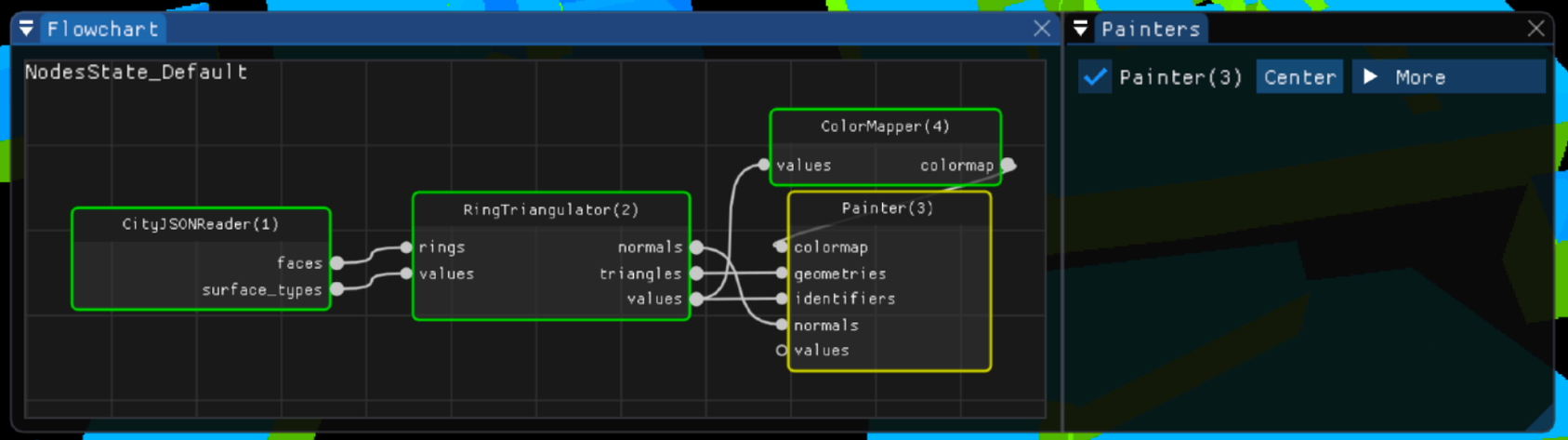
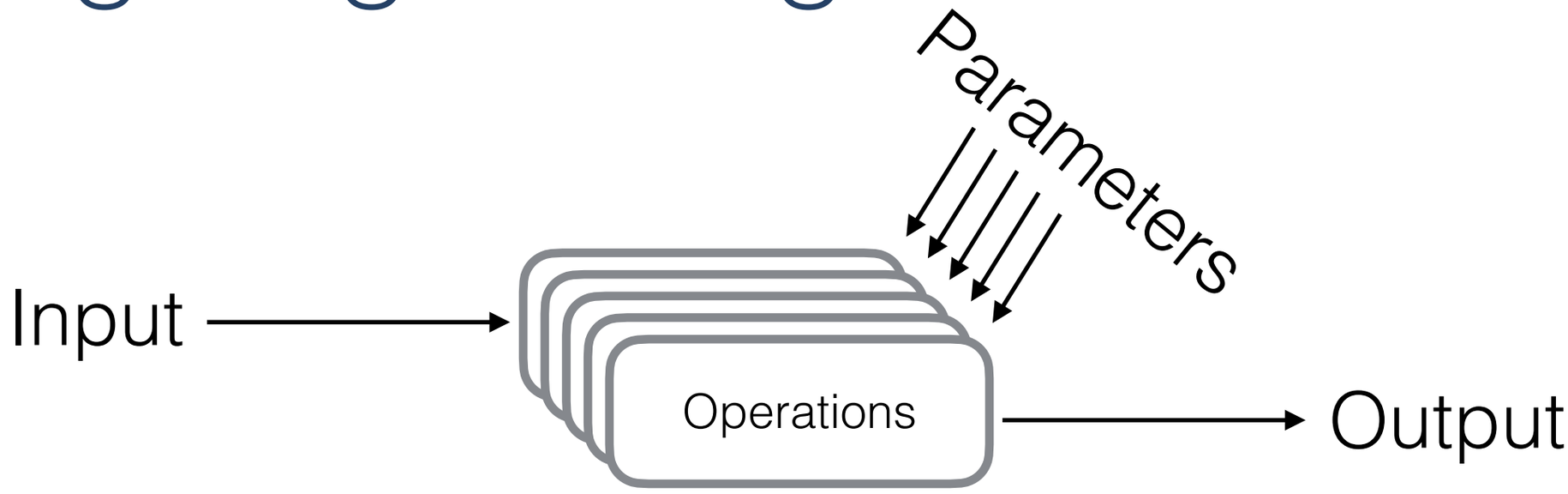
Geoflow

