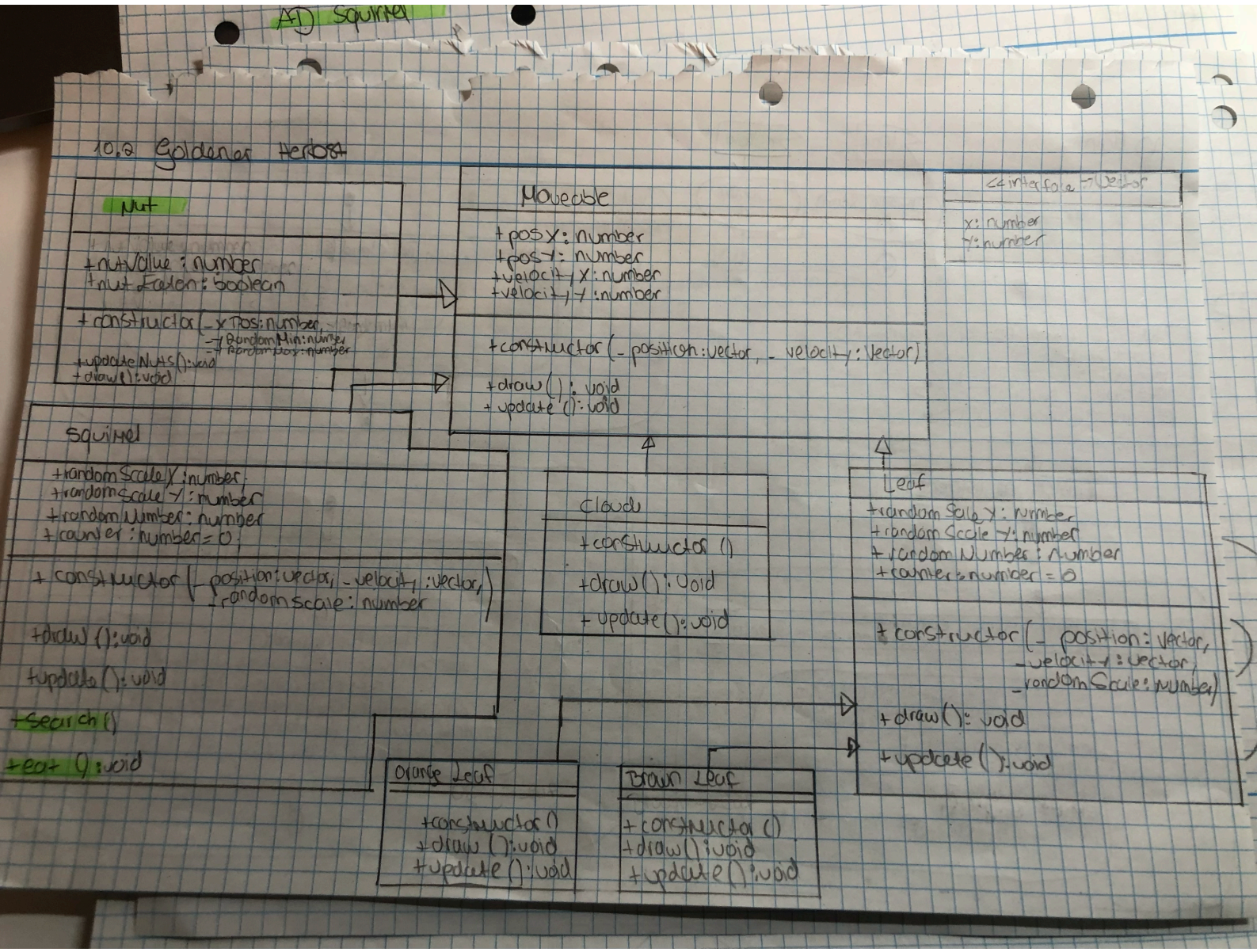
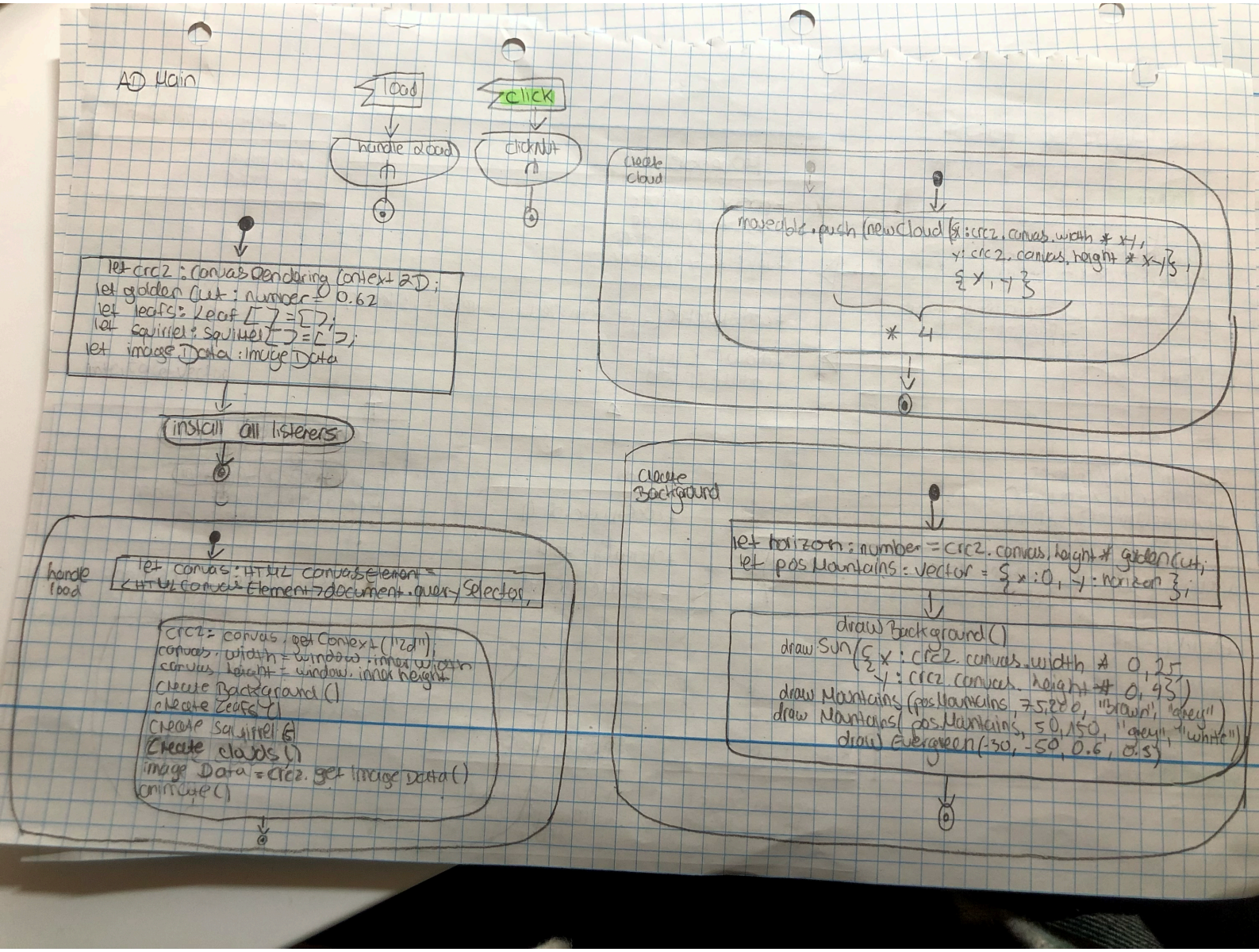
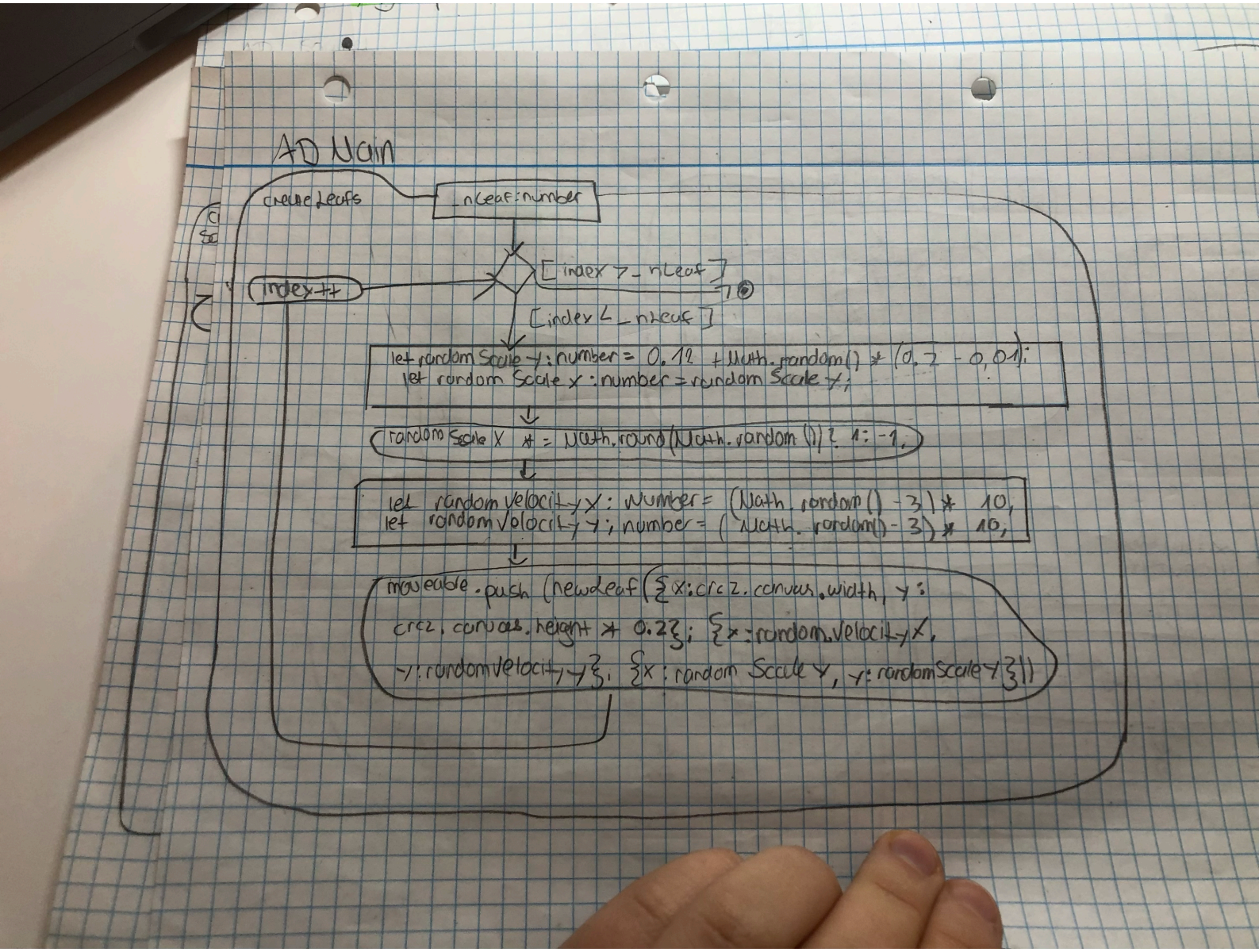
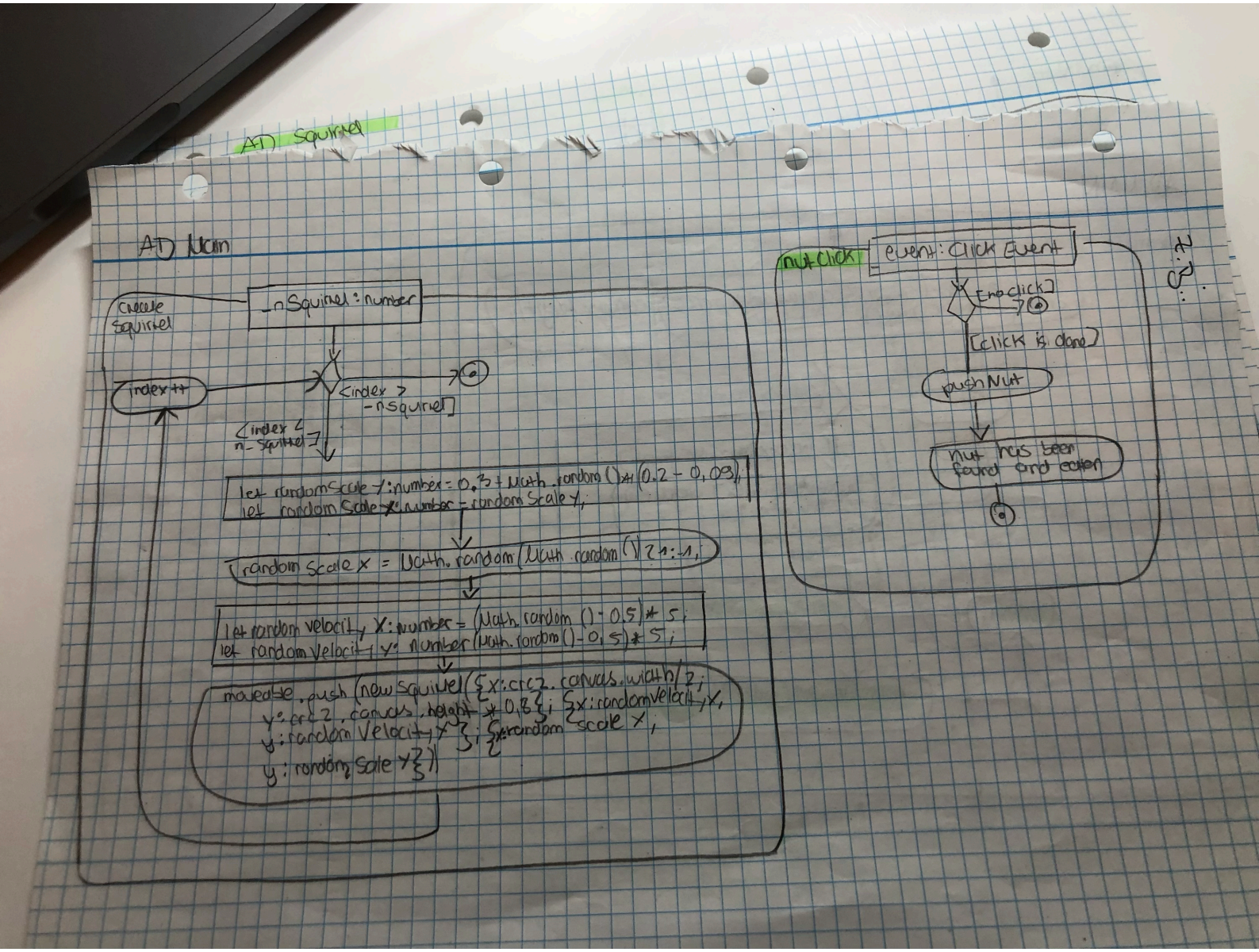
A1) Squirrel

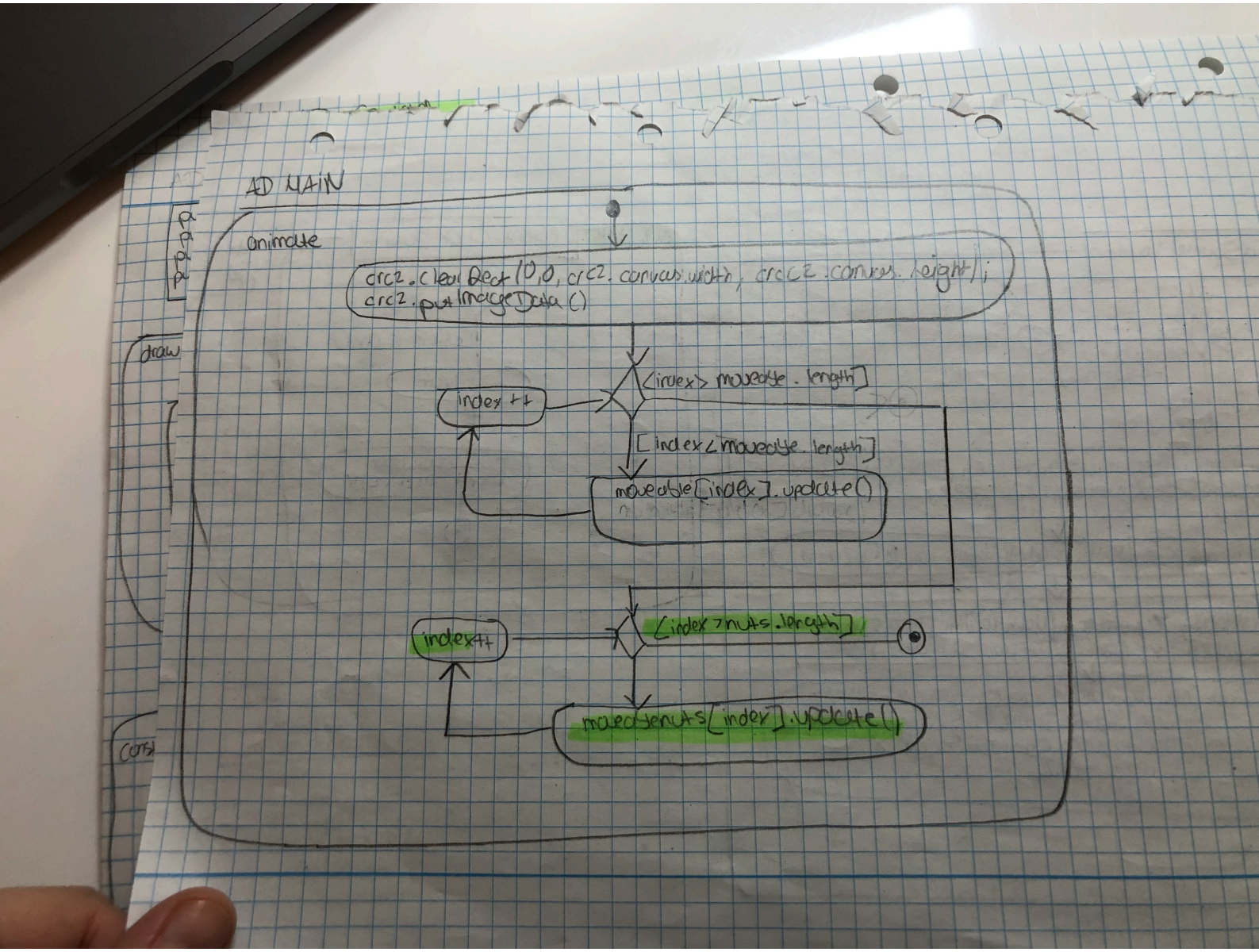
10.0 Goldener Herbst

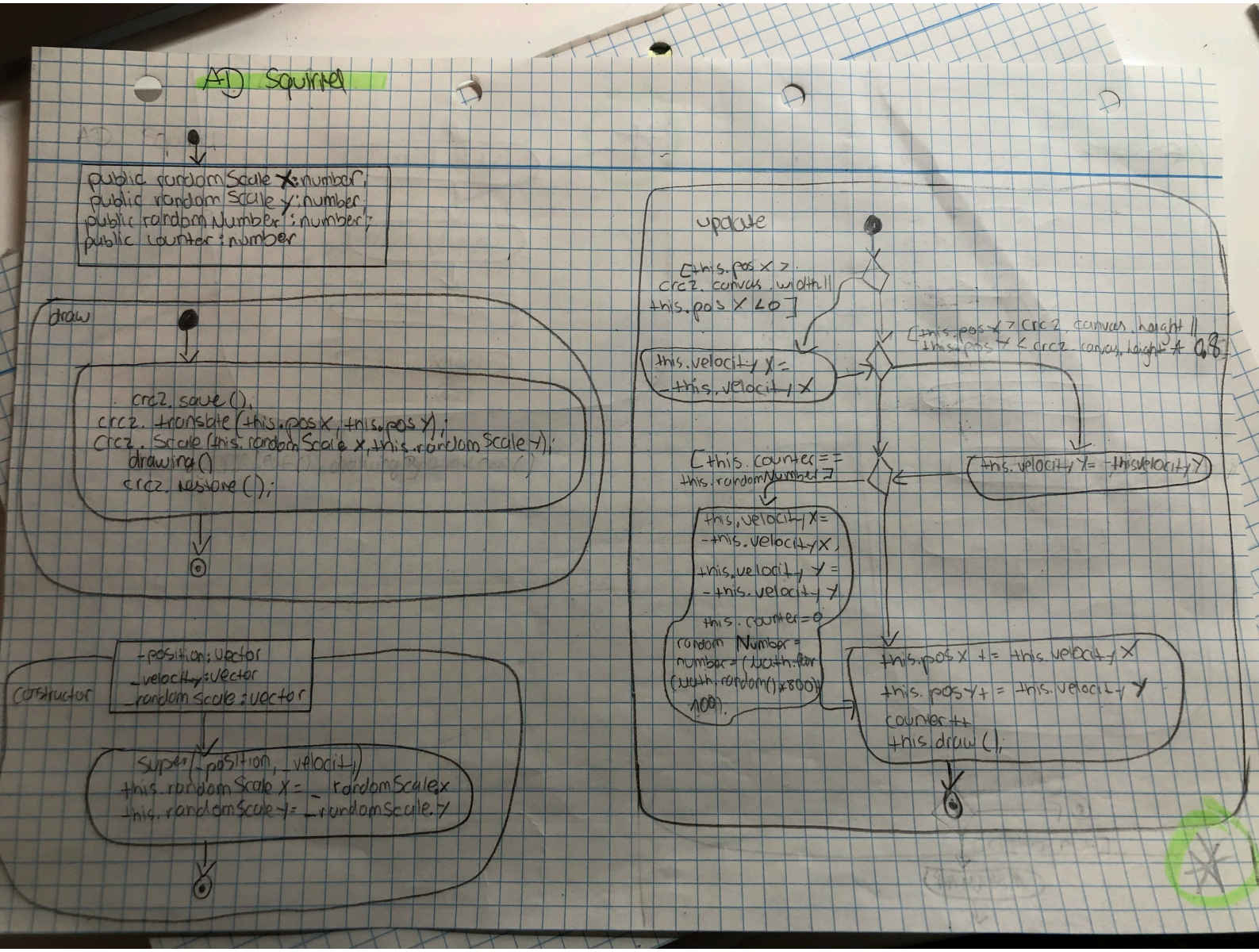


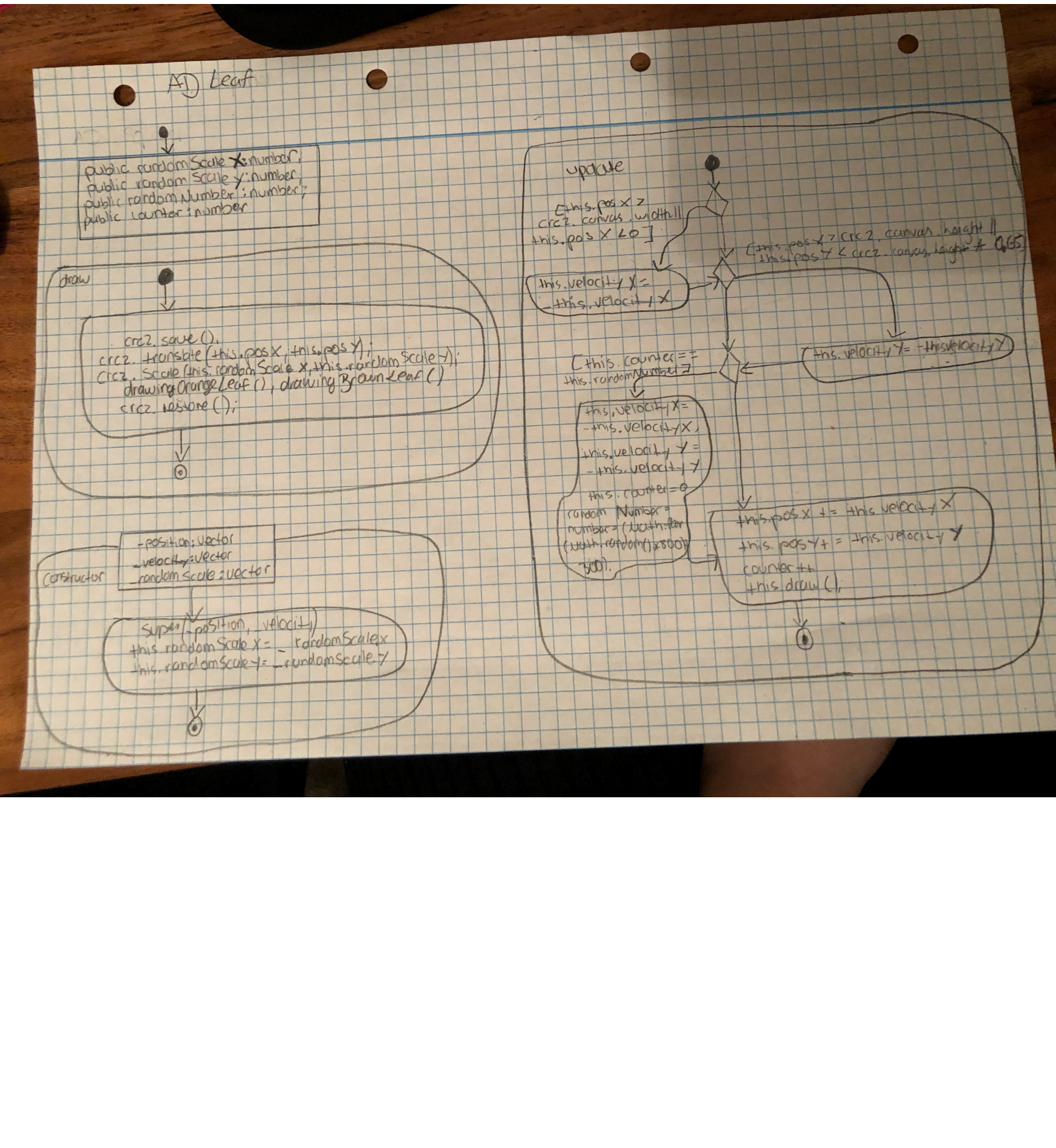

AD Main
load
handle load
click
clickAt
let ctx2 : CanvasRenderingContext2D;
let golden Cut : number = 0.62
let leafs: Leaf[] = [];
let squirrel: Squirrel[] = [];
let imageData : ImageData
install all listeners
handle load
let canvas: HTMLCanvasElement =
