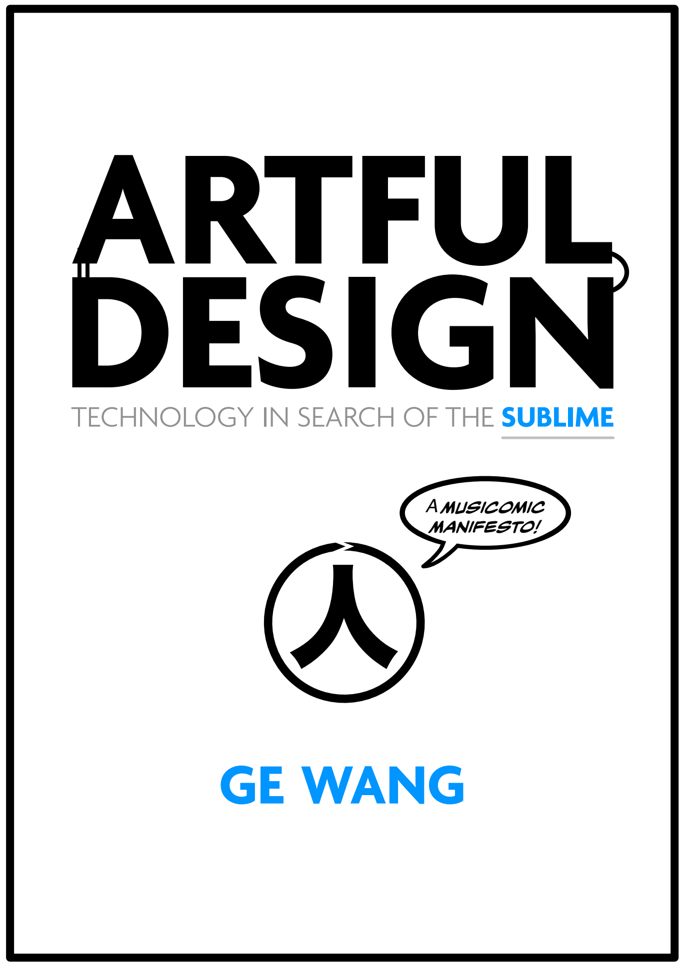
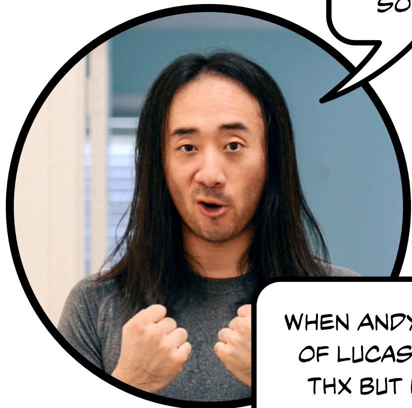
"Design of the THX Deep Note"

excerpt (pp. 176-181) from *Artful Design*,
Chapter 4 "Programability and Sound Design"



https://**artful.design**/

THE THX DEEP NOTE!



TO ILLUSTRATE SOUND SYNTHESIS BY WAY OF PARAMETRIC EVOLUTION, WE ARE GOING TO ***RECREATE*** ONE OF THE MOST RECOGNIZABLE PIECES OF COMPUTER-GENERATED SOUND EVER DESIGNED: *THE* ***THX DEEP NOTE!***

DESIGNED AND PROGRAMMED IN 1982 BY ***JAMES ANDY MOORER*** (ALSO A FOUNDING MEMBER OF CCRMA), THE ***DEEP NOTE*** WAS FIRST INTRODUCED WITH THE 1983 PREMIER OF ***RETURN OF THE JEDI*** AND HAS BEEN HEARD IN COUNTLESS THX TRAILERS FOR MOVIES AND VIDEO GAMES!