Ravi Peters

Delft, 26-02-2019

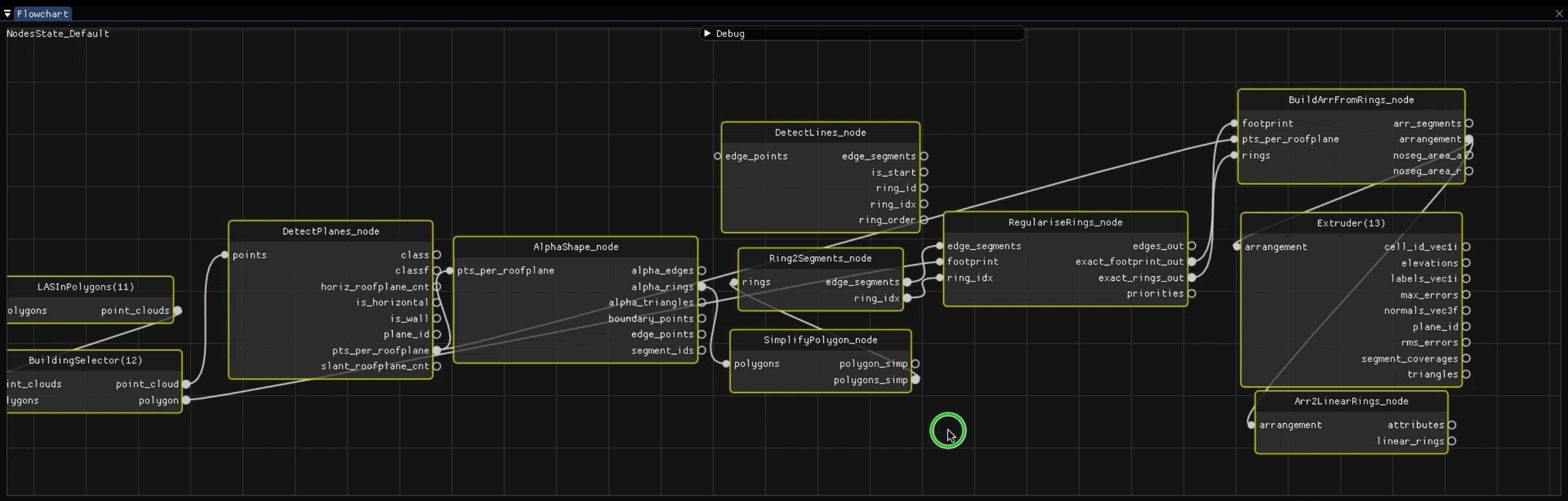


Designing 3D algorithms



can be a painful process…

1. Parameter handling & tuning
2. Visualisation
3. Managing different alternatives in the pipeline
4. Data conversions, I/O

