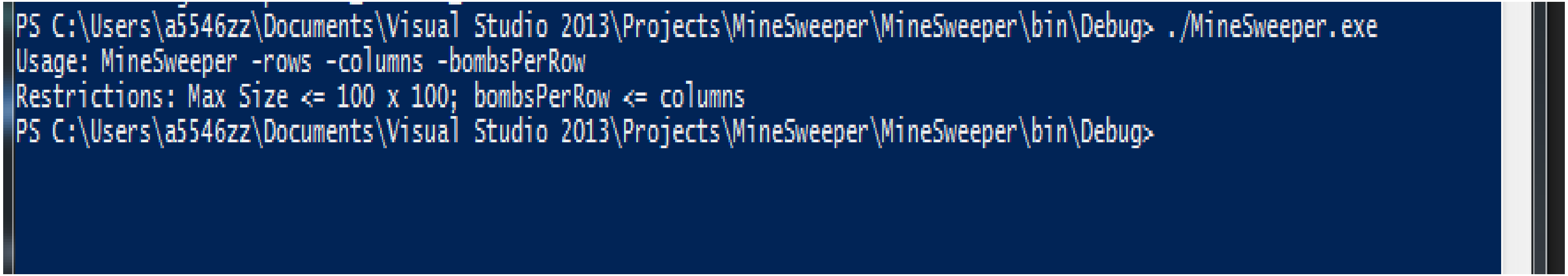
MineSweeper

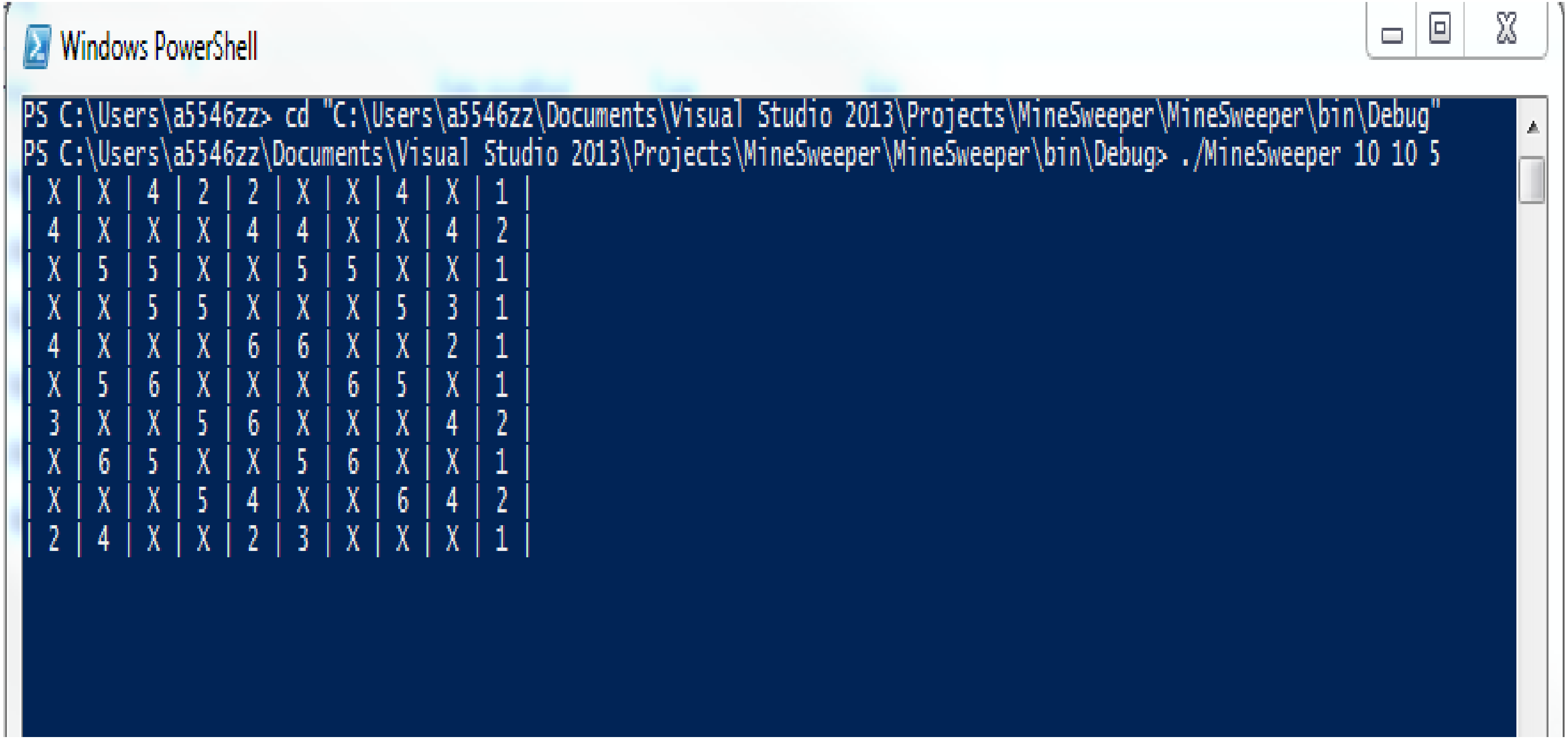
To use the command-line application, navigate to the bin\Debug (or bin\Release) directory and call the binary passing parameters for the number of rows, columns, and boms-per-row.

If any of the paramters is incorrect, a usage message will display and the program will terminate. The following restrictions apply, presently.

