

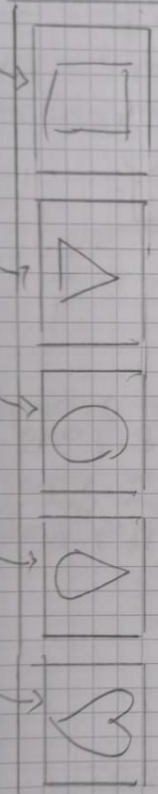
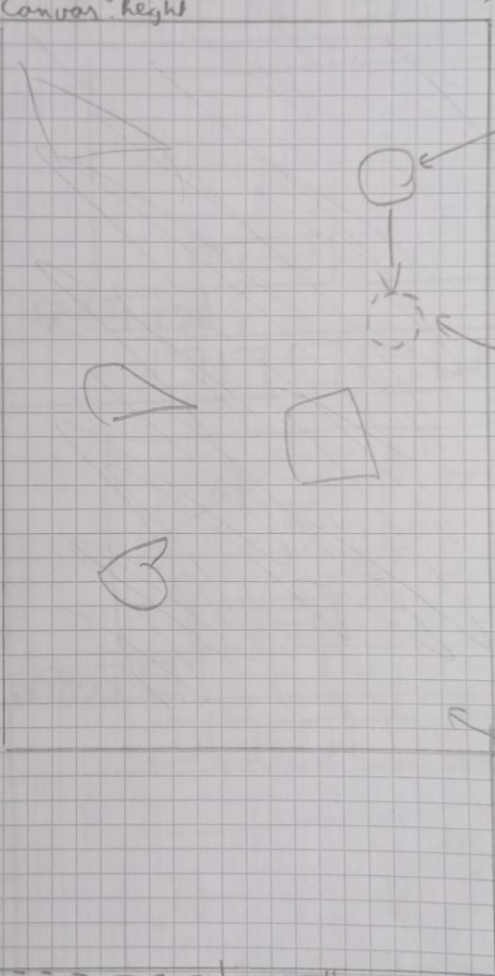
Move object with mouse down + move

U1 - Scribble

Draw object in new position on mouse up canvas width

Draw Background with last height last BG color

- * New = reload
- Load = previously saved
- Save Current
- Delete Current



SAVE POS

Values for Create Background

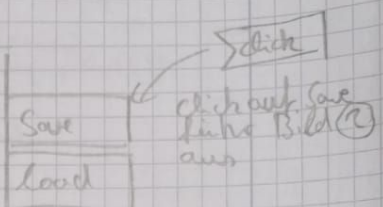
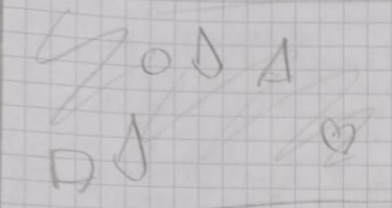
Save Position of all Objects