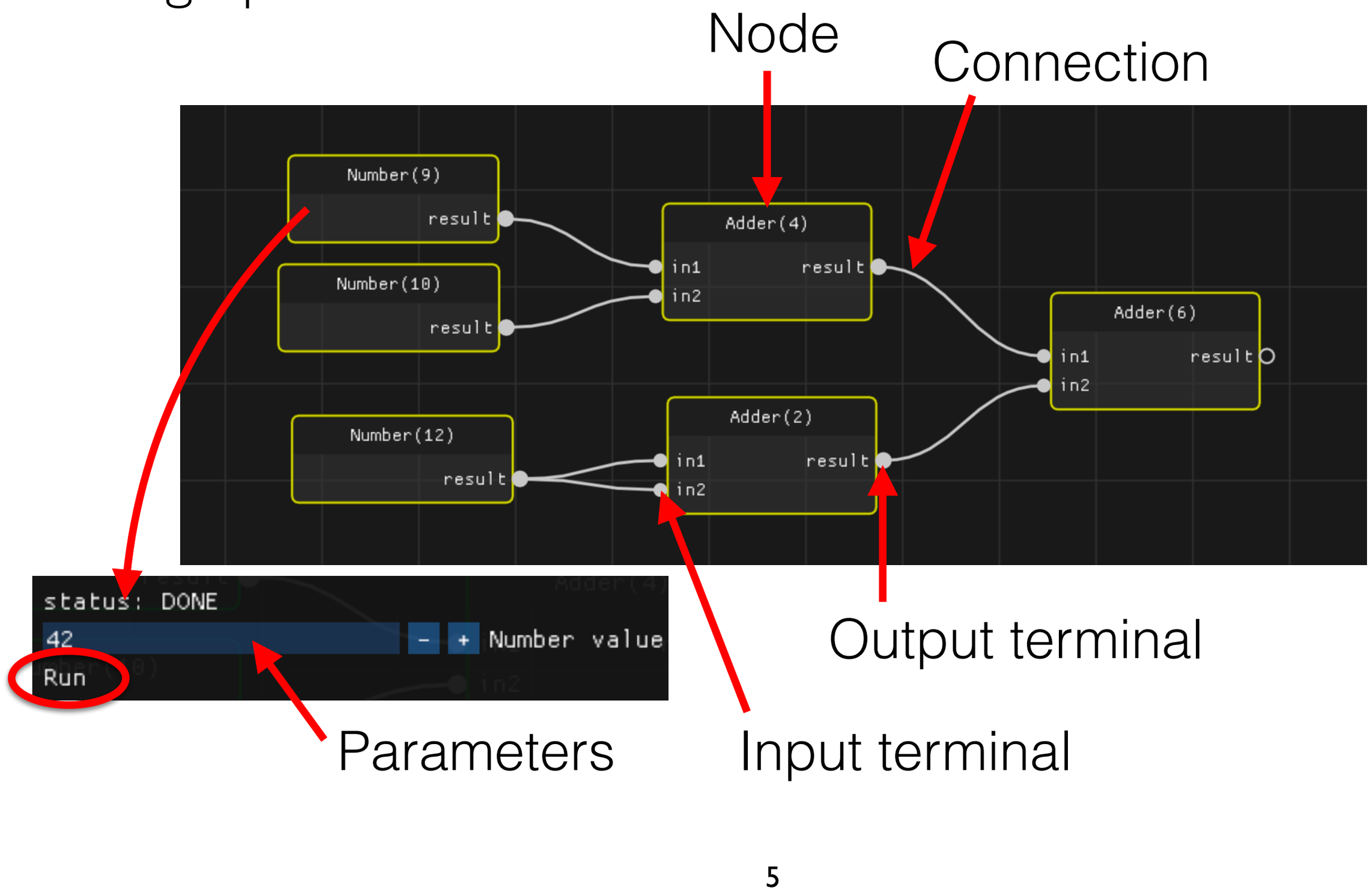

Flowchart
NodesState_Default
Debug
BuildArrFromRings_node
footprint
pts_per_roofplane
rings
arr_segments
arrangement
noseg_area_a
noseg_area_r
DetectLines_node
edge_points
edge_segments
is_start
ring_id
ring_idx
ring_order
RegulariseRings_node
edge_segments
footprint
ring_idx
edges_out
exact_footprint_out
exact_rings_out
priorities
Extruder(13)
arrangement
cell_id_vec1i
elevations
labels_vec1i
max_errors
normals_vec3f
plane_id
rms_errors
segment_coverages
triangles
DetectPlanes_node
points
class
classf
horiz_roofplane_cnt
is_horizontal
is_wall
plane_id
pts_per_roofplane
slant_roofplane_cnt
AlphaShape_node
pts_per_roofplane
alpha_edges
alpha_rings
alpha_triangles
boundary_points
edge_points
segment_ids
Ring2Segments_node
rings
edge_segments
ring_idx
SimplifyPolygon_node
polygons
polygon_simp
polygons_simp
LASInPolygons(11)
olygons
point_clouds
BuildingSelector(12)
int_clouds
lygons
point_cloud
polygon
Arr2LinearRings_node
arrangement
attributes
linear_rings
3D Viewer
Painters

Geoflow features

- Cross platform (windows, linux, mac)
- Built in C++17
- Minimal dependencies (core uses only stdlib)
- Easy to add and re-use *nodes*
- Interactive GUI:
  - 3D visualisation everywhere in the pipeline
  - Immediate feedback on parameter change
  - Rerun only what changes
- Execute flowcharts without GUI on headless server
- Flowcharts completely serialisable

