

# Mozart Waltz Generator

Project idea from David Costanzo and Kevin Wayne at Princeton – thanks!

**NOTE:** playing even waltzes will be distracting to others so please use headphones! I'll lend you some.

In 1787, Wolfgang Amadeus Mozart created a dice game ([Mozart's Musikalisches Würfelspiel](#)). In the game, you compose a two part waltz by pasting together 32 of 272 pre-composed musical elements at random.

There are  $11^{16} * 6^{16}$  different possible results, some of which are more likely than others. Since this is over  $10^{23}$  different possibilities, each time you play the game you are likely to compose a piece of music that has never been heard before! Mozart carefully constructed the measures to obey a rigid harmonic structure, so each waltz reflects Mozart's distinct style. Unfortunately, due to the rigidity, the process never results in anything truly extraordinary

**The waltz.** The waltz consists of two parts - the minuet and the trio. Each is comprised of 16 measures, which are generated at random according to a fixed set of rules, as described below.

- *Minuet.* The minuet consists of 16 measures. There are 176 possible Minuet measures, named `M1.wav` through `M176.wav`. To determine which one to play, roll two fair dice, and use the following table.

|    | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  | 16  |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 2  | 96  | 22  | 141 | 41  | 105 | 122 | 11  | 30  | 70  | 121 | 26  | 9   | 112 | 49  | 109 | 14  |
| 3  | 32  | 6   | 128 | 63  | 146 | 46  | 134 | 81  | 117 | 39  | 126 | 56  | 174 | 18  | 116 | 83  |
| 4  | 69  | 95  | 158 | 13  | 153 | 55  | 110 | 24  | 66  | 139 | 15  | 132 | 73  | 58  | 145 | 79  |
| 5  | 40  | 17  | 113 | 85  | 161 | 2   | 159 | 100 | 90  | 176 | 7   | 34  | 67  | 160 | 52  | 170 |
| 6  | 148 | 74  | 163 | 45  | 80  | 97  | 36  | 107 | 25  | 143 | 64  | 125 | 76  | 136 | 1   | 93  |
| 7  | 104 | 157 | 27  | 167 | 154 | 68  | 118 | 91  | 138 | 71  | 150 | 29  | 101 | 162 | 23  | 151 |
| 8  | 152 | 60  | 171 | 53  | 99  | 133 | 21  | 127 | 16  | 155 | 57  | 175 | 43  | 168 | 89  | 172 |
| 9  | 119 | 84  | 114 | 50  | 140 | 86  | 169 | 94  | 120 | 88  | 48  | 166 | 51  | 115 | 72  | 111 |
| 10 | 98  | 142 | 42  | 156 | 75  | 129 | 62  | 123 | 65  | 77  | 19  | 82  | 137 | 38  | 149 | 8   |
| 11 | 3   | 87  | 165 | 61  | 135 | 47  | 147 | 33  | 102 | 4   | 31  | 164 | 144 | 59  | 173 | 78  |
| 12 | 54  | 130 | 10  | 103 | 28  | 37  | 106 | 5   | 35  | 20  | 108 | 92  | 12  | 124 | 44  | 131 |

For example, if you roll an 11 for measure 3, then play measure [165](#).

- *Trio.* The trio consists of 16 measures. There are 96 possible Trio measures named `T1.wav` through `T96.wav`. To determine which one to play, roll one fair die, and use the following table.

|   | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 72 | 6  | 59 | 25 | 81 | 41 | 89 | 13 | 36 | 5  | 46 | 79 | 30 | 95 | 19 | 66 |
| 2 | 56 | 82 | 42 | 74 | 14 | 7  | 26 | 71 | 76 | 20 | 64 | 84 | 8  | 35 | 47 | 88 |
| 3 | 75 | 39 | 54 | 1  | 65 | 43 | 15 | 80 | 9  | 34 | 93 | 48 | 69 | 58 | 90 | 21 |
| 4 | 40 | 73 | 16 | 68 | 29 | 55 | 2  | 61 | 22 | 67 | 49 | 77 | 57 | 87 | 33 | 10 |
| 5 | 83 | 3  | 28 | 53 | 37 | 17 | 44 | 70 | 63 | 85 | 32 | 96 | 12 | 23 | 50 | 91 |
| 6 | 18 | 45 | 62 | 38 | 4  | 27 | 52 | 94 | 11 | 92 | 24 | 86 | 51 | 60 | 78 | 31 |

**Start by generating the waltz.** For this part, your goal is to generate a random sequence of measures according to the rules described above. The file `waltz_start.py` contains an 13-by-17 array `minuet` initialized such that `minuet[i][j]` is the musical element to play in measure `j` if you roll `i`. The same file also contains a 7-by-17 array `trio` such that `trio[i][j]` is the musical element to play in measure `j+16` if you roll `i`.

It will be useful for you to generate a list of filenames chosen that can be played later.

**Then play the waltz.** Play each measure using your list of filenames and the `winsound` module (details below)

**Data files.** The 272 measures take up quite a bit of space so let me know if you want to attempt this exercise and I'll give them to you from my jump drive.

**Playing audio.** Python has a built-in module called `winsound` that will let you play wav files. In order to use it, you will need to import it. You should only need to use the `PlaySound` function, but the full module documentation can be found at <http://docs.python.org/library/winsound.html>

Here is the code needed to play the file "music.wav" WHICH MUST BE IN THE SAME DIRECTORY AS YOUR CODE:

```
from winsound import *
PlaySound("music.wav", SND_FILENAME)
```

## EXTENSIONS

Can you generate other kinds of music?

- Read about music generation algorithms:  
[http://en.wikipedia.org/wiki/Algorithmic\\_composition](http://en.wikipedia.org/wiki/Algorithmic_composition)
- Download some beats from <http://free-loops.com>, for example, and see if you can come up with interesting random patterns!