

Funktionale- und Technische-Analyse

Das Spiel soll eine Art Sidescroller verkörpern. Damit ist gemeint, dass sich der Hintergrund nach rechts immer wieder loopt und so eine Illusion von unendlicher Bewegung in eine Richtung entsteht.

Als Spieler hat man die Kontrolle über einen Fisch, der sich im Canvas mit Hilfe der Pfeiltasten frei bewegen lässt. Daher ist das Spiel für den PC gedacht, da außerdem auch eine ausreichende Fenstergröße für den Canvas gewährleistet werden muss.

Sollte der Fisch an eine Kante des Canvas gelangen, wird das Spiel beendet und man wird gebeten seinen Nickname einzugeben. Sollte der Gamescore unter den top fünf aller Zeiten liegen, wird dieser auf dem Leaderboard angezeigt und dafür werden die restlichen Playerranks dementsprechend angepasst.

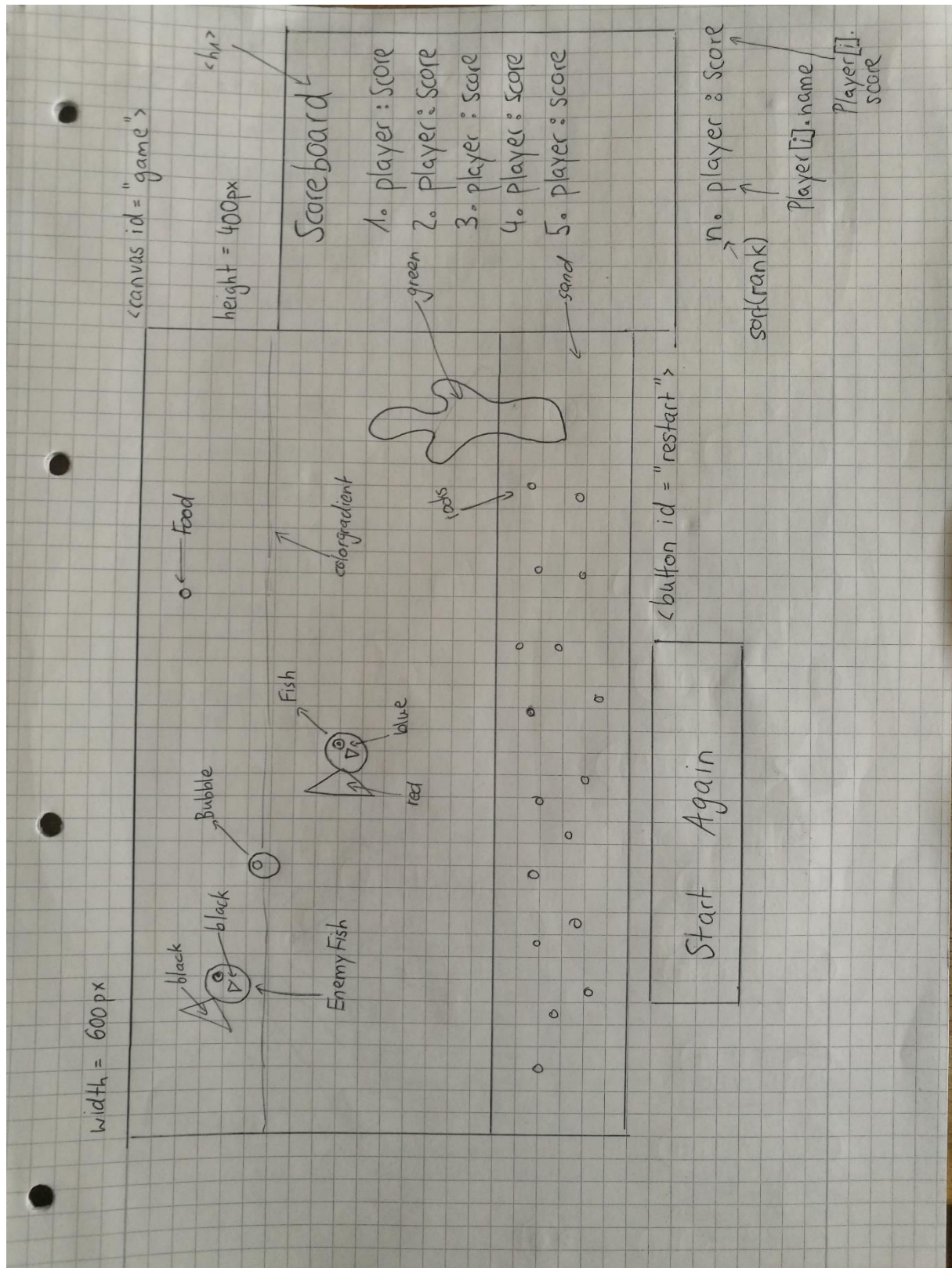
Ziel des Spiels ist deshalb, den höchsten Gamescore zu erreichen und andere Spieler von ihrem Thron zu stürzen. Gamescore wird einmal durch einfaches überleben auf Zeit erhöht und durch das Fressen von Futter und feindlichen Fischen. Trifft man auf eine Luftblase, werden Punkte wieder abgezogen.