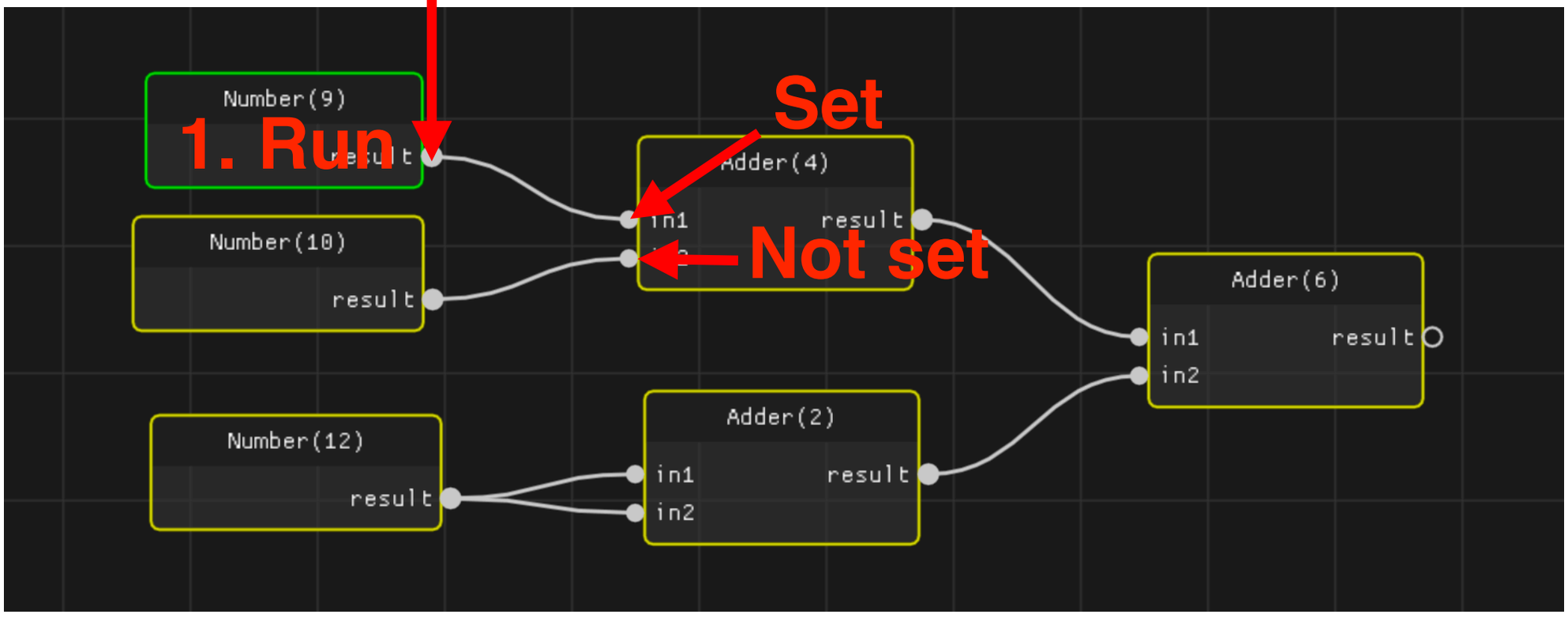
The internals

It's a graph



The internals

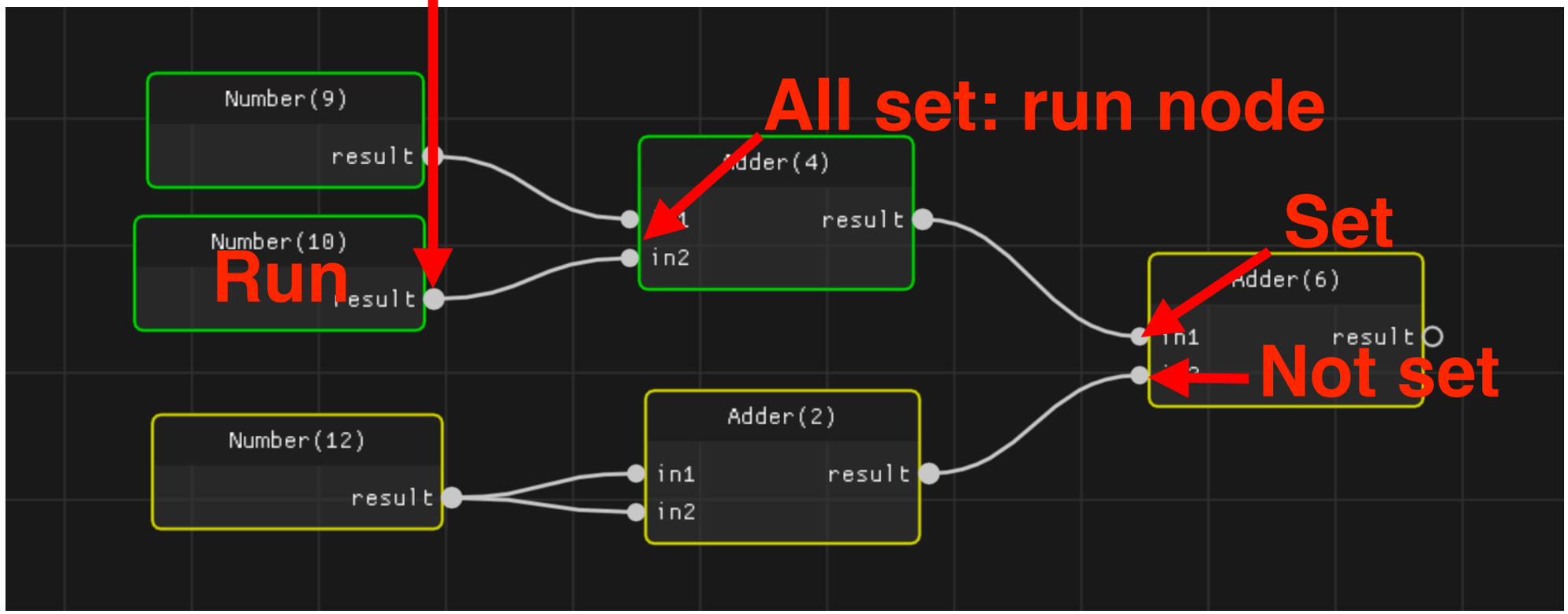
