

# **Workshop 05:**

## **TidalCycles Practical Tutorial 2**

### **Chords and Scale.**

# Notes and Melodies

## From Drums to Synthesizers

- The sequence parts are mainly sampling (pre-recorded sounds) like drums.
- In this part, we will use pitched instruments like pianos and synthesizers.



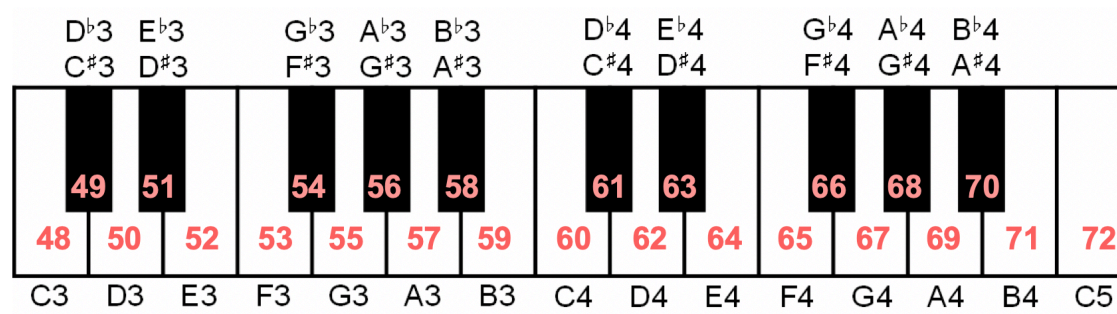
## Specifying Pitch with Numbers

```
-- tidalcycles  
d1 $ note "0 4 7 11" # s "superpiano"
```

```
// Strudel  
note("48 52 55 59").sound("piano")
```

- Notes are generated with `note()`.
- A sequence of numbers becomes the pitch.
- The numbers are MIDI note numbers.
- Middle C on the piano is 60 (or 0 in TidalCycles).

MIDI note numbers  
(MIDI pitches)



## Specifying Notes by Letter

```
-- tidalcycles  
d1 $ note "c e g b" # s "superpiano"
```

```
// Strudel  
note("c e g b").sound("piano")
```

- Specify by English note names.
- c (Do), d (Re), e (Mi), f (Fa), g (Sol), a (La), b (Si)

## Specifying the Octave

```
-- tidalcycles  
d1 $ note "c4 e4 g4 b4" # s "superpiano"
```

```
// Strudel  
note("c4 e4 g4 b4").sound("piano")
```

- Note name + number: specifies the octave.

## Chromatic Scale (Sharps and Flats)

Flat (  $\flat$  ): Lower by a semitone

```
-- tidalcycles  
d1 $ note "c4 d4 ef4 g4" # s "superpiano"
```

```
// Strudel  
note("c4 d4 ef4 g4").sound("piano")
```



## Sharp ( # ): Raise by a semitone

```
-- tidalcycles  
d1 $ note "c4 d4 e4 gs4" # s "superpiano"
```

```
// Strudel  
note("c4 d4 e4 gs4").sound("piano")
```

- Flat (lower by a semitone): note name + f
- Sharp (raise by a semitone): note name + s

## Changing the Timbre

```
-- tidalcycles  
d1  
  d1 $ note "c4 e4 g4 b4" # s "supersaw"
```

```
// Strudel  
note("c4 e4 g4 b4").sound("supersaw")
```

- Try changing the content of `sound()` to something else → e.g., `supersaw`.
- Many other timbres are also available.

# Timbre Sequence

```
-- tidalcycles  
d1  
  $ note "c4 d4 e4 [g4, b4]"  
  # s "superpiano supersaw"
```

```
// Strudel  
note("c4 d4 e4 [g4, b4]")  
  .sound("piano gm_electric_guitar_muted")
```

- List multiple instruments in `sound()` separated by spaces.
- The timbre changes during the sequence.
- It's also possible to build up a sequence.

## Layering Timbres

```
-- tidalcycles  
d1  
  $ note "c4 d4 e4 [g4, b4]"  
  # s "superpiano, supersaw"
```

```
// Strudel  
note("c4 d4 e4 [g4, b4]")  
  .sound("piano, gm_electric_guitar_muted")
```

- Separate with a comma ( , ) instead of a space in `sound()`.
- This plays multiple instruments simultaneously, not as a sequence.
- This is called stacking instruments.