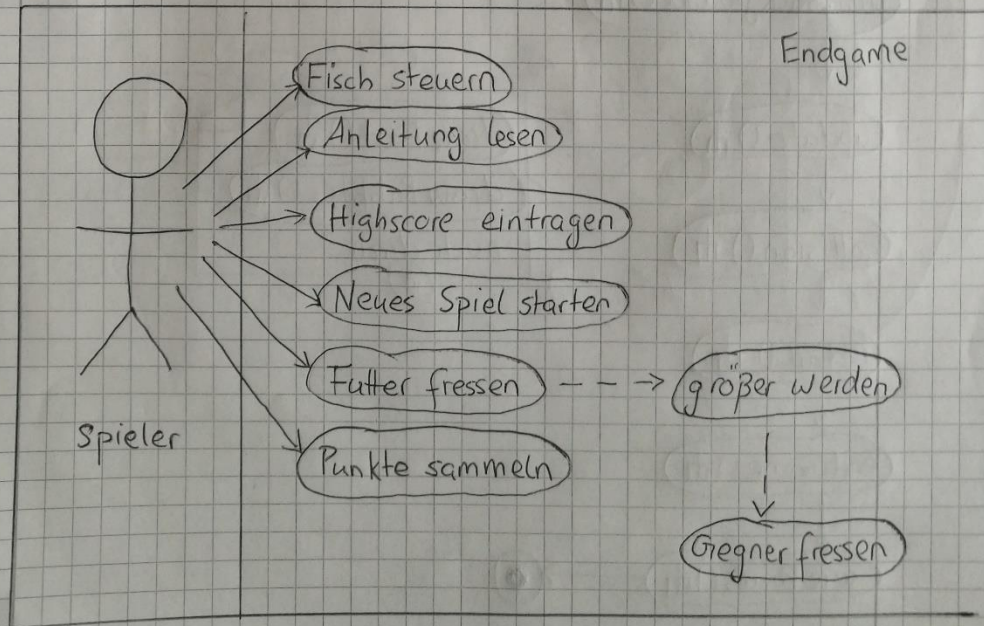
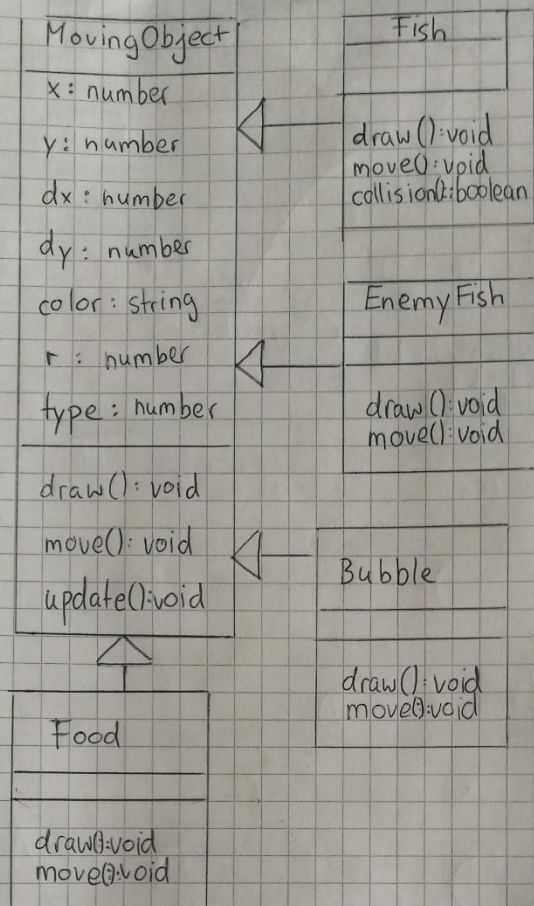
Es existieren immer gleich viele feindliche Fische, sollte einer gefressen werden, dann wird ein neuer erstellt. Untereinander interagieren sie nicht, sie können aber mit dem Spieler kollidieren und so das Spiel beenden. Ihre Bewegungsgeschwindigkeit und Richtung ist randomisiert und ändert sich jedes Mal, wenn sie den Canvas verlassen. Sie sind alle größer als der Spielerfisch, daher können sie nur nach dem Essen eines Futterstückes besiegt werden.

Das Futter wird alle sieben Sekunden neu am oberen Ende des Canvas gespawnt und wandert dann auch mit zufälliger Geschwindigkeit zum unteren Ende, wo es dann nicht mehr gefressen werden kann und gelöscht wird. Feindliche Fische können das Futter nicht fressen, nur der Spieler kann bei Kontakt einen Effekt auslösen. Das Futter bewirkt, dass der Spielerfisch für vier Sekunden wächst und in dieser Zeit alle feindlichen Fische fressen kann.