2. Push results to outputs



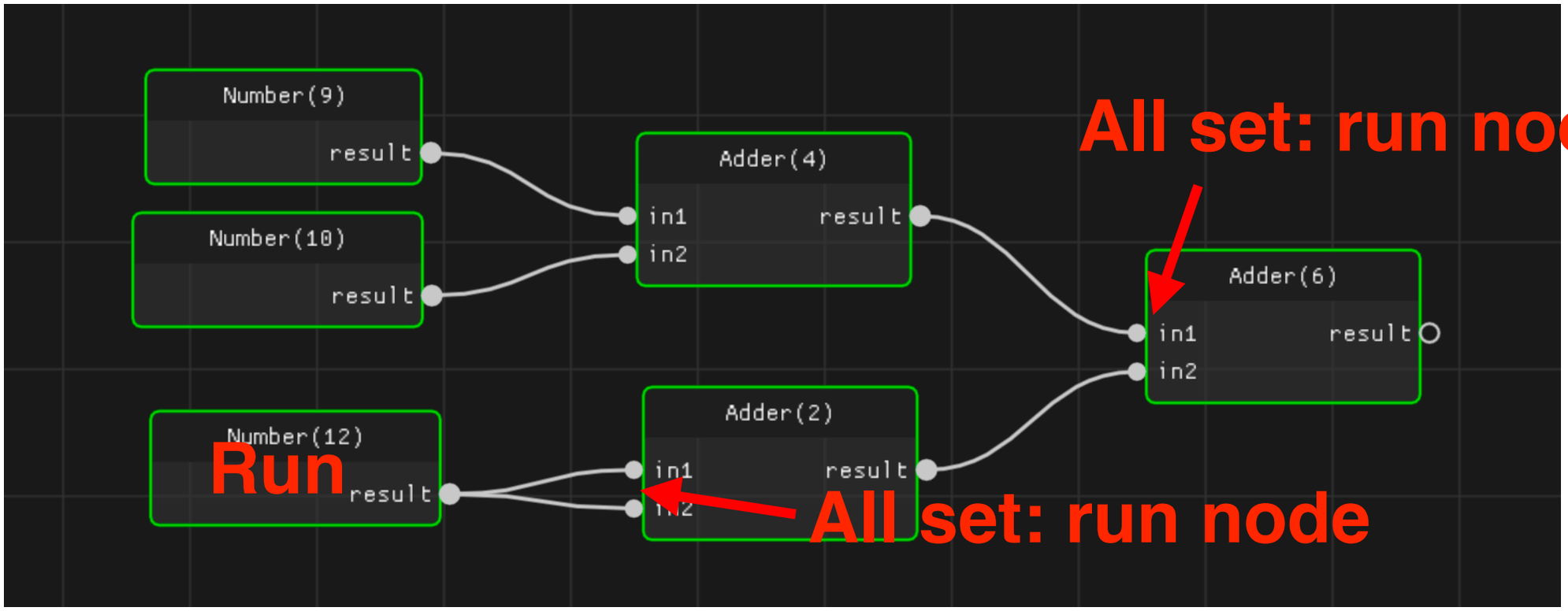
3. Check for child nodes if all inputs are set

