

# CC2 TAMAGOTCHI PLANUNG

Florian Pflücker SoSe 2021

# IDEE

**Tamagotchi Prinzip** anhand eines kleinen Tieres, welches Emotionen besitzt und zu Beginn ausgebrütet werden muss.

Der Spieler muss sich um die Pflege, Durst, Hunger und weitere Bedürfnisse kümmern

# GROBE WIREFRAMES

SPIELNAME

*\*Hintergrundgrafik\*  
evtl. animiert*

Schwierigkeitsgrad wählen

LEICHT

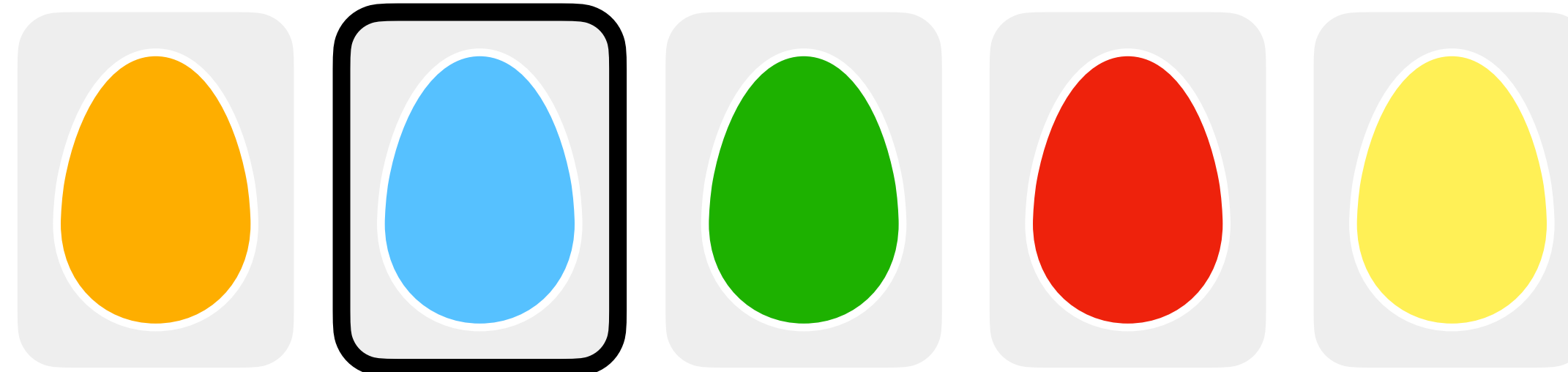
MITTEL

SCHWER



Namen geben

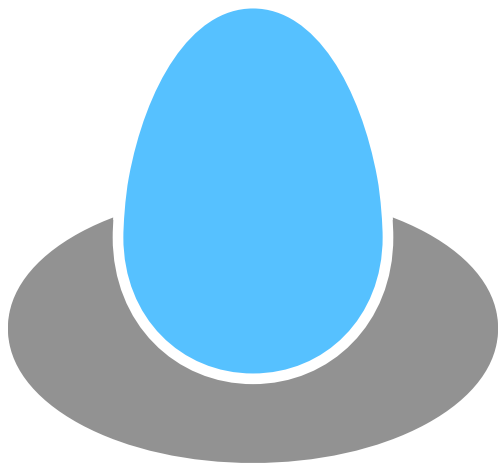
Farbe des Eis wählen



STARTEN

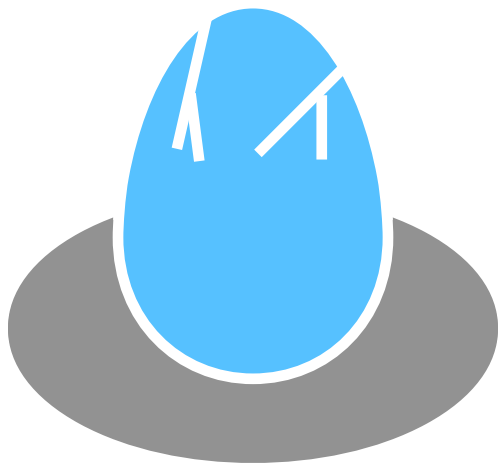
*\*Hintergrundgrafik\*  
evtl. animiert*

