KAPITEL 27

Minesweeper

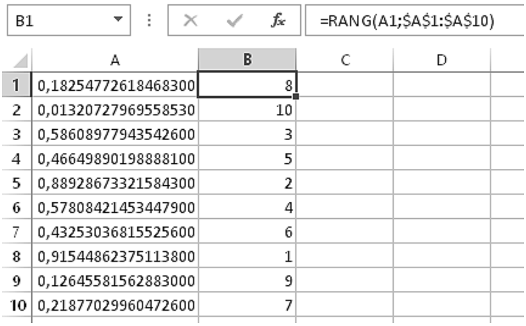
Wer mit Microsoft aufgewachsen ist, kennt auch die beiden Spiele Minesweeper und Solitaire, die früher auf keinem unter Windows laufenden PC gefehlt haben. In den Jahren, in denen es noch ein seltener Luxus war, im Internet surfen zu können, waren diese Klassiker vor allem für Azubis oft überlebenswichtig, um den tristen Büroalltag zu überstehen. Im Laufe der Zeit haben aber immer mehr hartherzige Chefs die beliebten Langeweilevertreiber von den PCs verbannen lassen und ab Windows 8 sind die Spiele gar nicht mehr standardmäßig an Bord. Echt fies! Als Excel-Fan lassen Sie sich das natürlich nicht gefallen. Machen Sie Excel zum größten Minesweeper-Feld aller Zeiten!

Das Spiel ist schnell erklärt. In einer Matrix sind zufällig Minen verteilt und versteckt und die Aufgabe besteht darin, alle Zellen der Matrix aufzudecken, bis auf die Zellen, die eine Mine enthalten. Das Spiel ist gewonnen, wenn nur noch die Zellen übrig sind, die Minen enthalten. Zur Ortung der Minen enthält jede aufgedeckte Zelle einen Wert, der die Anzahl Minen wiedergibt, die in den acht Nachbarzellen der aufgedeckten Zelle versteckt sind.

Zur Demonstration fangen wir wieder klein an, mit einem 10 x 10 Zellen großen Minenfeld. Wem das zu brutal ist, kann sich statt Minen auch Fettnäpfchen oder Kuhfladen vorstellen.

27.1 Zufallszahlen ohne Wiederholung

Die erste Hürde, die genommen werden muss, besteht darin, eine bestimmte Anzahl Minen zufällig auf dem Feld zu verteilen. Immer wenn in Excel etwas zufällig geschehen soll, kommt die Funktion ZUFALLSZAHL() zum Zuge. Wir benötigen 100 Zufallszahlen, genau so viele, wie Zellen auf dem Spielfeld enthalten sind, die ja alle mit gleicher Wahrscheinlichkeit eine Mine enthalten können. In der Kombinatorik wird dieser Fall als »Ziehen ohne Zurücklegen« (oder »ungeordnete Kombinationen ohne Wiederholung«) bezeichnet. Jede der 100 Zellen hat eine Nummer von 1 bis 100 und nun werden n zufällige Zellen »gezogen«, die eine Mine enthalten. Ganzzahlige Zufallszahlen ohne doppelt vorkommende (»ohne Zurücklegen«) können mithilfe der Funktion RANG erzeugt werden, was wir uns an einem Beispiel mit zehn Zahlen ansehen (*Abb. 27.1*):



B1 | =RANG(A1;$A$1:$A$10)

	A	B	C	D
1	0,18254772618468300	8		
2	0,01320727969558530	10		
3	0,58608977943542600	3		
4	0,46649890198888100	5		
5	0,88928673321584300	2		
6	0,57808421453447900	4		
7	0,43253036815525600	6		
8	0,91544862375113800	1		
9	0,12645581562883000	9		
10	0,21877029960472600	7		

Abbildung 27.1: Ganzzahlige Zufallszahlen ohne Wiederholung

`=RANG(A1;$A$1:$A$10)` ergibt 8, da in A1 die achthöchste Zufallszahl der Spalte A steht. Die 10 erscheint dort, wo die kleinste Zufallszahl steht, nämlich die 0,01320727969558530 in A2. Vorraussetzung dafür, dass alle Zahlen von 1 bis 10 in B1:B10 genau einmal vorkommen, ist, dass alle Zufallszahlen unterschiedlich groß sind.