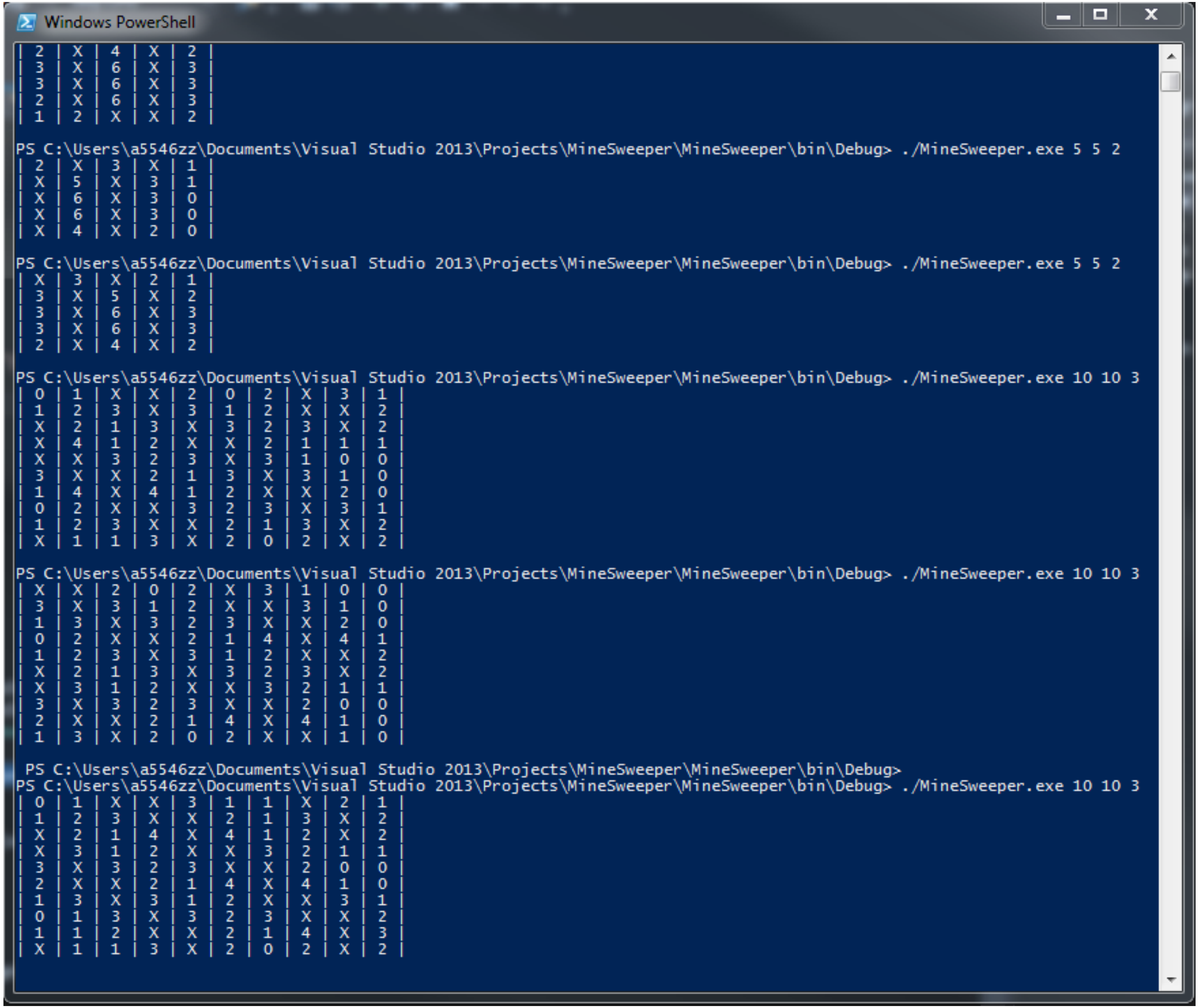
- All three parameters must be valid.
- Rows and columns are restricted to the range { 1:100 }
- The value of bombsPerRow paramger is limited to the range { 0 : number_of_rows }



```
PS C:\Users\a5546zz\Documents\Visual Studio 2013\Projects\MineSweeper\MineSweeper\bin\Debug> ./MineSweeper.exe
Usage: MineSweeper -rows -columns -bombsPerRow
Restrictions: Max Size <= 100 x 100; bombsPerRow <= columns
PS C:\Users\a5546zz\Documents\Visual Studio 2013\Projects\MineSweeper\MineSweeper\bin\Debug>
```



