Juegos no tan serios para el aprendizaje de la gestión de proyectos de despliegue de redes de computadores

Juan J. Ramos-Muñoz[1,*,†], Felix Delgado-Ferro[1], Jonathan Prados-Garzon[1], Julia Caleya-Sánchez[1], Pablo Rodríguez Martín[1], Oscar Adamuz-Hinojosa[1], Natalia Chinchilla-Romero[1] and Jorge Navarro-Ortiz[1]

[1]*Department of Signal Theory, Telematics and Communications, University of Granada.*

Abstract

Parte del trabajo que realizará una persona que se dedique a la ingeniería consistirá en interaccionar con usuarios reales. Esta interacción debería trabajarse como una de las competencias transversales en las asignaturas de las carreras técnicas. En este trabajo se presenta la experiencia del uso de videojuegos serios para la simulación de proyectos de diseño de redes de computadores. En este tipo de videojuegos se implementan sistema de diálogos para simular conversaciones con posibles clientes en proyectos realistas.

Keywords

Juegos serios, redes de computadores, ingeniería informática, Godot Engine

1. Introducción

Uno de los aspectos muy valorados por las empresas relacionadas con las ingenierías es el de las competencias transversales o *soft skills*. En muchos de los puestos que se ofertan la persona contratada tendrá que ser la interlocutora de la empresa con la clientela. Normalmente es difícil trabajar algunas de estas competencias en las asignaturas durnate la carrera.

En este trabajo corto se describen los aspectos básicos del desarrollo en curso de un videojuego realizado para simular entrevistas con clientes en el ámbito de diseño e implementación de redes de computadores, de forma que hace ágil el desarrollo de nuevos juegos.

2. Contextualización

Las asignatura de *Planificación y Gestión de Proyectos Informáticos* del Máster Universitario en Ingeniería Informática de la Universidad de Granada aborda la gestión de proyectos, incluyendo desde las propias metodologías de desarrollo, planificación, estimación de costes, gestión de la calidad, entre otros.

Uno de los contenidos de la asignatura es el diseño, planificación y despliegue de redes, incluyendo su ciclo de vida, operaciones de gestión y mantenimiento. Esta parte de la asignatura tiene asignada 15 horas para su desarrollo. En esta parte de la asignatura se sigue le modelo de ciclo de vida de CISCO descrito en [1]. En este libro se incluye material tal como tanto listas de comprobación de tareas, plantillas para el análisis y diseño de las redes, etc. Esta metodología requiere recopilar datos acerca de los requisitos de diseño y implementación, además de la interacción con los usuarios de la red a implementar en la empresa pertinente.

Para su realización se diseñan unas sesiones prácticas en las que el estudiantado aborda un caso desde los distintos puntos de vista tratados en la parte teórica. Concretamente, se propone el análisis y diseño

IV Congreso Español de Videojuegos

*Corresponding author.

jjramos@ugr.es (J. J. Ramos-Muñoz); felixdelgado@ugr.es (F. Delgado-Ferro); jpg@ugr.es (J. Prados-Garzon); jcaleyas@ugr.es (J. Caleya-Sánchez); pablorodrimar@ugr.es (P. R. Martín); oadamuz@ugr.es (O. Adamuz-Hinojosa); nataliachr@ugr.es (N. Chinchilla-Romero); jorgenavarro@ugr.es (J. Navarro-Ortiz)

0000-0001-5428-3913 (J. J. Ramos-Muñoz); 0000-0003-3494-9487 (F. Delgado-Ferro); 0000-0001-9373-5767 (J. Prados-Garzon); 0000-0002-1703-8641 (J. Caleya-Sánchez); 0000-0002-7345-8923 (P. R. Martín); 0000-0002-7797-1150 (O. Adamuz-Hinojosa); 0000-0001-5992-8717 (N. Chinchilla-Romero); 0000-0003-4719-3156 (J. Navarro-Ortiz)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



Figure 1: Captura de pantalla del juego.

básico de una red dado un caso realista. Para poder incluir la parte de interacción con personas usuarias de la red a implantar, se define un juego de roles, en los que las personas a entrevistar se simulan dentro de un videojuego. A esto se le suma que el personaje se moverá por las instalaciones de la empresa cliente. Se procura de esta manera hacer también más amenas las tareas de análisis y diseño.

3. Diseño general del juego

El videojuego se diseña con un formato simplificado de RPG (Role-Playing Game). Concretamente, un personaje protagonista se mueve por escenarios donde puede hablar con otros personajes. Los diálogos entre personajes consisten en intercambio de frases que pueden elegirse de entre una lista. El personaje protagonista puede abrir puertas en los distintos escenarios. Las puertas pueden estar bloqueadas. Se pueden desbloquear tras hablar con otros personajes.

