

## SUPERKLASSE

## Moving Objects

```
x: number;
y: number;
xSpeed: number;
ySpeed: number;
Start: number;
End: number;
```

## SUBKLASSE

## Food

```
radius: number;
color: string;
```

```
move();
draw();
```

## SUBKLASSE

## Fish

```
color1: string;
color2: string;
color3: string;
color4: string;
color5: string;
```

```
move() move();
draw() draw();
```

## SUBKLASSE

## Bubble

```
radius: number;
color1: string;
color2: string;
```

```
move();
draw();
```

## main.ts

Arrays  $\Rightarrow$  nur noch ein Array movingObjects: MovingObjects[]

in init():  $\Rightarrow$  addEventListener("click", feed)  $\rightarrow$   $\text{H}$

feed(): void

$\text{H}$

click

let anzahl

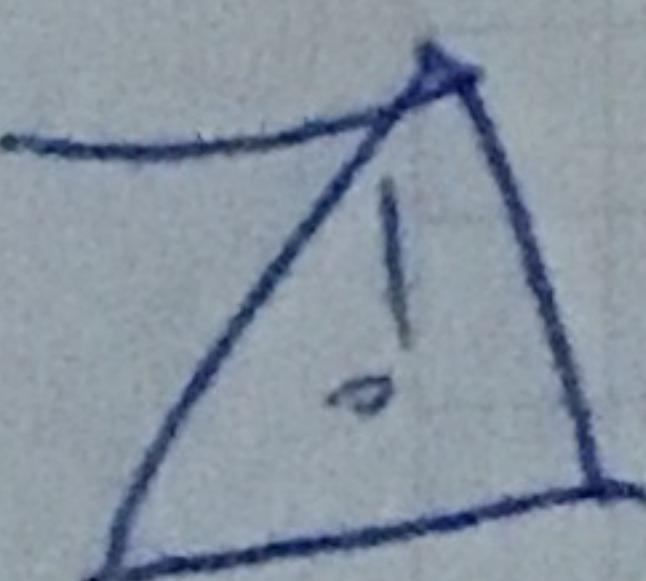
Mausposition  
x, y

FOR

let i = 0; i < anzahl; i++

i++

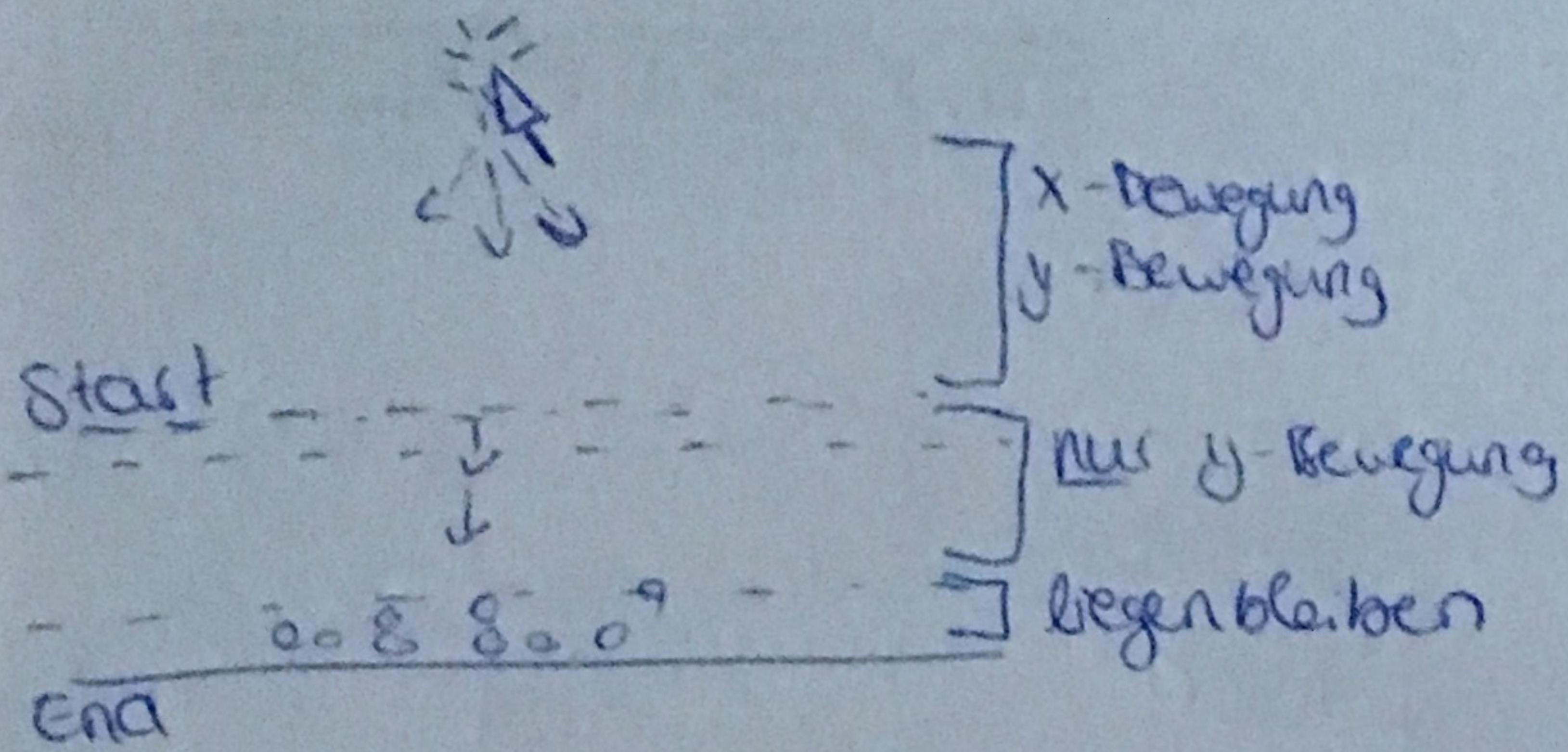
let food: Food = new Food(movingObjects.push(food))