WHEN ANDY CREATED THE ***DEEP NOTE***, HE WAS AN EMPLOYEE OF LUCASFILM'S COMPUTER DIVISION (WHICH NOT ONLY LED TO THX BUT EVENTUALLY PIXAR). THX CREATOR ***TOM HOLMAN*** ASKED ANDY TO CREATE A ***SOUND LOGO*** THAT "COMES OUT OF NOWHERE AND GETS ***REALLY, REALLY BIG.***"

IN 1982, IT TOOK ANDY MOORER 325 LINES OF C CODE RUNNING ON A SPECIALIZED HARDWARE AND SOFTWARE ***AUDIO SIGNAL PROCESSOR.*** HERE WE ARE GOING TO RECREATE IT IN ***CHUCK!*** IT WON'T BE EXACTLY THE SAME, BUT WE WILL TRY TO CAPTURE THE ESSENCE OF THE SOUND DESIGN!

THE ***DEEP NOTE*** WAS ***SYNTHESIZED*** USING ***30*** VOICES WITH RANDOMIZED STARTING FREQUENCIES BETWEEN 40HZ TO 350HZ. THESE VOICES SMOOTHLY ***GLIDE*** TOWARD A PREDETERMINED ***CHORD*** SPANNING 6 OCTAVES, OVER A DURATION OF 30 SECONDS.

IT'S A WONDERFUL DEMONSTRATION OF THE POWER OF ***PRECISELY*** CONTROLLING TIME-VARYING AUDIO -- AND USING ***SIMPLE*** BUILDING BLOCKS TO CREATE A ***COMPLEX*** SOUND!

A PLAN...

0 ***SETUP STAGE***: CREATE PROVISIONS FOR 30 VOICES. IN OUR CASE, WE WILL INSTANTIATE 30 SAWTOOTH WAVE GENERATORS, ***RANDOMIZING*** THEIR RESPECTIVE ***STARTING*** FREQUENCIES (OUR EMULATION WILL USE 160-360HZ AS THE STARTING RANGE). EACH VOICE WILL EVENTUALLY REACH ONE OF 9 PREDETERMINED ***TARGET*** FREQUENCIES.

1 ***INITIAL STAGE***: BEGIN THE SOUND BY ***RAMPING*** UP THE VOICES IN ***AMPLITUDE*** (WHILE ***HOLDING*** THE STARTING FREQUENCIES CONSTANT). THE ORIGINAL DEEP NOTE DOES SOMETHING MORE SOPHISTICATED -- WE'LL ONLY APPROXIMATE IT HERE. THE GOAL IS TO CREATE THE PART OF THE SOUND THAT "COMES OUT OF NOWHERE."

2 ***CONVERGING STAGE***: GRADUALLY ***CHANGE*** THE FREQUENCIES OF ALL THE VOICES ***TOWARD*** THEIR RESPECTIVE ***TARGET FREQUENCIES***, ACCOMPLISHED BY UPDATING EACH VOICE'S FREQUENCY EVERY SO OFTEN (EVERY 10::MS), SO THAT IT SMOOTHLY APPROACHES THE TARGET (MUCH LIKE OUR ZENO'S INTERPOLATOR IN CHAPTER 3, EXCEPT THIS INTERPOLATION IS ***LINEAR***). HERE, THE SOUND GETS "REALLY BIG"!

3 ***TARGET STAGE***: ALL VOICES ***REACH*** THEIR TARGET FREQUENCIES AT PRECISELY THE ***SAME TIME***, SOUNDING OUR PREDETERMINED ***CHORD*** AND CREATING AN ***EPIC*** AND UNMISTAKABLE SENSE OF ***ARRIVAL*** AND ***RESOLUTION!*** WE WILL HOLD THIS CHORD BRIEFLY BEFORE FADING OUT.

WE CAN ILLUSTRATE THE PROGRAM ***GRAPHICALLY*** -- 30 ***LINES*** REPRESENT THE ***FREQUENCIES*** OF THE 30 VOICES OVER TIME. OBSERVE THE ***THREE STAGES*** THE SOUND GOES THROUGH!

