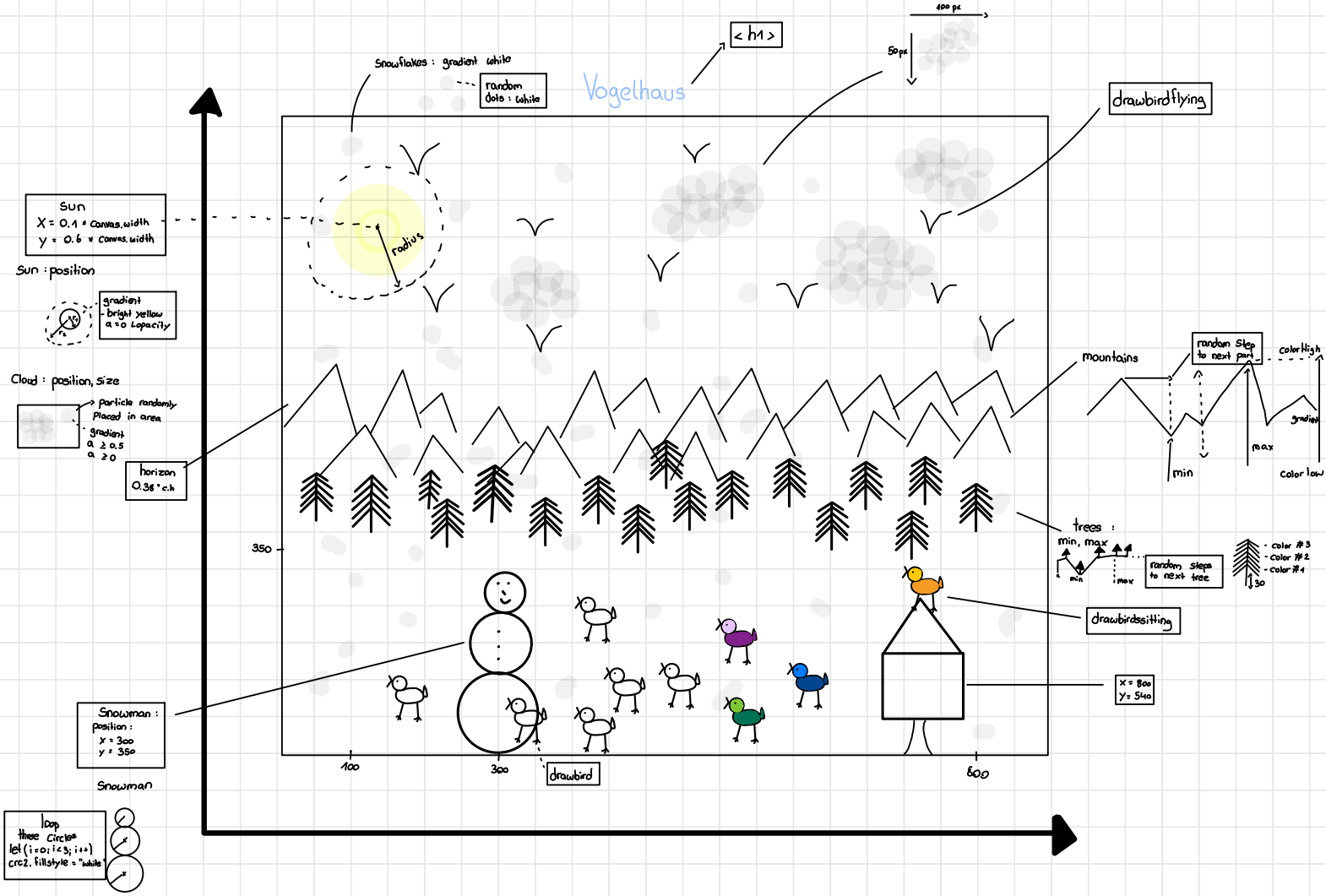
UI - Scribble ~ Vogelhaus



drawbirdflying

position x : Mathrandom (900 - 40) + 40

position y : Mathrandom (200 - 20) + 20

randomscale : Mathrandom (3 - 0,5 + 0,5)

```
Vogelhaus.crc2.moveTo(1,0)
Vogelhaus.crc2.bezierCurveTo(8,-5,15,-10,20,-2)
Vogelhaus.crc2.moveTo(-1,0)
Vogelhaus.crc2.bezierCurveTo(-8,-5,-15,-10,-20,-2)
```

drawbirdsitting

let position x : number = 800;

let position y : number = 34 ;


```
let radius : number = 10
crc2.arc(positionx - 10, positiony - 10, radius, 0.2 * Math.PI)
Vogelhaus.crc2.fillstyle = randomColor();
```

```
let radius3 = 1
crc2.arc(positionx - 11, positiony - 11, radius3, 0.2 * Math.PI)
Vogelhaus.crc2.fillstyle = "black";
```

```
let radius = 12
Vogelhaus.crc2.arc(positionx - 10, positiony - 10, radius, 0.2 * Math.PI);
Vogelhaus.crc2.fillstyle = randomColor()
```

