Funktionale Analyse

Klassendiagramm:

```
        +------------------------+
        |        general         |
        +------------------------+
        |  x : number            |
        |  y : number            |
        |  color : string        |
        +------------------------+
        |  move () : void {}     |
        |  draw () : void {}     |
        +------------------------+
                    △
                    |
        +------------------------+        +----------------------+
        |       Movement         |        |        birds         |
        +------------------------+   ◁――  +----------------------+
        |  dx : number           |        |  state : String      |
        |  dy : number           |        +----------------------+
        +------------------------+
        |  move : void {}        |◈
        +------------------------+ ↘
                    △                  +--------+
                    |                  |  snow  |
        +------------------------+     +--------+
        |       snowball         |
        +------------------------+
        |  time : number         |
        +------------------------+
```

Interface:

Menü:

```
        +----------------------------------+
        |                                  |
        |      +--------------------+       |
        |      |     Play Game      |       |
        |      +--------------------+       |
        |      |     Highscores     |       |
        |      +--------------------+       |
        |      |     Manual         |       |
        |      +--------------------+       |
        |                                  |
        +----------------------------------+
```

In - Game: cursor

clouds

snow

birds

tree

sun
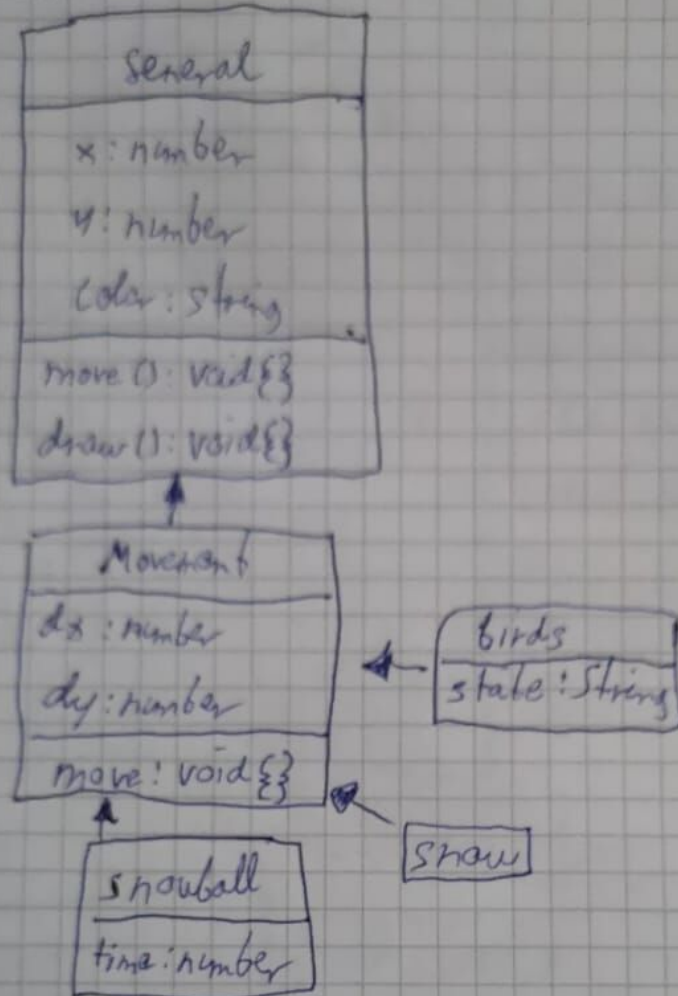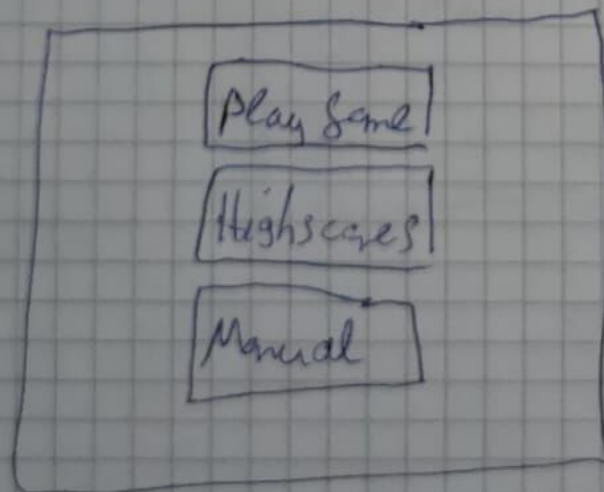
- bei linker-1click
und auf cursor-position
Schneeball geworfen
- bei 1click auf space
wird Fackel gewählt