Zufallszahlen liegen zwischen 0 und 1 und haben in der Regel 15 Dezimalstellen. Nur in der Regel – aber nicht immer – und zwar deshalb, weil ca. 10 % der Zufallszahlen 16 Nachkommastellen haben und ca. 1 % sogar 17 Nachkommastellen haben. Das hängt damit zusammen, dass es nicht genau 10 hoch 15 unterschiedliche Zufallszahlen gibt, sondern 2 hoch 52, das sind 10 hoch 15,653559774527 oder anders ausgedrückt 4.503.599.627.370.500 verschiedene Zufallszahlen. Dass dann zwei gezogene Zufallszahlen identisch sind, ist zwar theoretisch möglich, praktisch aber so gut wie ausgeschlossen. Die Wahrscheinlichkeit, dass Sie zweimal hintereinander im Lotto sechs Richtige haben, ist größer. Glauben Sie nicht? Die Wahrscheinlichkeit für einen Sechser im Lotto ist 1 zu 13.983.816, denn es gibt rund 13 Mio. Kombinationsmöglichkeiten 6 aus 49 (ohne zurücklegen) zu ziehen. Die Formel

```
=KOMBINATIONEN(49;6)
```

offenbart es.

Die Wahrscheinlichkeit, zweimal hintereinander sechs Richtige zu haben, ist 1 zu

```
=KOMBINATIONEN(49;6)^2= 195.547.109.921.856 = 10 hoch 14,3
```

und damit 20-mal höher als die Wahrscheinlichkeit, zwei gleiche Zufallszahlen zu ziehen.

Die Wahrscheinlichkeit, dass sich bei zehn Zufallszahlen zwei gleichen, ist dagegen wesentlich höher, nämlich rund 1 zu 10 ^ 14. Bei 100 Zufallszahlen sind es nur noch rund 1 zu 10 hoch 12 (eine Billion).

Noch nicht überzeugt? Wenn Sie so hartnäckig sind, müssen wir noch etwas weiter ausholen. Vielleicht kennen Sie das in der Kombinatorik bekannte Geburtstagsparadoxon. Es stellt die Frage, wie hoch die Wahrscheinlichkeit ist, dass auf einer Party zwei Personen am gleichen Tag Geburtstag haben. Die erstaunliche Antwort ist, dass bei mindestens 23 Partygästen mit größerer Wahrscheinlichkeit zwei Leute am gleichen Tag Geburtstag haben, als dass alle an unterschiedlichen Tagen Geburtstag haben, nämlich genau 50,7 %. Berechnet wird das mit der Formel

```
1 - (365/365)x(364/365) x(363/365)x...x(343/365) = 50,7 %
```

In unserer Muttersprache Excel ausgedrückt:

```
{=RUNDEN((1-PRODUKT(ZEILE(INDIREKT("365:343"))/365))*100;1)}
```

Das Geburtstagsparadoxon lässt sich nun 1:1 auf die Zufallszahlen projizieren. Statt 365 mögliche Tage, an denen man Geburtstag haben kann, gibt es 10 hoch 15,65... mögliche Zufallszahlen und die 100 gezogenen Zufallszahlen sind die 100 Partygäste. Angenommen, dass c die Anzahl möglicher Zufallszahlen ist, nämlich 2^52, dann berechnet sich die Wahrscheinlichkeit für zwei gleiche Zufallszahlen bei 100 Ziehungen aus

1 zu 1 /(1-(c/c)x((c-1)/c) x((c-2)/c)x...x((c-99)/c)) = 1 zu 10 hoch 11,95895.

