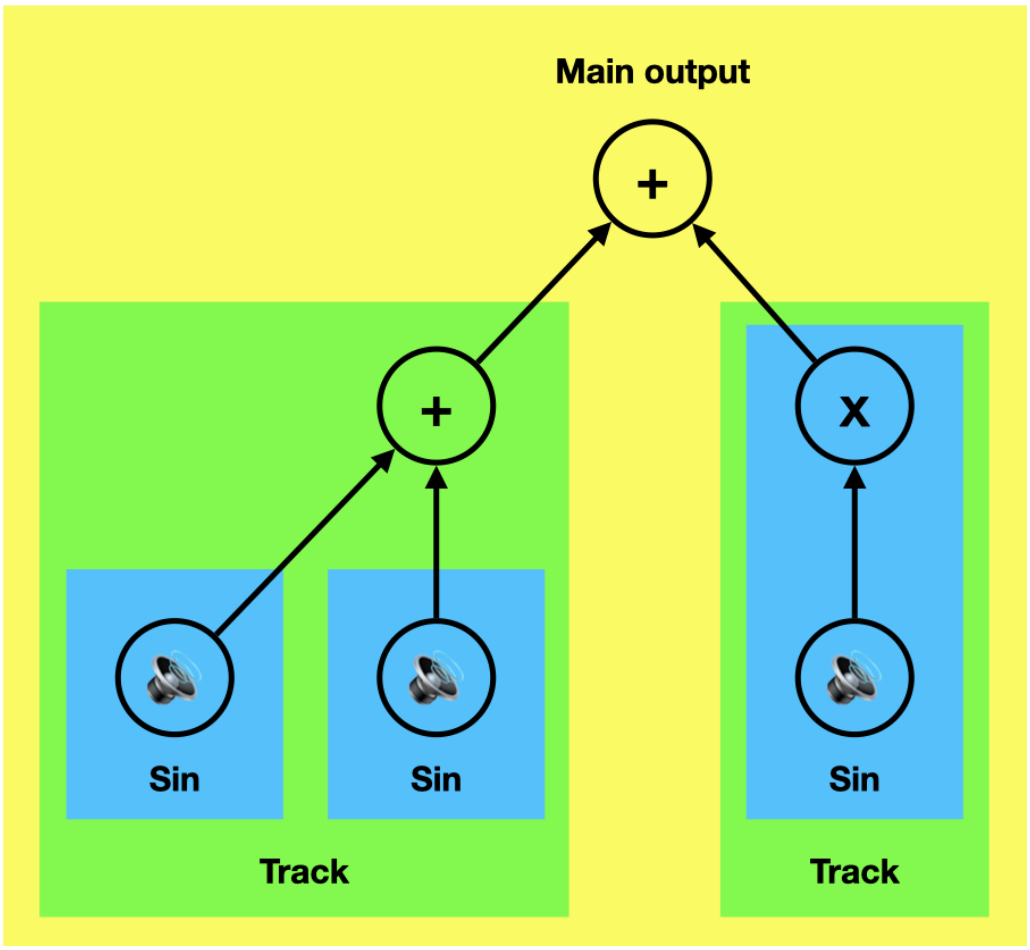

Main output
+
+
x
Sin
Sin
Sin
Track
Track

Main output

+

+

x

Sin

Sin

Sin

Track

Track

```
// Make track one
```

```
float clipGain = 1.0f;
```

```
// Make track two
```

```
[clipGain] { return clipGain; });
```

```
// Make main output node
```

```
trackOneClipNodes.push_back (std::make_unique<SinNode> (220.0f, 1));
```

```
std::vector<std::unique_ptr<Node>> trackOneClipNodes;
```

```
auto trackTwoClipNode = std::make_unique<SinNode> (220.0f, 1);
```

```
trackNodes.push_back (std::move (trackOneNode));
```

```
auto trackTwoNode = std::make_unique<GainNode> (std::move (trackTwoClipNode),
```

