GAME MUSIC COMPOSER

REVIEW TEMPO, METER, HARMONICS

Creative Track Emphasis

Junior Dev League

DAY:THREE

Background <=>Mechanics
Adaptations
Synthesis

ONE DAY

GAME MUSIC COMPOSER

REVIEW 10-20 MINUTES
CONCEPTS| FORM AND STRUCTURE 10-15 MINUTES
DEVELOPING SKILLS PRACTICE 50 MINUTES
SHORT BREAK 10 MINUTES
MEETING WITH DEVELOPER 15 MINUTES @11:00 AM
PROJECT PLANNING MEETING WITH INSTRUCTOR 10-15 MINUTES
PRACTICE EXERCISE IN SKILLS WITH SLIGHT VARIATIONS 30 MINUTES
REVIEW CONCEPTS | HOMEWORK 10-20 MINUTES @11:30 AM

Github

https://github.com/junior-devleague/GameProductionProgram/tree/master/GameMusicComposition/Week3

http://ciocan.github.io/angular-wheel-rhythm/

REVIEW

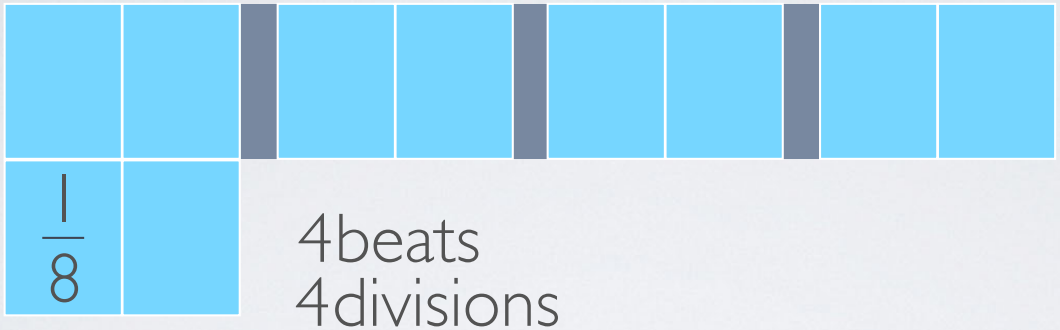

IBAR

1 BAR

4divisions

1 BAR

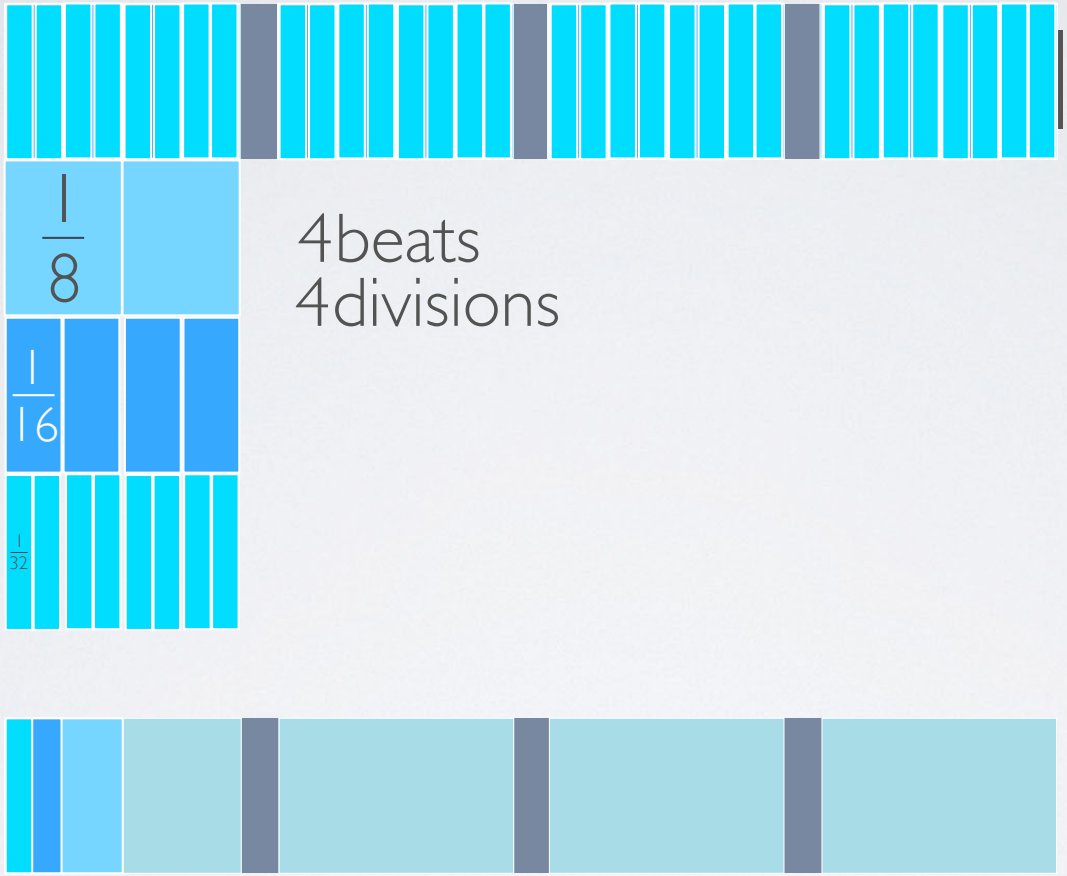
4beats
4divisions



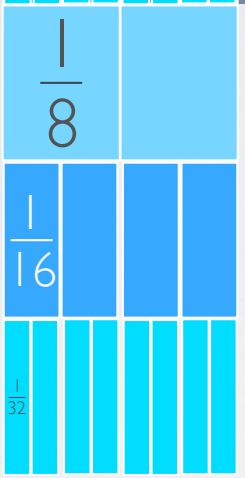
1 BAR

4beats
4divisions


1 BAR
1/8
1/16
4beats
4divisions


1 BAR
1/8
4beats
4divisions
1/16
1/32

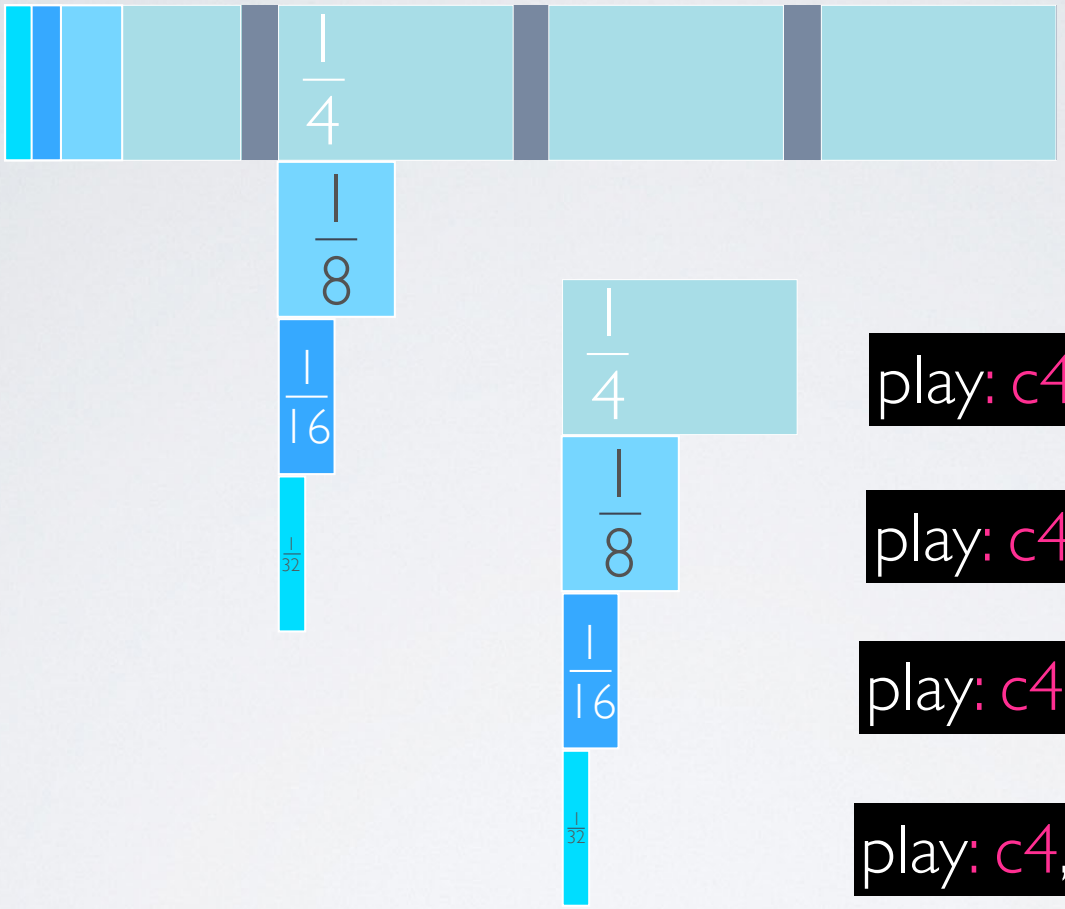
1 BAR



4beats Tempo bpm : heart rate
4divisions 120 beats : 120 quarter notes per minute

$$\frac{120 \text{ beats}}{60 \text{ seconds}} = \frac{2 \text{ beats}}{1 \text{ seconds}}$$