Das Problem ist, dass Excel in diesem hohen Zahlenbereich nicht mehr genau rechnen kann, für Excel ergibt

```
=2^52=2^52-1=WAHR
```

Excel kann zwar Zahlen bis ca. 10^308 darstellen, aber nur bis zu 15 signifikanten Stellen genau. Trotzdem gibt es zwei Möglichkeiten, Aufgaben mit solch großen Zahlen wie 2^52 genau zu berechnen, entweder über die VBA-Typumwandlungsfunktion CDEC oder einige spezielle Berechnungen, auf die in Kapitel 12 »Zahlensysteme« näher eingegangen wird.

27.2 Minen zufällig verteilen

Jetzt aber endlich zurück zum Thema Minesweeper. Wir schreiben in A1:A100 `=ZUFALLSZAHL()`. Als Nächstes wird in Spalte B eine Rangfolge dieser 100 Zufallszahlen gebildet.

`B1: =RANG(A1;$A$1:$A$100)` (kopieren bis B100)

Nun stehen in B1:B100 alle Zahlen von 1 bis 100 ungeordnet untereinander.

Als Nächstes definieren wir eine Zelle *n*, die die Anzahl zu verteilender Minen auf dem Feld vorgibt. In C1 schreiben Sie

```
C1: =WENN(Zeile()>n;0;B1)
```

und kopieren die Formel bis C100. Die Spalte C verfolgt den Zweck, aus Spalte B genau *n* Zufallszahlen aus dem Zahlenbereich herauszupicken, beispielsweise für n = 10

```
{1;21;98;68;64;47;7;5;86;48}
```

AH19

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	0,394942	64	64		Minen:		1	2	3	4	5	6	7	8	9	10											
2	0,567110	38	38		10		11	12	13	14	15	16	17	18	19	20		0	0	0	0	0	1	0	0	0	0
3	0,490001	54	54				21	22	23	24	25	26	27	28	29	30		0	0	0	0	0	0	0	0	0	0
4	0,078042	94	94				31	32	33	34	35	36	37	38	39	40		0	0	0	0	0	0	0	0	0	0
5	0,604370	36	36				41	42	43	44	45	46	47	48	49	50		0	0	0	0	0	1	0	1	0	0
6	0,028198	98	98				51	52	53	54	55	56	57	58	59	60		1	0	0	0	0	0	0	0	0	0
7	0,553483	41	41				61	62	63	64	65	66	67	68	69	70		0	0	0	1	0	0	0	0	0	0
8	0,965183	6	6				71	72	73	74	75	76	77	78	79	80		0	0	0	1	0	0	0	0	0	0
9	0,256932	78	78				81	82	83	84	85	86	87	88	89	90		0	1	0	0	0	0	0	1	0	0
10	0,331785	72	72				91	92	93	94	95	96	97	98	99	100		0	0	0	0	0	0	0	0	0	0
11	0,158384	85	0															0	0	0	1	0	0	0	1	0	0
12	0,303885	75	0																								
13	0,498949	51	0																								
14	0,683244	30	0																								
15	0,026730	99	0																								
16	0,443757	58	0																								

Abbildung 27.2: Zehn Zellen der 10x10-Matrix zufällig auswählen

Die 100 Zellen des Minenfelds müssen von 1 bis 100 durchnummeriert werden, damit jede der 10 zufällig gezogenen Nummern einer Zelle zugeordnet werden kann. Sie können die Matrix G1:P10 in *Abb. 27.2* relativ leicht manuell durchnummerieren, aber wenn Sie es noch bequemer haben wollen, nehmen Sie dazu die Formel:

`G1: =10*(ZEILE(1:1)-1)+SPALTE(A:A)` (kopieren bis P10)

Die Formellösung ermöglicht später bei Bedarf eine flexiblere Erweiterung der Spielfeldgröße.

