Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
IIC2113 Diseño Detallado de Software

Entrega 1: Shin Megami Tensei

Francisco Ignacio Gazitúa Requena
Cristian Andrés Hinostroza Espinoza

Introducción

En esta entrega debes implementar el flujo principal del combate, lo que abarca el flujo completo del combate (aproximadamente desde la Sección 1 hasta la Sección 6 del enunciado) excluyendo cualquier habilidad, todas las afinidades menos **Neutral** y las acciones de invocar y pasar turno. Es decir, debes implementar:

1. La validación de los equipos.
2. El flujo de rondas entre los jugadores.
3. Las acciones atacar, disparar y rendirse.
4. El setup del tablero
5. Leer las unidades desde archivos json.
6. Cálculos de daño sin considerar afinidades.
7. Flujo completo del combate, considerando el consumo de turnos y el termino del combate.

Los test no simulan combates donde el rival tenga afinidades distintas a **Neutral** a los ataques del rival, por lo que solo debes implementar un caso del sistema de turnos.

Test cases

Para esta entrega debes completar los siguientes grupos de tests:

- E1-BasicCombat.
- E1-InvalidTeams.
- E1-Random

Cada test consiste en un archivo de texto que contiene el input y output esperado de tu programa. Todos los test cases se encuentran en el archivo `data.zip`. El siguiente es un extracto de uno de estos:

```
Elige un archivo para cargar los equipos
0: 001.txt
1: 002.txt
2: 003.txt
3: 004.txt
4: 005.txt
5: 006.txt
6: 007.txt
7: 008.txt
8: 009.txt
9: 010.txt
10: 011.txt
```

```
INPUT: 0
----------------------------------------
Ronda de Flynn (J1)
----------------------------------------
Equipo de Flynn (J1)
A-Flynn HP:971/971 MP:527/527
B-
C-
D-
Equipo de Itsuki (J2)
A-Itsuki HP:1136/1136 MP:413/413
B-
C-
D-
----------------------------------------
Full Turns: 1
Blinking Turns: 0
----------------------------------------
Orden:
1-Flynn
----------------------------------------
Seleccione una acción para Flynn
1: Atacar
2: Disparar
3: Usar Habilidad
4: Invocar
5: Pasar Turno
6: Rendirse
INPUT: 1
----------------------------------------
Seleccione un objetivo para Flynn
1-Itsuki HP:1136/1136 MP:413/413
2-Cancelar
INPUT: 1
----------------------------------------
Flynn ataca a Itsuki
Itsuki recibe 88 de daño
Itsuki termina con HP:1048/1136
----------------------------------------
Se han consumido 1 Full Turn(s) y 0 Blinking Turn(s)
Se han obtenido 0 Blinking Turn(s)
```

Tu programa debe generar el mismo output que aparece en el test para estar correcto. Notar que el test incluye el keyword "`INPUT: `" en algunas líneas (e.j., en la línea 13). Esto indica que hay que pedir un input al usuario (en vez de escribir algo) y el input ingresado por el usuario será el número que aparece luego de "`INPUT: `". Por ejemplo, si "`INPUT: 2`" entonces el número ingresado por el usuario será `2`.

Los tests de esta entrega comprueban que se cumplan dos escenarios.

El primer escenario consiste en verificar que el equipo elegido por cada jugador sea válido. Recordar que un equipo es inválido cuándo:

- No tiene samurai.
- Tiene más de un samurai.
- Tiene más de 8 unidades (incluyendo al samurai).
- Tiene alguna unidad repetida.
- Un samurai tiene más de 8 habilidades.
- Un samurai tiene alguna habilidad repetida.

Si un equipo es inválido, se muestra un mensaje indicando que la selección es inválida y termina el programa:

```
Elige un archivo para cargar los equipos
0: 01.txt
1: 02.txt
2: 03.txt
3: 04.txt
INPUT: 3
Archivo de equipos no válido
```

El segundo escenario es que ambos equipos ingresados sean válidos. Si esto ocurre, el juego comenzará, momento en el que los jugadores decidirán qué acciones tomarán sus unidades. El juego continuará normalmente hasta que uno de los jugadores se rinda o quede sin unidades vivas en su tablero. Este caso será explicado con más detalle en las siguientes secciones.