All Buttons down with circle, get triggers with mouse down on their position

Save Position of all Objects

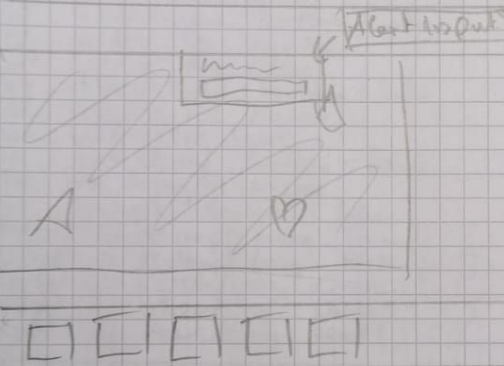
User Interface Save/Load

①



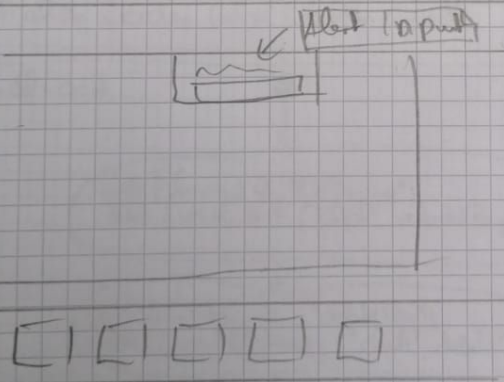
Click auf Load führt zu Bild 3

②



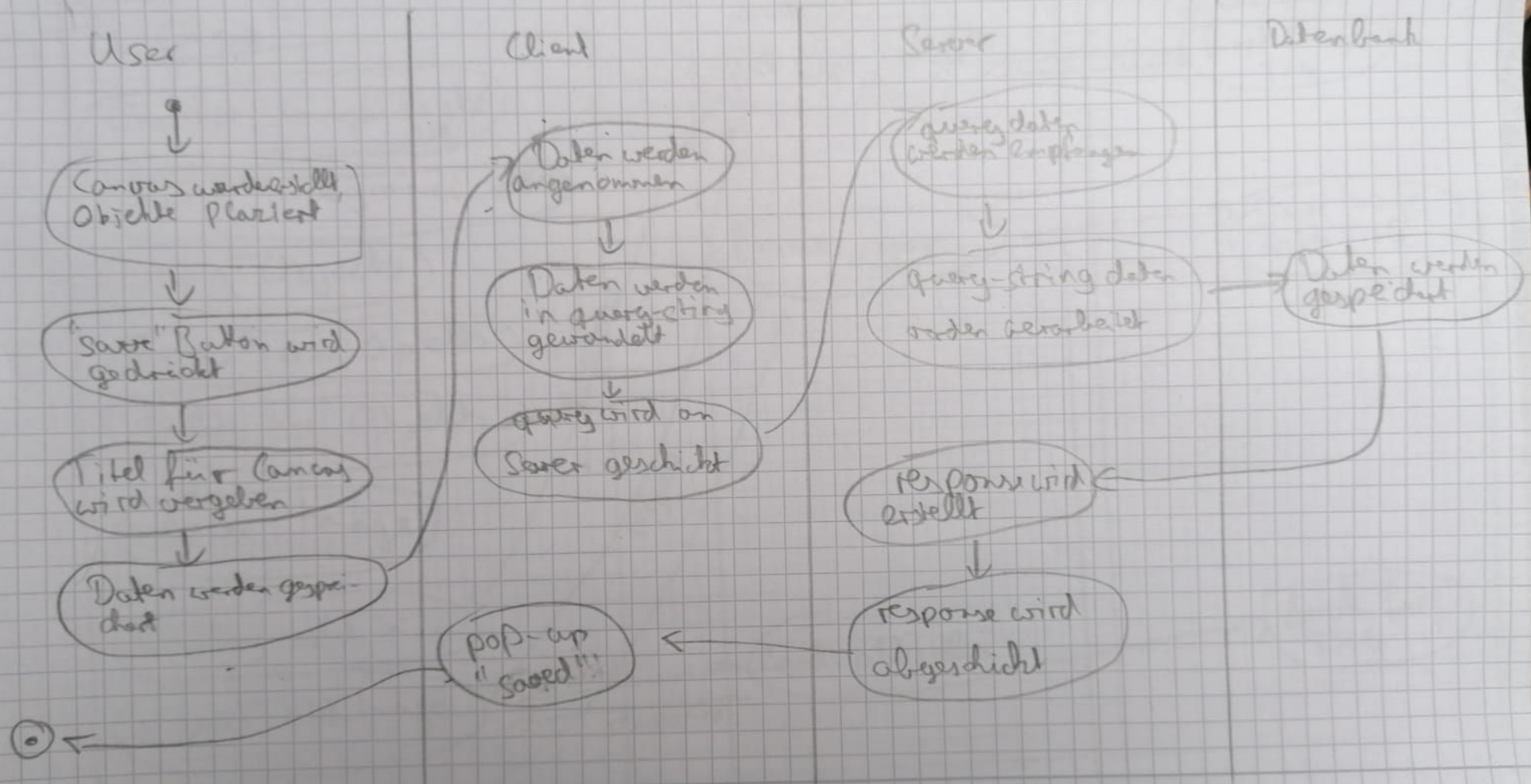
Save
Ein Alert Popst auf in dem man einen Namen eingeben muss.
Das Bild wird unter diesem gespeichert

