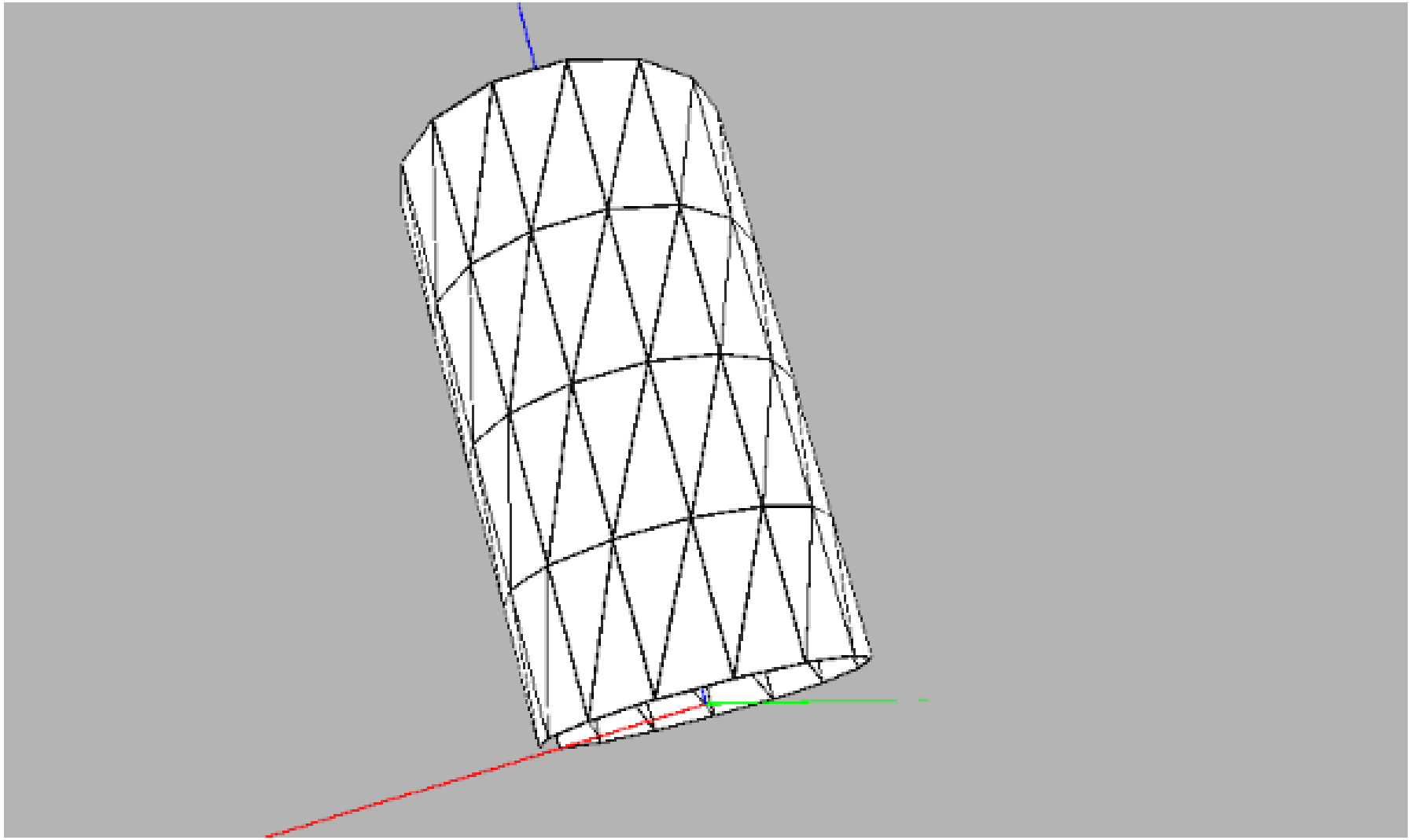
