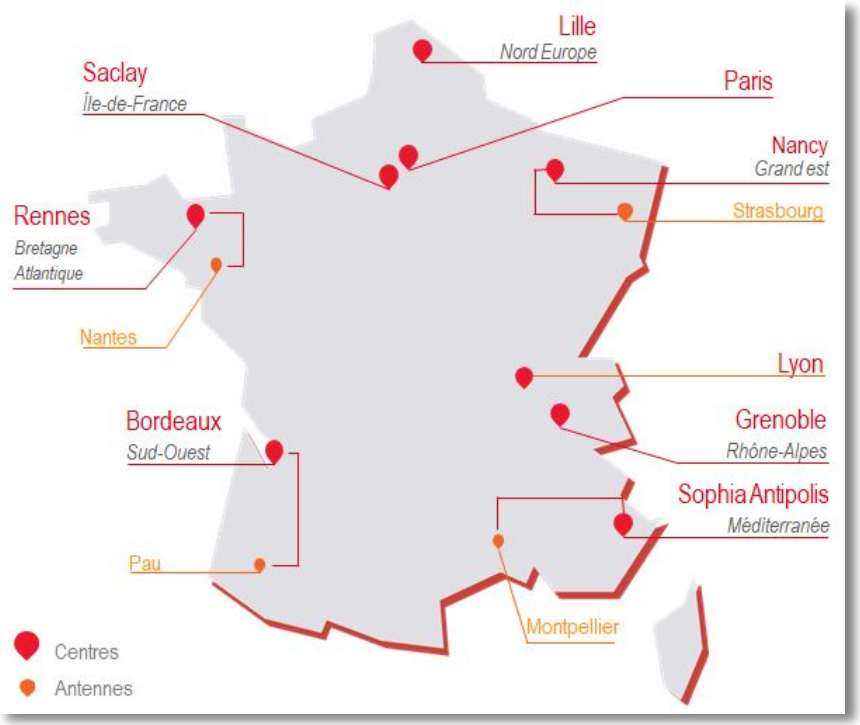
Music Live Coding en Pharo

Antoine Delaby

30 juin 2022

Tuteurs de stage
Santiago Bragagnolo
Clément Quinton

INRIA



200 équipes de recherche, 3900 chercheurs

10 centres de recherche

Création de l'IRIA en 1967

RMOD

Membres

Permanents

- **Professeur**
  - Anne Etien
- **Directeur de recherche**
  - Stéphane Ducasse (Responsable)
- **Maîtres de conférences**
  - Nicolas Anquetil (hdr)
  - Vincent Aranega
- **Chargés de recherche**
  - Steven Costiou
  - Marcus Denker
- **Ingénieurs**
  - Christophe Demarey
  - Guillermo Andres Polito
  - Pablo Adrian Tesone

Non permanents

- **Doctorants**
  - Nour jihene Agouf
  - Santiago Bragagnolo
  - Aless Hosry
  - Pierre Misse-Chanabier
  - Théo Rogliano
  - Maximilian Willembrinck
  - Oleksandr Zaitsev
- **Ingénieurs**
  - Soufyane Labsari
  - Clotilde Toullec

