



UNIVERSIDAD DE LOS ANDES
Facultad de Ingeniería y Ciencias Aplicadas
Conversión de Energía 2019-10

Documentación Entrega 4

Integrantes:

Alejandra Izquierdo, Matías Reyes

Pedro Donoso

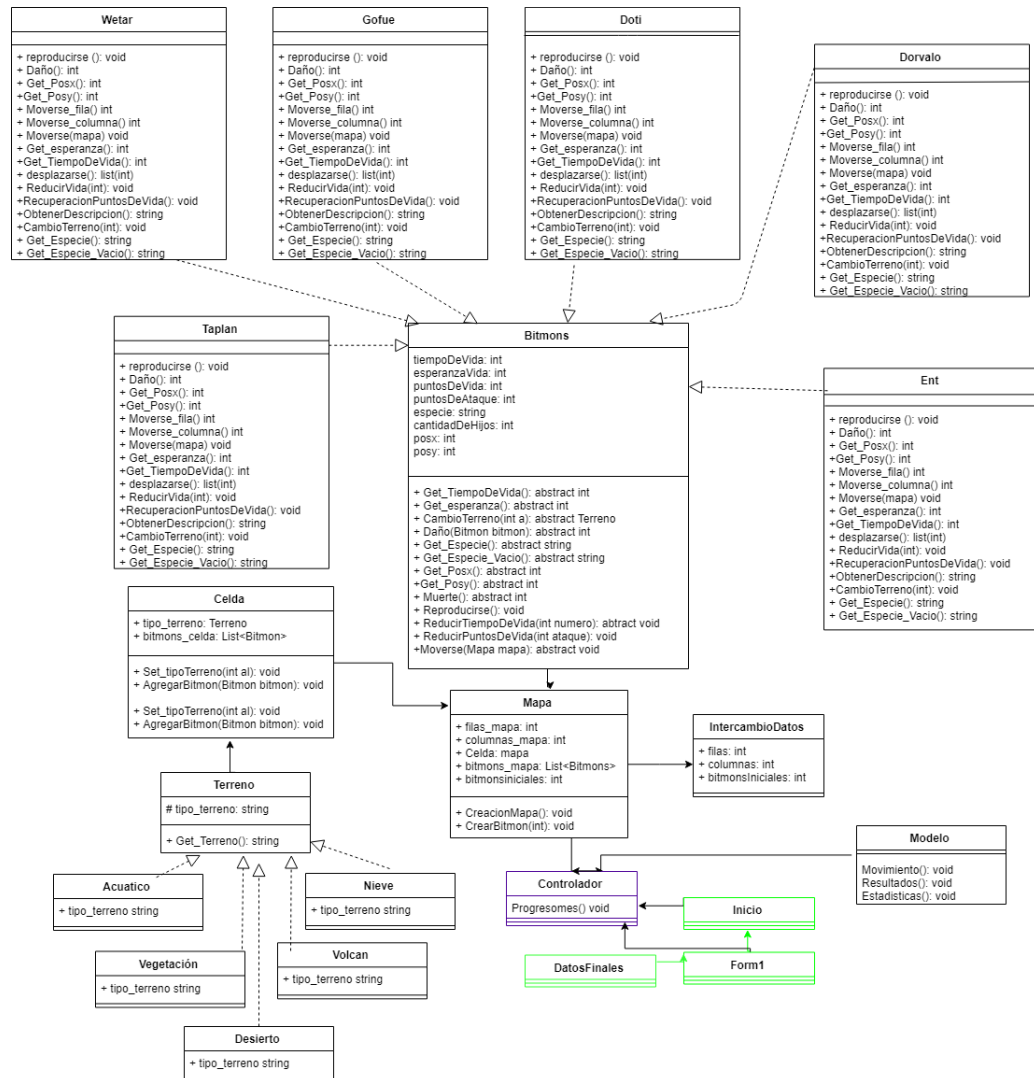
Profesor: Nicolás Gomez

Ayudante: Sebastián Baixas, Benjamín Perez

12/06/2019



1. Modelo de Clases



En esta entrega se puso como propósito mejorar la interacción con el usuario y las fallas de la entrega anterior. Las clases de este proyecto son:

1. Intercambio Datos: Esta clase estática fué creada para esta entrega



con el fin de unificar la información entregada. Sus atributos son:

- a) filas de tipo int
- b) columnas de tipo int
- c) Bitmon Iniciales de tipo int

Por otro lado, no tiene métodos.