In der Matrix R1:AA10 werden letztlich die Minen verteilt, in R1 steht

`R1: =ISTZAHL(VERGLEICH(G1;$C:$C;0))*1` (kopieren bis AA10).

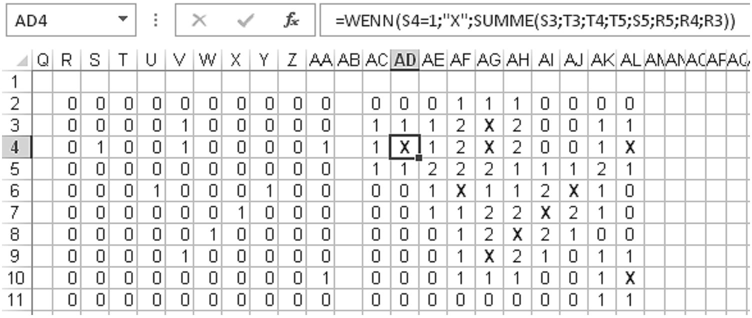
Wenn also G1 (die entsprechende Zelle der benachbarten Matrix G1:P10) in Spalte C gefunden wird, erzeugt die Formel in R1 eine 1, andernfalls eine 0. Jede 1 symbolisiert eine Mine. Sie haben jetzt *n* zufällige Zellen der Matrix in R1:AA10 mit Minen bestückt. Schön und gut, doch zum Spielen von Minesweeper haben Sie noch ein elementares Problem zu lösen: Jedes Mal, wenn Sie F9 drücken, werden die Minen neu verteilt, doch das ergibt keinen Sinn. Wenn die Minen einmal verteilt wurden, dürfen diese sich bis zum Ende des Spiels nicht mehr bewegen.

Jetzt kommt wieder die Excel-Iteration ins Spiel. Sie können mit ihr nicht nur iterative Berechnungen durchführen, Sie können mit ihr auch Zellwerte je nach Bedarf »einfrieren«. Genau wie beim Spiel des Lebens definieren Sie eine Zelle namens *Start*. Diese kann den Wert FALSCH bzw. 0 oder den Wert WAHR bzw. 1 annehmen. Nun erweitern Sie die Formel von C1 (dito C2:C100) auf

```
C1: =WENN(Start;WENN(Zeile()>n;0;B1);C1)
```

Dies hat zur Folge, dass in Spalte C nur dann neue Werte berechnet werden, wenn *Start* WAHR ist, ansonsten bleibt der eigene Zellwert erhalten, da sich die Zelle auf sich selbst bezieht. Wenn in den Excel-Optionen das Kontrollkästchen für die Excel-Iteration gesetzt ist, kommt Excel mit diesem Zirkelbezug klar. Damit wird der Zellwert so lange »eingefroren«, bis *Start* wieder den Wert WAHR erhält.

Als Nächstes benötigen Sie eine weitere Matrix (AC2:AL11), in der zum einen die Minen durch ein »X« markiert sind und in der zum anderen die Anzahl Minen in den acht angrenzenden Zellen zurückgegeben wird (*Abb. 27.3*):


AD4 =WENN(S4=1;"X";SUMME(S3;T3;T4;T5;S5;R5;R4;R3))

	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
1																						
2		0	0	0	0	0	0	0	0	0	0		0	0	0	1	1	1	0	0	0	0
3		0	0	0	0	1	0	0	0	0	0		1	1	1	2	X	2	0	0	1	1
4		0	1	0	0	1	0	0	0	0	1		1	X	1	2	X	2	0	0	1	X
5		0	0	0	0	0	0	0	0	0	0		1	1	2	2	2	1	1	1	2	1
6		0	0	0	1	0	0	0	1	0	0		0	0	1	X	1	1	2	X	1	0
7		0	0	0	0	0	0	1	0	0	0		0	0	1	1	2	2	X	2	1	0
8		0	0	0	0	0	1	0	0	0	0		0	0	0	1	2	X	2	1	0	0
9		0	0	0	0	1	0	0	0	0	0		0	0	0	1	X	2	1	0	1	1
10		0	0	0	0	0	0	0	0	0	1		0	0	0	1	1	1	0	0	1	X
11		0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	1