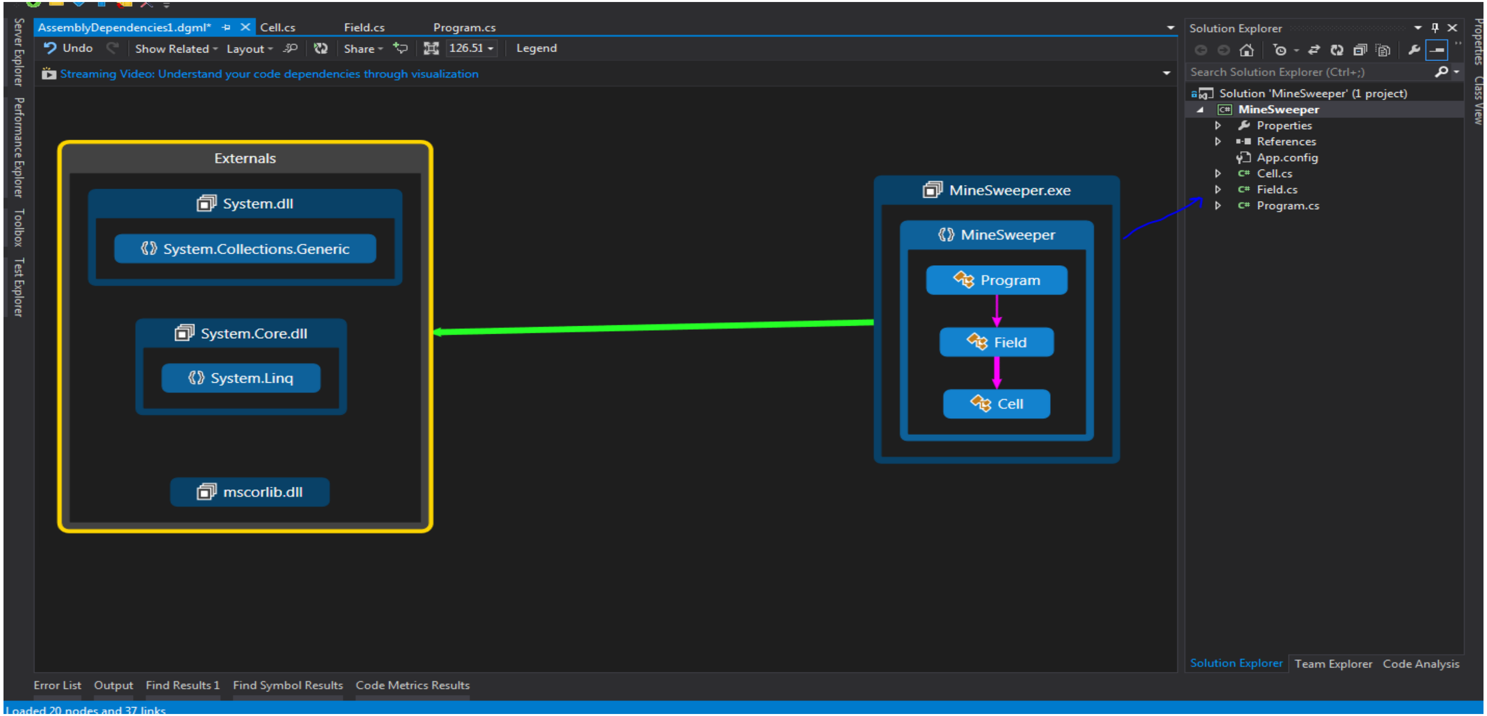
```
PS C:\Users\a5546zz> cd "C:\Users\a5546zz\Documents\Visual Studio 2013\Projects\MineSweeper\MineSweeper\bin\Debug"
PS C:\Users\a5546zz\Documents\Visual Studio 2013\Projects\MineSweeper\MineSweeper\bin\Debug> ./MineSweeper 10 10 5
| X | X | 4 | 2 | 2 | X | X | 4 | X | 1 |
| 4 | X | X | X | 4 | 4 | X | X | 4 | 2 |
| X | 5 | 5 | X | X | 5 | 5 | X | X | 1 |
| X | X | 5 | 5 | X | X | X | 5 | 3 | 1 |
| 4 | X | X | X | 6 | 6 | X | X | 2 | 1 |
| X | 5 | 6 | X | X | X | 6 | 5 | X | 1 |
| 3 | X | X | 5 | 6 | X | X | X | 4 | 2 |
| X | 6 | 5 | X | X | 5 | 6 | X | X | 1 |
| X | X | X | 5 | 4 | X | X | 6 | 4 | 2 |
| 2 | 4 | X | X | 2 | 3 | X | X | X | 1 |
```