2 beats per second
1 beats per 0.5 second
1 qtr. note per 0.5second



2 beats per second

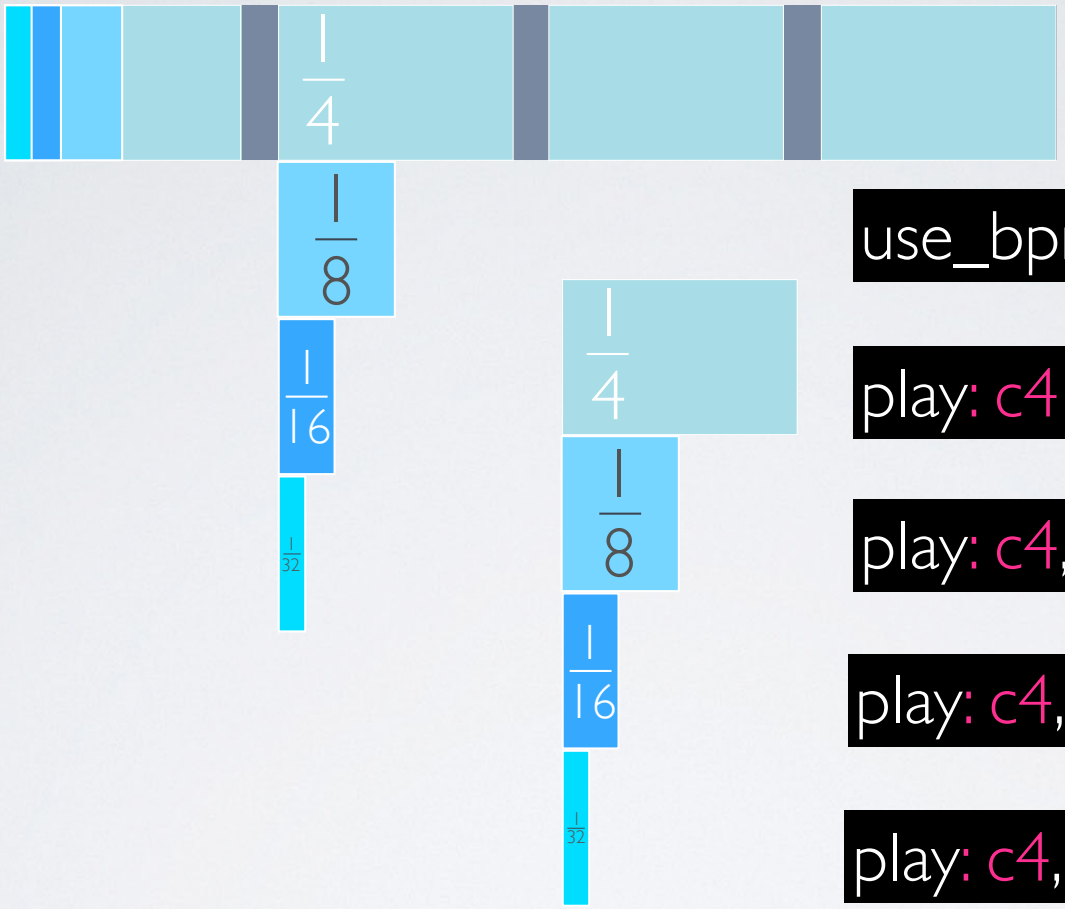
1 BAR : MEASURE

play: c4, release: 0.5

play: c4, release: 0.25

play: c4, release: 0.125

play: c4, release: 0.0625



2 beats per second

1 BAR : MEASURE

```
use_bpm 120
```

```
play: c4
```

```
play: c4,  release: 0.5
```

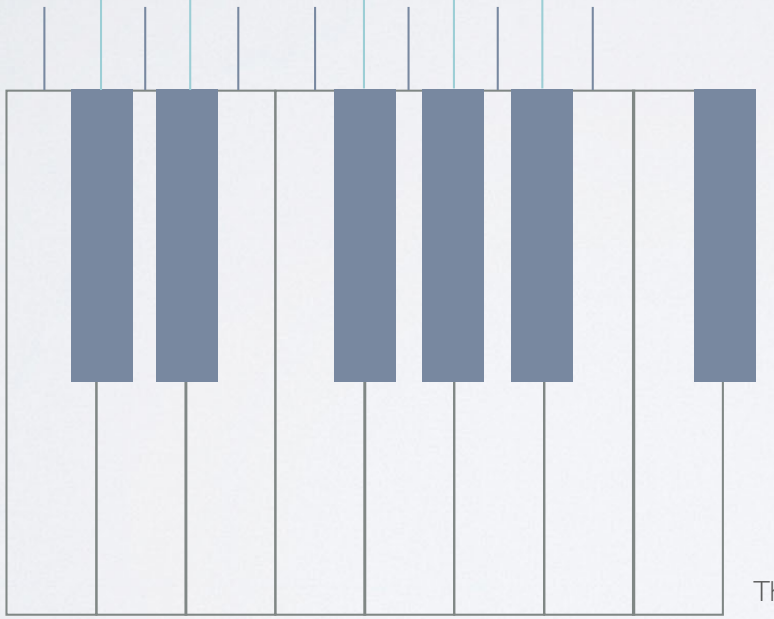
```
play: c4,  release: 0.25
```

```
play: c4,  release: 0.125
```

	120	121	122	123	124	125	126	127					
8	108	109	110	111	112	113	114	115	116	117	118	119	
7	96	97	98	99	100	101	102	103	104	105	106	107	Octave
6	84	85	86	87	88	89	90	91	92	93	94	95	
5	72	73	74	75	76	77	78	79	80	81	82	83	
4	60	61	62	63	64	65	66	67	68	69	70	71	
3	48	49	50	51	52	53	54	55	56	57	58	59	
2	36	37	38	39	40	41	42	43	44	45	46	47	
1	24	25	26	27	28	29	30	31	32	33	34	35	
0	12	13	14	15	16	17	18	19	20	21	22	23	
-1	0	1	2	3	4	5	6	7	8	9	10	11	

MIDI no.

