

## MANUAL TECNICO

El siguiente manual esta organizado y descrito de acuerdo a todos lo algoritmo que se utilizaron para realizar con éxito el sistema de juego “Super Auto Pets”, el cual se entrega dividido por paquetes y clases las cuales se describirán a continuación para entender el funcionamiento de las mismas.

**PAQUETE MASCOTAS:** en este paquete se encuentran las siguientes clases:

- **Clase Animal**

Inicio

var nombre  
var puntosAtaque  
var puntosVida  
var nivel  
var nivelJuego  
var fusión

Constructor Animal(var nombre, var puntosAtaque, car puntosVida, var nivel, var nivelJuego, var fusión)

Getters y Setters respectivo a cada variable

Fin

- **Clase Tipos, esta clase extiende de la Clase Animal**

Inicio

var figura  
var tipo  
var arregloMascota[54]  
var efecto

Constructor Tipos (var figura, var tipo, var arregloMascota, var efecto)  
Getters y Setters respectivo a cada variable

Métodos

inicializarAnimales(): en este método se encuentran creadas todas las mascotas que se utilizan dentro del desarrollo del juego.

mostrarDatos(Tipos figura): este método muestra los datos de cada mascota como el nombre, figura, puntos de vida, puntos de ataque y el nivel en que se encuentre la mascota actualmente.

Fin

- **Clase BufYNerfeo, esta clase extiende de la clase Tipos**

Inicio

Constructor BufYNerfeo(var nombre, var figura, var puntosAtaque, var puntosVida, var nivel, var nivelJuego, var tipo, var efecto, var fusion)

Métodos

bufearMascotaReptil( Tipos[] reptil): este método otorga un buf en los puntos de vida y ataque de todas aquellas mascotas que sean de tipo Reptil.

bufearMascotaVolador(Tipos[] volador): este método otorga un buf en los puntos de vida y ataque de todas aquellas mascotas que sean de tipo Volador.

bufearMascotaAcuatica( Tipos[] acuático): este método otorga un buf en los puntos de vida y ataque de todas aquellas mascotas que sean de tipo acuático.

bufearMascotaTerrestreMamiferoSolitario( Tipos[] terrestre): este método otorga un buf en los puntos de vida y ataque de todas aquellas mascotas que sean de tipo Terrestre/Mamífero, y nerfeo los puntos de ataque de todas las mascotas que sean de tipo Solitarias.

bufearMascotaDomesticoMamifero( Tipos[] domestico): este método otorga un buf en los puntos de vida y ataque de todas aquellas mascotas que sean de tipo Domestico/Mamífero, y nerfeo los puntos de ataque de todas las mascotas que sean de tipo Solitarias.

bufearMascotaSolitaria( Tipos[] solitario): este método otorga un buf en los puntos de vida y ataque de todas aquellas mascotas que sean de tipo Solitarias.

bufearMascotaDisertica( Tipos[] disertico): este método otorga un buf en los puntos de vida de todas aquellas mascotas que sean de tipo Diserticas.

Fin

- **Clase Comida**

Inicio

var efecto

var nombre

var figura

var nivelJuego

var descripcion

var alimentos [18]

var comidaTienda [2]

Constructor Comida(var nombre, var figura, var nivelJuego, var efecto, var descripción)

Getters y Setters respectivo a cada variable

Métodos

inicializarAlimentos(): en este método se encuentran creados todos los alimentos que se utilizan dentro del desarrollo del juego.

mostrarAlimentos(): este método muestra los datos de cada comida como el nombre, figura, puntos de vida, puntos de ataque, y una descripción de lo que hace cada alimento.

mostrarComidaEnTienda(Comida[] alimentos, var Ronda): este método muestra la comida que aparece en tienda dependiendo del nivel y la ronda.

comprarComidaEnTienda(Comida[] alimento, var contRonda, Tipos[] miMascota): este método de permite al jugador compran la comida que el desee para sus mascotas.

darComidaAMascotas(Comida alimentos, Tipos mascotas, Tipos[] arreglo): este método le permite al usuario seleccionar la mascota que desee darle de comer, dependiendo de que comida compre, esta le otorgara un buf a la mascota que el usuario desee darle dicha comida

darVida2Random(Tipos[] mascotas): este método depende de la comida que el usuario compre, si dicha comida comprada es ENSALADA, entonces esta le dará un buf a dos mascotas aleatorias dentro de nuestro mazo de pelea.

darVida3Random(Tipos[] mascotas): este método depende de la comida que el usuario compre, si dicha comida comprada es SUSHI, entonces esta le dará un buf a tres mascotas aleatorias dentro de nuestro mazo de pelea.

Fin

- **Clase Datos Jugador**

Inicio

var oro

var vida

var contVictorias

var contRonda

var contDerrotas

Getters y Setters respectivo a cada variable

Fin

PAQUETE JUGAR

- **Clase Batalla**

Inicio

var Tipos mascota

var Tipos abeja

Métodos

iniciarPelea( Tipos[] miMascota, Tipos[] mascotaRival, var ronda): este método genera la batalla entre las mascotas compradas por el usuario(mi mazo pelea) contra las mascotas rival, conforme están posicionadas las mascotas se irán peleando una por una, haciendo el daño correspondiente a sus puntos de ataque, luego evaluará cual de los dos equipos de (mi mazo pelea) y (mazo rival) tiene al menos una mascota con vida y se le notificara que a ganada esa ronda de batalla, si ambos equipos mueren todas las mascotas entonces le notificara de una empate en esa ronda. También evaluara si mi equipo (mi mazo pelea) gano, entonces se le suma una victoria, caso contrario mi equipo pierda, entonces se le restara puntos de vida.

restarVida( var ronda): este método resta la vida, dependiendo la ronda en la que se este jugando.

presionarTecla(): este método hace el que usuario presione una tecla para seguir avanzando.