Spielende:

Your Score:

-21466

Type your Name

send

Highscore-List:

Highscores

1.) Name 1 Sccre 1

2.) Name 2 Score 2

3.) Name 3 Score 3

Anwendungsfalldiagramme:

Spieler
- kann Spiel starten
- kann Manual lesen
- kann Bälle werfen und somit Score generieren
- kann Futter werfen und somit Vögel anlocken
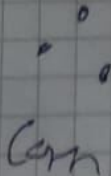- kann Name eintragen und Highscore einsehen

Birds
- können zu Ziel fliegen
- können zu Ziel zu Futter ändern und wieder zurück
- können durch abschießen einen Score generieren
- Spawnen an zufälliger Pos.
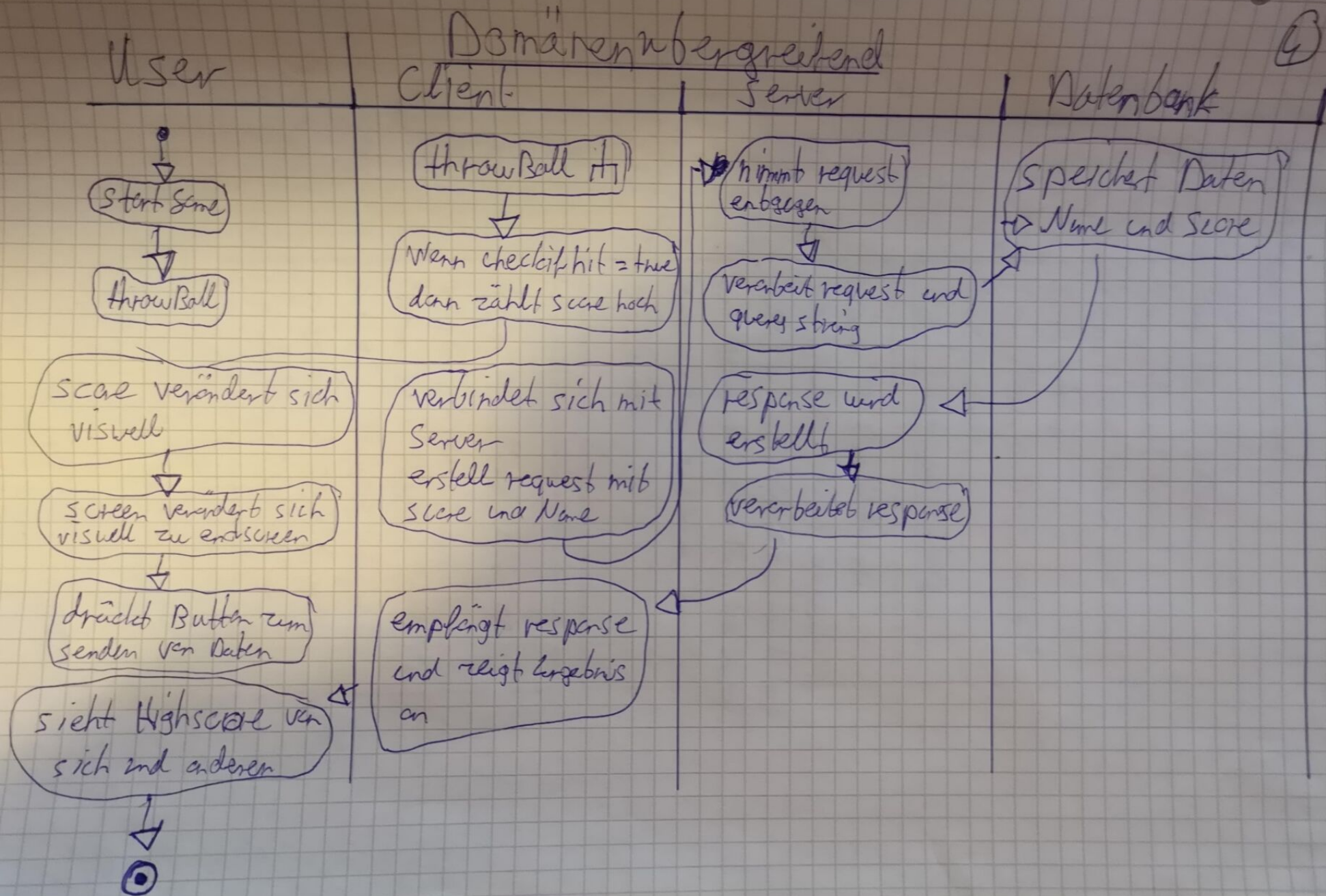- Haben zufällige Geschwindigkeit
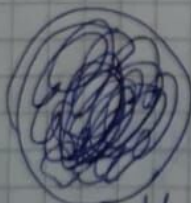
Snowball
- kann mit Klick geworfen werden
- kann mit Vögeln kollidieren
- hat eine Flugzeit und Größe abhängig zur Zeit