3 ***"ORDER + RESOLUTION"*** *(AND A* ***BIG CHORD!****)*

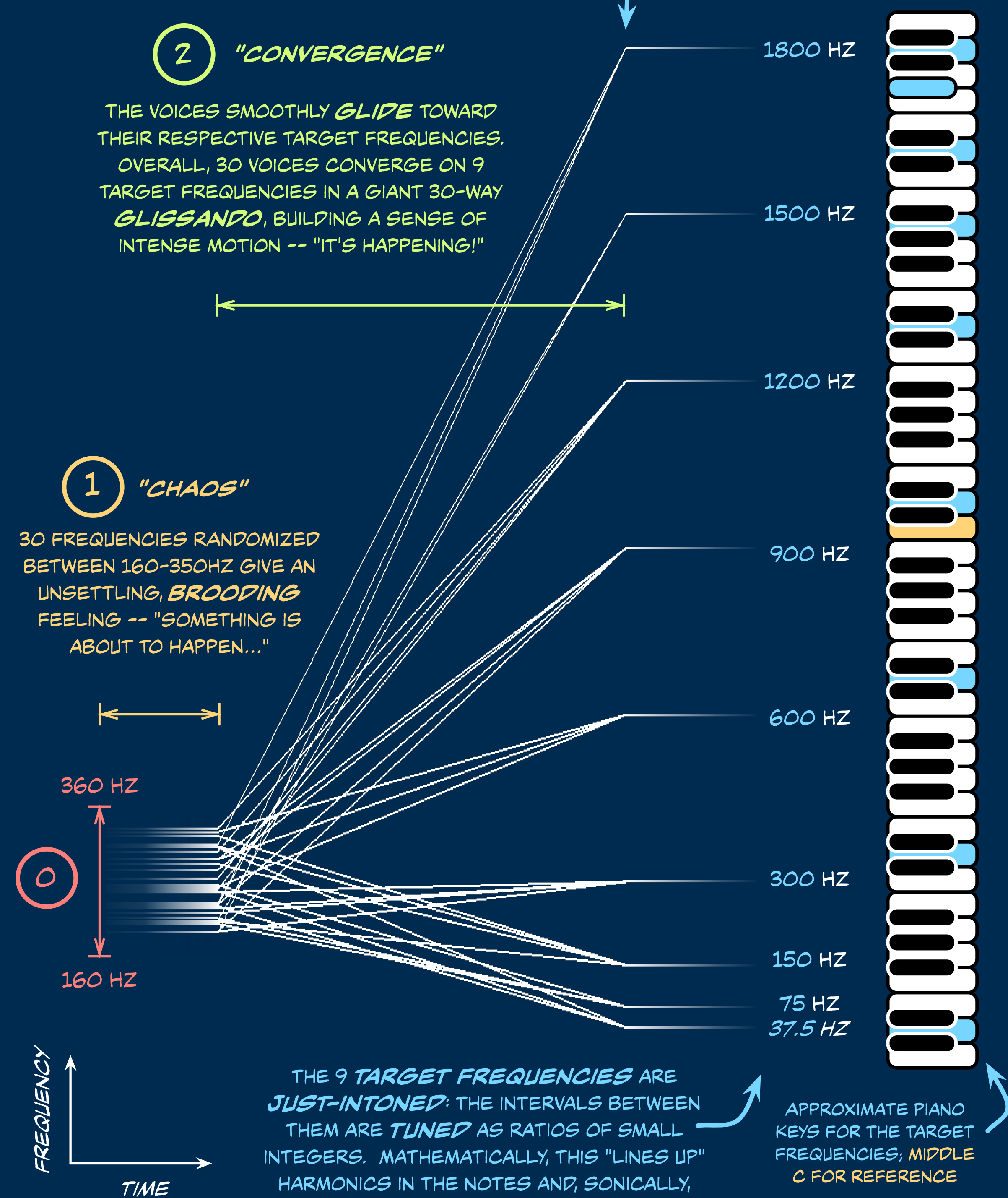
THE TARGET FREQUENCIES STACK UP TO A ***BIG CHORD*** SPANNING MULTIPLE OCTAVES AND GIVING A SENSE OF EPIC ***RESOLUTION*** AND ***ARRIVAL*** -- WHOA.

