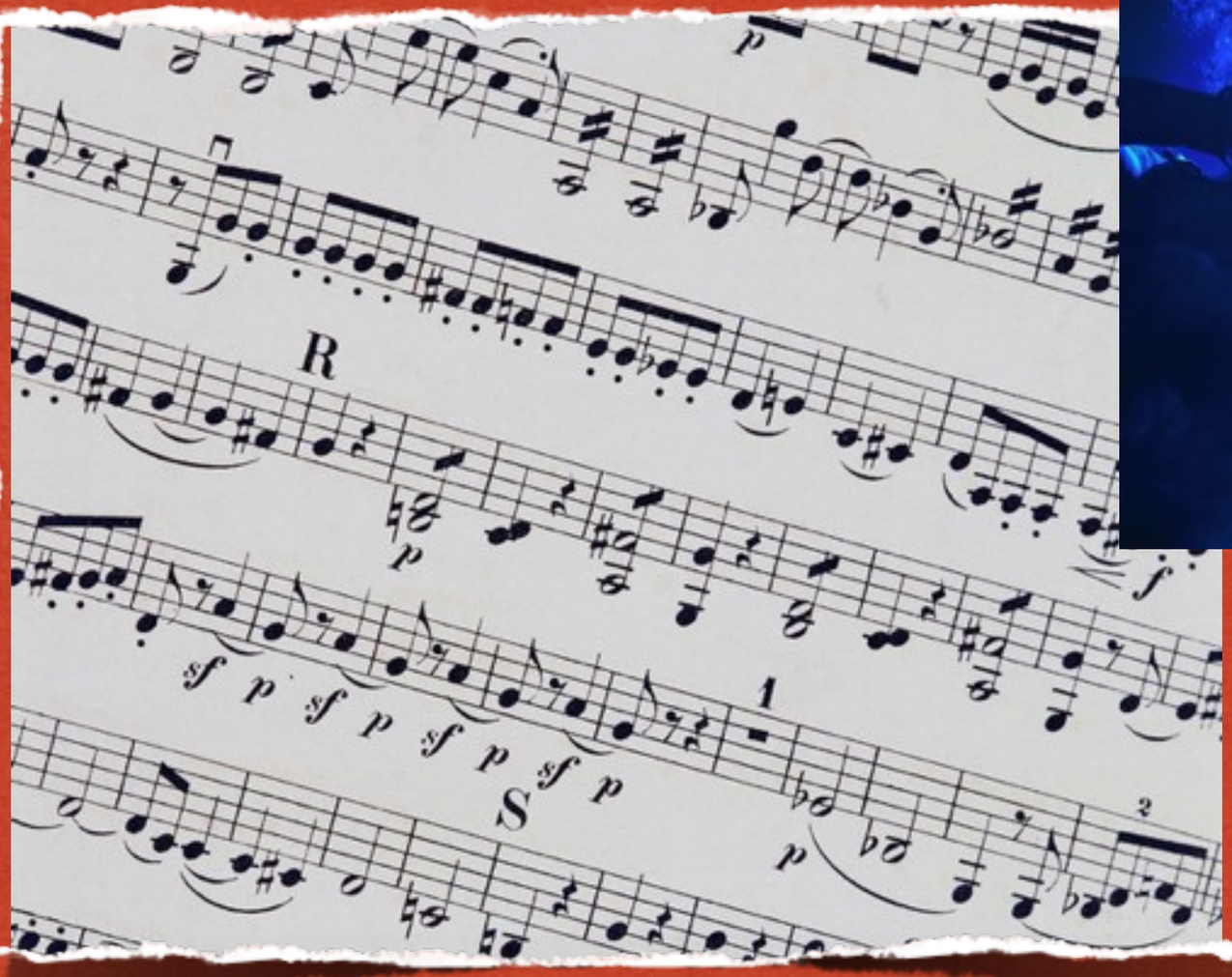
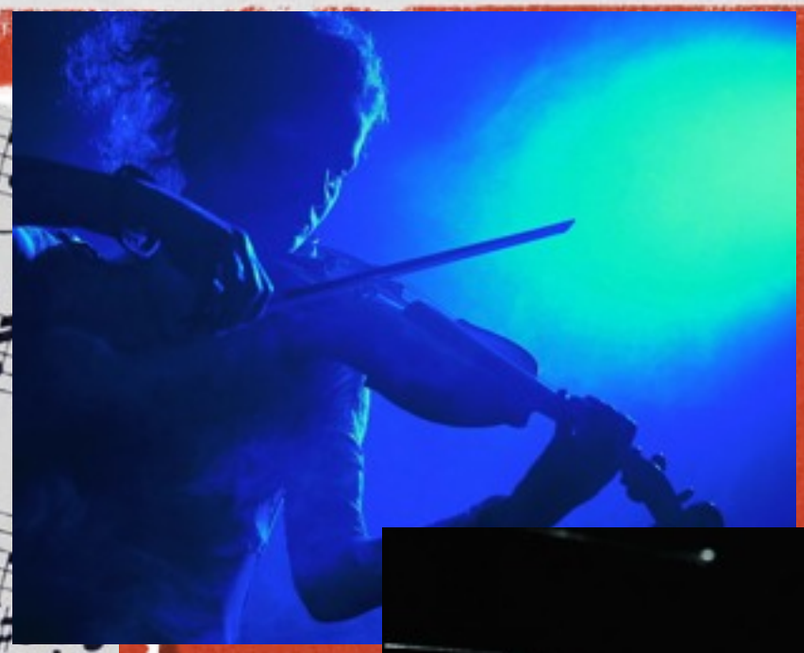