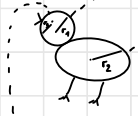
drawbird

maxwidth : 800;

minwidth : 100;

maxHeight : 515;

minHeight : 530;

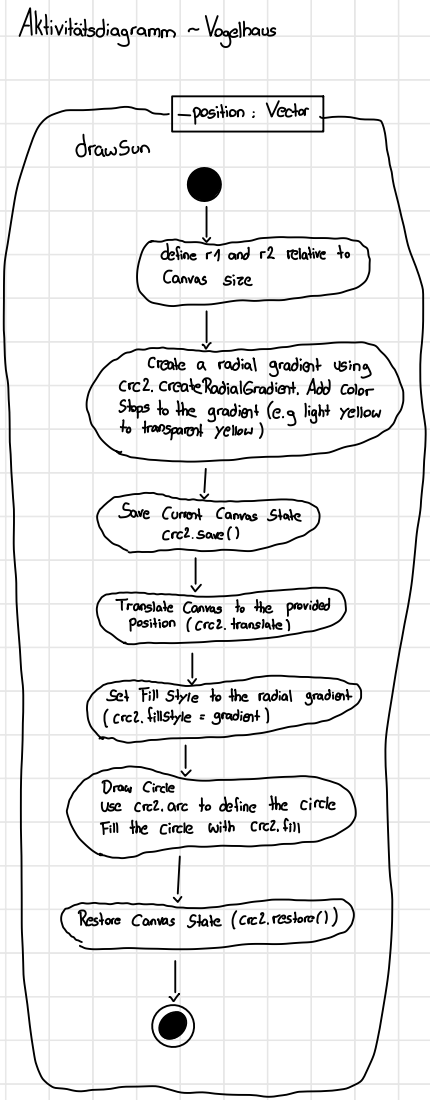


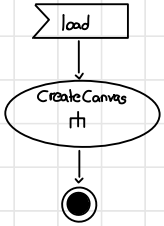

```
let radius = 10
Vogelhaus.crc2.arc(positionx - 10, positiony - 10, radius, 0.2 * Math.PI)
Vogelhaus.crc2.fillstyle = randomColor();
```

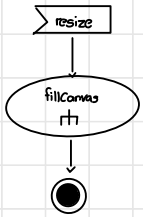
```
let radius2 = 15
Vogelhaus.crc2.arc(positionx, positiony, radius2, 0.2 * Math.PI)
Vogelhaus.crc2.fillstyle = randomColor();
```

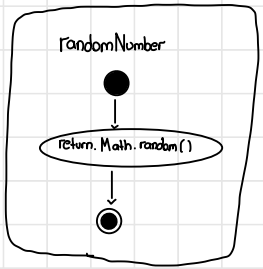
```
let radius3 = 1
Vogelhaus.crc2.arc(positionx - 11 positiony - 11, radius3, 0.2 * Math.PI);
Vogelhaus.crc2.fillStyle = "black";
```