Recherche en Génie Logiciel

Utilisation du langage Pharo

Ma mission

Implémentation du protocole MIDI dans Pharo

Appels à PortMidi

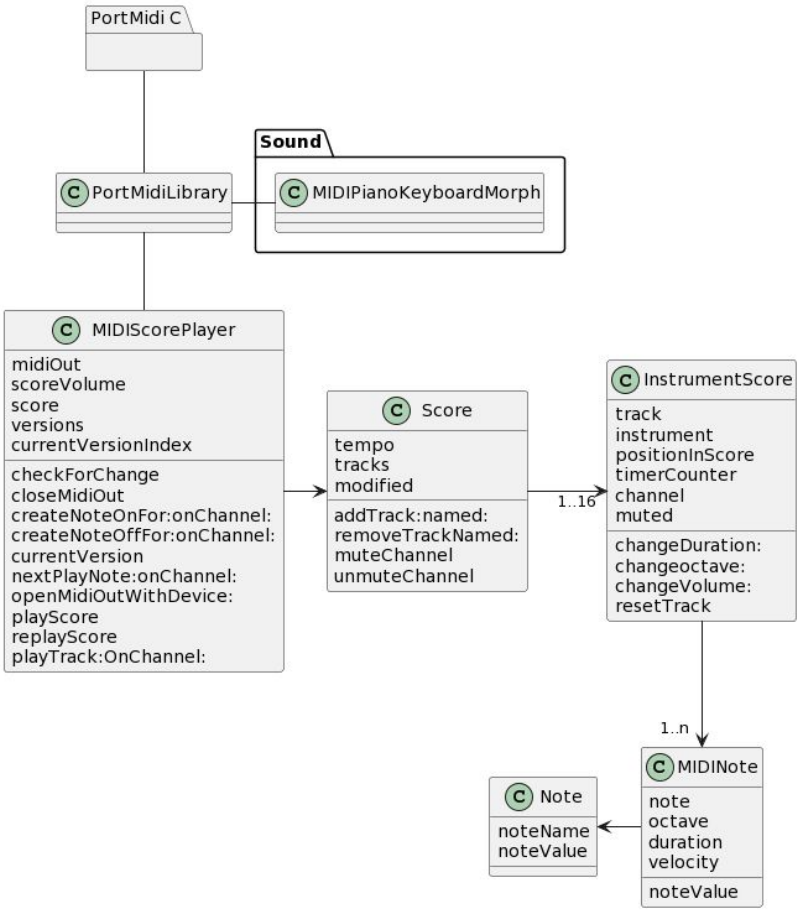
Support de live coding dans Pharo

Annonce du plan

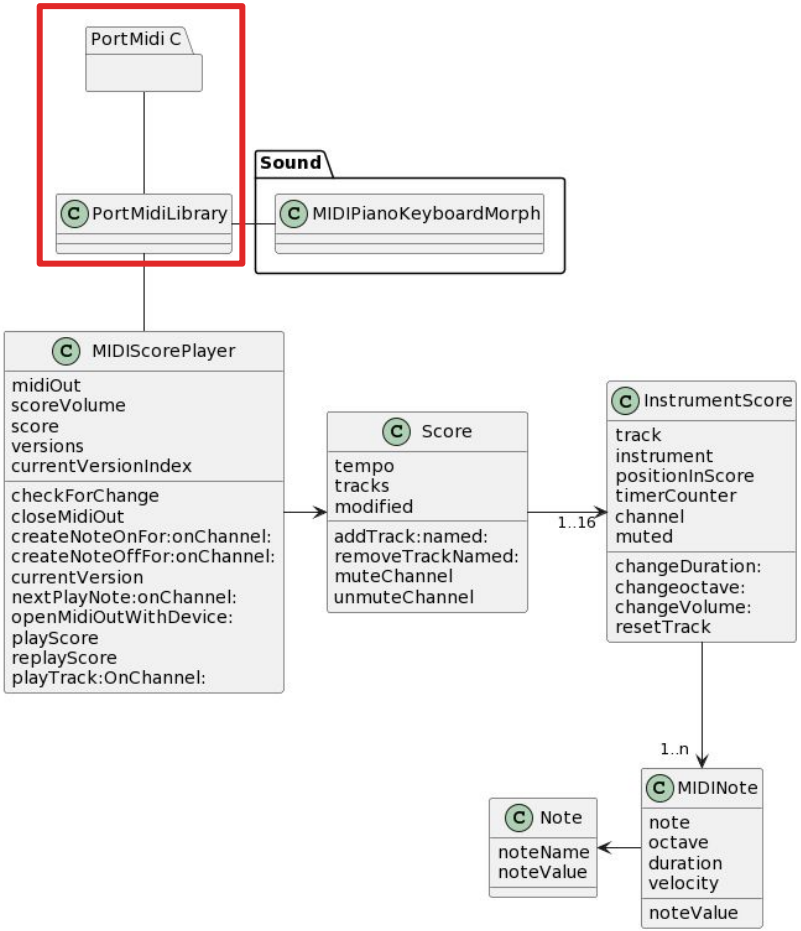
- *Vision globale du projet*
- *Notions musicales*
- *Le protocole MIDI*
- *PortMidi*
- *PortMidi dans Pharo*
- *Support live coding dans Pharo*
- *Conclusion*

