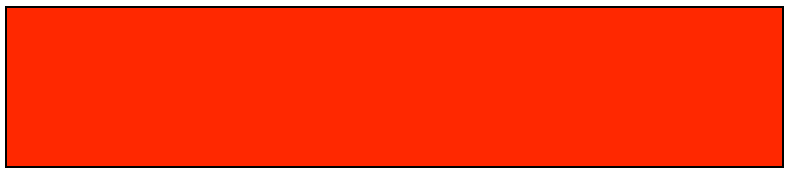
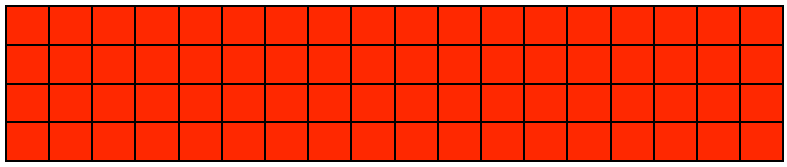
Die Idee einer zufälligen Labyrinthprogrammierung

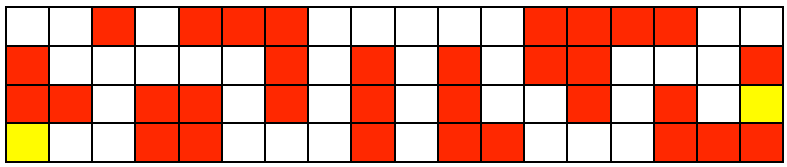
Zu Beginn wird ein beliebig oder definiert großes Rechteck erstellt. Was für uns so aussieht,



besteht für den Computer aus unzähligen kleinen Rechtecken, etwa so:



Doch wie macht man daraus ein Labyrinth? Nun, indem man dem Computer sagt, er soll Gänge „graben“. Er soll also die Mauern zwischen den einzelnen Zellen einreißen, wodurch begehbare Gänge entstehen. Der Weg wird durch weiße Kästchen dargestellt, etwa so: (Von Gelb zu Gelb)



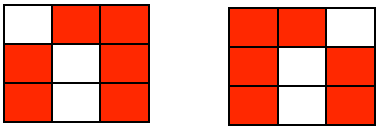
Das ist eine von unzähligen Möglichkeiten, dieses Rechteck mit einer Art Labyrinth zu versehen. Dem Computer soll jetzt gesagt werden, dass er aus allen Möglichkeiten zufällig eine auswählt und zeichnet.
Ein Algorithmus sagt dem Computer wo es lang geht:
Nach jedem von ihm zufällig im Rechteck gewählten Kästchen soll der Computer ein angrenzendes Kästchen wählen, weiß einfärben und zu diesem Kästchen springen. Vo

dort aus geht es weiter. Damit kein Rundgang oder eine Abkürzung entsteht, werden dem Algorithmus noch Vorschriften gestellt.
In C++ könne diese etwa so aussehen:

```
if (GetGrid (r, c-1) == PATH)
{
        if (GetGrid (r-1, c+1) == PATH || GetGrif (r+1, c+1) == PATH)
        {
                return (false);
        }
}
```

Was diese sogenannte if-Bedingung bewirkt?

Kurz gesagt, er verhindert diese Ereignisse beim Tunnelgraben:



Denn man könnte ja annehmen, dass bei Kästchen, die sich diagonal überschneiden, ein Durchgang entstehen könnte, den man verhindern möchte.