2. Bitmons: En esta clase es una clase abstracta. Se tuvo que agregar un atributo de esperanza de vida y algunos métodos, con el fin de corregir la entrega pasada, mejorar la comunicación del programa y facilitar el uso de esta clase en sí. Sus atributos son:

- a) Tiempo de Vida de tipo int
- b) Puntos de vida de tipo int
- c) Puntos de ataque de tipo int
- d) Especie de tipo String
- e) Cantidad de Hijos de tipo int
- f) Esperanza de vida de tipo int
- g) Posy: Coordenada de ubicación del Bitmon en el eje Y de tipo int
- h) Posx: Coordenada de ubicación del Bitmon en el eje X de tipo int

Por otra parte, sus métodos son:

- a) Get Esperanza: Este método abstracto busca entregar la esperanza de vida, retorna una variable de tipo int.



- b) Muerte: Clase de tipo abstracta con el fin de informar cuando el Bitmon ha muerto, retorna un bool.
- c) Get TiempoDeVida: Este ayuda a obtener el tiempo de vida para agilizar el programa. Es un método abstracto que retorna una variable de tipo int.
- d) Daño: Este método ayuda a retornar en valores enteros el Daño causado por otro Bitmon hacia el. Es de tipo Abstracto y tiene un retorno de enteros.
- e) Get Especie: Este método ayuda a obtener el tipo de especie del Bitmon. Es de tipo abstract y devuelve un String.
- f) Get Especie Vacío: Retorna el nombre de la especie vacío en caso de no lograr agregarlo. Es es un método abstracto y retorna un string.
- g) Get posx: Este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje x. Es abstracto y retorna un entero.
- h) Get posy: Este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje y. Retorna un valor entero y es abstracto.
- i) Reproducirse: Este método se mantuvo de la entrega anterior, es abstracto y sin retorno.
- j) ReducirPuntosDeVida: Este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.
- k) moverse: Ayuda a realizar el movimiento del Bitmon por el mapa, es abstracto y no tiene retorno.

3. Dorvalos: En esta clase es una clase es heredada de Bitmon. Se realizaron ciertos cambios con único fin de adaptarse a la Clase Bitmons



reparada. Sus atributos son heredados y serían:

- a) Esperanza de Vida de tipo int
- b) Tiempo de Vida de tipo int
- c) Puntos de vida de tipo int
- d) Puntos de ataque de tipo int
- e) Especie de tipo String
- f) Cantidad de Hijos de tipo int
- g) Posy: Coordenada de ubicación del Bitmon en el eje Y de tipo int
- h) Posx: Coordenada de ubicación del Bitmon en el eje X de tipo int

Por otra parte, sus métodos son:

- a) Get Esperanza: Este método abstracto busca entregar la esperanza de vida, retorna una variable de tipo int.
- b) Muerte: Clase de tipo abstracta con el fin de informar cuando el Bitmon ha muerto, retorna un bool.
- c) Get TiempoDeVida: Este ayuda a obtener el tiempo de vida para agilizar el programa. Es un método que retorna una variable de tipo int.
- d) Daño: Este método ayuda a retornar en valores enteros el Daño causado por otro Bitmon hacia el. Tiene un retorno de enteros.
- e) Get Especie: Este método ayuda a obtener el tipo de especie del Bitmon. Devuelve un String.



- f)* Get Especie Vacío: Retorna el nombre de la especie vacío en caso de no lograr agregarlo. Retorna un string.
 - g)* Get posx: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje x. Retorna un entero.
 - h)* Get posy: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje y. Retorna un valor entero.
 - i)* Reproducirse: este método se mantuvo de la entrega anterior, es sin retorno.
 - j)* ReducirPuntosDeVida: este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.
 - k)* Moverse: ayuda a realizar el movimiento del Bitmon por el mapa. No tiene retorno.
4. Doti: En esta clase es una clase es heredada de Bitmon. Se tuvo que realizar ciertos cambios con el fin de corregir la entrega pasada, facilitar el uso de esta clase en el programa y adaptarse a la Clase Bitmons. Sus atributos son heredados y serían:
- a)* Esperanza de Vida de tipo int
 - b)* Tiempo de Vida de tipo int
 - c)* Puntos de vida de tipo int
 - d)* Puntos de ataque de tipo int
 - e)* Especie de tipo String
 - f)* Cantidad de Hijos de tipo int
 - g)* posy: Coordenada de ubicación del Bitmon en el eje Y, retorna un int