C D E F G A B



The numbers refer to note at different octave range and pitch

12 pitches

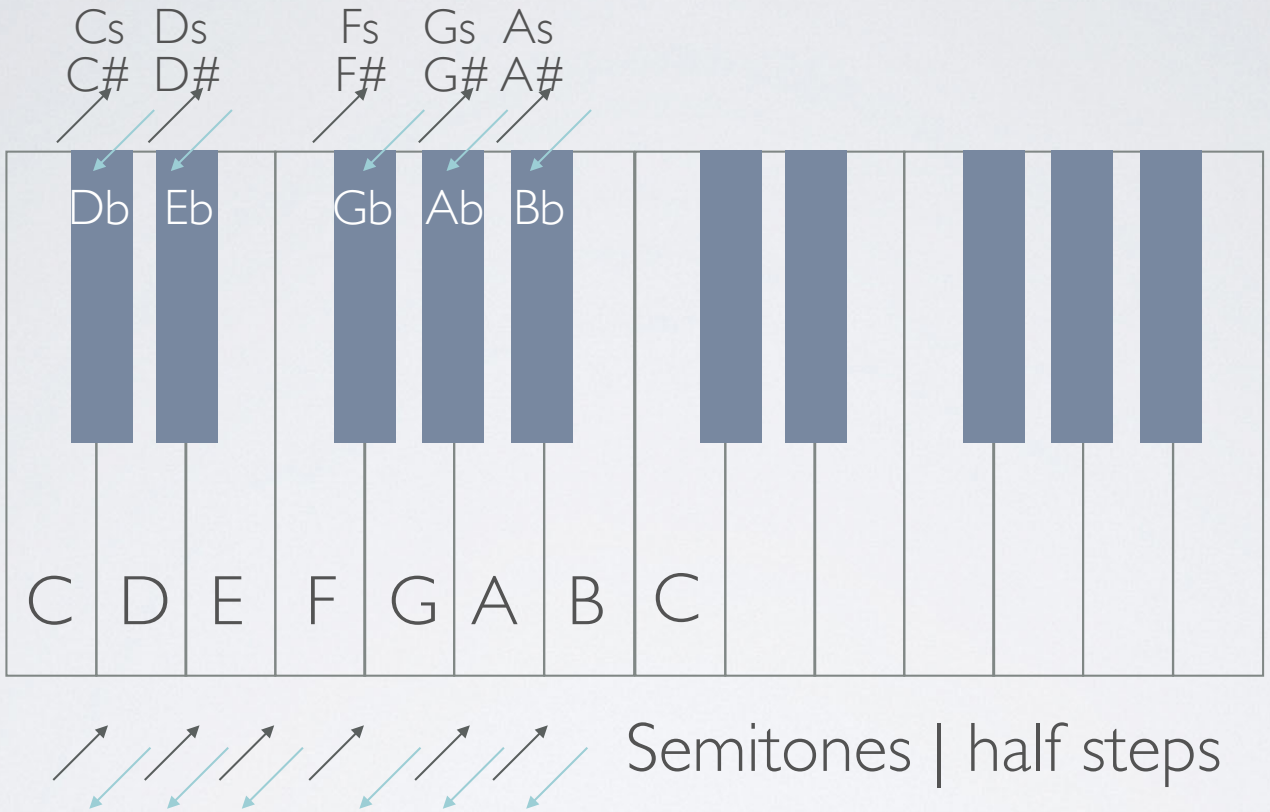
Basic MIDI Chart

These are the numbers translated in Sonic PI when you type pitches

HARMONY

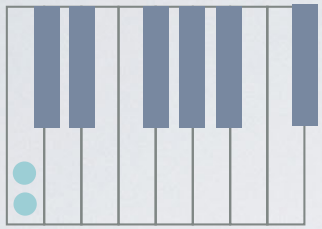
AN INTERVAL IS…

- An interval is the difference of (at minimal) two pitches.

