

Bot guía para un ambiente virtual inmersivo.

Alejandro Altube¹, Iris Sattolo¹, Liliana Lipera¹,

¹ Facultad de Informática Ciencias de la Comunicación y Técnicas Especiales, Universidad de Morón- Cabildo 183 Morón

alealtube@hotmail.com, iris.sattolo@gmail.com, llipera@unimoron.edu.ar,

Abstract. Los Ambientes Virtuales Inmersivos (AVIs) presentan características que son aprovechables para la simulación ya que los objetos que lo componen pueden contener script resultando ser objetos inteligentes. La posibilidad de incorporar Agentes Virtuales Autónomos (AVAs), para poder guiar a usuarios novatos dentro del metaverso construido en la Universidad de Morón, dio origen a este trabajo. En el mismo se presentan las características de un agente reactivo simple y su implementación en un entorno virtual inmersivo, utilizando las herramientas que brinda la plataforma Opensim. El bot resultante utiliza funciones propias de la plataforma NPC y funciones del lenguaje LSL.

Keywords: Inteligencia Artificial, Ambientes Virtuales Inmersivos, Agentes Reactivos Simples.

1 Contexto

Este trabajo es parte del proyecto de investigación que se está llevando a cabo en la FICCTE (Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales) perteneciente a la Universidad de Morón titulado: “Ambientes virtuales inmersivos, elementos interactivos”. Es la continuación del proyecto realizado en los años 2014-2016: “Ambientes virtuales inmersivos de aprendizaje orientados a la capacitación y al desarrollo de planes de evacuación”.

Para la construcción del metaverso se utilizó el software Opensim (aplicación servidora multiplataforma y multiusuario). Durante el período de investigación se presentaron los avances en los congresos CACIC 2013, WICC 2014 y 2015, TE&ET 2014 y 2015.

Actualmente, éste espacio virtual está montado en una isla definida en un servidor de prueba en la Universidad de Morón al cual se accede en forma local a través de una red LAN.

Una de las líneas de investigación del presente proyecto plantea la posibilidad de dotar al entorno y a los diferentes tipos de objetos de autonomía o inteligencia propia.

Dado que el usuario novel no está preparado para vivir situaciones desconocidas, al entrar las primeras veces al mundo, se propuso la incorporación de agentes inteligentes que habiten el metaverso aun cuando no existan usuarios reales y le brinden alguna ayuda.

2 Introducción

Se presenta en esta sección una síntesis de los ambientes virtuales inmersivos, la plataforma utilizada para el desarrollo del trabajo y una introducción a los agentes racionales.

2.1 Ambientes virtuales inmersivos

Los mundos virtuales inmersivos, (ambientes virtuales inmersivos, AVIs o metaversos) son entornos tridimensionales generados por computadora que recrean escenarios reales o imaginarios. En ellos el usuario puede interactuar a través de un avatar, su representación virtual, con el mundo y con otros usuarios.

Actualmente los ambientes virtuales muestran distintas actividades, como ejemplo de ellas podemos citar en el área de educación y pedagogía, el espacio desarrollado en Second Life (Harvard Law School) [1], donde los estudiantes interactúan entre sí como expositores y jueces de un tema. En turismo, según Cruz Ruiz en su tesis de grado [2], Grecia lidera un proyecto de turismo virtual de sus Islas (Epsilon, s.f.).

En cuanto al desarrollo de agentes inteligentes en los AVIs el trabajo propuesto por Fabiana Santiago Sgobbi et al [3] propone un asistente para el cuidado de la salud de los ancianos. En Gonzales Martínez [4] se propone un agente conversacional utilizando herramientas de procesamiento natural. En la aplicación desarrollada por Johnson et al [5], utiliza el entorno virtual para mostrar el uso de agentes virtuales y su interacción con los objetos inteligentes del entorno.

Estos espacios, en educación, permiten a los usuarios relacionarse en forma remota admitiendo mayor interactividad en contraparte con las plataformas bidimensionales utilizadas actualmente. El desarrollo de mundos virtuales, como innovación en la práctica educativa, ha creado nuevas posibilidades en la adquisición de conocimientos y la formación de procesos de aprendizaje en línea.

