

2016

Arcane Quest

Juego de rol



Arcane Quest

Índice

Introducción	4
Jugador	5
Variables de Jugador	5
Métodos de Jugador	5
Posicion.....	5
Variables de Posicion	5
Movimiento	5
Variables de Movimiento	5
Mapa.....	6
Variables de Mapa	6
Métodos de Mapa	6
Partida	6
Variables de Partida.....	6
Métodos de Partida	6
Nivel.....	6
Variables de Nivel	6
Métodos de Nivel	6
PersistenciaNivel	7
Variables de PersistenciaNivel.....	7
Turno	7
Variables de Turno.....	7
Métodos de Turno	7
Elemento	8
Variables de elemento.....	8
Animado	8

Variables de animado	8
Métodos de animado	8
Héroe	8
Variables de Héroe	8
Métodos de héroe	8
Interface Mágico.....	9
Métodos de la interfaz mágico	9
Enemigo	9
Inanimado.....	9
Muro	9
Suelo	10
Escalera.....	10
Cofre	10
Trampa.....	10
Intrerface ObjetoEspecial	11
Método de ObjetoEspecial	11
Poción	11
Variables de poción	11
Hechizo	11
Variables de hechizo.....	11
Fuego	12
BolaDeFuego.....	12
Incinerar.....	12
Aire	12
Tierra	12
Meteoro.....	12
Tornado de agua.....	12
AuraCurativa.....	12
Variables de AuraCurativa	12
Inteface HechizoCuracion.....	13
Variables de hechizo.....	13

Introducción

Arcane Quest es un juego de rol por turnos que utiliza una mecánica y una estética muy similar a la del clásico juego de mesa Hero Quest. Es decir, tendremos que adentrarnos en mazmorras llenas de peligros con nuestro grupo de aventureros, para matar a los monstruos y saquear el tesoro.

Todas las acciones del juego se llevan a cabo por turnos y utilizando dados. Es decir, para mover a nuestro personaje tendremos que tirar los dados, para atacar tendremos que tirar los dados, y para hacer prácticamente cualquier acción tendremos que tirar los dados.

Ahora bien, estas tiradas de dados se pueden ver alteradas gracias a los diferentes atributos que tendrán nuestros héroes, o los distintos objetos que podemos utilizar para ayudarles: anillos de protección, escudos especiales, espadas, etcétera.

En Arcane Quest encontraremos varias aventuras diferentes, cada una con su nivel de dificultad correspondiente. Así, encontraremos alguna ideal para principiantes, que podremos superar con tan solo dos héroes; pero también tendremos misiones solo aptas para aventureros experimentados.

Este juego está inspirado en la versión gratuita del mismo.

<https://play.google.com/store/apps/details?id=com.nex.xquestlite&hl=es>

Jugador

Un jugador es aquel que elige los héroes con los cuales se iniciará una partida. También podrá comprar, vender, dar equipamiento a los héroes que haya seleccionado.

Variables de Jugador

- **heroes: Heroe[]** : Son los personajes que ha escogido el jugador.
- **equipamiento: Equipamiento[]** : Son los objetos que se mantienen entre partida y partida, y se pueden comprar en la tienda. Por defecto será Punho.
- **oro: Oro** : Moneda de cambio que se utiliza en la tienda.

Métodos de Jugador

- **darEquipo(heroe: Heroe, equipamiento: Equipamiento)**: Consiste en asignar un objeto de tipo Equipamiento a un personaje de tipo Heroe.
- **darHechizo(heroe: Heroe, hechizo: Hechizo)**: Consiste en dar un hechizo a un héroe mágico.
- **comprar(tienda: Tienda)**: Consiste en comprar un objeto de tipo Equipamiento en la tienda.
- **vender(equipamiento: Equipamiento)**: Consiste en vender un objeto de tipo Equipamiento en la tienda. (Al venderse un objeto, este desaparece del array y se incrementa el oro).
- **elegirHeroe(héroe: Heroe)**: Añade al array de héroes el personaje pasado por argumentos.

Posicion

Clase que determina la posición de un elemento en el mapa a través de dos coordenadas.

Variables de Posicion

- **x: int** : Determina la posición de un elemento en el eje X.
- **y: int** : Determina la posición de un elemento en el eje Y.

