



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

Documentación Entrega 3

Integrantes:

Alejandra Izquierdo, Matías Reyes

Pedro Donoso

Profesor: Nicolás Gomez

Ayudante: Sebastián Baixas, Benjamín Perez

30/05/2019



1. Modelo de Clases

Para esta entrega se tuvo que adaptar nuestro programa a Windows Forms, por lo que se tuvo que incluir clases que no estaban contempladas en el plan inicial ni en la entrega 2. Las clases de este proyecto son:

1. Bitmons: En esta clase es una clase abstracta. Se tuvo que realizar ciertos cambios en la manera de escribir sus coordenadas y algunos métodos, con el fin de corregir la entrega pasada y facilitar el uso de esta clase en el programa. 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)* posy: Coordenada de ubicación del Bitmon en el eje Y
- g)* posx: Coordenada de ubicación del Bitmon en el eje X.

Por otra parte, sus métodos son:

- a)* 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.
- b)* 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.



- c) Get Especie: Este método ayuda a obtener el tipo de especie del Bitmon. Es de tipo abstract y devuelve un String.
 - d) 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.
 - e) 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.
 - f) 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.
 - g) Muerte: Este método analiza cuando el Bitmon muere. Retorna un int y es abstracto.
 - h) Reproducirse: Este método se mantuvo de la entrega anterior, es abstracto y sin retorno.
 - i) ReducirPuntosDeVida: Este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.
 - j) moverse: Ayuda a realizar el movimiento del Bitmon por el mapa, es abstracto y no tiene retorno.
2. Dorvalos: En esta clase es una clase es heredada de Bitmon. Se tuvo que realizar ciertos cambios en la manera de escribir sus coordenadas y algunos métodos, 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) 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) posy: Coordenada de ubicación del Bitmon en el eje Y
- g) posx: Coordenada de ubicación del Bitmon en el eje X.

Por otra parte, sus métodos son:

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



- i)* ReducirPuntosDeVida: este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.
 - j)* moverse: ayuda a realizar el movimiento del Bitmon por el mapa. No tiene retorno.
- 3. Doti: En esta clase es una clase heredada de Bitmon. Se tuvo que realizar ciertos cambios en la manera de escribir sus coordenadas y algunos métodos, 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)* 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)* posY: Coordenada de ubicación del Bitmon en el eje Y
 - g)* posX: Coordenada de ubicación del Bitmon en el eje X.

Por otra parte, sus métodos son:

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



- d)* Get Especie Vacío: Retorna el nombre de la especie vacío en caso de no lograr agregarlo. Retorna un string.
 - e)* Get posx: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje x. Retorna un entero.
 - f)* Get posy: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje y. Retorna un valor entero.
 - g)* Muerte: Este método analiza cuando el Bitmon muere. Retorna un int.
 - h)* Reproducirse: este método se mantuvo de la entrega anterior, es sin retorno.
 - i)* ReducirPuntosDeVida: este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.
 - j)* moverse: ayuda a realizar el movimiento del Bitmon por el mapa. No tiene retorno.
4. Gofue: En esta clase es una clase es heredada de Bitmon. Se tuvo que realizar ciertos cambios en la manera de escribir sus coordenadas y algunos métodos, 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)* 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)* posy: Coordenada de ubicación del Bitmon en el eje Y
- g)* posx: Coordenada de ubicación del Bitmon en el eje X.

Por otra parte, sus métodos son:

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



5. En esta clase es una clase es heredada de Bitmon. Se tuvo que realizar ciertos cambios en la manera de escribir sus coordenadas y algunos métodos, 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) 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) posy: Coordenada de ubicación del Bitmon en el eje Y
- g) posx: Coordenada de ubicación del Bitmon en el eje X.

Por otra parte, sus métodos son:

- a) Get TiempoDeVida: este ayuda a obtener el tiempo de vida para agilizar el programa. Es un método que retorna una variable de tipo int.
- b) Daño: Este método ayuda a retornar en valores enteros el Daño causado por otro Bitmon hacia el. Tiene un retorno de enteros.
- c) Get Especie: Este método ayuda a obtener el tipo de especie del Bitmon. Devuelve un String.
- d) Get Especie Vacío: Retorna el nombre de la especie vacío en caso de no lograr agregarlo. Retorna un string.
- e) Get posx: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje x. Retorna un entero.



- f)* Get posy: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje y. Retorna un valor entero.
 - g)* Muerte: Este método analiza cuando el Bitmon muere. Retorna un int.
 - h)* Reproducirse: este método se mantuvo de la entrega anterior, es sin retorno.
 - i)* ReducirPuntosDeVida: este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.
 - j)* moverse: ayuda a realizar el movimiento del Bitmon por el mapa. No tiene retorno.
6. Wetar: En esta clase es una clase heredada de Bitmon. Se tuvo que realizar ciertos cambios en la manera de escribir sus coordenadas y algunos métodos, 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)* 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)* posy: Coordenada de ubicación del Bitmon en el eje Y
 - g)* posx: Coordenada de ubicación del Bitmon en el eje X.

Por otra parte, sus métodos son:



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

7. Ent: En esta clase es una clase es heredada de Bitmon. Se tuvo que realizar ciertos cambios en la manera de escribir sus coordenadas y algunos métodos, 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) 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) posy: Coordenada de ubicación del Bitmon en el eje Y
- g) posx: Coordenada de ubicación del Bitmon en el eje X.