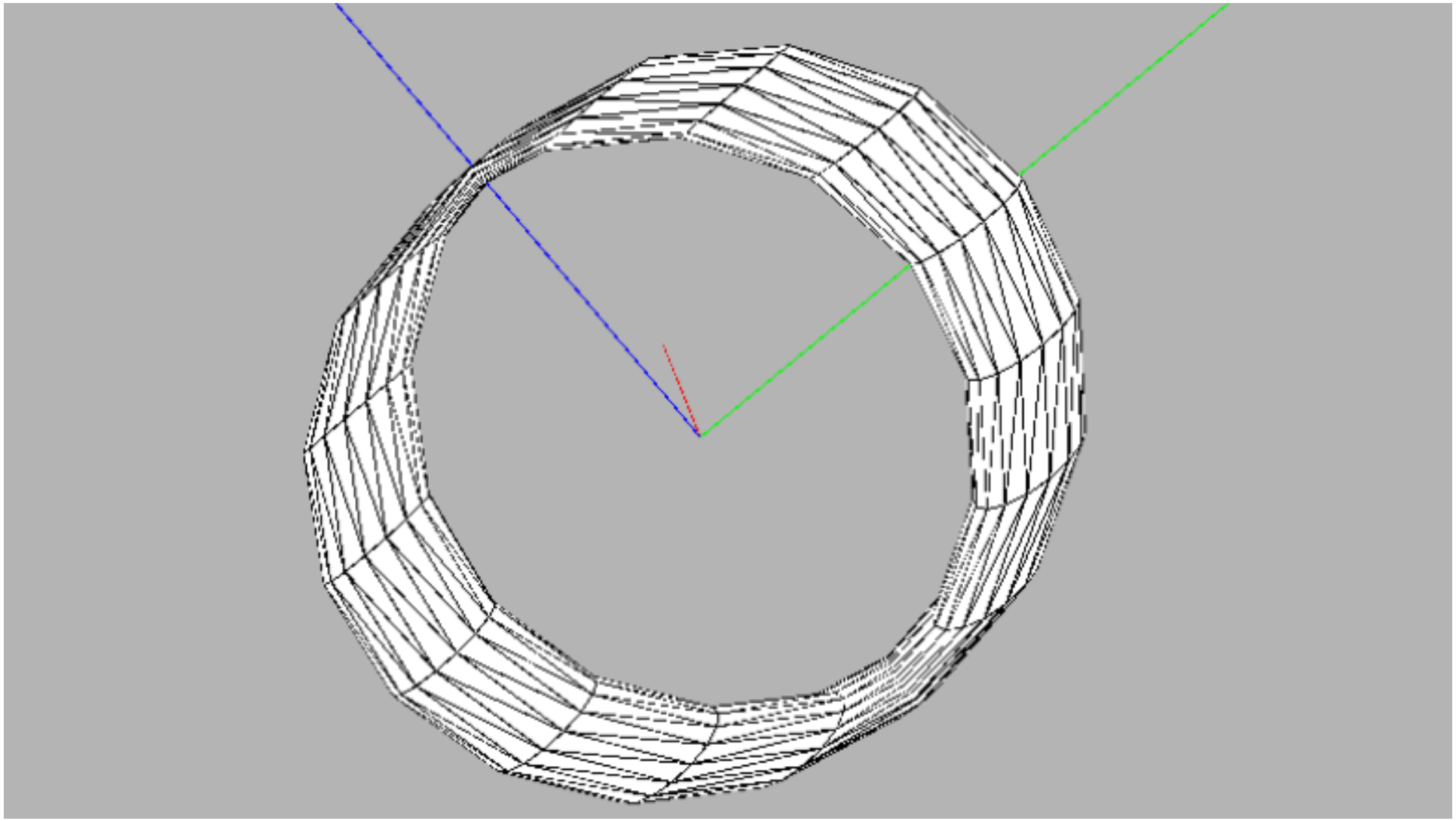
Übungen Woche 3