- ***Vision globale du projet***
- Notions musicales
- Le protocole MIDI
- PortMidi
- PortMidi dans Pharo
- Support live coding dans Pharo
- Conclusion

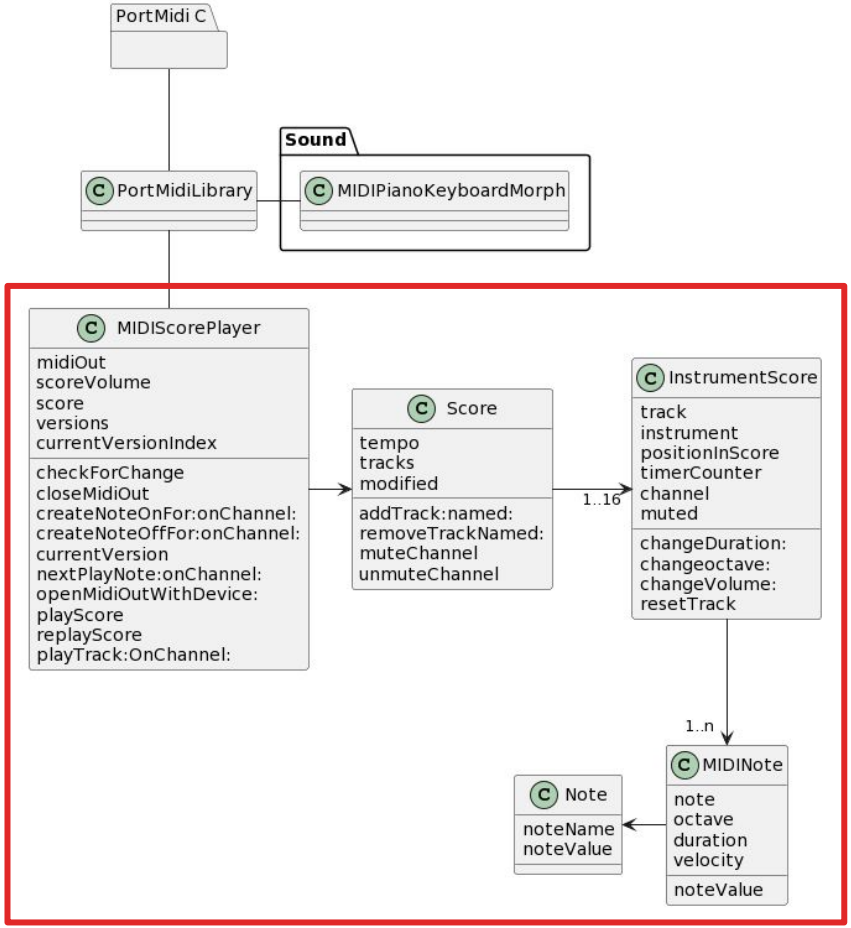
Vision globale du projet



Vision globale du projet



Vision globale du projet



- Vision globale du projet
- ***Notions musicales***
- Le protocole MIDI
- PortMidi
- PortMidi dans Pharo
- Support live coding dans Pharo
- Conclusion

