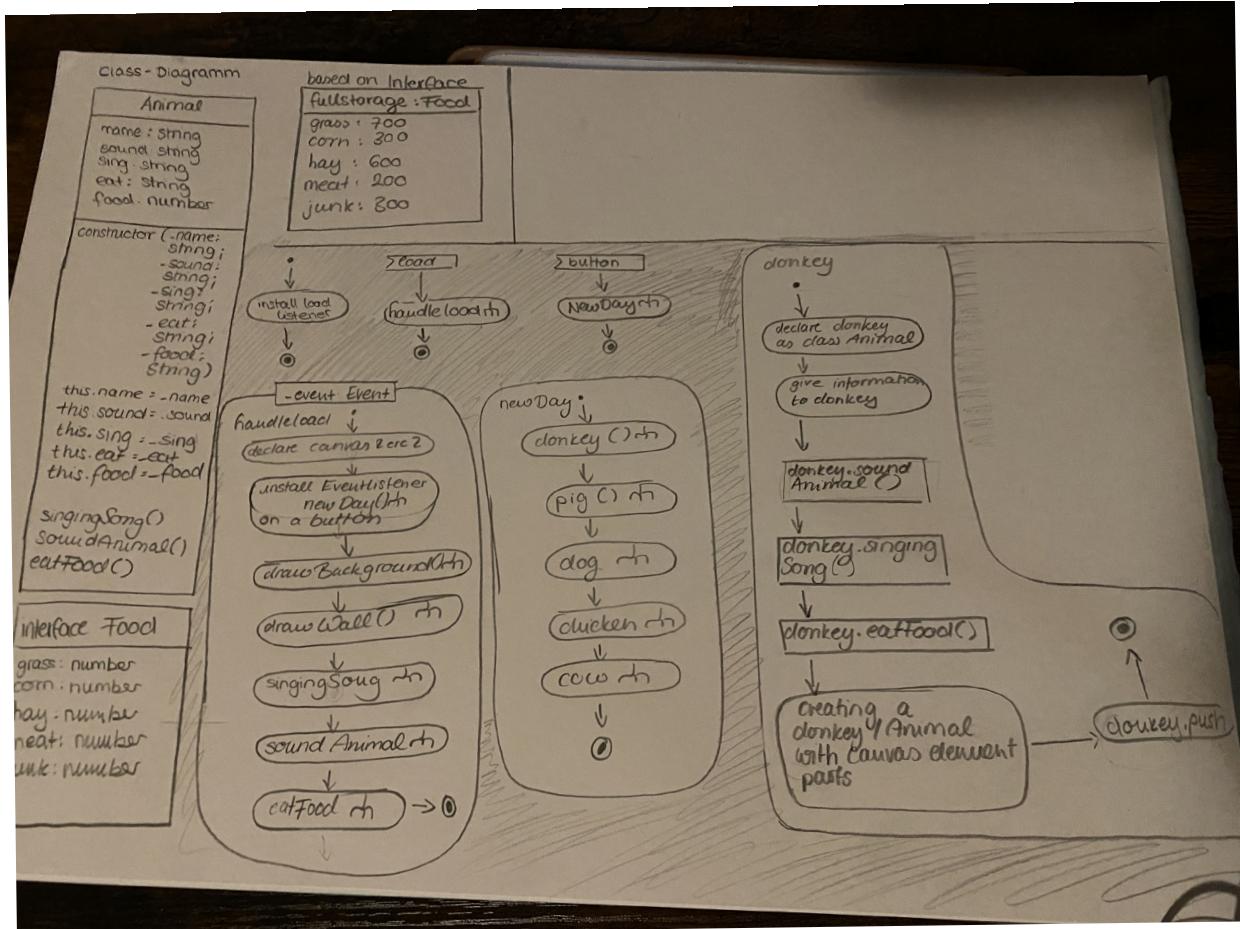


Hello Lisa! :)

Das ist mein kleiner Erklärungsaufsatz fürs Konzept.