Formato equipos

Lo primero que debe hacer tu programa es pedirle al usuario que seleccione un equipo. Cada grupo de test cases tiene un conjunto de equipos posibles diseñados para verificar el correcto funcionamiento de tu programa. Los equipos también se encuentran en `data.zip`. Por ejemplo, los equipos usados en los test cases `E1-BasicCombat` están en `data/E1-BasicCombat/`.

Para saber qué equipos se pueden utilizar, la clase `Game.cs` (que es la entrada a tu programa) recibe en su constructor el parámetro `teamsFolder`. Este parámetro contiene la ruta a la carpeta con todos los equipos disponibles según el test case que se esté ejecutando. Esta carpeta tendrá un archivo `.txt` por cada equipo posible para cada jugador. Los equipos deben ser mostrados al usuario y luego se le debe pedir como input que elija alguno de ellos. En el siguiente ejemplo, hay 6 equipos posibles y el usuario elige el equipo `0`:

```
Elige un archivo para cargar los equipos
0: 000.txt
1: 001.txt
2: 002.txt
3: 003.txt
4: 004.txt
5: 005.txt
INPUT: 0
```

Luego de elegido el equipo hay que leer el archivo y ver si el equipo es válido. Los archivos de equipos tienen el siguiente formato. La primera línea indica que comienza la sección del primer jugador, seguida inmediatamente de los nombres de sus unidades. Luego de las unidades del primer jugador habrá una línea indicado el inicio de la sección del segundo jugador, seguida por el nombre de las unidades de su equipo. Por ejemplo, este es un posible equipo:

```
Player 1 Team
[Samurai] Joker (Holy Wrath,Needle Shot)
Kabuso
Black Rider
Enku
Apsaras
Astaroth
Jueyuan
Player 2 Team
[Samurai] Nanashi
Zouchouten
Israfel
Patrimpas
Lorelei
Tam Lin
```

Los samurai siempre vienen precedidos por el indicador `[Samurai]`, que permite diferenciarlo del resto de unidades. En caso que un samurai tenga habilidades, estas se encuentran en la misma línea del nombre del samurai entre paréntesis. Si un samurai tiene más de una habilidad, estas se encontrarán dentro del paréntesis delimitadas por una coma sin espacio entre ellas. En el ejemplo anterior, Joker tiene las habilidades Holy Wrath y Needle Shot, mientras que Nanashi no tiene habilidades.

Formato Habilidades

La información de las habilidades del juego está en el archivo: `skills.json`. Por cada habilidad se indica su nombre, tipo, costo, skill power, objetivo, hits y efectos. Sin embargo, en este entrega solo usaremos el nombre de la habilidad para verificar si un equipo es válido. La lógica para implementar cada habilidad será agregada en las siguientes entregas.

```
[
  {...},
    {
        "name": "Mabufula",
        "type": "Ice",
        "cost": 20,
        "power": 120,
        "target": "All",
        "hits": "1",
        "effect": "Medium Ice attack. Target: All enemies"
    },
    {
        "name": "Mabufudyne",
        "type": "Ice",
        "cost": 32,
        "power": 180,
        "target": "All",
        "hits": "1",
        "effect": "Heavy Ice attack. Target: All enemies"
    },
  {...}
]
```

Formato Unidades

La información de las unidades del juego se encuentra en el archivo `json`. Los atributos de cada unidad te permitirán computar cuánto daño realizan al rival durante sus batallas y el orden en que actuarán (entre otras cosas).

Los datos correspondientes a los Samurai se encuentra en el archivo `samurai.json`, el cual cuenta con el siguiente formato:

```
[
  {...},
  {
        "name": "Demi-Fiend",
        "stats": {
            "HP": 1227,
            "MP": 834,
            "Str": 273,
            "Skl": 125,
            "Mag": 300,
            "Spd": 161,
            "Lck": 151
        },
        "affinity": {
            "Phys": "Nu",
            "Gun": "-",
            "Fire": "Nu",
            "Ice": "Nu",
            "Elec": "Nu",
            "Force": "Nu",
            "Light": "Nu",
            "Dark": "Nu"
        }
    },
    {
        "name": "Tadano",
        "stats": {
            "HP": 388,
            "MP": 193,
            "Str": 27,
            "Skl": 23,
            "Mag": 28,
            "Spd": 30,
            "Lck": 43
        },
        "affinity": {
            "Phys": "Rs",
            "Gun": "-",
            "Fire": "-",
            "Ice": "-",
            "Elec": "-",
            "Force": "-",
            "Light": "Nu",
            "Dark": "-"
        }
    },
  {...}
]
```

