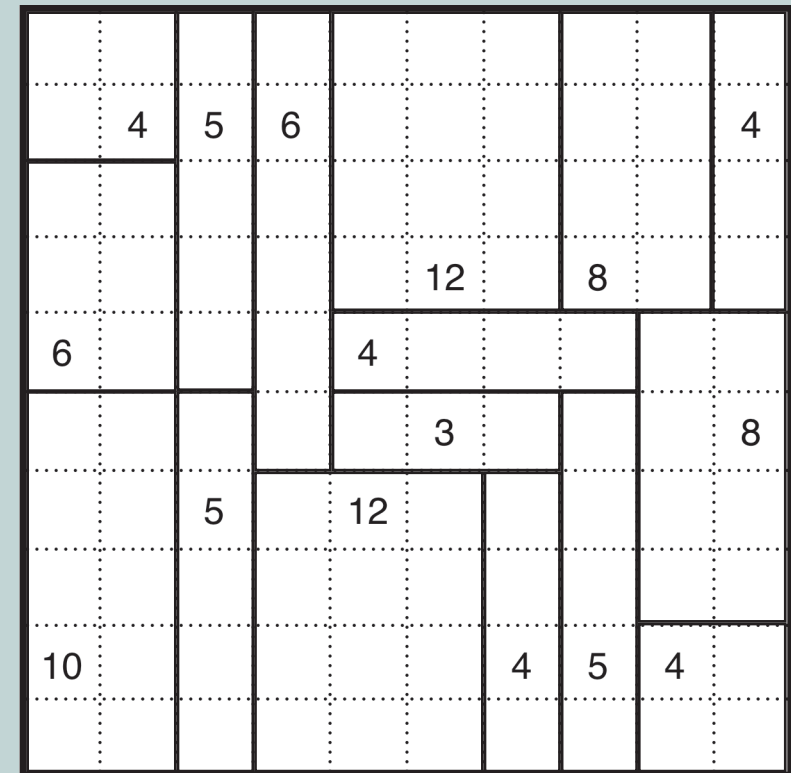
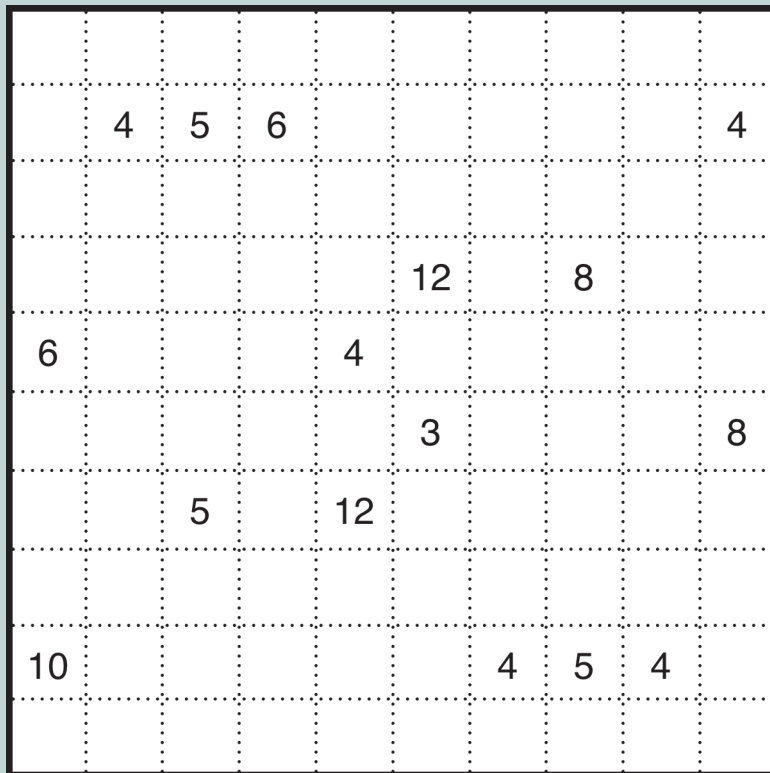