③

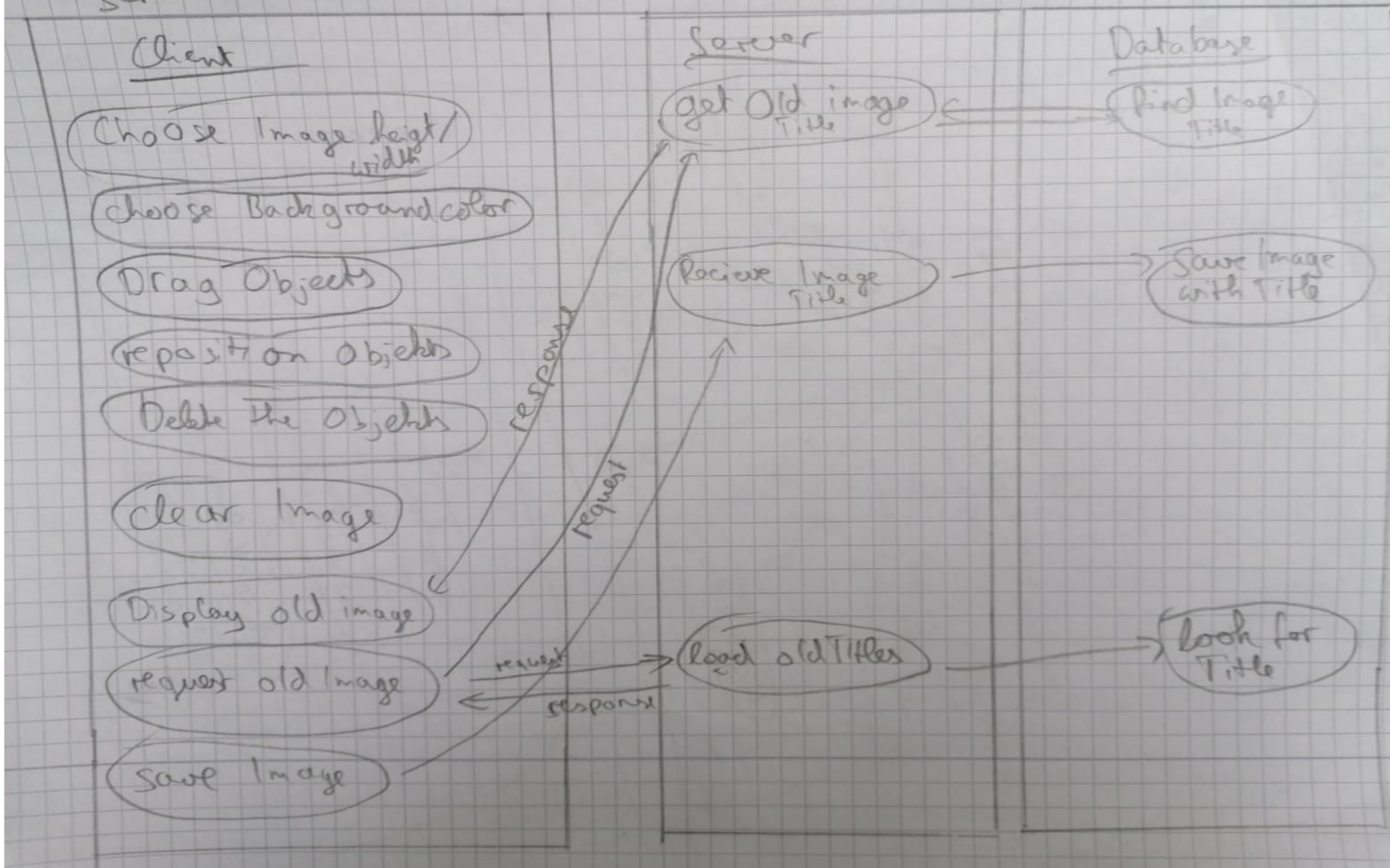


Load
Beim Click auf Load öffnet sich ein Alert, Dort kann man seine zuvor erstellten Bilder neu laden.

