September 18th PLT Group Meeting

GitHub

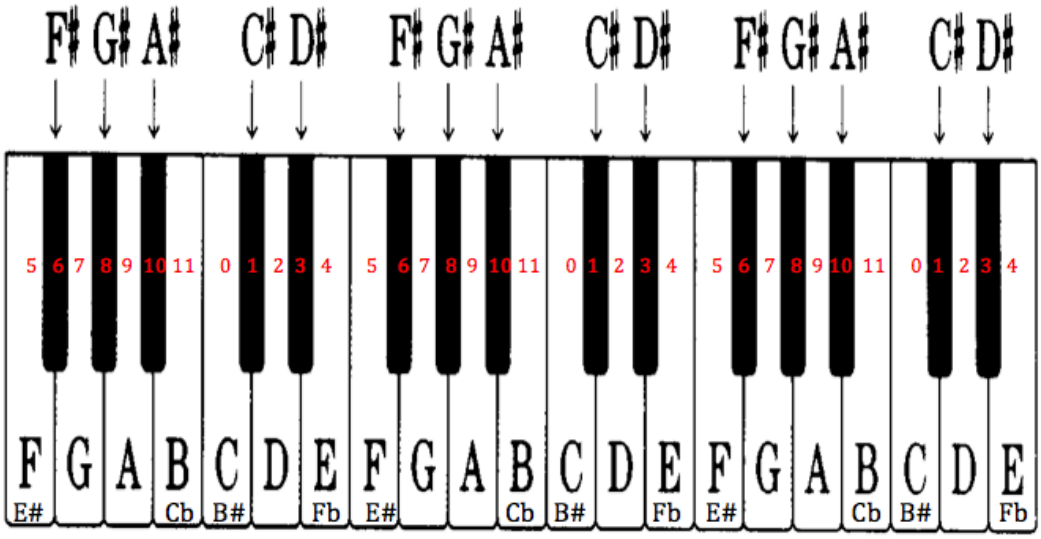
- **ACTION**: make own branch, make proposal file
- Look at http://gitimmersion.com/ for tutorial if needed
    - Basic commands: add, commit, push

Proposal Action Items

- Background: **Richard**
    - What is serialism
    - Notation
- Motivation: **Lianne**
    - Why functional language
    - Why compiling to C
    - How helpful for composers
- Language Description: **Lindsay**
    - Types (static, immutable)
    - Operators
    - Keywords
    - Functions
- Examples: **Van**
    - Library functions (transposition, etc)
    - Writes music (with C output): Take Figure and "show"/compose it

Serialism

Pitch classes (pc):
(Original image: http://www.music-mind.com/Music/Srm0038.GIF)



- Creating the matrix
  - Matrix is a tool for composers: pick and choose things, then compose
  - P_#
    - P_0 = prime row, only thing composer needs to decide
    - # subscript !=0, the subscript indicates the number of transpositions to do on the original row
      - Example: P_10 takes every pc in P_0 and adds 10 to it
  - I_# = inverse
    - # subscript is inverse of whatever P row that # corresponds to
  - R_# = retrograde
  - RI_# = inverse retrograde

Example

1. Pick P_0
2. Compute I_0: difference between P_0's 1st entry and current entry, inverted and mod(12)
3. Compute P subscript numbers for all other rows
4. Take each P_0 entry number add P subscript to it
5. Match I subscripts with P row subscripts

	I_0	-3 I_3	+7 I_5	+1 I_11	+2 I_10	+3 I_9	
P_0	3	0	10	4	5	6	R_0
+3 P_3	6	3	1	7	8	9	R_3
-7 P_5	8						R_5
-1 P_11	2						R_11
-2 P_10	1						R_10
-3 P_9	0						R_9
	RI_0	RI_3	RI_5	RI_11	RI_10	RI_9	

Proposal Discussion

- Functional Language → Compiler → C & OpenGL → gcc → .o file (music score)
  - Pro: no one has done this before
  - Pro: programmable to do serialism
  - Con: might be hard if we want to add rhythm
    - Start with whole notes only, incrementally add other note lengths
- Specifics
  - Immutable memory – no global variables, just types
  - No I/O: initial in (prime row), ultimate out (image of score)
  - Haskell syntax is good jumping off point (see tutorials if needed)
- Keywords
  - let (rec?)
  - ->
  - :: specify type (ex. retro :: [int] -> … -> [int] -> int)
    - last [int] = return type
    - all other [int] =arguments
  - if, else, then (else required)

- | (pipe for guards)

- Types
  - Primative: int, boolean
  - Note, Chord, Figure
    - Note: (pc: int, beat: int, register: int)
      - pc = 0-11
      - beat = powers of 2
        - using 4:4 timing (4 beats in measure, each beat gets ¼ note)
        - 1 – whole note
        - 2 – half note
        - 4 – quarter note
        - 8 – eighth note
        - 16 – sixteenth note
        - 32 – thirty second note, highest we will allow
      - register = -2 – 2
        - middle C and above = positive numbers (treble clef)
        - below middle C = negative numbers (base clef)
    - Chord: [Note] – need type check that notes all have same beat number
    - Figure: [Chord]
  - **OR:** Lists and Tuples and write std library with note, chord, figure
    - All elements of list must have same type
    - Elements of tuple can be different types
- Operators
  - + plus, - minus, ++ concat, : cons
  - % mod (make sure handles negatives)
  - /* */ nested comments, // in-line comments
  - <, >, <=, >=, == comparisons
- Functions
  - Declaration
    - Must declare type
    - Can declare general type
    - Use "->" to specify arguments + return type
    - Declaring a function must be on own line
  - Can use pattern matching, guards, or if-then-else
    - Pattern Matching: each pattern must be on own line

      ```
      retro [] = []
      retro (x : rest) = (retro rest) ++ [x]
      ```
    - Guards: new line vs no new line doesn't matter

      ```
      retro list
      | (null list) = []
      | otherwise = (retro (tail list)) ++ [(head list)]
      ```

        - if-else
            retro list
            if (null list) then []
            else (retro (tail list)) ++ [(head list)]
- end of line = \n
    - need to account for Windows? \r\n