# Лабораторна робота №1.

**Об’єктно-орієнтоване програмування та SOLID.**

***Мета роботи:*** повторення принципів ООП та вивчення принципів SOLID

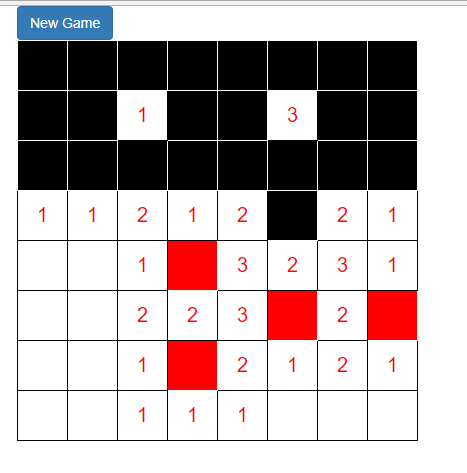


Рис.1. – Гра сапер.

Код написаний з дотриманням принципів SOLID.

Приклад коду.

import { Component, Injectable } from '@angular/core';

import { MineField } from '../Minner/MineField';

import { IField } from '../Minner/Interface/IField';

@Injectable()

export class MineFieldInterface implements IField<MineField> {

public Click(Field: MineField, n: number, m: number) {

if (Field.Field[n][m] == 1) {

Field.Field[n][m] = 5;

Field.GameState = false;

}

if (Field.Field[n][m] == 0) {

Field.Field[n][m] = 4;

if (Field.MineAround[n][m] == 0) {

this.ClearEmptyCell(Field ,n ,m);

}

}

}

private ClearEmptyCell(Field: MineField, n: number, m: number) {

for (var i = n - 1; i <= n + 1; i++) {

for (var j = m - 1; j <= m + 1; j++) {

if (i >= 0 && j >= 0 && i < Field.Height && j < Field.Width) {

if (Field.Field[i][j] == 0) {

Field.Field[i][j] = 4;

if (Field.MineAround[i][j] == 0)

this.ClearEmptyCell(Field, i, j);

}

}

}

}

}

public SetFlag(Field: MineField, n: number, m: number) {

if (Field.Field[n][m] == 0) {

if (Field.MaxFlags > Field.Flags) {

Field.Field[n][m] = 2;

Field.Flags++;

}

}

else

if (Field.Field[n][m] == 2) {

Field.Field[n][m] = 0;

Field.Flags--;

}

if (Field.Field[n][m] == 1) {

if (Field.MaxFlags > Field.Flags) {

Field.Field[n][m] = 3;

Field.Flags++;

}

}

else

if (Field.Field[n][m] == 3) {

Field.Field[n][m] = 1;

Field.Flags--;

}

}

public GetCode(Field: MineField, n: number, m: number): number {

var index = 0;

if (Field.Field[n][m] == 2 || Field.Field[n][m] == 3)

index = 1;

if (Field.Field[n][m] == 4)

index = 2;

if (Field.Field[n][m] == 5)

index = 3;

return index;

}

public MineAround(Field: MineField, n: number, m: number): string {

var index = 0;

if (Field.Field[n][m] == 4) {

index = Field.MineAround[n][m];

}

return index == 0 ? "" : index.toString();

}

}