

Update firing pointers

```
public void UpdateFiringEnemiesList(int i)
{
    //Retrieve the values in the jagged array to access the pointer and
    //to which the next enemy in the list is to fire
    float tempX = 0, tempY = 0;
    int? nextFire = 0, tempPointer = 0;

    bool corrected = false;
    int count = 1;

    tempX = enemies[i].Number.X;
    tempY = enemies[i].Number.Y;

    if (enemies[i].prevState == Enemy.previousState.HasLineofSight)
    {
        if (tempY != null && tempX != null)
        {
            nextFire = enemyBlock[(int)tempX][(int)tempY];

            if (nextFire != null)
            {
                enemies[(int)nextFire].changeState(idle, hasLineofSight);
            }

            enemyBlock[(int)tempX][(int)tempY] = null;
        }
        else
        {
            nextFire = enemyBlock[(int)tempX][(int)tempY];

            if (tempX != null)
            {
                enemies[(int)nextFire].changeState(idle, hasLineofSight);
            }

            enemyBlock[(int)tempX][(int)tempY] = null;
        }

        checkJagged();
    }
    else
    {
        tempX = enemies[i].Number.X;
        tempY = enemies[i].Number.Y;

        tempPointer = enemyBlock[(int)tempX][(int)tempY];
        Console.WriteLine("temp pointer = " + tempPointer);

        enemyBlock[(int)tempX][(int)tempY] = null;

        while (corrected == false)
        {
            if (enemyBlock[(int)tempX + count][(int)tempY] != null)
            {
                enemyBlock[(int)tempX + count][(int)tempY] = tempPointer;
                corrected = true;
                break;
            }
            else
            {
                count++;
            }
        }

        checkJagged();
    }
}
```