Tutorial: Beep

Ich möchte mich in diesem Tutorial mit der Windows-Funktion Beep beschäftigen.
Diese Funktion lässt den SystemSpeaker einen Ton mit einer bestimmten Frequenz und für eine bestimmte Dauer erklingen.

== Gerüst ==
Hierzu ist
Quellcode

```
1. #include <windows.h>
2. #include <iostream>
3. using namespace std;
```

erforderlich. Außerdem importiert ihr iostream und verwenden den namespace std für Ausgaben in der Konsole.
Jetzt bestimmt ihr die Frequenz der Töne als const int, weil sie ja nicht verändert werden müssen.

== Musiklehre ==
Wichtig ist zu wissen, dass die Hälfte einer Frequenz, der gleiche Ton eine Oktave tiefer, und, dass das Doppelte der Frequenz eine Oktave höher ist.
Wenn man die Frequenz selber berechnen möchte, muss man die 12. Wurzel aus 2 mit der Frequenz des Tones, der eine halbe Stufe unter dem gesuchten Ton liegt multiplizieren.

Ein Beispiel:
- Ton A = 440Hz (ein normal temperierte Klaviersaite des Tons A schwingt mit dieser Frequenz (es kann auch mit der Hälfte oder dem Doppelten gerechnet werden))
- 12. Wurzel aus 2 * Frequenz des Tones = Halbton höher (Ais) = 466
- beliebig oft wiederholen 🙂

== Frequenzen ==
Also, für diejenigen, die die Frequenzen nicht selber berechnen möchten:
Quellcode

```
1. const int C = 261;
2. const int Cis = 277;
3. const int D = 293;
4. const int Dis = 311;
5. const int E = 329;
6. const int F = 349;
7. const int Fis = 369;
8. const int G = 391;
9. const int Gis = 415;
10. const int A = 440;
11. const int Ais = 466;
12. const int H = 493;
13. const int Takt = 2000;
```

Alles anzeigen

Was die Variable Takt soll, könnt ihr euch sicherlich denken.
Recht praktisch ist, jeden Beep so zu gestalten:
So ertönt der Ton C1 als Viertelnote.
Quellcode

```
1. Beep(C * 1, Takt / 4);
```

Wieder der Ton C, allerdings doppelt so lang und eine Oktave höher.

Quellcode

```
1. Beep(C * 2, Takt / 2);
```

Und nochmal der Ton C, diesmal jedoch 4 Mal so lang und 2 Oktaven höher.

Quellcode

```
1. Beep(C * 4, Takt / 4);
```

Um einen Ton eine Oktave niedriger erklingen zu lassen, einfach das * Zeichen mit / ersetzen.
Eventuell ist es erforderlich, das ihr hinter einen jeden Beep Aufruf das Programm kurz mit Sleep(1); anhalten und danach weiterlaufen lassen, beispielsweise bei 16tel Noten.
Ihr könnt ja mal Probieren "Alle meine Entchen" zu programmieren.

== Beispiel Tetris ==
Hier noch der Sourcecode von Tetris.

Quellcode

```
1. #include <windows.h>
2. #include <iostream>
3. using namespace std;
4. 
5. const int C = 261;
6. const int Cis = 277;
7. const int D = 293;
8. const int Dis = 311;
9. const int E = 329;
10. const int F = 349;
11. const int Fis = 369;
12. const int G = 391;
13. const int Gis = 415;
14. const int A = 440;
15. const int Ais = 466;
16. const int H = 493;
17. const int Takt = 1700;
18. 
19. int main() {
20. cout << "Tetris" << endl << " (Enter dr" << (unsigned char)129 << "cken um fortzufahren)";
21. getchar();
22. while (1) {
23. Sleep(Takt / 4);
24. Beep(E * 2, Takt / 4);
25. Beep(H * 1, Takt / 8);
26. Beep(C * 2, Takt / 8);
27. Beep(D * 2, Takt / 4);
28. Beep(C * 2, Takt / 8);
29. Beep(H * 1, Takt / 8);
30. Beep(A * 1, Takt / 4);
31. Beep(A * 1, Takt / 8);
32. Beep(C * 2, Takt / 8);
33. Beep(E * 2, Takt / 8);
34. Beep(E * 2, Takt / 8);
35. Beep(D * 2, Takt / 8);
36. Beep(C * 2, Takt / 8);
```

```
37.  Beep(H * 1, Takt / 2.5);
38.  Beep(C * 2, Takt / 8);
39.  Beep(D * 2, Takt / 4);
40.  Beep(E * 2, Takt / 4);
41.  Beep(C * 2, Takt / 4);
42.  Beep(A * 1, Takt / 4);
43.  Beep(A * 1, Takt / 4);
44.  Sleep(Takt / (8 / 3));
45.  Beep(D * 2, Takt / 3.25);
46.  Beep(F * 2, Takt / 8);
47.  Beep(A * 2, Takt / 8);
48.  Beep(A * 2, Takt / 8);
49.  Beep(G * 2, Takt / 8);
50.  Beep(F * 2, Takt / 8);
51.  Beep(E * 2, Takt / 3);
52.  Beep(C * 2, Takt / 8);
53.  Beep(E * 2, Takt / 8);
54.  Beep(E * 2, Takt / 8);
55.  Beep(D * 2, Takt / 8);
56.  Beep(C * 2, Takt / 8);
57.  Beep(H * 1, Takt / 4);
58.  Beep(H * 1, Takt / 8);
59.  Beep(C * 2, Takt / 8);
60.  Beep(D * 2, Takt / 4);
61.  Beep(E * 2, Takt / 4);
62.  Beep(C * 2, Takt / 4);
63.  Beep(A * 1, Takt / 4);
64.  Beep(A * 1, Takt / 4);
65.  }
66.  return 0;
67.  }
```

Alles anzeigen