Mein Tier



wärmen

Mein Tier



progress

wärmen

Mein Tier

A



B



C



Handlungsmöglichkeiten die  
Parameter beeinflussen  
Evtl. „Drag n Drop“



A

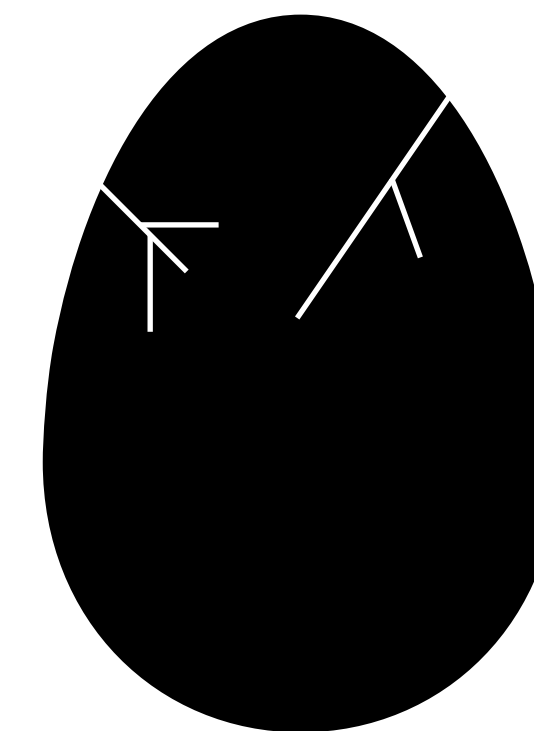
B

C



# „SCHLÜPFEN“

Durch Wärmen oder eine bestimmte „Aktion“, die ausgeführt werden kann, wird der Brutvorgang beschleunigt



Wärmen



# ENTWICKLUNGSSTADIEN & EMOTIONEN

## Eigene Klasse für Entwicklung

### Methoden

- > zeichnen je nach Fortschritt verschiedene Erscheinungsbilder
- > verschiedene Körper werden ohne Gesicht gezeichnet

## Eigene Klasse für Emotionen

### Methoden

- > glücklich
- > traurig
- > neutral



Verschiedene Gesichter werden je nach Zustand verschieden gezeichnet und über den aktuellen Körper drüber gezeichnet

# PARAMETER

Hunger

Durst

Emotionen/Aufmerksamkeit

Permanent abnehmende Parameter  
durch Füttern, Streicheln etc. werden die  
jeweiligen Parameter wieder „aufgefüllt“

## Eigene Klasse für Parameter

```
display.“jeweiligerParameter”();
```

-> soll jeweiligen Parameter Status zeichnen  
-> jeder Parameter ist hierdurch individuell  
ansprechbar

```
display.parameters();
```

-> alle Parameter werden hier mit einer  
Methode gezeichnet



NOTIZEN

## Parameter

- Fortschritt beim Schlüpfen
- Hunger
- Durst
- Spaß
- (Schlaf) → Zeitraffer, wird kurz abgedunkelt

class Parameters { }

- > hunger ()
  - > thirst ()
  - > fun
- display. "versch. Anzeigen",
  - calculate. "———";

• Bsp.: hunger () { }

variable = variable - 0,5; }

⇒ Variable nimmt ständig ab

- Durch Füttern wird bspw. der hunger() Parameter wieder um 100 Einheiten erhöht

class Button {

const(x, y, w, h, action) { }

}

Maße  
(wie bei Avatar)

↑  
was bewirkt der  
jeweilige Button?

Wann wird der nächste Entwicklungszustand gezeichnet?

= wenn das Tier n mal gefüttert wurde, dann wird es größer

↳ Variable, die speichert, wie oft gewisse Buttons gedrückt wurden

= if-Bedingung für die versch. Methoden, die das Tier zeichnen

Entwicklung

class Animal {

constructor() { }

baby() {

PNG einfügen if ( )

}

child() {