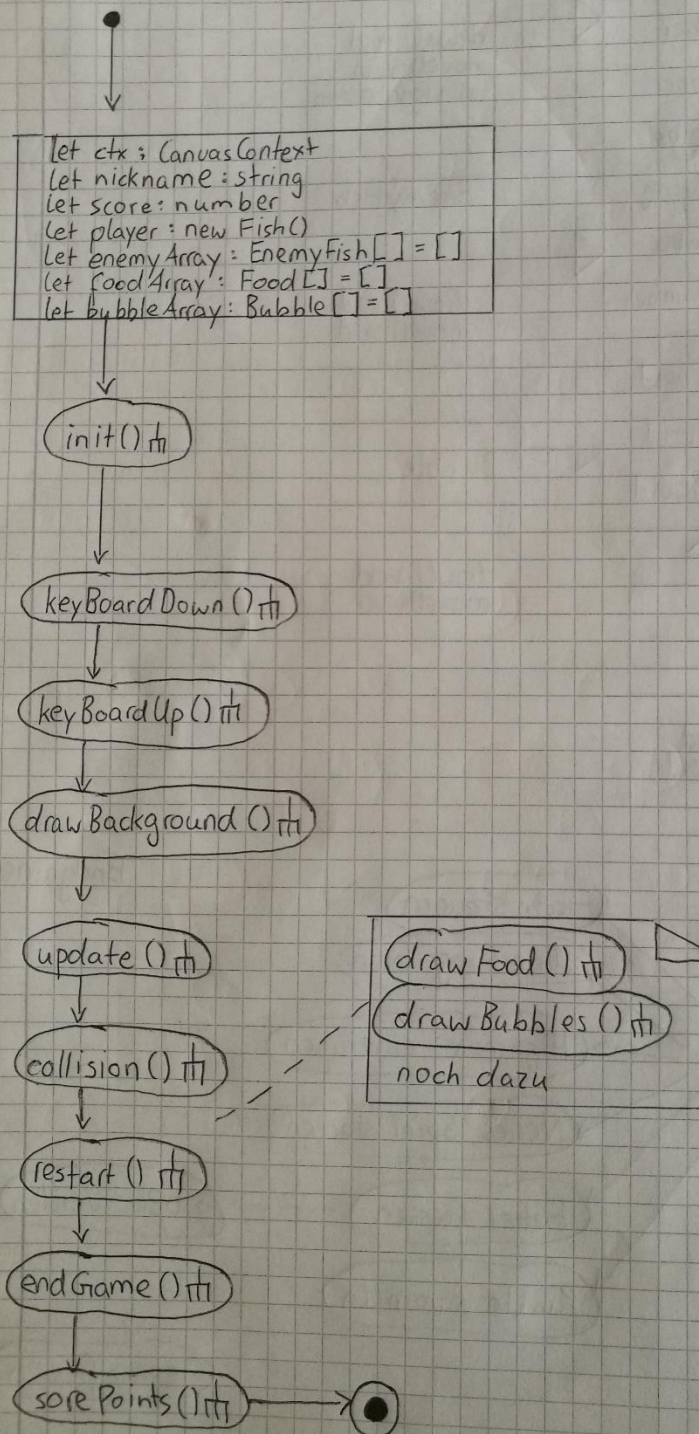
Zuletzt spawnen alle zehn Sekunden eine neue Luftblase. Nur der Spielerfisch kann von ihnen beeinflusst werden. Sollte er sie berühren, wird seine Bewegungsgeschwindigkeit stark erhöht und der Spieler muss die Kontrolle wiedererlangen. Die Luftblasen wandern von unten nach oben und werden danach wieder zufällig auf der X -Achse im unteren Ende des Canvas platziert, von wo sie wieder nach oben wandern. Das bedeutet, dass mit der Zeit immer mehr Luftblasen im Spiel existieren, welches den Schwierigkeitsgrad stetig erhöht.

