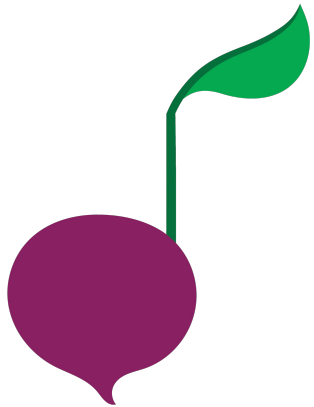
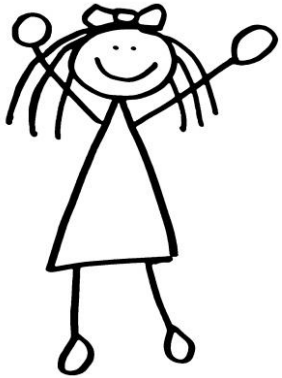


Sick Beets

Team Roles

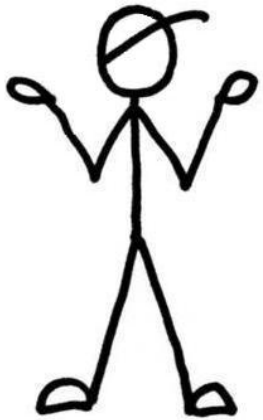
Courtney

Manager



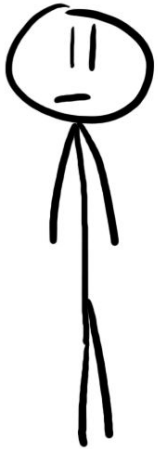
Angel

Language Guru



Kevin

System Architect



Jin

Tester

Project Goal

Create a language that makes it easy to compose music by rendering MIDI files

Language Basics & Control Flow

bool

if-else

int

string

for loop

float

array

while loop

Language Specifics

Notes

print

Durations

setTempo

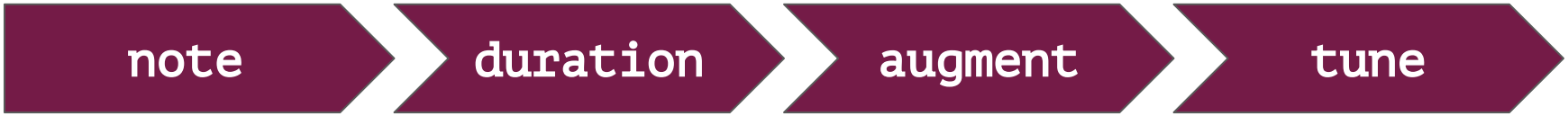
setInstrument

Tunes

addLayer

render

Basics of Making Music



Format: i.e. eb6
NoteAccidental
Octave-Offset

Accidentals:
‘b’ or ‘#’ or ‘’

Octaves:
middle C = 5

Durations:
w: whole
h: half
q: quarter
i: eighth
s: sixteenth
t: thirty-second

* can combine

Augment:
uses the operator ‘:’

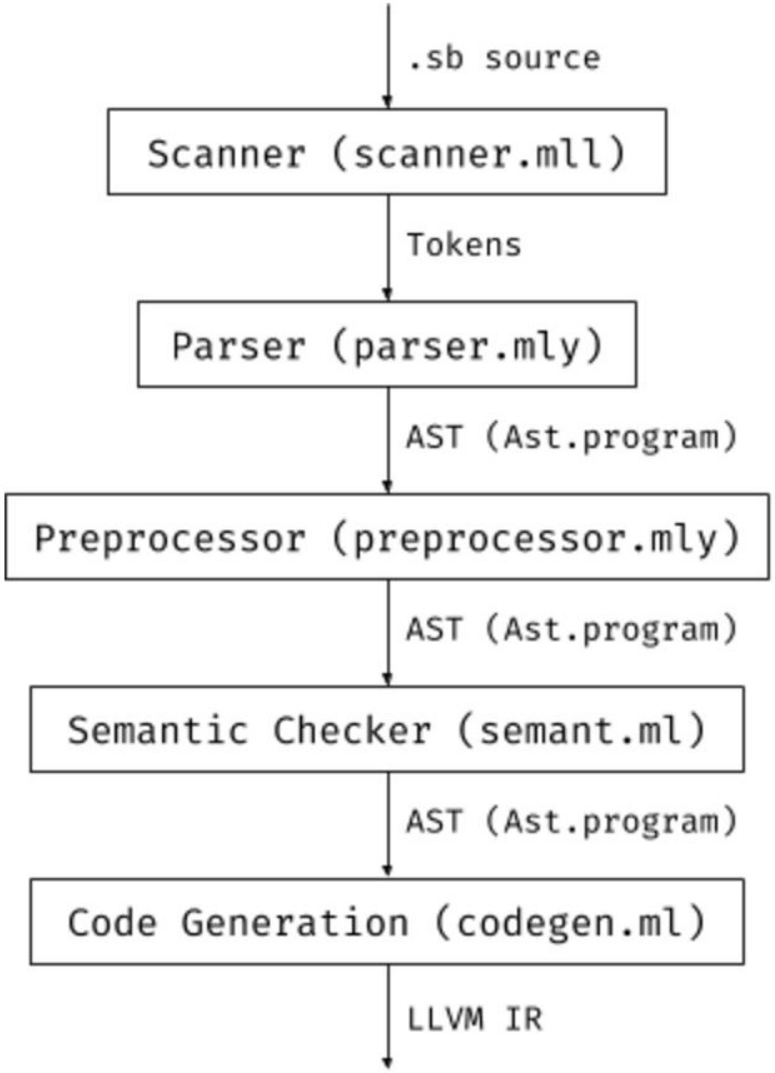
i.e. tune = note(s) : duration(s)

Types of Tune Manipulation:
- concat ‘.’
- setTempo
- setInstrument
- addLayer

Mary Had a Little Lamb

```
1 tune x = [e,d,c,d,e,e,e,r] : i
2 tune y = [d,d,d,r,e,g,g,r] : i
3 tune z = [e,d,d,e,d,c] : i
4
5 tune mary = x.y.x.z
6 tune maryflute = setInstrument(mary, "flute")
7 tune marypiano = setInstrument(mary,"piano")
8
9 render (marypiano.maryflute, "mary.midi")
```

Compiler Architecture



Test Suite

```
------------------------------------------------------------------------
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ --
52 out of 54 tests passed

Failing Tests:
 test-error
1c1
< 115
\ No newline at end of file
---
> not 115
\ No newline at end of file
 test-error2
Fatal error: exception Parsing.Parse_error
------------------------------------------------------------------------
```

Running a Program + Demo

1. **Compile the CFugue Library**

```
cd CFugue
cmake CMakeLists.txt
make
cd ..
```

2. **Make the compiler**

```
make
```

3. **Compile the .sb file llvm and run the llvm code**

```
./sb.native < test.sb > test.ll
lli test.ll
```


Thank You!
vevo