Por otra parte, sus métodos son:

- a) Get TiempoDeVida: este ayuda a obtener el tiempo de vida para agilizar el programa. Es un método que retorna una variable de tipo int.
- b) Daño: Este método ayuda a retornar en valores enteros el Daño causado por otro Bitmon hacia el. Tiene un retorno de enteros.
- c) Get Especie: Este método ayuda a obtener el tipo de especie del Bitmon. Devuelve un String.
- d) Get Especie Vacío: Retorna el nombre de la especie vacío en caso de no lograr agregarlo. Retorna un string.
- e) Get posx: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje x. Retorna un entero.
- f) Get posy: este ayuda a ver en que parte del mapa se sitúa el Bitmon en el eje y. Retorna un valor entero.



- g)* Muerte: Este método analiza cuando el Bitmon muere. Retorna un int.
 - h)* Reproducirse: este método se mantuvo de la entrega anterior, es sin retorno.
 - i)* ReducirPuntosDeVida: este reduce los puntos de vida por algún ataque, no tiene retorno y requiere de un valor dado.
 - j)* moverse: este aplica un Get exception.
8. Mapa: Esta cambió considerablemente debido a que el cambio de sistema utilizado para esta entrega no era compatible con el usado anteriormente. Sus atributos son:
- 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)* n taplan: corresponde al número de Taplan de tipo int
 - f)* n Wetar: corresponde al número de Wetar de tipo int
 - g)* n Gofue: corresponde al número de Gofue de tipo int
 - h)* n Dorvalos: corresponde al número de Dorvalos de tipo int
 - i)* n Doti: corresponde al número de Doti de tipo int
 - j)* n Ent: corresponde al número de Ent de tipo int

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.

9. Celda: Esta fué creada para esta entrega con el fin de agilizar el uso de los botones y el mapa. 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.

10. Terreno: Esta clase se creó en esta entrega para poder manejar mejor el terreno dentro de las celdas. Su atributo es:

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

Por otra parte, su método es:

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

11. Acuático: Esta clase se creó en esta entrega para poder manejar mejor el terreno dentro de las celdas. Su atributo heredado es:

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



No tiene métodos.

12. Desierto: Esta clase se creó en esta entrega para poder manejar mejor el terreno dentro de las celdas. Su atributo heredado es:

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

No tiene métodos.

13. Nieve: Esta clase se creó en esta entrega para poder manejar mejor el terreno dentro de las celdas. Su atributo heredado es:

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

No tiene métodos.

14. Vegetacion: Esta clase se creó en esta entrega para poder manejar mejor el terreno dentro de las celdas. Su atributo heredado es:

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

No tiene métodos.

15. Volcán: Esta clase se creó en esta entrega para poder manejar mejor el terreno dentro de las celdas. Su atributo heredado es:

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

No tiene métodos.

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.



2. División de tareas

En esta entrega tuvimos dificultades muy distintas a la 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 plantearon las clases, métodos y atributos para esta entrega, en conjunto con las asociaciones, composiciones, agregaciones y herencias.

Matías se reunió directamente con el ayudante y Alejandra se encargó de solucionar las dudas a la distancia con ellos y resolver las dudas del proyecto con el profesor después de la clase y mediante reunión.

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 principalmente presencial, juntándose en Pirque para trabajar.



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 Los Bitmons con la misma especie tienen rangos parecidos de ataque y vida.
3. 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 = Acuático.
4. Se asume que los Ent no pueden reproducirse con otras especies ni pueden defenderse.
5. Se asume que la simulación es de $n = 200$ meses.
6. 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.



7. La tasa de natalidad estará calculado por la sumatoria de los hijos de cada Bitmon del juego dividido en 2 (debido a la presencia de padre y madre). Por otro lado, esta lista no considera los ents nacidos cada 3 meses, ya que son un dato constante en cada simulación e inflaría mucho los resultados.
8. El tiempo de vida de los Bitmons está dado por el promedio del puntaje de edad alcanzada al minuto de morir (calculado a partir de los Bitmons del Bithala).
9. Al morir, su cuerpo muerto permanecerá en el mapa, pero será agregado al Bithala y no podrá moverse.
10. El mapa se crea de manera aleatoria para agilizar su uso.



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.

Para comenzar, debe primero insertar las dimensiones del eje X, seleccionando del ComboBox inferior la opción de la cantidad de columnas que desea y 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 meses de la simulación 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, Iniciar Juego o Salir, en conjunto con la imagen de cada uno de los Bitmons existentes y la cantidad presente en el mapa.

Presione el botón iniciar juego para que aparezca el mapa y comience el juego. El Mapa muestra como un conjunto de cuadrados de colores 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

Este mapa contiene dentro de el a los Bitmons iniciales de la simulación, los cuales se irán moviendo al pasar los meses de Bitmons que son alrededor de 1 segundo por mes. Al pasar los meses, los Bitmons se irán reproduciendo, naciendo y muriendo de acuerdo a la situación. La cantidad de Bitmons finales por cada tipo podrá verse reflejado en el lado derecho del programa, donde aparece un Dibujo de cada Bitmon, en conjunto con la cantidad que hay de ellos presentes en el mapa.

Al finalizar los meses de simulación, usted podrá reiniciar la simulación o salir del programa apretando los botones.