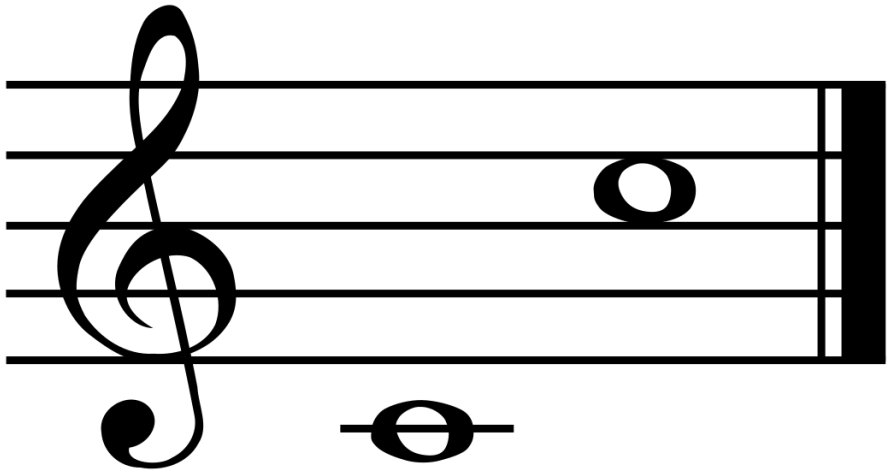
Notions musicales

Notions de base

- Type de note
- Hauteur de la note
- Durée de la note
- Intensité de la note

Notions musicales

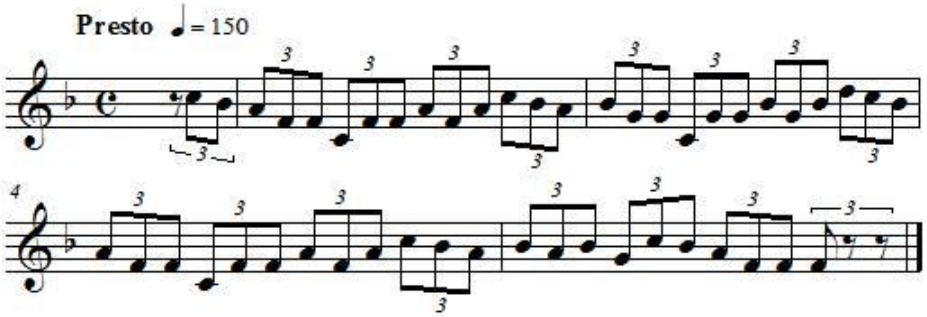
Notions de base



Octave Do grave au Do aigü

Notions musicales

Notions de base

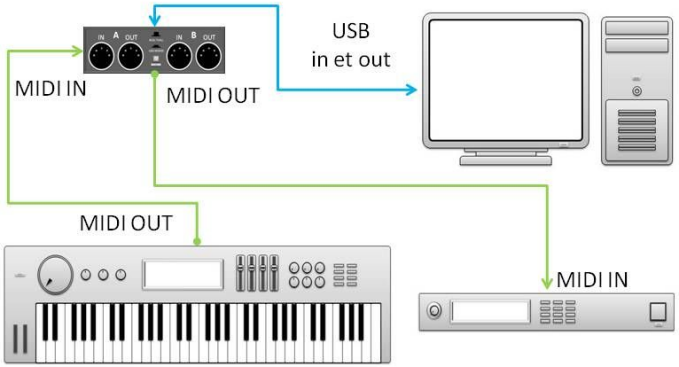


Rythme de la note et tempo

- Vision globale du projet
- Notions musicales
- ***Le protocole MIDI***
- PortMidi
- PortMidi dans Pharo
- Support live coding dans Pharo
- Conclusion

Le protocole MIDI

MIDI = Musical Instrument Digital Interface



Normaliser les échanges d'informations musicales

Le protocole MIDI

Message MIDI

Status Byte	Data Byte 1	Data Byte 2
1001 1111	00111100	01111111
Note / MIDI Channel 16	Note Number 60 (C3)	Note Velocity 127