Corn
- kann mit Space geworfen werden
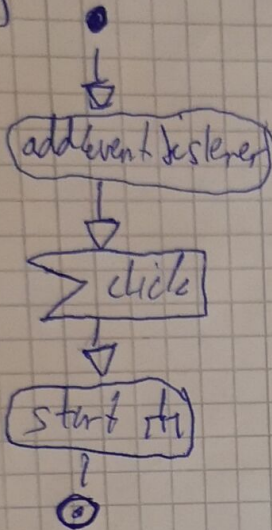- kann Vögel anlocken
- hat zufällige Lebenszeit

Domänenübergreifend

User	Client	Server	Datenbank

User:
- (Start) ●
- Start Game
- throwBall
- Score verändert sich visuell
- Screen verändert sich visuell zu endscreen
- drückt Button zum senden von Daten
- sieht Highscore von sich und anderen
- (Ende) ◉

Client:
- throwBall 🏀
- Wenn checkif hit = true dann zählt score hoch
- verbindet sich mit Server erstell request mit score und Name
- empfängt response und zeigt Ergebnis an

Server:
- nimmt request entgegen
- verarbeit request und query string
- response wird erstellt
- verarbeitet response

Datenbank:
- speichert Daten
- Name und Score

4

Technische Analyse

init

1.)

```
        ●
        ↓
   ┌─────────────────┐
   │ addEventListener │
   └─────────────────┘
        ↓
   ┌─────────────────┐
   │ ⚡ click         │
   └─────────────────┘
        ↓
   ┌─────────────────┐
   │ start ttt       │
   └─────────────────┘
        ↓
        ◉
```

2.)

```
        ●
        ↓
   ┌─────────────────┐
   │ addEventListener │
   └─────────────────┘
        ↓
   ┌─────────────────┐
   │ ⚡ click         │
   └─────────────────┘
        ↓
   ┌─────────────────┐
   │ showManual ttt  │
   └─────────────────┘
```

3.)

```
        ●
        ↓
   ┌─────────────────┐
   │ addEventListener │
   └─────────────────┘
        ↓
   ┌─────────────────┐
   │ ⚡ click         │
   └─────────────────┘
        ↓
   ┌─────────────────┐
   │ show Highscore ttt │
   └─────────────────┘
```
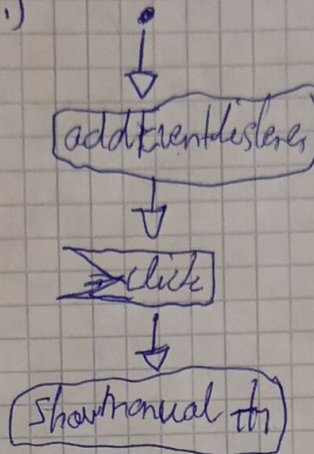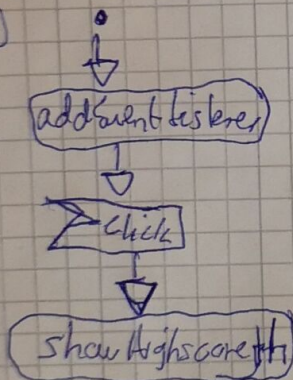
update

start

↓
(Menü screen wird verbergen)
↓
(Canvas wird angezeigt
document.getElementsByTagName
(canvas)[0].style.display initial)
↓
(Scorefeld anzeigen lassen)
↓
(add event listener)
↓
[click]
↓
(throw Ball ())
↓
(create Child ())
↓
(Schnee wird erstellt)
↓
show wird in array
gepusht Snowflakes.push (snow)
↓
(update ())
↓
◎