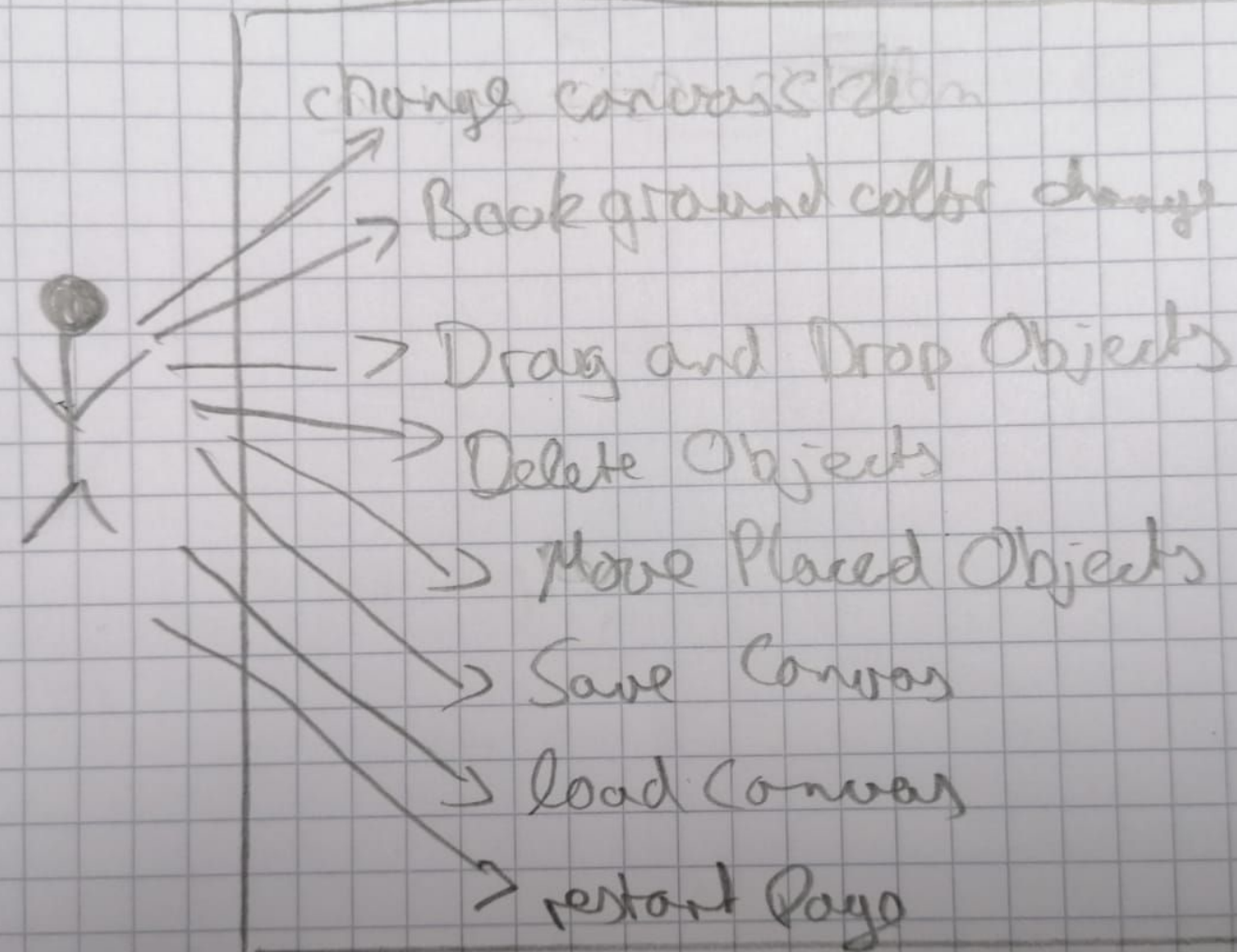
Dom übergreifendes Aktivitätsdiagramm



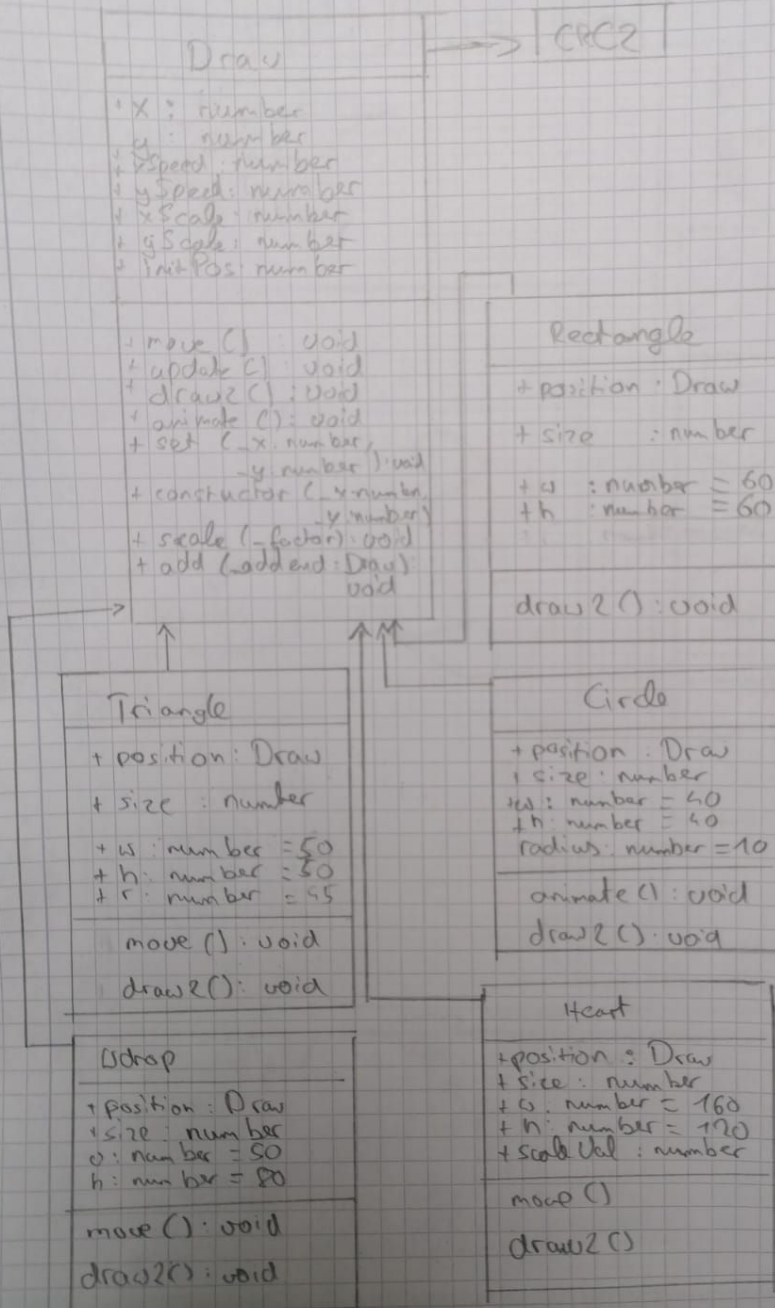
Server - Client - Communication



Use case Diagram



Class Diagram



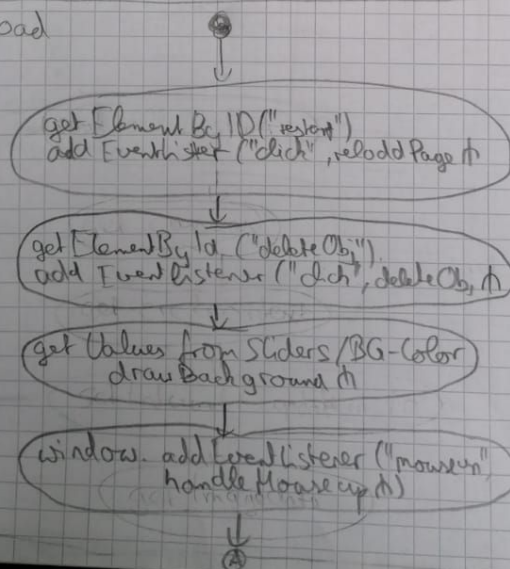

```

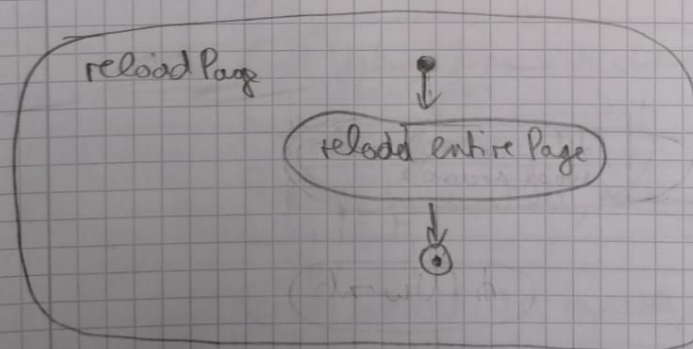
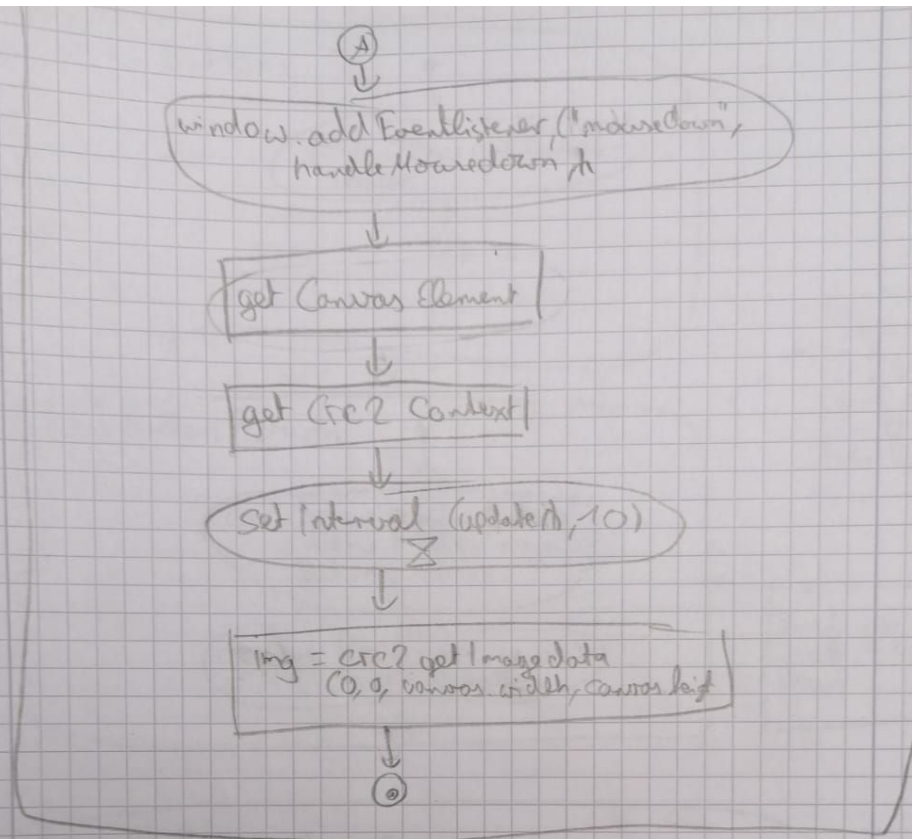