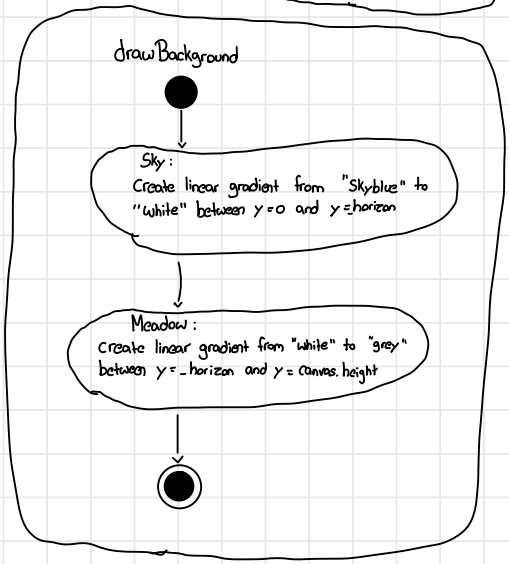
Aktivitätsdiagramm ~ Vogelhaus

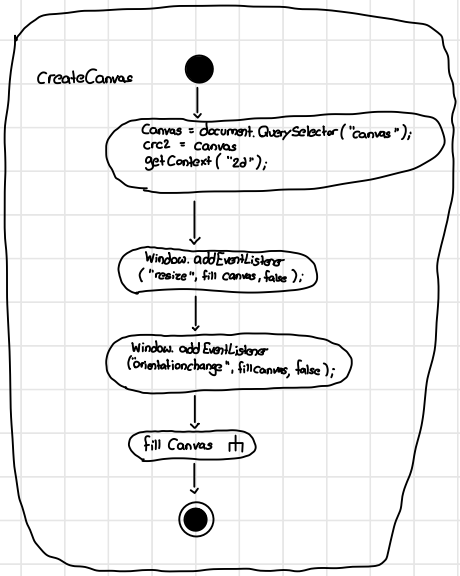


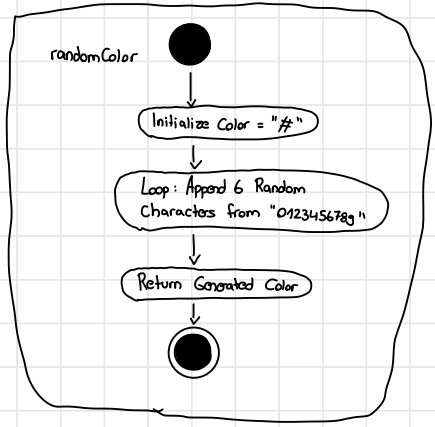


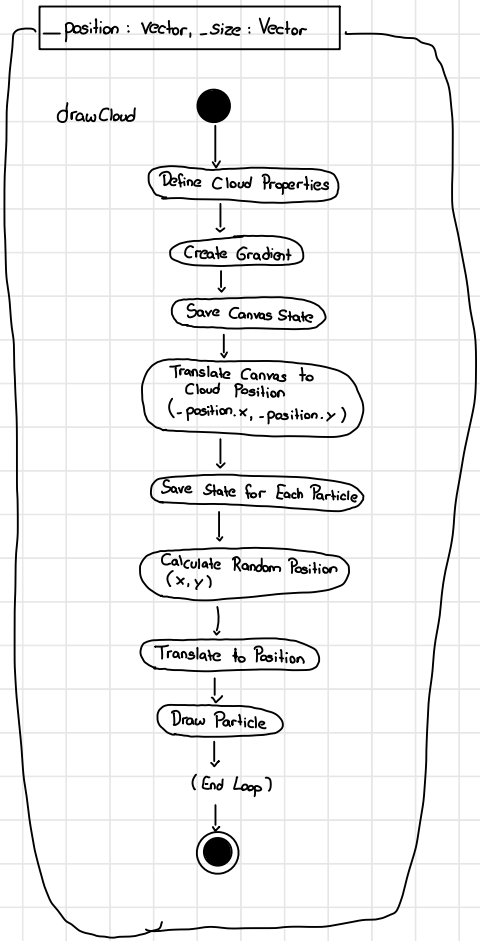


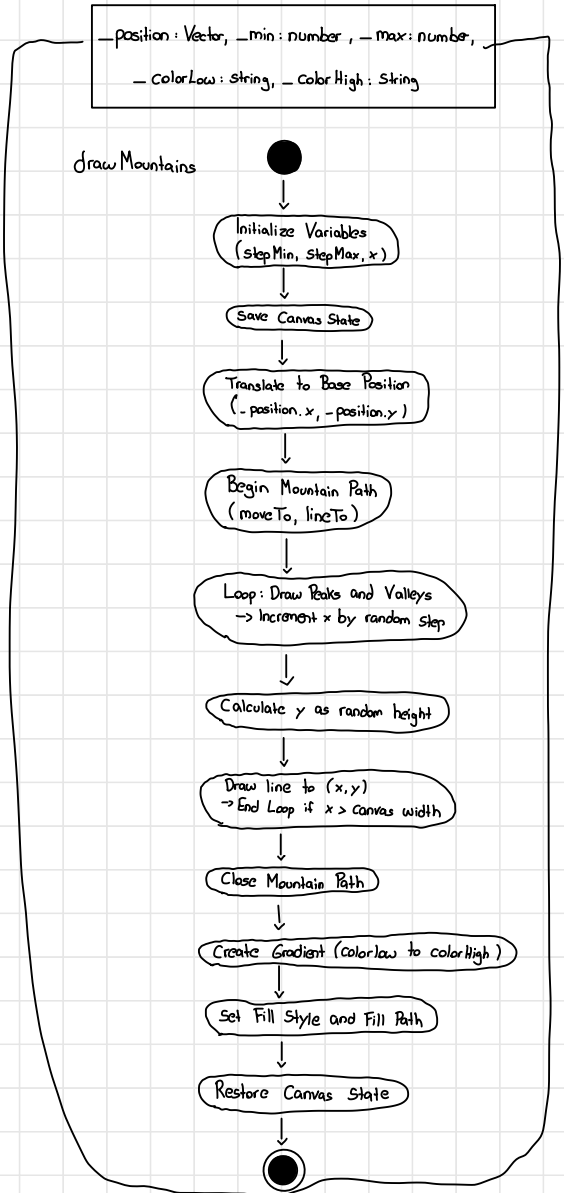


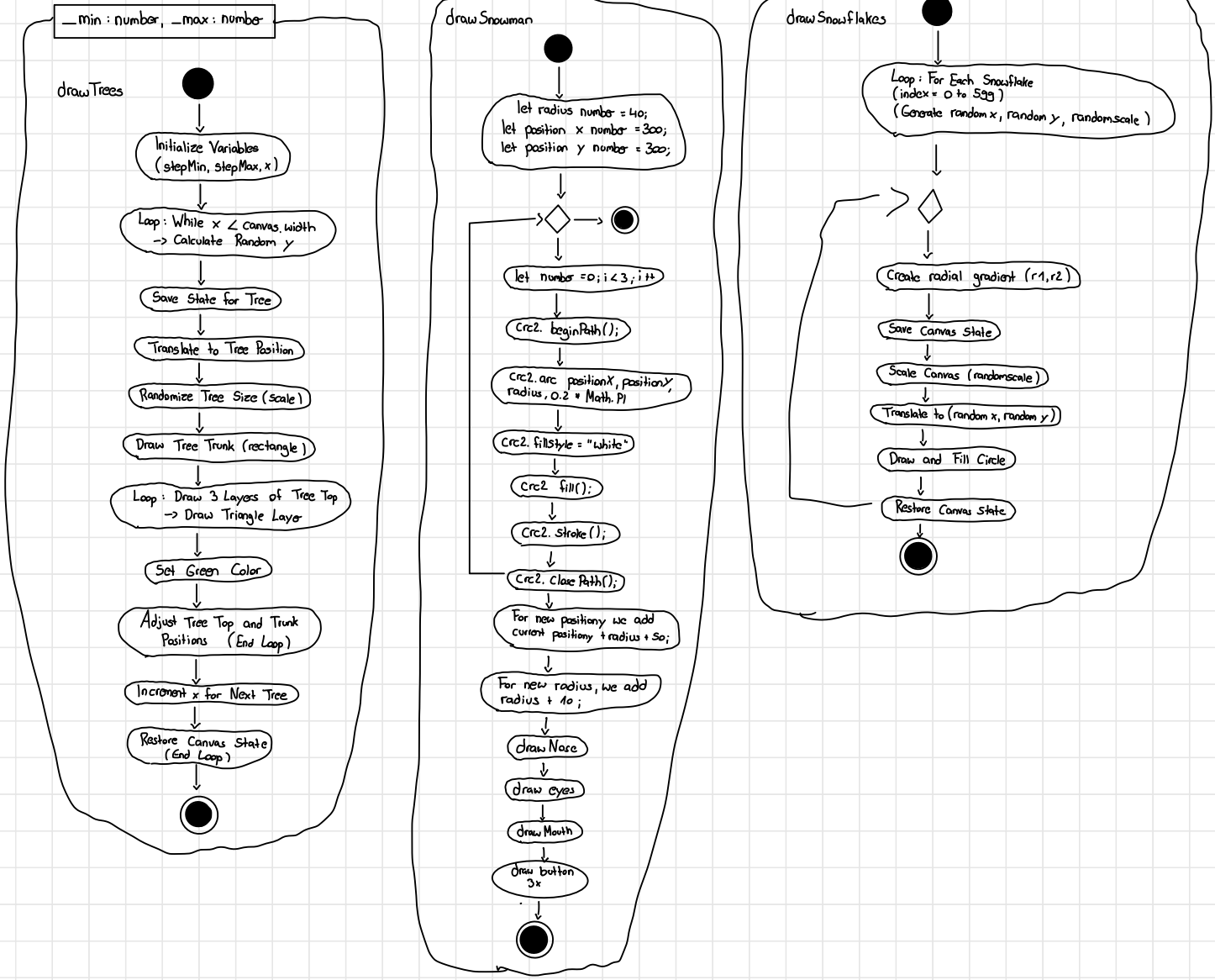


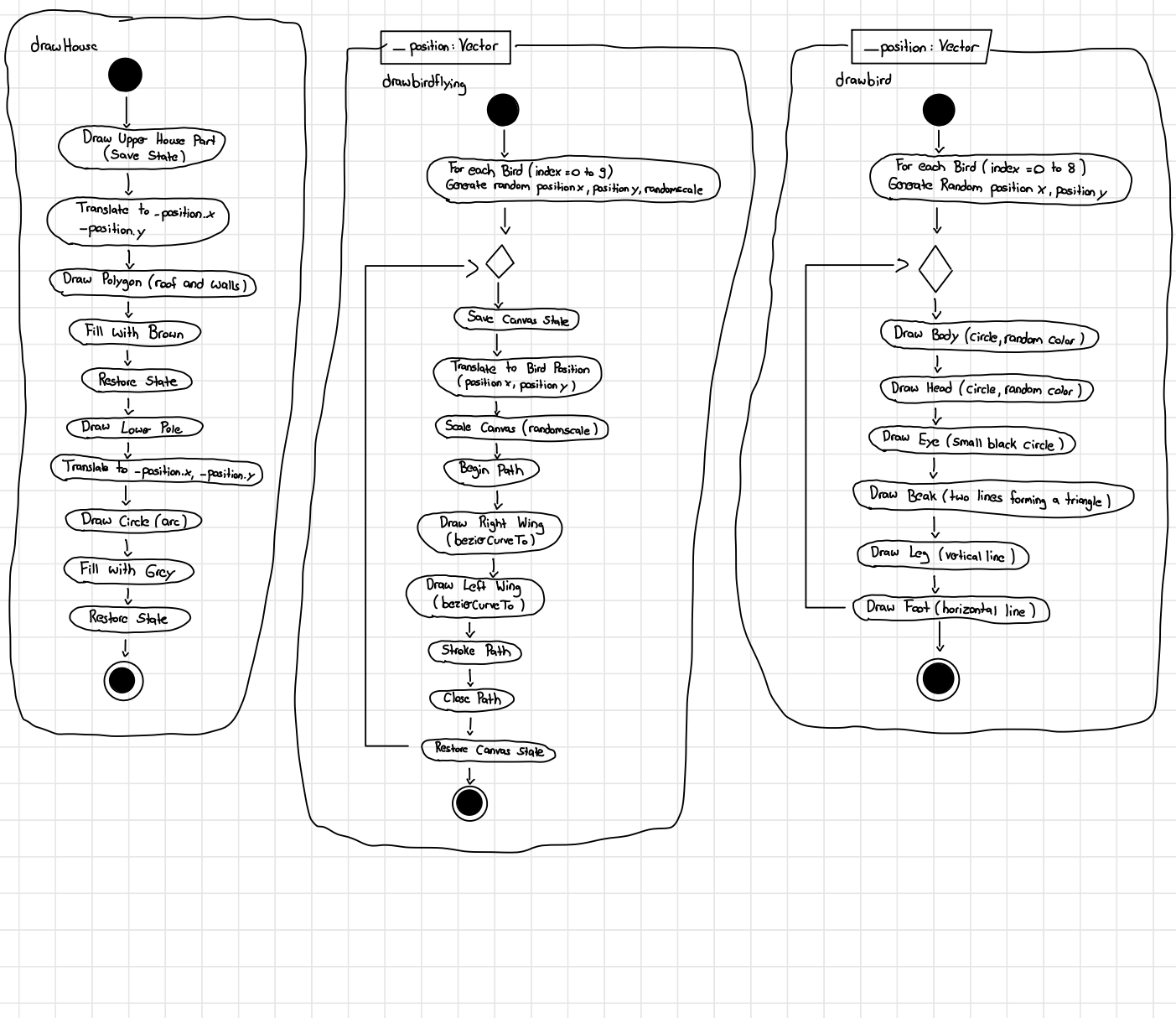


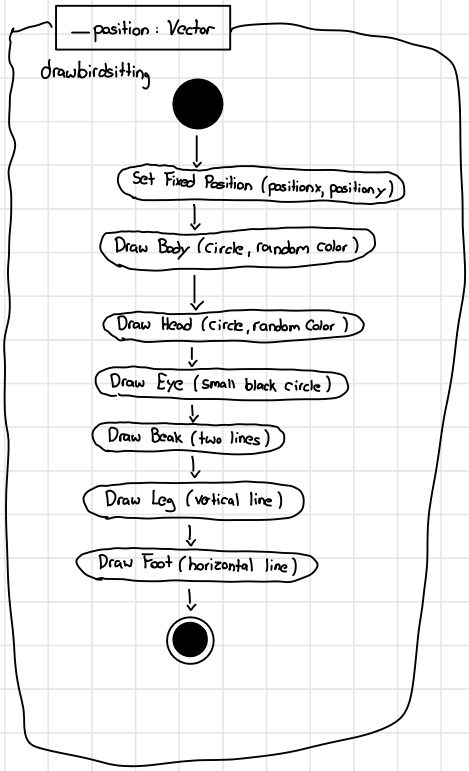



_position : Vector, _size : Vector
drawCloud
Define Cloud Properties
Create Gradient
Save Canvas State
Translate Canvas to Cloud Position (-position.x, -position.y)
Save State for Each Particle
Calculate Random Position (x, y)
Translate to Position
Draw Particle
(End Loop)


_position : Vector, _min : number , _max : number,
_colorLow : string, _colorHigh : string
drawMountains
Initialize Variables (stepMin, stepMax, x)
Save Canvas State