Ich hatte ursprünglich ein anderes, wo ich dann mitten drin aufgehört habe als ich versucht hab das Ding zu Implementieren und dann masslos gescheitert bin. Daher lege ich dir die nur eine vorhandene Hälfte hin mit meiner ersten Idee des ADs und wie ich es potentiell implementieren wollte. Dann siehst du im Anhang von die Veränderung :)



Leider hab ich es nicht geschafft via interface das Essen noch das fressen zu implementieren. Wenn dann hat es mir den Code zerschossen, das HTML zersetzt oder es hat dann irgendwas mit der implementierung nicht gepasst. Auch hat das sound und sing irgendwie nicht wollen, weder als span element noch im console log.

Mir kam dann Judith zur Rettung, die genau das geschafft hat, was ich machen wollte nur mit einer Food-Klasse. Daher hab ich sie auch erwähnt im HTML-Dokument als hilfreiche Unterstützung. Musste dann das Konzept zum Teil überarbeiten. Das kommt im Anschluss.

animal.ts Klassen

```

Animal
name: string
sound: string
sing: string
eat: string
food: number

constructor(
  -name: String
  -sound: String
  -sing: String
  -eat: String
  -food: String)
  this.name = -name
  this.sound = -sound
  this.sing = -sing
  this.eat = -eat
  this.food = -food

```

```

drawDonkey()
drawCow()
drawChicken()
drawDog()
drawPig()

```

```

drawBlock(x: number, y: number, color: string)

```

const xPos = x * this.blockSize
const yPos = y * this.blockSize

draw Block with fillStyle

```

Food
blockSize: number
context: CanvasRenderingContext2D

constructor(
  blockSize: number,
  context: CanvasRenderingContext2D)
  this.blockSize = blockSize
  this.context = context

```

```

drawManager()
drawBlock()
drawManager()
drawFood()
eat(animal: Animal)

```

```
clearCanvas()
```

```

constructor(
  -name: String
  -sound: String
  -sing: String
  -eat: String
  -food: String)
  this.name = -name
  this.sound = -sound
  this.sing = -sing
  this.eat = -eat
  this.food = -food

drawChicken()
drawPig()
drawDonkey()
drawDog()
drawCow()

donkey.name = " "
donkey.sound = " "
donkey.sing = " "
donkey.eat = " "
donkey.food = " "

create head, ear, eyes, body and nose with canvas code parts like fillRect, stroke etc

```

animals.ts

```

-a: number, -b: number, -c: number
-d: number, color1: String, color2: String,
-color3: String, -color4: String

```

```

drawFood
  i < 13
  i < 24
  i < 33
  i < 40

  i: number = 0
  increase i and -a
  increase i and -b
  increase i and -c
  increase i and -d

  this.drawBlock(-a, 41, color1)
  this.drawBlock(-b, 40, color2)
  this.drawBlock(-c, 39, color3)
  this.drawBlock(-d, 38, color4)

  return

```

```

eat
animal: Animal
if (animal == dog)
  draw the food for dog in its declared color and portion
  console.log(`portion`)