Según el informe de Gartner (2007) *“Las nuevas arquitecturas, los nuevos espacios interconectados, los nuevos estándares abiertos posibilitan que los usuarios de Internet tengan algún tipo de presencia dentro de los mundos virtuales basados en la imagen del usuario a través de su avatar o de entornos multiusuario y, aparentemente esta presencia ha aumentado”* [6]. A estos entornos Digitales Inmersivos se los puede considerar partes vitales del Web3D emergente.

En el año 2001 Castronova [7] definió las tres características más importantes de los mundos virtuales: Corporeidad, Persistencia, Interactividad.

“Corporeidad : las personas acceden al programa a través de una interfaz, la cual simula un entorno físico donde el usuario observa y se mueve en primera persona; el entorno generalmente está sometido a las leyes de la física y se caracteriza por la escasez de recursos”.

“Persistencia: el programa sigue funcionando independientemente, aun cuando los usuarios estén conectados o no, recuerda la localización de personas y cosas como también quién es propietario de los objetos”.

“Interactividad: es la cantidad posibles de interacciones que el usuario puede realizar en el mundo virtual, siendo la interacción la capacidad que el usuario tiene de modificar el medio”.

2.2 Opensim

Opensim brinda un espacio vacío, posibilita la creación de un ambiente con objetos editables que pueden simular entes reales al adquirir comportamiento mediante scripts. Estos elementos pueden interactuar entre sí o con los avatares/usuarios. Los espacios consisten en una o más regiones que componen el metaverso. Para acceder al servidor Opensim desde los usuarios, se utilizan visores (programas clientes) que permiten controlar el avatar, editar objetos y crear scripts. Una gran variedad de ellos existen en el mercado, creados por grupos independientes y de uso libre.

A los objetos que demuestran la posibilidad de interacción con el entorno, se los suele llamar objetos inteligentes.

Otra posibilidad que permiten estas plataformas es la incorporación de Agentes Virtuales Autónomos (AVAs) los cuales se definen como: “entidades de software que representan personajes virtuales inmersos en escenarios 3D, teniendo la capacidad de reaccionar al entorno que los rodea, simulando tener vida propia”. Estos agentes dentro de Opensim son llamados bots.

2.3 Agentes

Se presentan las características de los agentes y se intenta catalogar a los AVAs (Agentes virtuales autónomos) dentro de las diferentes clasificaciones.

Existen distintas definiciones sobre agentes dependiendo el punto de vista utilizado, según Franklin y Graesser: “*un agente autónomo es un sistema que ubicado dentro de un entorno, percibe dicho entorno y actúa sobre él, a lo largo del tiempo, en busca de cumplir sus objetivos, y por tanto para llevar a cabo lo que percibe en el futuro*” [8].

Según Wooldridge y Jennings [9] existen dos nociones de agente, una débil y una fuerte. Un agente débil es aquel que goza de las siguientes propiedades:

- Autonomía: Operan sin una intervención directa de humanos, tienen un cierto grado de control de sus acciones y estado interno.
- Habilidad social: Se comunican con humanos u otros agentes a través de sus interfaces de comunicación.
- Reactividad: Actúa como resultado de cambios en su entorno. En este caso, un agente percibe el entorno y esos cambios dirigen su comportamiento.
- Proactividad: No solo actúan en respuesta a su ambiente, sino que son capaces de presentar un comportamiento oportunista, dirigido por objetivos, tomando iniciativas cuando sea apropiado.

Un agente fuerte, además de las características anteriores, posee una o más de las siguientes características:

- Nociones mentales: Posee creencias, deseos e intenciones.
- Racionalidad: Realizan acciones con el propósito de lograr objetivos.
- Adaptabilidad: Son capaces de aprender y cambiar su comportamiento en base a sus propias experiencias.
- Veracidad: No son capaces de comunicar conscientemente información falsa.

No es el fin de este trabajo nombrar las distintas características propuestas por los diversos autores, por tal motivo, se resolvió adoptar el esquema propuesto por Osorio et al[10] el cual sugiere que el agente puede ser catalogado según las características observada en él (Tipo de entidad, la similitud con el ser humano, según la arquitectura de control, las tareas que realiza, grado de autonomía, localización, tipo de entorno, tipo de actividad, tipo de interacción). En la tabla 1 se resumen las características observadas (columna 1), su parametrización (columna 2) y la categorización que nuestro bot tendría dentro de los AVAs (columna 3).

