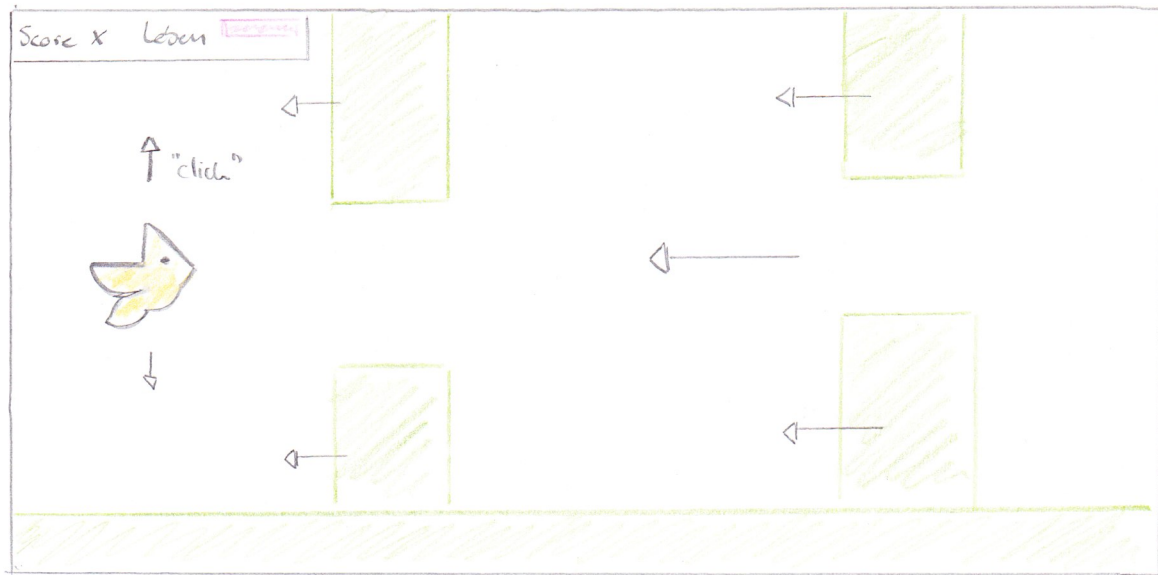


# Funktionale Analyse

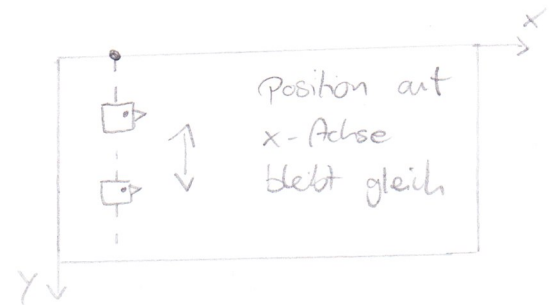
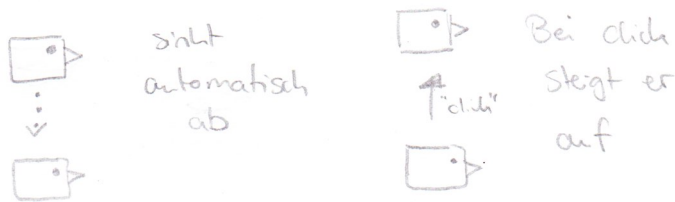


- Die Balken wandern in einer bestimmten Geschwindigkeit von rechts nach links. Aus dem Bild raus und neue ins Bild rein.
- Der Vogel sinkt konstant ab und steigt bei Klick ein Stück auf, die Position auf der x-Achse bleibt gleich.
- Der Spieler muss nur versuchen den Vogel so springen und absinken zulassen, dass keine der Rechtecke berührt werden.
- Überschneidet sich die Fläche des Vogels mit der Fläche eines Rechtecks so bekommt er Schaden. Ein paar mal hält der Vogel dies aus, doch irgendwann wenn sein Leben leer ist, ist das Spiel vorbei.
- Immer wenn der Vogel erfolgreich durch eine Lücke fliegt gibt es einen Punkt.
- Ziel ist es so viele Punkte wie möglich zu erreichen.
- Es sollen mehrere lvl geben, nach einer gewissen Zeit oder Distanz soll die Geschwindigkeit der Rechtecke erhöht werden, damit die Schwierigkeit und Herausforderung steigt.
- Die Balken sollen eine variable Höhe bekommen.

