



Implementing Optimized Line of Sight based

# Fog of War

in Unity

Bacheloroppgave 086

Sindre Haugland Paulshus

Mai 2020



# Oppgaven: Hvem?

STARTUP 2016

**Pineleaf Studio**

TRONDHEIM

SPILLUTVIKLING

DWARFHEIM





# Oppgaven: Hva?

Utvikle en “Fog of War” løsning

Med “Line of Sight” mechanic

Unity med C#

Optimalisert

Modulært

Visuelt

“How can a Line of Sight based Fog of War solution for a Real Time Strategy game that maintains desired game performance be implemented in Unity?”

DWARFHEIM

# Oppgaven: Hvorfor?

Spennende fagområde

Interessert i spillutvikling

Hadde erfaring i nødvendig programvare

Var kjent med Pineleaf Studio fra før





```

StaticXY (viewDistance, tolerance)
int[] edgeArrayX, edgeArrayY
float viewDistSq = (viewDistance + tolerance) *
    (viewDistance + tolerance)

int dy = viewDistance
int dx, index

//find up->right oct
while (dy >= dx)
    if (dx+dx+dy+dy <= viewDistSq) //Pythagoras
        edgeArrayX[index] = dx
        edgeArrayY[index] = dy
        dx++
        index++
    else
        dy--

int octantLen = index
int[] newEdgeArrayX = new int[octantLen * 8]
int[] newEdgeArrayY = new int[octantLen * 8]

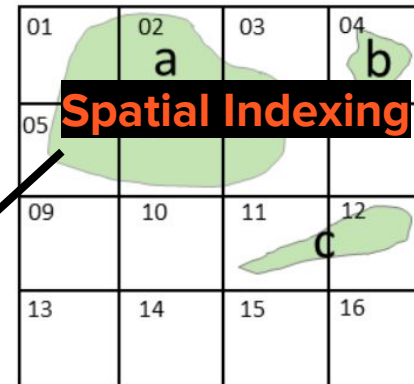
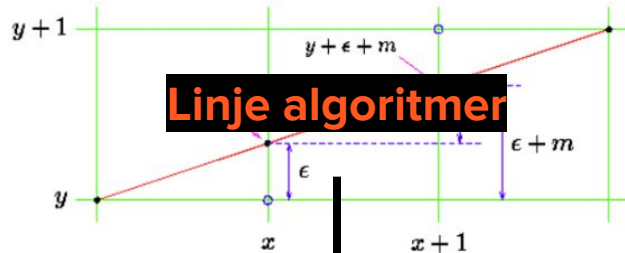
for (int i = 0; i < octantLen; i++)
    newEdgeArrayX[i] = edgeArrayX[i]
    newEdgeArrayY[i] = edgeArrayY[i]

//right->up oct with zdy shifting
for (int i = 0; i < octantLen; i++)
    dx = newEdgeArrayX[i]
    dy = newEdgeArrayY[i]
    newEdgeArrayX[index] = dx
    newEdgeArrayY[index] = dy
    index++

/* Similarly for other octants (not shown)
 * All octants values are:
 * (nz, ny = new values, x, y = original)
 * up->right: (x,y)
 * right->up: nz = y, ny = x
 * up->left:  nz = -x, ny = y
 * left->up:  nz = -y, ny = x
 * left->down: nz = -y, ny = -x
 * right->down: nz = y, ny = -x
 * down->left: nz = -x, ny = -y
 * down->right: nz = x, ny = -y
 */
return newEdgeArrayX, newEdgeArrayY

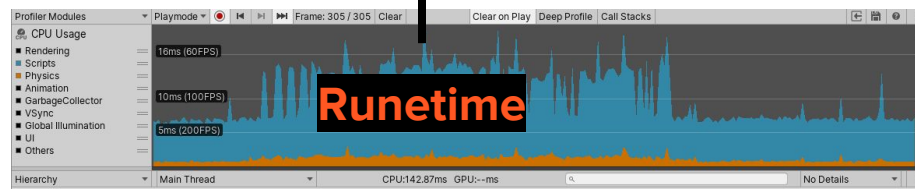
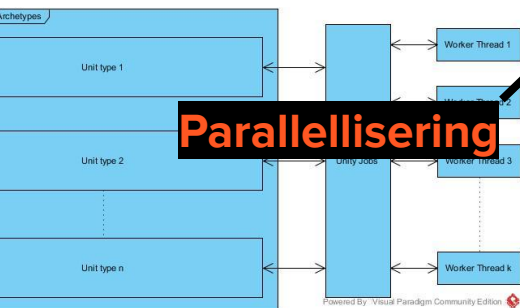
```

**Unit Vision basert**



01	a
02	a
03	a
04	b
...	...
11	c
12	c

**Løsningen**



```

[BurstCompile]
public struct MyJob : IJob
{
    public void Execute()
    {
        // behaviour here
    }
}