Movimiento

Clase que indica los movimientos de los Elementos Animados del mapa.

Variables de Movimiento

- **posicionInicial: Posicion** : Indica la posición inicial del personaje antes de moverse.
- **posicionDestino: Posicion** : Indica la posición final del personaje tras realizar su movimiento.

Mapa

Clase que representa el mapa y que contiene todos los elementos de tipo Elemento que se encuentra en él.

Variables de Mapa

- **elementos: Elemento[]** : Son todos los elementos que hay en el mapa incluyendo animados, e inanimados, básicamente es una colección de todos los elementos que tiene el mapa.

Métodos de Mapa

- **eliminarElemento(elemento: Elemento)**: Cambia a nulo el elemento del mapa que se pasa por argumentos.
- **anadirElemento(elemento: Elemento)**: Coloca el elemento en el mapa que se pasa por parámetros.

Partida

Conjunto de niveles que juega un jugador.

Variables de Partida

- **jugador: Jugador** : Es el jugador que juega la partida actual.
- **movimientos: Movimiento[]** : Son los movimientos realizados hasta ahora en la partida.
- **nombre: String** : Es el nombre de la partida.

Métodos de Partida

- **cargarNivel(persistenciaNivel: PersistenciaNivel, numNivel: int): Nivel**: Se pasa por argumentos los niveles en almacenamiento y su número para cargarlo, y lo cambia por el actual.

Nivel

Nivel de partida que puede tener uno o varios mapas y en el que solo se permite comprar antes de empezarlo.

Variables de Nivel

- **contadorFinNivel: int** : Es un número que se incrementa cada vez que un héroe usa las escaleras para así saber cuando han pasado todos.
- **mapas: Mapa[]** : Son los mapas en los que se juega cada nivel.
- **tienda: Tienda**: Es la tienda en la que se puede comprar entre niveles. Solo se puede comprar antes de empezar un nivel.

Métodos de Nivel

- **resetearHechizos(héroes: Heroe[])**: Cambia el estado de uso de los hechizos de los personajes mágicos del array de héroes de usado a no usado antes de empezar el nivel.

- **limpiarInventario(héroes: Heroe[]):** Vacía el inventario del array de héroes que se pasan por argumentos cada vez que se cambia de mapa.
- **cambiarMapa: Mapa:** Cambia el mapa del nivel en el que se está jugando cuando todos los héroes hayan pasado por las escaleras por el mapa siguiente. El mapa ya tendrá todos sus elementos colocados.
- **mover(personaje: Animado, movimiento: Movimiento):** Cambia la posición del héroe que se pase por argumentos a la posición especificada en la misma. Si la posición del personaje es una escalera, entonces el héroe llama al método usarEscaleras() y se incrementa el valor de contadorFinNivel. Si contadorFinNivel es igual al número de héroes vivos en la partida, entonces se llama al método cambiarMapa().
- **resetearComprado():** Tienda: Cambia el estado de los objetos de la tienda de comprado a no comprado.

PersistenciaNivel

Mantiene guardados los niveles que se jugarán en cada partida.

Variables de PersistenciaNivel

- **niveles: Nivel[] :** Son todos los niveles guardados en el juego.

Turno

Clase que controla el cambio de turno entre los Animados.

Variables de Turno

- **animados: Animado[] :** Son los personajes actuales que se encuentran en el mapa actual.
- **numTurno: int:** Variable que se va incrementando cada vez que se acaba el turno de un personaje y cambia al siguiente para así mantener un orden concreto.
- **cambioTurno: boolean:** Booleano que sirve para saber si es el turno de los enemigos o de los héroes.

Métodos de Turno

- **cambiarTurno():** Método que se encarga de cambiar el turno. Cuando un personaje hace todos sus movimientos que pueda realizar o lance un hechizo o ataque, el boolean se cambiará y le dará el turno al siguiente personaje.

Elemento

Un elemento puede ser animado o inanimado y conoce su posición en el mapa.

Variables de elemento

- **posicion : Posicion** : En esta variable se almacenara la posición de dicho objeto animado o inanimado.

Animado

Un héroe o un enemigo son personajes animados, que tienen los siguientes variables y métodos.