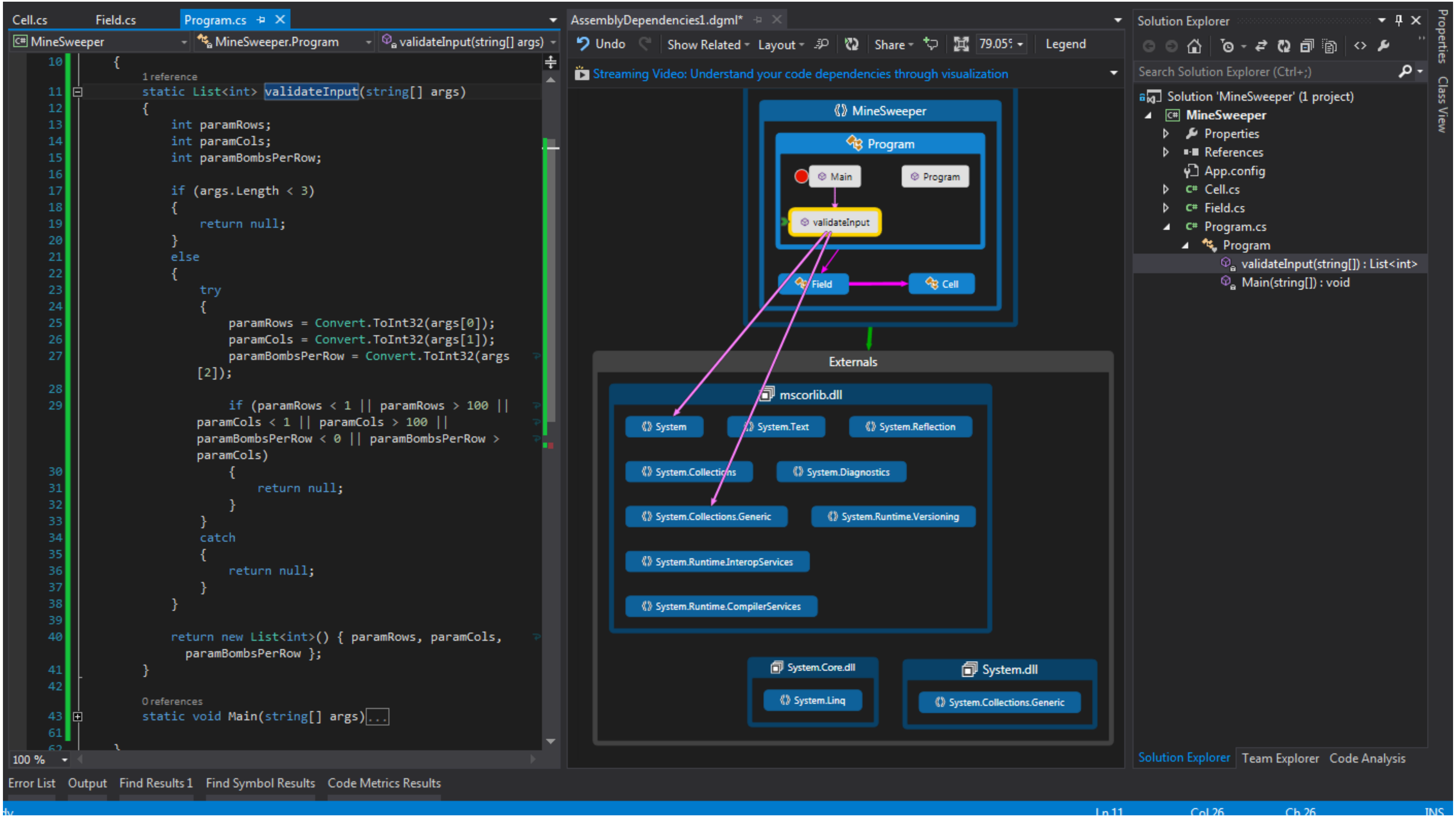
```
| 2 | X | 4 | X | 2 |
| 3 | X | 6 | X | 3 |
| 3 | X | 6 | X | 3 |
| 2 | X | 6 | X | 3 |
| 1 | 2 | X | X | 2 |

PS C:\Users\a5546zz\Documents\Visual Studio 2013\Projects\MineSweeper\MineSweeper\bin\Debug> ./MineSweeper.exe 5 5 2
| 2 | X | 3 | X | 1 |
| X | 5 | X | 3 | 1 |
| X | 6 | X | 3 | 0 |
| X | 6 | X | 3 | 0 |
| X | 4 | X | 2 | 0 |

PS C:\Users\a5546zz\Documents\Visual Studio 2013\Projects\MineSweeper\MineSweeper\bin\Debug> ./MineSweeper.exe 5 5 2
| X | 3 | X | 2 | 1 |
| 3 | X | 5 | X | 2 |
| 3 | X | 6 | X | 3 |
| 3 | X | 6 | X | 3 |
| 2 | X | 4 | X | 2 |

PS C:\Users\a5546zz\Documents\Visual Studio 2013\Projects\MineSweeper\MineSweeper\bin\Debug> ./MineSweeper.exe 10 10 3
| 0 | 1 | X | X | 2 | 0 | 2 | X | 3 | 1 |
| 1 | 2 | 3 | X | 3 | 1 | 2 | X | X | 2 |
| X | 2 | 1 | 3 | X | 3 | 2 | 3 | X | 2 |
| X | 4 | 1 | 2 | X | X | 2 | 1 | 1 | 1 |
| X | X | 3 | 2 | 3 | X | 3 | 1 | 0 | 0 |
| 3 | X | X | 2 | 1 | 3 | X | 3 | 1 | 0 |
| 1 | 4 | X | 4 | 1 | 2 | X | X | 2 | 0 |
| 0 | 2 | X | X | 3 | 2 | 3 | X | 3 | 1 |
| 1 | 2 | 3 | X | X | 2 | 1 | 3 | X | 2 |
| X | 1 | 1 | 3 | X | 2 | 0 | 2 | X | 2 |

PS C:\Users\a5546zz\Documents\Visual Studio 2013\Projects\MineSweeper\MineSweeper\bin\Debug> ./MineSweeper.exe 10 10 3
| X | X | 2 | 0 | 2 | X | 3 | 1 | 0 | 0 |
| 3 | X | 3 | 1 | 2 | X | X | 3 | 1 | 0 |
| 1 | 3 | X | 3 | 2 | 3 | X | X | 2 | 0 |
| 0 | 2 | X | X | 2 | 1 | 4 | X | 4 | 1 |
| 1 | 2 | 3 | X | 3 | 1 | 2 | X | X | 2 |
| X | 2 | 1 | 3 | X | 3 | 2 | 3 | X | 2 |
| X | 3 | 1 | 2 | X | X | 3 | 2 | 1 | 1 |
| 3 | X | 3 | 2 | 3 | X | X | 2 | 0 | 0 |
| 2 | X | X | 2 | 1 | 4 | X | 4 | 1 | 0 |
| 1 | 3 | X | 2 | 0 | 2 | X | X | 1 | 0 |

 PS C:\Users\a5546zz\Documents\Visual Studio 2013\Projects\MineSweeper\MineSweeper\bin\Debug>
PS C:\Users\a5546zz\Documents\Visual Studio 2013\Projects\MineSweeper\MineSweeper\bin\Debug> ./MineSweeper.exe 10 10 3
| 0 | 1 | X | X | 3 | 1 | 1 | X | 2 | 1 |
| 1 | 2 | 3 | X | X | 2 | 1 | 3 | X | 2 |
| X | 2 | 1 | 4 | X | 4 | 1 | 2 | X | 2 |
| X | 3 | 1 | 2 | X | X | 3 | 2 | 1 | 1 |
| 3 | X | 3 | 2 | 3 | X | X | 2 | 0 | 0 |
| 2 | X | X | 2 | 1 | 4 | X | 4 | 1 | 0 |
| 1 | 3 | X | 3 | 1 | 2 | X | X | 3 | 1 |
| 0 | 1 | 3 | X | 3 | 2 | 3 | X | X | 2 |
| 1 | 1 | 2 | X | X | 2 | 1 | 4 | X | 3 |
| X | 1 | 1 | 3 | X | 2 | 0 | 2 | X | 2 |
```

