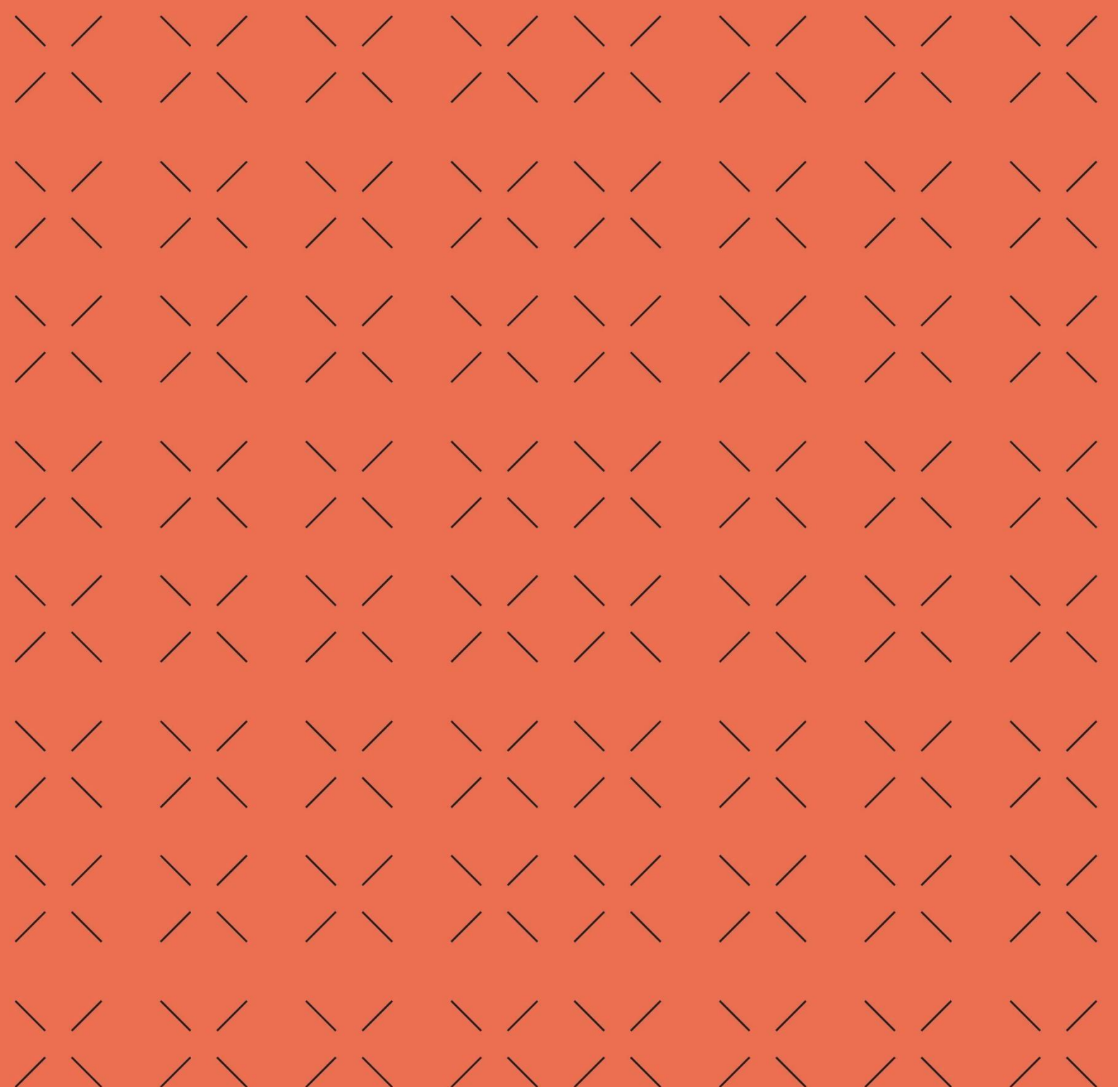


DESARROLLO DE INTERFACES

Actividad Evaluable 1ª Evaluación



Actividad Evaluable 1ª Evaluación: Buscaminas

La tarea consiste en implementar el juego del Buscaminas.

En caso de no conocer el juego, a continuación tenéis una pequeña explicación.

La interfaz del juego original es la siguiente:

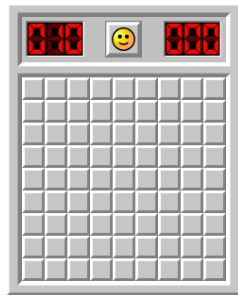


Ilustración 1. Interfaz del juego buscaminas

El juego es muy básico. Consiste en un tablero de 9 x 9, en su versión de dificultad fácil, en el que hay colocadas 10 minas. El juego consiste en hacer *click* en las casillas sin hacer estallar las minas. Cuando pulsamos en una casilla vacía, esta se descubre. Si hay minas cercanas, se pinta un número con la cantidad de minas adyacentes a esta casilla. En el caso de no haber ninguna mina adyacente, se descubren todas las casillas de este tipo, es decir, que no tengan ninguna casilla adyacente. Un ejemplo de esto lo podéis ver en la siguiente imagen:



Ilustración 2. Ejemplo de casilla descubierta

Como podéis observar en la ilustración 2, se han descubierto las casillas adyacentes a la que se ha hecho *click* en el caso de que no haya minas a su alrededor.

