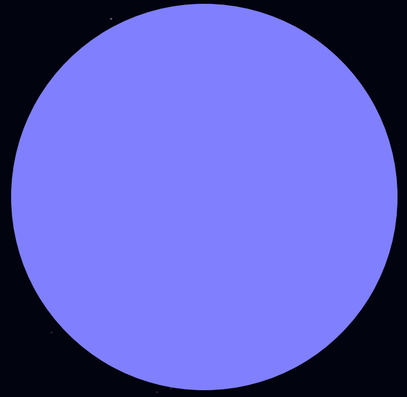
Star field

Designing the stars

We have 2 types of stars:

- Round
- Diffraction spikes (JWST style)

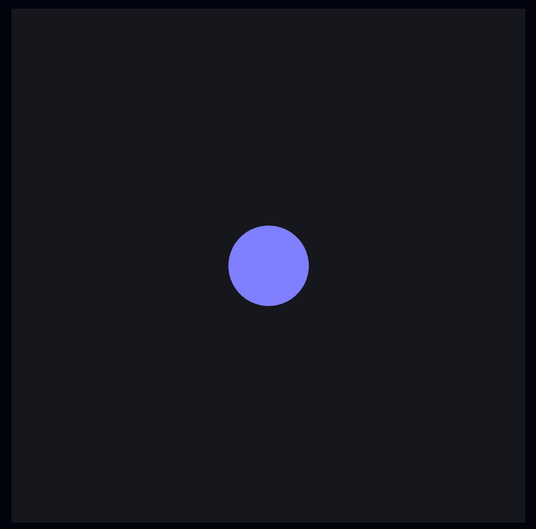


Designing the stars

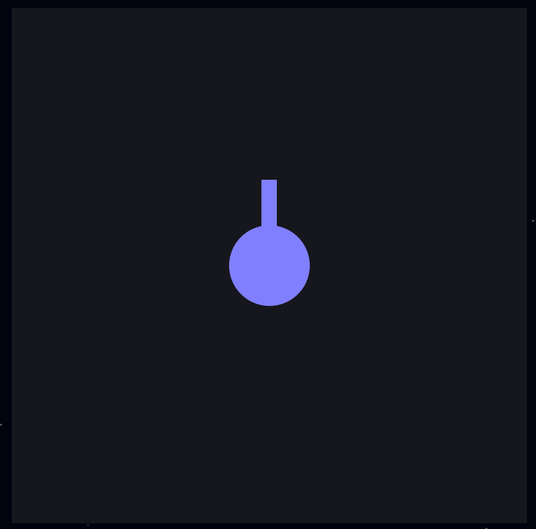
Designing the stars

To draw the stars, we will use the **canvas** in HTML.

The steps to follow to draw the stars with diffraction spikes:

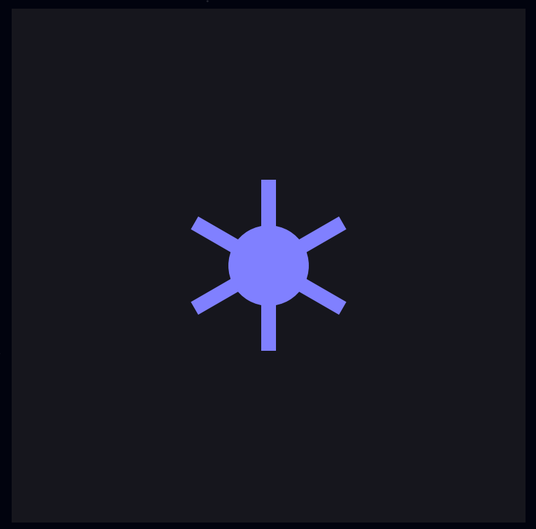


Draw a circle



Draw the 1st half spike

Draw the 2nd half spike



Draw the last half spike

Designing the stars

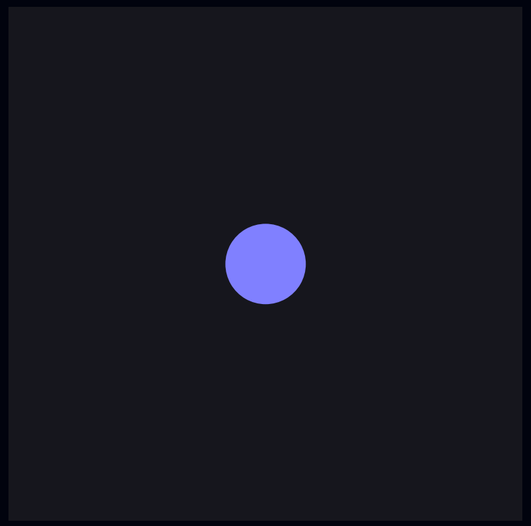
Draw a circle

```
drawStar() {
  this.context.clearRect(0, 0, this.width, this.height);

  // Draw the star circle
  this.context.globalAlpha = this.alpha;
  this.context.beginPath();
  this.context.arc(positionX, positionY, this.radius, 0, 2 * Math.PI);
  this.context.fillStyle = color;
  this.context.shadowColor = color;
  this.context.shadowBlur = this.glowIntensity;
  this.context.fill();
},
```

Designing the stars

Draw a circle

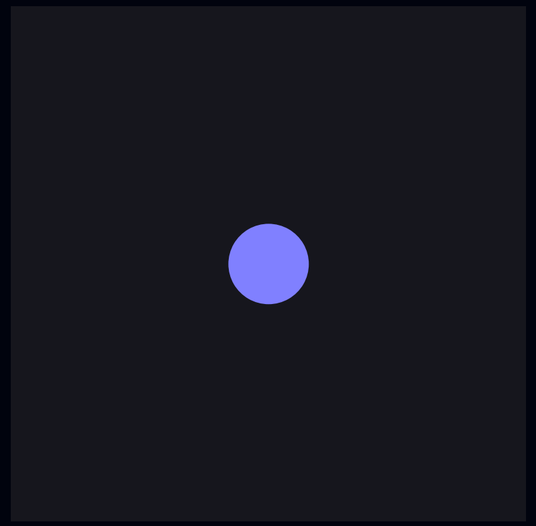


```
drawStar() {
  this.context.clearRect(0, 0, this.width, this.height);

  // Draw the star circle
  this.context.globalAlpha = this.alpha;
  this.context.beginPath();
  this.context.arc(positionX, positionY, this.radius, 0, 2 * Math.PI);
  this.context.fillStyle = color;
  this.context.shadowColor = color;
  this.context.shadowBlur = this.glowIntensity;
  this.context.fill();
},
```

Designing the stars

Draw the spikes



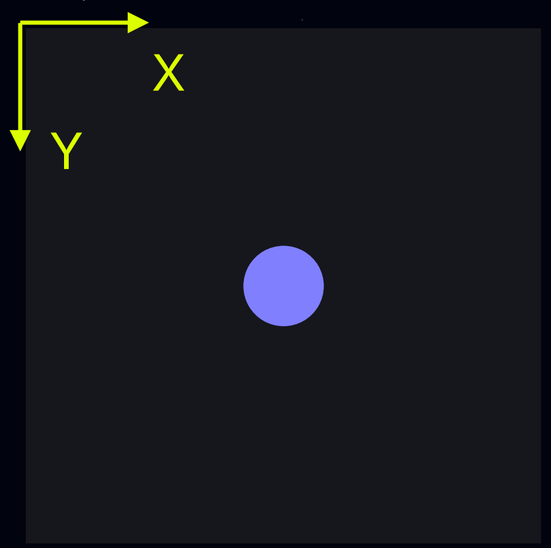
```
drawStar() {
  for (let i = 0; i < numSpikes; i<+) {
    this.context.save();

    const angle = (i * Math.PI) / (numSpikes / 2);
    this.context.translate(positionX, positionY);
    this.context.rotate(angle);
    this.context.beginPath();
    this.context.moveTo(0, 0);
    this.context.lineTo(0, -spikeLength);
    this.context.strokeStyle = color;
    this.context.shadowColor = color;
    this.context.shadowBlur = this.glowIntensity;
    this.context.lineWidth = spikeWidth;
    this.context.globalAlpha = this.alpha * 0.7;
    this.context.stroke();

    this.context.restore();
  }
},
```

