# Sample Refactoring Documentation for Project “Battle Field 5”

Team “Xenon”

Промени от Атанас

1. Redesigned the project structure:

* Added Unit test project BattleField.Tests and method in it Explosion.Tests

1. Reformatted the source code:

* Changed in IsInsideField() method:

if (isXOutsideOfField || isYOutsideOfField)

{

return false;

}

return true;

into:

if (isXOutsideOfField || isYOutsideOfField)

{

return false;

}

else

{

return true;

}

* Removed in ExplosionOne() method:

Mine URcorner = new Mine(mine.X - 1, mine.Y - 1);

Mine ULcorner = new Mine(mine.X - 1, mine.Y + 1);

Mine DRcorner = new Mine(mine.X + 1, mine.Y - 1);

Mine DLcorner = new Mine(mine.X + 1, mine.Y + 1);

* Changed in ExplosionOne() method:

if (IsInsideField(field, URcorner.X, URcorner.Y))

{

field[URcorner.X, URcorner.Y] = DetonatedCell;

}

if (IsInsideField(field, ULcorner.X, ULcorner.Y))

{

field[ULcorner.X, ULcorner.Y] = DetonatedCell;

}

if (IsInsideField(field, DRcorner.X, DRcorner.Y))

{

field[DRcorner.X, DRcorner.Y] = DetonatedCell;

}

if (IsInsideField(field, DLcorner.X, DLcorner.Y))

{

field[DLcorner.X, DLcorner.Y] = DetonatedCell;

}

Into:

if (IsInsideField(field, mine.X - 1, mine.Y - 1))

{

field[mine.X - 1, mine.Y - 1] = DetonatedCell;

}

if (IsInsideField(field, mine.X + 1, mine.Y - 1))

{

field[mine.X + 1, mine.Y - 1] = DetonatedCell;

}

if (IsInsideField(field, mine.X - 1, mine.Y + 1))

{

field[mine.X - 1, mine.Y + 1] = DetonatedCell;

}

if (IsInsideField(field, mine.X + 1, mine.Y + 1))

{

field[mine.X + 1, mine.Y + 1] = DetonatedCell;

}

* Removed from method ExplosionThree():

Mine Up = new Mine(mine.X - 2, mine.Y);

Mine Down = new Mine(mine.X + 2, mine.Y);

Mine Left = new Mine(mine.X, mine.Y - 2);

Mine Right = new Mine(mine.X, mine.Y + 2);

* Changed from method ExplosionThree():

if (IsInsideField(field, Up.X, Up.Y))

{

field[Up.X, Up.Y] = DetonatedCell;

}

if (IsInsideField(field, Down.X, Down.Y))

{

field[Down.X, Down.Y] = DetonatedCell;

}

if (IsInsideField(field, Left.X, Left.Y))

{

field[Left.X, Left.Y] = DetonatedCell;

}

if (IsInsideField(field, Right.X, Right.Y))

{

field[Right.X, Right.Y] = DetonatedCell;

}

Into

if (IsInsideField(field, mine.X, mine.Y - 2))

{

field[mine.X, mine.Y - 2] = DetonatedCell;

}

if (IsInsideField(field, mine.X + 2, mine.Y))

{

field[mine.X + 2, mine.Y] = DetonatedCell;

}

if (IsInsideField(field, mine.X, mine.Y + 2))

{

field[mine.X, mine.Y + 2] = DetonatedCell;

}

if (IsInsideField(field, mine.X - 2, mine.Y))

{

field[mine.X - 2, mine.Y] = DetonatedCell;

}

* Changed in ExplosionFour() method:

if (isInUpperLeftCorner) continue;

if (isInBottomLeftCorner) continue;

if (isInUpperRightCorner) continue;

if (isInBottomRightCorner) continue;

into

if (isInUpperLeftCorner || isInBottomLeftCorner || isInUpperRightCorner || isInBottomRightCorner)

{

continue;

}

1. Renamed variables:

* rowCondition to isXOutsideOfField in IsInsideField() method
* colCondition to isYOutsideOfField in IsInsideField() method
* i to row and j to col in ExplosionTwo() method
* i to row and j to col in ExplosionFour() method
* UR to isInUpperLeftCorner in ExplosionFour() method
* UL to isInBottomLeftCorner in ExplosionFour() method
* DR to isInUpperRightCorner in ExplosionFour() method
* DL to isInBottomRightCorner in ExplosionFour() method
* i to row and j to col in ExplosionFive() method

1. Renamed methods:

* ExplodeOne into ExplosionOne
* ExplodeTwo into ExplosionTwo
* ExplodeThree into ExplosionThree
* ExplodeFour into ExplosionFour
* ExplodeFive into ExplosionFive

1. Renamed classes:

* Renamed Explosion.Tests.cs to ExplosionTests.cs and GameFieldTest.cs to GameFieldTests.cs

1. Extracted Explosion class again, moved method IsInsideField() from GameServices to Explosion, moved const DetonatedCell from GameServices to Explosion.