

# Feed the Squirrel

## Funktions Analyse:

Der Spieler sieht zunächst einen Startbildschirm, welchen er mit einem Klick auf den Canvas durch ein kurzes Tutorial ersetzt. In dem das Ziel des Spieles, die Steuerung und die Vergabe der Punkte kurz erklärt wird.

Dieses Tutorial kann man ebenfalls durch einen Klick auf den Canvas mit dem Spiel ersetzen.

Im Spiel kann der Spieler ein Eichhörnchen mit der rechten und linken Pfeiltaste (oder mit dem linken und rechten Button) nach links oder nach rechts bewegen.

Von der oberen Canvas Seite fallen verschiedene Objekte nach unten. Der Spieler muss „essbare“ Objekte fangen und „nicht essbaren“ Objekten ausweichen.

Mit essbaren Objekten sind Objekte gemeint, die beim Fangen Punkte geben, sobald diese allerdings von dem Spieler nicht gefangen werden und die untere Canvas Seite erreichen, wird dem Spieler ein Leben abgezogen.

Nicht essbare Objekte ziehen dem Spieler beim Fangen Punkte ab, wenn sie jedoch unberührt die untere Canvas Seite erreichen, passiert nichts.

Es gibt unterschiedliche Objekte mit unterschiedlich hohen Punkten/Minuspunkten, es ist auch möglich durch ein Objekt ein Leben zu bekommen. Die Wahrscheinlichkeit, dass ein Objekt gezeichnet wird, ist für jedes Objekt unterschiedlich groß.

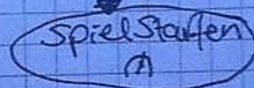
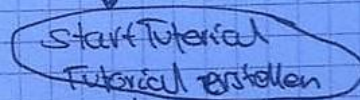
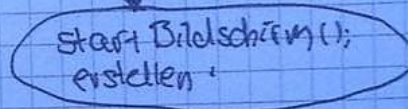
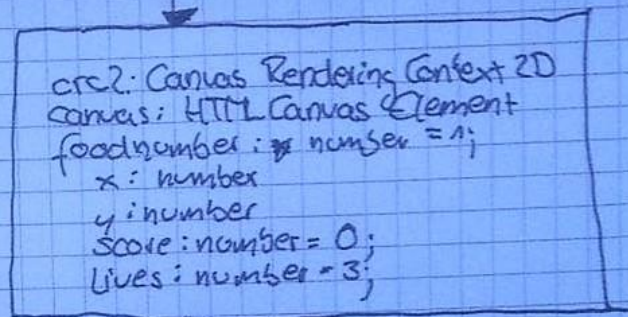
Der Spieler startet mit null Punkten und drei Leben. Das Spiel ist beendet, sobald der Score unter 0 Punkten ist oder dem Spieler seine Leben ausgehen.

Zum Schluss sieht der Spieler noch seinen Highscore.

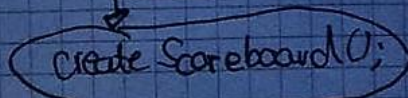
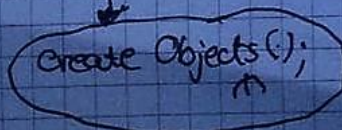
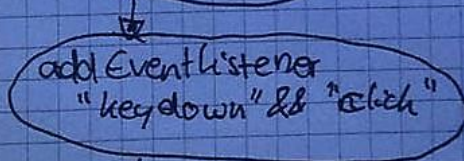
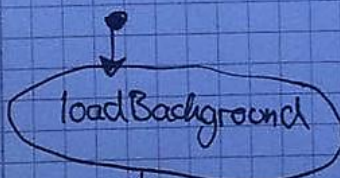
## Technische Konzeption:

Das Spiel basiert auf einem HTML Canvas und ist in Typescript / Javascript geschrieben.