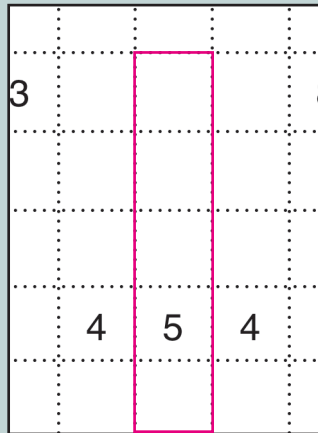
Ziel

- Die Belegung in Rechtecke zu zerlegen.
- In jedem Rechteck steht genau eine Zahl.
- Diese Zahl gibt die Anzahl der Felder an, aus denen das Rechteck besteht

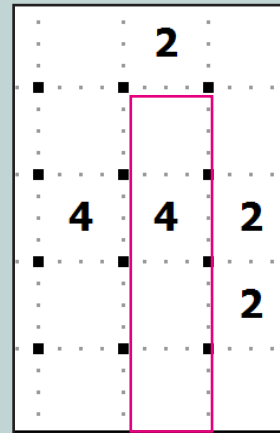


Lösungsstrategien

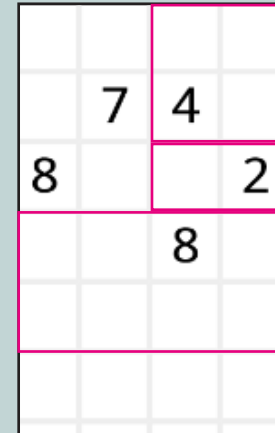
Die Rechtecke
mit Primzahlen
rausfinden



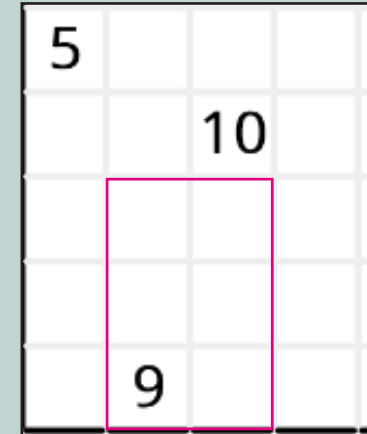
Die Rechtecke
deren Position
eindeutig ist



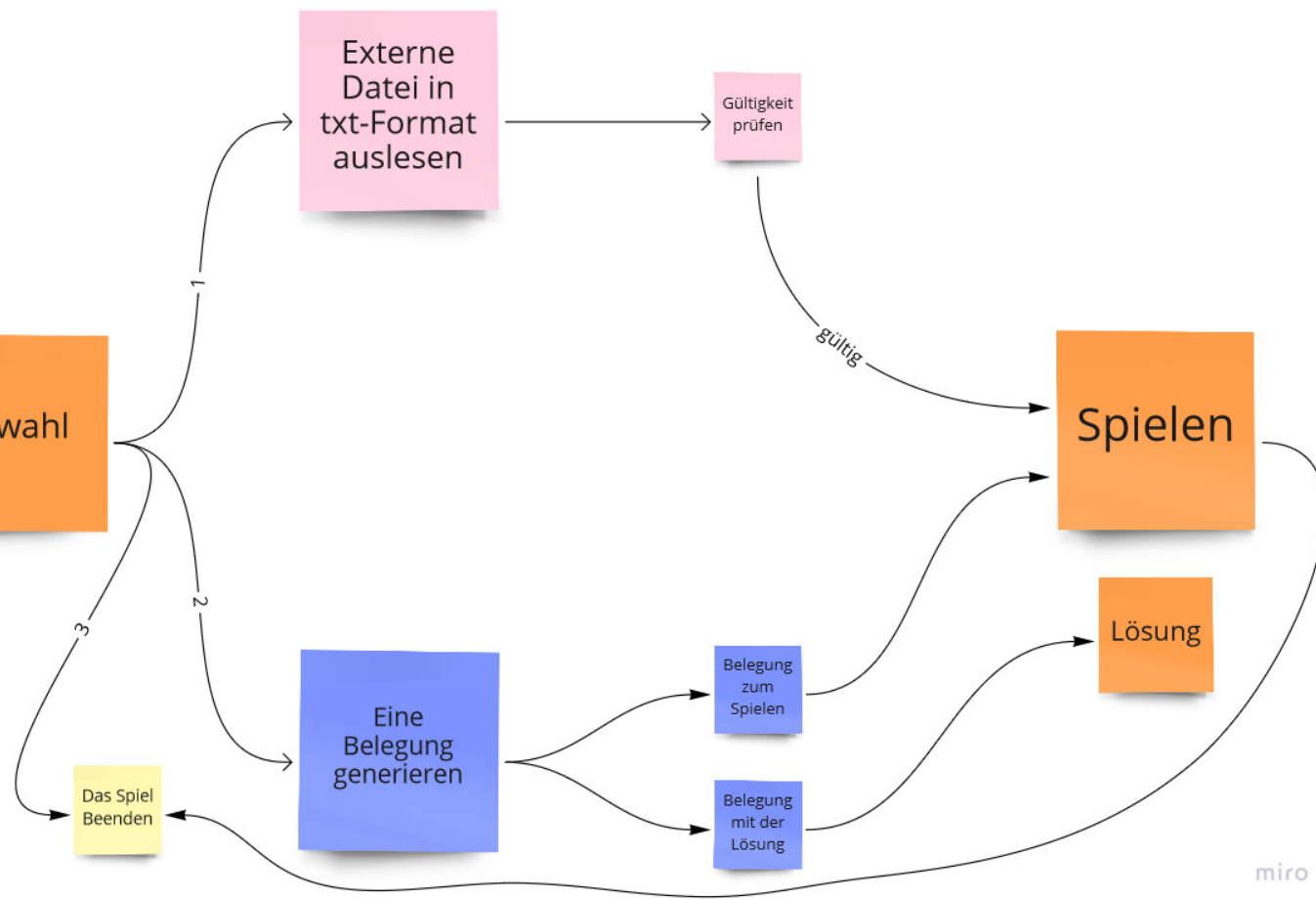
Ecken und Stellen
mit vielen Zahlen
nebeneinander
durchsuchen