ctx2 = canvas.getContext("2d");
canvas.width = window.innerWidth
canvas.height = window.innerHeight
createBackground()
createLeafs()
createSquirrel()
createClouds()
imageData = ctx2.getImageData()
animate()
createCloud
mouseDble.push(newCloud({x: ctx2.canvas.width * x/,
y: ctx2.canvas.height * x/5,
{x, y}
* 4
createBackground
let horizon: number = ctx2.canvas.height * golden Cut;
let posMountains: vector = {x: 0, y: horizon};
drawBackground()
drawSun({x: ctx2.canvas.width * 0,25
y: ctx2.canvas.height * 0,95})
drawMountains(posMountains, 75,200, "brown", "grey")
drawMountains(posMountains, 50,150, "grey", "white")
drawEvergreen(-30, -50, 0.6, 0.5)

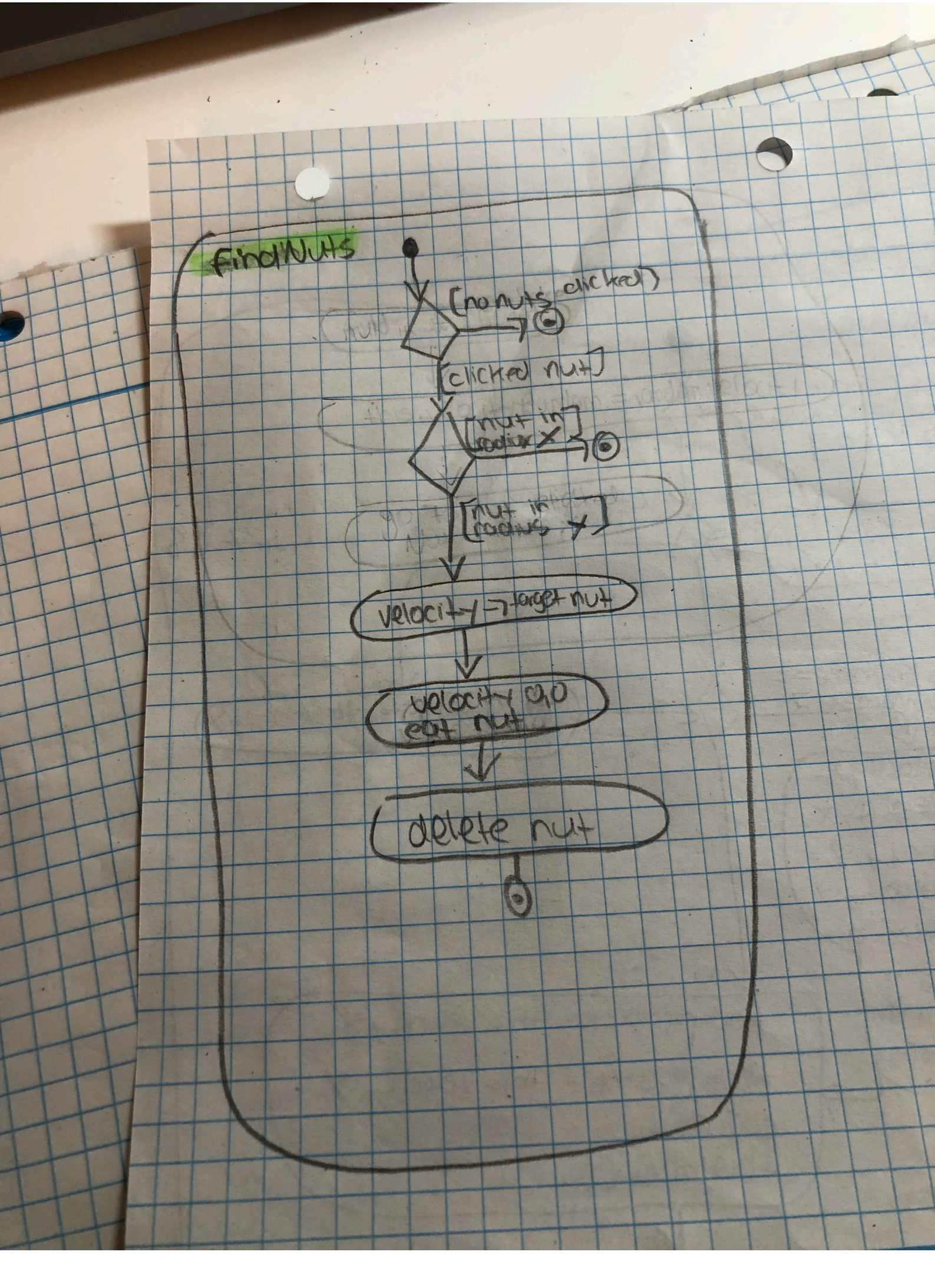

AD Nan
createLeafs
nLeaf:number
index++
[index >= nLeaf]
[index < nLeaf]
let randomScaleY:number = 0.1? + Math.random() * (0.7 - 0.01);
let randomScaleX:number = randomScaleY;
randomScaleX *= Math.round(Math.random()) ? 1 : -1;
let randomVelocityX: number = (Math.random() - 3) * 10;
let randomVelocityY: number = (Math.random() - 3) * 10;
movable.push(newLeaf({x:crc2.canvas.width, y:
crc2.canvas.height * 0.2}; {x:randomVelocityX,