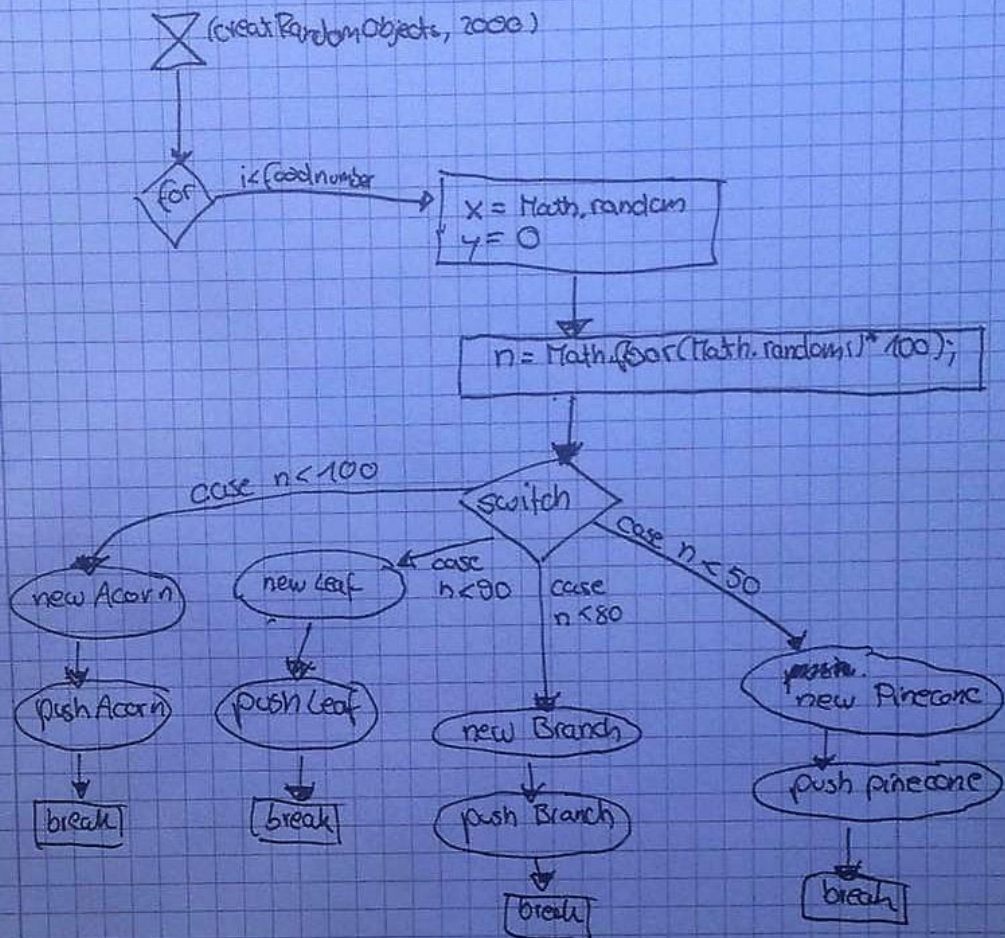
## Aktivitätsdiagramme und Klassendiagramme:



Spiel Starten():

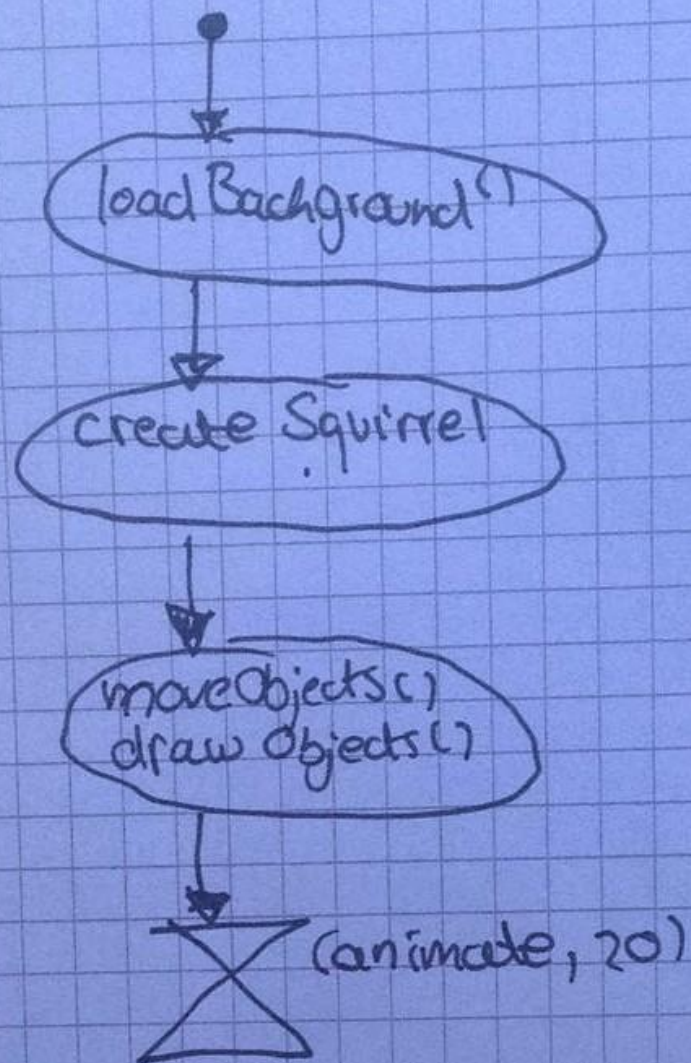


create Random Objects !:





animate():



# class Falling Object

x: number  
y: number  
type: string

draw();  
move();

constructor

(x: number, y: number,  
type: string)

{ this.x = x;  
this.y = y;  
this.type = type;

Pinecone

move();  
update();

Branch

move();  
update();

Leaf

move();  
update();

Acorn

move();  
update();



```

class Squirrel
{
    x: number;
    y: number;

    constructor (-x: number, -y: number)
    {
        this.x = -x;
        this.y = -y;
        this.move();
        this.draw();
    }

    move();
    draw();
    catch();
}

```

catch():