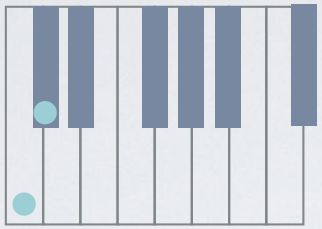


Semitones | half steps

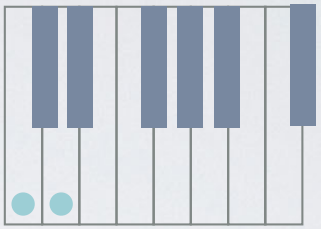
Identify a whole step



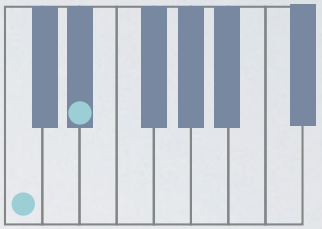
prime/unison



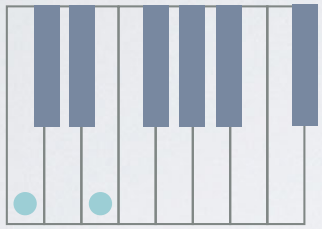
minor second



major second

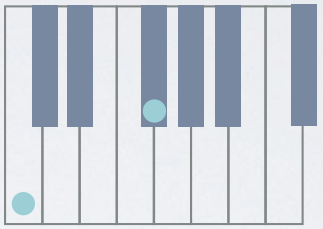


minor third

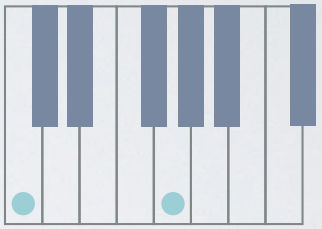


major third

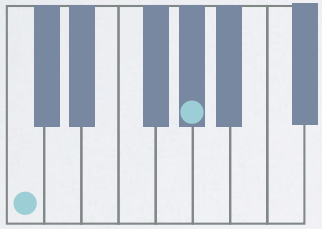
perfect fourth



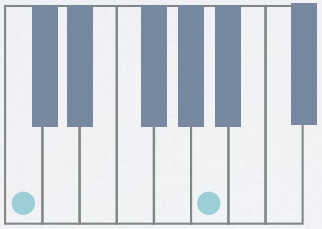
augmented fourth



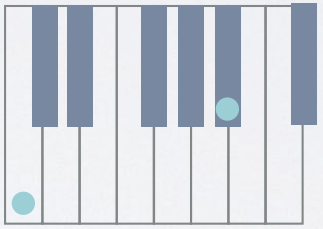
perfect fifth



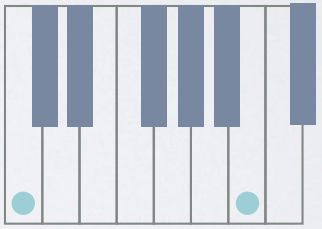
minor sixth



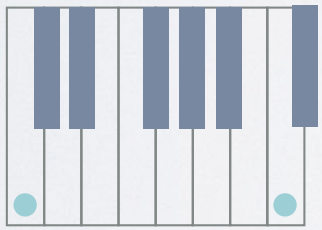
major sixth



minor seventh



major seventh



octave

13 interval harmonies in the octave via C note

Exercise| Translate this to sequence steps

http://onlinesequencer.net/

13 intervals | Programming Functions

Now let's learn to name your blocks so that there is no need to retype the two notes.

```
1   ##| unison/prime
2   play :C4
3   play :C4
4   sleep 1
```

-Must remove ":"
-Convert the notes to MIDI numbers
-Name the musical block in "lowercase"

```
1   define :unison do
2     play 60
3     play 60
4     sleep 1
5   end
```

```
66
67
68  unison
69  min2
70  maj2
71  min3
72  maj3
73  perfect4
74  aug4
75  perfect5
76  min6
77  maj6
78  min7
79  maj7
80  octave
81
```

Now you can play each interval with one line

```
1   ##| unison/prime
2   play :C4
3   play :C4
4   sleep 1
5   ##| minor second/m2
6   play :C4
7   play :Cs4
8   sleep 1
9   ##| major second /M2
10  play :C4
