

Junioraufgabe 2: Baulwürfe

Vorwort

Alle Methoden sind mit HTML und Javascript implementiert.

Das Ausführbare Programm kann man bei folgende Adresse finden:

https://tnthunter22.github.io/Bwinf_2020/Aufgabe%202/Aufgabe_2.HTML

Lösungsidee

Meine Lösungsidee war ein array in einem array zu machen und dann mit einem 'for' schleife immer Jede Zeile auf ein 'X' zu überprüfen. Dann überprüfe ich ob das 'X' und seine Nachbarn mit dem Muster identisch sind, wenn nicht andere ich das 'X' am anfang zu einem leeren platz damit es bei weiteren checken nicht stört. Wenn aber die beiden Muster identisch sind dann addiere ich zu dem anzahl an mustern 1 und ändere alle 'X' in dem muster zu leeren Plätzen damit das system sie nicht nochmal checkt und damit zeit verliert.

Umsetzung

Schritt 1: Den auf die Seite geladenen File speichern

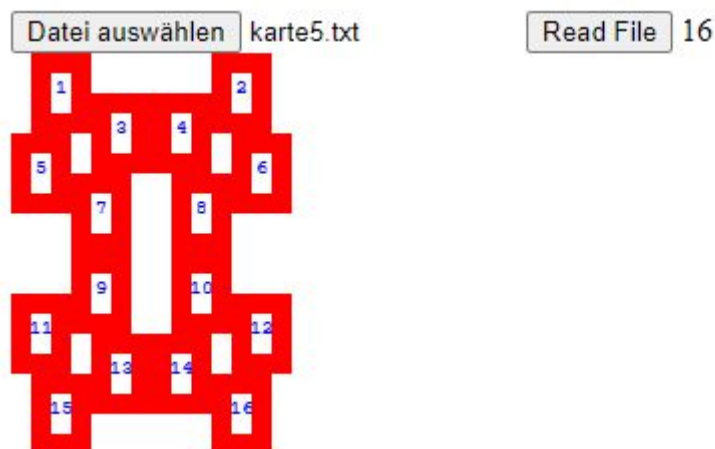
Schritt 2: Den File auf 'X' überprüfen

Schritt 3: Überprüfen ob ein Muster vorliegt

Schritt 4: Die Anzahl an Mustern rechnen und vorlegen

Beispiele

1:



2:



Quellcode

//Button

```
document.querySelector("#read-button").addEventListener('click', function() {
    //filescheck
    if(document.querySelector("#file-input").files.length == 0) {
        alert("Error : No file selected");
        return;
    }
    //number of Patterns setup
    var count = 0;
    //clearcanvas
    ctx.fillStyle = "White"
    ctx.fillRect(0,0,width,height)
    //filereader
    var file = document.querySelector("#file-input").files[0];
    var reader = new FileReader();
    reader.addEventListener('load', function(e) {
        input = [];
        var text = e.target.result;
        var lines = this.result.split("\n")
        var columncount = lines[0];
        var linecount = lines[1];
        for (var line = 0;line < linecount;line++){
            input[line] = [];
            for (column = 0;column < columncount ;column++){
                input[line][column] = lines[line + 2].charAt(column);
                var patternBlocks = new Block(line,column)
                if(input[line][column] === "X"){
                    patternBlocks.draw("Black");
                }
            }
        }
        $("#numberOfPatterns").text(" " + findx());
    });
    reader.readAsText(file);
```

});

//find all the X

```
var findx = function (){
    for (var k = 0;k < input.length - 3;k++){
        for (var l = 0;l < input[k].length - 2;l++){
            if(input[k][l] === "X"){
                if (checker(k,l)){
                    count++;
                    drawPattern(k,l)
                    clear(k,l)
                }
            }
        }
    }
    return count;
}
//check if the pattern is right
var Patternchecker = function(xPos,yPos) {
    for (var i = 0;i < searchedpattern.length;i++){
        for (var j = 0;j < searchedpattern[i].length;j++){
            if (input[xPos + i][yPos + j] != searchedpattern[i][j]){
                return false;
            }
        }
    }
    return true;
}
```

