



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
ING. NEFTALÍ CALDERON
AUX. JURGEN RAMIREZ

INTRODUCCIÓN A LA COMPUTACIÓN Y PROGRAMACIÓN 1

Manual Técnico

Practica 01. Batalla Pokémon

CARNET	NOMBRE
202000558	Sheila Elizabeth Amaya Rodríguez.

INTRODUCCION

La finalidad de este es la de proporcionar al lector las pautas de configuración y lógica con la que se ha desarrollado.

El programa “Batalla Pokémon” cuenta con una interfaz sencilla la cual consiste en que el usuario agregue o digite por teclado cada una de las instrucciones que desea para que el programa se ejecute de manera correcta, se debe de prestar atención a cada espacio requerido según lo solicitado por el programa.

El proyecto se basa en reproducir la batalla Pokémon entre dos jugadores los cuales lucharán con dos pokémones cada uno, además de ser por turnos esto hasta que ambos Pokémon de alguno de los dos jugadores llegue a su vida cero.

PLATAFORMA DE EJECUCIÓN

NetBeans

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las API de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

El NetBeans IDE permite el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

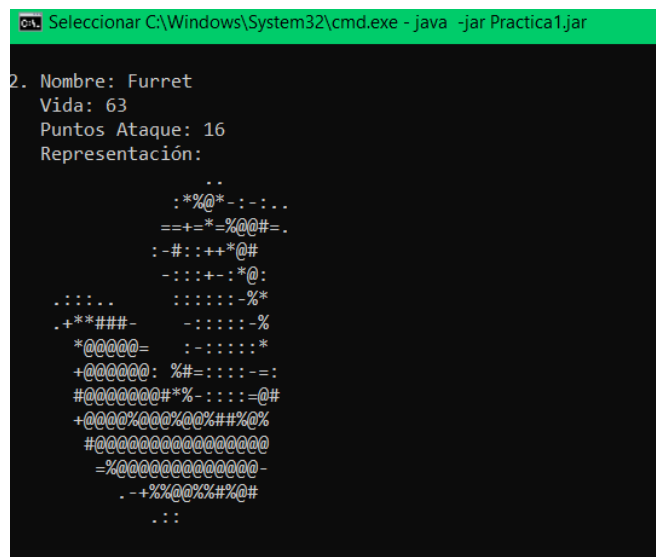
.Jar

Este suele albergar una biblioteca con varios archivos. La extensión .jar es una abreviatura de Java Archive y suele contener, como su nombre sugiere, varios archivos Java y metadatos que se envían de manera sintetizada y comprimida.

Este tipo de bibliotecas suelen utilizarse para simplificar la programación de aplicaciones y de applets de Java. El conjunto de metadatos y archivos .xml o .json, gráficas y archivos de audio puede integrarse como elemento individual en otros proyectos. Uno de los componentes más importantes de estas bibliotecas de Java es el llamado manifiesto, un archivo que contiene detalles acerca de la versión y el autor del código.

Requerimientos

Contar en nuestro sistema operativo java ya que sin esta las aplicaciones no se ejecutarán en Windows.



```
C:\Windows\System32\cmd.exe - java -jar Practica1.jar

2. Nombre: Furret
Vida: 63
Puntos Ataque: 16
Representación:

      .
      :*%@*-:-:..
      ==+*=*%@@#=.
      :-#::+*%@#
      -::+:-*%@:
      .:::..      :::-~%*
      .+*%###-    -:::-~%
      *@@@@@=    :-:::-~%
      +@@@@@@: %#=:::-~%
      #@@@@@@@#*%-:::-~%
      +@@@@@%@@@@%@@%##%@@%
      #@@@@@@@@@@@@@@@@
      =@@@@@@@@@@@@@@@@-
      .-+%@@%#%@@%
      .:::
```

DICCIONARIO MÉTODOS USADOS

CLASE PRACTICA1

Método “menú”: Nos muestra las opciones que pueden ser elegidas dependiendo si el usuario es un administrador o jugador.

Método “SubM_A”: Nos redirige al menú del administrador.

Método “SubM_U”: Nos redirige al menú del usuario normal el cual puede acceder únicamente a iniciar una partida.

Método “elegirPokemon”: Brinda las opciones disponibles.

Método “admi”: Nos brinda diferentes opciones para que el administrador pueda editar pokemones o visualizar el uso del programa últimamente.

Método “ListaP”: Permite visualizar la lista de Pokémon actual.

Método “addP” Permite que el administrador pueda agregar mas pokemones a la lista y el usuario pueda tener variedad.

Método “editP”: Permite modificar aspectos básicos de los Pokémon.

Método “Report”: Permite visualizar un historial de lo que se ha ejecutado en las últimas partidas.

Método “pDefecto”: Al iniciar el programa esto corre como primera instancia ya que se debe generar la vida random y ataque justo antes de iniciar una partida.

Método “Registros”: Nos permite visualizar un historial de las partidas jugadas.

Método “Listap_menos”: Permite visualizar un historial sobre los Pokémon menos utilizados.

Método “ListaP_mas”: Permite visualizar un historial sobre los Pokémon más utilizados gracias al método de ordenamiento burbuja.

Método “ListaG”: Nos permite recorrer el arreglo para obtener los datos de los ganadores y así visualizar a los usuarios con los que se han ganado las partidas gracias al método de ordenamiento burbuja.

Método “Lista P”: Nos permite recorrer el arreglo para obtener los datos de lo perdedores para visualizar a los usuarios con los que se han perdido las partidas.

CLASE POKEMON

Método “crearPokemon”: Recorre el arreglo y si dicho espacio se encuentra vacío añade al pokémon ingresado previamente.

Método “buscarPokemon”: Recorre el arreglo para determinar si los espacios en el se encuentran ocupados o vacíos.

Método “vidaRandom”: Retorna un número aleatorio entre un número min (incluido) y un maximo (excluido)

Método “ataqueRandom”: Retorna un número aleatorio entre un número min (incluido) y un maximo (excluido).

Método “restarVida”: Solicita un número que se encuentra en el rango de ataque y retorna un booleano para determinar así el estado actual del Pokémon.

Método “revivirPokemon”: Se usa para revivir los pokemones y también restaura la vida al iniciar otra partida.

Método “haSidoSeleccionado”: Define un contador para las veces que se selecciona un pokemon y un contador para atacas recibidos.

CLASE BATALLA

Método “iniciarJuego”: Permite solicitar datos a los usuarios que competirán para así poder dar inicio.

Método “mostrarDatos”: Permite mostrar la lista Pokémon con cada una de las características que poseen estos.

Método “elegirPokemon”: Se ejecuta en cada turno de la batalla

Método “elegirDefensor”: Ayuda a que cada jugador a pueda escoger a que Pokémon atacara.

