
OpenSpace User Meeting 2026
Track 2 – Multichannel Display Environments

SGCT

configurations

C-Troll

control applications in a distributed computing environment



SGCT – What is it?

- Simple Graphics Cluster Toolkit
  - https://github.com/sgct/sgct
  - https://sgct.readthedocs.io
- Used in both OpenSpace and C-Play
- Synchronized rendering in clustered environment
- Supports many projection types, computer environments and stereoscopic options
  - Examples: planar, cubemap, cylindrical, equirectangular, fisheye, spherical mirror
  - Examples: active, checkerboard, anaglyph, side-by-side, top-bottom
- Sharing via NDI and SPOUT
- *Theoretical* VR support

Note: Simpler configurations for OpenSpace can be created in the OpenSpace Launcher



SGCT – Key concept

- Each config file consists of one *cluster*
  - Each *cluster* contains one or more *nodes*
    - Each *node* contains one or more *windows*
      - Each *window* contains one or more *viewports*
        - Each *viewport* contains one or more *projections*

- Each of these layers also have options specific to them

```
{
  "version": 1,
  "masteraddress": "localhost",
  "nodes": [
    {
      "address": "localhost",
      "port": 20400,
      "windows": [
        {
          "size": { "x": 1280, "y": 720 },
          "viewports": [
            {
              "projection": {
                "type": "PlanarProjection",
                "fov": {
                  "hfov": 80.0,
                  "vfov": 50.534015846724
                }
              }
            }
          ]
        }
      ]
    }
  ],
  "users": [
    {
      "eyeseparation": 0.065,
      "pos": { "x": 0.0, "y": 0.0, "z": 0.0 }
    }
  ]
}
```

SGCT – Simple example

- *masteraddress*: local ip of the master computer
- *nodes*: list of *nodes* that exists in the *cluster*
  - Includes master computer
- *address*: local ip of the specific *node*
- *port*: network port used for communication
  - Unique and may not be used for something else
- *windows*: list of *windows* for the specific *node*
- *size*: width and height for the specific *window*
- *res*: rendering resolution
  - If specified it will decouple *window* size from rendering resolution
- *viewports*: list of viewports for the specific *window*
- *projection*: type of projection

```
{
  "version": 1,
  "masteraddress": "localhost",
  "nodes": [
    {
      "address": "localhost",
      "port": 20400,
      "windows": [
        {
          "size": { "x": 1920, "y": 1080 },
          "res": { "x": 7680, "y": 4320 },
          "viewports": [
            {
              "projection": {
                "type": "PlanarProjection",
                "fov": {
                  "hfov": 80.0,
                  "vfov": 50.534015846724
                }
              }
            }
          ]
        }
      ]
    }
  ],
  "users": [
    {
      "eyeseparation": 0.065,
      "pos": { "x": 0.0, "y": 0.0, "z": 0.0 }
    }
  ]
}
```

SGCT – Key concept

- Each config file consists of one *cluster*
  - Each *cluster* contains one or more *nodes*
    - Each *node* contains one or more *windows*
      - Each *window* contains one or more *viewports*
        - Each *viewport* contains one or more *projections*

- Each of these layers also have options specific to them

```
{
  "version": 1,
  "masteraddress": "localhost",
  "nodes": [
    {
      "address": "localhost",
      "port": 20400,
      "windows": [
        {
          "size": { "x": 1280, "y": 720 },
          "viewports": [
            {
              "projection": {
                "type": "PlanarProjection",
                "fov": {
                  "hfov": 80.0,
                  "vfov": 50.534015846724
                }
              }
            }
          ]
        }
      ]
    }
  ],
  "users": [
    {
      "eyeseparation": 0.065,
      "pos": { "x": 0.0, "y": 0.0, "z": 0.0 }
    }
  ]
}
```