The internals

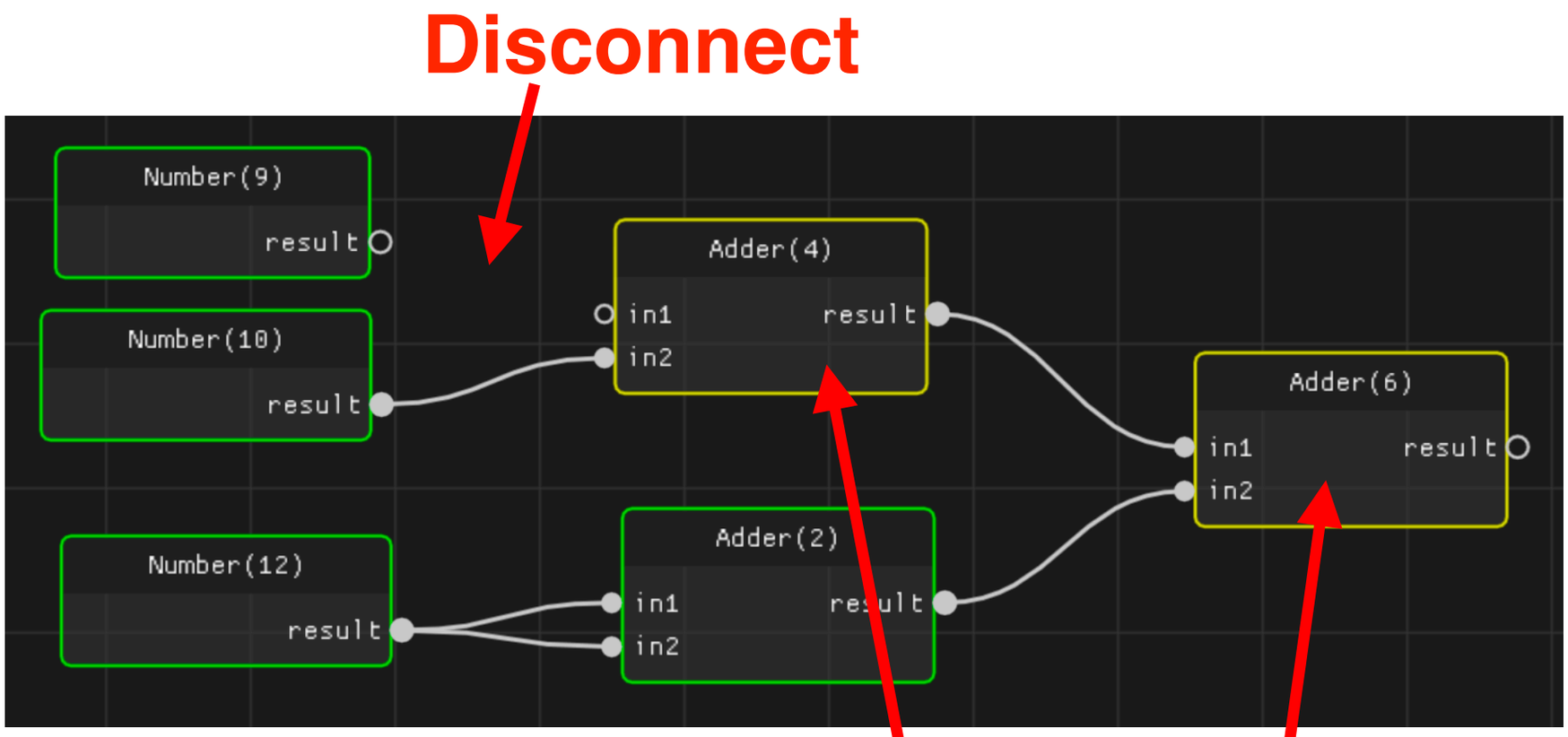
Push results to outputs



The internals



The internals

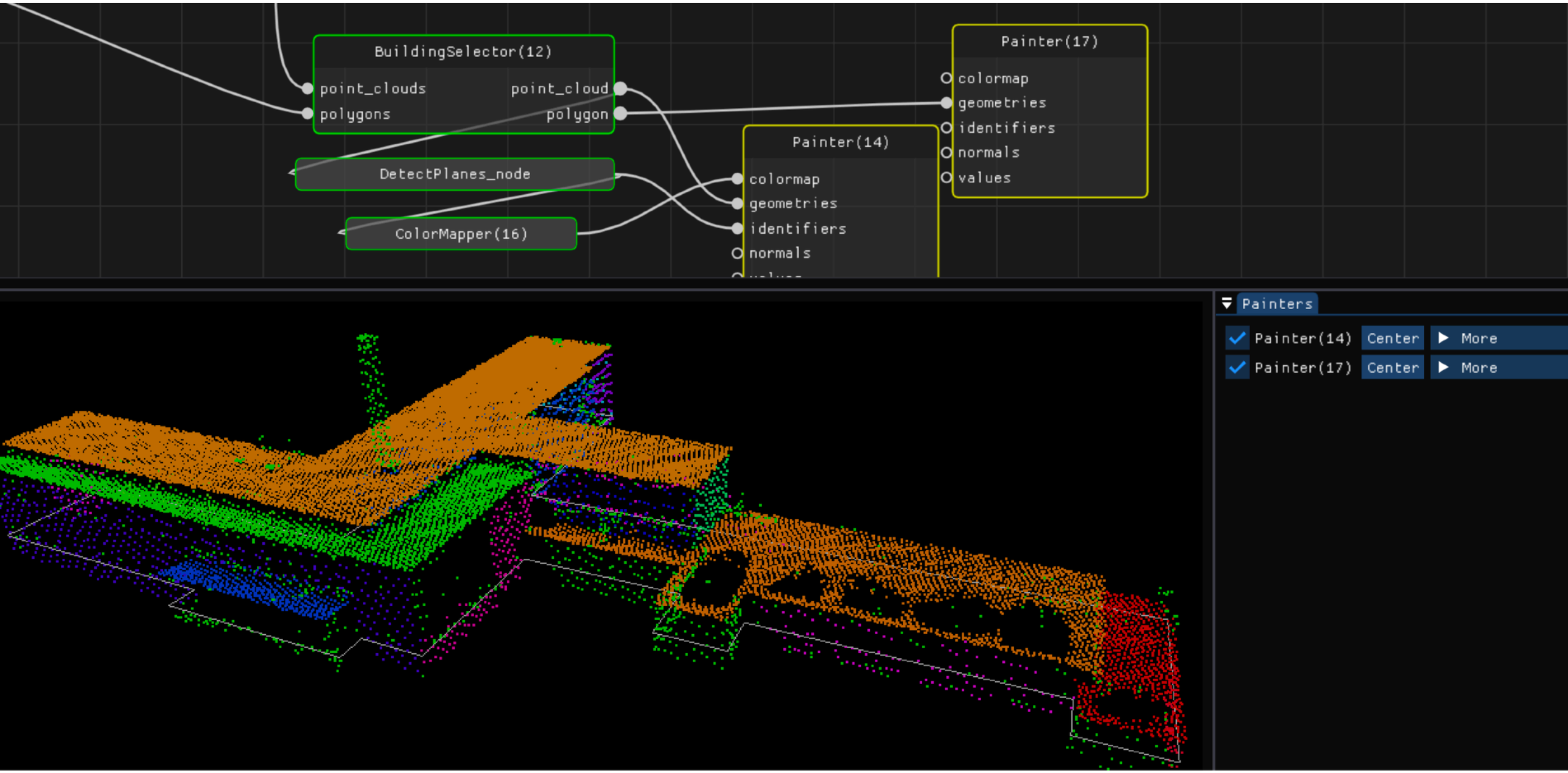


Geoflow rules

- Input can only have 1 connection
- Output can have multiple connections
- Loops are forbidden
- Connect/disconnect events immediately affect status of child nodes
- If all inputs are set, a node will automatically run
- Only nodes without inputs need to be ran manually
- Re-running a node will also re-run all child nodes