Por otro lado, los datos correspondientes a los monstruos se encuentra en el archivo `monsters.json`, el cual cuenta con el siguiente formato:

```
[
  {...},
  {
        "name": "Night Stalker",
        "stats": {
            "HP": 231,
            "MP": 103,
            "Str": 29,
            "Skl": 27,
            "Mag": 27,
            "Spd": 35,
            "Lck": 29
        },
        "affinity": {
            "Phys": "-",
            "Gun": "-",
            "Fire": "-",
            "Ice": "-",
            "Elec": "-",
            "Force": "-",
            "Light": "Wk",
            "Dark": "-"
        },
        "skills": [
            "Damascus Claw",
            "Dormina",
            "Life Bonus"
        ]
    },
    {
        "name": "Tattooed Man",
        "stats": {
            "HP": 290,
            "MP": 66,
            "Str": 35,
            "Skl": 31,
            "Mag": 24,
            "Spd": 28,
            "Lck": 23
        },
        "affinity": {
            "Phys": "-",
            "Gun": "-",
            "Fire": "-",
            "Ice": "-",
            "Elec": "-",
            "Force": "-",
            "Light": "Nu",
            "Dark": "Wk"
        },
        "skills": [
            "Taunt",
            "Fatal Sword",
            "Heat Wave",
            "Counter"
        ]
    },
  {...}
]
```

Como se puede ver, la unica diferencia entre estos archivos es que los monstruos tienen habilidades predefinidas.

Output del juego

El programa siempre comienza mostrando los equipos que se pueden elegir dentro de la carpeta `teamsFolder`. Luego de ello, si el equipo es inválido, se notifica al usuario y termina el programa:

```
Elige un archivo para cargar los equipos
0: 01.txt
1: 02.txt
2: 03.txt
3: 04.txt
INPUT: 3
Archivo de equipos no válido
```

En el caso contrario, comenzará una batalla entre los equipos del *Player 1* y del *Player 2*, donde siempre se comenzará con la ronda del *Player 1*.

Al iniciar la ronda del jugador, se mostrará un mensaje anunciándolo, el que incluye el nombre del samurai del jugador. El siguiente ejemplo muestra esto para el jugador 1, quien ha escogido al samurai Flynn.

```
----------------------------------------
Ronda de Flynn (J1)
```

Luego de esto, se mostrará el estado actual del tablero. En este se mostrarán siempre primero los puestos activos del jugador 1 seguido de los puestos activos del jugador 2. Solo para efectos del output, llamaremos a los puestos, de izquierda a derecha, como A, B, C y D respectivamente.

En orden, se mostrará el estado actual de cada puesto activo de izquierda a derecha, tengan estos unidades o no. En caso que tengan, se indicará su nombre, junto a su HP y MP; mientras que en caso de no tener, se mostrará el puesto vacío.

El siguiente ejemplo ilustra una situación donde ambos jugadores tienen el tablero lleno de unidades:

```
----------------------------------------
Ronda de Flynn (J1)
----------------------------------------
Equipo de Flynn (J1)
A-Flynn HP:971/971 MP:527/527
B-Dormarth HP:277/277 MP:269/269
C-Kabuso HP:137/137 MP:131/131
D-Inugami HP:214/214 MP:109/109
Equipo de Itsuki (J2)
A-Itsuki HP:1136/1136 MP:413/413
B-Ammut HP:709/709 MP:140/140
C-Wendigo HP:331/331 MP:66/66
D-Gremlin HP:140/140 MP:63/63
```

En este ejemplo, la linea `A-Flynn HP:971/971 MP:527/527` significa que en la posición A se encuentra la unidad de nombre Flynn, la cual actualmente tiene 971 HP de un máximo de 971 y tiene 527 MP de un máximo de 527.

