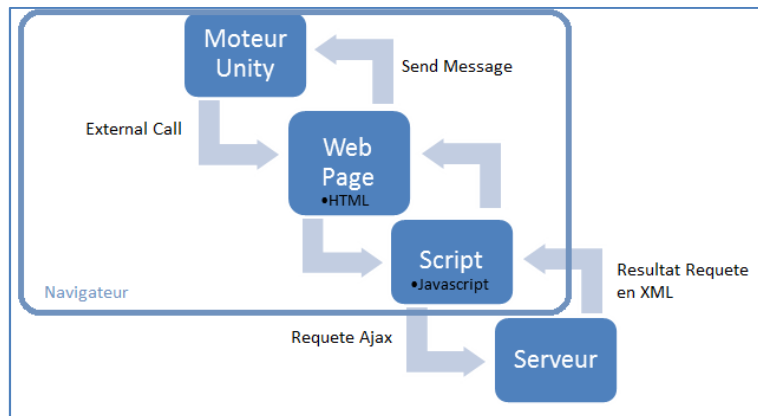


Figure 8 : Récupération des données en utilisant des requêtes AJAX

La solution a été de créer des scripts Javascript exécutant des requêtes AJAX puis de les inclure dans la page HTML contenant également le WebPlayer Unity3D. Une fois ces scripts inclus dans la page, les classes C# peuvent appeler des fonctions Javascript à l'aide de la méthode ExternalCall(). Une fois les résultats des requêtes obtenus les scripts Javascript les transfèrent au moteur Unity3D à l'aide de la méthode SendMessage().

6. Conclusion

Durant ce projet, nous avons donc été soumis à différentes contraintes qui nous ont permis de dégager une problématique plus appropriée au contexte de la session posters virtuelle. Il nous a paru plus opportun de créer plusieurs agents intelligents ayant chacun un rôle précis au sein de l'univers virtuel plutôt que de créer un seul agent « multitâches ». Il s'est donc dégagé 3 grands types d'agents :

- L'agent d'accueil qui présente le lieu et dirige l'utilisateur vers les posters souhaités. Le temps d'appréhension de la plate-forme s'en retrouve fortement diminué.

- L'agent de renseignement qui permet d'aller chercher des informations sur Wikipédia sans avoir à quitter l'univers virtuel. Cela ajoute de l'immersion dans le monde puisqu'il permet d'apporter du contenu « de l'extérieur » dans l'univers.
- L'agent de figuration qui est simplement là pour créer de la vie dans le monde afin que l'utilisateur n'ait pas la sensation d'être seul. Cela permet d'animer l'environnement virtuel.

Néanmoins, cette expérience peut être grandement améliorée en perfectionnant les différents agents intelligents. Par exemple, on pourrait intégrer la fonctionnalité text-to-speech. Ainsi les agents utiliseraient l'oral comme moyen d'expression et le réalisme de l'univers en sera renforcé. L'agent de renseignement pourrait également récupérer des informations à partir de plusieurs sources de données afin de les synthétiser pour l'utilisateur. Enfin, les agents de figuration pourraient interagir entre eux et avec l'utilisateur de manière à fournir des informations sur des événements à venir dans l'univers virtuel ou dans le monde réel.

7. Références

- [Bel08] BELL, M. (2008). *Toward a Definition of "Virtual Worlds"*. Indiana University.
- [BP98] BRASSAC, C., & PESTY, S. (1998). Simuler la conversation : un défi pour les systèmes multi-agents. Dans S. DELISLE, B. CHAÏB-DRAA, & B. MOULIN, *Analyse et Simulation de conversations : De la théorie des actes de discours aux systèmes multi-agents*. Lyon: L'interdisciplinaire,.
- [Bui05] BUISINE, S. (2005). *Conception et Evaluation d'Agents Conversationnels Multimodaux Bidirectionnels*. Paris.
- [Chi98] CHICOISNE, G. (1998). *Outils et pistes pour la pratique du dialoguisme entre agents*. JFIADSMA.
- [FG96] FRANKLIN, S., & GRAESSER, A. (1996). *Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents*. Consulté le 01 2012, sur <http://www.msci.memphis.edu/>: <http://www.msci.memphis.edu/~franklin/AgentProg.html>
- [GC05] GOLANSKI, C., & CAELEN, J. (2005). *Test d'utilisabilité de l'agent conversationnel « Angela »*. Grenoble, France.
- [MR07] MENNECKE, B., & ROCHE, M. (2007). *Second Life and other Virtual Worlds : A Roadmap for Research*. Montréal: Twenty-Eighth International Conference on Information Systems.