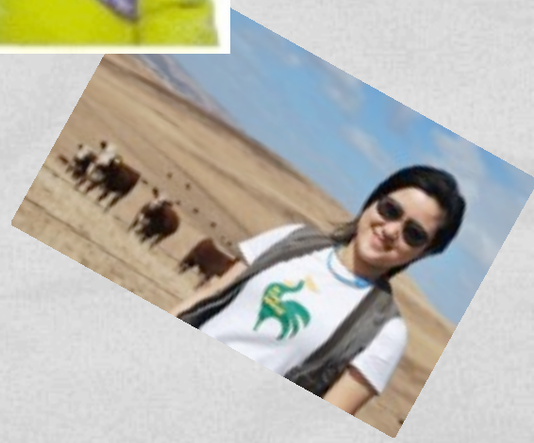


MFAN

A PROGRAMMING LANGUAGE FOR WRITING MUSIC

TEAM MEMBER

- Weiqi Ma (wm2244@columbia.edu)
- Guangwei Zhang (gz2155@columbia.edu)
- Chen Qian (cq2137@columbia.edu)
- Xue Bai (xb2115@columbia.edu)
- Zitong Wei (zw2158@columbia.edu)



WHAT IS IT?

- A declarative, interpreted language for writing music
- who will use it?
- What kind of music users are able to write in our language?

WHY DO WE DO IT ?

*Everybody loves music

*Easy to learn our language

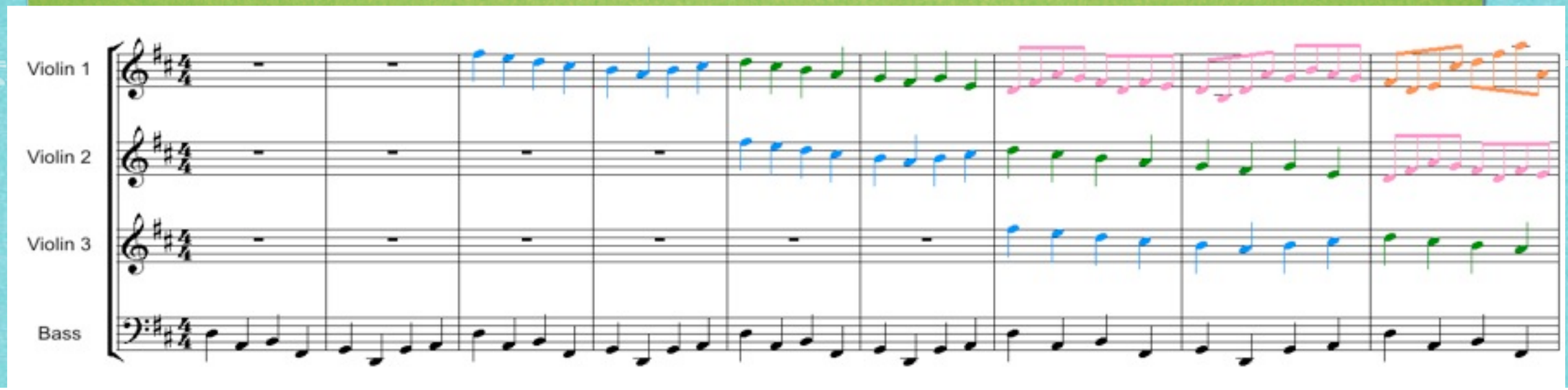
*Easy to test how your music sounds like

Basic Mfan Notation

- Basically, We convert the Modern musical symbols into the programming language like notation.