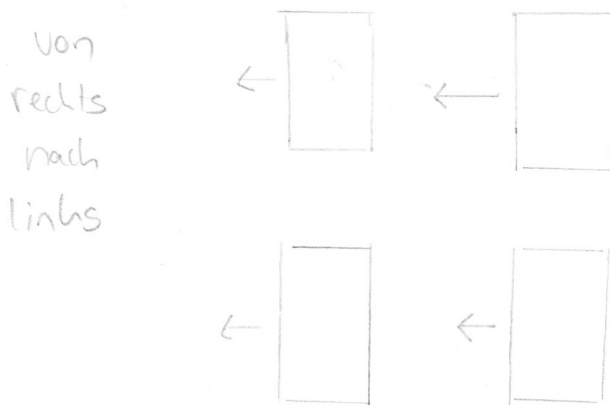
- Der Spieler soll die Möglichkeit bekommen zwischen 3 verschiedenen Vögeln auszuwählen. Sie unterscheiden sich in Farbe, Größe, Fallgeschwindigkeit und Sprungkraft.

	Vogel 1	Vogel 2	Vogel 3
Größe	●●●●	●●○○	●●●●
Fallgeschwindigkeit	●●○○	●●○○	●●●○
Sprungkraft	●●○○	●●●○	●○○○

- Vogel steuern



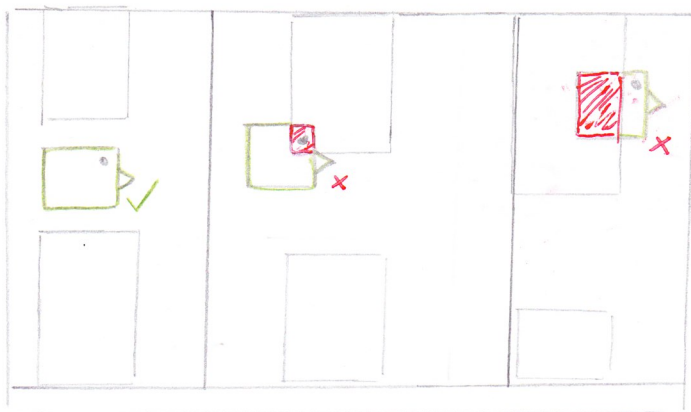
- Balhen bewegen sich



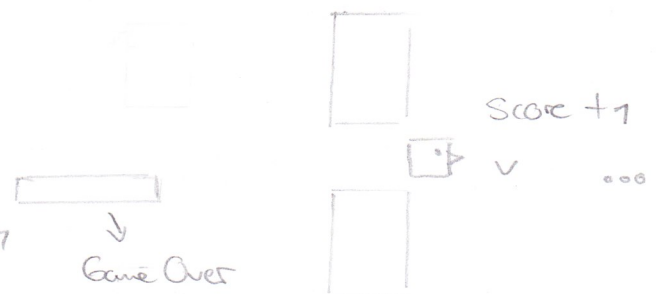
kommen rechts rein und gehen links wieder raus.  
Jeder Balhen hat seine eigene höhe. Breit ist immer gleich.  
Die Lücke muss immer groß genug zum Durchfliegen sein.

- Schaden

Überschneidet die Fläche des Vogels die des Rechtecks so bekommt er Schaden und die Lebensanzeige geht zurück.