Bei mehrere
Möglichkeiten eine
Mindestbelegung
bestimmen



C++ Programm



Benutzt werden zwei Klassen:

```

class Field
{
public:
    int size;
    vector<vector<string>> field;
    vector <string> row;
    string freeCell = " ";
    etc.
}

// Konstruktor
Field(int customSize)
{
    size = customSize;
    for(int i = 0; i < size; i++)
    {
        for(int j = 0; j < size; j++)
        {
            row.push_back(freeCell);
        }
        field.push_back(row);
    }
}

class Rectangle
{
friend class Field;
private:
    int width = 1;
    int height = 1;
    int area = 1;
    int x;           // x-Koordinate
    int y;           // y-Koordinate
    char sign;
// Konstruktor 2
Rectangle(int rectArea, char ch)
{...}

// Konstruktor 3
Rectangle(int w, int h, char ch)
{...}
    
```

Externe Datei

Belegung_01.txt

```

1 10|
2 2|-|7|-|-|-|-|-|5|
3 -|-|-|-|-|12|-|-|-|-|
4 -|-|-|-|-|2|-|4|-|
5 -|-|-|-|7|-|-|-|2|-|
6 2|-|3|-|4|-|-|-|-|-|
7 -|-|-|-|-|8|-|2|2|
8 -|-|-|-|8|-|-|-|-|-|
9 -|-|9|-|-|-|-|-|-|-|
10 2|-|2|3|-|-|-|-|-|-|
11 -|-|4|-|-|2|8|-|-|-|-|

```

Falls die Datei gültig ist, eine Belegung generieren, sonst das Spiel beenden

```

//Prüfen ob die gegebene Datei richtig sind
if(partitionFile.size() == pow(size, 2))
{
    playField.divideByFile(partitionFile);
    playField.print();
    play(playField, select);
}
else
{
    cout << "Die Datei ist falsch definiert!";
    out << partitionFile.size();
    exit(0);
}
break;

```

txt öffnen, auslesen und im Vektor speichern

```

// die Anordnung von txt-datei auslesen
bool readFile(string fileName, vector<string> &partitionFile)
{
    ifstream newFile;
    string str;

    newFile.open(fileName.c_str());
    if(!newFile)
    {
        cerr << fileName << " - Datei nicht gefunden";
        exit(-1);
    }
    while(getline(newFile, str, '|'))
    {
        partitionFile.push_back(str);
    }
    newFile.close();
    return true;
}

