Computer Music

Assignment 4

Set Multiplication

1 Objective

In this assignment you will implement a basic set multiplication algorithm and use it to compose a piece for cello and fixed media. You will automate the multiplication process and control musical parameters such as density, amplitude, and register using envelopes or other functions of time.

To get a general sense of what we are implementing, watch this video demonstrating the manual process.

`https://www.youtube.com/watch?v=-_I8aE_hyd4`

In the video, the process is done by hand. We will automate these processes and add dynamic control over musical parameters.

2 Java Implementation

2.1 Step 1: Copy Dependencies

Copy the following dependencies from the *Stepwise-Voice-Leading* exercise into the `java-version/src/main/java/org/delightofcomposition/voiceleading` directory of the current project:

- `Directed.java`
- `NoCommons.java`
- `NoCommonsDirected.java`
- `VoiceLeadingFramework.java`
- `VoiceLeading.java`

2.2 Step 2: Review the Interface

Consult `java-version/src/main/java/org/delightofcomposition/Multiplication.java` for detailed directions on what methods to implement.

2.3 Step 3: Create Implementation Class

In the `delightofcomposition` directory, create the file `SetMultiplication.java`.

2.4 Step 4: Add Imports

Add the following import statements to your file:

```
import java.util.ArrayList;
import java.util.Random;
```

2.5 Step 5: Extend the Abstract Class

Make your class extend `Multiplication.java`:

```
public class SetMultiplication extends Multiplication {
    // Your implementation here
}
```

2.6 Step 6: Compile and Execute

Once you've finished implementing, uncomment the block found in `Main.main` and execute `Main.java` either using VS Code or raw Maven. Within the `java-version` directory, run:

```
mvn clean compile exec:java
```

This should produce a file called `composition.wav` in the top-level directory.

2.7 Step 7: Listen and Adjust

Listen to the generated `composition.wav` file. Now you're ready to adjust the envelopes to shape the musical output.

The four envelopes control:

- **Env 1:** Range
- **Env 2:** Amplitude
- **Env 3:** Density
- **Env 4:** Pan

2.8 Step 8: Run the Envelope GUI

Launch the envelope editor to adjust the parameters:

```
java -jar ./src/main/java/org/delightofcomposition/envelopes/EnvGui.jar
```

After adjusting the envelopes in the GUI, rerun the compilation command from Step 6 to regenerate the audio with your new parameter settings.

2.9 Step 9: Optional Experiments

If you finish early, feel free to experiment with:

- Alternative functions to control various parameters
- Generating rhythmic cycles

Frameworks for such experiments may be found in:
`java-version/src/main/java/org/delightofcomposition/simpleparams`

3 Final Composition

Compose a brief work for cello and fixed media using the process you've just built. Your composition should demonstrate:

- Effective use of the set multiplication algorithm
- Thoughtful control of musical parameters through envelopes
- Musical coherence between the electronic fixed media and the cello part

The generated audio file will serve as the fixed media component. You are responsible for composing an appropriate cello part that interacts musically with the electronic material.