- Punkte  
gibt es bei jedem überstandenen Balhen-Paar



Life: [Progress bar] → [Progress bar] → [Progress bar] →

## Nutzerinteraktion

## Reaktion des Systems

### Startscreen

- Klick auf einen Vogel

- Vogel wird ausgewählt und Informationsscreen eingeblendet

### Informationsscreen

- Klick auf Canvas

- Spiel wird gestartet

### Hauptscreen

- Klick auf Canvas

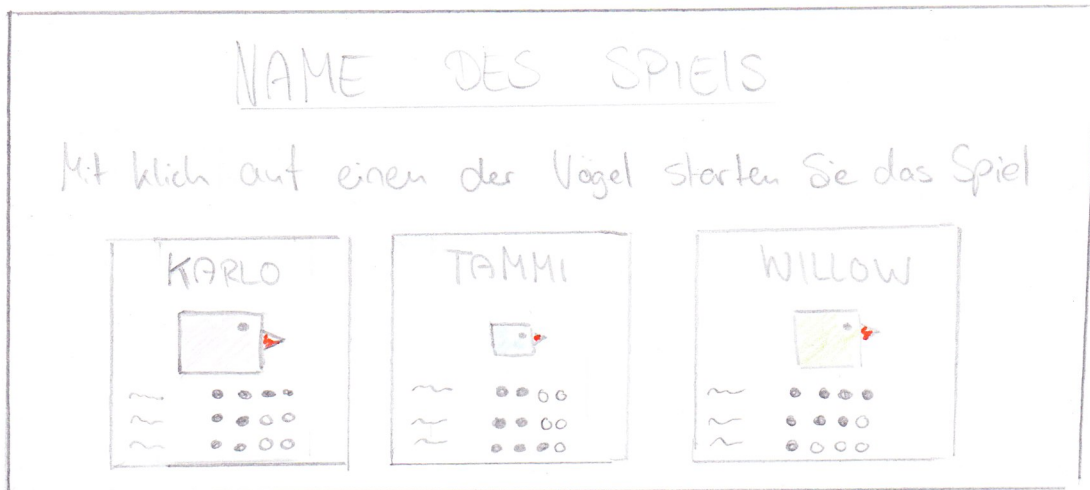
- Vogel springt nach oben

### Endscreen

- Klick auf Replay

- Startscreen wird eingeblendet

# Startscreen



click!



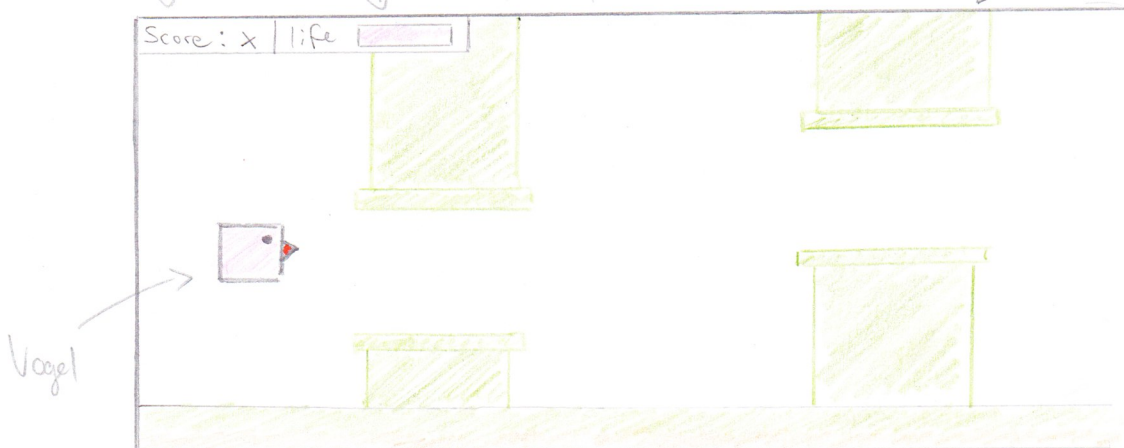
## Informationsscreen



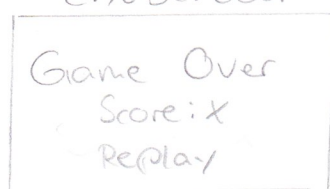
click!



