A componente eletrónica da instalação está presente nos cinco canais destinados às preparações digitais, projetados por colunas situadas nos quatro cantos da instalação e uma no centro, junto ao piano. A cada canal é associado um pentagrama da partitura em manuscrito, onde é utilizada quer notação musical comum, quer notações mais alternativas e ilustrativas do gesto textural das preparações.

A cada um dos canais é associado um modelo de preparações, também referido como galeria do *bitKlaiver*, que recebe como *input* notas MIDI. Durante a fase de experimentação, as preparações eram testadas utilizando módulos criados diretamente através do software, criando galerias para cada um dos cinco canais, e simulando-os individualmente. Porém, elaborar as galerias deste modo trazia algumas complicações. Primeiro, em praticalidade, pois introduzir manualmente todas as opções de cada uma das preparações consome muito tempo; segundo, em estabilidade, uma vez que algumas das galerias construídas desapareciam ou ficavam corrompidas e o software processava algumas excepções terminando o programa e descartando todo o progesso; terceiro, em performance, porque não só o sotware fica ligeiramente mais lento ao possuir diversas galerias no seu diretório, como dispende-se de tempo a trocar de galeria a cada peça e conduz a mais enganos em tempo real. Além disso, sendo estes modelos criados estaticamente, não havia espaço para a composição expontânea e randomizada de preparações, limitando toda a potência do software na instalação.

Assim, como solução a esta insuficiência artística e prática das galerias estáticas, elaborei uma biblioteca em C++, para a criação de galerias em tempo real, recorrendo a vários métodos que automatizam a produção de instâncias de preparações, cujos parâmetros eram rápida e aleatoriamente introduzidos, dando espaço à criação de modelos em tempo real. Uma vez que as funções diferentes do programa produziam todas as galerias necessárias para as desassete obras, todas essas galerias têm o mesmo nome e vão-se sobrepondo umas às outras, sendo assim apenas necessário recarregar o software e as novas galerias ficam automaticamente disponíveis.

Adicionalmente, na mesma biblioteca, desenvolvi a funcionalidade de criar sons sintetizados que podem ser utilizados como *samples* que definem um instrumento a ser utilizado numa Gallery. Vários tipos de onda e parâmetros harmónicos podem constituir esse sintetisador, assim como envelopes ADSR e alterar a excitação harmónica em função da

amplitude.

```cpp
#include "bitklavier.hpp"

using namespace bkr;

int main(int argc, char* argv[])
{
    int complexity = 800;
    BitKlavier* bk = new BitKlavier(XAttribute::bitKlavierAttributes.Name("TestGallery"));
    XDirect* direct = bk->CreateDirect();
    XKeymap* keymap = bk->CreateKeymap();
    direct->RandomDirect(complexity);
    keymap->RandomKeymap(complexity);
    keymap->Connect(direct);
    bk->Save();
}
```