

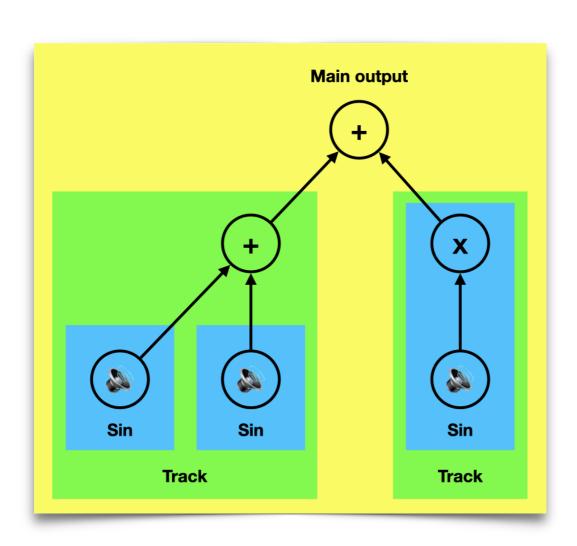








```
// Make track one
std::vector<std::unique ptr<Node>> trackOneClipNodes;
trackOneClipNodes.push back (std::make unique<SinNode> (220.0f, 1));
trackOneClipNodes.push back (std::make unique<SinNode> (220.0f, 1));
auto trackOneNode = std::make unique<SummingNode> (std::move (trackOneClipNodes));
// Make track two
auto trackTwoClipNode = std::make_unique<SinNode> (220.0f, 1);
float clipGain = 1.0f;
auto trackTwoNode = std::make unique<GainNode> (std::move (trackTwoClipNode),
                                                [clipGain] { return clipGain; });
// Make main output node
std::vector<std::unique ptr<Node>> trackNodes;
trackNodes.push back (std::move (trackOneNode));
trackNodes.push_back (std::move (trackTwoNode));
auto mainOutput = std::make_unique<SummingNode> (std::move (trackNodes));
// Play mainOutput!
```



std::vector<std::unique_ptr<

trackOneClipNodes.push_back

uniau

uniqu

> (std::move (trackOneClipNodes));

trackNodes.push_back (

uniqu

std::move (trackTwoClipNode),

unique<

std::vector<std::unique_ptr<Node>> track

unique<

auto trackTwoClip

```
> (std::move (trackNodes));
```

 \sim e>> trackOneClip

// Make track one

e<GainNode>

// Make main output node

auto trackTwoNode

Make track two

auto trackOneNode

auto mainOutput

std::move (trackOneNode)

e<SinNode> (220.0f,

SummingNode

// Play mainOutput!

std::move (trackTwoNode));



float clipGain = 1.0f;

[clipGain] { return clipGain; })

SummingNode





e<SinNode> (220.0f,















// Make track one

e<GainNode>

// Make main output node

auto trackTwoNode

Make track two

auto trackOneNode

auto mainOutput

std::move (trackOneNode)

e<SinNode> (220.0f,

SummingNode

// Play mainOutput!

std::move (trackTwoNode));



float clipGain = 1.0f;

[clipGain] { return clipGain; })

SummingNode





e<SinNode> (220.0f,