Inmediatamente seguido del estado del tablero, se mostrará cuántos turnos tiene el jugador. En este ejemplo, el jugador tiene 4 **Full Turns** y ningún **Blinking Turn**:

```
----------------------------------------
Ronda de Flynn (J1)
----------------------------------------
Equipo de Flynn (J1)
A-Flynn HP:971/971 MP:527/527
B-Dormarth HP:277/277 MP:269/269
C-Kabuso HP:137/137 MP:131/131
D-Inugami HP:214/214 MP:109/109
Equipo de Itsuki (J2)
A-Itsuki HP:1136/1136 MP:413/413
B-Ammut HP:709/709 MP:140/140
C-Wendigo HP:331/331 MP:66/66
D-Gremlin HP:140/140 MP:63/63
----------------------------------------
Full Turns: 4
Blinking Turns: 0
```

Inmediatamente luego de esto, se mostrará el orden actual en el que actuará el equipo del jugador.

```
----------------------------------------
Ronda de Flynn (J1)
----------------------------------------
Equipo de Flynn (J1)
A-Flynn HP:971/971 MP:527/527
B-Dormarth HP:277/277 MP:269/269
C-Kabuso HP:137/137 MP:131/131
D-Inugami HP:214/214 MP:109/109
Equipo de Itsuki (J2)
A-Itsuki HP:1136/1136 MP:413/413
B-Ammut HP:709/709 MP:140/140
C-Wendigo HP:331/331 MP:66/66
D-Gremlin HP:140/140 MP:63/63
----------------------------------------
Full Turns: 4
Blinking Turns: 0
----------------------------------------
Orden:
1-Flynn
2-Dormarth
3-Inugami
4-Kabuso
```

Luego de mostrar el estado del tablero, los turnos del jugador y el orden en que actúan las unidades, se le pedirá al usuario que seleccione una acción. Las posibles acciones que puede seleccionar dependerán de cuál sea el tipo de unidad que está actuando. Si la unidad es un samurai, se desplegarán las siguientes opciones:

```
----------------------------------------
Seleccione una acción para Flynn
1: Atacar
2: Disparar
3: Usar Habilidad
4: Invocar
5: Pasar Turno
6: Rendirse
INPUT: 1
```

Mientras que si la unidad es un monstruo, se desplegarán las siguientes opciones:

```
----------------------------------------
Seleccione una acción para Dormarth
1: Atacar
2: Usar Habilidad
3: Invocar
4: Pasar Turno
INPUT: 1
```

Al seleccionar la opción *Atacar*, se mostrarán las distintas unidades que el usuario tiene como objetivos. Las unidades se mostrarán en el orden en el cual se encuentran en el tablero del oponente:

```
----------------------------------------
Ronda de Flynn (J1)
----------------------------------------
Equipo de Flynn (J1)
A-Flynn HP:971/971 MP:527/527
B-Dormarth HP:277/277 MP:269/269
C-Kabuso HP:137/137 MP:131/131
D-Inugami HP:214/214 MP:109/109
Equipo de Itsuki (J2)
A-Itsuki HP:1136/1136 MP:413/413
B-Ammut HP:709/709 MP:140/140
C-Wendigo HP:331/331 MP:66/66
D-Gremlin HP:140/140 MP:63/63
----------------------------------------
Full Turns: 4
Blinking Turns: 0
----------------------------------------
Orden:
1-Flynn
2-Dormarth
3-Inugami
4-Kabuso
----------------------------------------
Seleccione una acción para Flynn
1: Atacar
2: Disparar
3: Usar Habilidad
4: Invocar
5: Pasar Turno
6: Rendirse
INPUT: 1
----------------------------------------
Seleccione un objetivo para Flynn
1-Itsuki HP:1136/1136 MP:413/413
2-Ammut HP:709/709 MP:140/140
3-Wendigo HP:331/331 MP:66/66
4-Gremlin HP:140/140 MP:63/63
5-Cancelar
INPUT: 4
```

Vale la pena destacar, que a diferencia de las opciones de acciones para las unidades, las cuales están separadas del número correspondiente a cada acción con ": " (dos puntos y un espacio), las selecciones para atacar están separadas por un guión. Además, cada opción contiene el nombre de la unidad, su HP y su MP.