```

```

~/Project
Die Belegung -----
x 2 |-| 7 |-|-|-|-|-| 5 |x
x -|-|-|-|-|12|-|-|-|-|x
x -|-|-|-|-| 2 |-| 4 |-|x
x -|-|-|-| 7 |-|-|-| 2 |-|x
x 2 |-| 3 |-| 4 |-|-|-|-|-|x
x -|-|-|-|-| 8 |-| 2 | 2 |x
x -|-|-|-| 8 |-|-|-|-|-|x
x -|-| 9 |-|-|-|-|-|-|-|x
x 2 |-| 2 | 3 |-|-|-|-|-|-|x
x -|-| 4 |-|-| 2 | 8 |-|-|-|-|x
x -----
Optionen:
wasd - Cursor bewegen
Leerzeichen - Grenze loeschen
r - alle Grenze zurücksetzen
z - Undo
q - Aktuelle Grenze zurücksetzen
l - Lösung zeigen
k - Lösung ausblenden
g - Neu generieren
x - Beenden

```

Belegung generieren

```
// Generiere eine Belegung
```

```
Field generateNewField(map<string, Rectangle> &partition, int size)
{
    Inizielisiere ein leeres Feld size x size
    Inizielisiere ein leeres Rechteck mit dem "default" konstruktor

    while(true)
    {
        Suche auf die erste freie Zelle
        Falls keine freie Zelle mehr gibt, breche ab.
        Sonst versuche ein Rechteck zu platzieren:
            Generiere zufällige Fläche und Größe -> platziere
            Falls nicht gelungen, drehe und versuche noch mal
            Falls erfolgreich platziert, speichere das im Wörterbuch
            Wiederhole alles von vorne
    } // Als Ergebniss -> Lösungsbelegung und die Menge der Rechtecke

    Inizielisiere ein neues Feld size x size
    Ersetze die Buchstaben mit den Zahle/Flaechen
    Gehe durch jedes Rechteck, lesen die XY-Koordinaten und die Fläche aus
    und generiere entsprechende Postiion für eine Zahl
    Als Ergebniss -> eine Belegung zum Spielen
}
```

```
// Übergebe ein leeres Wörterbuch (map) und die Größe des Feldes
```

```
// Field customField(size);  
// Rectangle rect = Rectangle();
```

```
// tie(x, y) = customField.findFirstFreeCell();  
// if(x == -1) break;
```

```
// area = rand()%11+2;
// rect = Rectangle(area, albet[i]);
// partition[key] = rect;
```

```
// Field numbersField(size);
// Field replaceWithNumbers(map<string, Rectangle> partition, int size)
```

```
// numbersField.field[yPos][xPos] = to_string(rect.getArea());
```

```

irisi@DESKTOP-M6IPS3D ~/Project
$ ./Main.exe

Die Belegung

```

A	B	B	B	B	B	B	B	B	B
A	C	D	D	E	F	F	G	G	H
A	C	D	D	E	F	F	G	G	H
A	C	I	I	E	F	F	G	G	H
A	C	J	J	E	F	F	G	G	K
A	C	J	J	E	F	F	L	M	K
A	N	N	O	O	P	P	L	M	K
A	N	N	O	O	P	P	L	M	Q
A	R	S	O	O	P	P	L	M	Q
A	R	S	O	O	P	P	L	T	T

	A	R	S	O	O	P	P	L	T	T
20										
x: 0		y: 6	Area: 10						Sign: A	
x: 9		y: 0	Area: 9						Sign: B	
x: 1		y: 2	Area: 5						Sign: C	
x: 2		y: 2	Area: 4						Sign: D	
x: 4		y: 1	Area: 5						Sign: E	
x: 6		y: 2	Area: 10						Sign: F	
x: 8		y: 4	Area: 8						Sign: G	
x: 9		y: 2	Area: 3						Sign: H	
x: 3		y: 3	Area: 2						Sign: I	
x: 2		y: 4	Area: 4						Sign: J	
x: 9		y: 4	Area: 3						Sign: K	
x: 7		y: 9	Area: 5						Sign: L	
x: 8		y: 6	Area: 4						Sign: M	
x: 1		y: 6	Area: 4						Sign: N	
x: 3		y: 8	Area: 8						Sign: O	
x: 5		y: 6	Area: 8						Sign: P	
x: 9		y: 7	Area: 2						Sign: Q	
x: 1		y: 8	Area: 2						Sign: R	
x: 2		y: 9	Area: 2						Sign: S	
x: 8		y: 9	Area: 2						Sign: T	

