

Proyecto: Juego de Rol

Alumno: M^a Beatriz Medina Yañez

Profesor: Luis Castillo

- 1 Justificación**
- 2 Objetivos**
- 3 Inteligencia**
- 4 Características del juego**
- 5 Estructura de diseño**
- 6 Herramientas de programación**

1 Justificación

¿Por qué un juego de rol?

Un juego de rol permite implementar un sistema multiagente de forma que se pueda obtener una simulación de un entorno o mundo imaginario en el que una serie de agentes autónomos puedan tomar decisiones y actuar en el mismo.

Es un modo de analizar cómo perciben y actúan un conjunto de agentes ante situaciones que podrían simular un comportamiento real.

Cada agente tendrá disponible una serie de acciones y tendrá que elegir la que le parezca más correcta frente a una situación determinada. Es decir, lo más importante es tomar una decisión frente a un estado del mundo. Una vez tomada dicha decisión, será beneficiosa o no, tendrá éxito o no, según determinadas circunstancias, ya sean asociadas a otros agentes o al propio mundo. De esta forma, para darle más realismo al juego y para que no constituya un conjunto de reglas fijas, serán necesarias las probabilidades. Cada acción tendrá una cierta probabilidad de que tenga éxito o no. Dichas probabilidades serán más o menos altas según las características de cada agente. Este aspecto hace que el juego simule un comportamiento real, ya que si nos damos cuenta, cualquier ser humano toma siempre una decisión, la que estima más correcta, consideremos por ejemplo subir una pared, a la hora de llevar dicha acción a cabo dependerá de las características de agilidad de la persona y de la superficie de la pared. Esta aleatoriedad se implementará en forma de tiradas de dados.

2 Objetivos

El objetivo del proyecto es poder comparar los distintos comportamientos de los agentes frente a los estados del mundo o situaciones. Cada agente tendrá asociado una cierta

inteligencia y por tanto, actuará de un modo distinto, es decir, su toma de decisiones se realizará de manera diferente a otro agente que tenga una inteligencia distinta. Poder comparar los comportamientos nos servirá para comprobar qué inteligencia actúa mejor y cuál peor en la toma de decisiones.

3 Inteligencia

Hasta ahora hemos hablado de inteligencia de agentes no humanos. Hay dos tipos de jugadores en un juego de rol:

- Jugadores PJ (Personaje Jugador), controlado por el usuario.
- Jugador PNJ (Personaje no Jugador), controlado por el ordenador.

Los PJ's no tendrán asociada ninguna inteligencia, ya que serán los personajes que controle el propio usuario. Por el contrario, los PNJ's serán los que tengan asociada una determinada inteligencia que les permita interactuar con el mundo y con el resto de personajes.

El proyecto se basará en el comportamiento de los PNJ's.

Ya hemos mencionado que el juego constituye un sistema multiagente porque cada jugador o personaje es un agente, es decir, es capaz de percibir y actuar, es temporalmente continuo, es autónomo y tiene objetivos propios. Los objetivos de los agentes PJ's vendrán determinados por el usuario que los maneje, mientras que para los PNJ's vendrán determinados por su inteligencia y características. La toma de decisiones se basará en el cumplimiento de un determinado objetivo.

De esta forma, podemos decir que la inteligencia de los PNJ's sirve para obtener un determinado objetivo a través de una serie de decisiones.

Vamos a tener de forma general dos tipos de inteligencia dando lugar a agentes computacionales reactivos y deliberativos.

Los agentes reactivos se basarán en un método conductista de estímulo-respuesta implementado a través de un sistema de producción que establezca las reglas de actuación.

En función de que se verifiquen unas determinadas condiciones (ya sean del mundo o del estado de los personajes) se realizará una determinada acción.

Los agentes deliberativos se basarán en búsquedas para encontrar la mejor acción a aplicar. A diferencia de los anteriores, estos realizarán un análisis de la situación en la que se encuentren y verán las distintas alternativas para elegir la más apropiada.

En conclusión, los PNJ' ya sean reactivos o deliberativos tendrán que tomar una decisión y dicha decisión estará basada en un sistema de producción o en búsquedas y heurísticas.