Variables de animado

- **danho: int** : Contiene los dados de daño de dicho personaje animado.
- **defensa: int** : Contiene los dados de defensa de dicho personaje animado.
- **vida: int** : Contiene los puntos de vida de dicho personaje animado.

Métodos de animado

- **movimientosPermitidos(mapa:Mapa): Movimiento[]** : En este método, le pasas una variable mapa, y te devuelve los movimientos permitidos de dicho animado.
- **atacar(atacado: Animado)**: Este método se utiliza para atacar a otro animado, como por ejemplo un héroe a un animado o viceversa.

Heroe

Los héroes son los personajes que va a llevar el que utiliza la aplicación (**En el diagrama de clases están incluido los datos de cada hijo**).

Variables de Heroe

- **datosMovimiento: int** : En esta variable se almacenaran los datos de movimiento de cada héroe.
- **arma:Arma** : En esta variable se almacenara el arma que lleva equipada dicho héroe, por defecto se creara con los puños, si tiene equipada otra arma y se le quita, volvería a ponerse los puños, digamos que es el arma default, la que tiene al no llevar nada.
- **escudo: Escudo** : En esta variable se almacenara el escudo de dicho héroe.
- **inventario:ObjetoEspecial[]**: En este array se almacenaran los objetos especiales que lleve equipado dicho héroe (Pociones).

Métodos de Heroe

- **lanzarDadosMovimiento() : int**: En este método se lanzaran los dados de movimiento, según los dados que tenga dicho héroe, saldrá más o menos.
- **desactivarTrampa(trampa:Trampa)**: En este método, un héroe podrá desactivar una trampa por turno, la trampa tendrá una determinada vida, si se tira los dados de ataque y si es igual o mayor

que ese número, la trampa se desactiva, si el héroe saca menos de esa cantidad, la trampa se desactiva pero le quita el daño que tenía la trampa al héroe que la intento desactivar, y se convierte en suelo en ambos casos.

- **abrirCofre(cofre:Cobre):** Con este método, el héroe podrá abrir un cofre, en el cual podrá dar oro o ser una trampa (esta trampa no se puede desactivar, ya que el héroe no se la espera).
- **usarEscaleras():** Con este método, el héroe podrá ir al siguiente nivel, este método se ejecuta cada vez que se mueve, comprobando lo que este pisando, si cuando se mueve, lo que hay abajo es una escalera, este personaje desaparece del mapa, pero no del juego, una vez se metan todos los personaje, se pasa al siguiente mapa.
- **consumirObjeto(objetoEspecial:ObjetoEspecial):** Con este método, podremos consumir un objeto especial (una poción).

Interface Mágico

Esta interface mágico los implementaran los héroes que puedan tener magias.

Métodos de la interfaz mágico

- **lanzarHechizo(hechizo:Hechizo, animado:Animado):** Lanza un hechizo a un objeto animado, suele como por ejemplo, el mago tira un Tornado de agua sobre un personaje enemigo, si intenta lanzar un hechizo dañino sobre un aliado, no hace nada, pero si el hechizo implementa la interface HechizoCuracion, este curará.
- **getHechizos(): Hechizo[]:** Devuelve los hechizos que lleva equipado actualmente dicho héroe mágico.
- **getNumHechizos(): int:** Este método devuelve el número de hechizos que tiene dicho héroe.

Enemigo

Clase en la que contendrá todos los enemigos del juego (**En el diagrama de clases están incluido los datos de cada hijo**) CaballeroLobo, Goblin, Ogro.

Inanimado

Un elemento inanimado, como el suelo, la escalera, un cofre, muro, suelo o trampa, todos los anteriores mencionados deben tener una posición para ser visualizados.

Muro

El muro es un elemento es cual delimita las salas y haciendo que los personajes no puedan traspasarlas.

Suelo

Es el terreno básico sobre el que se mueven los personajes animados del suelo y sobre el que puede haber otros elementos.

Escalera

La escalera es un elemento que marca la salida del mapa, es decir que el objetivo de todos los héroes es llegar hasta este. Cada vez que un héroe pase por la escalera se determina que el héroe ha salido del mapa.