Designing the stars

Draw the spikes

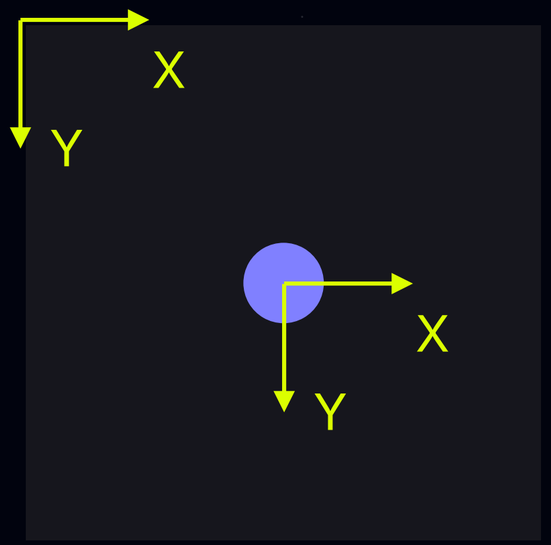


i = 0 ⇒ angle = 0°

```
1   drawStar() {
2     for (let i = 0; i < numSpikes; i<+) {
3       this.context.save();
4
5       const angle = (i * Math.PI) / (numSpikes / 2);
6       this.context.translate(positionX, positionY);
7       this.context.rotate(angle);
8       this.context.beginPath();
9       this.context.moveTo(0, 0);
10      this.context.lineTo(0, -spikeLength);
11      this.context.strokeStyle = color;
12      this.context.shadowColor = color;
13      this.context.shadowBlur = this.glowIntensity;
14      this.context.lineWidth = spikeWidth;
15      this.context.globalAlpha = this.alpha * 0.7;
16      this.context.stroke();
17
18      this.context.restore();
19    }
20  },
```

Designing the stars

Draw the spikes

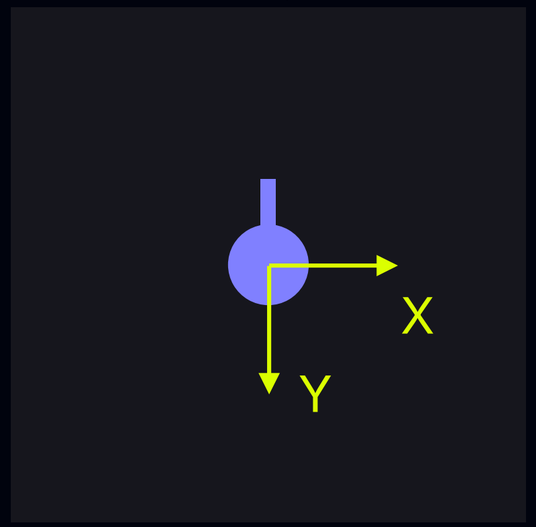


i = 0 ⇒ angle = 0°

```
1   drawStar() {
2     for (let i = 0; i < numSpikes; i<+) {
3       this.context.save();
4
5       const angle = (i * Math.PI) / (numSpikes / 2);
6       this.context.translate(positionX, positionY);
7       this.context.rotate(angle);
8       this.context.beginPath();
9       this.context.moveTo(0, 0);
10      this.context.lineTo(0, -spikeLength);
11      this.context.strokeStyle = color;
12      this.context.shadowColor = color;
13      this.context.shadowBlur = this.glowIntensity;
14      this.context.lineWidth = spikeWidth;
15      this.context.globalAlpha = this.alpha * 0.7;
16      this.context.stroke();
17
18      this.context.restore();
19    }
20  },
```