update

```
                    •
                    ↓
    ┌─────────────────────────────────────┐
    │ Window.setTimeout (update 1000/fps)  │
    └─────────────────────────────────────┘
                    ↓
           ┌──────────────────┐
           │ let i : number = 0 │
           └──────────────────┘
                    ↓
         ┌─┐    ◇ ──▷     Ⓐ
    (i++)◁┘    │
                    ↓
           ┌──────────────┐
           │ child[i].move │
           │ childd[i].draw │
           └──────────────┘
                    ↓
           ┌──────────────────┐
           │ child.splice(i,1); │
           └──────────────────┘
                    ↓
           ( create child +i )
```

```
Ⓐ ──▷ ┌──────────────────┐
       │ let i : number = 0 │
       └──────────────────┘
              ↓
   (i++)──▷ ◇ [ i = balls.length ]
        │
        ↓
      else  ◇ ── balls[i].time = 0
                      ↓
( snowballs[i].draw )      ( snowballs[i].draw )
                                ↑
( score += child[i].speed )   ┌──────────────────┐
                              │ let i1 : number = 0 │
        ↓                     └──────────────────┘
( child[i].state = shot )─▷  (i++)◇ ──▷ ( update )──▷ Ⓑ
                             ◁          ( score )
                             [ child.length ]
                             ↓
                             ◇ ──▷ ( check[i].hit )
                                    ↓ [shot]
```
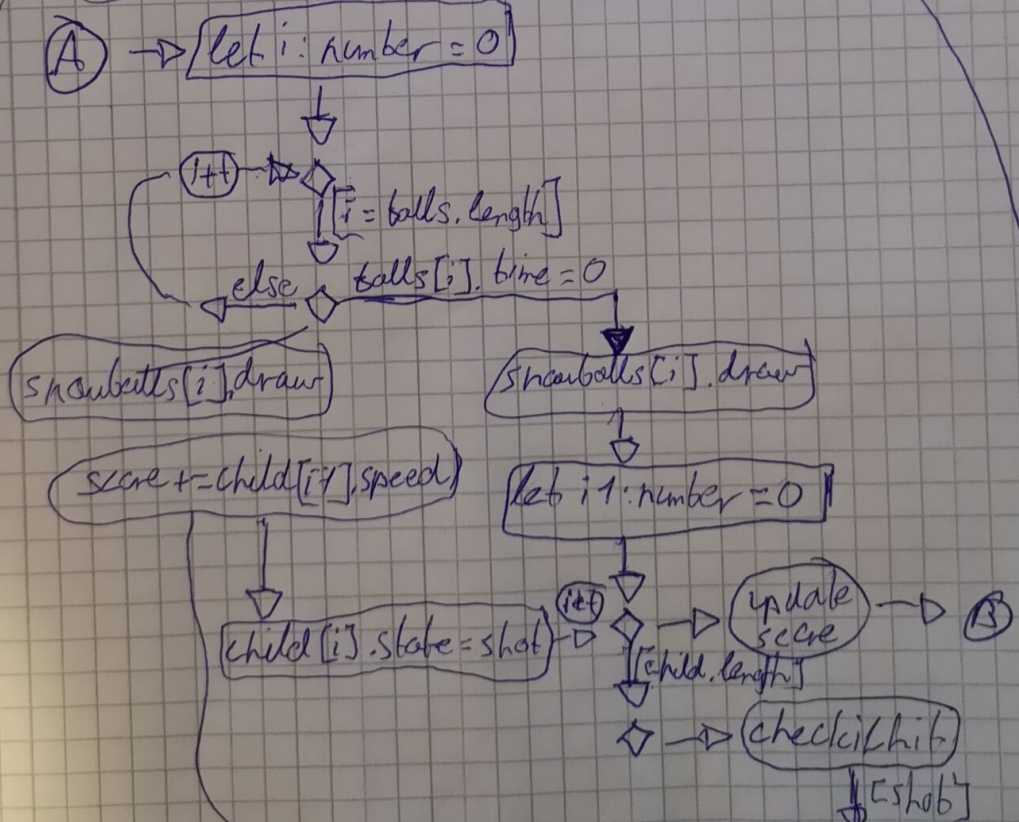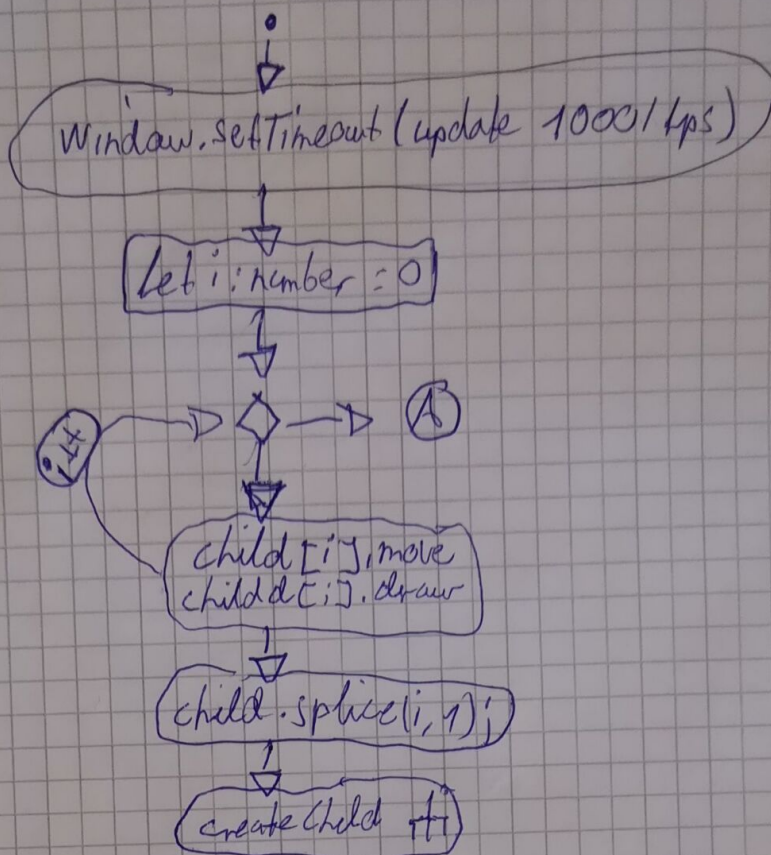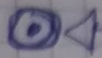
highscores / showManual