y:randomVelocityY}, {x:randomScaleX, y:randomScaleY}))


AD squirrel
AD Main
createSquirrel
_nSquirrel: number
index++
index < _nSquirrel
index > _nSquirrel
let randomScaleY: number = 0.3 + Math.random() * (0.2 - 0.03);
let randomScaleX: number = randomScaleY;
randomScaleX = Math.random(Math.random() > 0.1 ? 1 : -1;
let randomVelocityX: number = (Math.random() - 0.5) * 5;
let randomVelocityY: number (Math.random() - 0.5) * 5;
moveable.push(new Squirrel({x: crc2.canvas.width/2;
y: crc2.canvas.height * 0.8}; {x: randomVelocityX,
y: randomVelocityY}; {x: randomScaleX,
y: randomScaleY}))
nutClick
_event: ClickEvent
[no click]
[click is done]
pushNut
nut has been found and eaten


AD MAIN
animate
crc2.clearRect(0,0, crc2.canvas.width, crc2.canvas.height);
crc2.putImageData()
index++
[index > moveable.length]
[index < moveable.length]
moveable[index].update()
index++
[index > nuts.length]
moveNuts[index].update()


A1) Squirrel
public randomScale X:number;
public randomScale Y:number;
public randomNumber:number;
public counter:number
draw
crc2.save();
crc2.translate(this.posX, this.posY);
crc2.scale(this.randomScale X, this.randomScale Y);