Dado que uno de los pasos de la metodología consiste en hacer una caracterización de la infraestructura de red que ya existe en la localización en la que se va a desplegar la nueva red, se implementan escenas en las que se puede visualizar dispositivos de red ficticios. En estas escenas, se presentan cableado con etiquetas, y conexiones de cableado de red en distintos puertos de conmutadores o encaminadores de red. Este tipo de escenas pretenden emular lo que se puede encontrar en un armario técnico de red en un caso real. Además, la mecánica en la que se tiene que inspeccionar este tipo de escenas también tiene como objetivo hacer más amena la tarea de análisis, al contrario de lo que ocurriría con la simple revisión de un informe escrito.

En la Fig. 3 puede apreciarse un ejemplo de la interfaz general del juego en la que el jugador se desplaza por los escenarios para explorar la localización.

4. Implementación

Las primeras versiones del juego se implementaron utilizando el editor RPG Maker MV[2] debido a tres razones principalmente: editores de escenarios fáciles de usar, puede exportar el juego para que se ejecute en un navegador web, y dispone de una biblioteca muy amplia de gráficos para construir escenarios y personajes.

Sin embargo, aunque la funcionalidad de RPG Maker excede los requisitos del tipo de juegos que se pretende desarrollar, se identificaron algunos inconvenientes: la edición de sucesos no era intuitiva, los diálogos debían programarse en un lenguaje de programación propio, y había que realizar comprobaciones repetidas para comprobar que las conversaciones eran completas. Por eso se decidió desarrollar

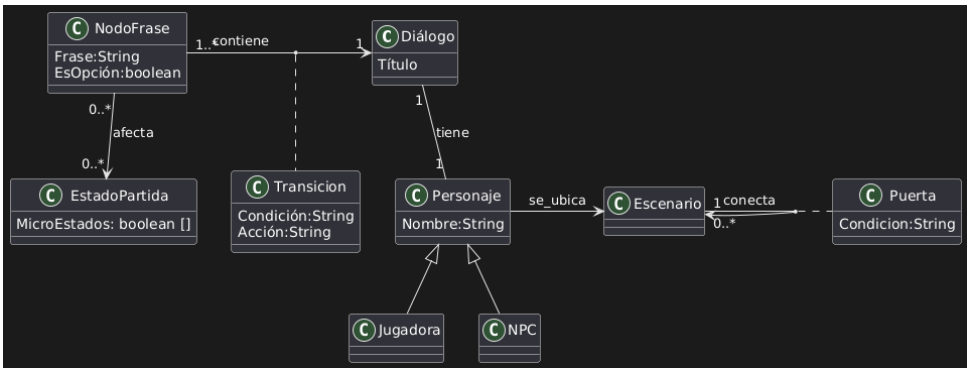


Figure 2: Diagrama de clases general del videojuego.

un marco más sencillo, pero más usable.

La biblioteca se desarrolló con Godot Engine 3.5 [3]. Se trata de un motor de juegos libre, que puede ser extendido, y cuyos *scripts* se basan en GDScript, muy similar a Python.

Aunque el diseño de la biblioteca tiene algunas dependencias con este motor, se procuró diseñarlo de forma que pueda portarse con los conceptos comunes de los motores de juego 2D. Así, se intenta aprovechar el uso de mapas de teselas, áreas simples de colisión o lectura y escritura de ficheros JSON (*JavaScript Object Notation*).

4.1. Diseño general de clases

Se lleva a cabo un diseño sencillo de clases con objeto de que pueda ser portable a otros lenguajes de programación, en caso de que se identifique un motor de videojuegos más adecuado en el futuro. Estas clases permiten implementar las funcionalidades básicas identificadas para este tipo de simulación. Este diseño de clases se esquematiza en la Fig. 4.1 a modo ilustrativo. El módulo más destacable es el de los diálogos, y su editor.

4.2. Módulo de diálogos

Los diálogos se estructuran como un grafo dirigido. Cada nodo es una frase de un personaje, y se establecen transiciones a los siguientes nodos. Los siguientes nodos son el listado de frases que aparecerán a continuación en la conversación.

Cada personaje tiene asociado un diálogo. Este diálogo se almacena como un fichero JSON.