Luego de seleccionar la opción, se mostrará el resultado del ataque y el resultado sobre los turnos del jugador:

```
----------------------------------------
Ronda de Flynn (J1)
----------------------------------------
Equipo de Flynn (J1)
A-Flynn HP:971/971 MP:527/527
B-Dormarth HP:277/277 MP:269/269
C-Kabuso HP:137/137 MP:131/131
D-Inugami HP:214/214 MP:109/109
Equipo de Itsuki (J2)
A-Itsuki HP:1136/1136 MP:413/413
B-Ammut HP:709/709 MP:140/140
C-Wendigo HP:331/331 MP:66/66
D-Gremlin HP:140/140 MP:63/63
----------------------------------------
Full Turns: 4
Blinking Turns: 0
----------------------------------------
Orden:
1-Flynn
2-Dormarth
3-Inugami
4-Kabuso
----------------------------------------
Seleccione una acción para Flynn
1: Atacar
2: Disparar
3: Usar Habilidad
4: Invocar
5: Pasar Turno
6: Rendirse
INPUT: 1
----------------------------------------
Seleccione un objetivo para Flynn
1-Itsuki HP:1136/1136 MP:413/413
2-Ammut HP:709/709 MP:140/140
3-Wendigo HP:331/331 MP:66/66
4-Gremlin HP:140/140 MP:63/63
5-Cancelar
INPUT: 4
----------------------------------------
Seleccione un objetivo para Flynn
1-Flynn HP:971/971 MP:527/527
2-Melchom HP:5/71 MP:67/67
3-Dybbuk HP:66/66 MP:63/63
4-Cancelar
INPUT: 2
----------------------------------------
Flynn ataca a Gremlin
Gremlin recibe 88 de daño
Gremlin termina con HP:52/140
----------------------------------------
Se han consumido 1 Full Turn(s) y 0 Blinking Turn(s)
Se han obtenido 0 Blinking Turn(s)
```

Por otro lado, la acción de *Disparar* se anuncia con el mismo formato que la acción anterior, pero se explicitará que la unidad usó un disparo:

```
----------------------------------------
Seleccione un objetivo para Flynn
1-Itsuki HP:1136/1136 MP:413/413
2-Ammut HP:709/709 MP:140/140
3-Wendigo HP:331/331 MP:66/66
4-Gremlin HP:140/140 MP:63/63
5-Cancelar
INPUT: 4
----------------------------------------
Flynn dispara a Gremlin
Gremlin recibe 110 de daño
Gremlin termina con HP:30/140
----------------------------------------
Se han consumido 1 Full Turn(s) y 0 Blinking Turn(s)
Se han obtenido 0 Blinking Turn(s)
----------------------------------------
```

Como podemos ver, al usuario se le provee la opción `Cancelar`. Si el usuario selecciona esta opción, se deberá desplegar nuevamente el menú con las acciones disponibles:

```
----------------------------------------
Seleccione una acción para Flynn
1: Atacar
2: Disparar
3: Usar Habilidad
4: Invocar
5: Pasar Turno
6: Rendirse
INPUT: 1
----------------------------------------
Seleccione un objetivo para Flynn
1-Itsuki HP:1136/1136 MP:413/413
2-Ammut HP:709/709 MP:140/140
3-Wendigo HP:331/331 MP:66/66
4-Gremlin HP:140/140 MP:63/63
5-Cancelar
INPUT: 5
----------------------------------------
Seleccione una acción para Flynn
1: Atacar
2: Disparar
3: Usar Habilidad
4: Invocar
5: Pasar Turno
6: Rendirse
INPUT: 1
```

Otra de las opciones disponibles es *Usar Habilidad*. Si bien, en esta entrega no es necesario que implementen las habilidades, deben implementar la funcionalidad de mostrar las habilidades disponibles. Solo se deben mostrar las habilidades que la unidad efectivamente puede utilizar, es decir, aquellas cuyo costo sea menor o igual al MP actual de la unidad. Podría suceder que una unidad tenga solo habilidades que no puede utilizar, caso en que solo se desplegará la opción `Cancelar`.