Architecture

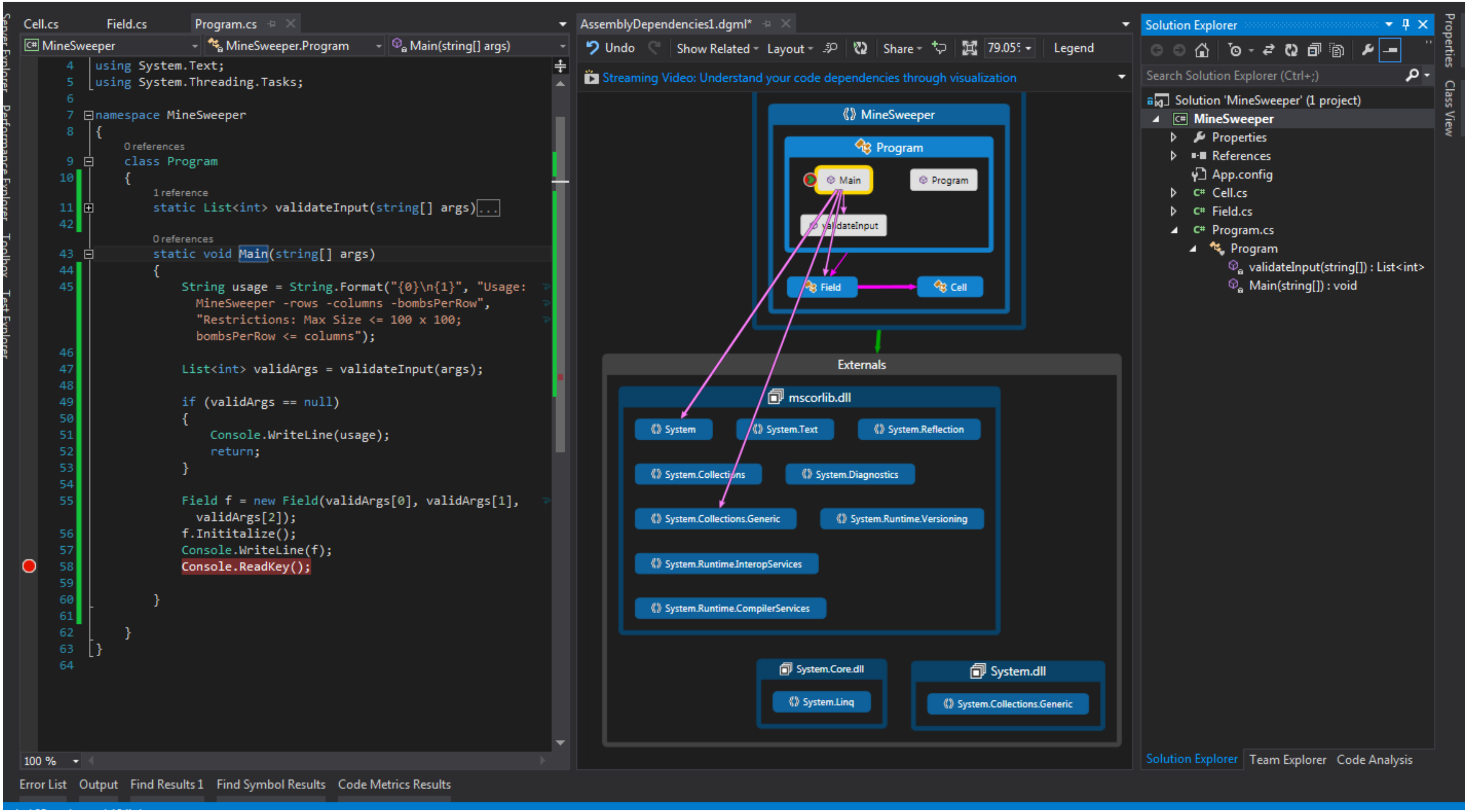
The solution has three classes, `Cell` (represents a cell in the game field), `Field` (represents the game field), `Program` (validates input and consumes the public interface of `Field` to generate and display a MineSweeper field).