Designing the stars

Draw the spikes



i = 0 ⇒ angle = 0°

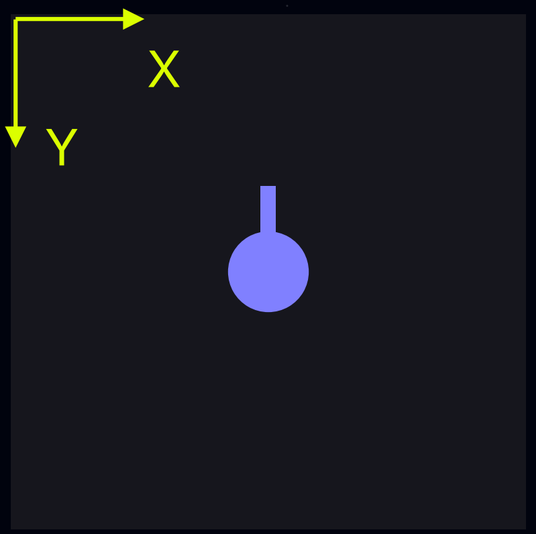
```
drawStar() {
  for (let i = 0; i < numSpikes; i<+) {
    this.context.save();

    const angle = (i * Math.PI) / (numSpikes / 2);
    this.context.translate(positionX, positionY);
    this.context.rotate(angle);
    this.context.beginPath();
    this.context.moveTo(0, 0);
    this.context.lineTo(0, -spikeLength);
    this.context.strokeStyle = color;
    this.context.shadowColor = color;
    this.context.shadowBlur = this.glowIntensity;
    this.context.lineWidth = spikeWidth;
    this.context.globalAlpha = this.alpha * 0.7;
    this.context.stroke();

    this.context.restore();
  }
},
```

Designing the stars

Draw the spikes

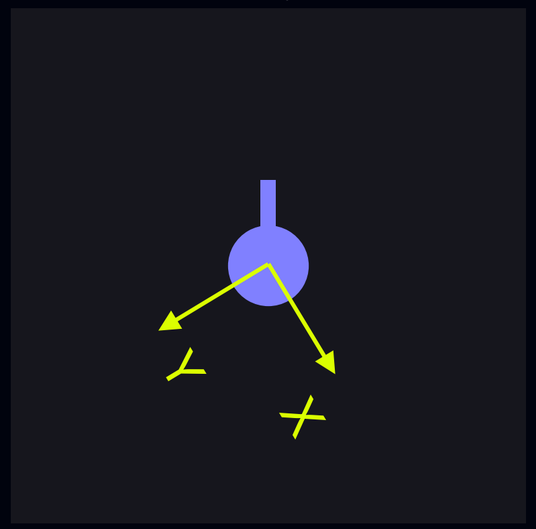


i = 0 ⇒ angle = 0°

```
1   drawStar() {
2     for (let i = 0; i < numSpikes; i<+) {
3       this.context.save();
4
5       const angle = (i * Math.PI) / (numSpikes / 2);
6       this.context.translate(positionX, positionY);
7       this.context.rotate(angle);
8       this.context.beginPath();
9       this.context.moveTo(0, 0);
10      this.context.lineTo(0, -spikeLength);
11      this.context.strokeStyle = color;
12      this.context.shadowColor = color;
13      this.context.shadowBlur = this.glowIntensity;
14      this.context.lineWidth = spikeWidth;
15      this.context.globalAlpha = this.alpha * 0.7;
16      this.context.stroke();
17
18      this.context.restore();
19    }
20  },
```