Para efectos de esta entrega, el jugador siempre escogerá la opción `Cancelar` luego de escoger utilizar una habilidad. El formato en que esto se realizará será el siguiente:

```
----------------------------------------
Seleccione una acción para Itsuki
1: Atacar
2: Disparar
3: Usar Habilidad
4: Invocar
5: Pasar Turno
6: Rendirse
INPUT: 3
----------------------------------------
Seleccione una habilidad para que Itsuki use
1-Agi MP:5
2-Cancelar
INPUT: 2
```

Una vez terminado el turno de una unidad, si el jugador aún tiene turnos disponibles, comenzará el turno de la siguiente unidad en el orden de acción. Al igual que en el caso anterior, se anunciará el estado actual del tablero, la cantidad de turnos disponibles, el nuevo orden de acciones y el input del usuario. Lo único que no se mostrará será el mensaje de inicio de ronda, ya que no ha cambiado la ronda del juego.

```
----------------------------------------
Equipo de Flynn (J1)
A-Flynn HP:971/971 MP:527/527
B-Dormarth HP:277/277 MP:269/269
C-Kabuso HP:137/137 MP:131/131
D-Inugami HP:214/214 MP:109/109
Equipo de Itsuki (J2)
A-Itsuki HP:1136/1136 MP:413/413
B-Ammut HP:709/709 MP:140/140
C-Wendigo HP:331/331 MP:66/66
D-Gremlin HP:52/140 MP:63/63
----------------------------------------
Full Turns: 3
Blinking Turns: 0
----------------------------------------
Orden:
1-Dormarth
2-Inugami
3-Kabuso
4-Flynn
----------------------------------------
Seleccione una acción para Dormarth
1: Atacar
2: Usar Habilidad
3: Invocar
4: Pasar Turno
INPUT: 1
```

Existe la posibilidad de que un espacio del tablero se encuentre vacío, sea porque un monstruo ha muerto o porque no se escogieron suficientes unidades al inicio del juego. En estos casos, los espacios se mostrarán vacíos. Es importante señalar que cuando un samurai muere, este no deja el tablero, por lo que sigue apareciendo en el anuncio del estado del tablero. Podemos ver el caso de espacios vacíos y samurai muertos en el siguiente ejemplo:

```
----------------------------------------
Equipo de Flynn (J1)
A-Flynn HP:917/971 MP:527/527
B-Xi Wangmu HP:269/418 MP:405/405
C-Yurlungur HP:107/202 MP:196/196
D-
Equipo de Itsuki (J2)
A-Itsuki HP:0/1136 MP:413/413
B-Ancient of Days HP:375/498 MP:484/484
C-
D-Anubis HP:453/497 MP:223/223
```

Una vez el jugador quede sin turnos, comenzará la ronda del otro jugador. Esta sucederá de la misma manera y se anunciará con las mismas reglas que la ronda del otro jugador, siendo la única diferencia que el mensaje de inicio de ronda mostrará los datos del otro jugador y el orden de las acciones corresponderá al del otro equipo.

Si en cualquier punto del juego un equipo queda sin unidades vivas en el tablero, entonces el juego terminará. En este caso se anunciará al ganador y el programa terminará su ejecución. El siguiente ejemplo muestra este caso:

```
----------------------------------------
Seleccione una acción para Ancient of Days
1: Atacar
2: Usar Habilidad
3: Invocar
4: Pasar Turno
INPUT: 1
----------------------------------------
Seleccione un objetivo para Ancient of Days
1-Flynn HP:49/971 MP:527/527
2-Cancelar
INPUT: 1
----------------------------------------
Ancient of Days ataca a Flynn
Flynn recibe 54 de daño
Flynn termina con HP:0/971
----------------------------------------
Se han consumido 1 Full Turn(s) y 0 Blinking Turn(s)
Se han obtenido 0 Blinking Turn(s)
----------------------------------------
Ganador: Itsuki (J2)
```