- **Pattern:** specify a section of melody
- **Song:** specify the name
- **Track:** specify the instrument
- **Melody:** specify the key and tempo

Sample Program : Canon Johann Pachelbel

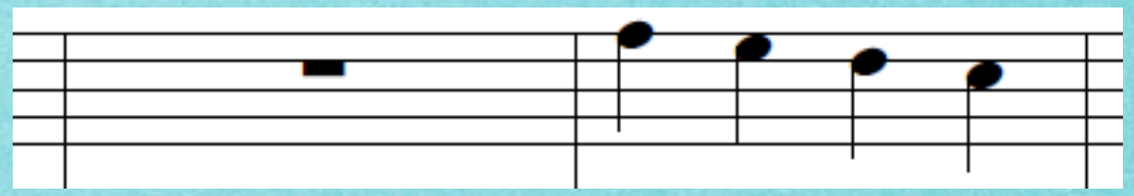



```
%%
Pattern “Rep_basso” | D A3 B3 F3 ||G3 D3 F3 A3 |
%%
Melody(“Canon_violin1”,DMAJOR, 4/4, Adagio)
{| rw||rw || F5 E5 D5 C5 || B A  B C5 ||D5 C5 B A||G F G E |   //etc.}

Melody(“Canon_violin2”,DMAJOR, 4/4, Adagio)
{|r(4)|| F5 E5 D5 C5 || B A  B C5 ||D5 C5 B A||G F G E |  //etc.}

Melody(“Canon_violin3”,DMAJOR, 4/4, Adagio)
{| r(6)|| F5 E5 D5 C5 || B A  B C5 ||D5 C5 B A||G F G E |  //etc.}

Melody(“Canon_basso”,DMAJOR, 4/4, Adagio)
{Rep_basso  Rep_basso Rep_basso}
```

```
Song(“Canon”)
{
        Track(Violin)
        {Canon_violin1}

        Track(Violin)
        {Canon_violin2}

        Track(Violin)
        {Canon_violin3}

        Track(Cello)
        {Canon_basso}
}
```

Basic Mfan Notation

- | | denote one bar
- C4w C denotes **Note Name,** 4 denotes **Octave, w** denotes **Note Value**
- Flat: #, Sharp: b, Natural n
- Rest: r



| rw| | F5q E5q D5q C5q|

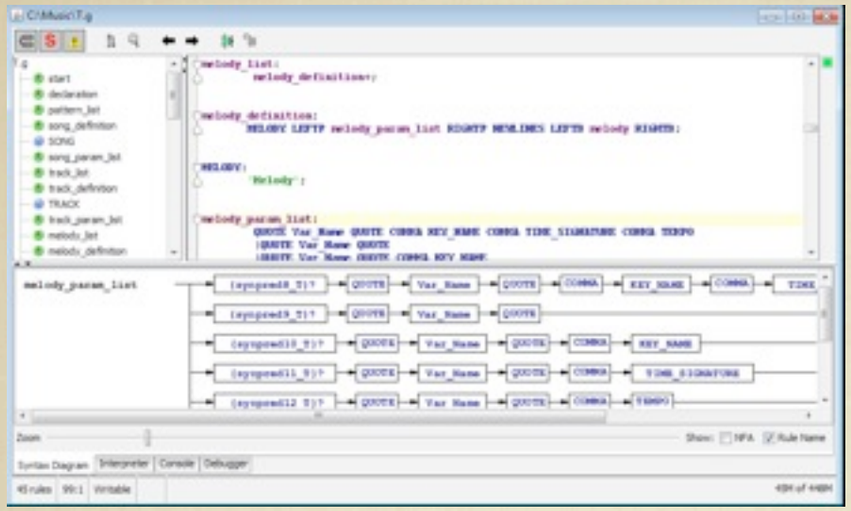
More advanced notation

- **Abbreviation:** C
- **Chord:** + C4w+E4w +G4w
- **Tuplet:** for example three note in one beat