```

**Unity Burst**

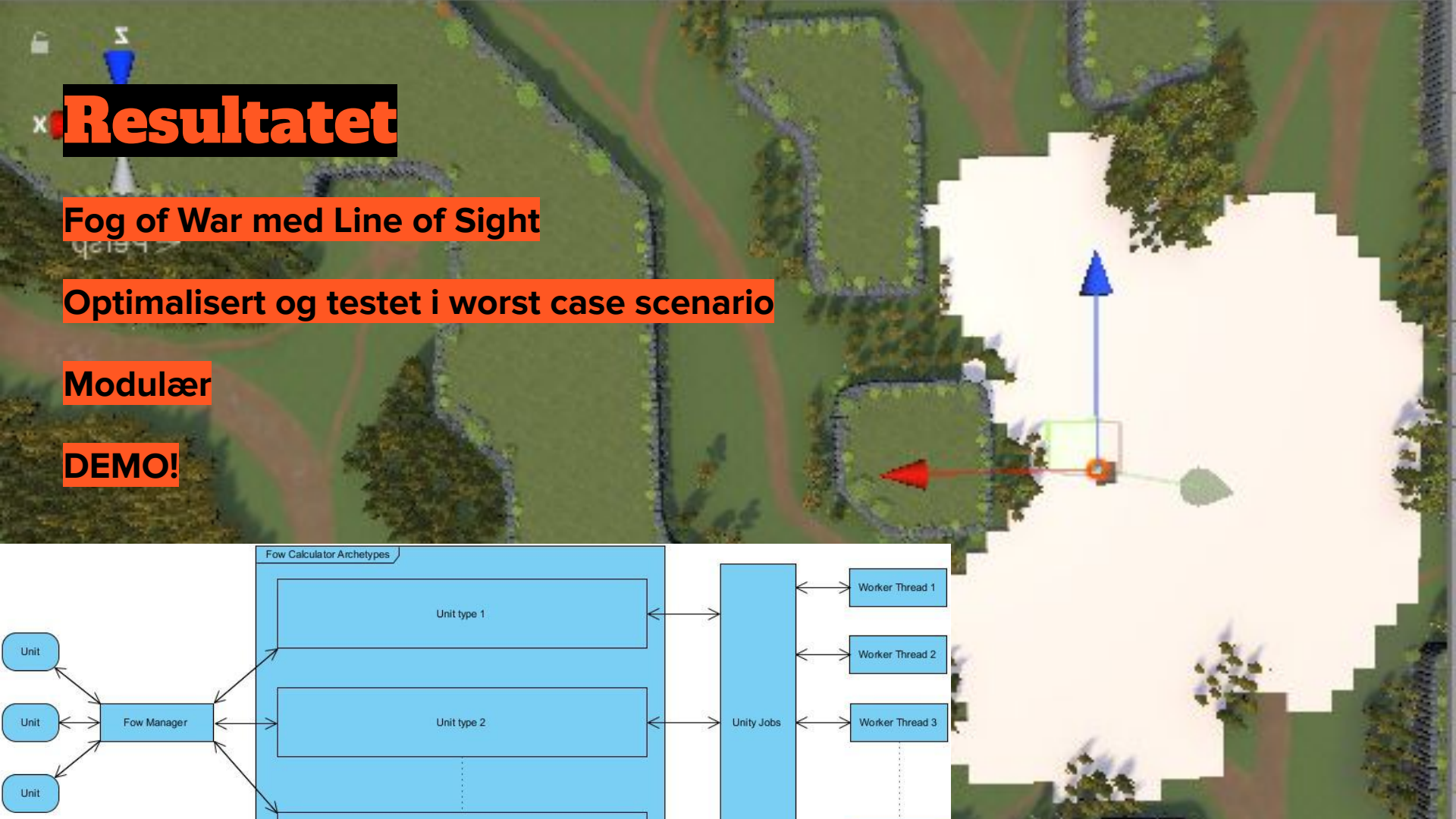
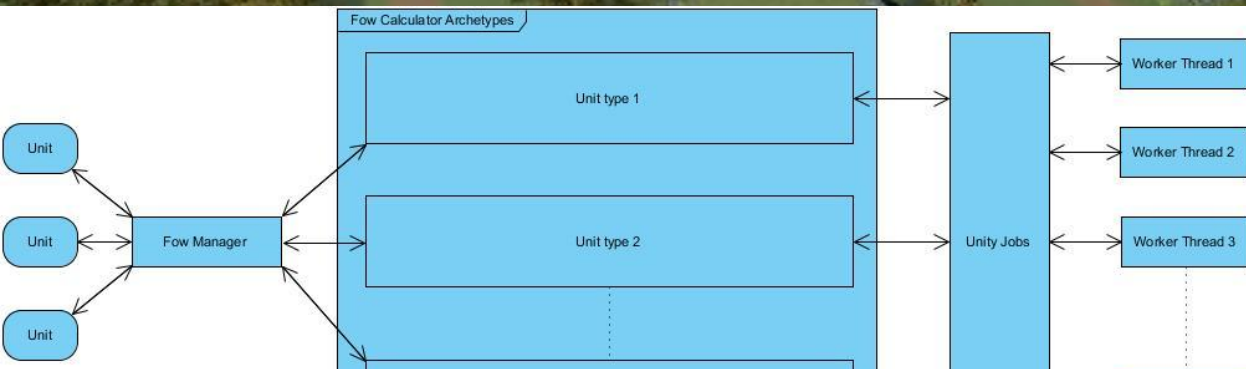
# Resultatet

Fog of War med Line of Sight

Optimalisert og testet i worst case scenario

Modulær

DEMO!





# Videre arbeid

Optimalisering

Bruk av andre metoder: Bresenham, pre-baking, data oriented design

Andre spillmotorer