Abbildung 27.3: Minen auf dem Minenfeld verteilen

Wichtig: Die Matrix von R1:AA10 wurde ausgeschnitten und eine Zeile tiefer wieder eingefügt, damit sich die Formel in AC2 bequem auf die acht Nachbarzellen der Zelle R2 der Vorgängermatrix beziehen und einheitlich auf den Bereich bis AL11 kopieren lässt.

Jetzt wird noch eine weitere Matrix benötigt, die endlich das tatsächliche Spielfeld des Minensuchers darstellt. Überlegen wir uns, welche Eigenschaften das Spielfeld haben muss. Vor allem darf zu Beginn des Spiels die Position der Minen nicht ersichtlich sein. Erst wenn ein Feld ausgewählt (selektiert) wird, kommt eine Mine oder die Anzahl benachbarter Minen dieses Felds zum Vorschein.

27.3 Rechnen mit der aktiven Zelle

Dazu müsste es erst mal eine Möglichkeit geben, die selektierte Zelle überhaupt zu identifizieren. In VBA kein Problem: *ActiveCell.Address* gibt die Adresse der aktiven Zelle zurück. Aber in Excel gibt es »offiziell« keine Funktion, die die aktive Zelle ermitteln kann. Wenn man fleißig sucht, findet man aber dennoch eine passende. Die Formel

```
=ADRESSE(ZELLE("Zeile";A1);ZELLE("Spalte";A1))
```

gibt die Adresse von A1 zurück, also »$A$1«. Interessanterweise kann man das Bezugsargument einfach weglassen. Die Funktion

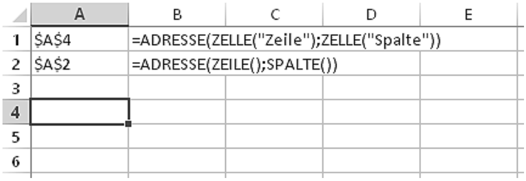
```
=ADRESSE(ZELLE("Zeile");ZELLE("Spalte"))
```

hat dann die Eigenschaft, die Adresse der Zelle zurückzugeben, die bei der letzten Neuberechnung aktiv war. Selektiert man eine Zelle und drückt dann [F9], wird also die aktive Zellenadresse ausgegeben.

Vielleicht kennen Sie die ähnlich aussehende und geläufigere Formel

`=ADRESSE(ZEILE();SPALTE())`

die die Adresse der Zelle wiedergibt, in der sie selbst steht (die aufrufende Zelle). Dass das nicht zum gleichen Ergebnis führt, zeigt das Beispiel in *Abb. 27.4*.



	A	B	C	D	E
1	\$A\$4	=ADRESSE(ZELLE("Zeile");ZELLE("Spalte"))			
2	\$A\$2	=ADRESSE(ZEILE();SPALTE())			
3					
4					
5					
6					

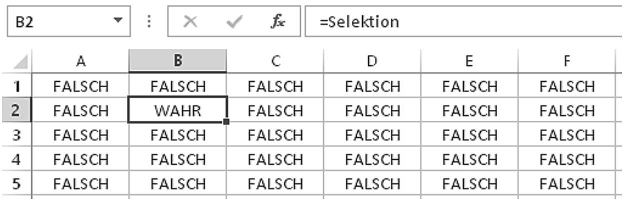
Abbildung 27.4: Unterschied zwischen der aktiven und der aufrufenden Zelle

Jetzt kommt der eigentliche Trick:

Definieren Sie den Namen *Selektion* bezogen auf die Formel

`=ADRESSE(ZELLE("Zeile");ZELLE("Spalte")) =ADRESSE(ZEILE();SPALTE())`

Um zu demonstrieren, wie sich der Name *Selektion* in einer Tabelle auswirkt, schreiben Sie in alle Zellen eines beliebigen Bereichs `=Selektion` (*Abb. 27.5*).