SGCT – Two windows

- Add another window into the ***windows*** list of the specific ***node***

```
"windows": [
  {
    "size": { "x": 1280, "y": 720 },
    "viewports": [
      {
        "projection": {
          "type": "PlanarProjection",
          "fov": {
            "hfov": 80.0,
            "vfov": 50.534015846724
          },
          "orientation": { "yaw": 20.0, "pitch": 0.0, "roll": 0.0 }
        }
      }
    ]
  },
  {
    "size": { "x": 1280, "y": 720 },
    "viewports": [
      {
        "projection": {
          "type": "PlanarProjection",
          "fov": {
            "hfov": 80.0,
            "vfov": 50.534015846724
          },
          "orientation": { "yaw": -20.0, "pitch": 0.0, "roll": 0.0 }
        }
      }
    ]
  }
]
```



```
{
  "version": 1,
  "masteraddress": "localhost",
  "nodes": [
    {
      "address": "localhost",
      "port": 20400,
      "windows": [
        {
          "size": { "x": 1280, "y": 720 },
          "viewports": [
            {
              "projection": {
                "type": "PlanarProjection",
                "fov": {
                  "hfov": 80.0,
                  "vfov": 50.534015846724
                },
                "orientation": { "yaw": 20.0, "pitch": 0.0, "roll": 0.0 }
              }
            }
          ]
        },
        {
          "size": { "x": 1280, "y": 720 },
          "viewports": [
            {
              "projection": {
                "type": "PlanarProjection",
                "fov": {
                  "hfov": 80.0,
                  "vfov": 50.534015846724
                },
                "orientation": { "yaw": -20.0, "pitch": 0.0, "roll": 0.0 }
              }
            }
          ]
        }
      ]
    }
  ],
  "users": [
    {
      "eyeseparation": 0.065,
      "pos": { "x": 0.0, "y": 0.0, "z": 0.0 }
    }
  ]
}
```

SGCT – Two windows

Important note for multiple *windows*…

- The first ***window*** **must** have the largest, or same, resolution if multiple ***windows*** are defined
  - Bug: https://github.com/OpenSpace/OpenSpace/issues/507

```
{
  "version": 1,
  "masteraddress": "localhost",
  "nodes": [
    {
      "address": "localhost",
      "port": 20400,
      "windows": [
        {
          "size": { "x": 1280, "y": 720 },
          "viewports": [
            {
              "projection": {
                "type": "PlanarProjection",
                "fov": {
                  "hfov": 80.0,
                  "vfov": 50.534015846724
                },
                "orientation": { "yaw": 20.0, "pitch": 0.0, "roll": 0.0 }
              }
            }
          ]
        },
        {
          "size": { "x": 1280, "y": 720 },
          "viewports": [
            {
              "projection": {
                "type": "PlanarProjection",
                "fov": {
                  "hfov": 80.0,
                  "vfov": 50.534015846724
                },
                "orientation": { "yaw": -20.0, "pitch": 0.0, "roll": 0.0 }
              }
            }
          ]
        }
      ]
    }
  ],
  "users": [
    {
      "eyeseparation": 0.065,
      "pos": { "x": 0.0, "y": 0.0, "z": 0.0 }
    }
  ]
}
```

SGCT – Other projections

- All projections: https://sgct.readthedocs.io/en/latest/users/configuration/projection/index.html

Equirectangular

```
{
  "version": 1,
  "masteraddress": "localhost",
  "nodes": [
    {
      "address": "localhost",
      "port": 20400,
      "windows": [
        {
          "size": { "x": 1280, "y": 720 },
          "viewports": [
            {
              "pos": { "x": 0.0, "y": 0.0 },
              "size": { "x": 1.0, "y": 1.0 },
              "projection": {
                "type": "EquirectangularProjection",
                "quality": "1k"
              }
            }
          ]
        }
      ]
    }
  ],
  "users": [
    {
      "eyeseparation": 0.06,
      "pos": { "x": 0.0, "y": 0.0, "z": 0.0 }
    }
  ]
}
```

Fisheye

```
{
  "version": 1,
  "masteraddress": "localhost",
  "nodes": [
    {
      "address": "localhost",
      "port": 20400,
      "windows": [
        {
          "size": { "x": 512, "y": 512 },
          "viewports": [
            {
              "type": "FisheyeProjection",
              "pos": { "x": 0.0, "y": 0.0 },
              "size": { "x": 1.0, "y": 1.0 },
              "projection": {
                "type": "FisheyeProjection",
                "fov": 180.0,
                "quality": "medium",
                "tilt": 0.0,
                "background": { "r": 0.1, "g": 0.1, "b": 0.1, "a": 1.0 }
              }
            }
          ]
        }
      ]
    }
  ],
  "users": [
    {
      "eyeseparation": 0.06,
      "pos": { "x": 0.0, "y": 0.0, "z": 0.0 }
    }
  ]
}
```

SGCT – Multiple nodes

- Uses same logic as we've seen before, but now we add other ***nodes*** that we have access to over the local network
  - https://sgct.readthedocs.io/en/latest/users/examples.html#two-nodes

- Added ***orientation*** which rotates our ***PlanarProjection*** such that "point our camera" at another part of the 3D scene