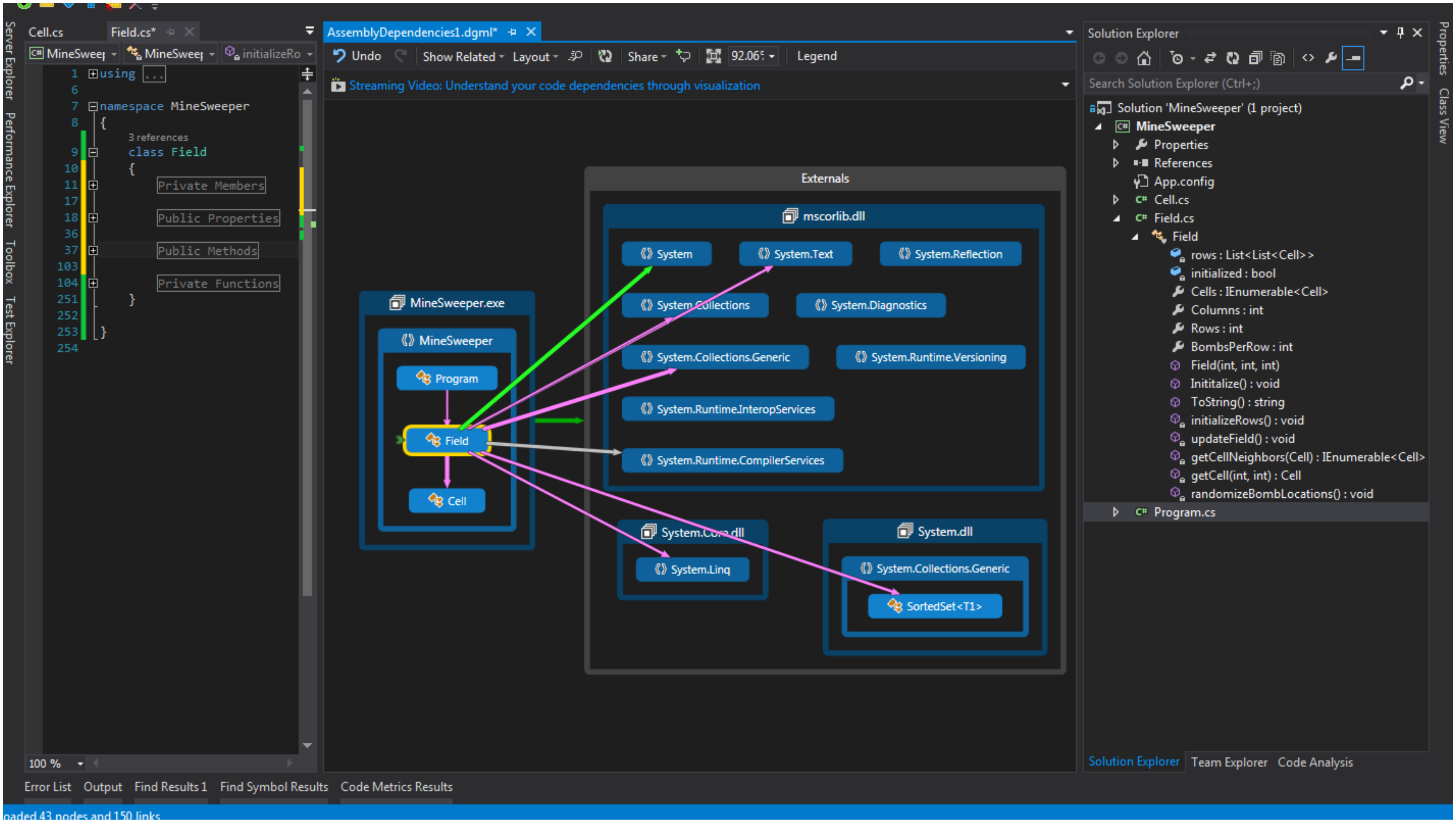
Loaded 20 nodes and 37 links



```
    {
        static List<int> validateInput(string[] args)
        {
            int paramRows;
            int paramCols;
            int paramBombsPerRow;

            if (args.Length < 3)
            {
                return null;
            }
            else
            {
                try
                {
                    paramRows = Convert.ToInt32(args[0]);
                    paramCols = Convert.ToInt32(args[1]);
                    paramBombsPerRow = Convert.ToInt32(args
                [2]);

                    if (paramRows < 1 || paramRows > 100 ||
                paramCols < 1 || paramCols > 100 ||
                paramBombsPerRow < 0 || paramBombsPerRow >
                paramCols)
                    {
                        return null;
                    }
                }
                catch
                {
                    return null;
                }
            }

            return new List<int>() { paramRows, paramCols,
              paramBombsPerRow };
        }

        static void Main(string[] args)...
```



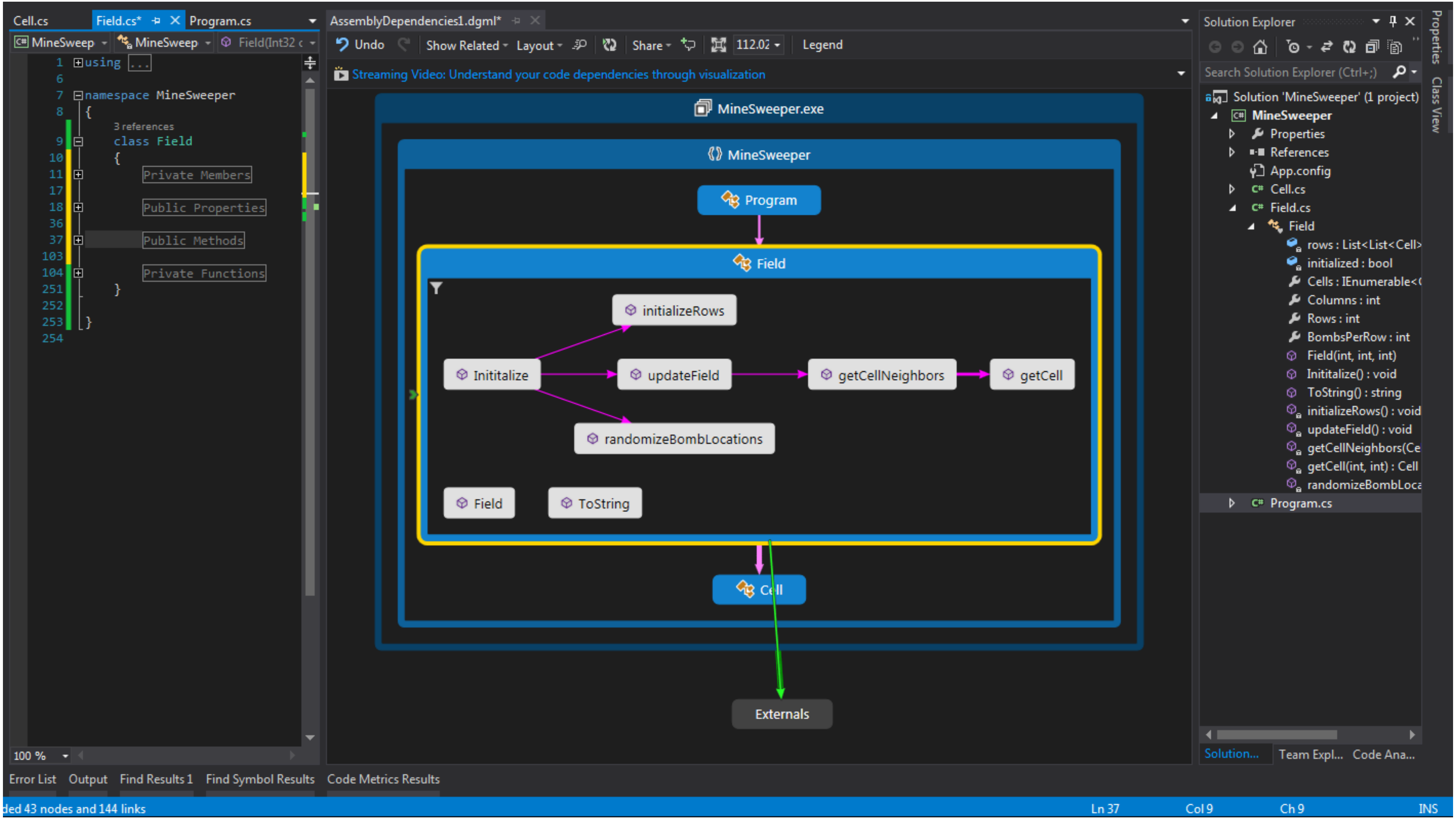
```
using System.Text;
using System.Threading.Tasks;

namespace MineSweeper
{
    class Program
    {
        static List<int> validateInput(string[] args)...

        static void Main(string[] args)
        {
            String usage = String.Format("{0}\n{1}", "Usage:
              MineSweeper -rows -columns -bombsPerRow",
              "Restrictions: Max Size <= 100 x 100;
              bombsPerRow <= columns");

            List<int> validArgs = validateInput(args);

            if (validArgs == null)
            {
                Console.WriteLine(usage);
                return;
            }

            Field f = new Field(validArgs[0], validArgs[1],
              validArgs[2]);
            f.Inititalize();
            Console.WriteLine(f);
            Console.ReadKey();

        }

    }
}
```



```
using ...

namespace MineSweeper
{
    3 references
    class Field
    {
        Private Members

        Public Properties

        Public Methods

        Private Functions
    }
}
```