  Tuplet(“C D E”, q)
- **Dynamics:** mf mp f p
- **Dotted note:** C4w.=C4w C4h
- **Repeat signs:** |: :|

  |: | 1st| :| 2nd| |

System Architecture



T.g

Mfan Source code :

```
%%
Pattern "Rep_basso" | D A3 B3 F3 ||G3 D3 F3 A3 |
%%
Melody("Canon_violin1",DMAJOR, 4/4, Adagio)
{| rw||rw || F5 E5 D5 C5 || B A  B C5 ||D5 C5 B A||G F G E |  //etc.}

Melody("Canon_violin2",DMAJOR, 4/4, Adagio)
{|r(4)|| F5 E5 D5 C5 || B A  B C5 ||D5 C5 B A||G F G E |  //etc.}

Melody("Canon_violin3",DMAJOR, 4/4, Adagio)
{| r(6)|| F5 E5 D5 C5 || B A  B C5 ||D5 C5 B A||G F G E |  //etc.}

Melody("Canon_basso",DMAJOR, 4/4, Adagio)
{Rep_basso  Rep_basso Rep_basso}
```

Lexical Analyzer

Tokens

T.token

Syntax Analyzer

Tlexer.java

Semantic Analyzer

Abstract Semantic Tree

Tparser.java

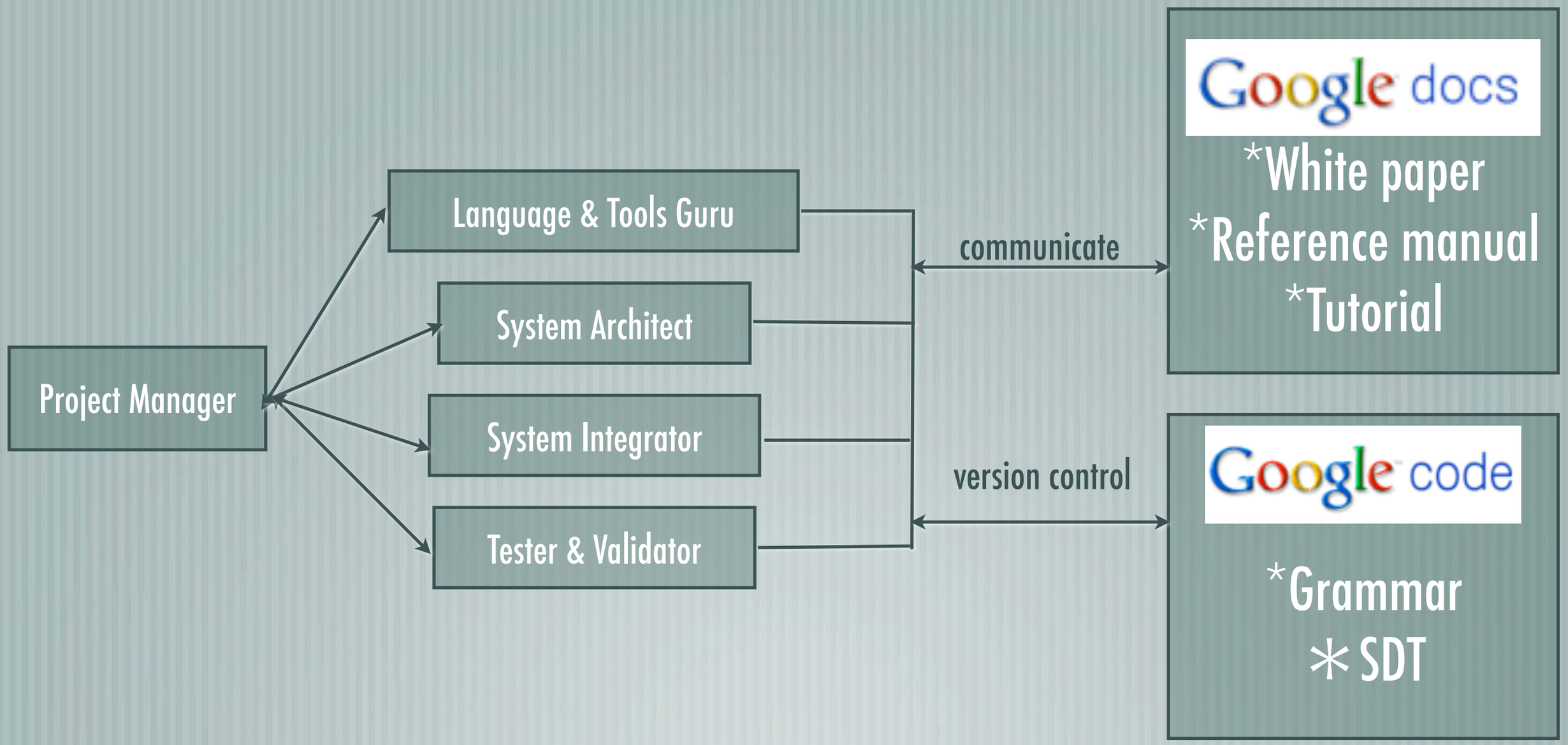
Interpreter

Target Program

System Integration

- Define the system integration environment
- Make sure the compiler components work together