CSS Positionierung ausgleichen

# Bewegungslogik Food

jood.ts



xPosition  
yPosition

aus jood(); übergehen

this.x = xPosition;

this.y = yPosition;

this.xSpeed = Zufallszahl minimal im - Bereich + minimal + Bereich

this.ySpeed = Zufallszahl "langsam" & nur leicht unterschiedlich

this.Start = 270 + Math.random() \* 50; // erster Bereich

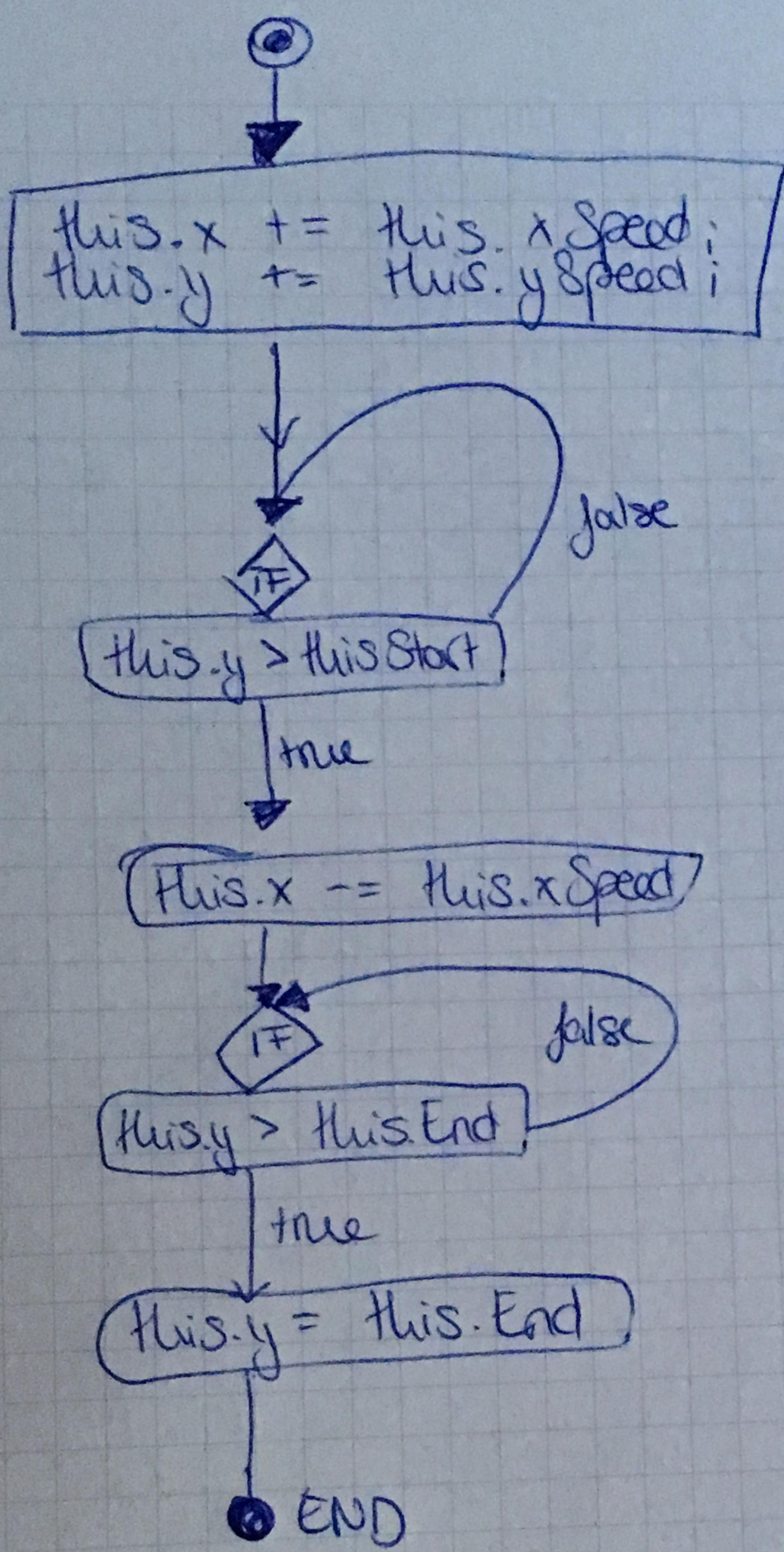
this.End = 580 + Math.random() \* 15; // zweiter Bereich