Note jouée

Hauteur de la note

Intensité de la note

- Vision globale du projet
- Notions musicales
- Le protocole MIDI
- ***PortMidi***
- PortMidi dans Pharo
- Support live coding dans Pharo
- Conclusion

PortMidi

```c
void sendMessage(){
    //buffer for message sending
    PmEvent buffer;
    // creation of C note (mid frequency C) noteOn
    int status = 0x90;
    int data1 = 0x3C;
    int data2 = 0x40;
    buffer.message = Pm_Message(status, data1, data2);
    //send message to midiOut
    if(Pm_Write(midi_out, &buffer, 1) == 0)
        printf("Messages begin note sent\n");
```

Protocole MIDI en langage C

Gestion des flux et des messages

```c
void receiveMessage(){
    //buffer for message sending
        PmEvent buffer;
    //message properties
    int status, data1, data2;
    if(Pm_Poll(midi_in)){
        //Reading received message
        Pm_Read(midi_in, &buffer, 1);
        status = Pm_MessageStatus(buffer.message);
        data1 = Pm_MessageData1(buffer.message);
        data2 = Pm_MessageData2(buffer.message);
```

https://github.com/PortMidi/portmidi

- Vision globale du projet
- Notions musicales
- Le protocole MIDI
- PortMidi
- ***PortMidi dans Pharo***
- Support live coding dans Pharo
- Conclusion

PortMidi dans Pharo

uFFI

```
FFILibrary subclass: #PortMidiLibrary
    instanceVariableNames: ''
    classVariableNames: ''
    package: 'PortMidiDemo-PortMidi'
```

uFFI = Unified Foreign Function Interface

Appeler des fonctions C dans Pharo

PortMidi dans Pharo

Fonction simple

```
portMidiInitialize
    "Calls the PortMidi C library method that initializes the library."
    ^ self ffiCall: #( int Pm_Initialize() )
```

Appel à ffiCall:

Type PmError -> int

PortMidi dans Pharo

Fonction complexe

```
openOutputForStream: aStream withDevice: outputDeviceID withOutputDriverInfos: infos
withBufferSize: outputBufferSize withTimeProcedure: timeProcedure withTimeInfos: timeInfos
andLatency: latency
   "Calls the PortMidi C library method that open an MIDI output where aStream is a pointer
that receives a pointer to the newly opened stream, outputDeviceID is the device to open.
Parameters infos, timeProcedure and timeInfos are optional."
   ^ self ffiCall: #( int Pm_OpenOutput( void** aStream, int outputDeviceID, void* infos,
uint outputBufferSize, void* timeProcedure, void* timeInfos, uint latency ) )
```

Paramètres facultatifs : void*

Mémoire dynamique - statique

- Vision globale du projet
- Notions musicales
- Le protocole MIDI
- PortMidi
- PortMidi dans Pharo
- ***Support live coding dans Pharo***
- Conclusion

Support live coding dans Pharo

Objets Notes

```
Object subclass: #MIDINote
    instanceVariableNames: 'note octave duration velocity'
    classVariableNames: 'DefaultDuration DefaultOctave DefaultVelocity'
    package: 'PortMidiDemo-Source'
```

Notation d'une note

Support live coding dans Pharo

Objets Partitions

```
Object subclass: #InstrumentScore
    instanceVariableNames: 'track instrument positionInScore timerCounter channel muted'
    classVariableNames: 'DefaultVelocity'
    package: 'PortMidiDemo-Source'
```

Une ou plusieurs partitions d'instrument

Mélodie à jouer

```
Object subclass: #Score
    instanceVariableNames: 'tempo tracks modified'
    classVariableNames: 'DefaultTempo'
    package: 'PortMidiDemo-Source'
```