	A	B	C	D	E	F
1	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH
2	FALSCH	WAHR	FALSCH	FALSCH	FALSCH	FALSCH
3	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH
4	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH
5	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH

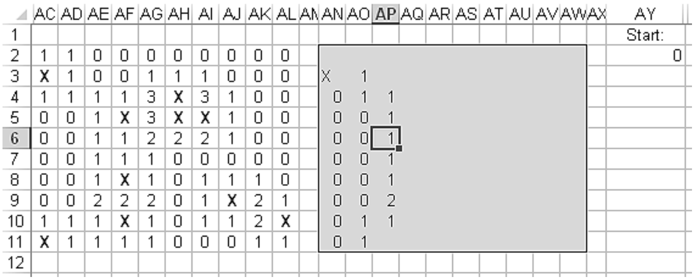
Abbildung 27.5: Rechnen mit der aktiven Zelle

Wenn Sie nun eine Zelle selektieren und F9 drücken, erscheint nur bei dieser Zelle WAHR. Und genau mit dieser Technik können Sie auf dem Minesweeper-Feld eine Zelle auswählen.

Die Matrix des Spielfelds geht von AN2:AW11. In AN2 steht die Formel

`AN2: {=WENN(start;""; WENN(AN2<>""; AN2; WENN( ODER((AM1:AO3=0)*(AM1:AO3<>"")); AC2; WENN(Selektion;AC2;"")))))}` (kopieren bis AW11).

Stellen Sie die Startzelle auf 1, um eine neue Startaufstellung des Minenfelds zu erhalten. Dann stellen Sie sie auf 0, um mit dem Minensuchen zu beginnen. Um eine Zelle zu betreten, beispielsweise AP6, drücken Sie [F9] *(Abb. 27.6)*.



	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY
1																							Start:
2	1	1	0	0	0	0	0	0	0	0													0
3	X	1	0	0	1	1	1	0	0	0		X	1										
4	1	1	1	1	3	X	3	1	0	0		0	1	1									
5	0	0	1	X	3	X	X	1	0	0		0	0	1									
6	0	0	1	1	2	2	2	1	0	0		0	0	1									
7	0	0	1	1	1	0	0	0	0	0		0	0	1									
8	0	0	1	X	1	0	1	1	1	0		0	0	1									
9	0	0	2	2	2	0	1	X	2	1		0	0	2									
10	1	1	1	X	1	0	1	1	2	X		0	1	1									
11	X	1	1	1	1	0	0	0	1	1		0	1										
12																							

Abbildung 27.6: Minesweeper – das Spiel beginnt.

Um zu verstehen, was da passiert ist, dröseln wir die Formel in AN2 auf:

- `{=WENN(start;""` solange *Start* auf 1 steht, bleibt das Spielfeld leer.
- `; WENN(AN2<>"";AN2` sobald in AN2 einmal ein Eintrag erscheint, bleibt er eingefroren, da sich die Zelle auf sich selbst bezieht.
- `WENN(ODER((AM1:AO3=0)*(AM1:AO3<>""));AC2` Wenn eine der acht Nachbarzellen eine 0 enthält, wird die aktive Zelle AN2 »aufgedeckt«, d. h., der entsprechende Wert der Matrix AC2:AL11, in diesem Fall AC2, wird übernommen. Denn dort, wo eine 0 steht, kann in einer der acht Nachbarn unmöglich eine Mine sein. Damit die Nullzellen nicht alle einzeln mit [F9] aufgedeckt werden müssen, geschieht das mit diesem Formelteil automatisch. Es entsteht eine Kettenreaktion, die alle zusammenhängenden Nullen auf einen Schlag aufdeckt. Genau so, wie das beim Original auch der Fall ist.
- `;WENN(Selektion;AC2;""))))}` Dieser Teil enthält das »Aufdecken« der ausgewählten Zelle. Wird bei Zellselektion [F9] gedrückt, liefert *Selektion* WAHR und der Wert aus AC2 wird übernommen.