window.addEventListener("load", handleLoad());
export let canvas: HTMLCanvasElement;
export let ctx: CanvasRenderingContext2D;
let img: ImageData;
let widthSlider: HTMLInputElement;
let heightSlider: HTMLInputElement;
let colorPicker: HTMLInputElement;
export let width: number;
export let height: number;
let currentRectangle: Rectangle;
let rectanglePresent: boolean = false;
let recArrayPos: number;
let currentTriangle: Triangle;
let trianglePresent: boolean = false;
let triArrayPos: number;
let currentCircle: Circle;
let circlePresent: boolean = false;
let cirArrayPos: number;
let currentUdrop: Udrop;
let udropPresent: boolean = false;
let udrArrayPos: number;
let currentHeart: Heart;
let heartPresent: boolean = false;
let heaArrayPos: number;
let dragged: boolean = false;
export let triangleArray: Triangle[] = [];
export let rectangleArray: Rectangle[] = [];
export let circleArray: Circle[] = [];
export let udropArray: Udrop[] = [];
export let heartArray: Heart[] = [];

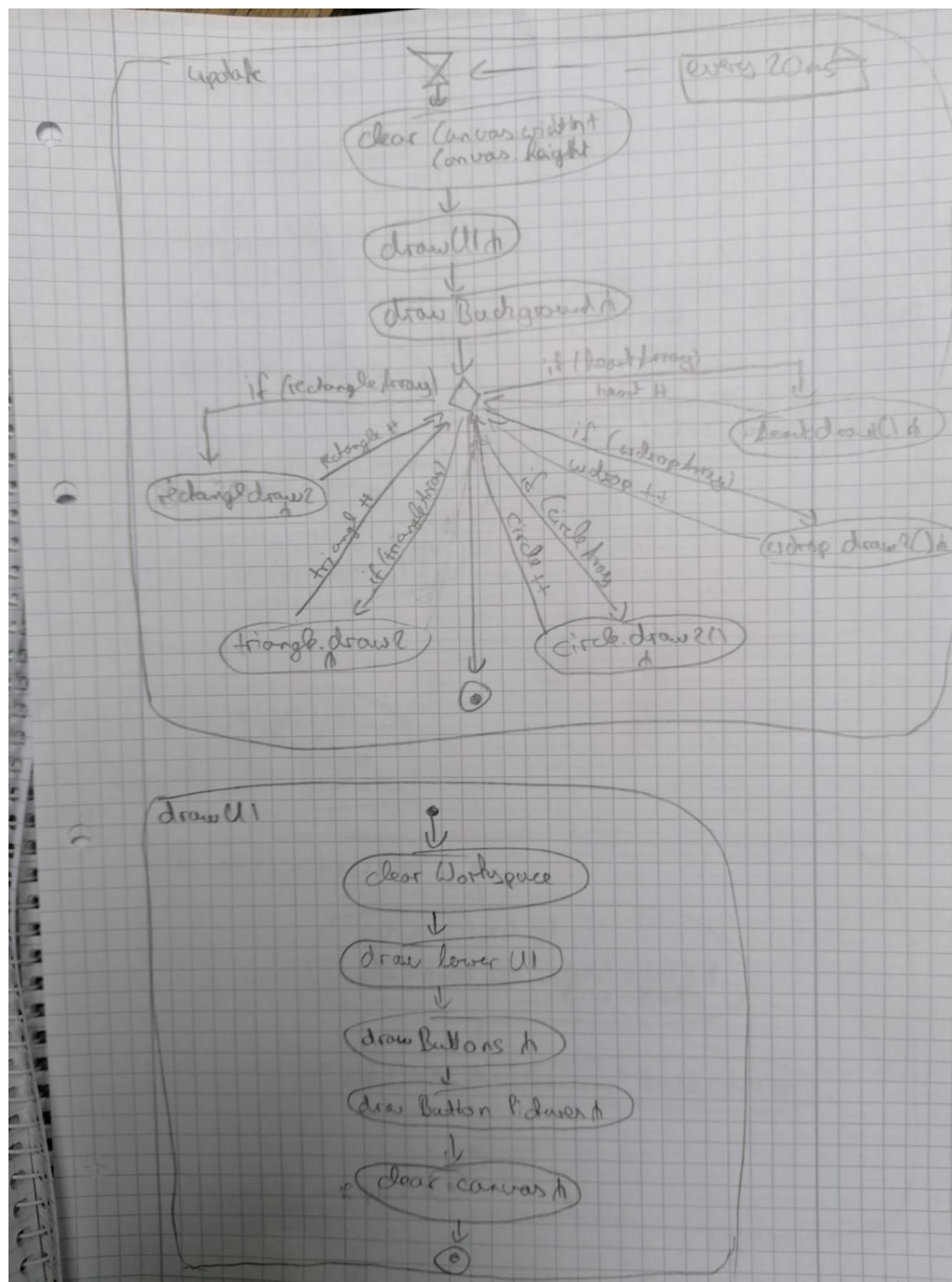
```