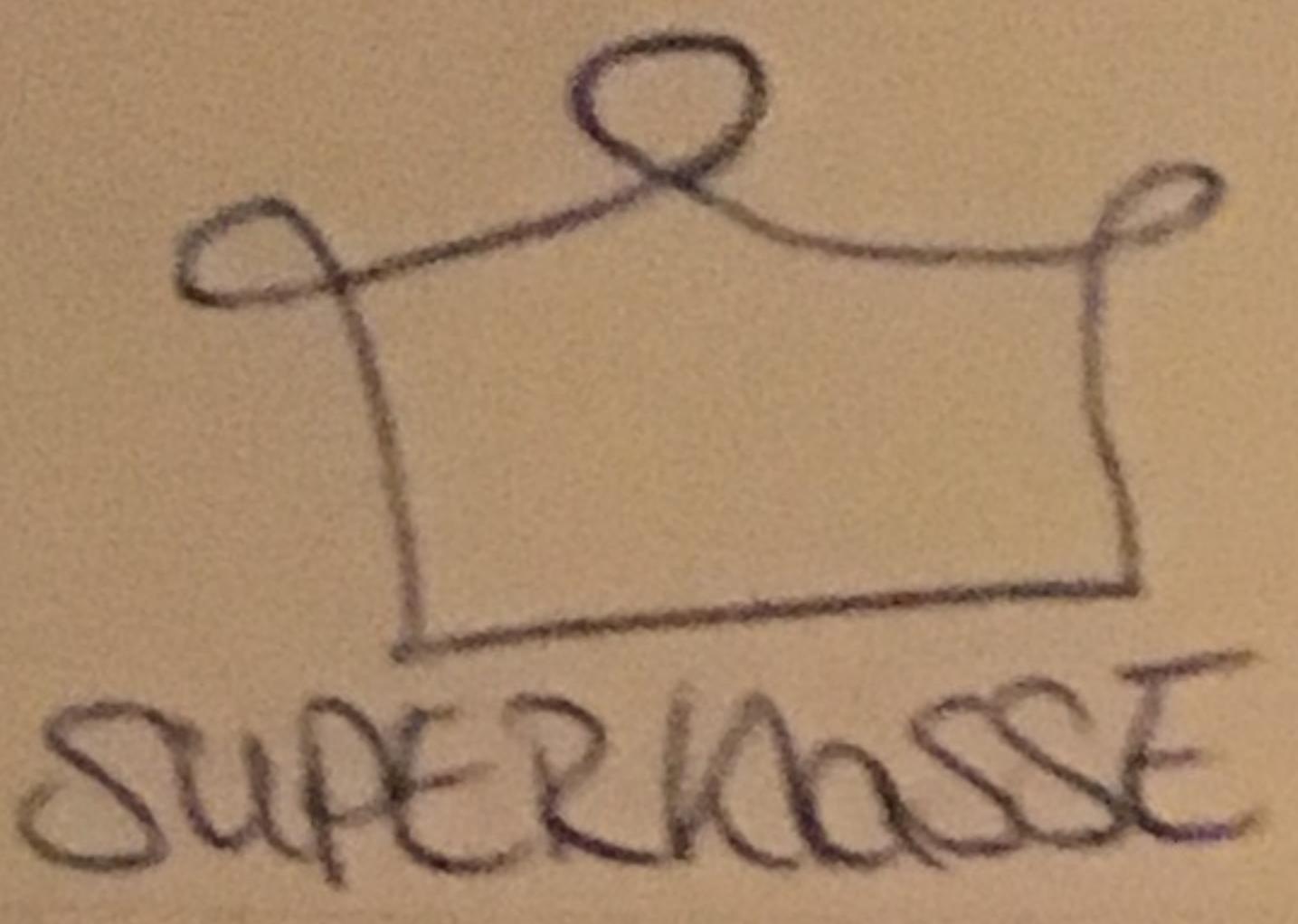
this.radius = 2 \* Math.random() \* 4;