4 Características del juego

En este apartado vamos a ver qué reglas de rol vamos a aplicar, es decir, cómo se juega al rol, independientemente de la inteligencia o implementación del proyecto. Las reglas estarán basadas en un juego de rol verdadero, RuneQuest.

Cuando nos referimos al mundo, hablamos de un conjunto de objetos, ya sean animados (personajes), como inanimados o estáticos, por ejemplo, terreno, muros, edificios, árboles, etc... Un estado del mundo es una situación determinada del mismo en que los objetos presentan unas determinadas características.

El juego se realizará por turnos. Cada personaje tendrá un turno para realizar la acción que desee.

Cada personaje tendrá una serie de campos. Dichos campos son extensibles a todos los objetos aunque no sean personajes. Para los objetos algunos campos tendrán valor nulo.

Los campos son los siguientes:

Características:

Nombre (NOM): Nombre del personaje. Proporcionado por el usuario si es PJ.

Especie (ESP): Especie a la que pertenece el personaje (Ej. humano, elfo, enano, hobbit, etc...). Seleccionado por el usuario en caso de que sea PJ.

Género (GEN): Género al que pertenece el personaje (masculino, femenino). Seleccionado por el usuario en caso de que sea PJ.

Fuerza (FUE): Fuerza del personaje. Influirá en el daño que puede causar, y en el tipo de armadura, armas, u otros objetos que pueda llevar.
Podrá aumentarse como máximo hasta CON+TAM.
Valor inicial: El obtenido en 3D6 (3 tiradas de dados de 6 caras).

Constitución (CON): Salud del personaje. Influirá en la cantidad de daño que puede soportar. Podrá aumentarse como máximo hasta FUE+TAM.
Valor inicial: El obtenido en 3D6.

Tamaño (TAM): Tamaño del personaje. Influirá para infligir y soportar daño. No podrá modificarse.
Valor inicial: 2D6 + 6 (2 tiradas de dados de 6 caras + 6 puntos adicionales).

Inteligencia (INT): Inteligencia del personaje. Influirá en el número de conjuros a memorizar. No podrá modificarse. No tiene nada que ver con la inteligencia que hemos mencionado antes, ya que ahora sólo hablaremos de cómo se juega.
Valor inicial: $2D6 + 6$.

Destreza (DES): Destreza del personaje. Determinará la rapidez de las acciones del personaje. Podrá aumentarse como máximo hasta $DES \times 1.5$.
Valor inicial: $3D6$.

Puntos Mágicos (PM): Puntos mágicos del personaje. Determinará el nivel de poder mágico para lanzar hechizos. Cada vez que se lance un hechizo se irá decrementando su valor. Cuando llegue a 0 ya no se podrá lanzar ningún hechizo a no ser que se recupere su valor en determinadas fuentes de energía mágica que consumirán varios turnos. Inicialmente, su valor estará a $3D6$ y éste será su máximo.

Puntos Golpe (PG): Puntos de golpe del personaje. Indicará cuánto daño es capaz de soportar. Será la media alta de CON y TAM. Cuando esté a 0 el personaje morirá. Los objetos inanimados también tendrán puntos de golpe. Dichos puntos serán necesarios para destruir el objeto.

Energía (ENE): Energía del personaje. Determinará el nivel de energía del personaje. Conforme transcurra el juego, el personaje irá perdiendo dicha energía y tendrá que recuperarla mediante descanso, comida, etc...
Su valor inicial será la suma de FUE y CON.
Cada dos turnos el personaje perderá un punto de energía. En caso de que realice un esfuerzo mayor, perderá más. Cuando su energía llegue a 0 empezará a perder un punto de golpe cada turno y en todas sus tiradas.

Modificadores:

Puntos de daño (PD): Puntos de daño del arma del personaje. Cada arma tendrá asociado un valor de daño. En caso de que el personaje no tenga ninguna arma se considerarán unos puntos básicos de arma natural ($1D6$).

Puntos de armadura (PA): Puntos de armadura del personaje. Cada armadura tendrá unos puntos de protección. Cuando el jugador no tenga ninguna armadura, se considerarán unos puntos básicos de armadura natural (3 puntos).
El daño que reciba un personaje será la diferencia entre sus puntos de armadura y los puntos de daño + modificador de daño del atacante.
El daño que reciba el personaje se le restará a sus puntos de golpe.
Los objetos tendrán también puntos de armadura. Cuando se sobrepasen dichos puntos se verán afectados sus puntos de golpe.