Simon Gisler

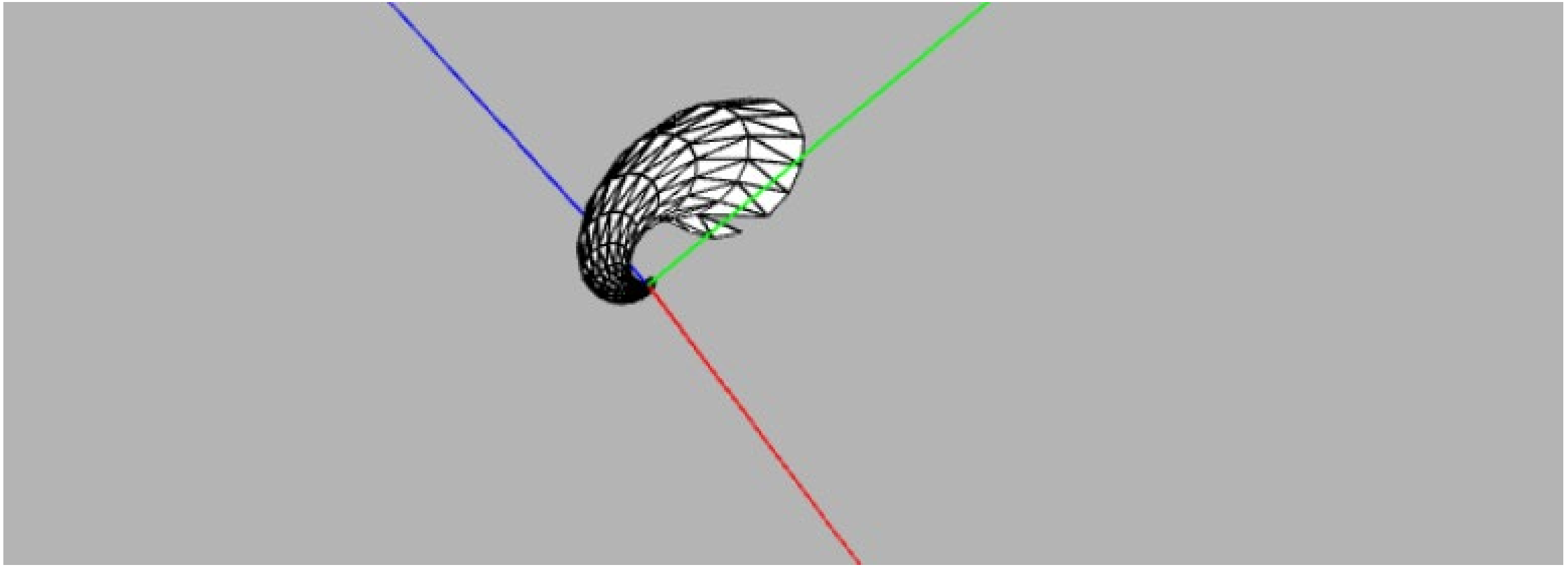
Rohr



Donut



Spirale



Code

```
function zipperJoin(firstInput, secondInput){
    let returnArray = [];
    if (firstInput.length == secondInput.length){
        for (let i = 0; i < firstInput.length; i++){
            returnArray.push(firstInput[i]);
            returnArray.push(secondInput[i]);
        }
    }

    return returnArray;
}
```

```
function getVertexRingVectors(middleX, middleY, middleZ, steps, radius){
  let returnArray = [];
  for(let i = 0; i < steps; i++){
    let angle = radians(i*(360/steps));

    let distanceY = radius*Math.sin(angle);
    let distanceX = radius*Math.cos(angle);

    returnArray.push(createVector(middleX + distanceX, middleY + distanceY, middleZ));
  }
  return returnArray;
}
```

```
let firstVertices;
for(let i = 0; i < STEPS; i++){
    let angle = radians(i * MAX_ANGLE/STEPS);

    radius = exp(angle * SPIRAL_GROWTH);
    thickness = exp(angle * THICKNESS_GROWTH);

    let x = SPIRAL_CENTER.x + radius * sin(angle);
    let y = SPIRAL_CENTER.y + radius * cos(angle);
    let z = SPIRAL_CENTER.z + exp(angle * HEIGHT_GROWTH);

    let newVertices = getVertexRingVectors(SPIRAL_CENTER.x, SPIRAL_CENTER.y, SP

    newVertices.forEach(coords => {
        coords.x = coords.x + x;
        coords.y = coords.y + y;
        coords.z = coords.z + z;
    });

    if (oldVertices == null){
        firstVertices = newVertices;
    } else
    {
        let joinedVertices = zipperJoin(oldVertices, newVertices);
        joinedVertices.forEach(coords => {
            allVertices.push(coords);
        })
    }

    oldVertices = newVertices
```