

MEMORIA PRACTICA FINAL POO

INDICE:

1. Comentando el código
2. Conclusión final

Conceptos previos:

Mi trabajo en esta práctica era programar el modo duo del battle royale, por lo que he hecho que cuando se pulse el botón de duo en el menú de juego se llame al constructor de mi clase DUO y desde este se vayan llamando a los distintos métodos.

Método SimulacionDeJuego():

Este método en general sirve para simular la partida entera. Empieza generando un número aleatorio entre 2 y el tamaño del ArrayList de personajes. Designamos que si la posición aleatoria en el arraylist de personajes no es nula, y es número par se añaden a otro array list de duo de contrincantes la posición aleatoria del arraylist de personajes y la posición aleatoria + 1 (esto es para crear los duos de contrincantes ya que si el número aleatorio es el 2, por ejemplo, su pareja será el tres). Después se establecen las variables con las posiciones de los contrincantes en el arraylist de personajes y se llama al método Decision() pasándole las variables de las posiciones. (En el caso de que el número aleatorio sea impar su pareja será su posición - 1). Si la posición fuese aleatoria en el arraylist de personajes es nula (ha muerto ese personaje) se vuelve a llamar a SimulacionDeJuego() hasta que se genere una posición no nula.

Método Decision():

Mientras que alguno de los duos se mantenga con vida se le pregunta a los jugadores humanos que desan hacer (excepto si alguno de los jugadores ha muerto que no se le pregunta nada). Para saber que quieren hacer se llama al método OpcionesJugadores(). Después de que decidan los jugadores se llama al método DecisionContrincantes() para que decidan los contrincantes. Finalmente si alguno de los duos muere se avisa al usuario mediante un JOptionPane.showMessageDialog().

Método Atacar():

Simplemente llama al método atacar de cada personaje.

Método DecisionContrincantes():

Si los contrincantes están muertos se retorna, de lo contrario, se simula de manera aleatoria como actuan los bots contrincantes (atacan o usan su habilidad y a quién atacan). También con este método me di cuenta de que debería haber puesto las posiciones de los contrincantes como pos1 la de menor valor, pos2 la de mayor valor.

Método Matar():

Simplemente se establece la posición que le pasemos, que será el jugador que ha muerto, como nula.

Método OpcionesJugadores():

Si la decisión del jugador es que quiere atacar se le ofrece la opción de atacar a los jugadores del duo contrincante que estén vivos y el escoge a quien de los dos ataca. Si alguno de los jugadores del duo contrincante está muerto no se le ofrecerá la opción de atacarle. Si escoge usar la habilidad se llama al método del personaje de habilidad.

Método SimulacionCombateBots():

Se rellena un arraylist con los jugadores bots que hay disponibles y se borran 4 aleatoriamente que serán los que combatirán con el método CombateBots().

Método CombateBots():

Cada bot seleccionado anteriormente decidirá atacar a otro bot de entre los otros dos seleccionados. `rand.nextBoolean()` ? `bot3` : `bot4` mediante este método el bot decide con un booleano aleatorio a quien atacar de entre los otros dos bots. Se imprime con una interfaz quien muere en cada ronda.

Método MostrarPersonajes():

Se muestra que personajes quedan vivos mediante un bucle for.

CONCLUSIONES FINALES:

De todas las practicas que hemos hecho en está asignatura esta es de la que menos orgulloso estoy de como me ha quedado. No he conseguido que funcione correctamente como quería. A demás creo que nos deberíamos haber organizado mejor para saber a donde quiere llegar cada uno. Para mi, la práctica no era nada fácil y presentaba bastante problemas a resolver ya que había que gestionar muchas cosas de la batalla.