Modificador de daño (MD): Modificador de daño del personaje. Indicará cuánto daño ha causado el personaje. Su valor podrá obtenerse con los dados a través de la tabla correspondiente sumando FUE + TAM. Al realizar un ataque se le sumará al daño

producido por el arma este modificador. Se añadiría sólo la mitad si el arma fuese arrojadiza.

Modificador de habilidad (MH): Modificador de habilidad del personaje. Se calculará en función de las características que posea el personaje. Cada personaje tendrá asociadas unas determinadas habilidades. Por ejemplo, buscar, caminar con sigilo, trepar, etc...

Posición (POS): Posición del personaje en coordenadas (x,y) del mundo. Cada personaje verá una parte del mundo limitada por su radio de visión.
Valor inicial: Posición aleatoria en el mundo.

Equipo (EQU): Lista de objetos limitada que podrá llevar el personaje. Cada índice de la lista se asociará con una parte del cuerpo del personaje:

Ej.

- | | |
|-------------------|--|
| 1- mano izquierda | - objeto: 1 escudo de algún tipo |
| 2- mano derecha | - objeto: 1 espada, hacha, lanza, etc... |
| 3- tronco | - objeto: malla protectora de algún tipo |
| 4- bolsillos | - objeto: monedas de oro, llaves, pergaminos, pociones, etc... |

Los objetos contenedores tales como los cofres, también tendrán un equipo constituido por los elementos que tengan en su interior. Por ejemplo, en los cofres puede haber monedas de oro o pergaminos.

Para realizar con éxito una habilidad, cada personaje tendrá que tirar 1D100 (1 dado de 100 caras). Si el valor resultante fuese menor o igual que el % asociado a la habilidad, entonces el personaje conseguirá realizar con éxito dicha habilidad.

Cuando se enfrenten dos personajes con habilidades, el personaje atacante tendrá que obtener un valor menor o igual a la diferencia entre el % de la habilidad del personaje defensor y el del atacante para tener éxito, en caso de que el defensor haya tenido éxito previamente. Si el defensor no tuviera éxito, el atacante sólo tendría que obtener una tirada de éxito simple sin tener que realizar ningún cálculo de diferencia.

Los personajes podrán aumentar el % de sus habilidades. Para ello será necesario un entrenamiento. El entrenamiento consistirá en realizar alguna misión que proponga algún PNJ que hará de maestro. Cuando se consiga dicha misión, se incrementará la habilidad correspondiente en 2 percentiles. El entrenamiento supondrá un coste en monedas de oro. A medida que se mejore la habilidad el coste se incrementará.

Las características de los personajes podrán aumentarse mediante estudio. El estudio consistirá en perder varios turnos en centros especializados que incrementen 1 percentil a la característica correspondiente. El estudio supondrá una pérdida considerable de energía.

Otro aspecto a tener en cuenta es el de enfrentar características entre sí. A este aspecto se le llama resistencia. Ejemplos de ello es la potencia de un veneno contra la constitución de un personaje, los enfrentamiento de puntos mágicos, forzar puertas o muros, levantar objetos... Para obtener su valor será necesario utilizar la tabla de resistencia, tirar 1D100 y obtener en la tirada un valor menor o igual al de la tabla.

Las caídas producirán 1D6 de daño.

La asfixia producirá 1D8 de daño.

El fuego producirá de 1D6 a 13D6 de daño (según sea una llama (valor mínimo) o un pozo de lava (valor máximo)).

Los venenos emplearán toda su potencia si el personaje no sale exitoso de su tirada de resistencia, en otro caso su efecto será atenuado a la mitad.

Cada veneno tendrá un antídoto.

Las armas y armaduras se comprarán en tiendas.

Las monedas de oro se obtendrán robando o buscando en cofres.

Las pociones se obtendrán buscando por el mundo.

Los pergaminos de magia se obtendrán en tiendas o buscando por el mundo.