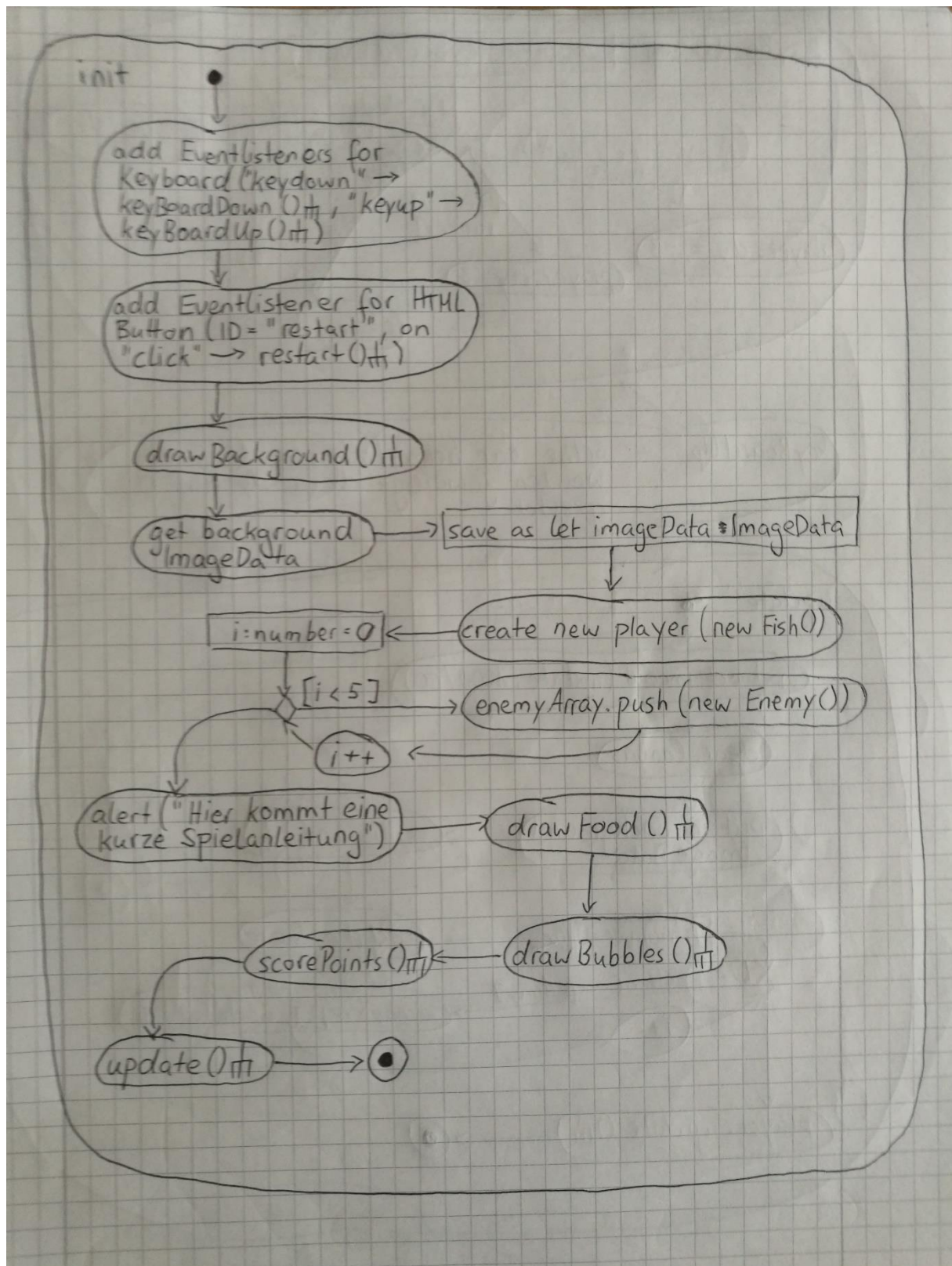
Der Spieler wird beim Aufrufen der index.html mit einer Alert-Nachricht begrüßt, die kurz die Spielmechanik beschreibt, danach befindet er sich direkt im Spielfluss und muss schnell reagieren. Bei Game Over durch Feind- oder Kantenkontakt soll er sich mit einem Namen für die Datenbank einschreiben. Danach kann er mit Hilfe des Buttons unter dem Canvas ein neues Spiel starten.





