Ing. Jeremy Mejía 19846 Ingeniería en Audio, UVG

**Ejercicio No. 5**

**(Septiembre, 2021)**

**ANÁLISIS**

1. **¿Qué debe hacer el programa?**

* Permitir elegir la cantidad de jugadores. (Máx. 3 y Mín. 1)
* Enemigo Jefe acompañado de uno o dos enemigos o jefes regulares. (Cantidad elegida al azar 0-2)
* El juego termina cuando Enemigo Jefe muere o cuando mueren todos los jugadores.
* Los jugadores pueden elegir ser Guerreros, Exploradores o Cazadores. Cada uno de ellos tendrá un acompañante y su habilidad será golpear únicamente. Si el combatiente muere, el acompañante desaparece y cualquier acción que haga el combatiente contra su enemigo, el acompañante también.
* El Cazador tiene como acompañante una mascota y su habilidad es lanzarla.
* Enemigo Jefe mucho más poderoso que un jefe regular. (más vida, más poder de ataque, puede clonar, variar y liberar). Clonar; enemigos y mascotas. Variar; elije las habilidades que tendrán los clones. Liberar; se deshace de los acompañantes clonados.
* Lanzar mascotas si aun no están en batalla y una vez lanzada, podrá volver a lanzarla tres turnos después.

1. **¿Qué clases conformarán su modelo para dar solución a la situación planteada?**

* Controlador
* Vista
* Combatiente
* Guerrero
* Explorador
* Cazador
* Acompañante
* Enemigos
* EnemigoJefeRegular
* EnemigoJefe

1. **¿Qué propiedades y métodos tendrá cada clase (incluyendo Vista y Controlador)? ¿Qué tipo deben tener las propiedades y métodos de cada clase? ¿Cuáles son los modificadores de visibilidad?**

**Clase Combatiente:**

* Propiedades:
  + Private Vidas: Int
  + Private Poderes: String
  + Private Nombre: String
* Métodos:
  + Public Acompanante(): String

**Clase Guerrero:**

* Propiedades:
  + Private Vidas: Int
  + Private Poderes: String
  + Private Nombre: String
* Métodos:
  + Public Golpear(): String
  + Public Herir(): String
  + Public Acompanante(): String

**Clase Explorador:**

* Propiedades:
  + Private Vidas: Int
  + Private Poderes: String
  + Private Nombre: String
* Métodos:
  + Public Golpear(): String
  + Public Herir(): String
  + Public Matar(): String
  + Public Acompanante(): String

**Clase Cazador:**

* Propiedades:
  + Private Vidas: Int
  + Private Poderes: String
  + Private Nombre: String
* Métodos:
  + Public Golpear(): String
  + Public herir(): String
  + PedirDatoFam(): String
  + AsignarPerro(): String
  + BuscarFam(): String

**Clase Controlador:**

* Propiedades:
* Métodos:
  + void Main(String[] args)

1. **¿Cuáles deben ser los modificadores de visibilidad de los miembros en cada clase?**

**Clase Perrera:**

* Propiedades:
  + Private Perro1: String Perro
  + Private Perro2: String Perro
  + …
  + Private PerroN: String Perro
  + Private Familia1: String Familia
  + Private Familia2: String Familia
  + …
  + Private FamiliaN: String Familia
* Métodos:
  + Publico AgregarPerro: String
  + Publico AgregarFamilia:String
  + Publico BuscarPerro: String
  + Publico BuscarFamilia: String
  + Publico AsignarPerro: int

**Clase Perro:**

* Propiedades:
  + Private NombrePerro: String
  + Private Tamaño: int
  + Private Raza: String
  + Private EdadProxima: int
  + Private EstadoSalud: String
  + Private Color: String
* Métodos:
  + Publico Detalles(): String

**Clase Familia:**

* Propiedades:
  + Private Apellido: String
  + Private Integrantes: int
  + Private Ninos: String
  + Private Edad: int
  + Private Mascota: String
* Métodos:
  + Publico DetallesFam(): String

**Clase Vista:**

* Propiedades:
  + Private Scan Scanner
* Métodos:
  + Public Menu() int
  + Public PedirDatoPerro(): String
  + Public PedirDatoFam(): String
  + Public AsignarPerro(): String
  + Public BuscarFam(): String

**Clase Controlador:**

* Propiedades:
* Métodos:
  + Public void Main (String[] args)

1. **¿Qué parámetros serán requeridos por los métodos en sus clases?**

Parámetros dentro de cada clase,

**Clase Perrera:**

* Métodos:
  + AgregarPerro: String
  + AgregarFamilia:String
  + BuscarPerro: String
  + BuscarFamilia: String
  + AsignarPerro: int

**Clase Perro:**

* Métodos:
  + Detalles(): String

**Clase Familia:**

* Métodos:
  + DetallesFam(): String

**Clase Vista:**

* Métodos:
  + Menu(): int
  + PedirDatoPerro(): String
  + PedirDatoFam(): String
  + AsignarPerro(): String
  + BuscarFam(): String

**Clase Controlador:**

* Métodos:
  + void Main(String[] args)

1. **¿Cómo proveerá de valores iniciales a sus objetos? ¿Qué valores iniciales les asignará?**

La clase perrera, tendrá tres familias ingresadas antes de iniciar el programa, estos datos serán dados por el programador, al igual que 17 perros.

La clase perro, no tendrá valores iniciales, los datos serán ingresados por el ususario.

La clase familia, no tendrá valores iniciales, los datos serán ingresados por el ususario.

La clase vista, recibirá los datos del usuario

La clase controlador, interactua en base a la interacción del usuario con la vista.

1. **¿ Dónde necesitará arreglos(propiedades, parámetros, etc.),y para qué?**

Necesito arreglos para agregar las familias y asignar perros a las familias, deben de tener parámetro String y la razón es porque debo de agregar muchos, solo declaro cuantos quiero y solos se iran generando.