- h)* posx: Coordenada de ubicación del Bitmon en el eje X, retorna un int

Por otra parte, sus métodos son:

- a)* Get Esperanza: Este método abstracto busca entregar la esperanza de vida, retorna una variable de tipo int.
- b)* Muerte: Clase de tipo abstracta con el fin de informar cuando el Bitmon ha muerto, retorna un bool.
- c)* Get TiempoDeVida: Este ayuda a obtener el tiempo de vida para agilizar el programa. Es un método que retorna una variable de tipo int.
- d)* Daño: Este método ayuda a retornar en valores enteros el Daño causado por otro Bitmon hacia el. Tiene un retorno de enteros.
- e)* Get Especie: Este método ayuda a obtener el tipo de especie del Bitmon. Devuelve un String.
- f)* Get Especie Vacío: Retorna el nombre de la especie vacío en caso de no lograr agregarlo. Retorna un string.
- g)* Get posx: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje x. Retorna un entero.
- h)* Get posy: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje y, retorna un valor entero.
- i)* Reproducirse: este método se mantuvo de la entrega anterior, es sin retorno.
- j)* ReducirPuntosDeVida: este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.



k) moverse: ayuda a realizar el movimiento del Bitmon por el mapa.
No tiene retorno.

5. Gofue: En esta clase es una clase es heredada de Bitmon. Se tuvo que realizar ciertos cambios con el fin de corregir la entrega pasada, facilitar el uso de esta clase en el programa y adaptarse a la Clase Bitmons Sus atributos son heredados y serían:

- a)* Esperanza de Vida de tipo int
- b)* Tiempo de Vida de tipo int
- c)* Puntos de vida de tipo int
- d)* Puntos de ataque de tipo int
- e)* Especie de tipo String
- f)* Cantidad de Hijos de tipo int
- g)* posy: Coordenada de ubicación del Bitmon en el eje Y de tipo int
- h)* posx: Coordenada de ubicación del Bitmon en el eje X de tipo int

Por otra parte, sus métodos son:

- a)* Get Esperanza: Este método abstracto busca entregar la esperanza de vida, retorna una variable de tipo int.
- b)* Get TiempoDeVida: este ayuda a obtener el tiempo de vida para agilizar el programa. Es un método que retorna una variable de tipo int.
- c)* Daño: Este método ayuda a retornar en valores enteros el Daño causado por otro Bitmon hacia el. Tiene un retorno de enteros.
- d)* Get Especie: Este método ayuda a obtener el tipo de especie del Bitmon. Devuelve un String.



- e)* Get Especie Vacío: Retorna el nombre de la especie vacío en caso de no lograr agregarlo. Retorna un string.
 - f)* Get posx: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje x. Retorna un entero.
 - g)* Get posy: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje y. Retorna un valor entero.
 - h)* Muerte: Clase de tipo abstracta con el fin de informar cuando el Bitmon ha muerto, retorna un bool.
 - i)* Reproducirse: este método se mantuvo de la entrega anterior, es sin retorno.
 - j)* ReducirPuntosDeVida: este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.
 - k)* moverse: ayuda a realizar el movimiento del Bitmon por el mapa. No tiene retorno.
6. En esta clase es una clase es heredada de Bitmon. Se tuvo que realizar ciertos cambios con el fin de corregir la entrega pasada, facilitar el uso de esta clase en el programa y adaptarse a la Clase Bitmons Sus atributos son heredados y serían:
- a)* Esperanza de vida de tipo int
 - b)* Tiempo de Vida de tipo int
 - c)* Puntos de vida de tipo int
 - d)* Puntos de ataque de tipo int
 - e)* Especie de tipo String
 - f)* Cantidad de Hijos de tipo int



- g)* Posy: Coordenada de ubicación del Bitmon en el eje Y de tipo `int`
- h)* Posx: Coordenada de ubicación del Bitmon en el eje X de tipo `int`

Por otra parte, sus métodos son:

- a)* Get Esperanza: Este método abstracto busca entregar la esperanza de vida, retorna una variable de tipo `int`.
- b)* Get TiempoDeVida: este ayuda a obtener el tiempo de vida para agilizar el programa. Es un método que retorna una variable de tipo `int`.
- c)* Daño: Este método ayuda a retornar en valores enteros el Daño causado por otro Bitmon hacia el. Tiene un retorno de enteros.
- d)* Get Especie: Este método ayuda a obtener el tipo de especie del Bitmon. Devuelve un `String`.
- e)* Get Especie Vacío: Retorna el nombre de la especie vacío en caso de no lograr agregarlo. Retorna un `string`.
- f)* Get posx: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje x. Retorna un entero.
- g)* Get posy: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje y. Retorna un valor entero.
- h)* Muerte: Clase de tipo abstracta con el fin de informar cuando el Bitmon ha muerto, retorna un `bool`.
- i)* Reproducirse: este método se mantuvo de la entrega anterior, es sin retorno.



j) ReducirPuntosDeVida: este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.

k) moverse: ayuda a realizar el movimiento del Bitmon por el mapa. No tiene retorno.

