

Was brauche ich:

1. Canvas erstellen beim Start dyn. pxl.
Anzahl evtl. limitiert auf maximale Größe
2. Canvas Elemente erstellen
3. Drag and drop func.
4. Formular mit Anzeige der platzierten Elemente
5. Animation der einzelnen Elemente
6. Möglichkeit Animation im Formular zu ändern
7. Löschen von platzierten Elementen
8. absenden von allen Elementen in die DB
9. Laden der Daten aus der DB
10. Ablage für Symbole

Main

- ↪ Event listeners (load, click, mousedown/up, send)
exemplified Ver
- handle Load = interface und canvas Zeichnen
 - ↪ update func aufrufen
- update func.
- draw interface
- save Picture
- unload Pic
- load Pic

Client

Server

HTML-Seite



update

Speed:



Rotation



color

Main:

move

rotate

index.html

Server / Client

Main.ts ✓

init. th ✓

update th ✓

draw th

move.ts

rotation.ts

color.ts

startPage th ✓

gamePage. th ✓

finalPage. th ✓

create RadioButton th ✓

handle Event Input th ✓

buttonTo Game Page th ✓

handle Event Slider th

loadCanvas th ?

createCanvas th ?

drag Geometry th ✓

manipulate Geometry th ✓

drawAll Objects th ✓

im den Klassen der Objekte
4 Eck 3 Eck Kreis etc.

Main.ts

eventListener

mouse

load

canvas

let festlegen global

↓
init ⌂

init.ts



startpage.th ✓

gamePage.th ✓

finalPage.th ✓

Start Page \rightarrow

HTML Page anzeigen

$id = \text{Startpage}$



Create Radio Buttons \rightarrow



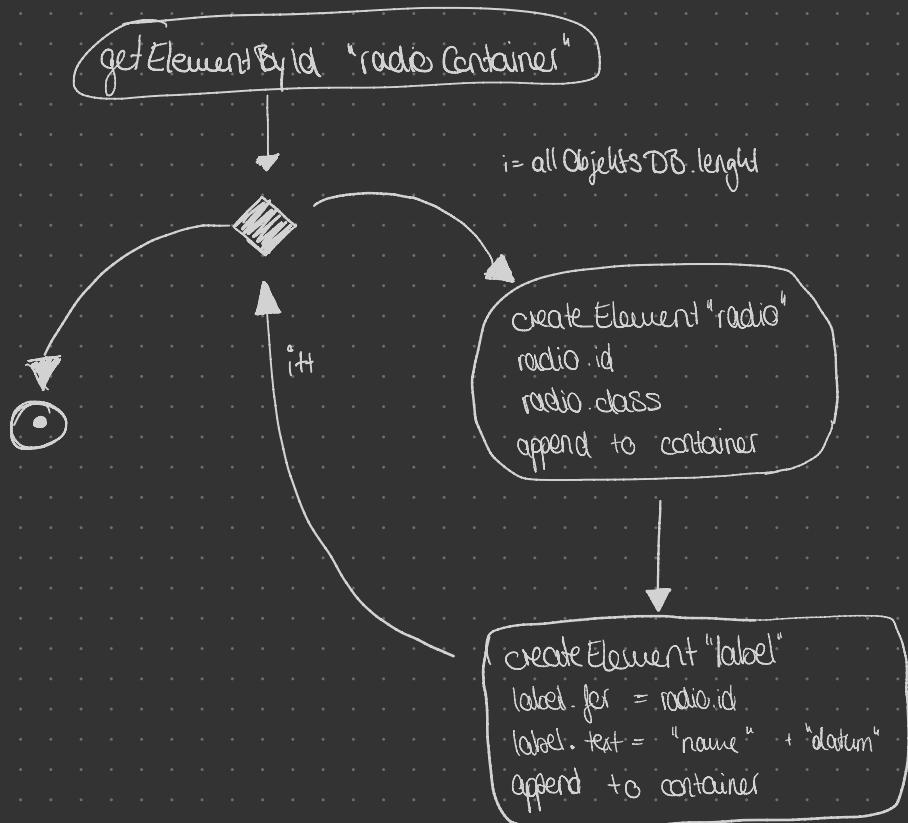
handle Event Input \rightarrow ✓



ButtonTo game Page \rightarrow ✓



Create Radio Buttons DB



handleEvent Input \vdash : Event (-event)

