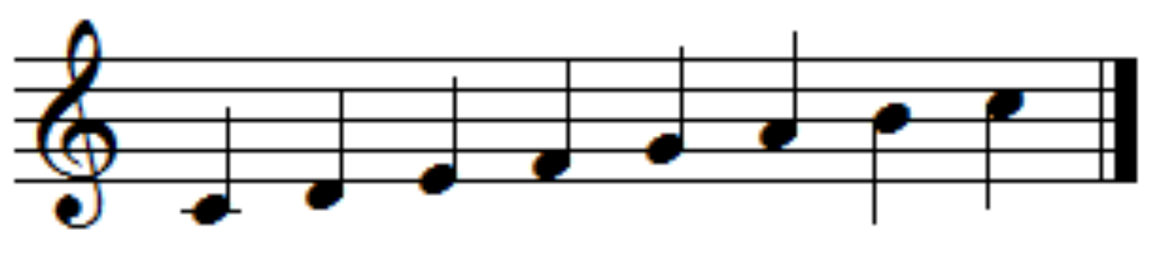
Using Music Notation for Teaching Computer Programming

Eunjeong KO, Kyogu LEE

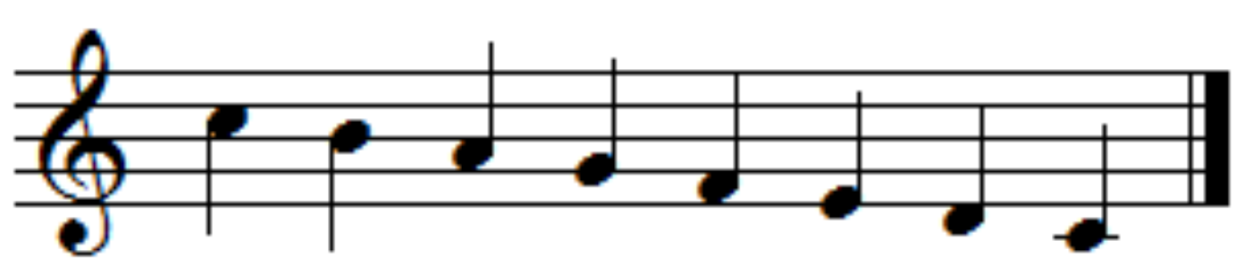
Music and Audio Research Group, Graduate School of Convergence Science and Technology,
Seoul National University, Republic of Korea
[konzung, kglee] @ snu.ac.kr

Simplified programming syntax examples in the code editor



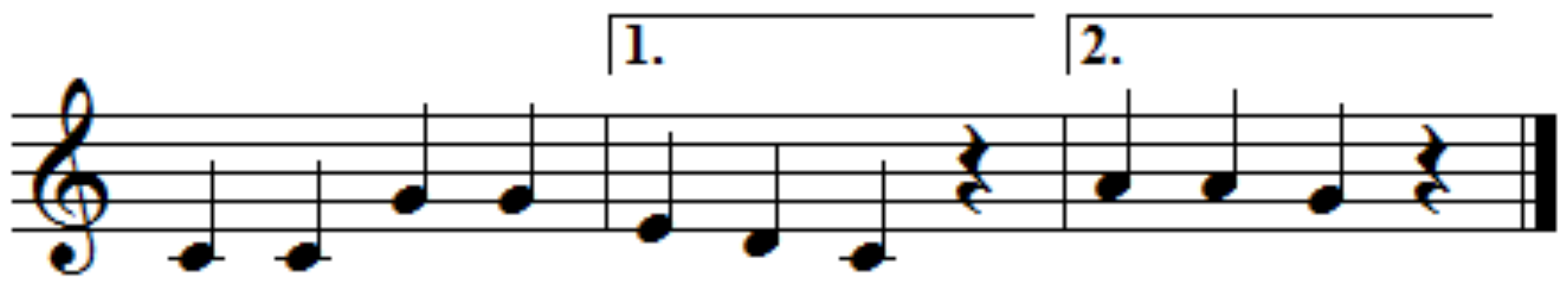
```
change(note) {
     note + change(note+1);
}

change(note) {
     note + change(note+2);
}
```