## One Sequence per Cycle

```
-- tidalcycles
d1
  $ note "<[c3 g3]*4 [c3 f3]*4 [d3 g3]*4 [c3 c4]*4>/2"
  # s "supersaw"
```

```
// Strudel
note("<[c3 g3]*4 [c3 f3]*4 [d3 g3]*4 [c3 c4]*4>/2")
  .sound("supersaw")
```

- The sequence changes with each cycle.
- Use `< >` to create a sub-sequence.
- They are played in order!

## Alternating Performance

```
-- tidalcycles  
d1  
  $ note "[c3 <e3 f3 g3 b3>]*2"  
  # s "supersaw"
```

```
// Strudel  
note("[c3 <e3 f3 g3 b3>]*2")  
  .sound("supersaw")
```

- Combine a normal sequence with `< >`.
- The performance alternates.

## Stretching a Note

```
-- tidalcycles  
d1  
  $ note "[c3@3 <e3 f3 g3 b3>]*2"  
  # s "supersaw"
```

```
// Strudel  
note("[c3@3 <e3 f3 g3 b3>]*2")  
  .sound("supersaw")
```

- A note is stretched with `note@number`.
- `@3` means three times the length.

## Replicating a Note

```
-- tidalcycles  
d1  
  $ note "[c3!3 <e3 f3 g3 b3>]*2"  
  # s "supersaw"
```

```
// Strudel  
note("[c3!3 <e3 f3 g3 b3>]*2")  
  .sound("supersaw")
```

- Replicate a note with `!`.
- It's similar to `@` but it retriggers the note instead of stretching it.



## Let's Try!

- Using what you've learned so far, try creating your own original sequence!
- Mix drum sounds, instruments, etc.

# Chords and Scales

## Playing Sounds Simultaneously

Enclosing notes in `[ ]` and separating them with commas plays them at the same time. This creates a list.

```
-- TidalCycles
d1
  $ s "superpiano*4"
  # note "[c, e, g]"
```

```
// Strudel
note("[c, e, g]").sound("piano*4")
```

# Playing a Chord

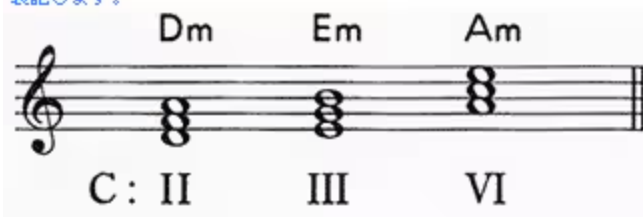
What is a chord?

It's when you play multiple (usually three or more) notes of different pitches at the same time.

長三和音はC(シー)、F(エフ)、G(ジー)、と表記します。



minorのmは必ずつきます。  
Dm(ディーマイナー) Em(イーマイナー) Am(エーマイナー)と  
表記します。



## Generating a List from a Chord Name → to a Chord

```
-- TidalCycles  
d1  
  $ s "superpiano*4"  
  # note "c'maj"
```

```
// Strudel  
$: sound("piano*4")  
  .n("[1, 2, 3, 4]")  
  .chord("C").voicing()
```

## Trying Different Chords

```
-- TidalCycles
d1
  $ s "superpiano*4"
  # note "<c'maj f'maj g'maj c'maj>"

-- For a nicer voicing
d1
  $ s "superpiano"
  # note "<c5'maj f4'maj'ii g4'dom7'ii c5'maj'i>"
```

```
//Strudel
$: sound("piano*4")
  .n("[1, 2, 3, 4]")
  .chord("<C F G C>").voicing()
```

## Pachelbel's Canon Progression - C, G, Am, Em, F, C, F, G

```
-- TidalCycles
d1
  $ s "superpiano*2"
  # note "<c'maj g'maj a'min e'min f'maj c'maj f'maj g'maj>"

-- Canon progression with nice voicing
d1
  $ s "superpiano*2"
  # note "<c4'maj'ii g4'maj a3'min'ii e4'min f4'maj c4'maj f4'maj g3'dom7'iii>"
```

```
//Strudel
$: sound("piano*4")
  .n("[0, 1, 2, 3, 4]")
  .chord("<C G Am Em F C F G>").voicing()
```

## “Royal Road” Progression (J-POP Style) - F△7–G7–Em7–Am

```
-- TidalCycles
d1
  $ s "superpiano*2"
  # note "<f5'maj7 g4'dom7'ii e5'min7 a4'min'ii>"
```

```
//Strudel
$: sound("piano*2")
  .n("[1, 2, 3, 4, 5]")
  .chord("<FM7 G7 Em7 Am>").voicing()
```



