This program simply plays the notes in the correct sequence with equal durations and pauses. Consequently, the resulting tune is somewhat stilted. With experimentation you can construct more complex programs that provide different intervals and durations for individual notes.

Sound Effects

By using two or three oscillators it is possible to play simple chords. The program below plays the chord of D major (F#, A and D) starting with the F# on its own, and adding the A and D after set delays of one second each. The chord then continues for a further two seconds.

```
10 POKE 36878,7
20 POKE 36874,233:D=TI
30 IF TI-D < 60 THEN 30
40 POKE 36875,219:D=TI
50 IF TI-D < 60 THEN 50
60 POKE 36875,147:D=TI
70 IF TI-D < 120 THEN 70
80 POKE 36878,0: POKE 36874,0
90 POKE 36875,0: POKE 36876,0
100 END
```

A lot can be done, however, to make the tone of these sounds more interesting. For instance, the volume can be varied over the duration of a note — rising and falling according to a variable. For example:

```
100 V = 36878
110 FOR I= 1 TO 12
120 POKE V,I
```

```
130 NEXT I
140 POKE V,0
```

This causes the volume to rise in steps of 1 to a peak of 12, where the total range is from 0 (off) to 15 (loud). Volume can be 'pulsed' by alternating a high and low volume setting, as well. The frequency can be similarly varied to 'bend' a note by changing line 120 above to:

```
POKE V-3,203+I
```

It is also worth trying different combinations of noise, oscillator frequencies and volumes. This can often result in a more pleasing tone. Whether making music or adding sound effects to games, the aim in computing is to reduce boredom by avoiding the constant repetition of monotonous notes.

We have shown how the simple sound facilities on the Vic-20 can be manipulated to produce interesting tones and note sequences. The main problem is the lack of sound commands, which involves the use of complex BASIC statements to carry out relatively simple tasks. This results in long program routines that prevent the BASIC interpreter from processing the code in between notes quickly enough. The only simple way to avoid this problem is to invest in one of the many commercial software packages that supply extra commands for music programming. Commodore's Super Expander cartridge provides a useful range of sound commands, as well as a facility for storing tunes written with the aid of the cartridge. However, if you require more than rudimentary sound or music facilities from a home computer it would be necessary to investigate other models, such as the BBC Micro, the Commodore 64, the Dragon 32 or the Oric-1.

The high resolution screen is defined with its origin in the bottom left-hand corner of the screen, regardless of the mode selected. Vertical values range from 0 to 1023, and horizontal values range from 0 to 1279. This consistent method of mapping the screen becomes very convenient when you decide to change the display from one mode to another. Incidentally, if the mode of display is changed during the course of a program then the screen is automatically cleared.

Background, text and graphics colours are set using the COLOUR and GCOL commands. The BBC Micro uses the interesting idea of logical and actual colours to allow the user to select a limited set of colours from the 16 allowed. To illustrate this it is best to use the example of using colour in MODE 0 where only two colours can be specified. Two possible foreground colours are given the logical colour numbers 0 and 1, and unless the computer is instructed to do otherwise, it takes 0 as black and 1 as white. The COLOUR command selects the text foreground colour. COLOUR 1 would select logical colour number 1 as the text colour, but it is possible to reset the logical text colour using one of the VDU commands. VDU19

defines the logical colour. To set logical colour 1 to green (actual colour number 2) the following command is needed:

```
VDU19, 1, 2, 0, 0, 0,
```

The three noughts on the end have no significance and are there for future expansion of the system.

The GCOL command has two numbers associated with it. The second number is the logical colour number for graphics display, the first relates to the way in which that colour is used on the screen. For the command GCOL a,b values of a can range from 0 to 4 allowing the user to specify whether the point or line should be displayed in the logical foreground colour, whether it should be ANDed, ORed or exclusive ORed with the colour already present, or whether the original colour should be inverted.

In a future part of the Sound And Light course we will return to the BBC Micro and explain high resolution capabilities, defining characters, and look more closely at the set of VDU commands.