Recursive Example 1



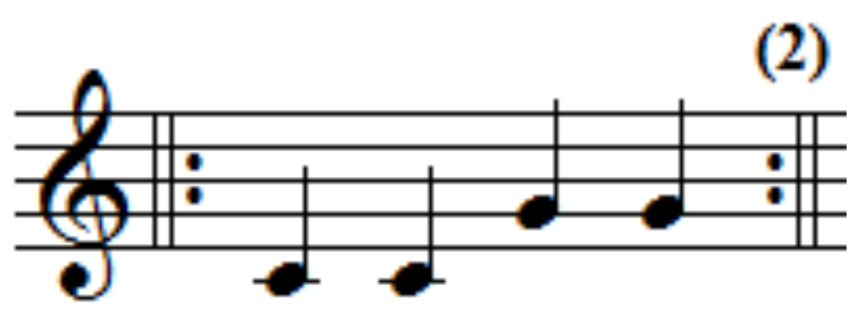
```
melody(note) {
  if(note == do) do;
  else
    melody(note-1) + melody(note-2)
}
```

Recursive Example 2



```
melody(happy) { do do sol sol; }
melody(first ending) { mi re do; }
melody(second ending) { la la sol; }

if(happy once) {
  first ending;
} else {
  second ending;
}
```

Conditional structure



```
melody(twinkle)
{ do do sol sol; }

repeat(twinkle,twice);
```

Loop structure

Introduction

- It is inevitable that novice programmers often feel difficulties in understanding abstract concepts without any similar phenomenon in the daily life for comparison
- Many novice programmers face difficulties and give up in the early stages, just because they are not familiar programming syntax and semantics

Objective

We present a method for programming language education using music notation with an aim to entice novice programmers to write their own programs

Music notation as an analogy to programming

- ♪ We use music notation as an analogy to programming, based on the observation that there are similar attributes between the two
- ♪ We provide users with on-line auditory feedback to immediately notify potential errors in a pleasant way
- ♪ We find several key concepts in programming language syntax and semantics, and translate them into music notation to help beginners

A survey of 132 participants showed that

90.2% participants **understand** western music scores

3.54 (scale of 1 to 5) an average degree of understanding

Method

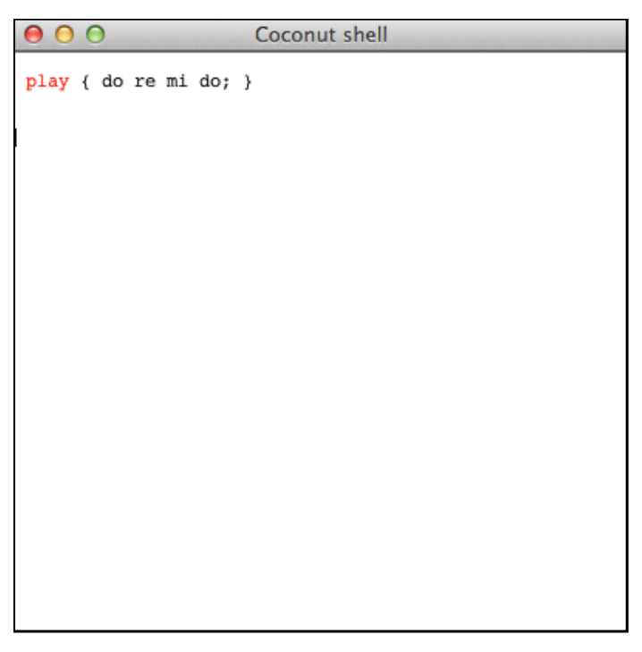
Musical settings as programming metaphors

