

Class Diagram :

CanvasRenderingContext ←

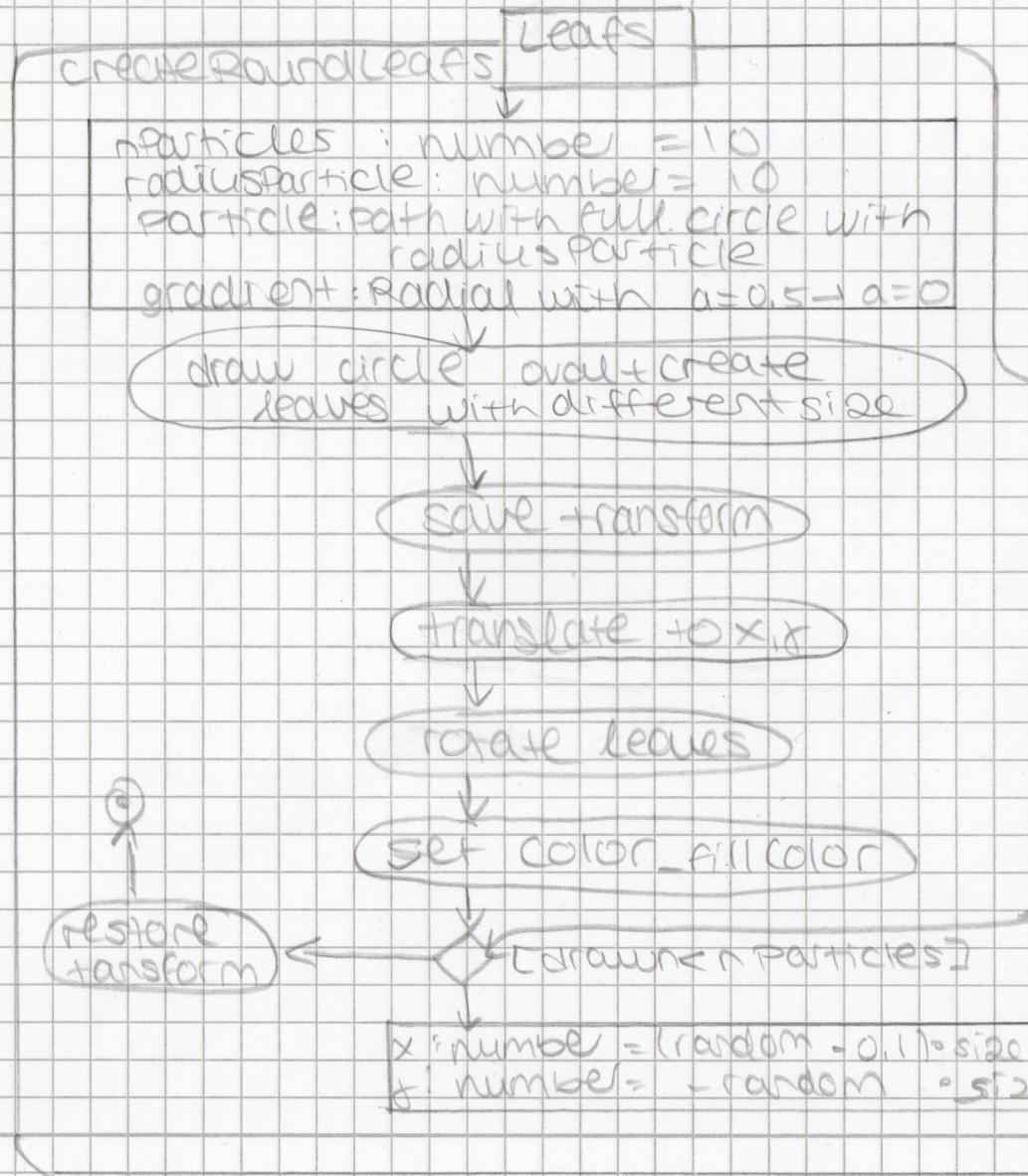
Leafs
x: number
y: number
size: number
fillColor: string
speedLeafs: number
radiusParticle: number
CreateRoundLeafs()
 ||
 ||
 ||

Cloud
x: number
y: number
speedCloud: number
CreateClouds()
moveForward()

Squirrel
x: number
y: number
speedSquirrel: number
CreateSquirrel()
moveSquirrel()

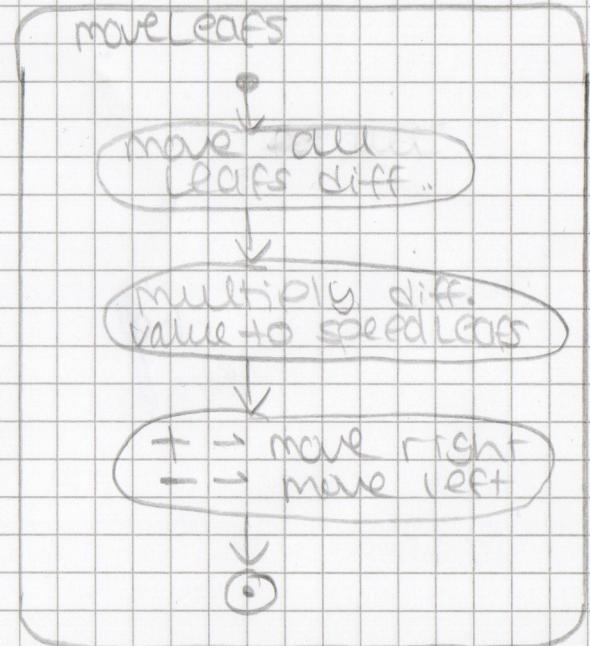
GoldenerHerbst
x: number
y: number
constructor()
drawBackground()
drawSun (-position: Vector): void
createGrass (-position: Vector, -size: Vector): void
drawMountains(-position: Vector, -min: number, -max: number,
 -colorLow: string, -colorHigh: string): void
drawTree (-x: number, -r: number): void
drawAnotherTree (-position: Vector, -size: Vector, -fillColor: string): void
drawFlowers(-position: Vector, -fillColor: string): void

Leafs: Activity Diagram

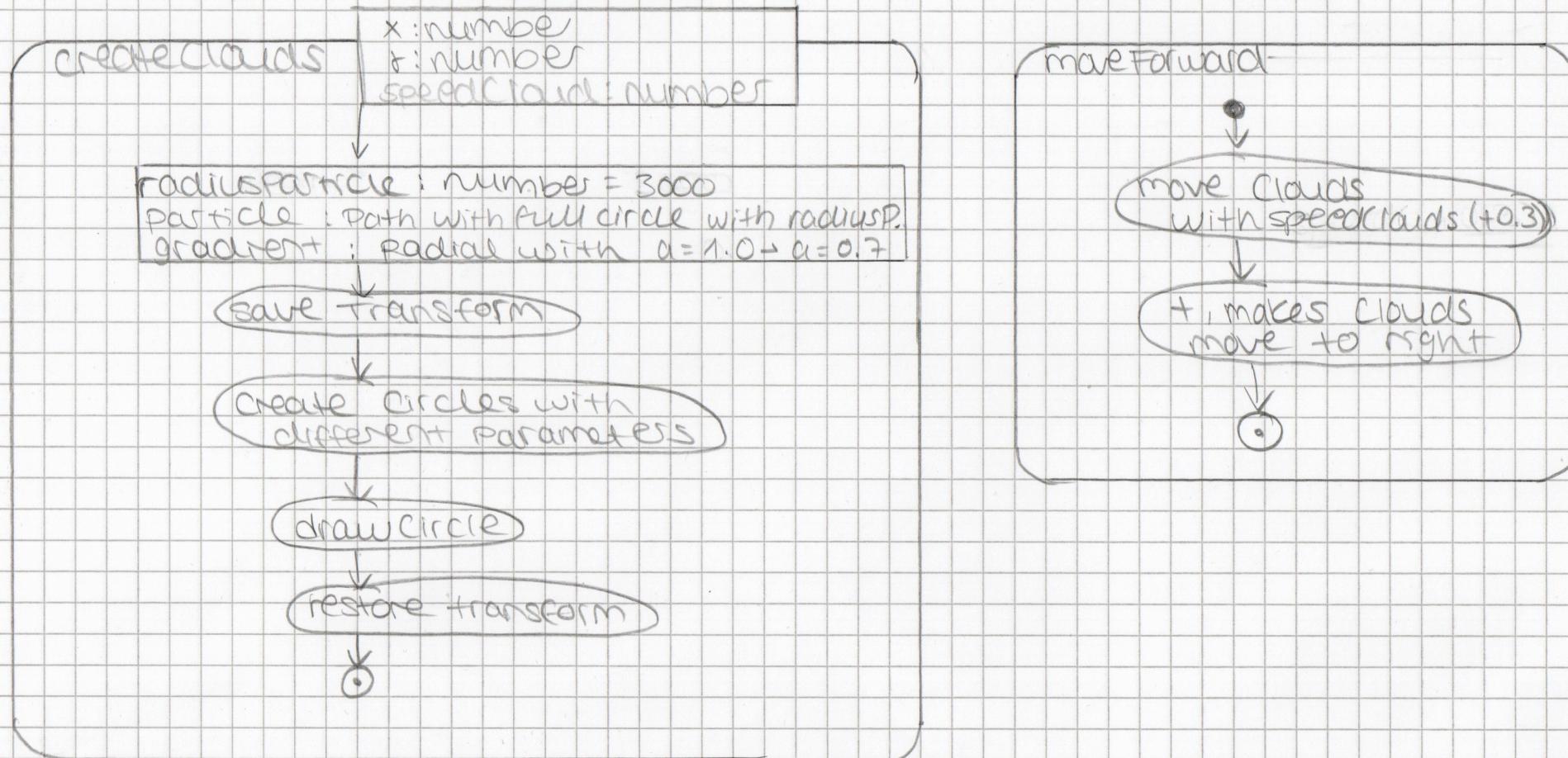


Leafs

<code>x : number</code>
<code>y : number</code>
<code>size : number</code>
<code>fillcolor : string</code>
<code>speedLeafs : number</code>
<code>radiusParticle : number</code>



Clouds : Activity Diagram



Squirrel : Activity Diagram

create Squirrel

$x: \text{number}$
 $y: \text{number}$
 $\text{speedsquirrel: number}$

save transform

set color with fillstyle

draw circles

create ear +
set color for ear

set $x + y$
for ear

draw ear

create eye + mouth
+ set color to black

Set $x + y$
for eye + mouth

draw eye + mouth

move Squirrel

move Squirrel
with speedSquirrel

multiply speedSquirrel
with (-0.2)

Squirrel will move
from right to left

restore transform

Goldener Herbst : Activity Diagram

constructor

set golden to 0.65
let horizon-canvas.height = golden

call all functions
with this and
set x & y for
all if needed

set color for
Mountains, AnotherTree,
Flowers

interface vector:

x: number

y: number

drawBackground

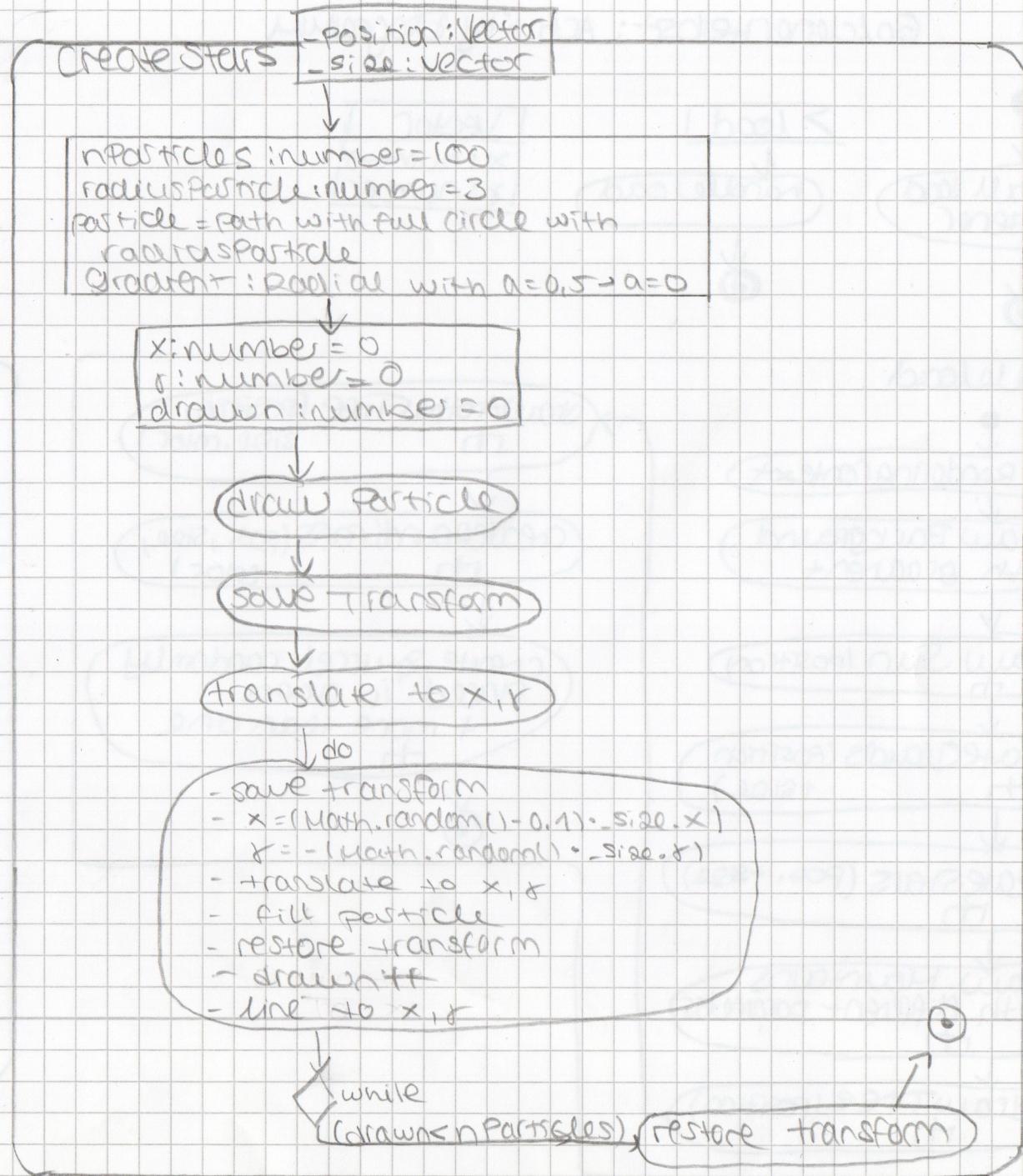
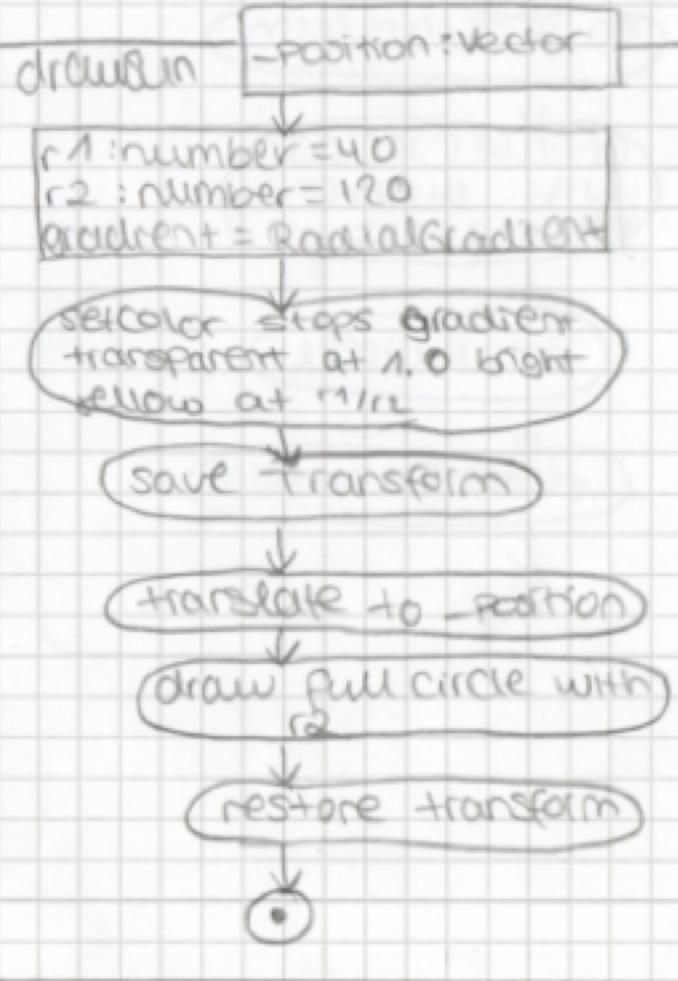
let gradient (0,0,10,400)

addColorStop to
darkblue at 0

at 0.7 → lightblue
+ 0.9 → white
+ 1.0 → grassgreen

fillStyle = gradient

fillRect(0,0,1000,1000)



F

drawMountains

```

- position: Vector
- min: number
- max: number
- colorLow: string
- colorHigh: string
  
```

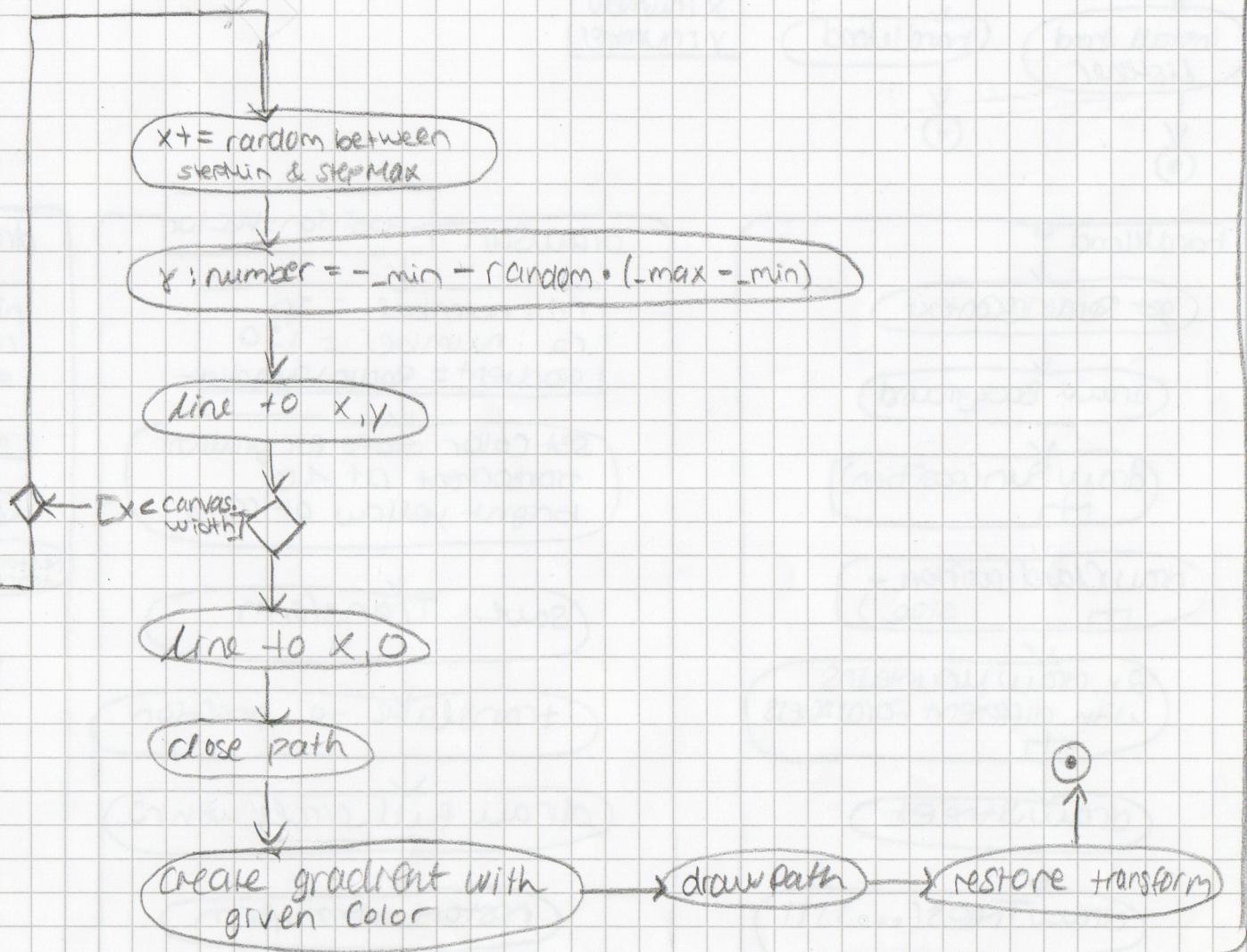
$\text{stepMin: number} = 10$
 $\text{stepMax: number} = 50$
 $\text{x: number} = 0$

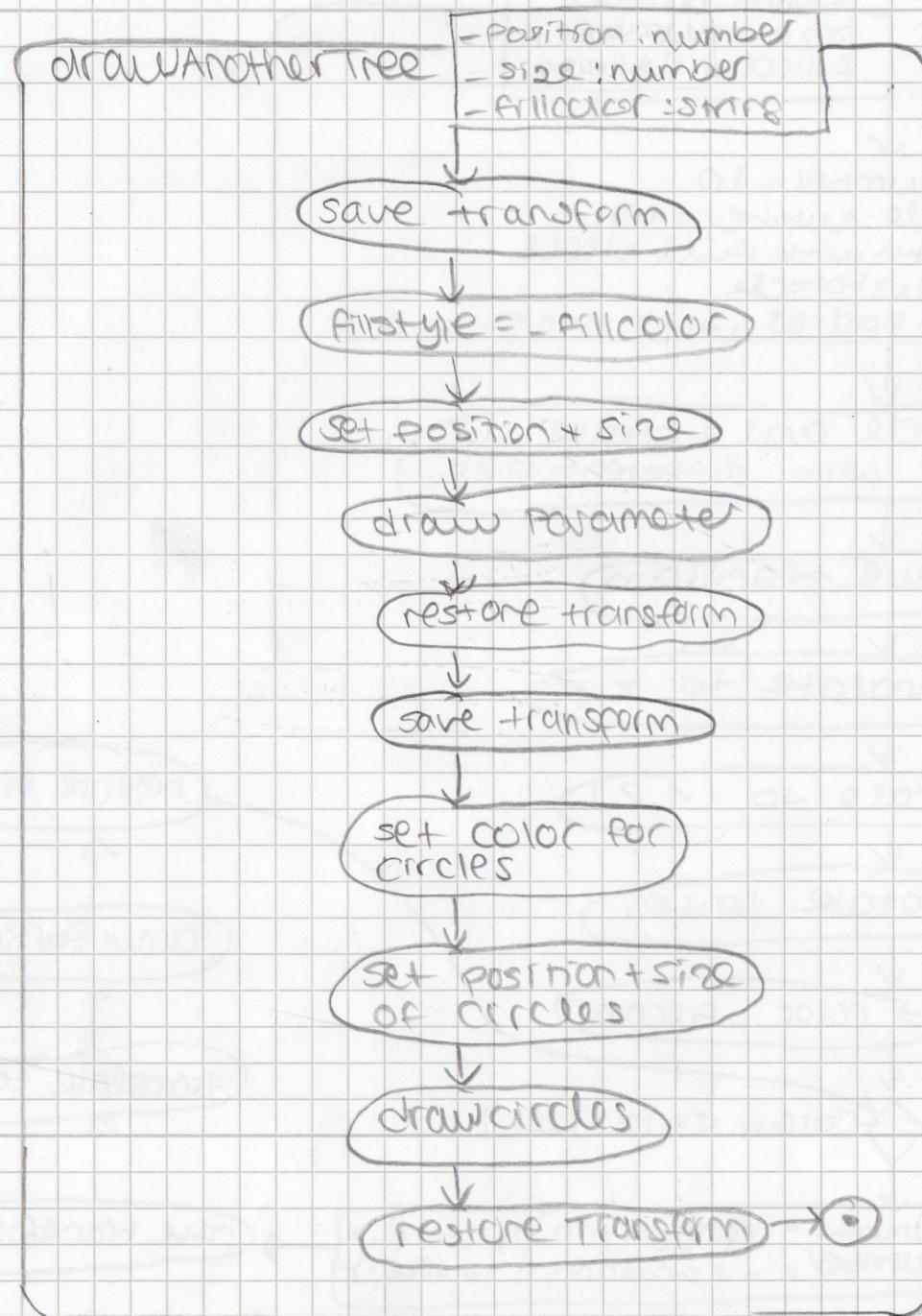
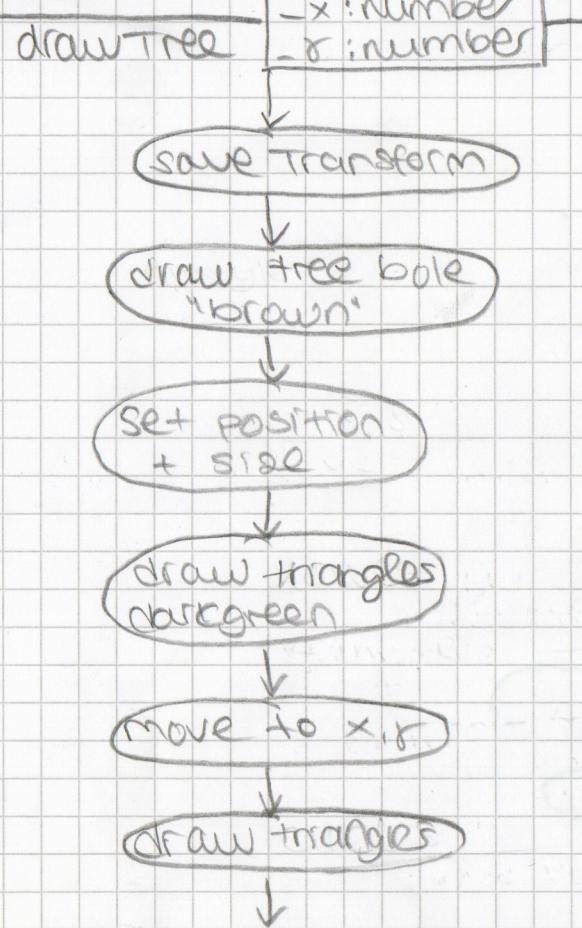
save - transform

translate to -position

move to 0,0

line to 0, -max





drawFlower

position : Vector
fillcolor : string

save Transform

translate to
position

strokeStyle to
"darkgreen"

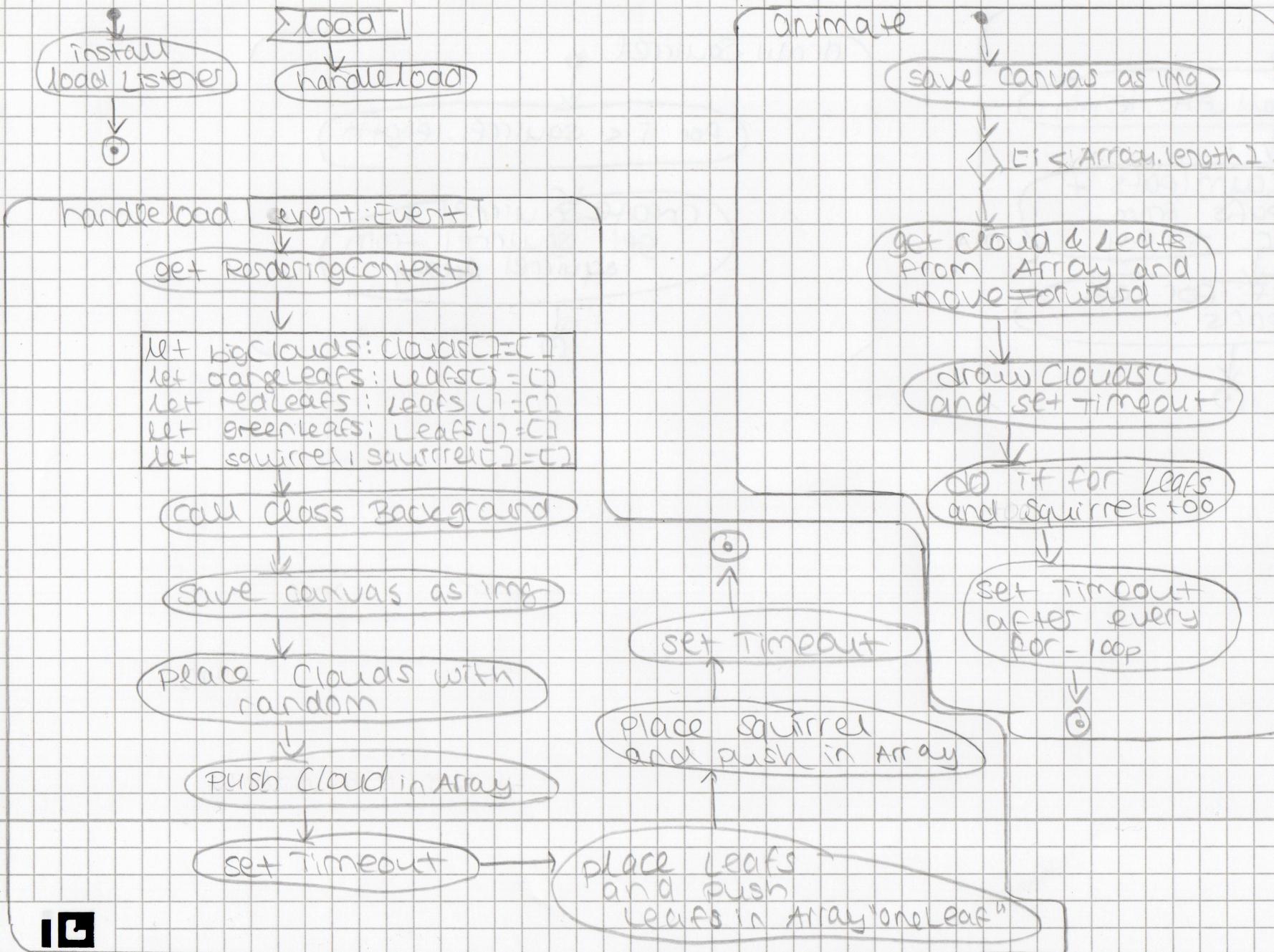
create stem with
moveTo & lineTo

fillStyle to
brown

create single
leaves +
add color

Create another
Flowers with
other color
and different
positions

Main : Activity Diagram



drawLeaves



for i < leaves.length



createLeaf +
get leafs from
their array

do it for all
leaves



draw Squirrel



for i < squirrel.length

createSquirrel +
get squirrel from
squirrel array