Tabla 1. Clasificación de Agentes de acuerdo a sus características.

Características observadas según	Clasificación	Agentes virtuales autónomos.
Tipo de entidad	Real (Humano, biológico, robot físico) Computacional (software, vida artificial)	Computacional (vida artificial)
Similitud con el humano	En cuanto a la estructura o Comportamiento	Estructura y comportamiento
Arquitectura de control	Reactivos, cognitivos, híbridos. Basados en estados, o en emociones	Reactivo Basado en Estados
Tareas que realiza	Transaccional, informativo, negocios, agentes de interfaz	Informativo
Grado de autonomía	Guiados, interactivos, autónomos, perceptivos.	Interactivos
Localización	Móvil, estacionario, distribuido	Móvil
Tipo de entorno	Escritorio, Red, Pedagógico, Virtual	Virtual
Tipo de actividad	Solitarios Social (grupos o colaborativos)	Colaborativo
Tipo de interacción	Con el usuario, otros agentes, con el entorno, o múltiples.	Interacciones múltiples.

2.4 Esquema de un agente reactivo simple

Según Nwana, los agentes reactivos se caracterizan porque pueden operar rápida y efectivamente, sin la necesidad de procesar una representación simbólica del entorno en que se encuentran, ya que constituyen una categoría de agentes que no posee modelos simbólicos externos [11]. Los agentes reactivos toman decisiones basadas en el presente, actúan siguiendo un esquema estímulo-respuesta según el estado actual del entorno en el que están inmersos.

Según Iglesia [12] un agente reactivo está formado por los siguientes tipos de módulos:

Módulos de interacción con el entorno (sensores y actuadores en la Figura 1).

Módulos de competencia (cada una de la capas en la Figura 1).

El funcionamiento es el siguiente: los sensores recogen información que es directamente enviada a los módulos de competencia. La llegada de información a uno

de estos módulos provoca en él una reacción que se transfiere al entorno a través de los actuadores.

Según el informe de la escuela Politécnica de Madrid, cuando los módulos de competencia trabajan en paralelo, lo normal es que aparezcan dependencias, por este motivo, se deben poder comunicar entre ellos. La comunicación normalmente utiliza tanto mecanismos como lenguajes de comunicación simples, y puede ser uno-a-uno o se puede realizar a través del entorno [13].

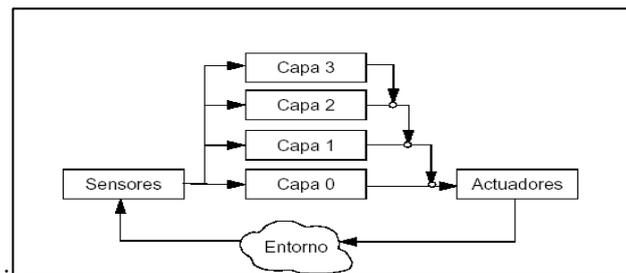


Figura 1 Arquitectura de un agente reactivo [12]

2.4 Bots

Existen dos alternativas que ofrece Opensim para la implementación de bots. La primera opción es crearlos mediante un script en LSL (Linden Scripting Language), y la segunda opción es mediante un cliente controlado por programa.

Los scripts son programados en LSL, lenguaje de programación basado en eventos, estados y funciones; creado para ser utilizado en ambientes de virtuales de Second Life.

LSL brinda una gran cantidad de funciones pero se han desarrollado en Opensim funciones propias llamadas OS. Dentro de las mismas, las NPC (Non Player Characters) que se utilizan para crear y manipular bots. Se describen aquí las más importantes para este trabajo:

Creación del bot: la función *osNpcCreate* permite su creación indicando nombre, apellido, posición inicial (dentro del ambiente) y apariencia. Esta apariencia se define dentro de una *notecard*.

Mover al bot: desde la posición actual hasta una fija. Siempre en línea recta. Para obtener un trayecto no recto se debe llamar varias veces a estas funciones. Se puede indicar que camine, corra o vuele.

