****

Módulo Profesional 06: Acceso a datos **UF1-ACT2**

**Mezclador**

**Revisado 7/11/2024**

**CICLO FORMATIVO DE GRADO SUPERIOR EN**

**Videojuegos y ocio digital**

**MODALIDAD PRESENCIAL**

**Nombre y apellidos del alumno**



**Introducción**



En esta actividad crearás un editor que te permitirá grabar y cargar puzzles para un minijuego.

|  |
| --- |
| **Objetivos** |
| Almacenar y recuperar datos en/desde ficheros binarios.  Almacenar y recuperar datos en/desde ficheros de texto usando serializadores automáticos |
| **Metodología de evaluación** |
| 40% Ejercicio1  40% Ejercicio2  20% Ejercicio3  En cada ejercicio se valorará   * 50% Metodología * 50% Funcionamiento. |
| **Entrega** |
| Entregar archivo comprimido con nombre UF1-ACT2-NombreApellidos.zip incluyendo enunciado cumplimentado, programas creados y vídeo explicativo. |

**Situación**

Disponemos de un minijuego tipo puzzle que actualmente genera puzzles de forma aleatoria, llamados patrones.

Queremos flexibilizar este mini juego permitiendo que:

* Disponga de una carpeta con diferentes patrones almacenados en ficheros binarios.
* Esos puzzles se puedan editar mediante una aplicación separada.
* Al arrancar, el juego cargue una de los puzzles almacenados de forma aleatoria.

**Crear el editor**

El editor, además de modificar los contenidos de la rejilla 8x8 que forma el patrón permitirá las siguientes operaciones:

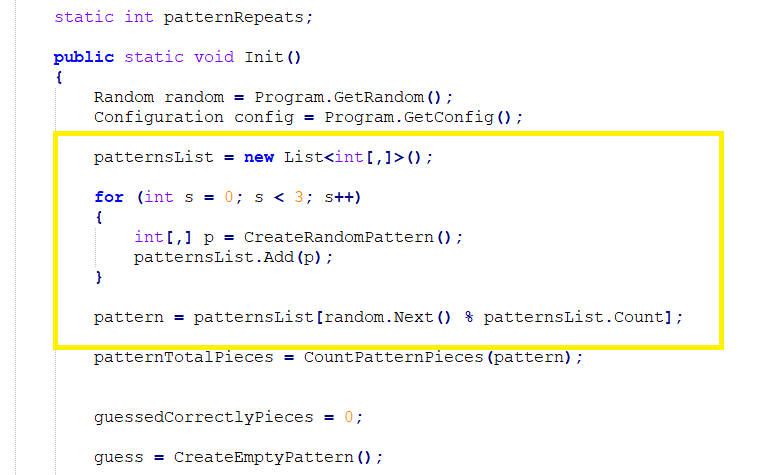
|  |  |
| --- | --- |
| **Operación** | **Descripción** |
| Nuevo | Reinicia el patrón |
| Cargar | Carga el patrón desde un fichero que el game designer elige |
| Guardar | Graba el patrón en un fichero, si no se había cargado el game designer podrá elegir el nombre y la ubicación, si se había cargado antes se sobreescribe el anterior |
| Guardar como | Graba el patrón en un fichero. El game designer podrá elegir el nombre y la ubicación |
| Salir | Cierra la aplicación |

* *Podéis formatear los datos dentro del fichero como creáis conveniente*

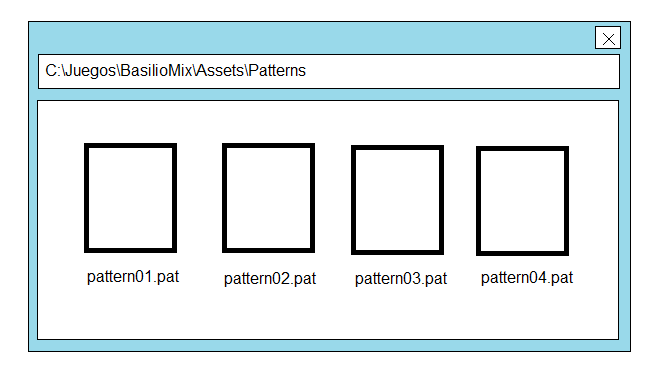
**Ejercicio1:** Crea el editor.