# “Royal Road” Progression Ensemble

```
-- TidalCycles
do
  d1
    $ s "supersaw(11, 16)"
    # note "<f5'maj7 g4'dom7'ii e5'min7 a4'min'ii>"
    # lpf (range 2000 8000 $ slow 16 $ sine)
    # resonance 0.2
    # room 0.5 # size 0.8
  d2
    $ s "supersaw(9, 16)"
    # note "<f2 g2 e2 a2>"
    |+ note "<0 12>*8"
    # lpf (range 1000 8000 $ slow 12 $ sine)
    # resonance 0.2
    # gain 1.5
  d3
    $ s "[bd(3, 8), hc(4, 8, 1)]"
    # gain 1.5
```

```
// Strudel
$: sound("supersaw(11, 16)")
  .n("[1, 2, 3, 4, 5]")
  .chord("<FM7 G7 Em7 Am>").voicing()
  .lpf(sine.slow(32).range(1000, 8000)).resonance(10)
  .room(0.9)
$: sound("supersaw(9, 16)")
  .note("<F2 G2 A2 E2>".add("<-12 0 12>*4"))
  .lpf(sine.slow(24).range(1000, 8000)).resonance(8)
  .room(0.9).gain(1.8)
$: sound("bd(3, 8), hh(4, 8, 1)").bank("RolandTR909")
  .gain(1.2)
```

## “Royal Road” Progression Ensemble + Slash Chords

```
--- TidalCycles
do
  d1
    $ s "supersaw(11, 16)"
    # note "<f5'maj7 g4'dom7'ii e5'min7 a4'min'ii>"
    # lpf (range 1000 8000 $ slow 16 $ sine)
    # resonance 0.2
    # room 0.5 # size 0.8
  d2
    $ s "supersaw(9, 16)"
    # note "<f2 e2 d2 c2>/2"
    |+ note "<0 12>*8"
    # lpf (range 1000 8000 $ slow 12 $ sine)
    # resonance 0.2
    # gain 1.5
  d3
    $ s "[bd(3, 8), hc(4, 8, 1)]"
    # gain 1.5
```

```
// Strudel
$: sound("supersaw(9, 16, 3)")
  .n("[1, 2, 3, 4, 5]")
  .chord("<FM7 G7 Em7 Am>").voicing()
  .lpf(sine.slow(32).range(1000, 8000)).resonance(10)
  .room(0.9)
$: sound("supersaw(9, 16)")
  .note("<F2 E2 D2 [C2 A1]>/2".add("<-12 0 12>*4"))
  .lpf(sine.slow(24).range(1000, 4000)).resonance(10)
  .gain(1.8)
$: sound("bd(4, 8), hh(4, 8, 1)").bank("RolandMC303")
  .delay(0.5).delaytime(3/8)
  .gain(1.3)
```

## **Scales (Modes)**

## What is a Scale (Mode)?

- A systematically arranged collection of notes used in music.
- It forms the foundation of a musical piece and is a key element in determining its mood and character.
- Representative Scales:
  - Major Scale
  - Minor Scale
  - Modes (discussed later)
  - Pentatonic Scale
  - Chromatic Scale

## Modal Jazz


- A style of jazz that involves improvisation based on a specific mode (scale).
- It does not rely on traditional chord progressions.
- It is characterized by a free approach that takes advantage of the characteristic sounds and atmosphere of the mode.
  - Example: Miles Davis's "Kind of Blue"
  - D Dorian mode → E  $\flat$  Dorian mode → D Dorian mode
  - [Spotify: Modal Jazz](#)

# The Seven Modes

Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian, Locrian

**"Major"**


Ionian (Major)      Lydian (Major with raised 4)      Mixolydian (Major with lowered 7)



The first staff of music shows three modes on a treble clef. The Ionian mode (Major) is represented by a natural scale: C, D, E, F, G, A, B. The Lydian mode (Major with raised 4) is represented by a scale with a sharp on the fourth degree: C, D, E, F#, G, A, B. The Mixolydian mode (Major with lowered 7) is represented by a scale with a flat on the seventh degree: C, D, E, F, G, A, Bb.

**"Minor"**

Aeolian (Natural Minor)      Dorian (Minor, raised 6)      Phrygian (Minor, lowered 2)      Locrian (Minor, lowered 2 & 5)



The second staff of music shows four modes on a treble clef. The Aeolian mode (Natural Minor) is represented by a scale with a flat on the second degree: C, Db, Eb, F, G, Ab, Bb. The Dorian mode (Minor, raised 6) is represented by a scale with a flat on the second degree and a sharp on the sixth degree: C, Db, Eb, F, G, F#, Ab, Bb. The Phrygian mode (Minor, lowered 2) is represented by a scale with a flat on the second degree and a flat on the first degree: C, Bb, Ab, G, F, Eb, Db, C. The Locrian mode (Minor, lowered 2 & 5) is represented by a scale with a flat on the second degree and a flat on the fifth degree: C, Bb, Ab, Gb, F, Eb, Db, C.