Obtener su posición, rotación y dueño (número identificador del usuario que lo creó).

Guardar/cambiar de apariencia.

Comunicarse: Se pueden crear canales de comunicación para que los objetos interactúen entre sí o con usuarios y, en este caso, los bots.

Sentarse y pararse tanto en objetos como en el suelo.

Tocar objetos.

Ejecutar animaciones: Animar al bot, se pueden generar movimientos o usar los genéricos del Opensim y aplicarlos al bot. De esta manera se logra que el bot baile, salude.

En la segunda opción, implementación de bot a través de un cliente controlado por programa, para realizar la conexión se debe crear un programa en C# .NET, utilizando la librería OpenMetaverse. Esta librería, creada por *OpenMetaverse Foundation* y de acceso libre, es una interfaz que permite conectar usuarios a Opensim y ejecutar las acciones del avatar mediante funciones. Lógicamente, se necesita un usuario de Opensim disponible para la conexión.

3 Desarrollo

Se propuso en este trabajo la utilización de las herramientas que brinda la plataforma Opensim, sin la introducción de herramientas de Inteligencia Artificial.

Se planteó adoptar la arquitectura de un agente reactivo simple, utilizando para la construcción del mismo la alternativa que ofrece el entorno, creando el bot con funciones NPC.

Adoptando la definición que Russell y Norvig ofrecen de agentes en forma genérica y conceptual reflejada en la figura 2, “Un agente es cualquier cosa capaz de percibir su medioambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores” [14], se analiza cómo, el bot, puede realizar esas acciones.

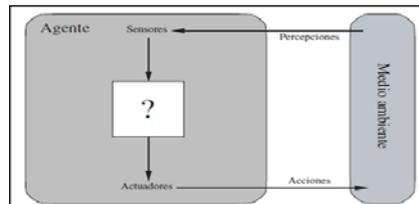


Figura 2 Arquitectura de un agente simple

De acuerdo con Aylett y Luck [15], la percepción es la interacción del agente con su entorno, el agente debe percibir diferentes objetos, pudiendo ser estos estáticos (como las paredes, muebles y otros), con movimiento (puertas que se abren), avatares en movimiento como también otros agentes.

El mecanismo empleado por el bot para esta interacción será la utilización de sensores para obtener información del entorno como disparador. Las funciones `llsensor`, `llSensorRepeat`, son provistas por el lenguaje pero no son propias del bot.

En cuanto al comportamiento que se espera del bot, será la acción que debe realizar de acuerdo a cada percepción. Podrá comunicarse, moverse y tocar objetos para interactuar con el mismo.

Existe la posibilidad de que el bot de respuesta a preguntas del usuario mediante el chat del Opensim, generando así un chatbot. En general existen movimientos pre-definidos que se pueden utilizar para dar mayor sensación de realismo. Los actuadores serán los responsables de realizar estas acciones. La figura 3 muestra la arquitectura propuesta para implementar el bot.

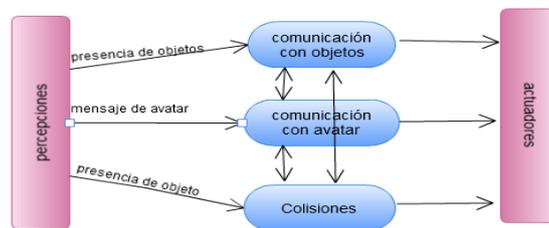


Figura 3. Arquitectura del bot.

3.1 Ambiente

El bot se moverá dentro del ambiente ya creado durante el proyecto “Ambientes virtuales inmersivos de aprendizaje orientados a la capacitación y al desarrollo de planes de evacuación” (figura 4) En este caso se propone un “bot guía” para dar soporte a posibles usuarios novatos que deseen conocer las distintas ubicaciones del entorno. El bot se encargará de guiar a los usuarios a locaciones específicas, enseñando las tareas que se pueden realizar en las mismas.



Figura 4 Ambiente creado en la Universidad de Morón

El bot se situará en el punto de inicio marcado con una flecha roja. Cuando el usuario ingrese al edificio el bot se acercará y preguntará, mediante el chat, si necesita dirigirse a alguna sala o locación en especial. Éstas se referencian en el mundo por las coordenadas (x,y,z). Tales como: Puerta de entrada (28,212,24); Sala de cascos (136,216,24); Sala de máquinas (18,214,24).