2 ***"CONVERGENCE"***

THE VOICES SMOOTHLY ***GLIDE*** TOWARD THEIR RESPECTIVE TARGET FREQUENCIES. OVERALL, 30 VOICES CONVERGE ON 9 TARGET FREQUENCIES IN A GIANT 30-WAY ***GLISSANDO***, BUILDING A SENSE OF INTENSE MOTION -- "IT'S HAPPENING!"

1 ***"CHAOS"***

30 FREQUENCIES RANDOMIZED BETWEEN 160-350HZ GIVE AN UNSETTLING, ***BROODING*** FEELING -- "SOMETHING IS ABOUT TO HAPPEN..."



THE 9 ***TARGET FREQUENCIES*** ARE ***JUST-INTONED***: THE INTERVALS BETWEEN THEM ARE ***TUNED*** AS RATIOS OF SMALL INTEGERS. MATHEMATICALLY, THIS "LINES UP" HARMONICS IN THE NOTES AND, SONICALLY, RESULTS IN A BIG, STABLE, AND PURE SOUND.

APPROXIMATE PIANO KEYS FOR THE TARGET FREQUENCIES; MIDDLE C FOR REFERENCE

9 ***TARGET FREQUENCIES***
WE ASSOCIATE EACH OF 30 VOICES WITH ONE OF THESE

SETUP

0

```
// D1,  D2, D3,  D4,  D5,  A5,  D6,   F#6   A6
[ 37.5, 75, 150, 300, 600, 900, 1200, 1500, 1800,
  37.5, 75, 150, 300, 600, 900, 1200, 1500, 1800,
  37.5, 75, 150, 300, 600, 900, 1200,       1800,
            150, 300,       900, 1200
] @=> float targets[];

float initials[30];
3.0::second => dur CHAOS_HOLD_TIME;
5.5::second => dur CONVERGENCE_TIME;
3.5::second => dur TARGET_HOLD_TIME;
2.0::second => dur DECAY_TIME;

SawOsc saw[30];
Gain gainL[30];
Gain gainR[30];
NRev reverbL => dac.left;
NRev reverbR => dac.right;
0.075 => reverbL.mix => reverbR.mix;

for( 0 => int i; i < 30; i++ )
{
    saw[i] => gainL[i] => reverbL;
    saw[i] => gainR[i] => reverbR;
    1.0 - gainL[i].gain() => gainR[i].gain;
    0.1 => saw[i].gain;
    Math.random2f( 160, 360 ) => initials[i] => saw[i].freq;
    Math.random2f( 0.0, 1.0 ) => gainL[i].gain;
}
```

DURATION FOR VARIOUS STAGES

MAKE ***30*** SAWTOOTH GENERATORS AS OUR VOICES, ALONG WITH ADDITIONAL SOUND OBJECTS FOR SIGNAL ROUTING

RANDOMIZE INITIAL FREQUENCIES AND ***PAN*** EACH SAWTOOTH (ALSO RANDOMLY) IN THE STEREO FIELD

FOR ***REFERENCE***, THIS IS OUR DEEP NOTE EMULATION ***ALGORITHM*** AS A ***CHUCK*** PROGRAM, IN FOUR SECTIONS CORRESPONDING TO OUR INITIAL PLAN.