11  play :D4
12  sleep 1
13  ##| minor third /m3
14  play :C4
15  play :Ds4
16  sleep 1
17  ##| major third/ M3
18  play :C4
19  play :E4
20  sleep 1
21  ##| perfect fourth /P4
22  play :C4
23  play :F4
24  sleep 1
25  ##| augmented fourth/ A4 (interval)
26  play :C4
27  play :Fs4
28  sleep 1
29  ##| perfect fifth
30  play :C4
31  play :G4
32  sleep 1
33  ##| minor sixth / m6
34  play :C4
35  play :Gs4
36  sleep 1
37  ##| Major sixth / M6
38  play :C4
39  play :A4
40  sleep 1
41  ##| minor seventh / m7
42  play :C4
43  play :As4
44  sleep 1
45  ##| Major seventh / M7
46  play :C4
47  play :B4
48  sleep 1
49  ##| Octave
50  play :C4
51  play :C5
52  sleep 1
```

Let's randomize the 13 intervals

The interval functions [unison, min2, maj2, min3, maj3,perfect4,aug4,perfect5,min6, maj6, min7, maj7,octave]

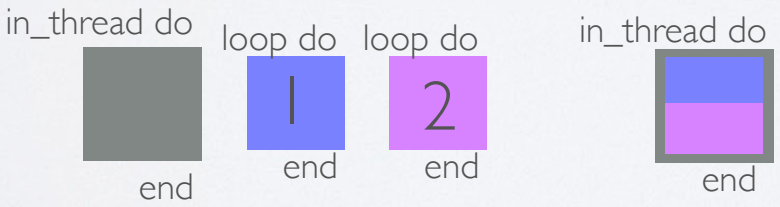
is not recognised as symbols 'c4' or MIDI notes 60

Therefore, the array will work.

So how do we randomize the 13 intervals in short code?

We will use the concept of Thread and the function 'rrand.'

Thread will sync two code blocks at a time.



As long as 'sleep' for
code block 1=2,
it won't fall out of sync

function ‘rrand’

rrand(minimal value, maximum value)

13 intervals consist of the pairing of C4 to each note of C4 scale + C5

What is the range in MIDI notes? 60 to 72

How do you randomize the 13 intervals?

```
1   in_thread do
2     use_synth :piano
3     loop do
4       play rrand_i(60, 72)
5       sleep 0.5
6     end
7   end
8
9   loop do
10    play 60
11    sleep 0.5
12  end
```