Fin

## **PAQUETE DESARROLLO JUEGO**

- **Clase Menú Juego**

Inicio

var opción

Métodos

mostrarMenuJuego()

1. MODO ARENA

2. MODO VERSUS

3. MODO CREATIVO

leer opción

Según:

una IA

Caso 1: lo lleva a un menú de batalla, en donde se enfrentara contra

Caso 2: lo lleva a un menú de batalla, en donde se enfrentará con x j jugadores.

Caso 3: en este caso el usuario podra elegir entre todas las mascotas sin ninguna restricción.

Finsegún

Fin

- **Clase Menú Batalla**

Inicio

var opción

var contadorMisMascotas

var miMascota [5]

var mascotaRival [5]

Getters y Setters respectivo a cada variable

Métodos

mostrarMenuBatallas():

Hacer

1. COMPRAR MASCOTAS

2. COMPRAR COMIDA

3. ORDENAR MASCOTAS

4. FUSIONAR MASCOTAS

5. VENDER MASCOTAS

6. INICIAR JUEGO

0. SALIR

Según:

Caso 1: mostrarAnimalesEnTienda(getArregloMascota(),  
getContRonda())

Caso 2: comprarComidaEnTienda(getAlimentos(), getContRonda(),  
miMascota)

Caso 3: ordenarMazoBatalla(miMascota)

Caso 4: fusionarMascotas(mascota, miMascota)

Caso 5: venderMascotas(miMascota)

Caso 6: mostrarCamposDeJuego()

FinSegun

Mientras(opcion != 0)

mostrarAnimalesEnTienda( Tipos[] mascotas, var Ronda): este método muestra los animales en tienda según el nivel y la ronda.

guardarMascotaComprada( Tipos[] mascotaComprada, Tipos[] miMascota, var posición) : este método va guardando mascotas en las castillas de nuestro arreglo, conforme vamos comprando mascotas, ingresando el nombre de la mascota que desee comprar.

imprimirMascotasCompradas( Tipos[] mascota): método que le permite ver al usuario las mascotas que lleva compradas.

imprimirMascotasEnTienda( var ronda)

ordenarMazoBatalla( Tipos[] mazo): en este método el usuario puede arreglar sus mascotas según su consideración y su forma lógica de llevarlas posicionadas a la batalla, ingresando el nombre de la mascota que desee que vaya en cada posición.

venderMascotas( Tipos[] mazo5): este método le permite al usuario vender las mascotas que lleve en su mazo, escribiendo el nombre de la mascota que desee vender.

fusionarMascotas( Tipos[] mascotaTienda, Tipos[] miMascota): este método le permite al usuario fusionar mascotas que lleve en su mazo con las de la tienda, dicho método evaluará si en la tienda hay una mascota igual a la que se lleva en el mazo de pelea, de ser así, entonces podrá fusionar ambas mascotas del mismo tipo, subiendo hasta un máximo nivel 3, si no hay mascotas repetidas para fusionar se le mostrará un mensaje.

mostrarMascotasRival( Tipos[] mascotaRival, var Ronda); este método genera la IA, es decir crea rivales aleatorios para ir a las diferentes rondas de batalla.

Fin

- **Clase Campos de Juego**

Inicio

var opción

Métodos

mostrarCamposDeJuego()

Mientras (opción!= -1) hacer

1. CAMPO POR DEFECTO
2. PANTANO
3. NUBES
4. MAR
5. BOSQUE
6. GRANJA
7. SABANA

-1. REGRESAR

Leer opción

Según opción

Caso 1:

```
bufearMascotaSolitaria(miMascota)
bufearMascotaSolitaria(mascotaRival)
iniciarPelea(miMascota, mascotaRival, ronda)
```

Caso 2:

```
bufearMascotaReptil(miMascota)
bufearMascotaReptil(mascotaRival)
iniciarPelea(miMascota, mascotaRival, ronda)
```

Caso 3:

```
bufearMascotaVolador(miMascota)
bufearMascotaVolador(mascotaRival)
iniciarPelea(miMascota, mascotaRival, ronda)
```

Caso 4:

```
bufearMascotaAcuatica(miMascota)
bufearMascotaAcuatica(mascotaRival)
iniciarPelea(miMascota, mascotaRival, ronda)
```

Caso 5:

```
bufearMascotaTerrestreMamiferoSolitario(miMascota)
bufearMascotaTerrestreMamiferoSolitario(mascotaRival)
iniciarPelea(miMascota, mascotaRival, ronda)
```

Caso 6:

```
bufearMascotaDomesticoMamifero(miMascota)
bufearMascotaDomesticoMamifero(mascotaRival)
```

iniciarPelea(miMascota, mascotaRival, ronda)

Caso 7:

bufearMascotaDisertica(miMascota)

bufearMascotaDisertica(mascotaRival)

iniciarPelea(miMascota, mascotaRival, ronda)

FinSegún

FinMientras

Fin