

DOKUMENTATION

REACT HOOKS

```
const [cards, setCards] = useState([])
```

Liste aller Karten, welche sich im Spiel befinden.

Wird initialisiert mit: leere Liste

Karten haben folgende Attribute:

- *source* (string): Pfad zur Bilddatei der jeweiligen Karte.
- *matched* (boolean): Gibt an, ob die Karte umgedreht als Paar vorliegt.
- *id* (float): ID der Karte, wird als Key für React verwendet.

```
const [choiceOne, setChoiceOne] = useState(null)
```

Speichert die Karte, welche als erste Wahl getroffen wird.

Wird initialisiert mit: null

```
const [choiceTwo, setChoiceTwo] = useState(null)
```

Speichert die Karte, welche als zweite Wahl getroffen wird.

Wird initialisiert mit: null

```
const [disabled, setDisabled] = useState(false)
```

Speichert, ob der Nutzer mit den Spielkarten interagieren kann.

Wird initialisiert mit: false

```
const [turns, setTurns] = useState(0)
```

Speichert die Anzahl der Züge.

Wird initialisiert mit: 0

FUNKTIONEN

```
const shuffleCards()
```

Mischt die Karten.

Es wird eine Liste, welches jedes Objekt aus [cardImages](#) zwei Mal enthält erstellt. Diese Liste wird anschließend gemischt und jeder Karte wird zusätzlich eine ID zugeordnet. Die beiden ReactHooks [choiceOne](#) und [choiceTwo](#) werden zurückgesetzt bzw. auf null gesetzt. Die erstellte Liste wird anschließend dem ReactHook [cards](#) zugewiesen und [turns](#) auf 0 gesetzt.

```
const restartGame()
```

Startet das Spiel neu.

Für jedes Objekt aus [cards](#) wird das Attribut [matched](#) auf false gesetzt. Die beiden ReactHooks [choiceOne](#) und [choiceTwo](#) werden zurückgesetzt bzw. auf null gesetzt. Am Ende wird [shuffleCards\(\)](#) mit einem Delay von 300 Millisekunden aufgerufen.

```
const handleChoice(card)
```

Entscheidet, ob die vom Nutzer gewählte Karte die erste oder zweite Wahl ist.

Ist das Attribut [id](#) von card gleich der id von [choiceOne](#), wird die Funktion abgebrochen, da sonst sowohl [choiceOne](#) als auch [choiceTwo](#) dieselbe Karte wäre und so beide Karten als „gleich“ bzw. als „Match“ angesehen werden würden. Danach wird die vom Nutzer gewählte Karte entweder [choiceOne](#) oder [choiceTwo](#) zugewiesen. Ist [choiceOne](#) null, so wurde noch keine erste Wahl getroffen und die gewählte Karte wird [choiceOne](#) zugewiesen. Ist [choiceOne](#) nicht null, wurde bereits eine erste Karte ausgewählt und die gewählte Karte wird [choiceTwo](#) zugewiesen.

```
const resetTurn()
```

Setzt den aktuellen Zug zurück und erhöht [turns](#) um 1.

Setzt [choiceOne](#) und [choiceTwo](#) zurück bzw. setzt deren Wert auf null und erhöht [turns](#) und damit die Anzahl der Züge um 1. [Disabled](#) wird auf false gesetzt.

USEEFFECT() FUNKTIONEN

```
useEffect(() => { [...] }, [ ])
```

Mischt die Karten automatisch beim Starten der App.

Die Hook-Liste ist leer, wodurch die Funktion direkt beim Starten der App aufgerufen wird und [shuffleCards\(\)](#) ausführt.

```
useEffect(() => { [...] }, [choiceOne, choiceTwo])
```

Wertet nach Auswahl zweier Karten diese aus.

Ist [choiceOne](#) oder [choiceTwo](#) null, wird abgebrochen, da noch keine zwei Karten gewählt wurden. Sind beide Karten gesetzt, wird fortgefahren und [disabled](#) wird auf true gesetzt, wodurch die Interaktion mit allen Karten blockiert wird. Als nächstes wird überprüft, ob das Attribut [source](#) von [choiceOne](#) gleich [source](#) von [choiceTwo](#) ist. Ist dies der Fall, wird [cards](#) mit einer neuen Liste aktualisiert, indem jede Karte der aktuellen Kartenliste, von welcher das Attribut [source](#) den gleichen Wert aufweist, wie [source](#) von [choiceOne](#) das Attribut [matched](#) auf true geändert bekommt. Dies führt dazu, dass die beiden Karten umgedreht bleiben, wenn das Spiel weitergeht. Zum Abschluss wird [resetTurn\(\)](#) aufgerufen. Ist das Attribut [source](#) von [choiceOne](#) jedoch nicht gleich [source](#) von [choiceTwo](#), wird [resetTurn\(\)](#) mit einem Delay von einer Sekunde aufgerufen, wodurch sich die beiden vom Nutzer gewählten Karten nach einer Sekunde wieder umdrehen.

SONSTIGE VARIABLEN

cardImages

Typ: Liste

Speichert Objekte mit dem Attribut [source](#) sowie [matched](#) und wird zum Erstellen der Kartenliste [cards](#) verwendet.

COMPONENTS

```
function Card({ card, handleChoice, flipped, disabled })
```

Übergabeparameter:

- *card* (objekt): Einzelne Karte dessen Attribute zum Data Binding verwendet werden.
- *handleChoice* (funktion): Wird aufgerufen, wenn eine Karte geklickt wird.
- *flipped* (boolean): Bestimmt, ob eine Karte visuell umgedreht werden soll.
- *Disbaled* (disabled): Bestimmt, ob die Nutzer-Interaktion ausgewertet werden soll, wenn eine Karte geklickt wird.