HTML Elemente werden angezeigt / verborgen

throwBall

event: Mouse Event

x : number
y : number

create new Ball
Snowball = new Snowball
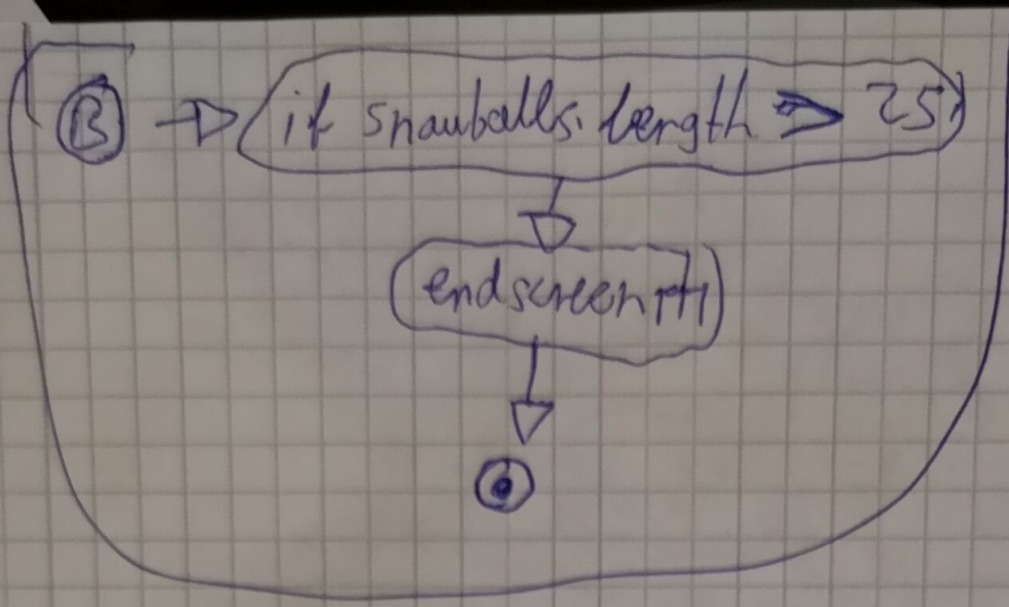
push new Ball in Array
Snow Snowballs.push

endscreen

HTML-Elemente werden angezeigt / verborgen

Send button wird angezeigt

highscore wird angezeigt

B → (if snawballs.length => 25)

⟶

(endscreen ())

⟶

◉

throw Corn

```
         ●
         ↓
    ┌──────────────────┐
    │ event: Keydown Event │
    └──────────────────┘
         ↓
    ┌──────────────────┐
    │ mouse_x : number │
    │ mouse_y : number │
    └──────────────────┘
         ↓
    ( create Canvas Corns )
         ↓
    ( generate hitbox radius )
         ↓
    ( check if Bird in Range )
         ↓
    ( decay after math. random frames )
         ↓
         ◉
```