Objetivos de la actividad

El objetivo principal de la actividad es que hagáis vosotros este juego, en su versión fácil. Es decir, se pide programar un tablero de 9 x 9 casillas en el que incluyáis 10 minas.

Requisitos

Los requisitos para la evaluación de la actividad son:

- La **interfaz** debe crearse con **WPF**, ya que es la **tecnología** que estamos utilizando en la **asignatura**.
- **Diseñar una interfaz** (no tiene por qué ser igual que el juego clásico) en la que se **incluya** un **botón** de **reinicio** (en el juego clásico es el emoji sonriente) y un **marcador** de las **minas activas**.
- Diseñar la **lógica** del **juego** original, es decir, que las **casillas adyacentes** se **abran** al **pulsar** sobre una **casilla** que **no** tenga **minas adyacentes** cercanas y que se **pinten** los **números** con **minas adyacentes** cercanas.
- Utilizad **colores distintos** para la **cantidad** de **minas cercanas**, para una mejor visualización del juego.
- Incorporad que cuando se **utilice** el **botón secundario** del **ratón**, se pueda **marcar** que en esa **posición** hay una **mina**. Debéis utilizar un **icono** para que el **usuario** pueda **ver** que ha establecido un **marcador** de **mina**. El **marcador** de **minas restantes** debe **disminuir** si **marcamos** que en una **casilla** hay una **mina**.

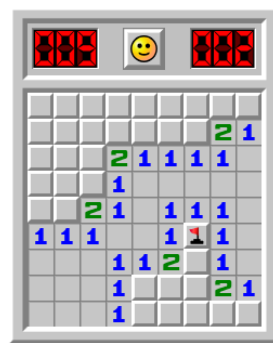


Ilustración 3. Ejemplo de marcado de minas

Como podéis observar en la imagen anterior, el contador ha pasado de 10 minas a 9 cuando hemos marcado donde creemos que hay una mina.

- Tenéis que hacer el **final** de **partida** como vemos en la imagen. Es decir, cuando el **jugador pincha** sobre una **mina**, que se **muestre** la **posición** de las **minas**. Se **valorará** también que se **marque** la **casilla** donde el **jugador** ha **pinchado** sobre la mina en **otro color**. También que el **icono** de **reinicio** **cambie**.



Ilustración 4. Ejemplo fin de partida

- Mostrad en una **ventana de diálogo** un **mensaje** de **victoria** o **derrota**.

A continuación, tenéis unas ampliaciones opcionales para subir nota en la actividad.

- **Opcional: contador de tiempo de partida.**
- **Opcional: mostrar al final en una ventana de diálogo el número de clics en la partida del usuario, el tiempo destinado y si ha ganado o ha perdido el juego.**
- **Opcional: selector para variar el número de casillas y/o de minas.**

Entregables

1. **Código fuente**
 - ✓ El código fuente de la aplicación, organizado de manera clara y comentado (haz un fichero comprimido con la carpeta del proyecto).
2. **Documentación**
 - ✓ Un documento explicando el **funcionamiento** y **diseño** de la aplicación, incluyendo capturas de pantalla.
 - ✓ Explicación de las **elecciones** que has tomado a la hora de **escoger componentes**, o cualquier tipo de **decisión importante** sobre la aplicación.
3. **Posibles mejoras**
 - ✓ Una **sección** que indique **posibles mejoras** o extensiones que podrían implementarse en el futuro.

Criterios de evaluación

Los criterios de evaluación, que se tendrán en cuenta para evaluar la actividad son:

- ✓ Cumplimiento de los requisitos funcionales y técnicos.
- ✓ Usabilidad y experiencia de usuario.
- ✓ Organización del código y uso adecuado de clases y métodos.
- ✓ Implementación de eventos y acciones.
- ✓ Correcto manejo de datos y enlace a componentes.

También se valorará que el alumno haya entregado todos los documentos que se solicitan.

Plazo de entrega

La entrega se debe realizar antes de las 23:59 del domingo 24 de noviembre de 2024.

Plagio

Esta actividad es individual. En caso de sospecha de originalidad en la realización, se requerirá una entrevista oral. Es posible incluso que se pida una entrevista aún sin sospechas de plagio.

Instrucciones de entrega

Se abrirá una tarea en Aules, en una sección llamada Actividad Evaluable 1ª Evaluación. Se debe entregar todo lo solicitado en un fichero comprimido.

Calificación

La entrega es obligatoria, con una nota mínima de 4 puntos para poder superar la actividad.