




Einführung in Rekursion und Algorithmik

Developer Akademie - Kundenevent



Herzlich willkommen!

Agenda

- Projektvorstellung
 - Installation (VS Code, GitHub, Live-Share)
 - Programmierung Teil 1 – User Interface (Gruppenarbeit)
 - Mittagspause (12:00 – 13:00)
 - Algorithmen und Rekursion
 - Programmierung Teil 2 – Spiel fertigstellen (Gruppenarbeit)
 - Q /A
- 
- The bottom of the slide features a decorative graphic consisting of several overlapping, organic shapes in shades of red, orange, and cream, creating a modern, flowing design.

Projektvorstellung



Projektvorstellung



- Verdeckte Felder
- Bombe = Verloren
- Zahl = Anzahl der Bomben in angrenzenden Feldern

Programmierung Teil 1

- Was brauchen wir?



Gruppeneinteilung



Projekteinrichtung

Projektordner, GitHub, Live-Share

Projekteinrichtung

JS script.js

JS script.js > ...

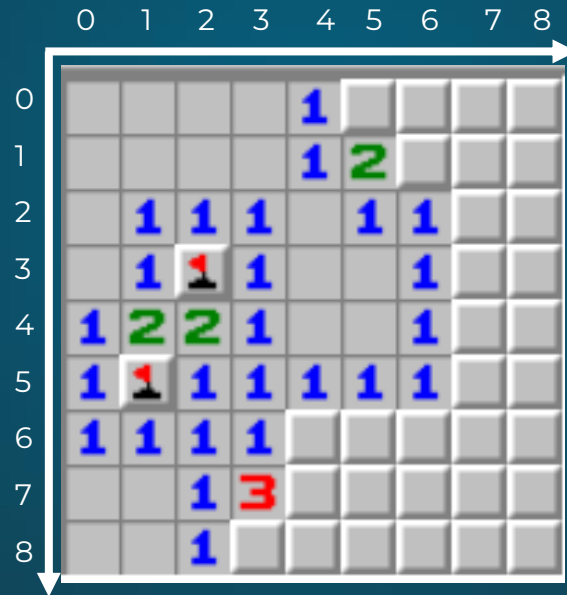
```
1  let fields = [];  
2  
3  function init() {  
4      fillFields(10, 10);  
5      render();  
6  }  
7  
8  function render() {  
9  
10 }  
11  
12
```

Projekteinrichtung

Grundaufbau

```
JS script.js •
JS script.js > ...
1  let fields = [];
2
3  function init() {
4      fillFields(10, 10);
5      render();
6  }
7
8  function render() {
9
10 }
11
12
```

Koordinaten



Projekteinrichtung

Grundaufbau

```
JS script.js •
JS script.js > ...
1  let fields = [];
2
3  function init() {
4    fillFields(10, 10);
5    render();
6  }
7
8  function render() {
9
10 }
11
12
```

Koordinaten



```
let fields = [ // Beispiel 3x3
  { x: 0, y: 0, number: 0, hasBomb: false },
  { x: 1, y: 0, number: 0, hasBomb: false },
  { x: 2, y: 0, number: 0, hasBomb: false },
  { x: 0, y: 1, number: 0, hasBomb: false },
  { x: 1, y: 1, number: 0, hasBomb: false },
  { x: 2, y: 1, number: 0, hasBomb: false },
  { x: 0, y: 2, number: 0, hasBomb: false },
  { x: 1, y: 2, number: 0, hasBomb: false },
  { x: 2, y: 2, number: 0, hasBomb: false },
];
```

render()

```
<div>
  <div onclick="selectField(0,0)"></div>
  <div onclick="selectField(1,0)"></div>
  <div onclick="selectField(2,0)"></div>
</div>
<div>
  <div onclick="selectField(0,1)"></div>
  <div onclick="selectField(1,1)"></div>
  <div onclick="selectField(2,1)"></div>
</div>
<div>
  <div onclick="selectField(0,2)"></div>
  <div onclick="selectField(1,2)"></div>
  <div onclick="selectField(2,2)"></div>
</div>
```



Gruppenarbeit

30 – 45 Min

Bomben platzieren

```
let fields = [ // Beispiel 3x3
  { x: 0, y: 0, number: 0, hasBomb: false },
  { x: 1, y: 0, number: 0, hasBomb: false },
  { x: 2, y: 0, number: 0, hasBomb: false },
  { x: 0, y: 1, number: 0, hasBomb: false },
  { x: 1, y: 1, number: 0, hasBomb: false },
  { x: 2, y: 1, number: 0, hasBomb: false },
  { x: 0, y: 2, number: 0, hasBomb: false },
  { x: 1, y: 2, number: 0, hasBomb: false },
  { x: 2, y: 2, number: 0, hasBomb: false },
];
```

