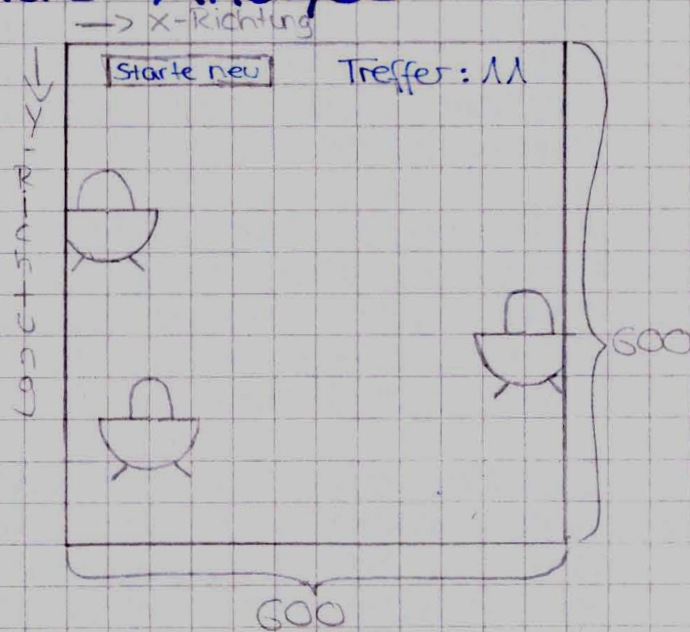



(A) Funktionale Analyse



Button **Starte neu** : Seite wird neu geladen, sodass der Nutzer von vorne anfangen kann.

Treffer: 11 : Zählt die abgeschossenen Ufos mit. Wenn ein Ufo getroffen wurde, kann es kein zweites mal abgeschossen werden.

Ufos  : Können durchs Anklicken abgeschossen werden. Wurde eines getroffen, fliegt dies nicht in x-Richtung weiter, sondern "fällt in y-Richtung runter".

Unterscheidung von 2 Klassen:

Ufos
x: number y: number color: string colorbody: string colorAlien: string status: boolean speed: number
constructor() update(): void move(): void draw(): void fall(): void

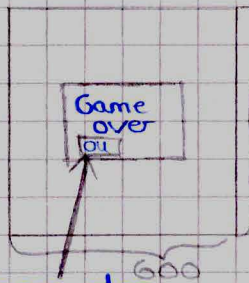
UfosRechts

$x = 600$
 $y = \text{Math.random()} * 500$

UfosLinks

$x = 0$
 $y = \text{Math.random()} * 500$

Canvas



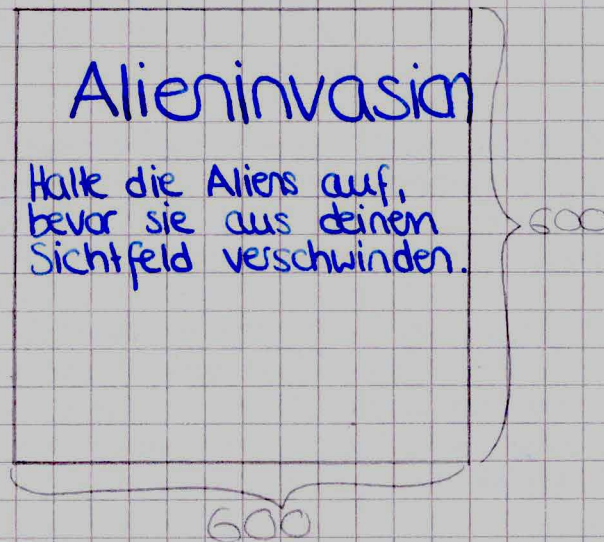
Drückt man den Button "OU" wird die Seite neu geladen und man kann das Spiel neu anfangen.

Fliegen die Ufos links oder rechts aus dem Canvas, werden sie in einem Array gespeichert. Sind in diesem Array mehr als 10 Ufos, hat man das Spiel verloren und es erscheint ein alert("Game Over").

Hat man ein Ufo abgeschossen und dieses fliegt in y-Richtung aus dem Canvas, wird es aus dem Array mit den ganzen Ufos gelöscht.

Durch ein bestimmtes Intervall wird immer wieder ein Ufo rechts und links hinzugefügt. Es wird im Array der Ufos gespeichert.

