

function drawSnow

Canvas innerwidth

function spw

function drawCk

let horizon =
ctx.canvas.height * 0.6

min

max

*

Canvas height

under Schneemann

birdhouse

- positive vector
 $x: 150$
 $y: 200$

ctx. save

translate(-posx posy)

beginPath()

ctx.beginPath()

moveTo

lineTo

closePath()

fillStyle = "brown"

restore

ctx.arc(0, 250, 150, 2 * Math.PI)

ctx.fillStyle = "black";

c(10)

lock {
x: 150
y: 200
}

ctx.save()

```
drawMountains  
- position: vector
```

```
- min: number
```

```
- max: number
```

```
- colorLow: string
```

```
- colorHigh: string
```



let bidPax
drawBirds()

i: number
X - [i < 10] -> let bid pax
increase 1by1

```
stepMin: 60,  
stepMax: 150,  
let x: number = 0
```

```
drawBirds ( )
```

```
saveTransform  
translate(-position.x, -position.y)
```

```
begInPath()  
ctxMove(0, 0)
```

```
ctxLineTo(0, -max)
```

```
ctxMove(0, 0), ctxLine(0, 0)
```

```
ctxSave  
ctxTranslate(-position.x, -position.y)
```

```
ctxFill()
```

```
ctxFillStyle = "hslc" + Math.random() * 180
```

```
x += stepMin + Math.random(stepMax - stepMin);  
let y: number = min - Math.random() * max - y
```

```
↓
```

```
lineColor()
```

```
↓
```

```
↓
```

```
gtxBeginPath()
```

```
↑
```

```
↑
```

```
ctf.lineTo( - - - )
```

```
ctf.fillStyle = "white"
```

```
↓
```

```
lineTo()
```

```
↓
```

```
↓
```

```
closePath()
```

```
↓
```

```
↓
```

```
restoreTransform()
```

```
↓
```

```
↓
```

```
drawPath()
```

```
↓
```

```
↓
```

Schneemann

[position: Vector]

ctx.save

ctx.translate(-positionx, -positiony)

ctx.beginPath()

ctx.arc(0, -5 / 1 = 80; r=2 = 100; r=150)

ctx.closePath

ctx

ctx

ctx.fillStyle = "black"

ctx.beginPath()

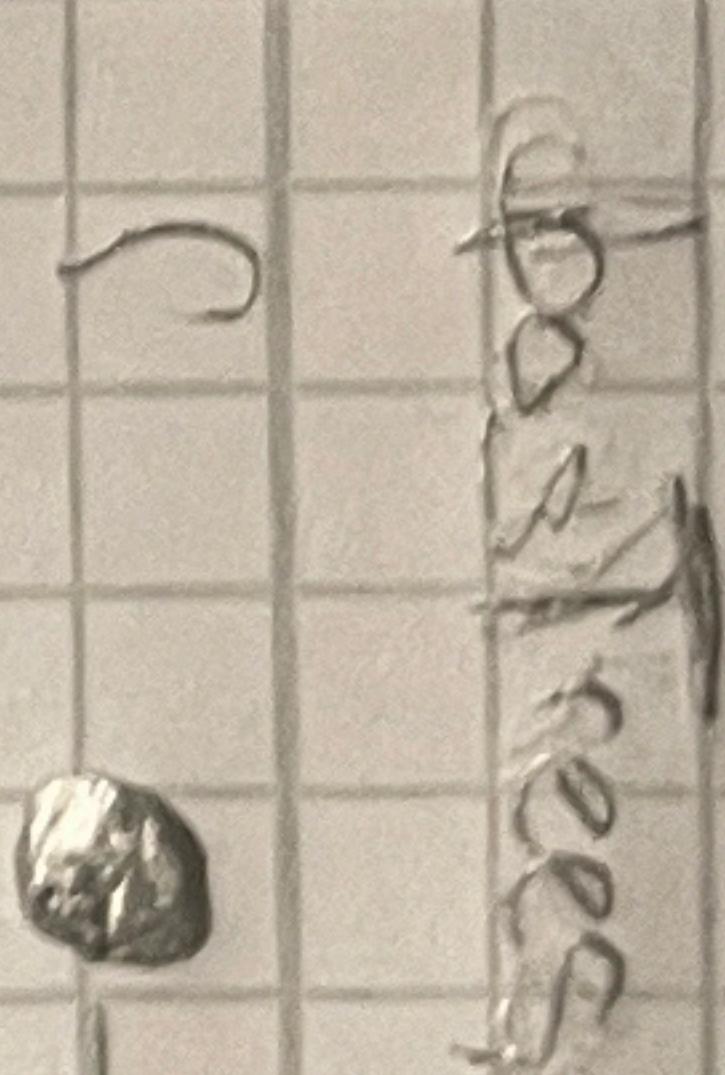
ctx.arc(0, 0, 100, 2 * Math.PI)

ctx.beginPath()

ctx.arc(0, 0, 100, 2 * Math.PI)

ctx.beginPath()

ctx.arc(0, 0, 100, 2 * Math.PI)



draws

(x: position
y: stem height)

past trees

11.11