Die Summe: 100
Die Belegung

```

Die Summe: 100
Die Belegung

```

									9
				5					
	5	4				10			3
			2						
		4						8	3
10	4				8			4	
									2
	2		8						
		2					5	2	

```

irisi@DESKTOP-M6IPS3D ~/Project
$

```

Das Spiel

```

Irisi@DESKTOP-M6IPS3D ~/Project
$ ./Main.exe

----- SHIKAKU -----

Wählen Sie eine Funktion aus:
1 - Externe Datei auslesen
2 - Die Anordnung generieren
3 - Das Spiel beenden

```

```

~/Project
----- Die Belegung -----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
x  |  |  |  |  | 5 |  |  |  | 3 |x
x .....
x  |  |  | 8 |  | 4 |  |  |  |  |x
x .....
x  |  | 4 |  |  | 8 |  |  |  |  |x
x .....
x 3 |  |  |  |  |  |  |  |  |  |x
x .....
x  |  | 2 |  | 2 |  |  |  | 4 |x
x .....
x  | 2 |  |  | 2 | 3 |  |  | 6 |x
x .....
x  |  |  | 5 |  | 8 |  |  |  |  |x
x .....
x  | 8 |  | 4 |  |  |  | 3 |  |x
x .....
x  |  |  |  |  |  |  |  |  |  |x
x .....
x  |  | 5 |  | 2 | 4 |  | 2 | 3 |x
x .....
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Optionen:
wasd - Cursor bewegen
Leerzeichen - Grenze loeschen
r - alle Grenze zurücksetzen
z - Undo
q - Aktuelle Grenze zurücksetzen
l - Lösung zeigen
k - Lösung ausblenden
g - Neu generieren
x - Beenden

```

```

~/Project
----- Die Lösung -----
----- Die Belegung -----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
x  A | A | A | A | B | B | B | B | C |x
x .....
x  A$| A | A | A | D | D | E | E | F | C |x
x .....
x  G | H | H | I | D | D | E | E | F | C |x
x .....
x  G | H | H | I | J | K | E | E | F | L |x
x .....
x  G | M | M | I | J | K | E | E | F | L |x
x .....
x  N | N | O | I | P | K | Q | Q | F | L |x
x .....
x  R | R | O | I | P | S | Q | Q | F | L |x
x .....
x  R | R | O | T | T | S | Q | Q | U | V |x
x .....
x  R | R | O | T | T | S | Q | Q | U | V |x
x .....
x  R | R | O | W | W | S | X | X | U | V |x
x .....
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Optionen:
wasd - Cursor bewegen
Leerzeichen - Grenze loeschen
r - alle Grenze zurücksetzen
z - Undo
q - Aktuelle Grenze zurücksetzen
l - Lösung zeigen
k - Lösung ausblenden
g - Neu generieren
x - Beenden

```

```

~/Project
----- Die Belegung -----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
x  5 |  |  |  |  |  |  |  |  |x
x .....
x  |  | 4 |  |  | 4 |  |  | 6 |x
x .....
x  | 5 |  |  |  | 5 |  |  | 2 |x
x .....
x 5 |  | 6 | 7 |  |  |  |  |x
x .....
x  |  |  |  |  |  | 10 | 6 |x
x .....
x  |  |  |  | 7 |  |  |  |x
x .....
x 3 |  |  |  |  |  |  |  |x
x .....
x  |  | 2 |  |  | 4 |  |  |x
x .....
x 3 |  | 2 |  |  |  |  |  |x
x .....
x  | 6 | 3 | 2 | 3 |  |  |x
x .....
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Optionen:
wasd - Cursor bewegen
Leerzeichen - Grenze loeschen
r - alle Grenze zurücksetzen
z - Undo
q - Aktuelle Grenze zurücksetzen
l - Lösung zeigen
k - Lösung ausblenden
g - Neu generieren
x - Beenden

```