

Trabajo Práctico 2 - Algoritmos 3

Brasburg. Cano. Raik.
FIUBA

October 8, 2023

Objetivos

Los objetivos de esta segunda parte del trabajo práctico son:

- Corregir las correcciones de la entrega anterior
- Demostrar el conocimiento de patrones de diseño de y de principios de diseño utilizándolos donde sea conveniente.
- Seguir las buenas prácticas de programación
- Lograr una cobertura de pruebas confiable
- Extender el modelo de la entrega anterior para satisfacer las nuevas necesidades
- (Opcional) Persistir datos del sistema del trabajo práctico
 - No es obligatorio tener realizada esta parte para esta entrega. Sin embargo, si tienen la posibilidad de avanzarla, es recomendable dado que sí será requerida para la última entrega del trabajo.

1 Requerimientos

La nueva entrega debe satisfacer los requerimientos mencionados en el enunciado de la entrega anterior y agregar las siguientes funcionalidades.

1.1 Estados

Hasta ahora, los Pokemon podían tener uno o ningún estado. Para ser un juego mas realista, se pide que los Pokemons puedan tener más de un estado. Por ejemplo puede estar dormido y envenenado en simultaneo.

Por otro lado, se necesita agregar un estado nuevo: "Confuso". Dicho estado se genera a partir de la habilidad "Confundir". Su efecto es que el Pokemon confundido tiene un tercio de posibilidad de herirse a si mismo al realizar cualquier habilidad, el daño recibido debe ser del 15% de la salud máxima del Pokemon. Se cura automáticamente luego de 3 turnos.

1.2 Items

Al ver el éxito de sus pociones, la fábrica decidió poner a la venta una nueva poción llamada "Poción molesta alumnos", que cura un tercio de los puntos de vida originales del Pokemon. Dicha poción debería ser tenida en cuenta

por el juego.

Además, para hacer el juego mas justo, la liga Pokemon decidió que cada jugador puede tener como máximo 1 (una) sola Hiper Poción.

1.3 Clima

Al notar un deficit de vitamina D en los Pokemon, se decidió realizar las batallas en el exterior por lo que hay un nuevo factor que puede afectar a los Pokemon: El clima. Algunos Pokemon tienen habilidades que permiten modificar el clima, por ejemplo hacer que llueva. Cada clima favorece a un tipo de Pokemon particular agregándole un 10% mas de poder a sus ataques. Por ejemplo: Si el clima es "Soleado" favorece a los Pokemon de tipo fuego, si el Pokemon hace un ataque que resulta en 50 puntos de ataque, finalmente atacará con 55. El clima es único y en caso de que un Pokemon utilice una habilidad que lo modifique, cambiará el actual. Cada clima desaparece luego de 5 turnos. Inicialmente se sorteará el clima inicial, teniendo "Sin clima" 2 tercios de probabilidad, mientras que el resto de los climas ocuparán el tercio restante, que serán equiprobables entre sí. Los climas disponibles son:

- Soleado: Favorece al tipo fuego
- Lluvia: Favorece al tipo agua y planta.
- Tormenta de arena: Favorece al tipo tierra y roca.
- Niebla: Favorece al tipo fantasma y psíquico.
- Tormenta de rayos: Favorece al tipo eléctrico.
- Huracán: Favorece al tipo volador

Podrán agregar climas a gusto del grupo.

Por otro lado, los climas "Tormenta de arena", "Tormenta" y "Huracán" dañan a ambos pokemon en el campo de batalla al comienzo de cada turno, restandoles un 3% de su vida máxima. Este daño no se efectúa si el clima favorece al tipo del Pokemon.

2 Pruebas

Para esta entrega se pide desarrollar pruebas unitarias obteniendo una cobertura de al menos 75% de todo el código, no solamente el código de esta entrega si no también de la anterior. Además, se requiere pruebas de integración del

modelo. Recomendación: Empezarlas en la clase madre del modelo "Juego" o como se le nombre. Dicha prueba de probar un flujo completo de batalla en la cual cada jugador tiene al menos dos Pokemon, tienen items (y los utilizan, al menos 2 tipos de items distintos) y los Pokemon utilizando distintos tipos de habilidades. Recordar que las pruebas muestran que el código funcione como se espera (al menos que se cumpla la prueba) por lo que se recomienda usarlas a su favor e ir implementandolas antes o a medida que se implementa el código fuente y no dejarlas para el final.

3 Patrones y principios

Si bien no se pide la utilización de ningún patron de diseño en particular, identificar aquellos que hayan sido utilizado y explicarlos, justificando su uso. Además, agregar una sección donde reconozcan principios de diseño incumplidos y bad smells de la primer entrega, y comentar como se corrigió (en caso de haberlo hecho).

4 Persistencia

Los Pokemon, los items y las habilidades estarán definidos en archivos de tipo JSON (<https://www.json.org/json-es.html>). También se dispondrá de un archivo con los datos de la partida, esto es el nombre de cada jugador (ya no se debe obtener por sistema) y referencia los Pokemon e items que dispone. La definición de la estructura de cada archivo está disponible en este documento pero los datos debe completarlos (inventarlos) cada grupo.