Se ha desarrollado un editor de diálogos sencillo que facilita su elaboración de forma ágil. Para ello, se utilizan los *nodos de grafo* (*GraphNode* [3]) de Godot Engine. Para crear un diálogo, se irán conectando un nodo detrás de otro, asignando la secuencia de frases que se hayan establecido en el guion. Los diálogos pueden constar de decenas de nodos. En cualquier momento se pueden ramificar secuencias de diálogos, para dar varias opciones y enriquecer la interacción.

En la 4.2 se puede apreciar una parte del diálogo de un personaje. La conversación comienza en el nodo que solo tiene transiciones de salida.

En cada nodo se identifican los siguientes campos:

- Nombre de personaje. Se trata del identificador de cada personaje. Esto permite que se puedan representar conversaciones de más de dos participantes, o incluso monólogos.
- Frase. Es una cadena de caracteres que representa la frase que aparecerá en la conversación.
- Atributos de nodo. Si un nodo es del tipo "Opción", significa que se va a mostrar en una lista, de forma que será elegible por la persona que juega, si se cumple la precondición.
- Precondición. Es una expresión lógica que utiliza variables booleanas del estado del juego. Si se cumple, la frase de este nodo se mostrará. En caso contrario, permanecerá oculta. Esto permite que el diálogo pueda cambiar según eventos sucedidos en el juego. También puede usarse para

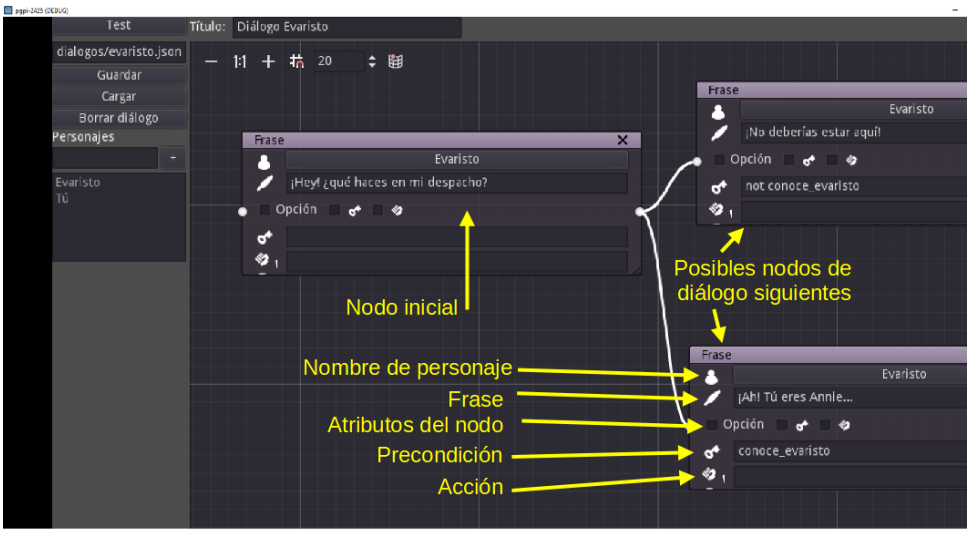


Figure 3: Detalle del editor de diálogos de un personaje.

no repetir alguna de las frases, si se crea una variable de estado que almacena si ha sido ya usada o no.

- Acción. Función que debe ejecutarse en caso de elegirse este nodo. Se aprovecha la facilidad que tiene GDScript para llamar a una función utilizando una cadena de caracteres con el nombre de la función a ejecutar.

5. Conclusiones y trabajo futuro

En este trabajo se ha presentado el diseño e implementación de un framework para el desarollo de juevos que permitan a estudiantes de ingeniería simular entrevistas con clientes y el análisis de infraestructuras de red. Este framework es adaptable a otros ámbitos, y agiliza el desarrollo de supuestos realistas. Hay que tener en cuenta que la adopción de supuestos distintos cada curso puede suponer una carga extra, que esta herramienta puede aliviar. Como trabajo futuro se pretende añadir plantillas para la generación de configuraciones automáticas de infraestructuras de red existentes, integrar recursos gráficos con licencias libres, y la inclusión de modelos generativos de texto para enriquecer los diálogos. Además, se espera realizar encuestas de satisfacción para evaluar la idoneidad de su uso en clase, frente a otras opciones tradicionales.

References

[1] P. Oppenheimer, Top-down network design, Pearson Education India, 2011.

[2] D. Perez, Beginning RPG Maker MV, The Expert's Voice in Game Development, 2nd ed. 2016. ed., Apress, Berkeley, CA, 2016.

[3] C. Bradfield, Godot Engine game development projects : build five cross-platform 2D and 3D games with Godot 3.0, 1st ed. ed., Packt Publishing, Birmingham, 2018 - 2024.