Exercise | Sequence|Code in SonicPI

1. Unison
2. Octave
3. Perfect fifth
4. Perfect fourth
5. Major third
6. Minor sixth
7. Minor third
8. Major sixth

Perfect to imperfect consonance

1. Major second

mild dissonance

2. Minor seventh

---

3. Minor second

severe dissonance

4. Major Seventh
5. Augmented fourth

Exercise | Sequence|Code in SonicPI

1. Unison
2. Octave
3. Perfect fifth
4. Perfect fourth
5. Major third
6. Minor sixth
7. Minor third
8. Major sixth

Perfect to imperfect consonance

1. Major second

mild dissonance

2. Minor seventh

---

3. Minor second

severe dissonance

4. Major Seventh
5. Augmented fourth

Exercise | Sequence|Code in SonicPI

1. Prime or unison
2. Octave
3. Perfect fifth
4. Perfect fourth
5. Major third
6. Minor sixth
7. Minor third
8. Major sixth
9. Major second
10. Minor seventh
11. Minor second
12. Major seventh
13. Augmented fourth/diminished fifth

Graduated spectrum of consonant to dissonant intervals

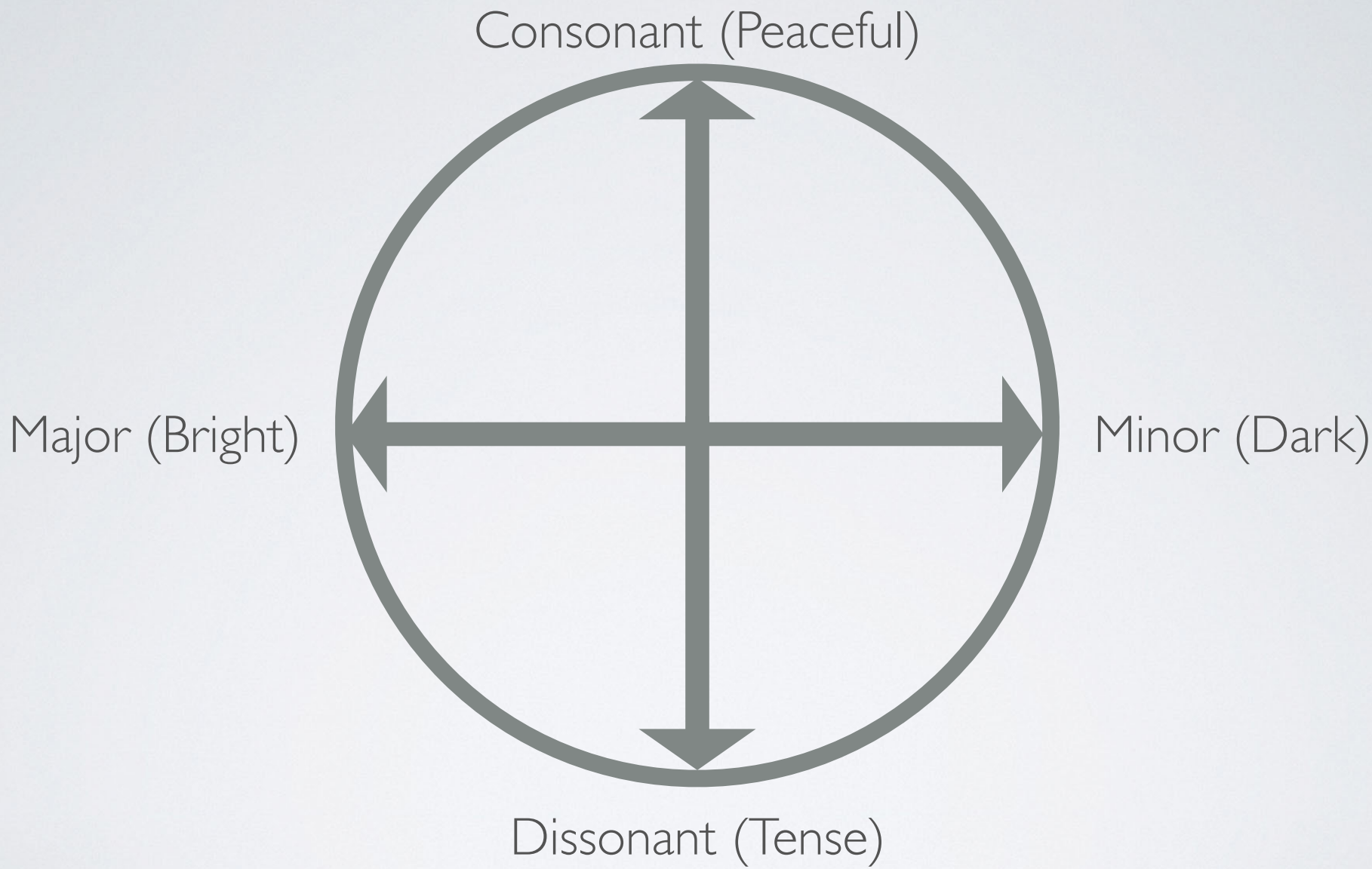

Consonant (Peaceful)
Major (Bright)
Minor (Dark)
Dissonant (Tense)