Especially for multiple nodes…

- Make sure that the ports are not blocked by network firewall or that they are used by another program
- Make sure that the same configuration file is copied to all computers in the cluster

```
{
  "version": 1,
  "masteraddress": "localhost",
  "nodes": [
    {
      "address": "localhost",
      "port": 20400,
      "windows": [
        {
          "size": { "x": 1280, "y": 720 },
          "viewports": [
            {
              "projection": {
                "type": "PlanarProjection",
                "fov": {
                  "hfov": 80.0,
                  "vfov": 50.534015846724
                },
                "orientation": { "yaw": 20.0, "pitch": 0.0, "roll": 0.0 }
              }
            }
          ]
        }
      ]
    },
    {
      "address": "192.168.0.101",
      "port": 20401,
      "windows": [
        {
          "size": { "x": 1280, "y": 720 },
          "viewports": [
            {
              "projection": {
                "type": "PlanarProjection",
                "fov": {
                  "hfov": 80.0,
                  "vfov": 50.534015846724
                },
                "orientation": { "yaw": -20.0, "pitch": 0.0, "roll": 0.0 }
              }
            }
          ]
        }
      ]
    }
  ],
  "users": [
    {
      "eyeseparation": 0.065,
      "pos": { "x": 0.0, "y": 0.0, "z": 0.0 }
    }
  ]
}
```

SGCT – Real dome config example

- The dome here in Norrköping
- *orientation* based on projector physical placement of the projectors



SGCT – What if…?

Q: What about projection warping?
A: Variably robust support for many ***mesh*** types:

- Recommended is Wavefront OBJ format
- Also support: SCISS, Scalable, Dome Projection CSV, Paul Bourke, SimCAD

Q: We have one projector per eye, now what?
A: Add each computer as a ***node*** and specify left or right ***eye*** for the specific ***viewport***

Q: What about blending?
A: We support using *blendmask* and *blacklevelmask*.

- *blendmask*: binary mask that removes parts of the rendered image where the mask pixel value is black
- *Blacklevelmask*: greyscale image which determines how much of the rendered image should be visible for each pixel. Black pixel means nothing of the rendered pixel value will be visible, white pixel means all will show, if it's 50% grey it means that half of the pixel value will show

OpenSpace



C-Troll

control applications in a distributed computing environment



C-Troll – What is it?

- Program used to launch and close programs in a distributed computing environment, such as a dome.
- Windows-only program
- Developed at Visualization Center C
  - https://github.com/c-toolbox/C-Troll





C-Troll – Key concept

- C-Troll setup contains program for *master* and *node* computers
  - Main ***C-Troll*** program is used as the control center from the *master* computer
  - Companion program ***Tray*** should run on all computers that communicate with ***C-Troll*** (including *master*)

- Additional ***Editor*** application can be used to create C-Troll configuration files

- Configuration files contain information about…
  - ***Nodes***: any computer that exist on the network
  - ***Clusters***: a subset of ***nodes*** which defines a group
  - ***Application***: specific configurations for programs that can be controlled

Read more here: https://github.com/c-toolbox/C-Troll/wiki/Configuration

OpenSpace



C-Troll – Nodes

- Requires…
  - *name*: used for identification when creating a *cluster*
  - *ip*: IPv4 or IPv6 of the *node*
  - *port*: which port C-Troll will communicate over

More optional settings: https://github.com/c-toolbox/C-Troll/wiki/Configuration

OpenSpace



C-Troll – Clusters

- Requires…
  - ***name***: used for identification referred to by ***applications***
  - ***nodes***: list of ***nodes*** which should be part of this ***cluster***. The ***nodes*** are identified using the ***names*** as defined in each ***node***

More optional settings: https://github.com/c-toolbox/C-Troll/wiki/Configuration



C-Troll – Applications

- Requires…
  - *name*: used for identification referred to by *applications*
  - *executable*: path to the program that should be started
  - *clusters*: list of *cluster names* that this application can run on

More optional settings: https://github.com/c-toolbox/C-Troll/wiki/Configuration





Demo

C-Troll

Demo

C-Troll, SGCT & OpenSpace

Tips, Tricks and Good-to-knows

- C-Troll:
  - You can define scripts as ***applications*** in C-Troll
  - Make most of ***applications*** in C-Troll by being smart with which arguments go in the ***parameter*** field for ***clusters*** vs ***configurations***

- SGCT:
  - First ***window*** must have the greatest resolution if more than one ***window*** is created per ***node***

- General:
  - Use *robocopy* script to sync data from master to nodes
    - Usefull for SGCT configs, scripts and OpenSpace ***Sync*** and ***User*** folders

OpenSpace