Score      Lebensanzeige      Hauptscreen      Rechtecke / Rohre

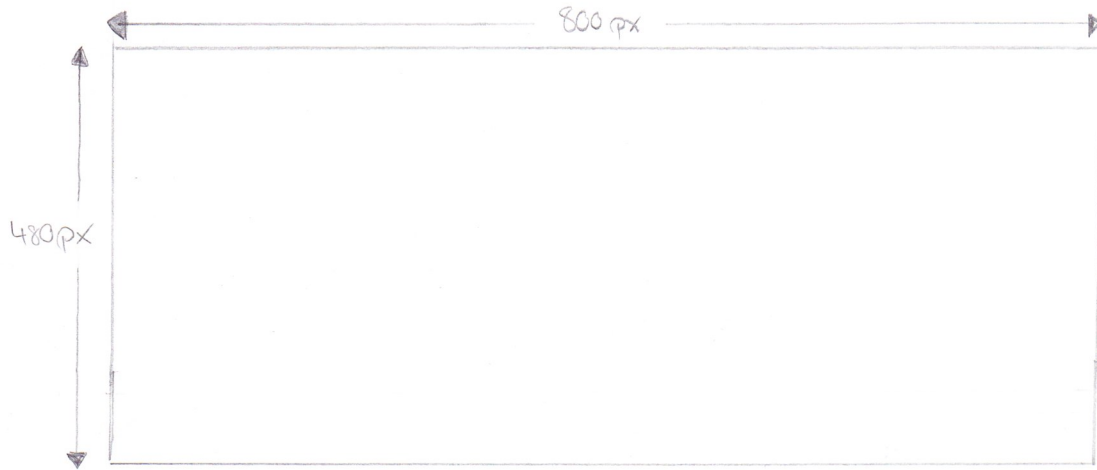


Endscreen

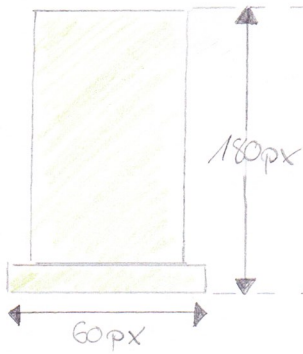


# Technische Analyse

## Canvas

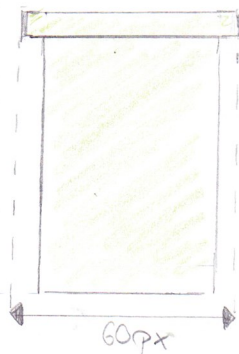


RohrOben



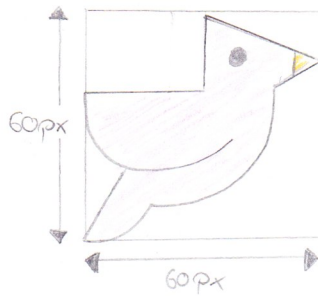
•png

RohrUnten



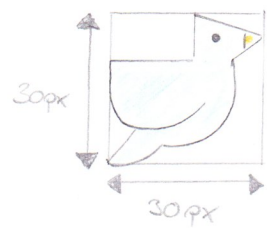
•png

Vogel 1 & 3



•png

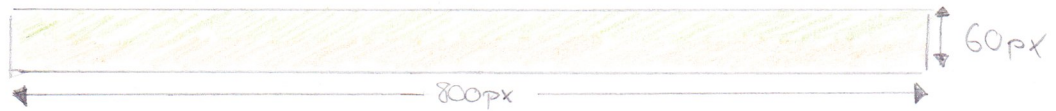
Vogel 2



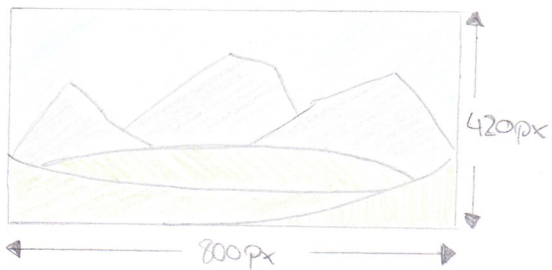
•png

Boden

•jpg

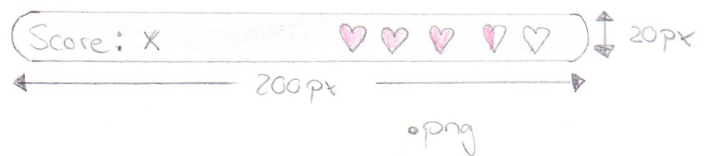


Hintergrund



•jpg

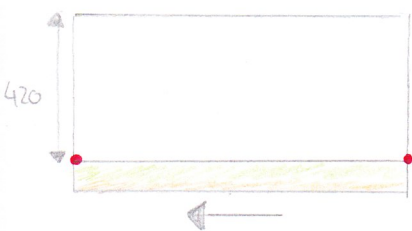
Score- + Lebensanzeige



•png



# Animierte Objekte im Canvas

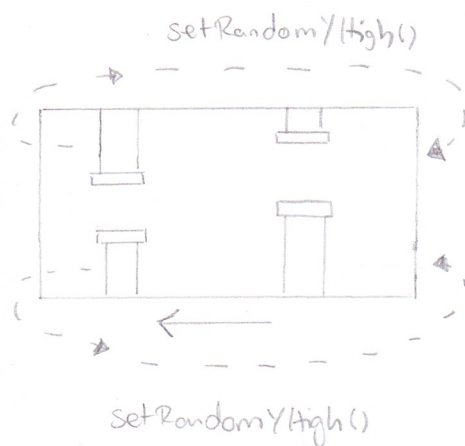
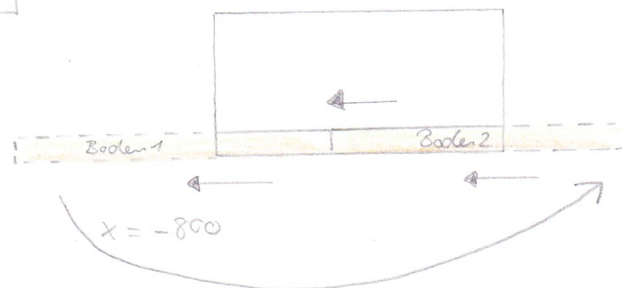