Designing the stars

Draw the spikes

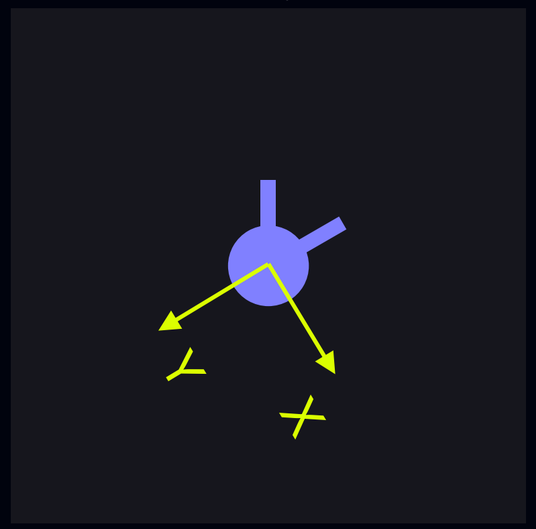


i = 1 ⇒ angle = 60°

```
1   drawStar() {
2     for (let i = 0; i < numSpikes; i<+) {
3       this.context.save();
4
5       const angle = (i * Math.PI) / (numSpikes / 2);
6       this.context.translate(positionX, positionY);
7       this.context.rotate(angle);
8       this.context.beginPath();
9       this.context.moveTo(0, 0);
10      this.context.lineTo(0, -spikeLength);
11      this.context.strokeStyle = color;
12      this.context.shadowColor = color;
13      this.context.shadowBlur = this.glowIntensity;
14      this.context.lineWidth = spikeWidth;
15      this.context.globalAlpha = this.alpha * 0.7;
16      this.context.stroke();
17
18      this.context.restore();
19    }
20  },
```

Designing the stars

Draw the spikes



i = 1 ⇒ angle = 60°

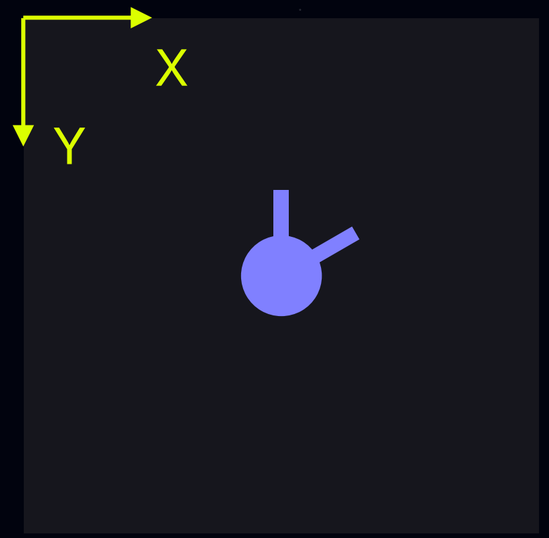
```
drawStar() {
  for (let i = 0; i < numSpikes; i<+) {
    this.context.save();

    const angle = (i * Math.PI) / (numSpikes / 2);
    this.context.translate(positionX, positionY);
    this.context.rotate(angle);
    this.context.beginPath();
    this.context.moveTo(0, 0);
    this.context.lineTo(0, -spikeLength);
    this.context.strokeStyle = color;
    this.context.shadowColor = color;
    this.context.shadowBlur = this.glowIntensity;
    this.context.lineWidth = spikeWidth;
    this.context.globalAlpha = this.alpha * 0.7;
    this.context.stroke();

    this.context.restore();
  }
},
```

Designing the stars

Draw the spikes



i = 1 ⇒ angle = 60°

```
1   drawStar() {
2     for (let i = 0; i < numSpikes; i<+) {
3       this.context.save();
4
5       const angle = (i * Math.PI) / (numSpikes / 2);
6       this.context.translate(positionX, positionY);
7       this.context.rotate(angle);
8       this.context.beginPath();
9       this.context.moveTo(0, 0);
10      this.context.lineTo(0, -spikeLength);
11      this.context.strokeStyle = color;
12      this.context.shadowColor = color;
13      this.context.shadowBlur = this.glowIntensity;
14      this.context.lineWidth = spikeWidth;
15      this.context.globalAlpha = this.alpha * 0.7;
16      this.context.stroke();
17
18      this.context.restore();
19    }
20  },
```

Designing the stars

Draw the spikes

```
drawStar() {
  for (let i = 0; i < numSpikes; i<+) {
    this.context.save();

    const angle = (i * Math.PI) / (numSpikes / 2);
    this.context.translate(positionX, positionY);
    this.context.rotate(angle);
    this.context.beginPath();
    this.context.moveTo(0, 0);
    this.context.lineTo(0, -spikeLength);
    this.context.strokeStyle = color;
    this.context.shadowColor = color;
    this.context.shadowBlur = this.glowIntensity;
    this.context.lineWidth = spikeWidth;
    this.context.globalAlpha = this.alpha * 0.7;
    this.context.stroke();

    this.context.restore();
  }
},
```

Star field

Code example