3.2 bot guía

Se muestra la implementación del módulo comunicación con el avatar, referenciando a cada percepción el comportamiento y el actuador correspondiente, a su vez, las acciones que realiza el bot. El paso entre ellas dependerá de las percepciones recibidas del usuario tanto en el sensor como en el chat y el menú (Figura 5).

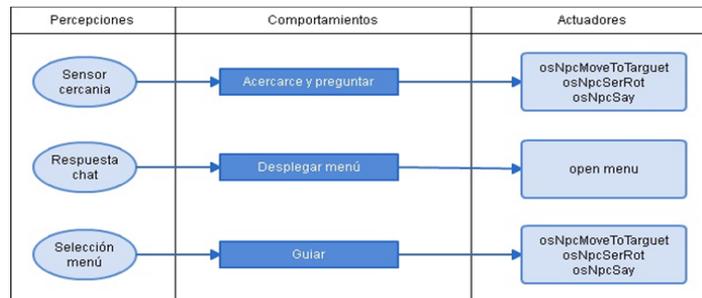


Figura 5 Percepciones y actuadores del bot.

Cuando el *Sensor cercanía* detecta a un avatar, el bot preguntará al usuario si necesita sus servicios esperando una respuesta por sí o por no. El accionar en cada alternativa será la siguiente:

1. Respuesta “sí”:
 - a. Se despliega un menú con las locaciones disponibles.
 - b. El usuario selecciona la locación.
 - c. El bot guía al usuario hasta la locación seleccionada y le brinda información útil sobre la misma.
2. Respuesta “no”:
 - a. El bot se despide y vuelve a su ubicación inicial.

3.4 Implementación de prueba

Las funciones NPC que brinda Opensim no son suficientes por sí solas para llevar a cabo el desarrollo del bot y sus funciones, por lo que se resolvió combinarlas con el resto de las funciones LSL. Ya se describieron las funciones NPC, en cuanto a las LSL usadas se necesitaron las siguientes: sensor de cercanía (del avatar con el bot), el menú de locaciones, y la comunicación entre bot y avatar vía chat, entre otras.

Para la implementación del bot se requirió la construcción de diferentes objetos (la figura 6 muestra una captura de pantalla de los mismos):

Objeto Bot guía:

Script ‘botguia’: contiene el `OsNpcCreate`, se ejecutará al hacer click sobre el objeto que lo contiene, se comunica con el objeto ‘Comandos del bot’, enviándole el número identificador del bot creado, a través de un canal establecido.

Apariencia: Notecard con la apariencia que tomara el `OsNpcCrate`.

Objeto Comandos del bot:

Script ‘comandosbot’: se ejecutará cuando reciba el “mensaje” con el número identificador del bot npc por el canal establecido. Contendrá el código del bot.

Objeto Copiar apariencia:

Script ‘copiarapariciencia’: simplemente copia la apariencia del avatar usuario. Lo importante es que al momento de copiar la apariencia se tenga adjuntado (lleve consigo) el objeto ‘Comandos del bot’, ya que esta apariencia será la que tome el objeto ‘Bot guía’ al momento de crear el bot.

AparienciaBot: *Notecard* que se creara al hacer click sobre este objeto.



Figura 6 Objetos para implementar el bot

Al no existir funciones NPC que permitan adjuntar un objeto al bot, una vez creados los objetos que contienen los scripts responsables de la comunicación, se deben colocar en el inventario del avatar que será copiado. Si bien resultará más laborioso a la hora de modificar el código, ya que se deberá repetir el proceso: *copiar apariencia y adjuntar objeto* (acoplamiento), se logra de este modo la comunicación entre el avatar y el bot mediante la instancia *comandos del bot*. Se muestra en la figura 7 un diagrama de clases para la implementación del mismo.