Variablen werden beim Starten erstellt

handleLoad







Draw Buttons

arc2 Draw all Button -
Shapes



arc

Draw Button Pictures

arc2 draw Specific Button
Element on previous
Button position

Declare Position x, y and
width + height of Button



Draw all
Elements
with different
functions into
the Buttons

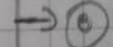
Rectangle, Triangle,
Circle, Outlined,
Filled

Draw Background

get Values from UTM to last
Bg - Color

Labels = newWidth, newHeight

arc2 fillRect (0,0, newWidth, newHeight)

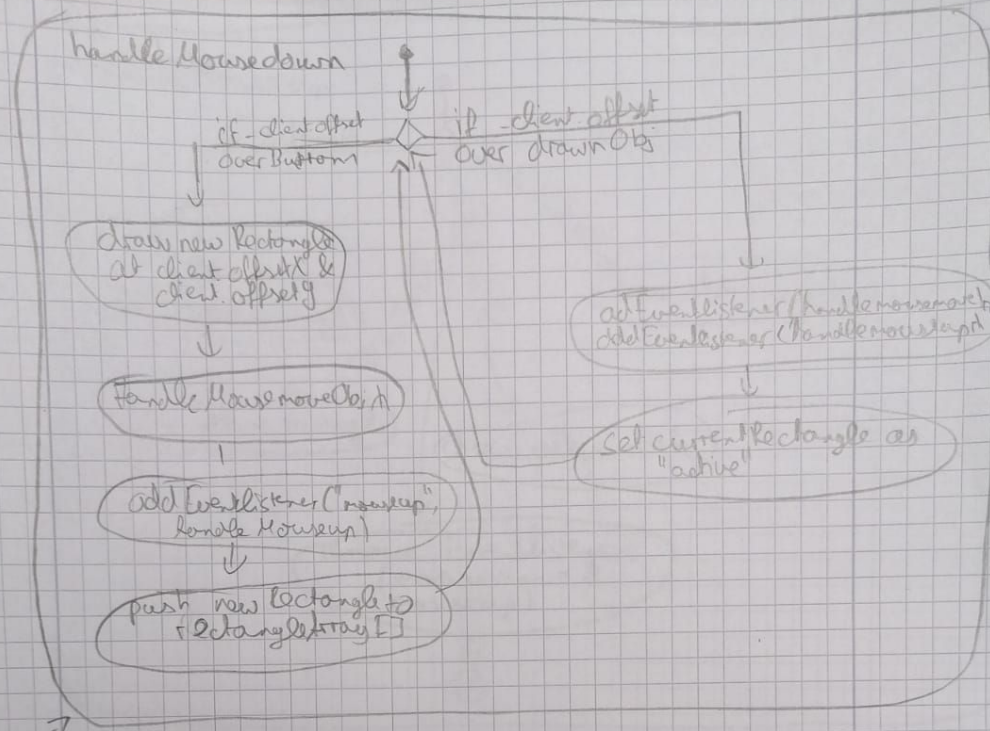


Clear Canvas

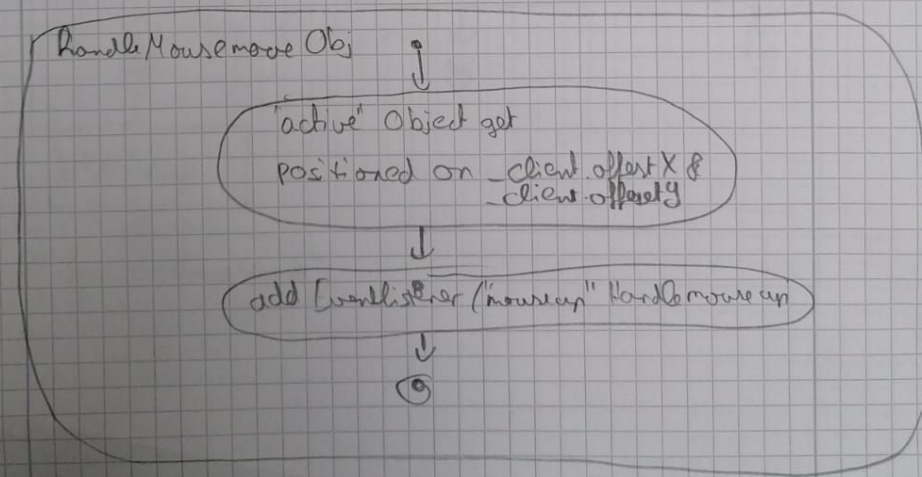
Get width and height
of overlapping workspace

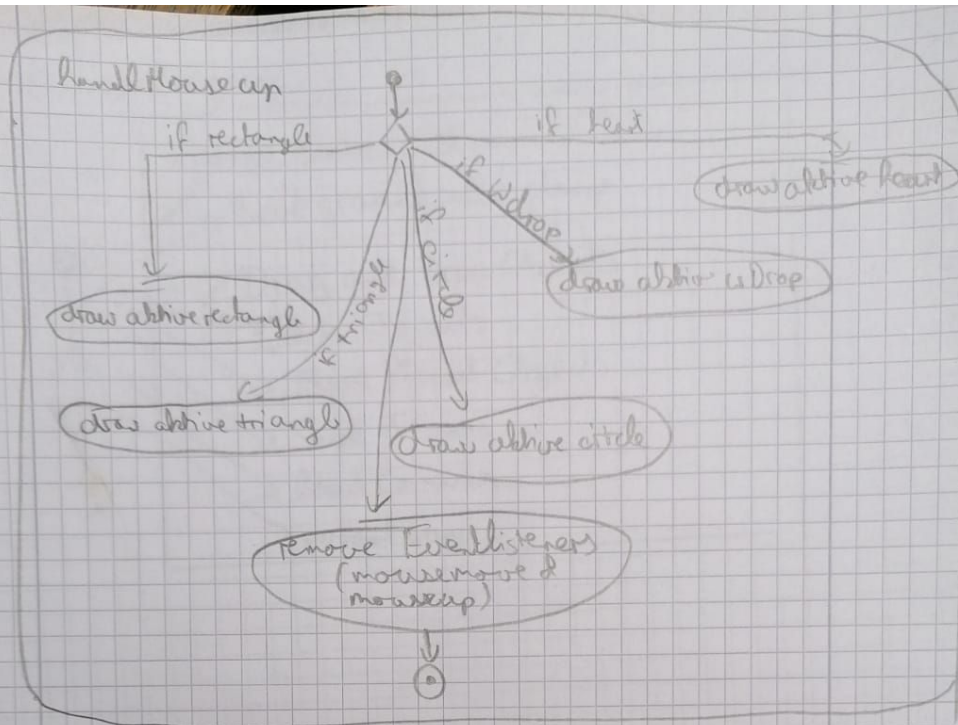
clearRect Rest of UI





if Bedingung bezieht sich nicht nur auf Rectangle, sondern auf alle Objekte. Sie steuern die selben Funktionen an.





load Window

handle load d



MouseDown

handle left click d



mouse move

handle mousemoveObj d



Mouse up

handle Mouse up d