**Cargar los niveles en el minijuego**

Actualmente el minijuego crea un patrón aleatoriamente en las siguientes líneas del archivo **Board.cs**.

****

Realizaremos las modificaciones necesarias para que el juego, en vez de mantener ese comportamiento, cargue aleatoriamente uno de los patrones almacenados, que podemos situar en una carpeta dentro de “Assets”.

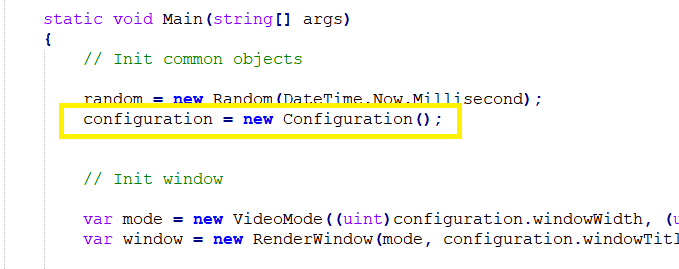


* *También podéis mantener el comportamiento anterior y que éste se active sólo en caso de que no exista ningún fichero de patrones.*

**Ejercicio2:** Realiza las modificaciones requeridas en el juego.

**Leer y escribir la configuración en un fichero**

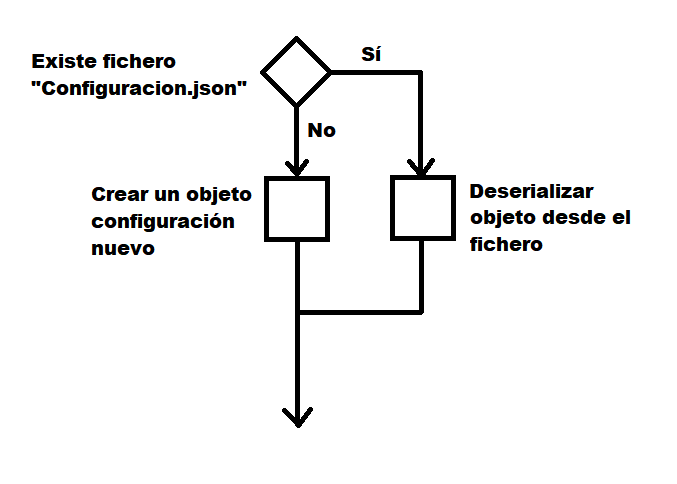
Actualmente la configuración del programa se almacena en un objeto que se crea en la inicialización.



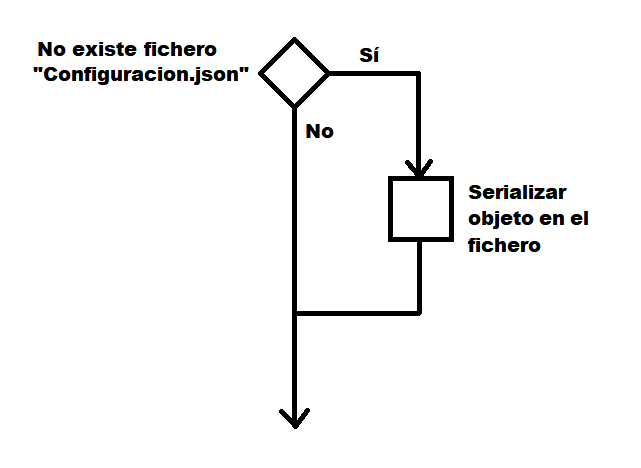
Queremos almacenar esta configuración en un fichero llamado “Configuracion.json” de forma que sea fácilmente editable.



Para ello, en el inicio del programa, antes de crear la ventana, realizaremos las siguientes acciones.



Al final del programa, después de cerrar la ventana, realizaremos las siguientes acciones.



Tanto la serialización como la deserialización la realizaremos de forma automática usando las clases disponibles en las librerías de soporte.

**Ejercicio3:** Realiza las modificaciones indicadas y graba un vídeo explicativo (2-3 minutos) explicando el funcionamiento del programa y el código introducido en los ejercicios. Asegúrate de mostrar que modificando el fichero de configuración se producen cambios en el juego. Si alojas el vídeo en un servicio externo, incluye el enlace a continuación.

|  |
| --- |
|  |