Translate to Base Position (-position.x, -position.y)
Begin Mountain Path (moveTo, lineTo)
Loop : Draw Peaks and Valleys
-> Increment x by random step
Calculate y as random height
Draw line to (x, y)
-> End Loop if x > canvas width
Close Mountain Path
Create Gradient (colorLow to colorHigh)
Set Fill Style and Fill Path
Restore Canvas State


_min : number, _max : number
drawTrees
Initialize Variables (stepMin, stepMax, x)
Loop : While x < canvas.width -> Calculate Random y
Save State for Tree
Translate to Tree Position
Randomize Tree Size (scale)
Draw Tree Trunk (rectangle)
Loop : Draw 3 Layers of Tree Top -> Draw Triangle Layer
Set Green Color
Adjust Tree Top and Trunk Positions (End Loop)
Incroment x for Next Tree
Restore Canvas State (End Loop)
drawSnowman
let radius number = 40;
let position x number = 300;
let position y number = 300;
let number =0; i<3; i++
Crc2. beginPath();
Crc2. arc positionX, positionY, radius, 0.2 * Math. PI
Crc2. fillStyle = "white"
Crc2 fill();
Crc2. Stroke();
Crc2. Clase Path();
For new positiony we add current positiony + radius + 50;
For new radius, we add radius + 10;
drawNose
draw eyes
drawMouth
draw button 3x
drawSnowflakes
Loop : For Each Snowflake (index = 0 to 599) (Generate random x, random y, random.scale)
Create radial gradient (r1,r2)
Save Canvas State
Scale Canvas (randomscale)
Translate to (random x, random y)
Draw and Fill Circle
Restore Canvas State


drawHouse
Draw Upper House Part (Save State)
Translate to -position.x -position.y
Draw Polygon (roof and walls)
Fill with Brown
Restore State
Draw Lower Pole
Translate to -position.x, -position.y
Draw Circle (arc)
Fill with Grey
Restore State
_position : Vector
drawbirdflying
For each Bird (index =0 to 9) Generate random positionx, positiony, randomscale
Save Canvas State
Translate to Bird Position (positionx, positiony)
Scale Canvas (randomscale)
Begin Path
Draw Right Wing (bezierCurveTo)
Draw Left Wing (bezierCurveTo)
Stroke Path
Close Path
Restore Canvas State
_position : Vector
drawbird
For each Bird (index =0 to 8) Generate Random positionx, positiony
Draw Body (circle, random color)
Draw Head (circle, random color)
Draw Eye (small black circle)
Draw Beak (two lines forming a triangle)
Draw Leg (vertical line)
Draw Feet (horizontal line)


__position : Vector
drawBirdsitting
Set Fixed Position (positionx, positiony)
Draw Body (Circle, random color)
Draw Head (Circle, random Color)
Draw Eye (small black circle)
Draw Beak (two lines)
Draw Leg (vertical line)
Draw Foot (horizontal line)