drawing()
crc2.restore();
constructor
-position:Vector
_velocity:Vector
_randomScale:Vector
super(_position, _velocity);
this.randomScale X = _randomScale.x
this.randomScale Y = _randomScale.Y
update
[this.posX > crc2.canvas.width || this.posX < 0]
this.velocity X = -this.velocity X
[this.posY > crc2.canvas.height || this.posY < crc2.canvas.height * 0.8]
this.velocity Y = -this.velocity Y
[this.counter == this.randomNumber]
this.velocity X = -this.velocity X,
this.velocity Y = -this.velocity Y
this.counter = 0
random Number = number = (width.for (width.random()*800)) 1000;
this.posX += this.velocity X
this.posY += this.velocity Y
counter++
this.draw();


A1) Leaf
public randomScaleX:number;
public randomScaleY:number;
public randomNumber:number;
public counter:number
draw
crc2.save();
crc2.translate(this.posX, this.posY);
crc2.scale(this.randomScaleX, this.randomScaleY);
drawingOrangeLeaf(), drawingBrownLeaf()
crc2.restore();
constructor
-position:Vector
_velocity:Vector
randomScale:Vector
super(_position, _velocity)
this.randomScaleX = _randomScale.x
this.randomScaleY = _randomScale.y
update
[this.posX > crc2.canvas.width || this.posX < 0]
this.velocityX = -this.velocityX
[this.posY > crc2.canvas.height || this.posY < crc2.canvas.height * 0.65]
this.velocityY = -this.velocityY
[this.counter == this.randomNumber]
this.velocityX = -this.velocityX;
this.velocityY = -this.velocityY
this.counter = 0
randomNumber = number = (width.for (width.random()*800)) 500).
this.posX += this.velocityX
this.posY += this.velocityY
counter++
this.draw();


FindNuts
[no nuts clicked]
[clicked nut]
[nut in radius X]
[nut in radius ✓]
velocity -> target nut
velocity 0,0 eat nut
delete nut