El curso provee un archivo de Pokemon de ejemplo que, no están obligados a utilizarlo, el sistema debe soportar. Dicho archivo NO establece las habilidades, cada Pokemon no tiene habilidades (vector vacio) y deberían completarlo.

4.1 pokemons.json

Este archivo es un vector de objetos JSON. Cada Pokemon tiene definidas sus estadísticas y tiene referencias a sus habilidades. Además, cada uno tiene un "Id" único para poder ser referenciado en los otros archivos. Ejemplo:

```
[
  {
    "nombre": "Bulbasaur",
```

```

    "id": 10,
    "nivel": 5,
    "tipo": "planta",
    "historia": "Esta es una historia",
    "vidaMaxima": 100,
    "velocidad": 20,
    "defensa": 3,
    "ataque": 10,
    "habilidades": [56, 78, 2]
  },
  {
    "nombre": "Charmander",
    "id": 7,
    "nivel": 6,
    "tipo": "fuego",
    "historia": "Esta es una historia",
    "vidaMaxima": 85,
    "velocidad": 15,
    "defensa": 5,
    "ataque": 15,
    "habilidades": [3, 98, 24]
  }
]

```

4.2 items.json

Este archivo es un vector de objetos JSON donde cada objeto es un item. Se especifica en este documento la estructura general, cada grupo debe agregar el efecto del item (depende del tipo de habilidad), su categoría y su "potencia" de forma que mejor se adecuó a su modelo. Ejemplo:

```

[
  {
    "nombre": "Hiper Poción",
    "id": 0,
    ...
  },
  {
    "nombre": "Defensa x",
    "id": 4,
    ...
  }
]

```

```

    },
    {
      "nombre": "Cura Total",
      "id": 67,
      ...
    }
  ]

```

4.3 habilidades.json

Este archivo es un vector de objetos JSON donde cada objeto es una habilidad. Se especifica en este documento la estructura general, cada grupo debe agregar el efecto de la habilidad (depende del tipo de habilidad), su categoría y su "potencia" de forma que mejor se adecuó a su modelo. Ejemplo:

```

[
  {
    "nombre": "Lanzallamas",
    "id": 3,
    ...
  },
  {
    "nombre": "Latigo cepa",
    "id": 2,
    ...
  },
  {
    "nombre": "Despejar clima",
    "id": 98,
    ...
  }
]

```

4.4 partida.json

Este archivo es un vector de dos objetos JSON, uno por cada jugador. Cada objeto tendrá definido el nombre del jugador, sus items en formato de diccionario donde la clave será el "id" del item y el valor la cantidad que dispone el jugador y sus pokemons en formato de vector de "Id"s de los mismos. Ejemplo:

```
[
  {
    "nombre": "Ash",
    "items": {
      "0": 1,
      "4": 3
    },
    "pokemons": [7]
  },
  {
    "nombre": "Brook",
    "items": {
      "4": 1,
      "67": 10
    },
    "pokemons": [10]
  }
]
```

4.5 Resumen de partida

Luego de la partida el sistema deberá generar un archivo "informe.json" (incluir en el nombre del archivo la hora de finalización de la partida) con el resultado de la partida indicando la persona ganadora y el estado final de cada Pokemon (incluyendo vida restando y estado si tuviese alguno) y sus items (cuantos de cada uno no utilizó). Ejemplo:

```
[
  {
    "nombre": "Ash",
    "ganador": true,
    "items": {
      "0": 0,
      "4": 1
    },
    "pokemons": [{
      "id": 7,
      "vidaRestante": 34,
      "estado": "envenenado"
    }]
  },
  {
    "nombre": "Brook",
    "ganador": false,
    "items": {
      "4": 1,
      "67": 10
    },
    "pokemons": [10]
  }
]
```

```

    {
      "nombre": "Brook",
      "ganador": false,
      "items": {
        "4": 1,
        "67": 4
      },
      "pokemons": [{
        "id": 10,
        "vidaRestante": 0,
        "estado": "debilitado"
      }]
    }
  ]

```

5

Formato de entrega

Para realizar la entrega se debe mandar un mail a algoritmos3.fiuba@gmail.com indicando los integrantes del grupo adjuntando el informe e indicando cual es el branch y el repositorio de la entrega. El repositorio tiene que ser privado. El trabajo práctico debe realizarse utilizando Git y Github. Para realizar la entrega se debe crear un branch llamado "tp-x" (en este caso tp-1) y no se debe modificar mas luego de enviado el mail. Se debe dar permisos en el repositorio a los 3 docentes del curso. Se debe contar con un README que explique como correr el programa "desde cero", como instalar las dependencias y como correr el juego. Incluir también los archivos base (pokemons.json, items.json, habilidades.json y partida.json) como para que el juego inicie sin tener que crear ninguno de ellos.

Fecha de entrega

La fecha de entrega es el 03/11/2023.

Criterios de aprobación

Para aprobar el trabajo práctico se deben cumplir con los objetivos y puntos detallados en el enunciado.