Table 2.1 Perceived Correlation between Emotions and Interval Harmonies

Interval	Notes	Emotion
Unison	C/C	Pleasing, peace
Octave	C/C	Pleasing, peace
Perfect fifth	C/G	Joy, triumph, courage
Perfect fourth	C/F	Excitement, contentment
Major third	C/E	Harmony, peace, joy
Minor sixth	C/Ab	Harshness, meanness, confusion
Minor third	C/Eb	Sadness, sorrow, annoyance, gloominess
Major sixth	C/A	Sweet, enjoyable, pleasing
Major second	C/D	Sadness, strangeness, tension
Minor seventh	C/Bb	Sadness, dismay, sorrow
Minor second	C/Db	Harshness, sinister, confusion, shock
Major seventh	C/B	Surprise, suspicion
Augmented fourth	C/F#	Suspense, shock, sorrow

Alex Carlin, Music Producer

The Interval. Harmony for Computer Musicians

We focused on
C major scale.
Here are the variations

(scale :C, :diatonic)
(scale :C, :ionian)
(scale :C, :major)
(scale :C, :dorian)
(scale :C, :phrygian)
(scale :C, :lydian)
(scale :C, :mixolydian)
(scale :C, :aeolian)
(scale :C, :minor)
(scale :C, :locrian)
(scale :C, :hex_major6)
(scale :C, :hex_dorian)
(scale :C, :hex_phrygian)
(scale :C, :hex_major7)
(scale :C, :hex_sus)
(scale :C, :hex_aeolian)
(scale :C, :minor_pentatonic)
(scale :C, :yu)
(scale :C, :major_pentatonic)
(scale :C, :gong)
(scale :C, :egyptian)
(scale :C, :shang)
(scale :C, :jiao)
(scale :C, :zhi)
(scale :C, :ritusen)
(scale :C, :whole_tone)
(scale :C, :whole)
(scale :C, :chromatic)
(scale :C, :harmonic_minor)
(scale :C, :melodic_minor_asc)
(scale :C, :hungarian_minor)
(scale :C, :leading_whole)
(scale :C, :augmented)
(scale :C, :purvi)
(scale :C, :chinese)
(scale :C, :lydian_minor)

(scale :C, :octatonic)
(scale :C, :messiaen1)
(scale :C, :messiaen2)
(scale :C, :messiaen3)
(scale :C, :messiaen4)
(scale :C, :messiaen5)
(scale :C, :messiaen6)
(scale :C, :messiaen7)
(scale :C, :super_locrian)
(scale :C, :hirajoshi)
(scale :C, :kumoi)
(scale :C, :neapolitan_major)
(scale :C, :bartok)
(scale :C, :bhairav)
(scale :C, :locrian_major)
(scale :C, :ahirbhairav)
(scale :C, :enigmatic)
(scale :C, :neapolitan_minor)
(scale :C, :pelog)
(scale :C, :augmented2)
(scale :C, :scriabin)
(scale :C, :harmonic_major)
(scale :C, :melodic_minor_desc)
(scale :C, :romanian_minor)
(scale :C, :hindu)
(scale :C, :iwato)
(scale :C, :melodic_minor)
(scale :C, :diminished2)
(scale :C, :marva)
(scale :C, :melodic_major)
(scale :C, :indian)
(scale :C, :spanish)
(scale :C, :prometheus)
(scale :C, :diminished)
(scale :C, :todi)

HOMEWORK

- What will you compose for the game?
- Practice defining functions. In what situation would you define your "musical code block"?
- Use Table 2.1 on slide 27 to create some melodies. Mix it up with single notes and triads.
    - Use Thread concept.
    - Use rrand fn.

CHALLENGE

How do you play the 13 interval randomisation code in different C scale on slide 28?

```
1   in_thread do
2     use_synth :piano
3     loop do
4       play rrand_i(60, 72)
5       sleep 0.5
6     end
7   end
8
9   loop do
10    play 60
11    sleep 0.5
12  end
```

Communicate with your fellow Game Music Composers. ^_^

@aisis or email: aisis@devleague.com

Links to codes

https://github.com/junior-devleague/GameProductionProgram/tree/master/GameMusicComposition/Week3/Aisis