Cofre

Es un objeto que se ha creado con la intención de que pueda contener otros objetos, dichos objetos deben implementar la interfaz "ContenidoCofre".

Variables de Cofre

- **contenido:ContenidoCofre:** En esta variable se guardará el contenido del cofre.

Trampa

Las trampas son objetos creados con la idea de hacer daño. El funcionamiento es el siguiente, todos los elementos tienen una posición, se determina que una trampa se activa cuando un "Heroe" tiene la misma posición que "Trampa". Recaltar que aunque "Heroe" tiene un método para desactivarla si se lo encuentra dentro de un cofre no tendrá otra opción que sufrir el daño.

Variables de Trampa

- **activada:boolean:** Este atributo es para determinar el estado de la trampa, es decir si esta activada o no. Por defecto este atributo tendrá un valor *true*, que quiere decir que esta activada. Cuando la trampa se utilice este atributo pasará ser *false*.
- **danho:int:** Determina el daño que recibe el personaje sobre el que se activa, en otras palabras otras palabras la cantidad de vida que le quita al personaje pasamos por argumentos.

Variables de Trampa

- **hacerDanho(heroe:Heroe):void:** Con este método modificamos la vida del "Heroe" pasado por argumentos. La cantidad de vida restada es igual al daño de "Trampa".

Interface ContenidoCofre

Se implementa esta interfaz a todos los objetos que se quieran meter en los cofres. Esta interfaz la implementan las clases "Oro", "Trampa" y lo hereda "ObjetoEspecial".

Interface Transitable

Con esta interfaz queremos controlar que un "Animado" pueda pasar por encima, o en otras palabras que la posición del "Animado" pueda tener la misma posición que el inanimado.

Interface ObjetoEspecial

Esta interfaz va a implementarla cualquier objeto especial, en nuestro caso poción.

Método de ObjetoEspecial

- **utilizar()** : Este método se implementa en poción, y está hecha para que un héroe pueda utilizar una poción.

Poción

Todas las pociones del juego son hijas de esta clase. (En el diagrama de clases están incluidos los datos de cada hijo).

Variables de poción

- **cantCuracion: int** : Esta variable indica cuánto cura una poción.

Hechizo

Magia que pueden ejecutar los personajes mágicos. (En el diagrama de clases están incluidos los datos de cada hijo).

Variables de hechizo

- **usado: boolean**: el cual nos indica si el personaje ha usado el hechizo en ese nivel o no, con lo consiguiente si el booleano esta con el valor "true" quiere decir que el personaje ya lo ha utilizado, si por el contrario el valor esta a "false" significa que el personaje no lo ha usado y lo puede usar en cualquier momento.
- **nombre: String**: con el que conseguimos que nos proporcione el nombre del hechizo.
- **descripcion:String** con el que nos dará la descripción del hechizo que hayamos elegido con su correspondiente valores como daño, los valores que le hace falta al enemigo para defenderse del ataque.
- **danho:int**: con el cual conseguiremos que los hechizos nos den el valor de su ataque.

Fuego

Tipo de hechizo que usa fuego.

BolaDeFuego

Hechizo que consiste en lanzar una potente bola de fuego haciendo un daño al enemigo.

Incinerar

Hechizo que consiste en envolver al enemigo en fuego haciéndole un daño al enemigo.

Aire

Tipo de hechizo que usa aire.

Tierra

Tipo de hechizo que usa tierra.

Meteoro

Hechizo que consiste en lanzar un potente meteorito el cual hace un daño al enemigo.

Tornado de agua

Consiste en lanzar un poderoso tornado de agua hacia el enemigo.

AuraCurativa

Es un tipo de hechizo que al lanzarse a otro personaje este será curado y si es lanzado a un enemigo este sufrirá un tipo de daño.

Variables de AuraCurativa

- **curacion:(int)** Cantidad de curación que se realizará. (Su valor está en el diagrama).

Interface HechizoCuracion

Esta interfaz va a implementarla cualquier hechizo que pueda curar

Variables de hechizo

- **getCuracion:int:** el cual se quedara como atributo en la clase y cuando la clase ejecute el método **lanzaHechizo** sobre un “Héroe” le hará un incremento de la vida igual al atributo **curacion**. El objeto **AuraCurativa** implementa esta interfaz.