Das Minesweeper-Feld ist bereits spielbar. Um dem Original von Microsoft noch etwas näher zu kommen, ergänzen wir die Formeln insoweit, als das komplette Minenfeld aufgedeckt wird, sobald der Spieler auf eine Mine getreten ist. Bei der jetzigen Formel könnte er trotz gesprengter Mine noch weiterspielen. Den letzten WENN-Ausdruck

```
WENN(Selektion;AC2;"")
```

ersetzen wir durch

```
WENN(Selektion;AC2;WENN(ZÄHLENWENN($AN$2:$AW$11;"X")>0;AC2;""))
```

um den gewünschten Effekt zu erzielen. Sobald ZÄHLENWENN ein »X« findet, werden die Zellen der Vorgängermatrix AC2:AL11 übernommen, das Minenfeld wird also komplett aufgedeckt.

Mit der Formel

```
=WENN(UND(ANZAHL(AN2:AW11)=100-n;ZÄHLENWENN(AN2:AW11;"x")=0);"gewonnen";"")
```

können Sie noch anzeigen lassen, wenn der Spieler gewonnen hat.

Der Rest ist Feinarbeit. Die Nullen können mit dem Zahlenformat `0;-0;""` ausgeblendet werden. Die Minen können statt mit einem »X« mit schöneren Symbolen angezeigt werden. Beispielsweise wird das M der Schriftart Wingdings durch eine Bombe dargestellt. Die nicht aufgedeckten Felder können über die bedingte Formatierung farblich unterlegt werden.

Das optisch getunte Endergebnis könnte dann beispielsweise wie in *Abb. 27.7* aussehen.

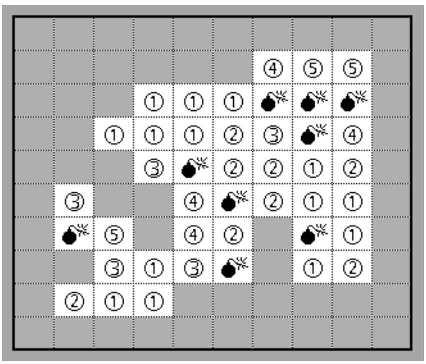
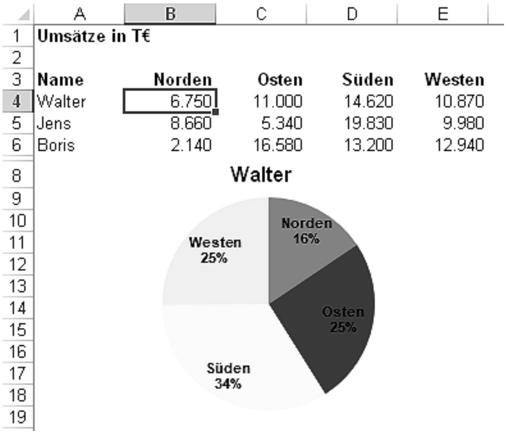


Abbildung 27.7: Ein optisch getuntes Spielfeld

27.4 Die aktive Zelle beeinflusst Diagramme

Wenn Sie genug Minesweeper gespielt haben, können Sie sich überlegen, wie Sie sich das Gezeigte noch anderweitig zu Nutze machen können. Per Formel auf die aktive Zelle zugreifen zu können, ist nämlich auch in anderen Excel-Welten sehr praktisch, beispielsweise bei Diagrammen (*Abb. 27.8*).