-event.target

RB
gespeicherte Bilder
→

Slider

handleEvent RadioButton \vdash



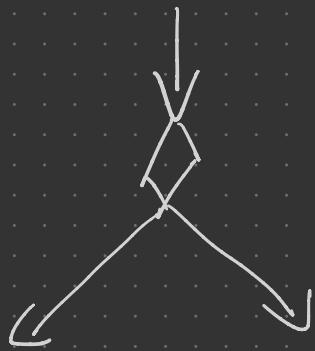
handleEvent Slider \vdash



handle Event RadioButton

dB

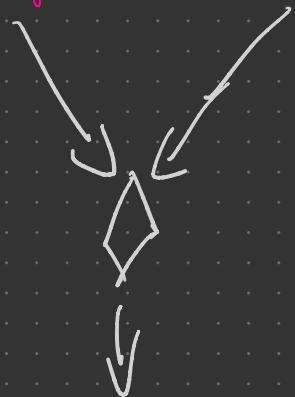
Button To Game Page ✓



Load CV ✓

Infos aus <input> auslesen

Create New CV ✓



Gamepage ✓ ✓



Same Page \rightarrow



Create Canvas \checkmark



DRAG Geometry \rightarrow



[img aus DB gewählt]

Load Array Elements
from DB \rightarrow

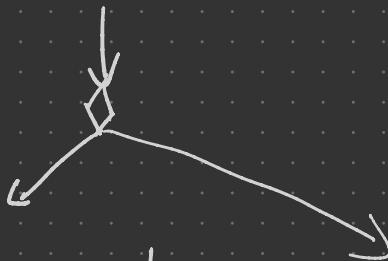


update \rightarrow



buttonToFinalPage \rightarrow

finalPage()



handleEvent SaveCanvas()



nameleEvent Restart()



Startpage() ✓

update it

[once it over
interface]

ingData

drag New Geometry it

→ would it make sense to add Eventlisteners
to all Objects? or is it better to have them
on the Array?

[Object von
Objects-Array = -event. target]

manipulate Geometry it

draw all Objects
in Array:
ObjectsToDraw

draw All Objects it

↓
savingData

drag New Geometry in

[if dragged Obj
is over drag target]

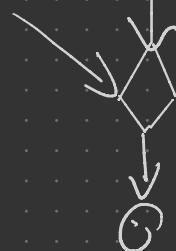
drag Enter



drag Drop



push Object Info
in ObjectsToDraw
Array



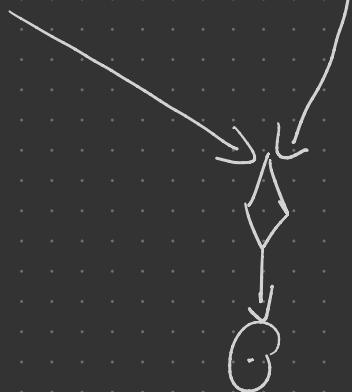
manipulate Geometry it



represent current
Properties (Speed, Rotation...) of clicked Obj.
in Sliders

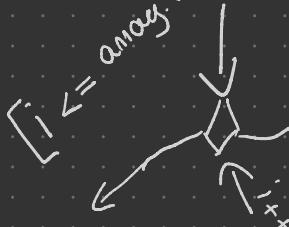
[if Properties settings changed]

Push new Values from Selected
Geo in ObjectsToDraw Array



draw All Objects ↗

$i \leq \text{array.length}$



draw Object[i]