(Also: Locrian: Phrygian, lowered 5)



# Playing a Scale

## Ionian Scale

```
-- TidalCycles
d1
  $ s "supersaw*8"
  # note (scale "ionian" "[0..7]")
```

```
// Strudel
$: sound("piano*8")
  .n("0 1 2 3 4 5 6 7")
  .scale("C:ionian")
```

## Pre-defined Scales

A large number of scales are pre-defined! Let's try them out!

```
-- TidalCycles
minPent majPent ritusen egyptian kumai hirajoshi iwato
chinese indian pelog prometheus scriabin gong shang jiao
zhi yu whole augmented augmented2 hexMajor7 hexDorian
hexPhrygian hexSus hexMajor6 hexAeolian major ionian
dorian phrygian lydian mixolydian aeolian minor locrian
harmonicMinor harmonicMajor melodicMinor
melodicMinorDesc melodicMajor bartok hindu todi purvi
marva bhairav ahirbhairav superLocrian romanianMinor
hungarianMinor neapolitanMinor enigmatic spanish
leadingWhole lydianMinor neapolitanMajor locrianMajor
diminished diminished2 chromatic
```

```
// Strudel
major pentatonic, major, minor, major blues, minor
blues, melodic minor, harmonic minor, bebop, diminished,
dorian, lydian, mixolydian, phrygian, locrian, ionian
pentatonic, mixolydian pentatonic, ritusen, egyptian,
neopolitan major pentatonic, vietnamese 1, pelog,
kumoiyoshi, hirajoshi, iwato, in-sen, lydian pentatonic,
malkos raga, locrian pentatonic, minor pentatonic, minor
six pentatonic, flat three pentatonic, flat six
pentatonic, scriabin, whole tone pentatonic, lydian #5P
pentatonic, lydian dominant pentatonic, minor #7M
pentatonic, super locrian pentatonic, minor hexatonic,
augmented, piongio, prometheus neopolitan, prometheus,
mystery #1, six tone symmetric, whole tone,
messiaen's mode #1, messiaen's mode #5, locrian major,
double harmonic lydian, altered, locrian #2, mixolydian
b6, lydian dominant, lydian augmented, dorian b2,
ultralocrian, locrian 6, augmented heptatonic, dorian
#4, lydian diminished, leading whole tone, lydian minor,
phrygian dominant, balinese, neopolitan major, harmonic
major, double harmonic major, hungarian minor, hungarian
major, oriental, flamenco, todi raga, persian,
enigmatic, major augmented, lydian #9, messiaen's mode
#4, purvi raga, spanish heptatonic, bebop minor, bebop
major, bebop locrian, minor bebop, ichikosucho, minor
six diminished, half-whole diminished, kafi raga,
messiaen's mode #6, composite blues, messiaen's mode #3,
messiaen's mode #7, chromatic
```

## Playing a Scale Randomly

```
-- TidalCycles
d1
  $ s "superpiano*8"
  # note (scale "hirajoshi" (irand 12))
```

```
// Strudel
$: sound("piano*8")
  .n(irand(12))
  .scale("C:hirajoshi")
```

# Improvising with Scales

```
-- TidalCycles
do
  let mode = "minPent"
  let root = "c"
  d1
    $ jux ((" [10, 5]" / 8) <~)
    $ s "superhammond(7, 8, [0, 5])"
    # voice 4
    # note "c2"
    |+ note (scale mode (irand 24))
    |+ note root
    # room 0.8 # size 0.9
  d2
    $ s "superhammond*16"
    # voice 3
    # note "c5"
    |+ note (scale mode (irand 7))
    |+ note root
    # room 0.8 # size 0.9
```

```
$: sound("supersaw(5, 8)")  
  .n(irand(12))  
  .scale("C1:hirajoshi").jux(rev).room(0.8)  
  .delay(0.7).delaytime(3/8)  
  .lpf(rand.range(400, 2000)).resonance(12)  
$: sound("gm_church_organ*16")  
  .n(irand(19))  
  .scale("C3:hirajoshi").jux(rev).room(0.8)  
  .lpf(rand.range(400, 8000)).resonance(15)  
  .delay(0.7).delaytime(5/8)
```