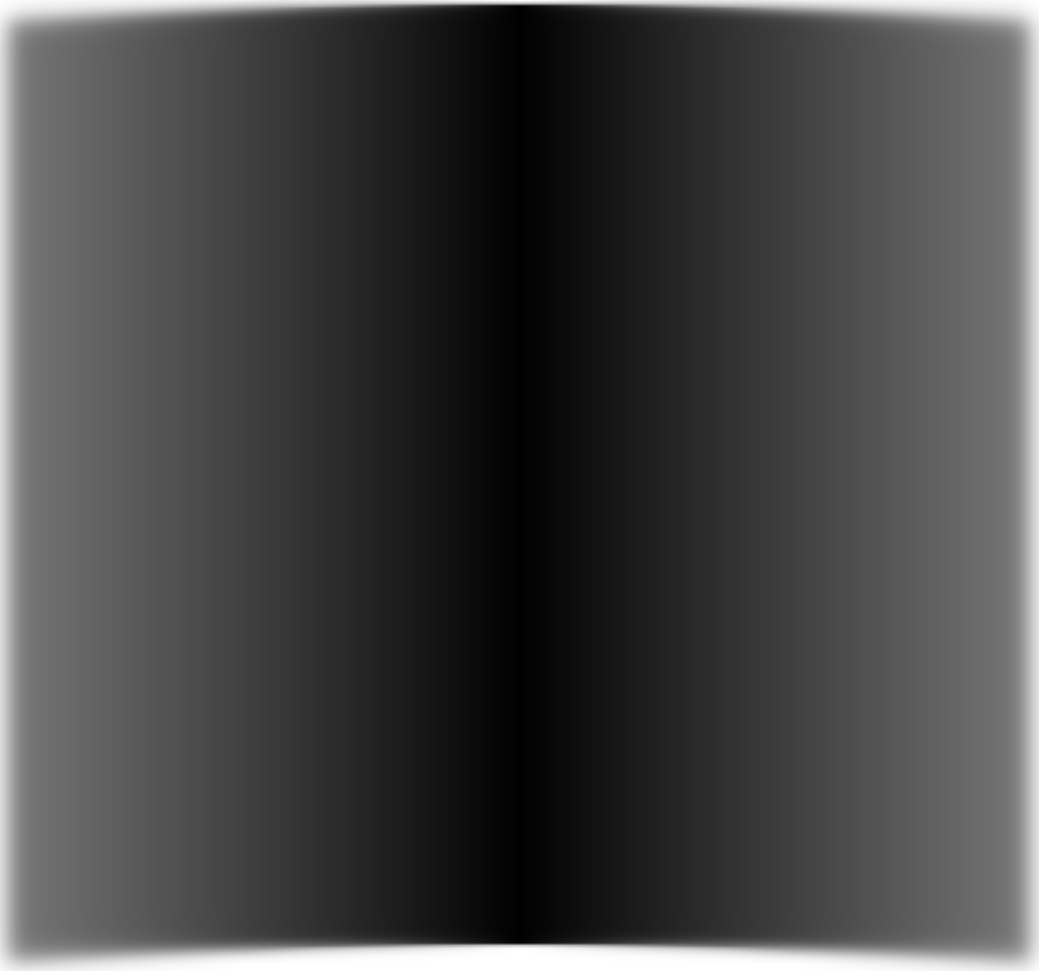
```
auto mainOutput = std::make_unique<SummingNode> (std::move (trackNodes));
```

```
auto trackOneNode = std::make_unique<SummingNode> (std::move (trackOneClipNodes));
```

```
std::vector<std::unique_ptr<Node>> trackNodes;
```

```
// Play mainOutput!
```

```
trackNodes.push_back (std::move (trackTwoNode));
```



Main output

+

+

x

Sin

Sin

Sin

Track

Track