Bomben platzieren

```
let fields = [ // Beispiel 3x3
  { x: 0, y: 0, number: 0, hasBomb: false },
  { x: 1, y: 0, number: 0, hasBomb: false },
  { x: 2, y: 0, number: 0, hasBomb: false },
  { x: 0, y: 1, number: 0, hasBomb: false },
  { x: 1, y: 1, number: 0, hasBomb: false },
  { x: 2, y: 1, number: 0, hasBomb: false },
  { x: 0, y: 2, number: 0, hasBomb: false },
  { x: 1, y: 2, number: 0, hasBomb: false },
  { x: 2, y: 2, number: 0, hasBomb: false },
];
```

- 3 Bomben
- Zufällig 3 Elemente aus dem Array
- Alle umliegenden Felder finden
- field.number++;

Algorithmen und Rekursion

```
<div>
  <div onclick="selectField(0,0)"></div>
  <div onclick="selectField(1,0)"></div>
  <div onclick="selectField(2,0)"></div>
</div>
<div>
  <div onclick="selectField(0,1)"></div>
  <div onclick="selectField(1,1)"></div>
  <div onclick="selectField(2,1)"></div>
</div>
<div>
  <div onclick="selectField(0,2)"></div>
  <div onclick="selectField(1,2)"></div>
  <div onclick="selectField(2,2)"></div>
</div>
```

```
let fields = [ // Beispiel 3x3
  { x: 0, y: 0, number: 0, hasBomb: false },
  { x: 1, y: 0, number: 0, hasBomb: false },
  { x: 2, y: 0, number: 0, hasBomb: false },
  { x: 0, y: 1, number: 0, hasBomb: false },
  { x: 1, y: 1, number: 0, hasBomb: false },
  { x: 2, y: 1, number: 0, hasBomb: false },
  { x: 0, y: 2, number: 0, hasBomb: false },
  { x: 1, y: 2, number: 0, hasBomb: false },
  { x: 2, y: 2, number: 0, hasBomb: false },
];
```

Algorithmen und Rekursion

```
let field = fields.find(f => f.x == 1 && f.y == 1);
```

```
<div>
  <div onclick="selectField(0,0)"></div>
  <div onclick="selectField(1,0)"></div>
  <div onclick="selectField(2,0)"></div>
</div>
<div>
  <div onclick="selectField(0,1)"></div>
  <div onclick="selectField(1,1)"></div>
  <div onclick="selectField(2,1)"></div>
</div>
<div>
  <div onclick="selectField(0,2)"></div>
  <div onclick="selectField(1,2)"></div>
  <div onclick="selectField(2,2)"></div>
</div>
```

```
let fields = [ // Beispiel 3x3
  { x: 0, y: 0, number: 0, hasBomb: false },
  { x: 1, y: 0, number: 0, hasBomb: false },
  { x: 2, y: 0, number: 0, hasBomb: false },
  { x: 0, y: 1, number: 0, hasBomb: false },
  { x: 1, y: 1, number: 0, hasBomb: false },
  { x: 2, y: 1, number: 0, hasBomb: false },
  { x: 0, y: 2, number: 0, hasBomb: false },
  { x: 1, y: 2, number: 0, hasBomb: false },
  { x: 2, y: 2, number: 0, hasBomb: false },
];
```


Algorithmen und Rekursion

```
function selectField(x, y){  
  // 1. Feld finden  
  if (field.number == 0 && !field.selected) {  
    // Alle Felder drum herum aufdecken  
    selectField(x - 1, y-1); // Links oben  
    selectField(x, y-1); // oben  
    selectField(x + 1, y-1); // Rechts oben  
    // Links  
    // Rechts  
    // Links unten  
    // Mitte unten  
    // Rechts unten  
  }  
  // 3. field.selected = true; // Wird aufgedeckt  
  // 4. render  
}
```

Algorithmen und Rekursion

```
function selectField(x, y){  
  // 1. Feld finden  
  if (field.number == 0 && !field.selected) {  
    // Alle Felder drum herum aufdecken  
    selectField(x - 1, y-1); // Links oben  
    selectField(x, y-1); // oben  
    selectField(x + 1, y-1); // Rechts oben  
    // Links  
    // Rechts  
    // Links unten  
    // Mitte unten  
    // Rechts unten  
  }  
  // 3. field.selected = true; // Wird aufgedeckt  
  // 4. render  
}
```

Algorithmen und Rekursion

Kein doppeltes
ausführen

```
function selectField(x, y){  
  // 1. Feld finden  
  if (field.number == 0 && !field.selected) {  
    // Alle Felder drum herum aufdecken  
    selectField(x - 1, y-1); // Links oben  
    selectField(x, y-1); // oben  
    selectField(x + 1, y-1); // Rechts oben  
    // Links  
    // Rechts  
    // Links unten  
    // Mitte unten  
    // Rechts unten  
  }  
  // 3. field.selected = true; // Wird aufgedeckt  
  // 4. render  
}
```

Algorithmen und Rekursion



Algorithmen und Rekursion



Algorithmen und Rekursion