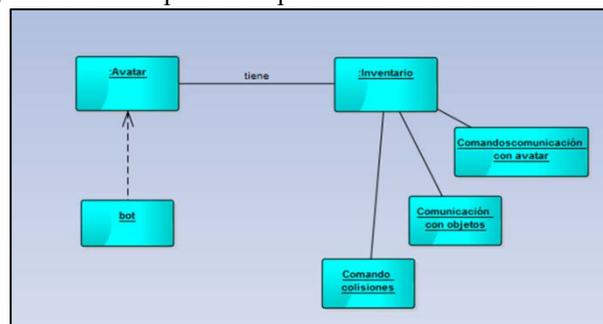


Figura. 7 Diagrama de clases para implementar el bot

Conclusiones y futuras líneas

Se logró que el bot guía realice tareas con las características de agentes débiles: autonomía, habilidad social y reactividad.

Resultó ser un agente reactivo simple, ya que actúa sobre el entorno en base a las percepciones recibidas.

Mayor interacción entre el usuario y el ambiente al crear un bot que simule un personaje más real.

En cuanto a las futuras líneas de investigación se nombran:

- Posibilidad de implementación de un bot a través de un cliente/usuario controlado por programa.
- Creación de agentes conversacionales, llamados chatbots, mediante AIML (Artificial Intelligence Mark-up Language).

- Implementación de multiagentes en ambientes virtuales.
- Búsqueda de posibilidad de creación de un agente virtual que cumpla con definiciones de agente fuerte, tales como: Agregar emociones a los agentes virtuales, Aprendizaje en agentes virtuales, Adaptar comportamiento de agentes en base a estado anímico y experiencia.

Referencias

1. <http://maps.secondlife.com/secondlife/Berkman/123/219/24?title=Harvard+Law+School&msg=The+Berkman+Center+for+Internet+%26+Society+at+Harvard+Law+School+has+held+multiple+classes+within+Second+Life>.
2. <http://eprints.ucm.es/13048/1/memoria.pdf>
3. Fabiana Santiago Sgobbi, Felipe Becker Nunes, Liane Taroucco A UTILIZAÇÃO DE AGENTES INTELIGENTES NO APOIO AO AUTOCUIDADO DE IDOSOS- <http://seer.ufrgs.br/index.php/renote/article/view/53520/33032>
4. Gonzalez Martínez. Agente conversacional con habilidades para la venta de productos en un mundo 3D - <http://www.informaticahabana.cu/sites/default/files/ponencias/EDU058.pdf>
5. W. L. Johnson, and J. Rickel, "Steve: An Animated Pedagogical Agent for Procedural Training in Virtual Environments", Sigart Bulletin, ACM Press, vol. 8, number 1-4, 16-21, 1997.
6. Gartner Inc. (2007). Gartner Says 80 Percent of Active Internet Users Will Have A "Second Life" in the Virtual World by the End of 2011. Retrieved May 10, 2008 from <http://www.gartner.com/it/page.jsp?id=503861>
7. Castronova, E. (2001). Virtual Worlds: A first-hand account of market and society on the cyberian frontier .
8. Intelligent Agents III Agent Theories, Architectures, and Languages is i tan agent, or just a program? A taxonomy for autonomous agents. http://www.springer.com/la/book/9783540625070?wt_mc=ThirdParty.SpringerLink.3.EPR653>About_eBook
9. Wooldridge M., Jennings N. R. (1995). Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, Vol. 10:2, pp. 115-152
10. Osorio, Musse, Santos, Heinen, Braun, Silva. Intelligent Virtual Reality Environments .Principles, Implementation, Interaction, Examples, and Practical Applications.
11. Nwana, H. (1996). Software agents: an overview. The Knowledge Engineering Review, 11(3):205-244. <http://www.btexact.com/projects/agents/publish/papers/review3.htm>.
12. Iglesias, C.A (1997) Fundamentos de los Agentes Inteligentes. Informe Técnico UPM/DIT/GSI 16/97.
13. Universidad politécnica de Madrid. Informe técnico. Arquitecturas de Agentes
14. 12 Rusell S, Norvig P. (2004). Inteligencia artificial: Un enfoque moderno 2da edición. Editorial Pearson.
15. Aylett, R. and Luck, M. (2000) "Applying Artificial Intelligence to Virtual Reality: Intelligent Virtual Environments". Applied Artificial Intelligence.