Structuring Music

The Organization of JMusic Objects
Thought Experiment

• How are they doing that?
• How can there be any number of Notes in a Phrase, Phrases in a Part, and Parts in a Score?
• (Hint: Not Arrays.)

How do we explore composition?

• We want to quickly and easily throw together notes in different groupings and see how they sound.
• The current jMusic structure models music.
  • Let’s try to create a structure that models thinking about music as bunches of riffs/SongElements that we want to combine in different ways.

Version 1: Notes in an array

• We can store our notes in an array.
• Traverse the array to gather the notes up into a Phrase, then use View to notate the Phrase.
Using an array to structure Notes

```java
Note[] someNotes = new Note[100];
for (int i = 0; i < 100; i++)
    someNotes[i] =
        new Note((int) (128 * Math.random()), 0.25);
// Now, traverse the array and gather them up.
Phrase myphrase = new Phrase();
for (int i = 0; i < 100; i++)
    myphrase.addNote(someNotes[i]);
View.notate(myphrase);
```

//---//---// Methods //---//---//
public void addNote(Note n) {
    notes.add(n);
}

public Phrase getPhrase()
{
    Phrase phr = new Phrase();
    for (int i = 0; i < notes.size(); i++)
        phr.addNote(notes.get(i));
    return phr;
}

Version 1: Note Array

```java
public class NoteArray
{
    //---//---// Fields //---//---//
    private ArrayList<Note> notes;

    //---//---// Constructors //---//---//
    public NoteArray()
    {
        notes = new ArrayList<Note>();
    }

    public static void main(String[] args)
    {
        NoteArray na = new NoteArray();
        na.addNote(new Note(JMC.D2, JMC.ENT));
        na.addNote(new Note(JMC.FS2, JMC.ENT));
        na.addNote(new Note(JMC.A2, JMC.ENT));
        na.addNote(new Note(JMC.A1, JMC.ENT));
        na.addNote(new Note(JMC.CS2, JMC.ENT));
        na.addNote(new Note(JMC.E2, JMC.ENT));
        na.addNote(new Note(JMC.G1, JMC.ENT));
        na.addNote(new Note(JMC.B1, JMC.ENT));
        na.addNote(new Note(JMC.D2, JMC.ENT));
        na.addNote(new Note(JMC.A1, JMC.ENT));
        na.addNote(new Note(JMC.CS2, JMC.ENT));
        na.addNote(new Note(JMC.E2, JMC.ENT));

        Phrase phr1 = na.getPhrase();
        Part part1 = new Part("Bass", JMC.PBASS, phr1);
        Score score = new Score("Buster", 100.0, part1);
        View.notate(score);
    }
```

version 1: repeat

can

public void repeat(int n) {
    int size = notes.size();
    for(; n > 0; n--)
        notes.add(notes.get(i));
}

version 1: doubleNote

public void doubleNote(int index) {
    Note orig = notes.get(index);
    Note half = new Note(orig.getPitch(),
                          orig.getRhythmValue() * 0.5);
    notes.set(index, half);
    notes.add(index + 1, half);
}
Critique of Version 1

- So where’s the music?
  - We don’t think about notes as just one long strand.
- Where are the phrases/riffs/elements?
  - We just have one long line of notes.
- How do we explore patterns like this?
  - `insertAfter` and `delete` are just as hard here as in sampled sounds!