THOMAS ELLING, HILA GUTFREUND, EMILY LEMONIER, WILLIAM FALK-WALLACE
Why a music language?

- Music lends itself to programming
- Interesting field to explore
- Allows for the use of Java libraries
- Combines artistry with computer science
Intro to DJ

- DJ abstracts the difficulties of MIDI programming
- DJ makes music programming intuitive
- DJ Focuses on Conventions and Simplified Programming
- DJ enables serial and parallel addition to extend chords and tracks
Language tutorial:

1. JMusic
2. Data Types & Hello World
3. Programmatic Structure
4. Control Flow
5. Functionality
6. Example: Hello World
JMusic

- Music library for Java
- Abstracts difficult midi music composition
- Allows for the creation of notes, chords, tracks, and scores.
- Built in instrument library including: flute, piano, guitar, xylophone, etc.

http://explodingart.com/jmusic/
song score ( ) {
	double pitchA = 141.32;
	double volume = 100;
	double duration = 2;
	double piano = 0;

	note n = note (pitchA, volume, duration);

cord c = chord ( n );

track t = track ( 0 );

t = t.c;

score s = score( t );

return s;
}
DJ: Programmatic Structure

Main function:
song score () { ... }

Global Variables

Inline initialization

double C4 = 261.63;

song score () {
    double pitchA = C4;
    double volume = 100;
    double duration = 5;
    note n;

    n = note (pitchA, volume, duration);
    track t = track (5);
    score s = score( t );

    return s;
}
DJ: Control Flow

For

If/Else

While

Loop

song score () {
    double i;
    for (i = 0 ; i < 5 ; i = i + 1) {...}
    loop (5) {...}
    score s = score();
    return s;
}
DJ: Functionality

- Serial Add
- Parallel Add
- Note Attribute Accessor

```java
createOtherNote note (note n) {
    /* creates + returns new note*/
    double p = n -> pitch + 40;
    double v = n -> vol + 10;
    double d = n -> dur + 5;
    return note(p, v, d);
}

song score () {
    note n1 = note(440, 100, 5);
    note n2 = createOtherNote(n1);
    note n3 = createOtherNote(n2);
    chord c = chord(n1);
    c = c:n2;
    c = c:n3;
    track t = track(26);
    t = t.c;
    score s = score(t);
    return s;
}
```
Example: HelloWorld

Making Noise!

```c
song score ()
{
    double pitchA;
    double volume;
    double duration;

    pitchA = 440;
    volume=50;
    duration=4;

    note n = note (pitchA, volume, duration);

    chord c = chord(n);
    track t = track( 0 );
    t = t . c;
    score s = score(t);

    return s;
}
```
Implementation:

1. Process
2. Structure
3. Statistics
Implementation Process: semcheck
Implementation Process: Javagen
Implementation Process: More Javagen
Structure of DJ

1. Input File
2. Scanner
3. AST
4. Parser
5. Semcheck
6. Java Code
7. Java Compile Utility (compile.sh)
8. Midi File
9. SAST
10. Java Generation (compile.ml)
Statistics

- 540 Commits to Master
- 25 Feature Branches
- 30 Issues and Pull Requests Assigned
Lessons Learned

● Do your research into any extra libraries you’ll need
● Everyone should be tangentially involved in every step of the way
● Don’t shy away from trying a hard language
● To fully create a language, you are going to have to leave time to make it twice
All code can be found on our github repository at:

http://whet-plt.github.io/wdjc/

Thank you!
Demo: Legend of Zelda!