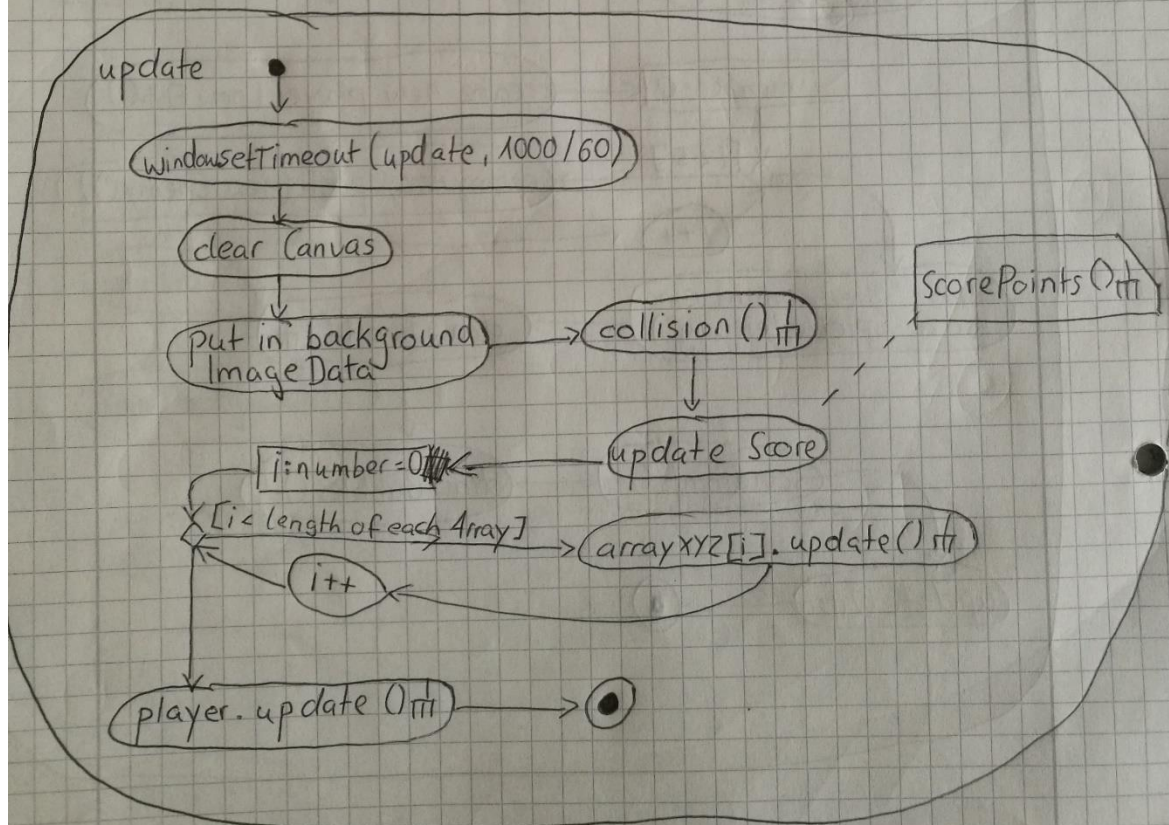
Canvas.ts

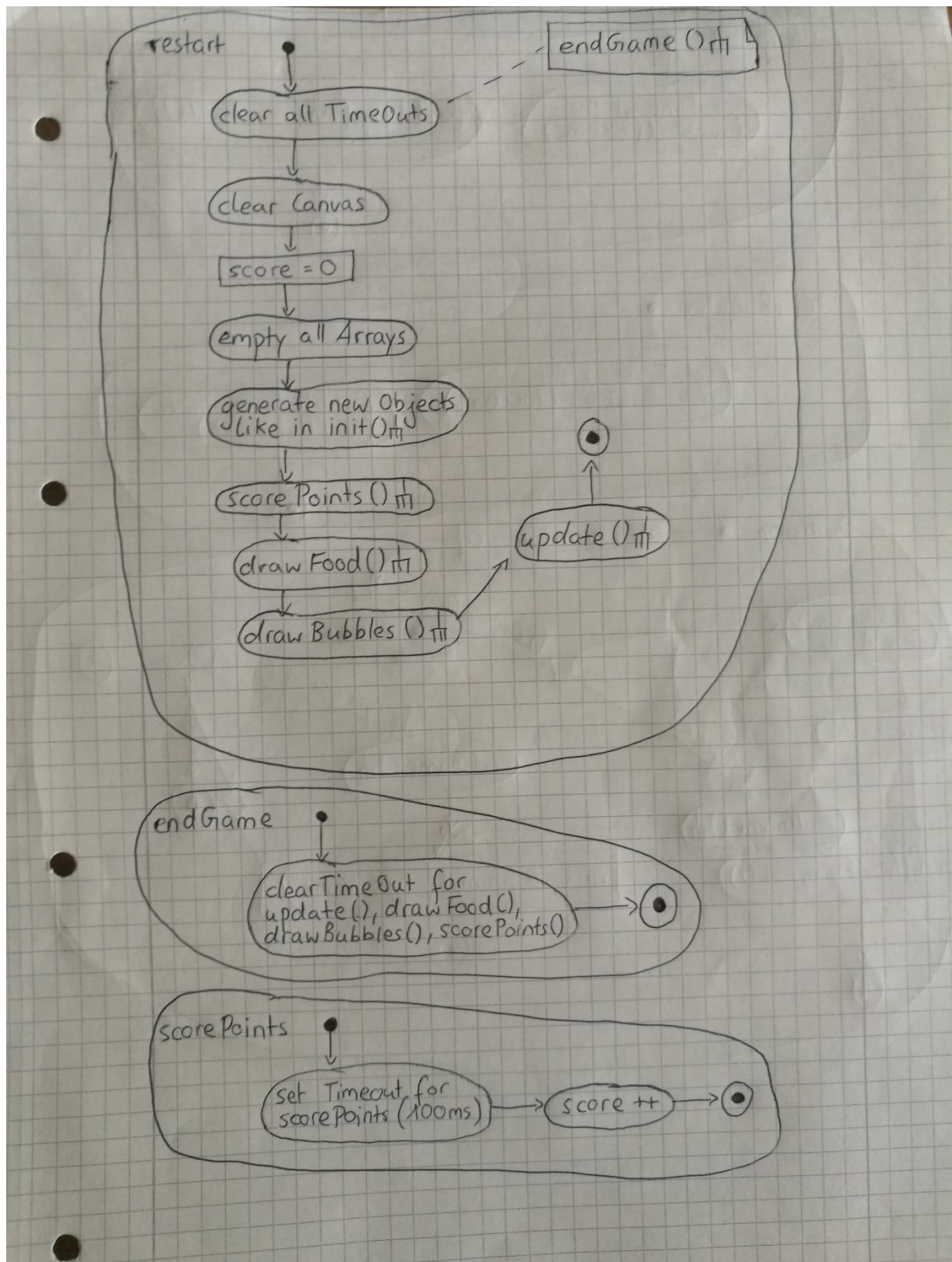


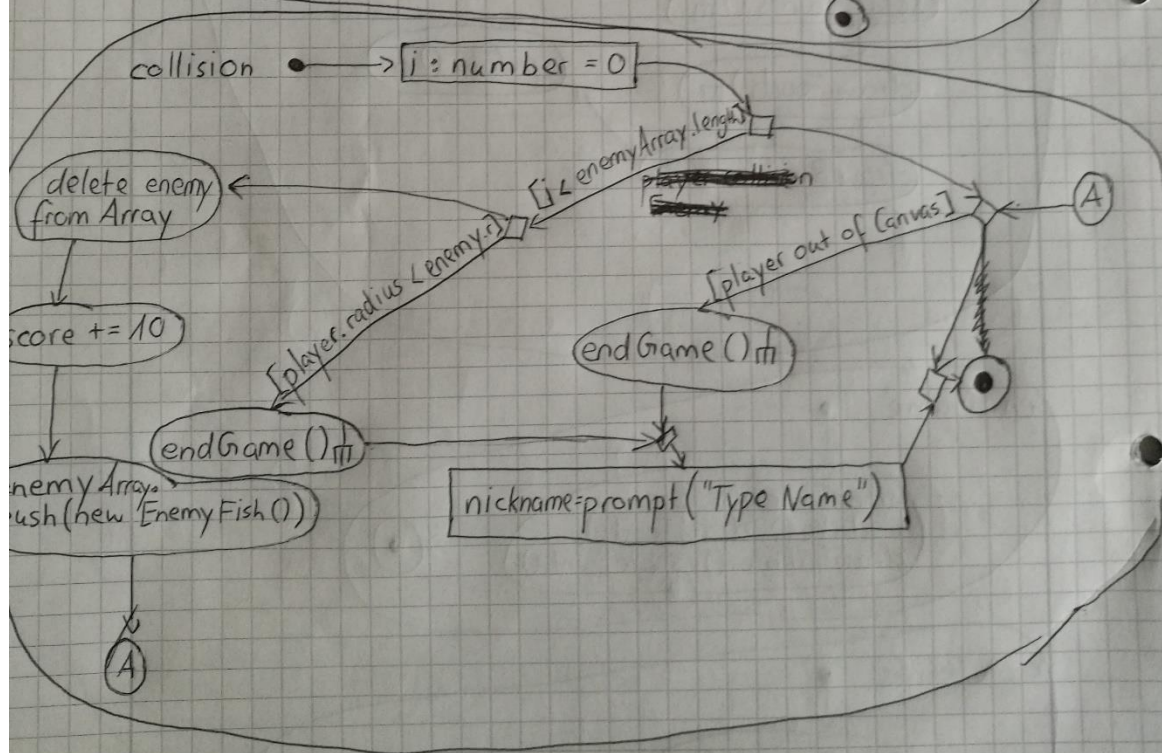
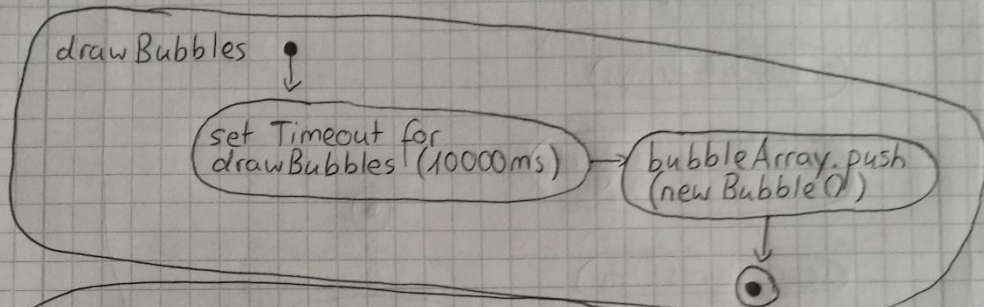