**Boden:** läuft von rechts nach links  
(zwei Stück)

Boden 1 (0, 420)    Boden 2 (800, 420)

↳ this.x 0 bis -800  
und dann  
if (this.x <= -800) {  
  this.x = 0  
}

update()  
// speed  
this.x += -6.5



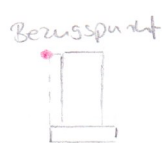
**Rohre**    4x rohrOben.png    4x rohrUnten.png

(x, y)  
RohrOben1 - (0+x, y)  
RohrOben2 - (400+x, y)  
RohrOben3 - (800+x, y)  
RohrOben4 - (1200+x, y)

RohrUnten1 (0+x, y)  
RohrUnten2 (400+x, y)  
RohrUnten3 (800+x, y)  
RohrUnten4 (1200+x, y)

x → update()  
this.x += -6.5

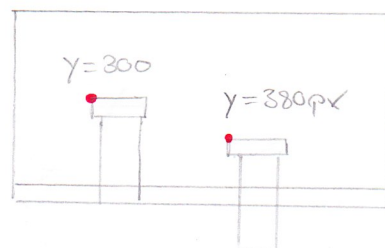
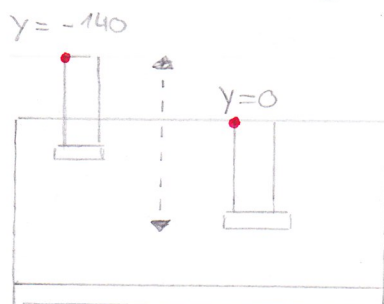
y → setRandomY(tigh())  
this.y = math.random



oben  
zwischen  
-140 px  
bis  
0



unten  
zwischen  
300 px  
bis  
380 px

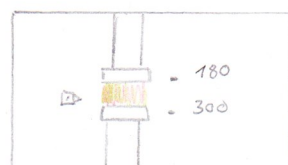


**Wichtig!**

Zwischen dem oberen  
Rohr und dem unteren  
Rohr muss immer  
genug Platz zum  
durchfliegen sein.

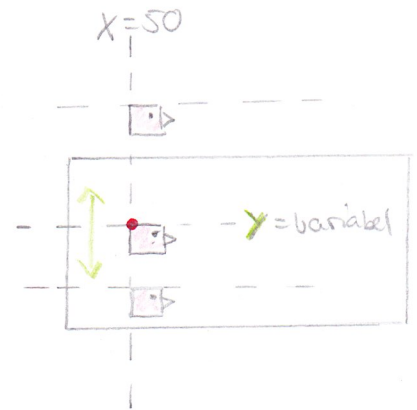
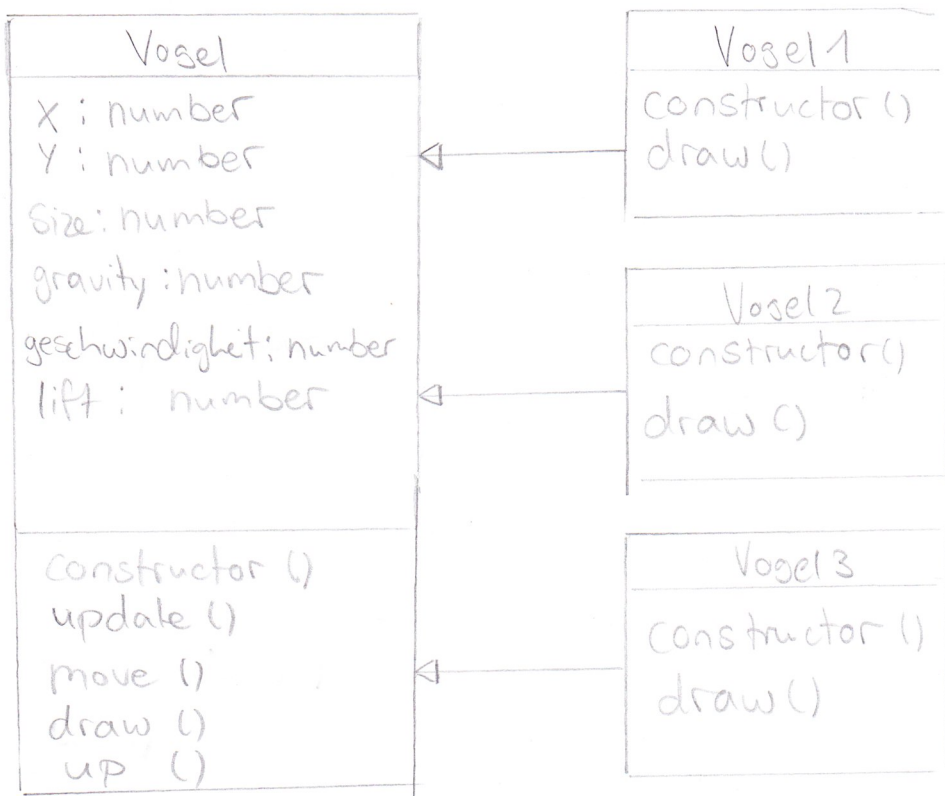
max y-oben = 0

min y-unten = 300



mindestens  
120 px platz

**Vogel** : Fliegt hoch und runter aber nicht vorwärts / rückwärts  
Drei Vogel Arten



`drawImage (x, y, 60, 60)`

$x = 50$  immer 50  
 $y$  = verändert sich

• Vogel sinkt konstant ab

→  $gravity : number = 1$   
 $geschwindigkeit : number = 0$   
 $lift = -30$

`update()`

`move()`

$geschwindigkeit += gravity,$   
 $geschwindigkeit *= 0.9,$   
 $y = geschwindigkeit,$   
`}`

• Vogel steigt auf bei  
Mausklick

click Event

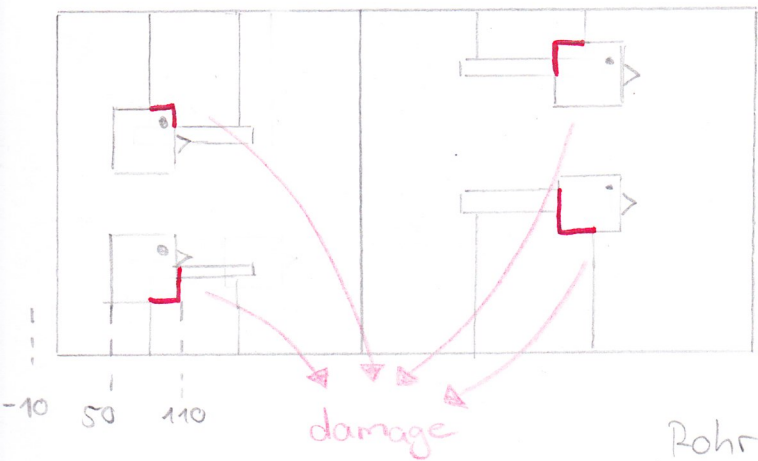
`canvas.addEventListener "click"`

↓

`up()`

$geschwindigkeit += lift$   
`}`

# Damage



- Bei jeder Überschneidung  
damage ++

- Life = 1000 - damage

```
if (Life == 0) {
    gameOver
}
```

Bedingung 1:

Rohr muss zwischen  
 $x = -10$  und  $x = 110$  sein

Bedingung 2 (Oben):

$(y_{\text{Rohr}} - 180) > y_{\text{Vogel}}$

Bedingung 2 (Unten):

$y_{\text{Rohr}} < (y_{\text{Vogel}} + 60)$

Wenn beide Bedingungen  
erfüllt sind  $\rightarrow$  damage ++

Score:

Immer wenn das Rohr  $x = 50$   
erreicht hat.

Score ++;

Ausgabe: innerText

Life

life = 1000

Lebensbalken  $\uparrow$  Länge = life

FillRect (100, 12, life, 10)

life = 500

life = 0

Life

prog  $\rightarrow$

+

+

rect  $\rightarrow$

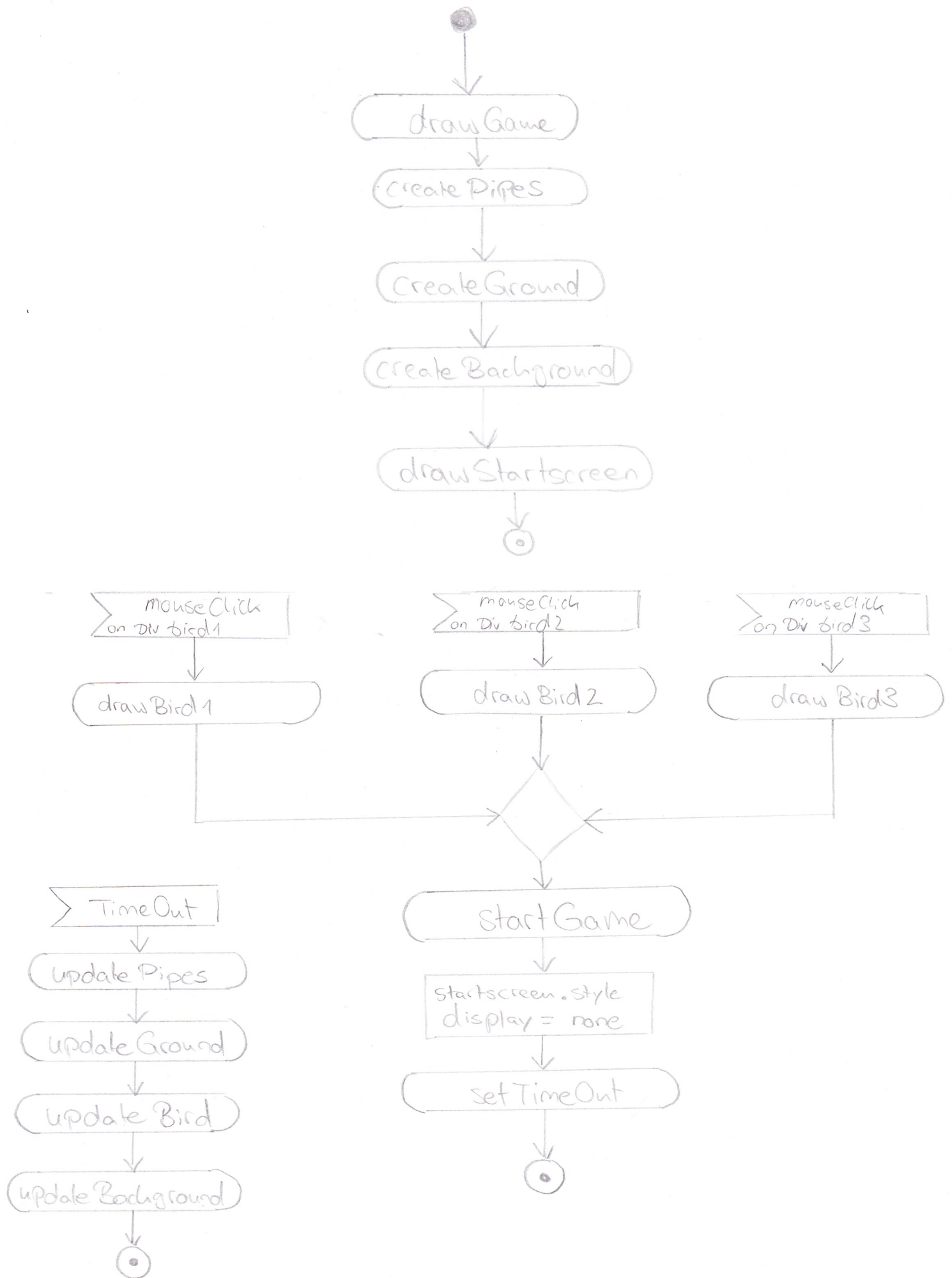
=

=

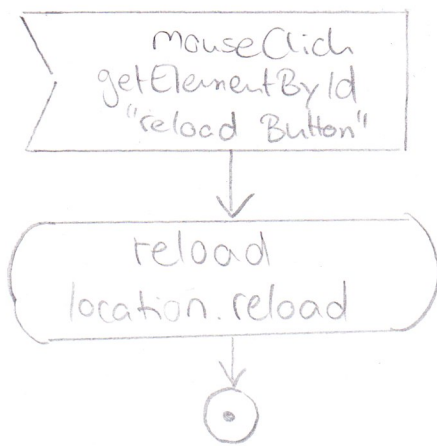
Lebensanzeige  $\rightarrow$



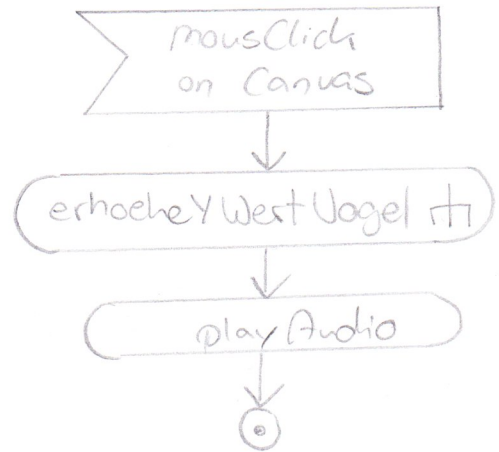