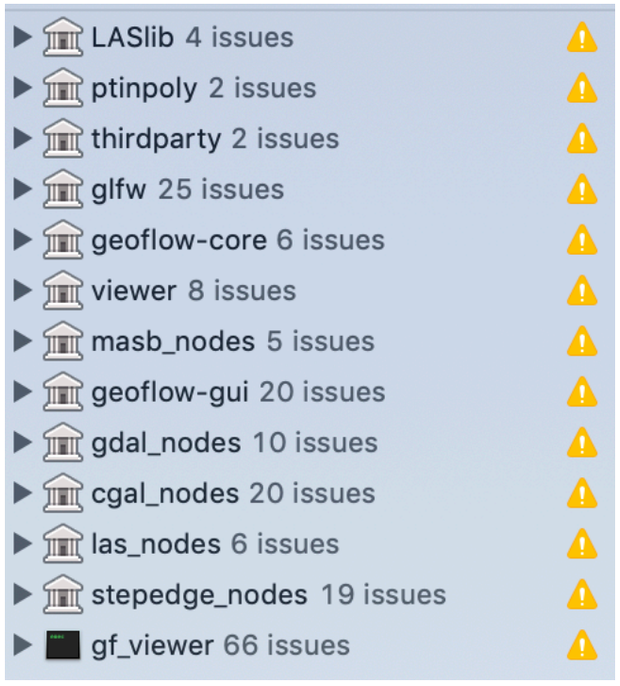
Geoflow painters

- Visualise any Geoflow geometry (eg 3D shapefile)



Status

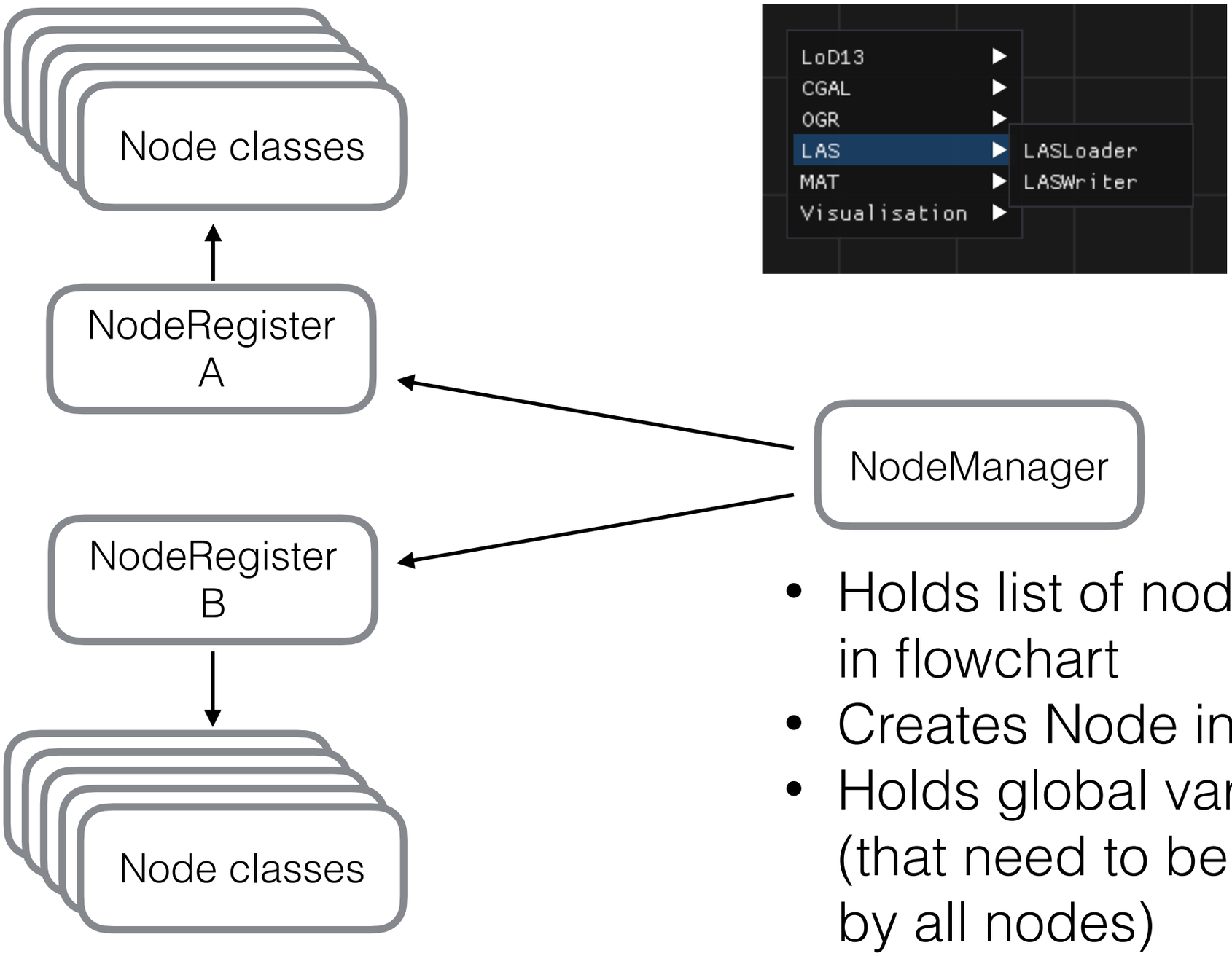
- A work in progress
- Tested on mac/windows and linux
- Todo list: https://github.com/tudelft3d/geoflow/issues/1