Support live coding dans Pharo

MIDIScorePlayer

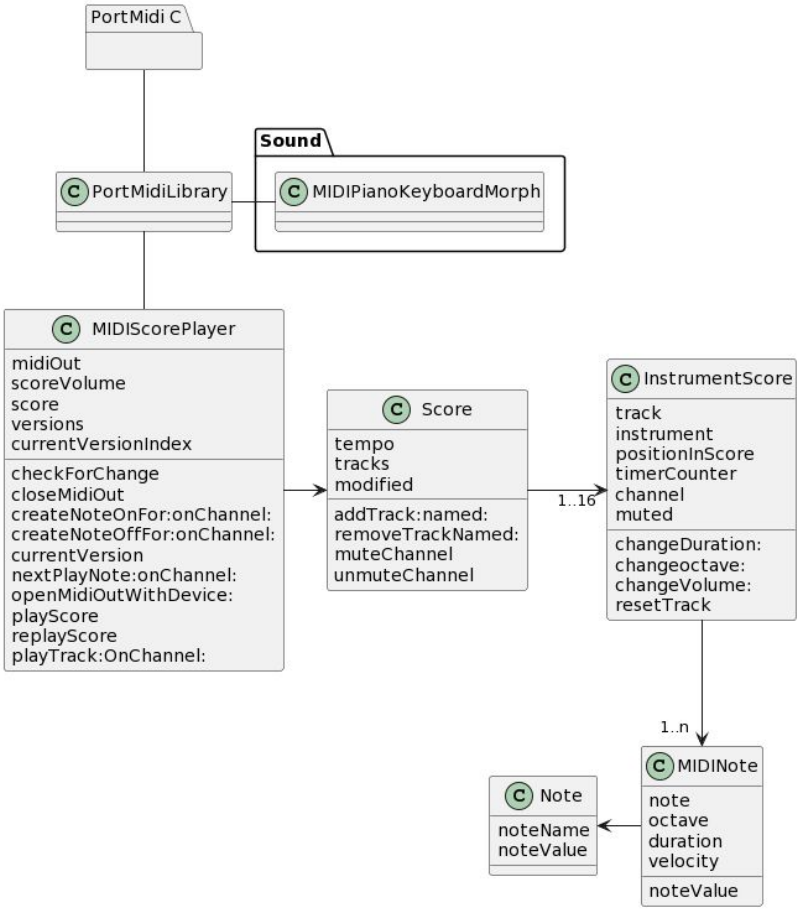
```
Object subclass: #MIDIScorePlayer
    instanceVariableNames: 'midiOut scoreVolume score versions currentVersionIndex'
    classVariableNames: 'DefaultOutput DefaultVolume'
    package: 'PortMidiDemo-Source'
```