Concepts	Musical settings
Arithmetic operation	Using a chord as the addition of notes
Function definition	Using a melody and making a sequence with a variety of predefined melody
Conditional structures	First and second endings for an if-else statement
Loop structures	Repetition for loops
Recursion	Calling a rhythm and melody repeatedly in a sequence
Solving a certain task	Task with sorting example
Finding a bug	Using an on-line auditory feedback for communication

Programming with music notation and auditory result

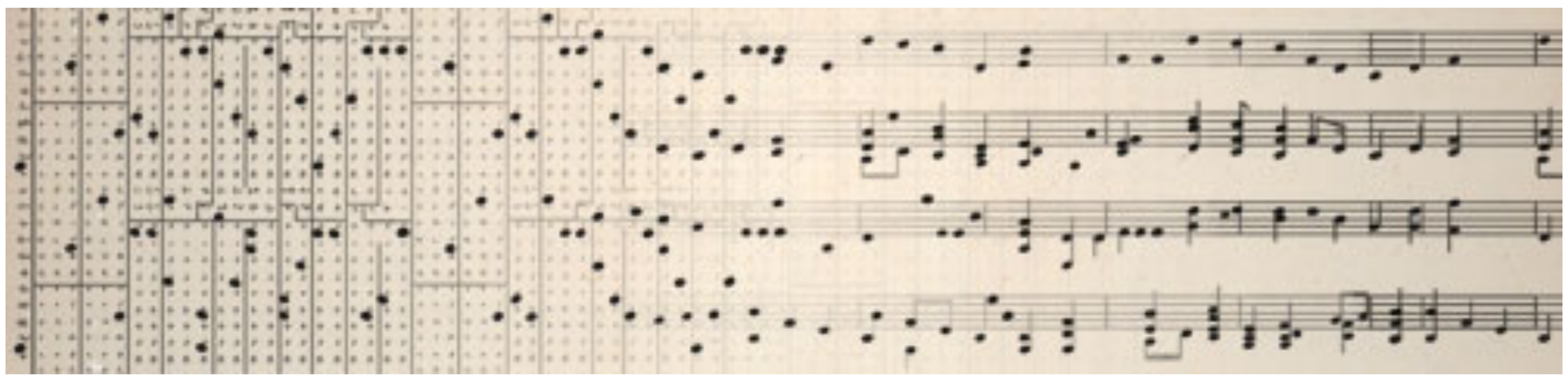
Java API with audio arguments

- We utilize JFugue to implement our method for on-line auditory feedback as a communication medium
- JFugue is a Java API that allows users to create MIDI files in the Java language



```
play { do re mi do; }
```

Code editor framework
Make it possible for beginners to write their own programs and listen to the results immediately through the code editor



from Google Image search

Preliminary user study

- Five teenage students (aged 12-16 years), non of whom had previous experience with computer programming
- First hypothesis: developing a positive first impression in programming, motivating the novices to learn programming
- Second hypothesis: on-line auditory feedback aid error finding in the programming process
- After half an hour of tutoring, the students could follow our instructions, enticing them to freely use the code editor environment

Conclusion

- We view our research as a gateway to language like Java
- Our research would bring a more pleasant and enjoyable environment for programming language education
- In the next step, we will focus on the evaluation and the implementation of our method

References

- J. M. Wing. Computational thinking. Communications of the ACM, 49(3), 33-35. 2006.
- F. K. Dannenberg, R. B. Dannenberg, and P. L. Miller. Teaching Programming to Musicians, 1984.
- I. Utting, S. Cooper, M. Kölling, J. Maloney, and M. Resnick. Alice, greenfoot, and scratch–a discussion. ACM Transactions on Computing Education (TOCE), 10(4), 17. 2010.
- M. J. Lee, and A. J. Ko. Personifying programming tool feedback improves novice programmers' learning. In Proceedings of the seventh international workshop on Computing education research (pp. 109-116). ACM. 2011.
- D. S. Hughes. Introducing programming logic in a one-credit course. In Proceedings of the 50th Annual Southeast Regional Conference (pp.48-52). ACM. 2012.