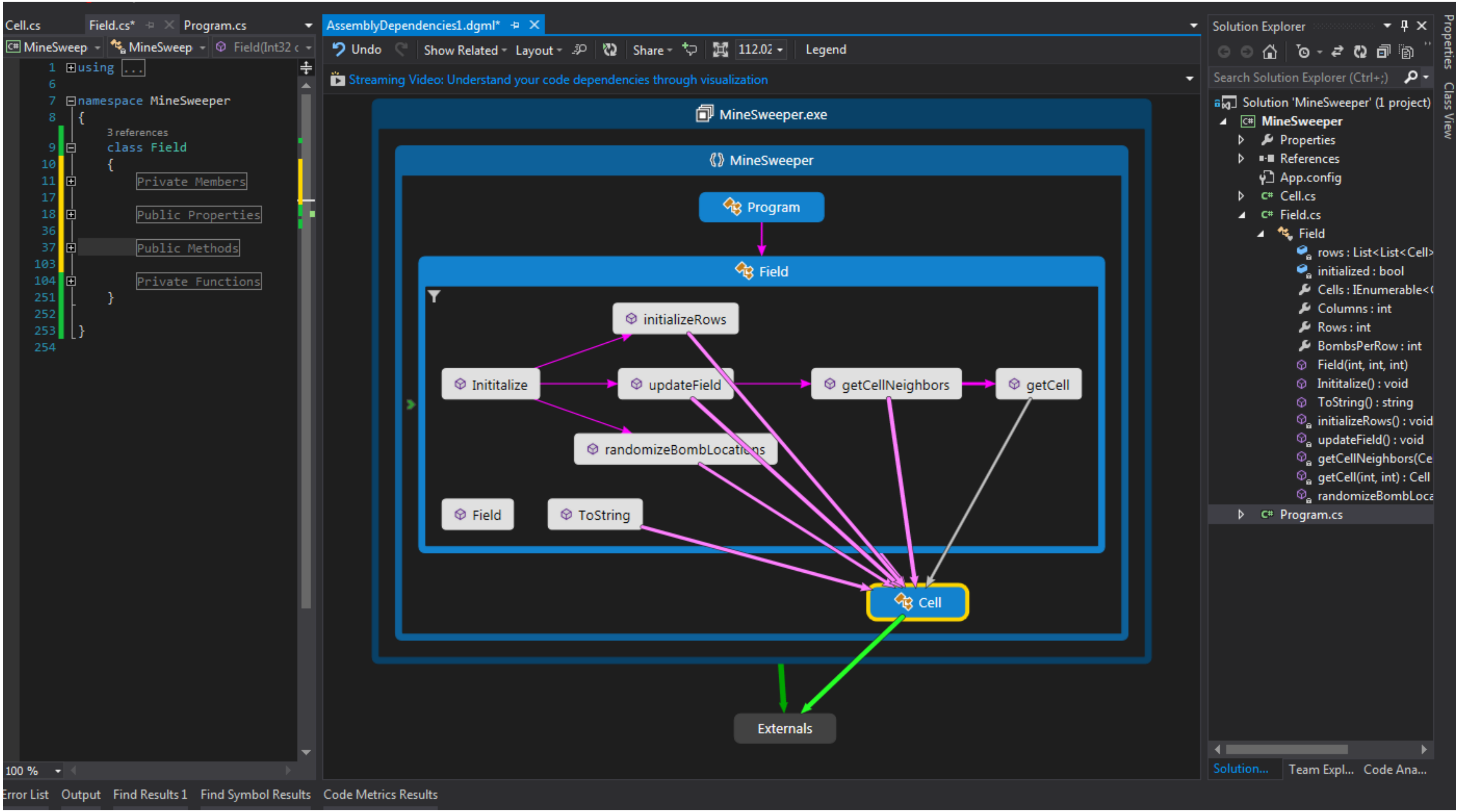
Solution 'MineSweeper' (1 project)
- MineSweeper
  - Properties
  - References
  - App.config
  - Cell.cs
  - Field.cs
    - Field
      - rows : List<List<Cell>>
      - initialized : bool
      - Cells : IEnumerable<Cell>
      - Columns : int
      - Rows : int
      - BombsPerRow : int
      - Field(int, int, int)
      - Inititalize() : void
      - ToString() : string
      - initializeRows() : void
      - updateField() : void
      - getCellNeighbors(Cell) : IEnumerable<Cell>
      - getCell(int, int) : Cell
      - randomizeBombLocations() : void
  - Program.cs