Startbildschirm:



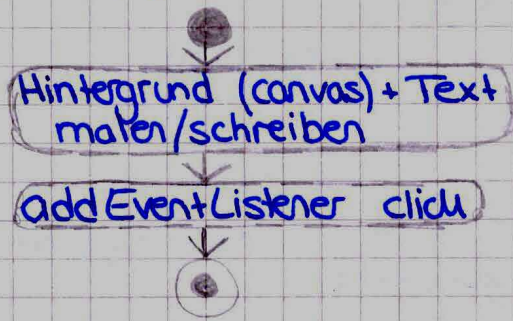
Klick auf den Canvas lässt das Spiel beginnen.

Nutzerinteraktionen & die zugehörige Reaktion

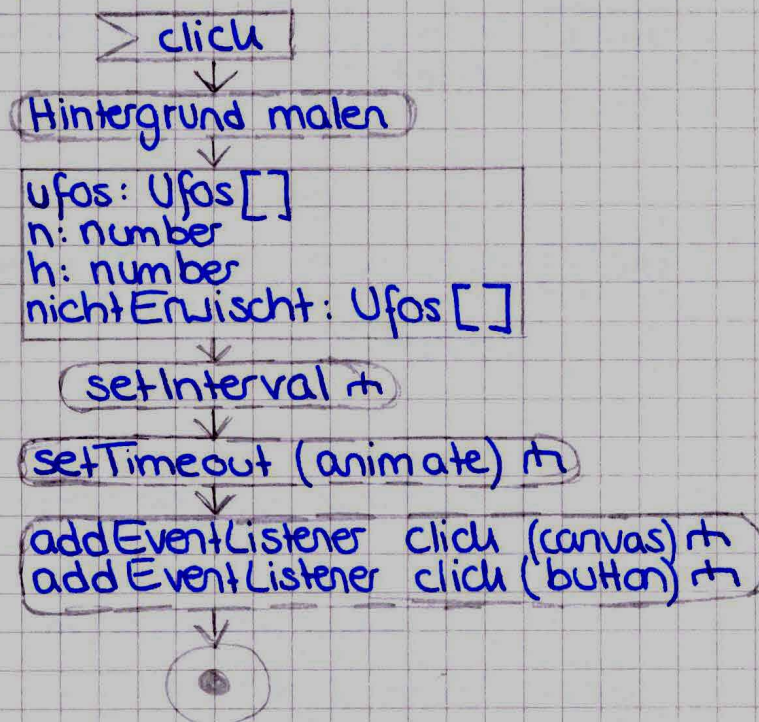
1. Startbildschirm Klick auf Canvas:
Spiel beginnt
2. Klick auf "Starte neu" Button:
Seite lädt neu → neues Spiel
3. Klick auf Ufo:
Abgeschossenes Ufo fliegt in y-Richtung aus Canvas
4. Klick auf "OU" beim alert("Game Over")
Seite lädt neu → neues Spiel

(2) Technische Analyse

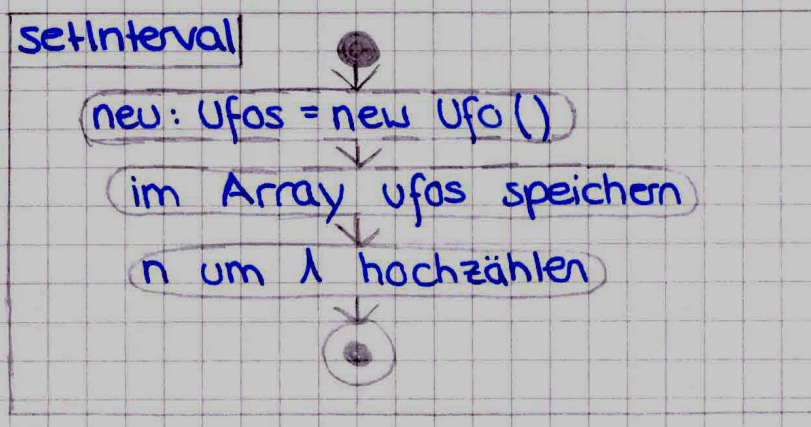
Startbildschirm:



Spielaufbau:



Spielfunktionen:



setTimeout(animate)