# Spielablauf



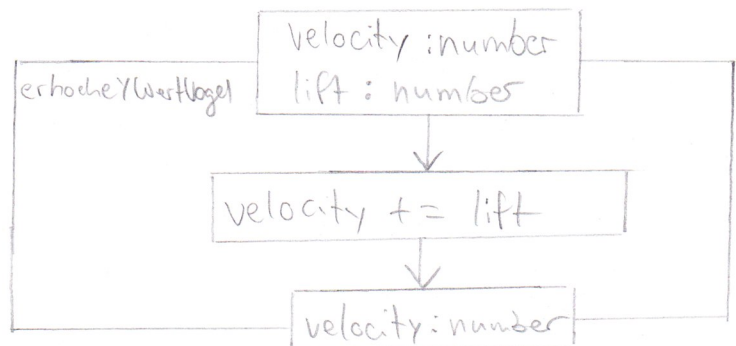
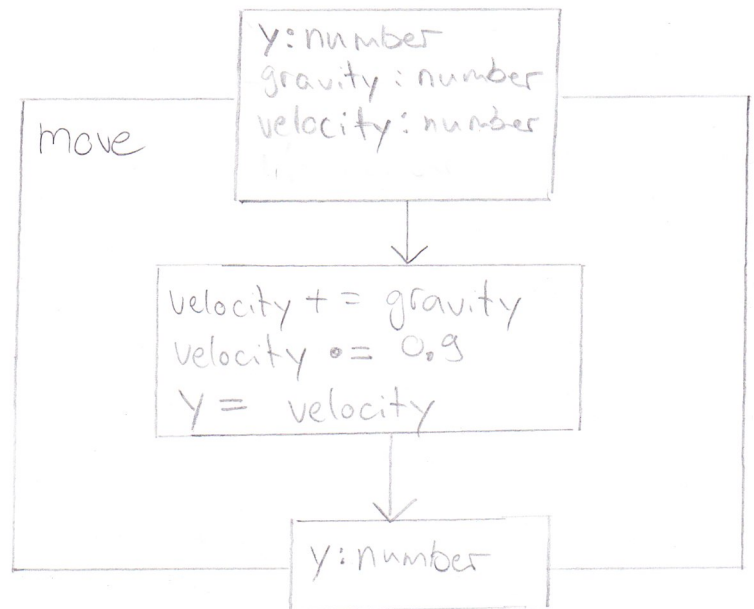
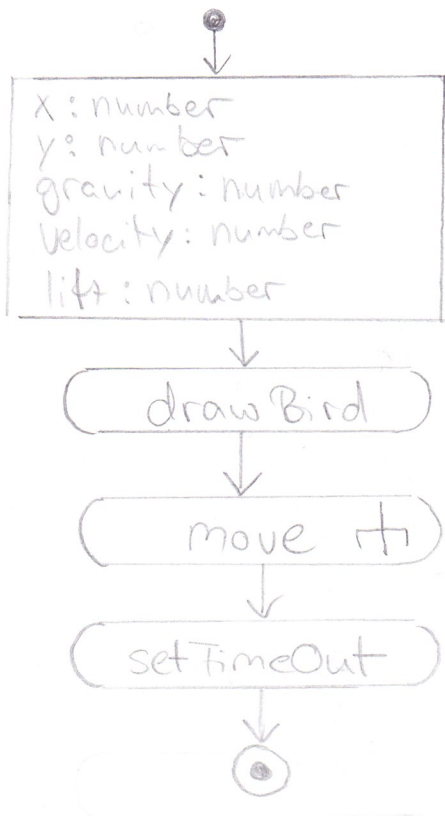
## Reload



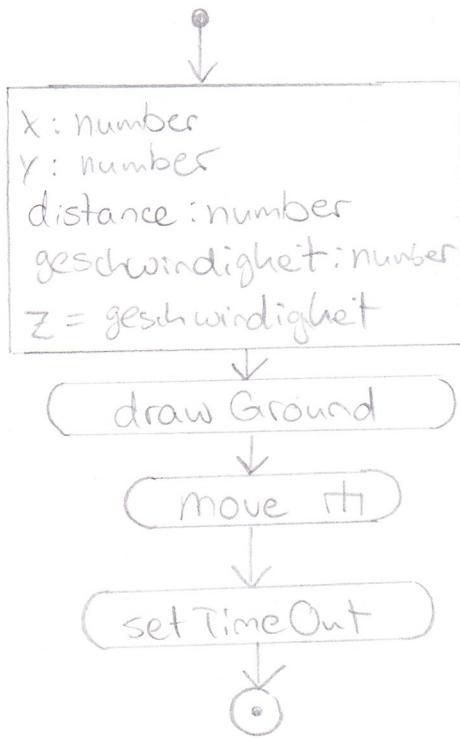
## Vogel springen lassen



## Vogel



# Boden



# Pipes