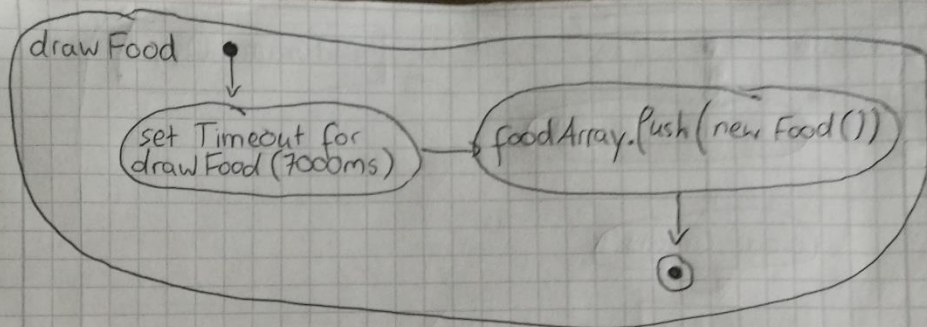




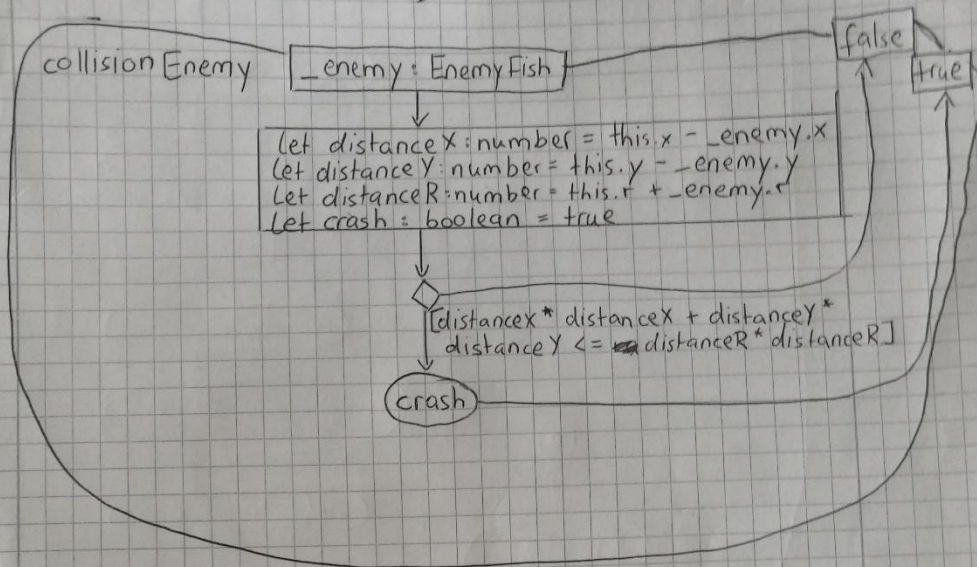
keyBoard Up = selbe nur beim Loslassen der Tasten werden jeweilige dx u. dy auf 0 gesetzt