Joue la musique en MIDI en live

Permet la modification de la partition

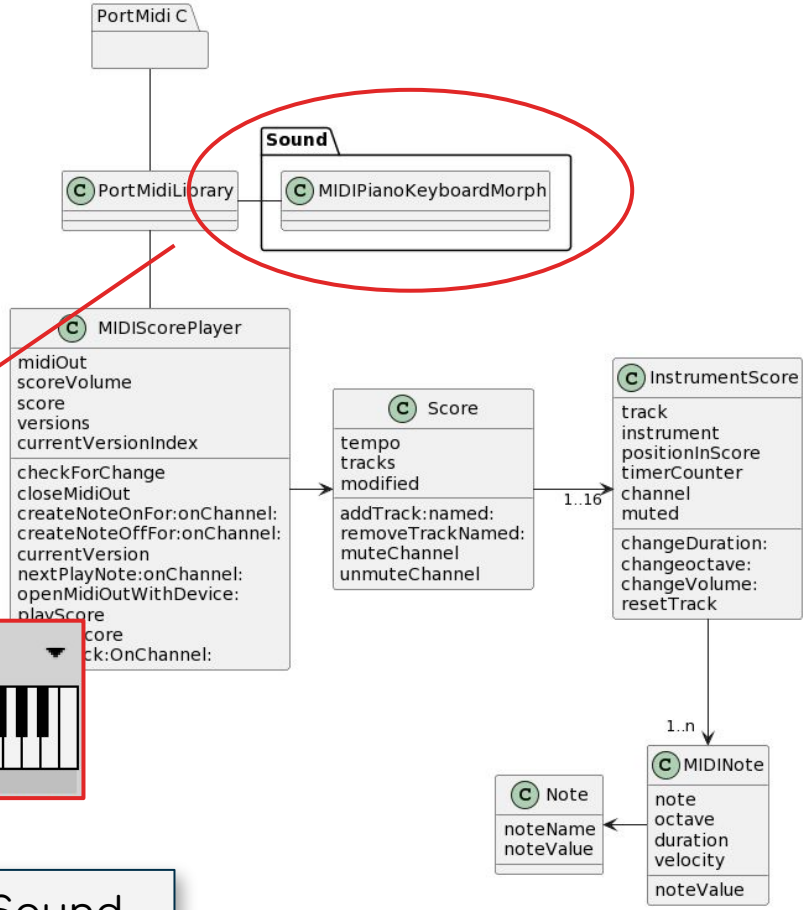
Support live coding dans Pharo

Vision globale du projet



Support live coding dans Pharo

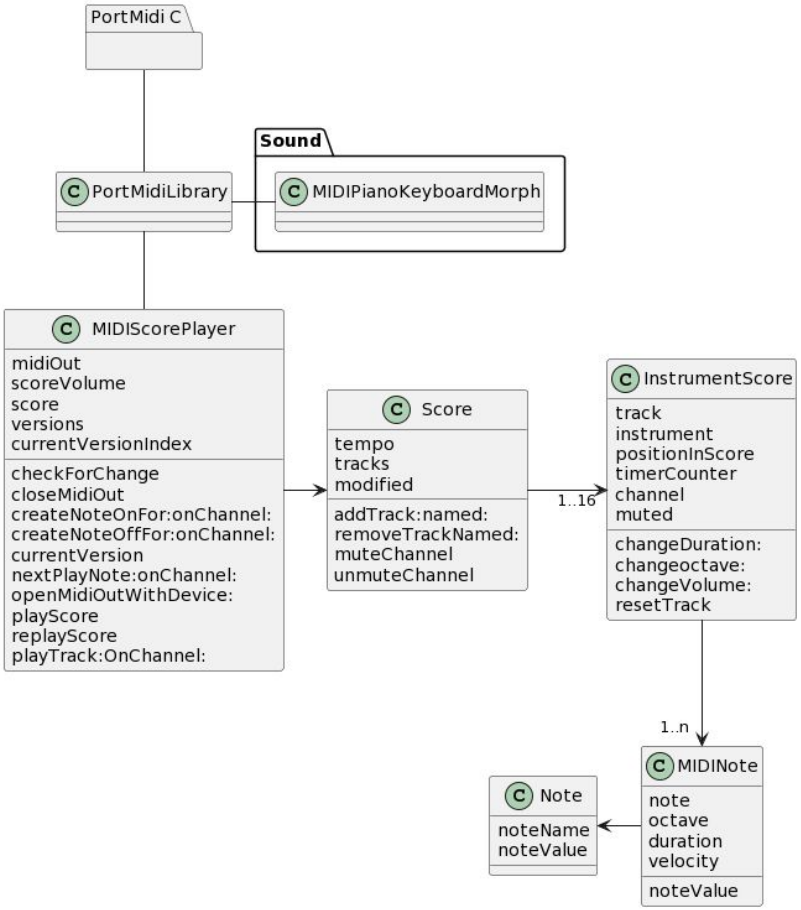
Sound



https://github.com/pharo-contributions/Sound

Support live coding dans Pharo

Vision globale du projet



Conclusion

- Support MIDI dans Pharo
- Bibliothèque C PortMidi
- Outils CMake et miniAudicle
- uFFI
- Paquet Sound

Université de Lille