Finalmente, el usuario también puede seleccionar la acción `Rendirse`. Si selecciona esta opción, se anunciará que el jugador se rindió, quién fue el ganador del juego y, finalmente, el programa terminará de ejecutarse. Sin embargo, no se mostrará el consumo de turnos:

```
----------------------------------------
Seleccione una acción para Flynn
1: Atacar
2: Disparar
3: Usar Habilidad
4: Invocar
5: Pasar Turno
6: Rendirse
INPUT: 6
----------------------------------------
Flynn (J1) se rinde
----------------------------------------
Ganador: Itsuki (J2)
```

Cálculo de daño

Tal como indica el enunciado general del proyecto, los cálculos de daño pueden generar números decimales. Cuando ello ocurre, hay que truncar el número a su entero más bajo. Esto se puede realizar en C# utilizando la función `Math.Floor(...)`. Luego el resultado puede ser convertido a entero con `Convert.ToInt32(...)`.

Input-Output

En tu proyecto **NO** debes usar `Console.WriteLine(...)` ni `Console.ReadLine()` para mostrar y pedir texto al usuario. Esto se debe a que nuestro código para comparar la salida de tu programa con los test cases ignora los mensajes mandados directamente a consola.

Para que el input-output de tu programa sea verificado por nuestros test cases debes usar el objeto `view` que te entregamos en el constructor de `Game.cs`. Ese objeto tiene los siguientes métodos:

- `ReadLine()`: Solicita un string al usuario y retorna el `string` correspondiente.
- `WriteLine(string message)`: Muestra `message` en consola.

El objeto `view` hace dos cosas. Por un lado, guarda los mensajes que se escriben mediante su método `WriteLine(...)`. Esos mensajes son comparados con el test case para verificar si tu programa está correcto. Por otro lado, cuando se llama a `ReadLine()` automáticamente retorna el `INPUT:` indicado en el test case.

En resumen, todo input pedido y mensaje mostrado mediante `Console` es ignorado al momento de evaluar los test cases. Si quieres que un input o texto sea considerado debes utilizar los métodos de `view`.

Rúbrica

Para evaluar tu entrega usaremos 3 grupos de test cases: `E1-Random`, `E1-InvalidTeams` y `E1-BasicCombat`.

Esta entrega tiene puntaje por funcionalidad y por limpieza de código. Para calcular tu puntaje de funcionalidad se le asignará a cada grupo de test un puntaje máximo, el cual será el limite superior del puntaje que obtendrás en dicho grupo de tests; el puntaje que obtengas variará proporcionalmente a la cantidad de tests del grupo que pases. Los puntajes se distribuyen de la siguiente manera:

- [**0.7 puntos**] Porcentaje de test cases pasados en `E1-InvalidTeams`.
- [**3.0 puntos**] Porcentaje de test cases pasados en `E1-BasicCombat`.
- [**2.3 puntos**] Pasar todos los test cases en `E1-Random`.

Por ejemplo, digamos que tu entrega todos los test cases `E1-InvalidTeams` , todos los test cases `E1-Random` y el 80 % de los test cases `E1-BasicCombat`. Entonces tu puntaje de funcionalidad será: $0{,}7+2{,}3+3{,}0\cdot 0{,}8 = 5{,}4$.

Por otro lado, el puntaje por limpieza de código es en base a descuentos. Es decir, se parte con 6 puntos y se descuenta en base a las violaciones de los principios de los capítulos de Clean Code que presente tu código. Los descuentos máximos por capítulo son los siguientes:

- [**-2.0 puntos**] **No** sigue los principios del cap. 2 de *clean code*.
- [**-2.5 puntos**] **No** sigue los principios del cap. 3 de *clean code*.

Finalmente, tu nota final será igual al promedio geométrico entre el puntaje por funcionalidad y el puntaje por limpieza de código (más el punto base), donde el promedio geométrico entre x e y es igual a $\sqrt{xy}$.

Por ejemplo, si tienes 3 puntos por funcionalidad y 5 puntos por limpieza de código entonces tu nota será $\sqrt{3 \cdot 5} + 1 = 4{,}9$. Pero si tienes 6 puntos en funcionalidad y 1.5 en limpieza de código entonces tu nota será $\sqrt{6 \cdot 1{,}5} + 1 = 4{,}0$.

Importante: No está permitido modificar los test cases ni el proyecto `Shin-Megami-Tensei.Tests`. Hacerlo puede conllevar una penalización que dependerá de la gravedad de la situación