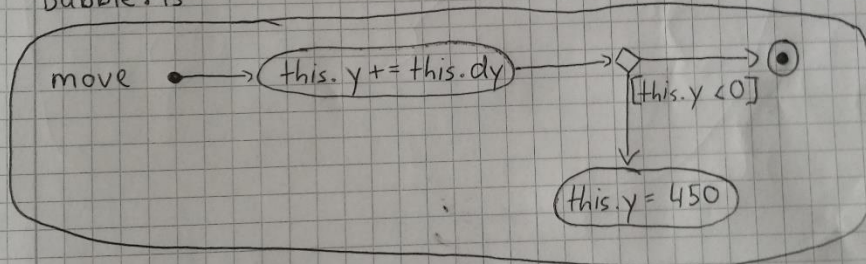




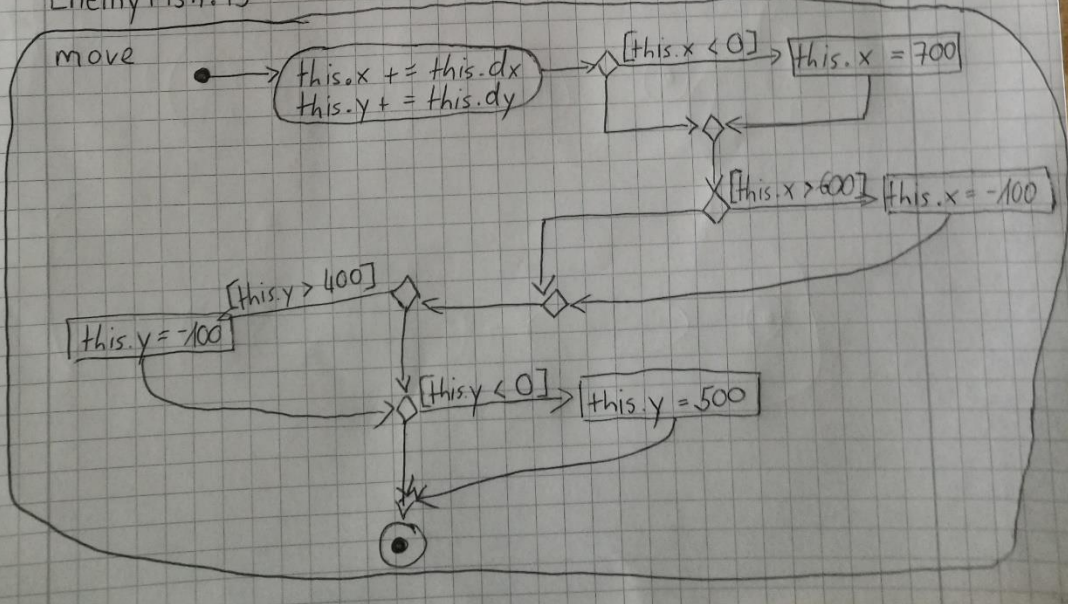
Fish.ts



Bubble.ts



EnemyFish.ts



Worm

```
ctx: CanvasRenderedContext2D  
name: string  
id: number  
score: number  
position: Coordinate  
speed: Coordinate  
angle: number  
length: number  
maxSize: number  
size: number  
color: string  
  
draw(): void  
move(): void  
checkEndOfMap(): void  
collisionFood(): void  
collisionWormAI(): void  
increaseScore(): void  
increaseSize(): void  
gameOver(): void
```

WormAI

Coordinate

```
x: number  
y: number
```

Food

```
ctx: CanvasRenderedContext2D  
pos: Coordinate  
sizeMin: number  
sizeMax: number  
color: string  
size: number
```