7. Wetar: En esta clase es una clase heredada de Bitmon. Se tuvo que realizar ciertos cambios con el fin de corregir la entrega pasada, facilitar el uso de esta clase en el programa y adaptarse a la Clase Bitmons. Sus atributos son heredados y serían:

- a) Esperanza de Vida de tipo int
- b) Tiempo de Vida de tipo int
- c) Puntos de vida de tipo int
- d) Puntos de ataque de tipo int
- e) Especie de tipo String
- f) Cantidad de Hijos de tipo int
- g) posy: Coordenada de ubicación del Bitmon en el eje Y
- h) posx: Coordenada de ubicación del Bitmon en el eje X.

Por otra parte, sus métodos son:

- a) Get Esperanza: Este método abstracto busca entregar la esperanza de vida, retorna una variable de tipo int.
- b) Get TiempoDeVida: este ayuda a obtener el tiempo de vida para agilizar el programa. Es un método que retorna una variable de tipo int.
- c) Daño: Este método ayuda a retornar en valores enteros el Daño causado por otro Bitmon hacia el. Tiene un retorno de enteros.



- d)* Get Especie: Este método ayuda a obtener el tipo de especie del Bitmon. Devuelve un String.
 - e)* Get Especie Vacío: Retorna el nombre de la especie vacío en caso de no lograr agregarlo. Retorna un string.
 - f)* Get Posx: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje x. Retorna un entero.
 - g)* Get Posy: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje y. Retorna un valor entero.
 - h)* Muerte: Clase de tipo abstracta con el fin de informar cuando el Bitmon ha muerto, retorna un bool.
 - i)* Reproducirse: este método se mantuvo de la entrega anterior, es sin retorno.
 - j)* ReducirPuntosDeVida: este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.
 - k)* Moverse: ayuda a realizar el movimiento del Bitmon por el mapa. No tiene retorno.
8. Ent: En esta clase es una clase es heredada de Bitmon. Se tuvo que realizar ciertos cambios con el fin de corregir la entrega pasada, facilitar el uso de esta clase en el programa y adaptarse a la Clase Bitmons Sus atributos son heredados y serían:
- a)* Esperanza de Vida de tipo int
 - b)* Tiempo de Vida de tipo int
 - c)* Puntos de vida de tipo int
 - d)* Puntos de ataque de tipo int



- e)* Especie de tipo String
- f)* Cantidad de Hijos de tipo int
- g)* Posy: Coordenada de ubicación del Bitmon en el eje Y
- h)* Posx: Coordenada de ubicación del Bitmon en el eje X.

Por otra parte, sus métodos son:

- a)* Get Esperanza: Este método abstracto busca entregar la esperanza de vida, retorna una variable de tipo int.
- b)* Get TiempoDeVida: este ayuda a obtener el tiempo de vida para agilizar el programa. Es un método que retorna una variable de tipo int.
- c)* Daño: Este método ayuda a retornar en valores enteros el Daño causado por otro Bitmon hacia el. Tiene un retorno de enteros.
- d)* Get Especie: Este método ayuda a obtener el tipo de especie del Bitmon. Devuelve un String.
- e)* Get Especie Vacío: Retorna el nombre de la especie vacío en caso de no lograr agregarlo. Retorna un string.
- f)* Get posx: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje x. Retorna un entero.
- g)* Get posy: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje y. Retorna un valor entero.
- h)* Muerte: Clase de tipo abstracta con el fin de informar cuando el Bitmon ha muerto, retorna un bool.
- i)* Reproducirse: este método se mantuvo de la entrega anterior, es sin retorno.



j) ReducirPuntosDeVida: este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.

k) moverse: este aplica un Get exception.

9. Mapa: En esta entrega se tuvo que realizar algunos cambios en este item, quedando los siguientes atributos:

a) Filas mapa de tipo int

b) Columnas mapa de tipo int

c) Mapa de tipo Celda[,]

d) Bitmons mapa de tipo lista de Bitmons

e) bitmon iniciales: utiliza los datos entregados para crear los bitmons.