Cada jugador podrá llevar un sólo escudo y una sola arma de ataque. Cuando coja otra nueva se sustituirá por la que tenía anteriormente.

Lo mismo pasará cuando se supera el máximo número de pergaminos a memorizar limitado por el valor de INT.

Acciones de los personajes:

- Atacar
- Realizar un hechizo
- Moverse a una posición no obstaculizada
- Coger algún objeto
- Usar algún objeto
- Abrir/Cerrar puertas
- Acciones relacionadas con las habilidades (buscar, trepar, nadar, etc...)
- Comprar algún objeto
- Recuperar energía
- Recuperar poder mágico
- Estudiar/Entrenar

5 Estructura de diseño

Como ya hemos dicho, el mundo estará formado por una serie de elementos, tanto estáticos como dinámicos. Cada elemento tendrá una serie de propiedades.

De este modo, podríamos tener el mundo formado por listas. Cada lista se asociaría a cada tipo de elemento. Por ejemplo, tendríamos una lista de personajes, donde cada elemento sería un personaje concreto; otra lista para los terrenos o tipos de suelo, para los edificios, y demás objetos.

El mundo estaría cargado en memoria en forma de listas. Ésta sería la estructura de datos del mundo.

Cuando un elemento del mundo se modifique, habrá que buscarlo en la lista correspondiente y modificar sus propiedades. En el caso de que aparezca o desaparezca un elemento del mundo habrá que insertarlo/eliminarlo de la lista.

Para todo ello, será necesario que las listas sean óptimas en las búsquedas y en las operaciones de inserción/eliminación.

Cuando los jugadores realicen sus acciones, modificarán el mundo, por lo que habrá que modificar la estructura de datos y visualizarla nuevamente con los cambios.

El mundo vendrá limitado por una serie de casillas. El número de casillas que tendremos será un valor máximo determinado. Cuando se llegue a dicho valor se volverá a la casilla inicial, produciendo, de este modo, un mundo redondo. Para ello, las operaciones de casillas se realizarán módulo el máximo valor.

El juego se realizará por turnos. Es decir, hasta que un personaje no realice una acción correcta no se pasará el turno a otro jugador.

Como pueden participar varios PJ's (usuarios) en el juego, la implementación estará basada en una arquitectura cliente/servidor.

Cada cliente sería un PJ.

El servidor se encargaría de controlar el mundo incluyendo los PNJ's.

De este modo, el funcionamiento seguiría el siguiente esquema:

- El cliente solicita al servidor realizar una acción determinada en el mundo.
- El servidor comprueba con la estructura de datos del mundo (listas) si esa acción es posible de realizar.
En caso afirmativo, realiza los cambios pertinentes y pasa turno.
En otro caso, informa del fallo y espera que el cliente le solicite otra acción
- Una vez pasado el turno del cliente, el servidor realiza las acciones de los PNJ's (turno del servidor) y después cede el turno al siguiente cliente.

El orden de los turnos vendrá determinado por el orden de conexión de los clientes al servidor.

El servidor será el encargado de llamar a la inteligencia artificial de cada PNJ para realizar una acción determinada.

En conclusión, la estructura de diseño del proyecto será un conjunto de listas para almacenar todos los elementos del mundo. Cada ítem de la lista será una estructura con varios campos correspondientes a las distintas características del elemento.

La arquitectura del proyecto o armazón estará formado por un cliente/servidor, donde cada cliente será un jugador usuario y el servidor será el encargado de controlar el mundo.

6 Herramientas de programación

La arquitectura cliente/servidor será programada utilizando C/C++. Éste será el lenguaje utilizado principalmente.

Para los gráficos, ya que el juego es una aplicación visual, y los resultados de las acciones se verán gráficamente, se utilizará la librería gráfica OpenGL. En cuanto a la inteligencia artificial, para que ésta presente un carácter dinámico, es decir, que pueda modificarse independientemente del resto del programa, se utilizará un lenguaje interpretado Python. De este modo, si queremos cambiar de tipo de inteligencia, no será necesario recompilar todo el código, ya que podremos hacerlo sobre la marcha mediante el intérprete. Este aspecto hará que el proyecto presente un carácter dinámico y abierto a cualquier ampliación.

El esquema principal del proyecto es el siguiente:

