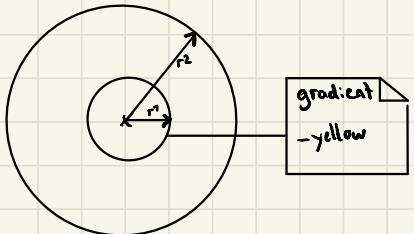
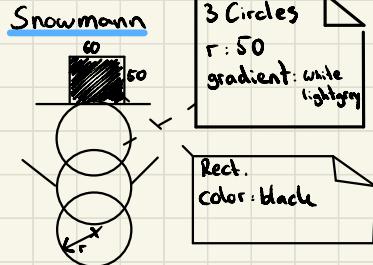


SUN



gradient
- yellow

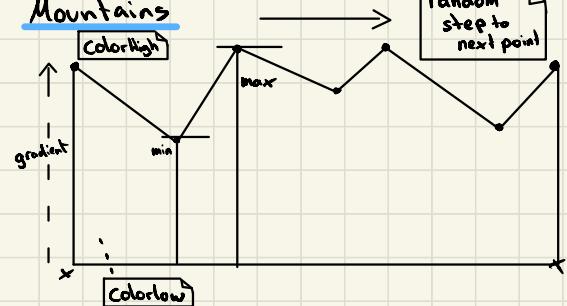


Snowmann

3 Circles
 $r: 50$
gradient: white
lightgray

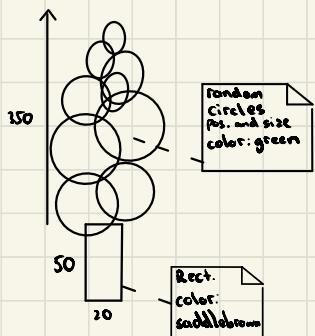
Rect.
color: black

Mountains



random
step to
next point

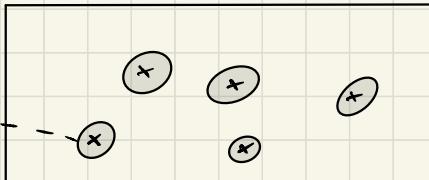
Tree



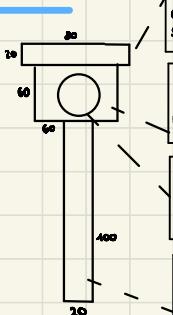
Random
Circles
Pos. and size
color: green

Snow flakes

particles
random
position



Birdhouse



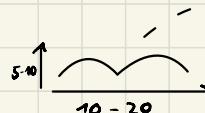
Rect.
color:
saddlebrown

Rect.
color:
burlybrown

Circle
 $r: 45$
color:
black

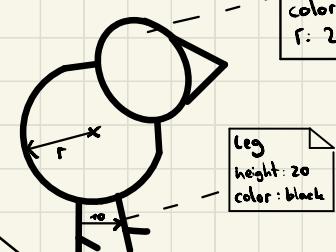
Rect.
color:
saddlebrown

Flying Birds



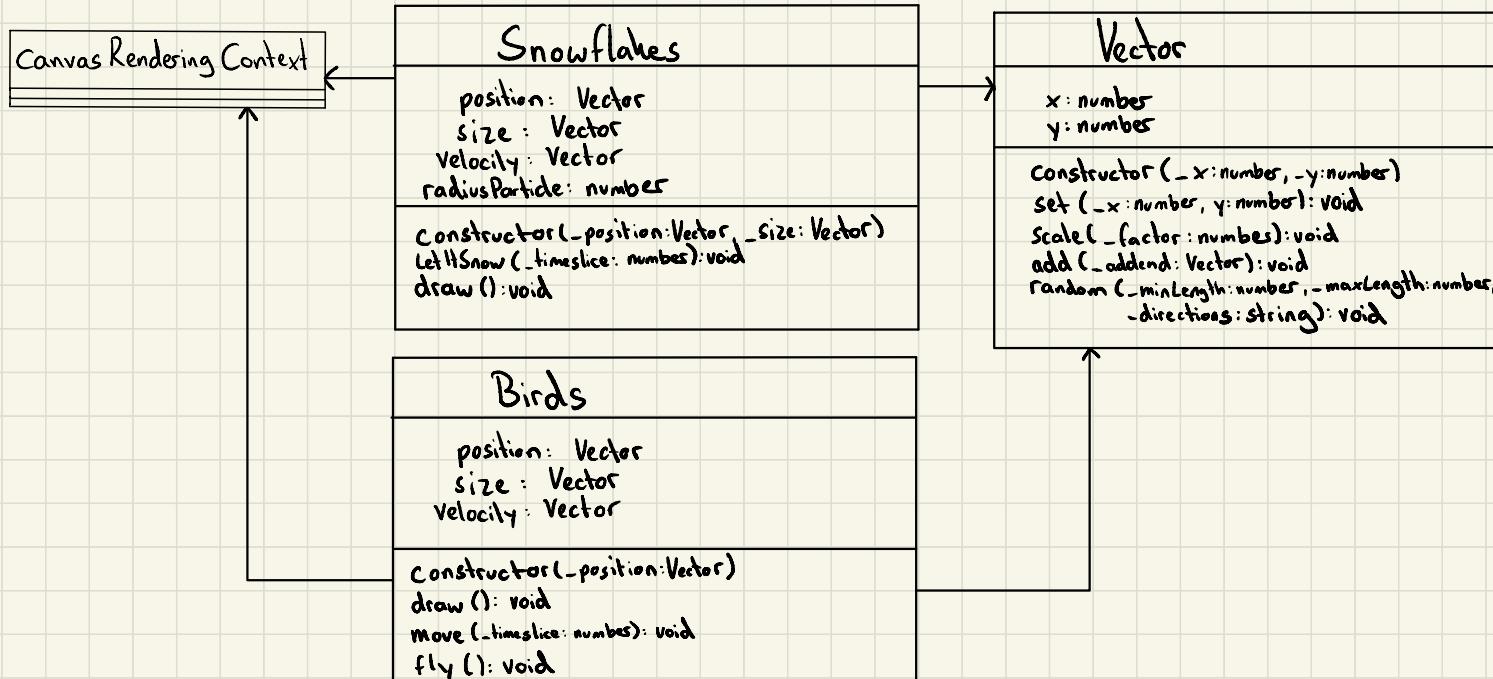
stroke black
2 Bezier
scale: random

Bird

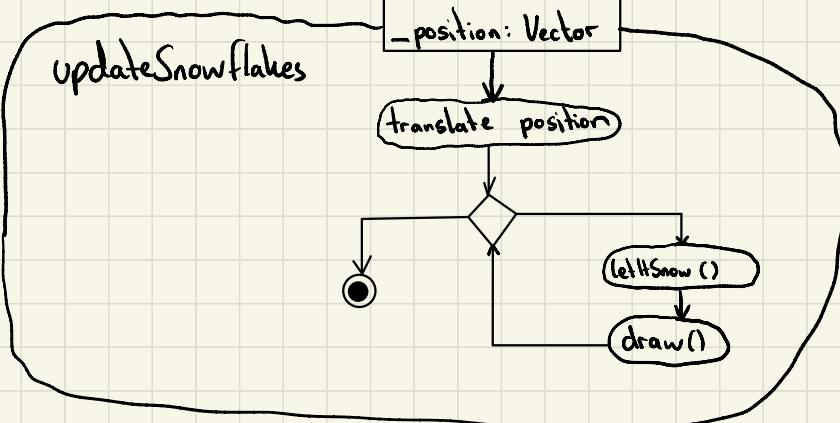
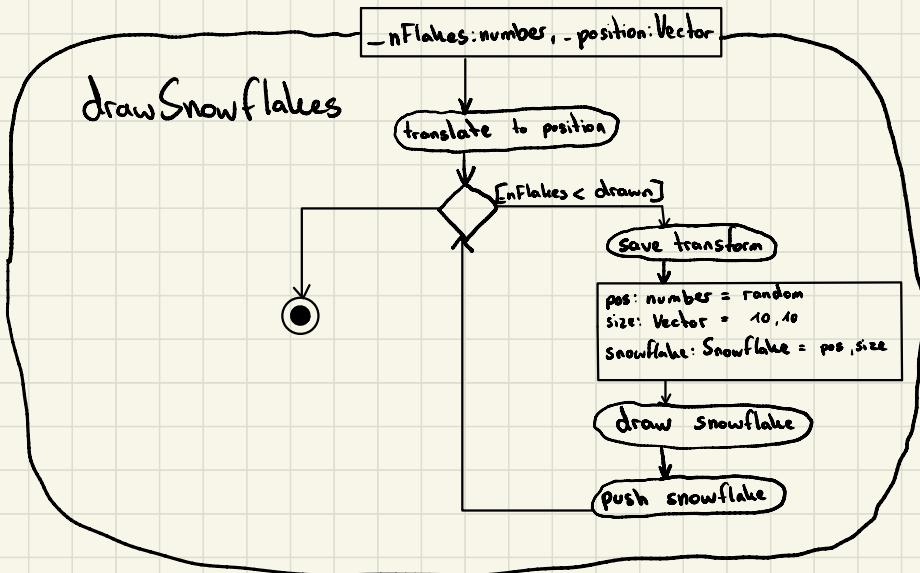
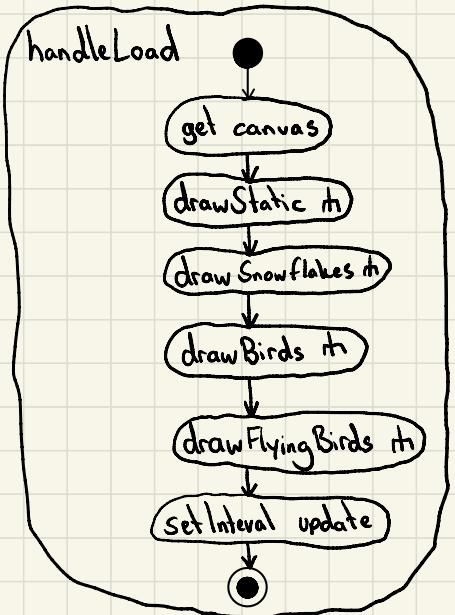
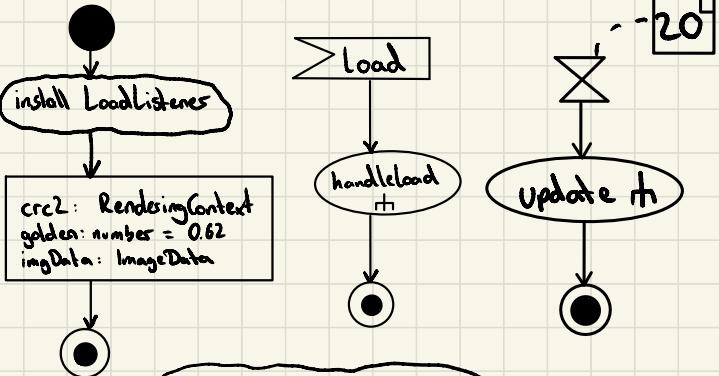


2 Circles
color: random
 $r: 20$

L09.2 - Klassendiagramm



Aktivitätsdiagramm: main



draw Birds

- nBirds: number, - position: Vector

translate to position

[nBirds < drawn]

Save transform

pos: number = random
size: Vector = 20, 20
bird: Bird = pos, size

draw Bird

push Bird

updateBird

Console "update Bird"

draw Flying Birds

- nFBirds: number, - position: Vector

translate to position

[nFBirds < drawn]

Save transform

pos: number = random
size: Vector = 10, 10
flyingBird: FlyingBird = pos, size

draw flyingBird

push flyingBird

Random Number

- min: number
- max: number

returnNumber: number = Math.floor(Math.random() · (-max - min)) + min

return returnNumber

Aktivitätsdiagramm: background

drawStatic

```

horizon: number = canvas.height * golden
sunPos: Vector = 400, 125
mountains: Vector = 0, horizon
cloudPos: Vector = 550, 125
cloudSize: Vector = 250, 75
treePos: Vector = 600, 350
treeMaxScale: Vector = 0.5, 0.5
snowmanPos: Vector = random
birdhousePos: Vector = random
  
```

drawBackground *th*

drawSun *th*

drawCloud *th*

drawMountains *th*

drawMountains *th*

drawTrees *th*

drawBirdhouse *th*

drawSnowman *th*

drawSun

position: Vector

```

r1 = 50
r2 = 150
gradient = RadialGradient
  
```

Set Colorstops
yellow to transparent

Save transform

translate position

draw circle

Restore transform

Draw Mountains

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

```

stepUp: number = 30
stepDown: number = 50
x: number = 0
  
```

Save transform

translate position

MoveTo (0,0)

(LineTo (0,-max))

Execution Wall?

x = random between stepUp and stepDown

*y: number = -min + Math.random() * (max - min)*

(LineTo (x,y))

Execution Wall?

(LineTo (x,0))

closePath
create gradient with given colors

drawCloud

- position: Vector
- size: Vector

```

nParticles = 60
radiusParticles = 40
particle = Path circle
gradient = RadialGradient
  
```

Set colorstops
0.5 to 0

translate position

restore transform

[*Edrawon nParticles*]

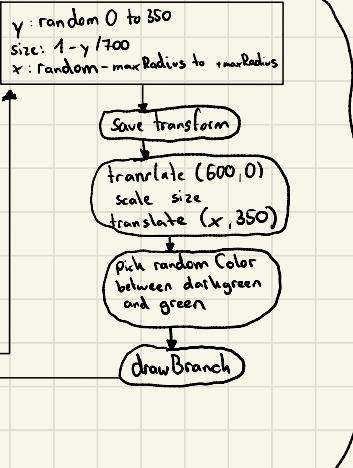
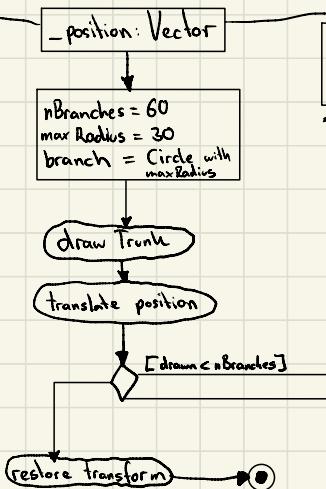
*x = (random - 0.5) * size.x*
*y = -random * size.y*

translate x, y

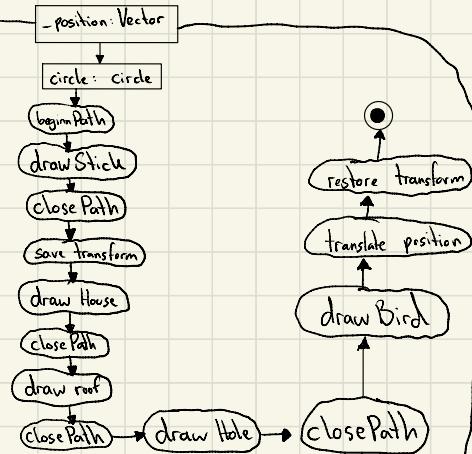
drawParticle

Restore transform

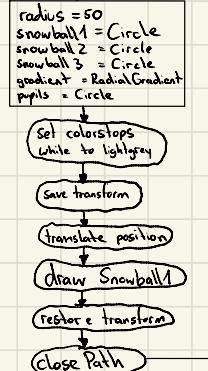
draw Tree



draw Birdhouse



draw Snowmann



Aktivitätsdiagramm: snowflakes