Por otra parte, sus métodos son:

a) Creacion Mapa: crea los terrenos celda por celda, requiere las filas y columnas.

b) CrearBitmon: Crea los bitmons para el inicio del juego y los posiciona en el mapa.

c) Crear Bitmon Reproduccion: Este método tiene por objetivo crear un bitmon y agregarlo en el mapa.

d) Crear Bitmon Ent: Estos Bitmons no pueden tener hijos entre ellos, por lo que se generan de manera espontánea cada 3 meses en el mapa gracias a este método.

10. Celda: Esta clase se mantiene constante en base a la entrega anterior.

Sus atributos son:



- a) Tipo de terreno de tipo Terreno
- b) Bitmons celda: Una lista de Bitmons para lograr posicionarlos en el mapa.

Por otra parte, sus métodos son:

- a) Set tipoTerreno: Este pretende obtener el tipo de terreno que tiene una celda. No tiene retorno y requiere un valor entero.
- b) AgregarBitmon: Este agrega los Bitmons creados durante el programa. Not tiene retorno.

11. Terreno: Esta clase se mantuvo constante en relación a la entrega anterior Su atributo es:

- a) Tipo terreno de tipo string, es protegido

Por otra parte, su método es:

- a) Get terreno: este obtiene el terreno de tipo string.

12. Acuático: Es una clase hija de terreno. Su atributo heredado es:

- a) Tipo terreno de tipo string, siendo estrictamente acuatico

No tiene métodos.

13. Desierto: Es una clase hija de terreno. Su atributo heredado es:

- a) Tipo terreno de tipo string, siendo estrictamente "Desierto"

No tiene métodos.

14. Nieve:Es una clase hija de terreno. Su atributo heredado es:



- a) Tipo terreno de tipo string, siendo estrictamente "nieve"

No tiene métodos.

15. Vegetacion: Es una clase hija de terreno. Su atributo heredado es:

- a) Tipo terreno de tipo string, siendo estrictamente "vegetacion"

No tiene métodos.

16. Volcán: Es una clase hija de terreno. Su atributo heredado es:

- a) Tipo terreno de tipo string, siendo estrictamente "volcán"

No tiene métodos.

17. Controlador: Esta clase es el controlador de la interfaz, siguiendo el diagrama UML. tiene los siguientes atributos:

- a) Inicio de la clase parcial inicio
- b) Form1 de la clase parcial Form1
- c) Datos finales de la clase parcial Datos Finales
- d) Modelo de la clase Modelo

Por otro lado, sus métodos son:

- a) Salir: de tipo void Este cumple con la función de cerrar el programa.
- b) Datos Iniciales: de tipo void con la fin de hacer el manejo de datos entre el usuario y el programa en sí.

18. Modelo: Esta clase tiene como función cambiar el texto de los labels, relacionando los datos del programa con la interface visual. No tiene atributos, su método es: Cambio: de retorno void o sin retorno.



Por otro lado, existen Clases Parciales o Mejor llamadas Windows Forms, que son la imagen del programa. Estas son:

1. Inicio: Esta es el inicio del juego, donde se selecciona el tamaño de mapa y se aprieta el botón para iniciar el juego. También se puede seleccionar salir.
2. Form1: Esta es la que hace funcionar el juego, va mostrando el mapa, como se transforma y sus bitmons se mueven.
3. Datos finales: Esta aparece antes de salir del programa, el cual da una estadística del mapa desde sus inicios hasta el tiempo actual, entregando datos de la poblacion.



2. División de tareas

Como se habló en la entrega anterior, uno de los integrantes desertó del proyecto debido a sus malas notas en el ramo, por lo que se tuvieron que redistribuir las tareas entre los que quedaron en el grupo.

Entre Matías y Alejandra se dedicaron a arreglar los errores o partes faltantes de la entrega anterior.

La comunicación hacia los ayudantes fue completamente vía mensaje o teléfono para esta entrega, debido a la falta de tiempo que ambas partes tienen. De esta se encargó Alejandra y, en conjunto con Matías, complementaban estas resolviendo dudas con el profesor entre las clases.

Alejandra se encargó del procesamiento de los datos entre forms y los resultados finales y su debido guardado en archivo. Por otro lado Matías se encargó de lograr el sistema de MVC y el uso del controller.