DON'T WORRY IF YOUR EYES START ***WATERING*** FROM LOOKING AT THIS CODE -- THIS IS JUST TO GIVE A GENERAL IDEA OF HOW WE CAN USE CODE TO CONTROL SOUND OVER TIME.

"CHAOS"

1

```
now + CHAOS_HOLD_TIME => time end;
while( now < end )
{
    1 - (end-now) / CHAOS_HOLD_TIME => float progress;
    for( 0 => int i; i < 30; i++ ) {
        0.1 * Math.pow(progress,3) => saw[i].gain;
    }
    10::ms => now;
}
```

RAMP UP VOLUME FOR EACH VOICE WHILE HOLDING ITS INITIAL FREQUENCY.

"CONVERGENCE"

2

```
now + CONVERGENCE_TIME => end;
while( now < end )
{
    1 - (end-now)/CONVERGENCE_TIME => float progress;
    for( 0 => int i; i < 30; i++ ) {
        initials[i] + (targets[i]-initials[i])*progress
            => saw[i].freq;
    }
    10::ms => now;
}
```

IN SMALL TIME INCREMENTS (10::MS) ***UPDATE*** FREQUENCIES TO APPROACH TARGETS SMOOTHLY!

VOICES ***ARRIVE*** AT THE ***TARGET*** FREQUENCIES SIMULTANEOUSLY; ***HOLD*** THE RESULTING CHORD.

"RESOLUTION"

3

```
TARGET_HOLD_TIME => now; // hold the chord!

now + DECAY_TIME => end;
while( now < end )
{
    (end-now) / DECAY_TIME => float progress;
    for( 0 => int i; i < 30; i++ ) {
        0.1 * progress => saw[i].gain; // fade
    }
    10::ms => now;
}
```

FADE TO SILENCE.

THERE ARE SEVERAL PROGRAMMABILITY AND DESIGN IDEAS ***IN MOTION*** HERE, INCLUDING ***PRECISION*** OF CONTROL, SONIC ***NARRATIVE***, AND ***STRENGTH*** IN ***NUMBERS*** OF SIMPLE ELEMENTS ACTING TOGETHER TO CULMINATE IN A SINGLE PRONOUNCED EFFECT.

PRINCIPLE 4.3

*BUILD COMPLEXITY AS THE **SUM** OF **SIMPLE** ELEMENTS*

AN AUDIO-SPECIFIC VERSION OF VISUAL DESIGN PRINCIPLE 3.5: BUILD COMPLEXITY FROM SIMPLICITY

COMPUTERS ARE REALLY GOOD AT ***MAKING COPIES.*** ONCE WE CAN PROGRAM ***ONE*** THING, IT'S TRIVIAL TO INSTANTIATE ***MORE*** OF IT. THE AIM IS NOT MERELY TO HAVE MORE, BUT TO CREATE SOMETHING ***NEW*** IN THE ***AMALGAM.***

FOR EXAMPLE, OUR ***DEEP NOTE*** EMULATION IS ACHIEVED THROUGH THE ADDITION OF 30 BASIC SAWTOOTH VOICES, MODULATING THEIR FREQUENCIES IN A SPECIFIC AND SYNCHRONIZED WAY. THIS PRODUCES THE SENSE OF A SINGLE, COHERENT SOUND! WE MIGHT STILL HEAR INDIVIDUAL VOICES IN THE MIX, BUT WE ALSO HEAR THE SUM TOTAL OF THE VOICES AS A ***CULMINATING,*** COHESIVE SOUND.

THE KEY HERE IS NOT ONLY THAT WE HAVE ***MANY*** VOICES, BUT THAT EACH ONE IS BOTH ***INDEPENDENTLY*** CHANGING IN FREQUENCY AND ***GLOBALLY COORDINATED*** WITH THE OTHER VOICES.