Game

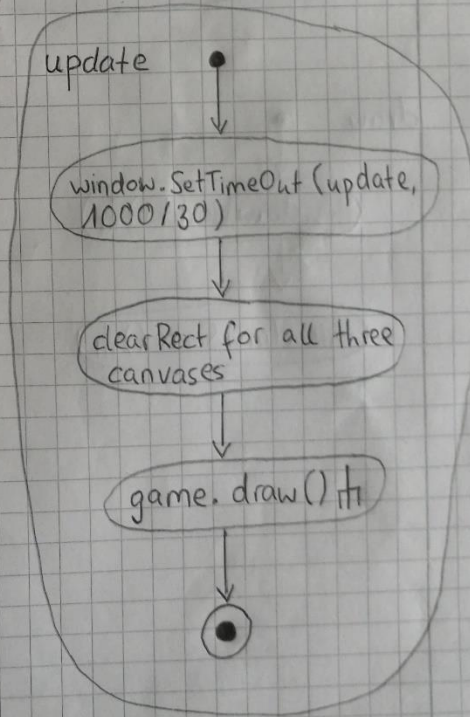
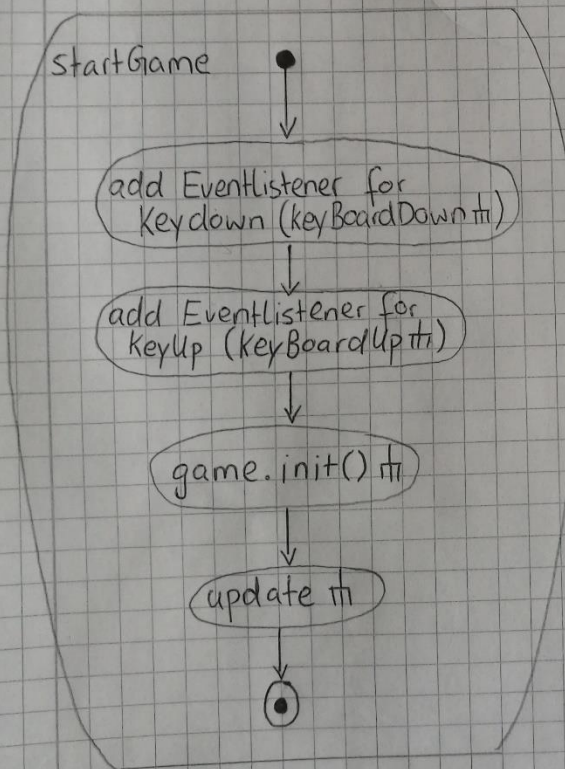
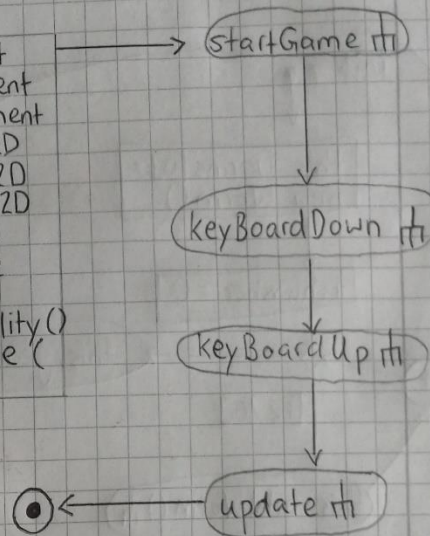
```
ctxWorm: CanvasRenderedContext2D  
ctxFood: CanvasRenderedContext2D  
ctxWorld: CanvasRenderedContext2D  
worldSize: Coordinate  
screenSize: Coordinate  
world: Coordinate  
worms: Worms[]  
food: Food[]  
  
init(): void  
draw(): void  
drawWorld(): void  
drawScore(): void  
newWorm(): void  
generateFood(): void
```

```
draw(): void  
eaten(): void
```

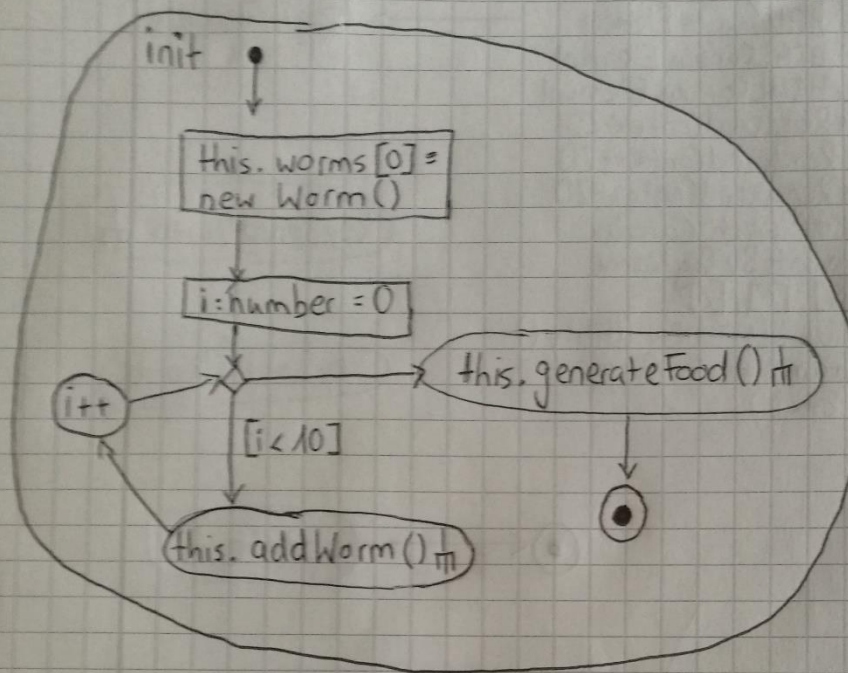

Canvas.ts

```

let canvas1: HTMLCanvasElement
let canvas2: HTMLCanvasElement
let canvas3: HTMLCanvasElement
ctx1: CanvasRenderingContext2D
ctx2: CanvasRenderingContext2D
ctx3: CanvasRenderingContext2D
get Context of each Canvas
per getElementByTagName
("canvas")[0], [1], [2]
let utility: Utility = new Utility()
let game: Game = new Game(
  ctx1, ctx2, ctx3)
  
```



Game.ts



draw

