

75.10 – Técnicas de Diseño
Trabajo Práctico – Nikoli Games
Actividad de Entrega 1

Objetivo

Se solicita incorporar al motor de generación de juegos el Juego **Inshi no Heya** y la funcionalidad de leer jugadas de un archivo y ejecutarlas secuencialmente.

Funcionalidad a Implementar Actividad 1

- Se deberá incorporar al motor el siguiente juego: **Inshi no Heya**
- Leer el tablero inicial desde un archivo json (leer aclaraciones)
- Leer una serie de jugadas del juego desde un json (leer aclaraciones) ,
- Procesar las jugadas y retornar el estado final del tablero en un formato json (leer aclaraciones).

Aclaraciones

- **Se deberá resolver esta actividad durante la presentación de la iteración 1**
- La lectura del archivo no requiere una interfaz de usuario,
- Verificar el funcionamiento con JUnit.
- Las jugadas invalidas no se ejecutan.
- Dichas nuevas funcionalidades deberán incorporarse y terminarse completamente para la segunda iteración
- Ver formato de archivo y reglas del juego debajo.

Especificaciones:

INSHI NO HEYA	REGLAS																																																												
<div><table><tr><td>6</td><td>4</td><td></td><td>5</td><td>40</td></tr><tr><td></td><td>3</td><td></td><td>4</td><td></td></tr><tr><td></td><td>15</td><td>40</td><td></td><td></td></tr><tr><td>4</td><td>10</td><td></td><td>6</td><td></td></tr><tr><td></td><td></td><td></td><td>3</td><td></td></tr></table></div> <div><table><tr><td>6</td><td>4</td><td>1</td><td>5</td><td>40</td></tr><tr><td>3</td><td>4</td><td>1</td><td>5</td><td>2</td></tr><tr><td>2</td><td>3</td><td>1</td><td>3</td><td>4</td></tr><tr><td>5</td><td>3</td><td>15</td><td>40</td><td>2</td></tr><tr><td>4</td><td>10</td><td>5</td><td>4</td><td>6</td></tr><tr><td>1</td><td>5</td><td>4</td><td>2</td><td>3</td></tr><tr><td>4</td><td>2</td><td>5</td><td>3</td><td>1</td></tr></table></div>	6	4		5	40		3		4			15	40			4	10		6					3		6	4	1	5	40	3	4	1	5	2	2	3	1	3	4	5	3	15	40	2	4	10	5	4	6	1	5	4	2	3	4	2	5	3	1	<ul style="list-style-type: none">La multiplicación de los números de cada “room” debe ser igual al pequeño número en la esquina superior izquierda de cada “room”Un número no puede repetirse en una misma fila o columna
6	4		5	40																																																									
	3		4																																																										
	15	40																																																											
4	10		6																																																										
			3																																																										
6	4	1	5	40																																																									
3	4	1	5	2																																																									
2	3	1	3	4																																																									
5	3	15	40	2																																																									
4	10	5	4	6																																																									
1	5	4	2	3																																																									
4	2	5	3	1																																																									
<div>Archivo de entrada:</div> <pre>{ "plays": [{ "number": 1, "position": [1, 1], "value": "3" }, { "number": 2, "position": [1, 2], "value": "4" }, { "number": 3, "position": [1, 3], "value": "1" }, { "number": 4, "position": [1, 3], "value": "5" }]}</pre>	<div>Archivo de salida:</div> <pre>{ "plays": [{ "number": 1, "boardStatus": "valid" }, { "number": 2, "boardStatus": "valid" }, { "number": 3, "boardStatus": "valid" }, { "number": 4, "boardStatus": "invalid" }], "board": { "status": "invalid", "values": [{ "position": [1, 1], "value": "3" }, { "position": [1, 2],</pre>																																																												

	<pre> "value": "4" } ...] } </pre>
--	---

Tablero inicial:

```

{
  "board": {
    "values": [{
      "position": [1, 1],
      "value": "6",
      "readonly": true
    }, {
      "position": [1, 2],
      "value": "4",
      "readonly": true
    }, {
      "position": [1, 4],
      "value": "5",
      "readonly": true
    }, {
      "position": [1, 5],
      "value": "40",
      "readonly": true
    }, {
      "position": [2, 2],
      "value": "3",
      "readonly": true
    }, {
      "position": [2, 4],
      "value": "4",
      "readonly": true
    }, {
      "position": [3, 2],
      "value": "15",
      "readonly": true
    }, {
      "position": [3, 3],
      "value": "40",
      "readonly": true
    }, {
      "position": [4, 1],
      "value": "4",
      "readonly": true
    }, {
      "position": [4, 2],
      "value": "10",
      "readonly": true
    }, {
      "position": [4, 4],
      "value": "6",
      "readonly": true
    }, {
      "position": [5, 4],
      "value": "3",
      "readonly": true
    }
  ]
}

```

Referencias útiles:

Para manejo de JSON se puede utilizar la siguiente librería: Gson

<https://google.github.io/gson/apidocs/com/google/gson/Gson.html>

Configuración vía Gradle:

compile group: 'com.google.code.gson', name: 'gson', version: '2.3.1'

Ejemplo de uso de Gson con una clase cualquiera:

```
Gson gson = new Gson(); // Or use new GsonBuilder().create();
```

```
MyType target = new MyType();
```

```
String json = gson.toJson(target); // serializes target to Json
```

```
MyType target2 = gson.fromJson(json, MyType.class); // deserializes json into target2
```