*TWO **KEY COMPONENTS** IN CREATING COMPLEXITY FROM SIMPLE ELEMENTS*

REMEMBER THIS FROM CHAPTER 3? ***ONE*** FLARE MULTIPLIED BY ***500***, ARRANGED IN A ***SHIMMERING STREAM***, WHERE ***EACH*** FLARE TWINKLES AND OSCILLATES ***INDEPENDENTLY***...

*IT'S AS IF THE SYSTEM HAS A **HIVE MIND** THAT GLOBALLY CONTROLS ALL THE ELEMENTS, BUT EACH ELEMENT IS ALSO LOCALLY FREE TO ACT INDEPENDENTLY WITHIN SPECIFIC RULES.*

THIS CASE STUDY ALSO REINFORCES PERHAPS THE MOST IMPORTANT ***ETHOS*** IN THIS CHAPTER...

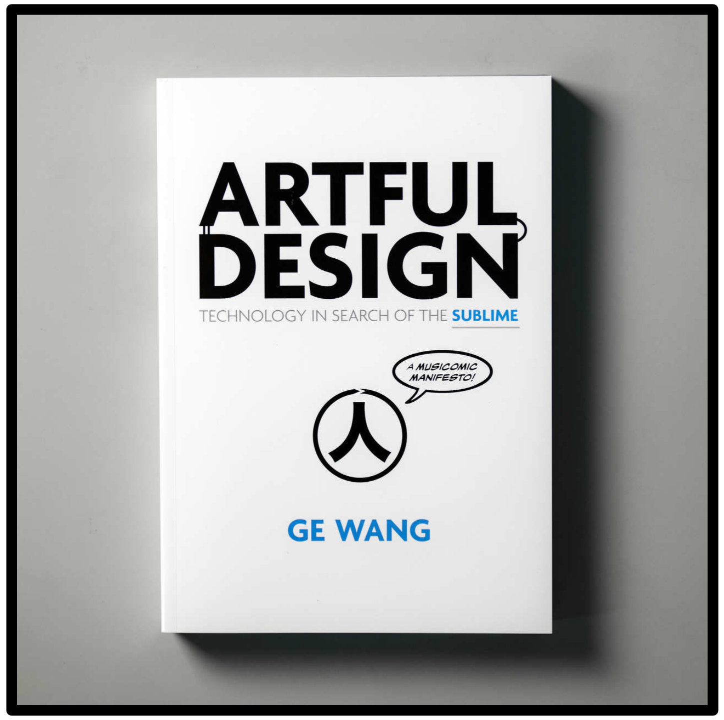
PRINCIPLE 4.5

*DESIGN THINGS **WITH** A COMPUTER THAT WOULD **NOT** BE POSSIBLE **WITHOUT!***

DO NOT SIMPLY COPY, PORT, DIGITIZE, OR EMULATE. RATHER, CREATE SOMETHING NOVEL AND UNIQUE TO THE ***MEDIUM*** -- SOMETHING THAT ***COULD NOT EXIST*** WITHOUT IT.

IT'S ***TEMPTING*** TO ***REMAKE*** WHAT ALREADY EXISTS. WHILE THAT REMAINS A USEFUL EXERCISE, MANY PEOPLE DO THAT BECAUSE IT'S ***OBVIOUS.*** BUT WITH NEW TECHNOLOGICAL MEDIUMS ALSO COME THE OPPORTUNITY AND ***RESPONSIBILITY*** TO DISCOVER WHAT THE MEDIUM IS ***INNATELY*** GOOD AT. ***DESIGN TO THE MEDIUM!***

THIS IS AN ESSENTIAL ***GUIDING PRINCIPLE*** OF ARTFUL DESIGN (WITH ***ANY*** MEDIUM OR TECHNOLOGY). LET'S APPLY THIS LENS AND DECONSTRUCT A COMPUTER MUSIC COMPOSITION -- ONE THAT USES THE COMPUTER AS A KIND OF ***PERSONAL MUSICAL FILTER*** TO THE WORLD.



https://**artful.design**/