this.color = [switch / case ~~findet über random number Zufallsfarbe~~  
random number zwischen 0 - max

case 0: farbe a  
case 1: farbe b  
case 2: farbe c

move() Food

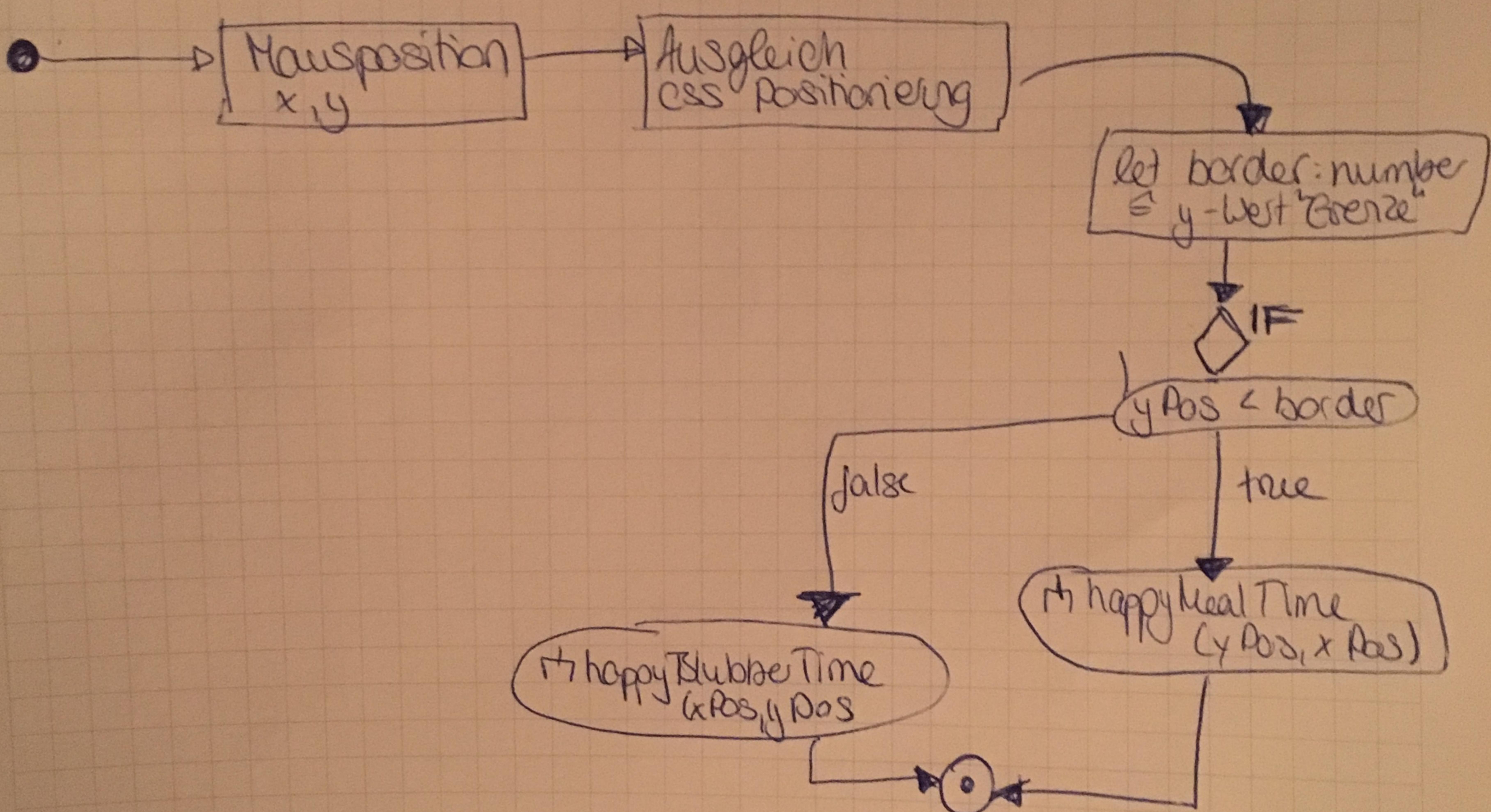




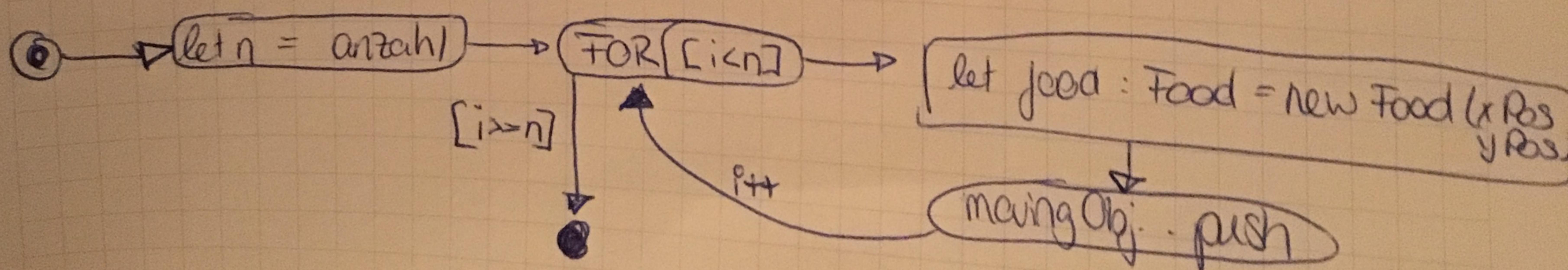