	A	B	C	D	E
1	**Umsätze in T€**				
2					
3	**Name**	**Norden**	**Osten**	**Süden**	**Westen**
4	Walter	6.750	11.000	14.620	10.870
5	Jens	8.660	5.340	19.830	9.980
6	Boris	2.140	16.580	13.200	12.940

Abbildung 27.8: Ein auf die aktive Zeile reagierendes Kuchendiagramm

Das Kuchendiagramm soll flexibel die Werte einer der drei Datenreihen anzeigen. Dazu selektieren Sie eine Zelle, führen mit [F9] eine Neuberechnung durch und das Diagramm soll die entsprechenden Werte anzeigen.

Die Diagrammüberschrift erhalten Sie mit der Formel

```
=INDIREKT(ADRESSE(ZELLE("Zeile");1))
```

`ZELLE("Zeile")` liefert ohne Bezugsangabe die Zeilennummer der aktiven Zelle. Die Spaltenangabe ist auf 1 fixiert, dann kann auch B1 selektiert werden und man erhält immer noch den richtigen Namen aus der Spalte A.

Bekanntlich können Sie für die Datenreihen eines Diagramms Namen definieren. Erstellen Sie also einen Namen *Daten*, dem der Bereich $B$4:$E$4 hinterlegt ist, und binden Sie ihn in das Diagramm ein. Die DATENREIHE-Definition, die Sie bei selektierter Datenreihe in der Bearbeitungsleiste sehen, lautet dann nicht mehr

```
=DATENREIHE(;Tabelle1!$B$3:$E$3;Tabelle1!$B$4:$E$4;1),
```

sondern

```
=DATENREIHE(;Tabelle1!$B$3:$E$3;Mappe1.xls!Daten;1)
```

Die Überschrift des Diagramms steht in der Zelle C8 mit der Formel

```
C8: =INDIREKT(ADRESSE(ZELLE("Zeile");1))
```

Wenn das geklappt hat, ersetzen Sie im Namensdialogfeld *Formeln>Namens-Manager* den Zellbezug des Namens durch die Formel

```
=INDEX(Tabelle1!$B:$E;ZELLE("Zeile");)
```

Diese Formel gibt einen Bereich zurück. Von Spalte B:E eine Zeile, und zwar die Zeile, die bei der letzten Neuberechnung aktiv war. Jetzt wird sich Ihr Kuchendiagramm immer an die aktive Zeile anpassen, vorausgesetzt, es findet eine Neuberechnung statt.

Falls Sie eine Zeile unterhalb des Datenbereichs selektieren, kann das Diagramm nach der Neuberechnung nichts mehr anzeigen und der Kuchen verschwindet. Wenn Sie das verhindern wollen, können Sie eine Bedingung ergänzen:

```
=INDEX(Tabelle1!$B:$E;MIN(ZELLE("Zeile");ANZAHL2(Tabelle1!$A$4:$A$100)+3);)
```

MIN in Kombination mit ANZAHL2 sorgt hier dafür, dass höchstens die letzte Zeile des Datenbereichs, im Beispiel die Zeile 6, zurückgegeben wird.

Vorsicht: Sobald Sie dem Namen *Daten*, der in das Diagramm eingebunden ist, die Formel mit der aktiven Zelle zuordnen, können Sie die Datenquellbezüge des Diagramms nicht mehr bearbeiten. Denn in dem Moment, in dem Sie das Diagramm bearbeiten, gibt es keine aktive Zelle, da das Diagramm den Fokus erhält. Aus diesem Dilemma kann sich Excel nicht befreien und streikt. Deshalb darf dieser Schritt erst ganz am Schluss erfolgen. Wenn Sie aber nachträglich andere Zellbezüge anpassen müssen, ordnen Sie zuvor dem Namen *Daten* wieder einen fixen Bereich wie $B$4:$E$4 zu.