System Integration



Mfan Source code

```
%%%
Pattern "Rep_basso" | D A3 B3 F3 ||G3 D3 F3 A3 |
%%%
Melody("Canon_violin1",DMAJOR, 4/4, Adagio)
{| rw||rw | F5 E5 D5 C5 || B A  B C5 ||D5 C5 B A||G F G E |   //etc.}

Melody("Canon_violin2",DMAJOR, 4/4, Adagio)
{|r(4)|| F5 E5 D5 C5 || B A  B C5 ||D5 C5 B A||G F G E |  //etc.}

Melody("Canon_violin3",DMAJOR, 4/4, Adagio)
{| r(6)|| F5 E5 D5 C5 || B A  B C5 ||D5 C5 B A||G F G E |  //etc.}

Melody("Canon_basso",DMAJOR, 4/4, Adagio)
{Rep_basso  Rep_basso Rep_basso}
```

Lexical Analyzer

Tokens

Syntax Analyzer

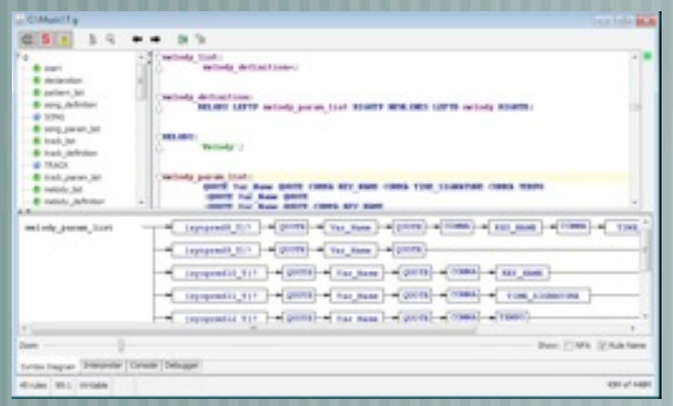
Abstract Semantic Tree

Tree Walker

ANTLR v3

Another Tool for Language Recognition

T.g



T.token
Tlexer.java
Tparser.java

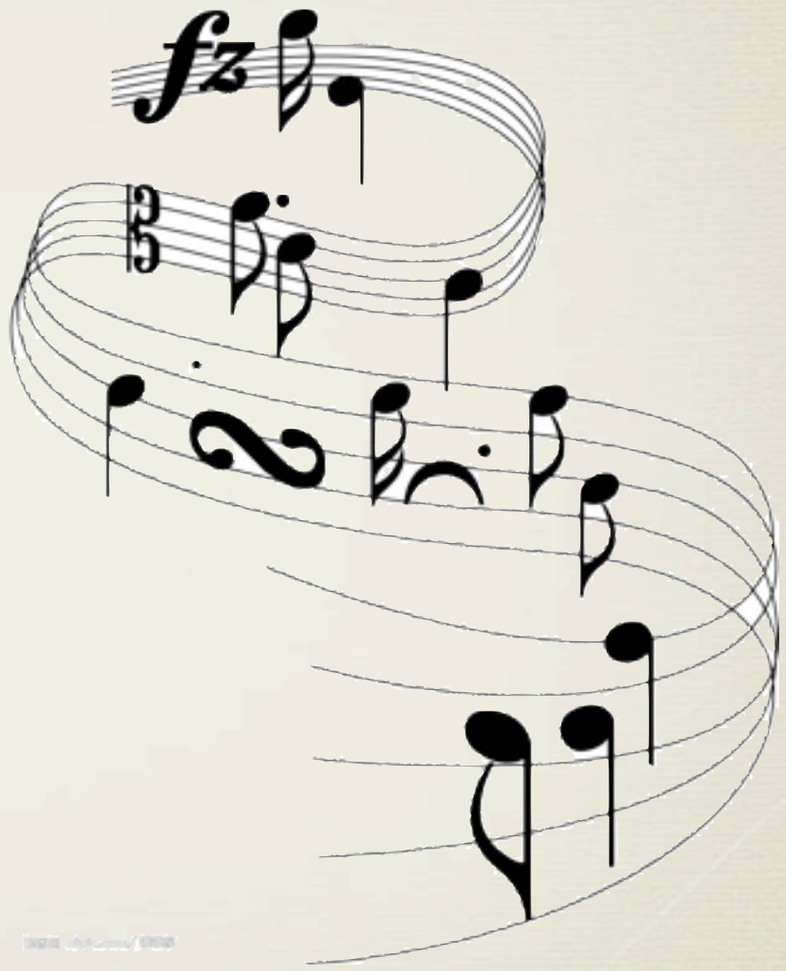
JFugue
Java API for Music Programming

Java
Virtual Machine

platform NetBeans

Test and Validation

- Test
  - Unit Testing
    - Black box, white box, automated test
  - Integration Testing
