

プログラマブルな音楽制作ソフトウェアの開発

—mimum:インフラとしての音楽プログラミング言語—

松浦 知也(九州大学 大学院芸術工学府 芸術工学専攻 博士後期課程)



- ・ 新しい音楽プログラミング言語“**mimum**”をゼロから開発
- ・ 複雑な内部状態を簡潔に、フルスクラッチで記述できる信号処理
- ・ オーディオドライバクロックに基づいたサンプル精度のイベント制御
- ・ LLVMバックエンドによる高い実行性能&多プラットフォームでの動作

<https://github.com/mimum-org/mimum>

信号処理

```
fn phasor(freq:float){
  return (self+freq/48000)%1
}
fn scaleTwopi(input){
  return input* 2 * 3.141595
}
fn osc(freq){
  return freq |> phasor |> scaleTwopi |> cos
}
fn dsp(time:float)->float{
  return osc(440)
}
```

- ・ **dsp**関数を書いて音を出力
- ・ **self**で関数が最後に出力した値を取得
- ・ パイプライン演算子(|>)でUGenライクな記述も

制御処理

```
ntrigger = 1
fn setN(val:float){
  ntrigger = val
}
fn playN(duration:float)->void{
  setN(1)@now
  setN(0)@(now+duration)
}
fn Nloop(period:float)->void{
  playN(50)
  nextperiod = if(random()>-0.3) period/2 else period
  Nloop(period)@(now+nextperiod)
}
Nloop(12000)@0
```

- ・ **@**をつけるとその時間に関数を実行
- ・ **now**でランタイムの現在時刻の取得
- ・ 時間を跨いだ再帰でリズムパターンを作成