canvas.addEventListener("click", insertObject)

click → ↗ insert Object

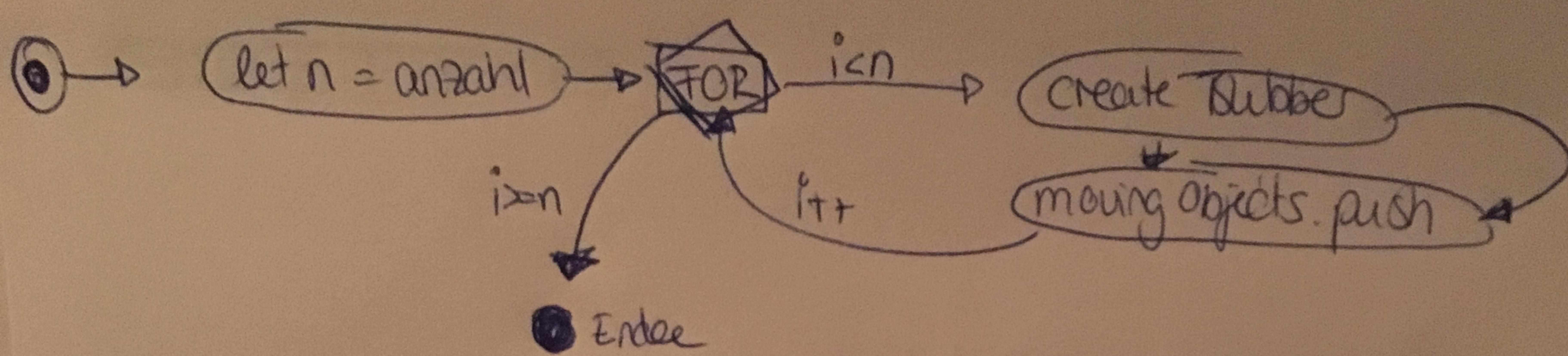
↗ insert Object (-event: MouseEvent)



↗ happy Meal Time (x Pos, y Pos)



## ↳ Happy Blubber Time (xPos, yPos)



class Blubber

> xSpeed anpassen