```

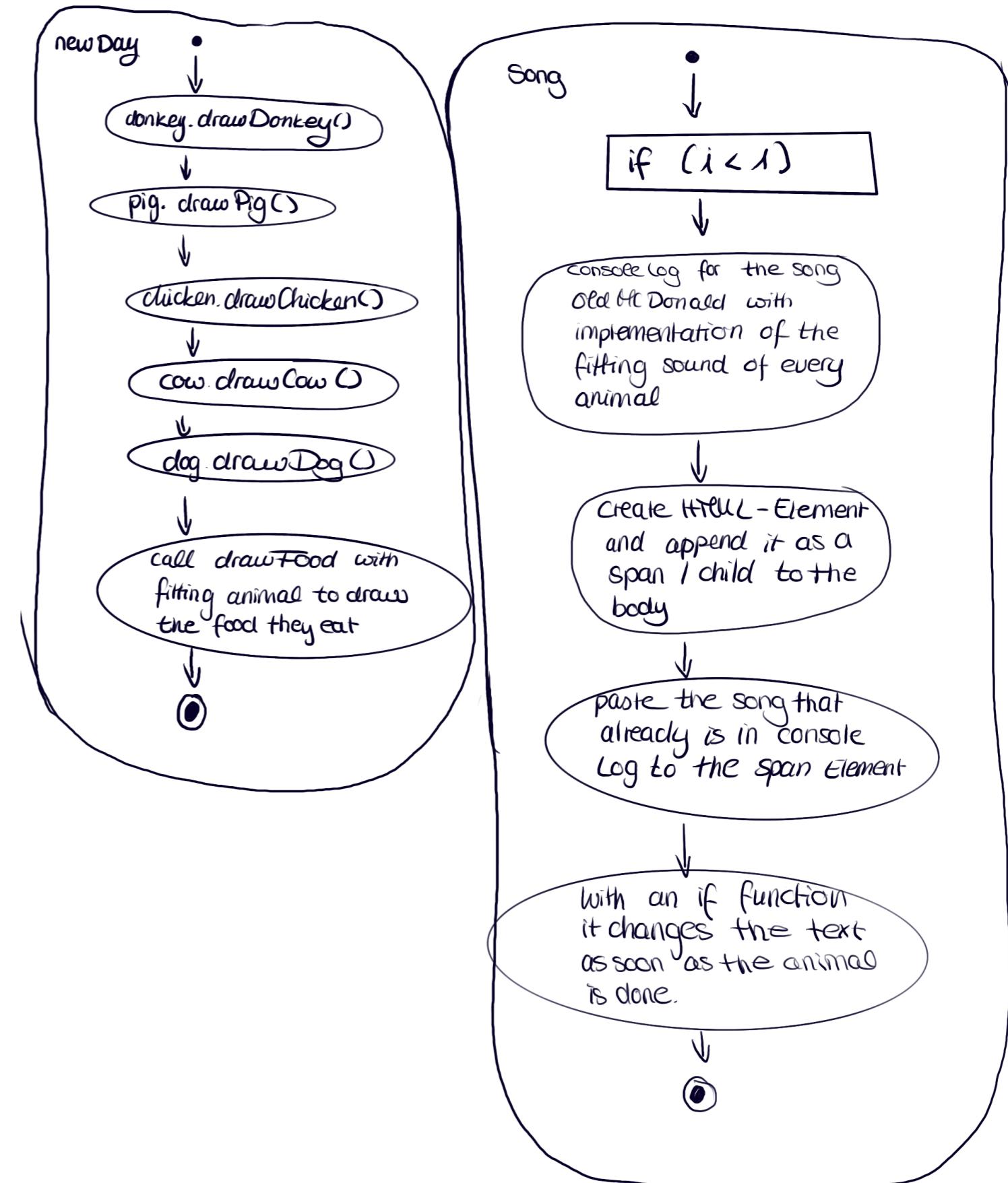
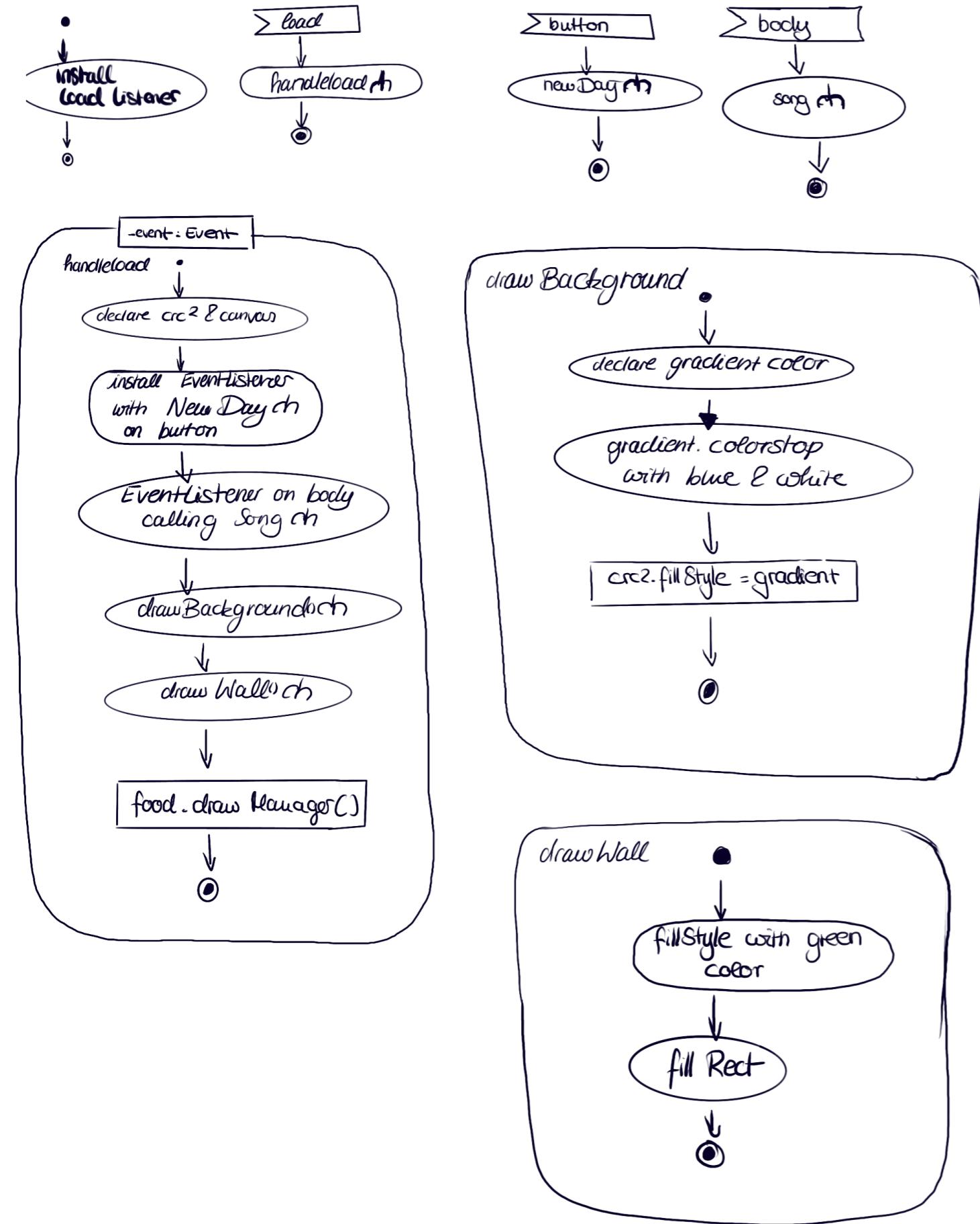
```

clearCanvas
  declare canvas in HTMLCANVASELEMENT
  this.context.clearRect(0, 0, canvas.width, canvas.height)

```

Bei eat hab ich nur das Beispiel dog genommen, da es basically dasselbe für alle Tiere sind, die ich definiert habe.

Hier möchte ich auch nochmal um Verbesserung fragen, da die Skizze als if funktion im Booklet mich sehr verwirrt hat und mir es einfach erschien das dann so zu schreiben :)



Hier hat dann irgendwie das mit dem span element immer noch nicht geklappt, versuche es aber definitiv weiter. Habs aus dem Code mal rausgenommen um dann Platz zu sparen. Außerdem ähnlich wie oben, habs für ein Tier geschrieben, ist aber gleich wie bei allen Tieren definiert = weil Klasse :D

Scribble