Some available nodes

- OGR reader/writer (vector)
- LAS reader/writer (point cloud)
- TIN simplification
- Various CGAL-based nodes
  - Triangulation, α-shape, line simplification,
- Medial Axis computation
- Plane detection, point in polygon

Node management



- Holds list of nodes instances in flowchart
- Creates Node instances
- Holds global variables (that need to be accessible by all nodes)

Example

```cpp
#include <geoflow/gui/flowchart.hpp>
#include <geoflow/core/geoflow.hpp>

#include <cityjson_nodes.hpp>
#include <utility_nodes.hpp>

namespace gfn = geoflow::nodes;

int main(int ac, const char * av[])
{
    geoflow::NodeManager N;

    // register nodes
    NodeRegister utility = gfn::utility::create_register();
    NodeRegister cityjson = gfn::cityjson::create_register();

    // create nodes
    NodeHandle cjreader = N.create_node(cityjson, "CityJSONReader", {200,100});
    NodeHandle ringtri = N.create_node(utility, "RingTriangulator", {600,100});

    // connect terminals
    connect(
        cjreader->output("faces"),
        ringtri->input("rings")
    );
    connect(
        cjreader->output("surface_types"),
        ringtri->input("values")
    );

    // Run the flowchart (no GUI required)
    // N.run(cjreader);

    // launch GUI
    launch_flowchart(N, {cityjson, utility});
}
```

coordinates in GUI flowchart

Does that sound interesting?

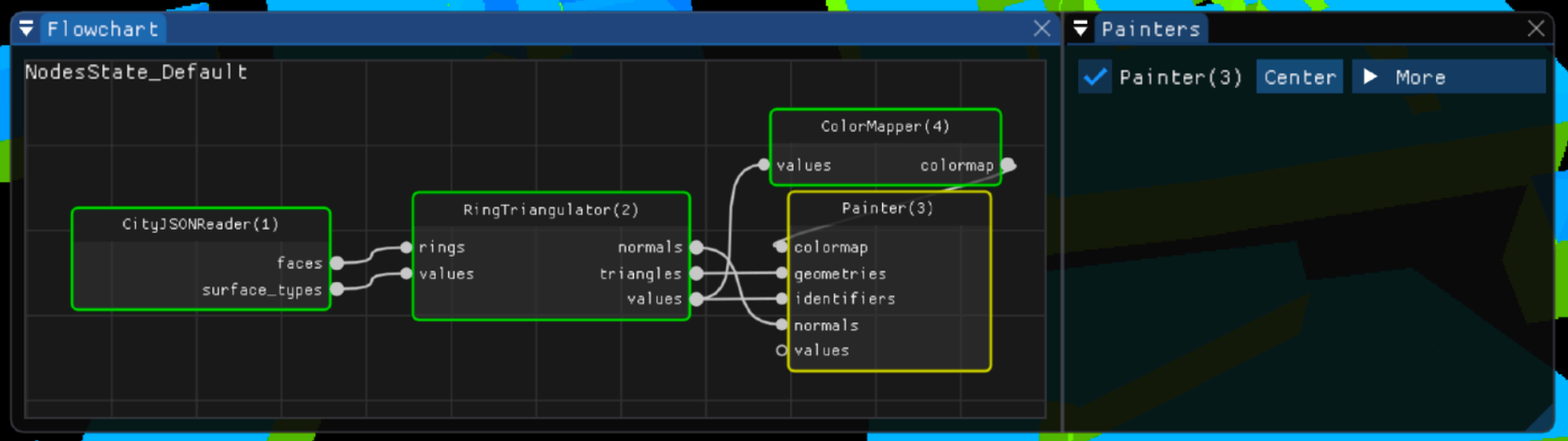
You are welcome to contribute

- Ideas and use cases
- Bug reports
- Pull requests

Thanks!

https://github.com/tudelft3d/geoflow-nodes

https://github.com/tudelft3d/geoflow



Details

- Use modern C++
  - Smart pointers, std::any/variant
- Why C++?
  - Most libs we care about are C++, bindings are often lagging behind and not feature complete
  - Fast, portable
- GUI built with
  - GLFW - https://www.glfw.org
  - Imgui - https://github.com/ocornut/imgui

Open problems

- Node management
  - As plugins/add-ons? Centrally managed