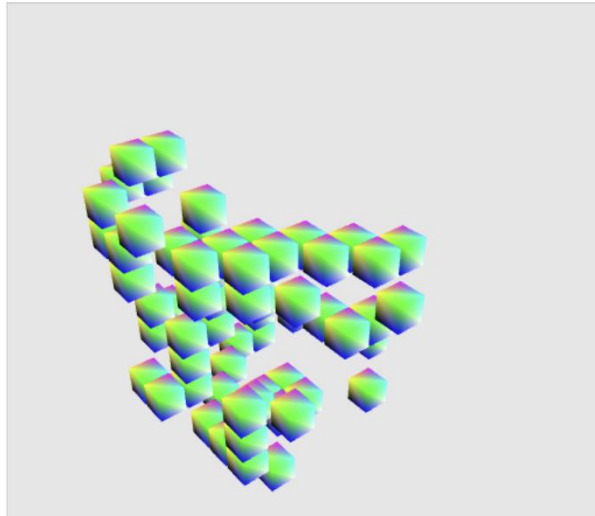


Nafn: Ásdís Valtýsdóttir

13.10. 2024

## Game of Life



Hlekkur á vefsíðuna mína: <https://asdisval.github.io/Einstaklingsverkefni2/GameOfLife.html>

Þetta verkefnið Conways Game of Life þar sem frumur lifa eða deyja út frá fjölda nágranna þeirra. Hér eru helstu atriðin í verkefninu

### Grindin

Ég bý til tvær grindur og með þeirra samspili næ ég að uppfæra frumurnar/kassana m.t.t. nágranna þeirra.

Ég læt grindina búa til slembna grind með hjálp Math.random.

```
//create the grid
function createGrid(gridSize)
{
  let grid = [];
  for(let x = 0; x < gridSize; x++) {
    grid[x] = [];
    for(let y = 0; y < gridSize; y++) {
      grid[x][y] = [];
      for(let z = 0; z < gridSize; z++) {
        grid[x][y][z] = Math.random() > 0.7 ? 1 : 0;
      }
    }
  }
  return grid;
}
```

## Nágrannar

Nágrannarnir segja til um hvort að fruma mun lifa(1) eða deyja(0). Ef að fruman er lifandi og nágrannarnir eru 5 til 7 er hún enþá á lífi, annars deyr hún.

Ef að fruma er dauð en það eru nákvæmlega sex nágrannar að þá lifnar hún við.

```
//draw the cubes that stayed the same
function drawCube(x, y, z, globalTransformMatrix)
{
    let mv = mat4();
    let xCoordinates = (x - centeringGrid) * spacing;
    let yCoordinates = (y - centeringGrid) * spacing;
    let zCoordinates = (z - centeringGrid) * spacing;
    mv = mult(mv, translate(xCoordinates, yCoordinates, zCoordinates));

    let transform = mult(globalTransformMatrix, mv);
    gl.uniformMatrix4fv(matrixLoc, false, flatten(transform));

    gl.drawArrays(gl.TRIANGLES, 0, numVertices);
}
```

```
// draw the cubes that need to be animated
function drawAnimatedCube(x, y, z, globalTransformMatrix, scale, rotation)
{
    let mv = mat4();

    let scaleSlowly = scale * scale * 0.95; // scale slowly for x, y and z coordinates

    let xCoordinates = (x - centeringGrid) * spacing;
    let yCoordinates = (y - centeringGrid) * spacing;
    let zCoordinates = (z - centeringGrid) * spacing;

    mv = mult(mv, translate(xCoordinates, yCoordinates, zCoordinates));
    mv = mult(mv, rotateX(rotation * (360/ Math.PI )))
    mv = mult(mv, rotateY(rotation * (360/ Math.PI )))
    mv = mult(mv, scalem(scaleSlowly, scaleSlowly, scaleSlowly));

    let transformedCubes = mult(globalTransformMatrix, mv);

    gl.uniformMatrix4fv(matrixLoc, false, flatten(transformedCubes));
    gl.drawArrays(gl.TRIANGLES, 0, numVertices);
}
```

```
// Update the Grid
function updateGrid()
{
    prevGrid = grid;
    let newGrid = createEmptyGrid(gridSize);

    for(let x = 0; x < gridSize; x++) {
        for(let y = 0; y < gridSize; y++) {
            for(let z = 0; z < gridSize; z++) {
                let neighbours = countNeighbours(x, y, z);

                if(grid[x][y][z] === 1) {
                    if(neighbours >= 5 && neighbours <= 7) {
                        newGrid[x][y][z] = 1;
                    } else {
                        newGrid[x][y][z] = 0;
                    }
                } else {
                    if(neighbours === 6) {
                        newGrid[x][y][z] = 1;
                    } else {
                        newGrid[x][y][z] = 0;
                    }
                }
            }
        }
    }

    grid = newGrid;
    lastUpdateTime = Date.now();
}
```

## Kassar/Frumur

Ég teikna annaðhvort venjulegan kassa, eða kassa sem deyr eða lifnar við. Það geri ég útfrá föllunum drawCube og drawAnimatedCube.

Þar staðset ég hvar kassarnir verða miðað við grindina.

Í drawAnimatedCube fannst mér einnig flott að kassarnir myndu minnka og stækka með auka snúningi og ég bætti því við í mv.

## Tímasetning

Til þess að breyta ekki strax kössunum þurfti að halda utanum tímasetninguna á því. Það gerði ég í fallinu `handleAnimationTiming`.

```
//Handle the animation timing
function handleAnimationTiming(currentTime, _lastUpdateTime, _animationLength, _updateInterval)
{
    const timeElapsed = currentTime - lastUpdateTime;

    if(timeElapsed > updateInterval) {
        updateGrid();
    }

    if( timeElapsed < animationLength ) {
        var progress = (timeElapsed % animationLength) / animationLength;
        var rotation = progress * fullRotation;
        return {animate: true, progress, rotation};
    }

    return { animate: false, progress: 1, rotation: 0};
}
```

## Auka

Til þess að hafa kassana litríka forritaði ég `colors.push(vertexColors[indices[i]]);` í staðinn fyrir `colors.push(vertexColors[a]);`

```
var vertexColors = [
    [ 0.0, 0.0, 0.0, 1.0 ],
    [ 1.0, 1.0, 0.0, 1.0 ],
    [ 0.0, 1.0, 0.0, 1.0 ],
    [ 0.0, 0.0, 1.0, 1.0 ],
    [ 1.0, 0.0, 0.0, 1.0 ],
    [ 1.0, 0.0, 1.0, 1.0 ],
    [ 0.0, 1.0, 1.0, 1.0 ],
    [ 1.0, 1.0, 1.0, 1.0 ]
];

var indices = [ a, b, c, a, c, d ];

for ( var i = 0; i < indices.length; ++i ) {
    points.push( vertices[indices[i]] );
    colors.push(vertexColors[indices[i]]);
}
```