Ambos se encargaron de corregir y adaptar el proyecto anterior y con esto lograr sacar el proyecto adelante. Se realizó mucho trabajo por separado, pero también hubo una reunión final para cerrar etapas.



3. Supuestos

1. Los Bitmons hijos solo pueden tener la especie de uno de los padres
2. Se asume que todos los Bitmons son asexuados
3. Las Los Bitmons con la misma especies tienen rangos parecidos de ataque y vida.
4. Las enemistades y terrenos son:
 - Doti, sin enemigos, sin ventaja, terreno beneficiario = vegetación
 - Taplan: debilidad Wetar, ventaja frente a Gofue, terreno = vegetación
 - Ent: sin ventaja, desventaja frente a Doti, terreno beneficiario = fuego
 - Dorvalos: debilidad Doti, ventaja Ent, terreno beneficiario = acuático
 - Gofue: debilidad Dorvalos, ventaja frente a Doti, terreno beneficiario = desierto
 - Wetar: debilidad Gofue, ventaja frente a Dorvalos, terreno beneficiario= Acuatico.
5. Se asume que los Ent no pueden reproducirse con otras especies ni pueden defenderse.
6. Se asume que los meses de simulación dependerán de los clicks al botón para esto y cada uno simula un mes en Bitmonlandia.
7. La tasa de mortalidad estará dada por los Bitmons que al pasar sus meses de vida esperados o morir en una pelea o caza, se irán al Bithala,



una lista y la tasa estaría dada por el largo de la lista en sí.

8. La tasa de natalidad estará dada por la diferencia entre los bitmons totales del juego y los bitmons iniciales.
9. Al reproducirse los Bitmons, estos pueden aparecer en cualquier parte del mapa.
10. El mapa se crea de manera aleatoria para agilizar su uso.
11. Al salir del programa, la información de la simulación es guardada en un archivo local de tipo txt de nombre Dato.



4. Manual de usuarios

Al ingresar al menú, tiene la opción de iniciar o salir mediante botones. Para poder iniciar, el usuario debe seleccionar las dimensiones del mapa. Estas corresponden a la cantidad de cuadrados de terreno presentes como mapa, cada cuadrado representará un tipo de terreno mediante una imagen que asimila su color y textura. Por otro lado, también debe elegir la cantidad de Bitmons iniciales en su programa.

Para comenzar, debe primero insertar las dimensiones del eje X, seleccionando del ComboBox inferior la opción de la cantidad de columnas que desea, luego las dimensiones en el eje Y, seleccionando del ComboBox inferior la opción de la cantidad de columnas que desea. Para el paso siguiente, debe seleccionar la cantidad de Bitmons para que habiten en los inicios de su Bitmonandia, seleccionando en el ComboBox que lo indica. Finalmente, presione aceptar.

Al aceptar, se abrirá una nueva ventana llamada Bitmonlandia, en esta saldrán diferentes opciones, como Reiniciar, siguiente mes de simulación, ver información actual o Salir, en conjunto con la imagen de cada uno de los Bitmons existentes y el mapa en sí. El Mapa muestra como un conjunto de cuadrados de colors y cada uno representa un tipo de terreno con su parecido en color y textura, identificandose como:

1. El terreno de Volcán se ve como lava o color Rojo
2. La Vegetación se ve como pasto o color verde
3. La nieve se ve como nieve o color blanco



4. El desierto se ve como arena o color beige
5. El Acuático se ve como agua o color azul

Presione el botón siguiente mes de simulación para hacer la simulación de un mes en Bitmonlandia, acá podrá ver como los bitmons se van moviendo y haciendo acciones en el mapa. Cada vez que presione el botón, pasará un mes de vida de los Bitmons y podrá ver los cambios de este en el mapa.

Al pasar los meses, los Bitmons se irán reproduciendo, naciendo y muriendo de acuerdo a la situación. Si desea ver la cantidad de Bitmons actuales de su mapa, seleccione el botón de ver info. actual, este hará que se muestre la cantidad donde aparece un Dibujo de cada Bitmon, en conjunto con la cantidad que hay de ellos presentes en el mapa.

Si desea comenzar una nueva simulación, puede presionar reiniciar, que volverá al menú de inicio para volver a pedirle todos los datos antes mencionados.

Si desea salir del juego, presione el botón salir, el cual antes de salir del juego, le mostrará todas las estadísticas de su simulación y el progreso de sus Bitmons. Para salir del programa, presione aceptar o la cruz de la esquina superior derecha.