oaded 43 nodes and 150 links



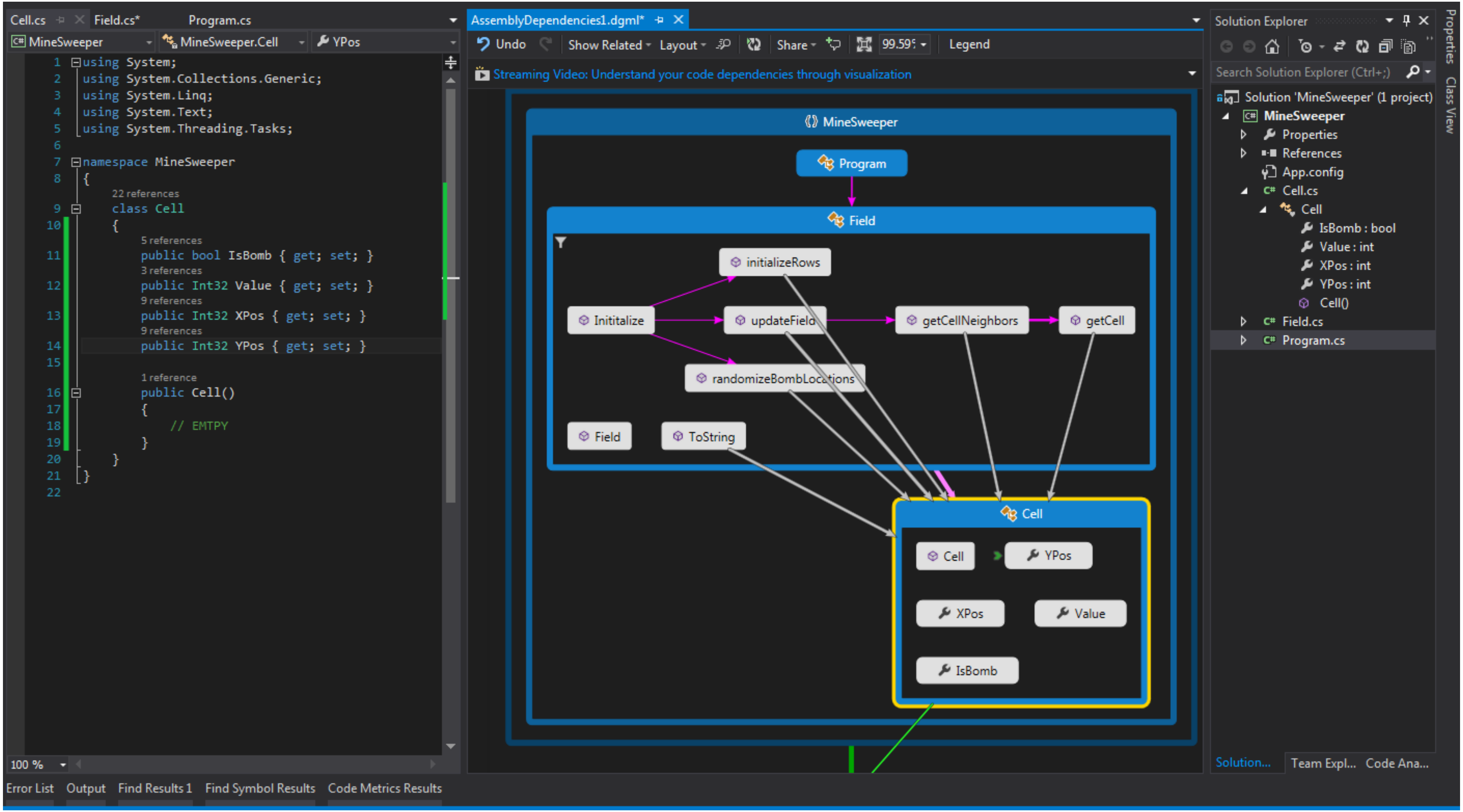
```
using ...
namespace MineSweeper
{
    3 references
    class Field
    {
        Private Members
        Public Properties
        Public Methods
        Private Functions
    }
}
```

ded 43 nodes and 144 links



```
using ...

namespace MineSweeper
{
    3 references
    class Field
    {
        Private Members

        Public Properties

        Public Methods

        Private Functions
    }
}
```

Solution 'MineSweeper' (1 project)
MineSweeper
Properties
References
App.config
Cell.cs
Field.cs
Field
rows : List<List<Cell>
initialized : bool
Cells : IEnumerable<(
Columns : int
Rows : int
BombsPerRow : int
Field(int, int, int)
Inititalize() : void
ToString() : string
initializeRows() : void
updateField() : void
getCellNeighbors(Ce
getCell(int, int) : Cell
randomizeBombLoca
Program.cs



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MineSweeper
{
    22 references
    class Cell
    {
        5 references
        public bool IsBomb { get; set; }
        3 references
        public Int32 Value { get; set; }
        9 references
        public Int32 XPos { get; set; }
        9 references
        public Int32 YPos { get; set; }

        1 reference
        public Cell()
        {
            // EMTPY
        }
    }
}
```