    - Subsystems (front end, back end)
    - Link to test the whole system
  - System/Acceptance Testing
    - Usability test (easiness, user need)
    - Performance test (large input, OS, hardware)

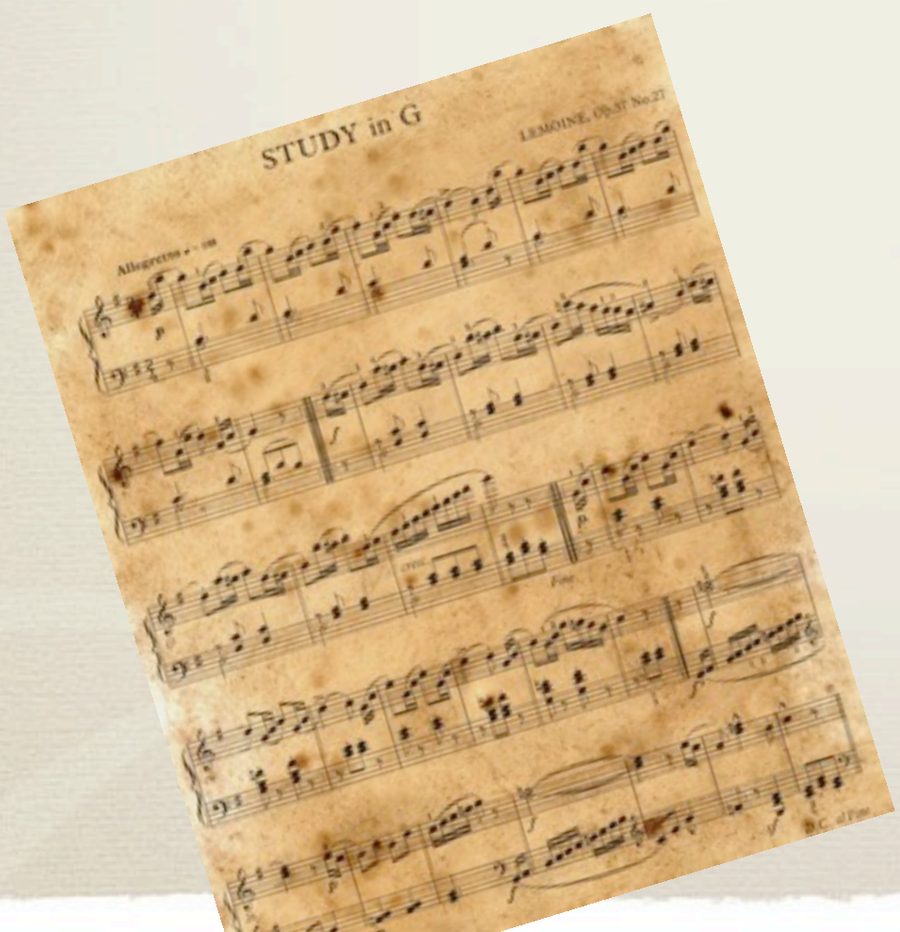


Schedule

Date	Milestone
Feb 24	White Paper ✓
Mar 24	Language Tutorial, Reference Manual ✓
Apr 20	Lexer and Parser ✓
Apr 26	Syntax Tree, Intermediate Code
May 5	Syntax Directed Translation
May 9	Testing and Final Report

Final Deliverable

- **A language to compose music on computer**
  - Write music in plain text files
  - Support variations on timbre, time signature, key signature and tempo.
  - Support multi-track music
  - Support code reuse for repeated melodies
- **A compiler that compiles the source code into a midi file**



MFan

THANK YOU!!!

The Mfan Team

Project Manager: Weiqi Ma

Language Guru: Guangwei Zhang

System Architect: Cheng Qian

System Integrator: Xue Bai

Tester and Validator: Zitong Wei