

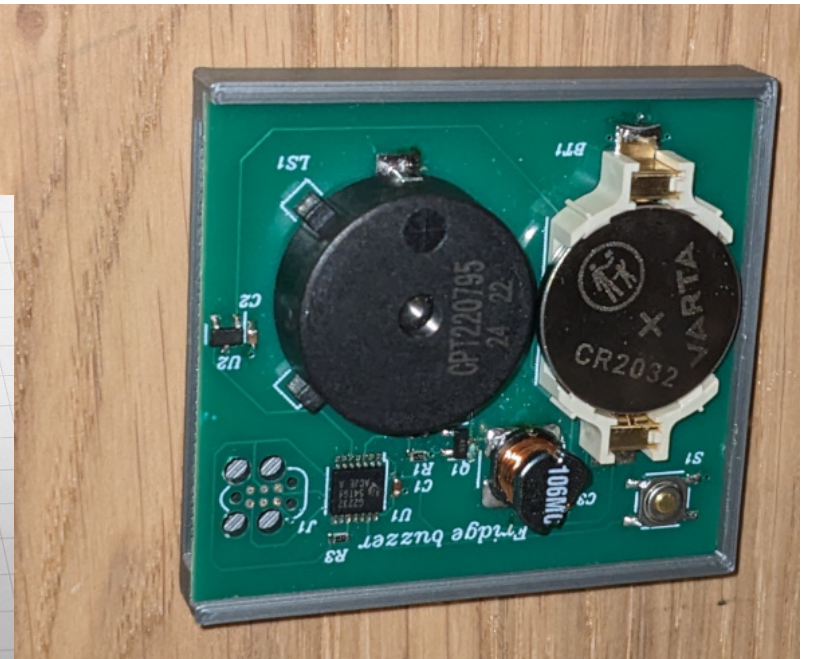
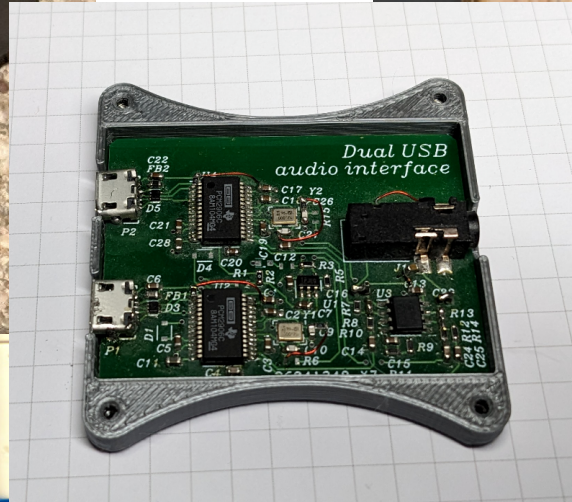
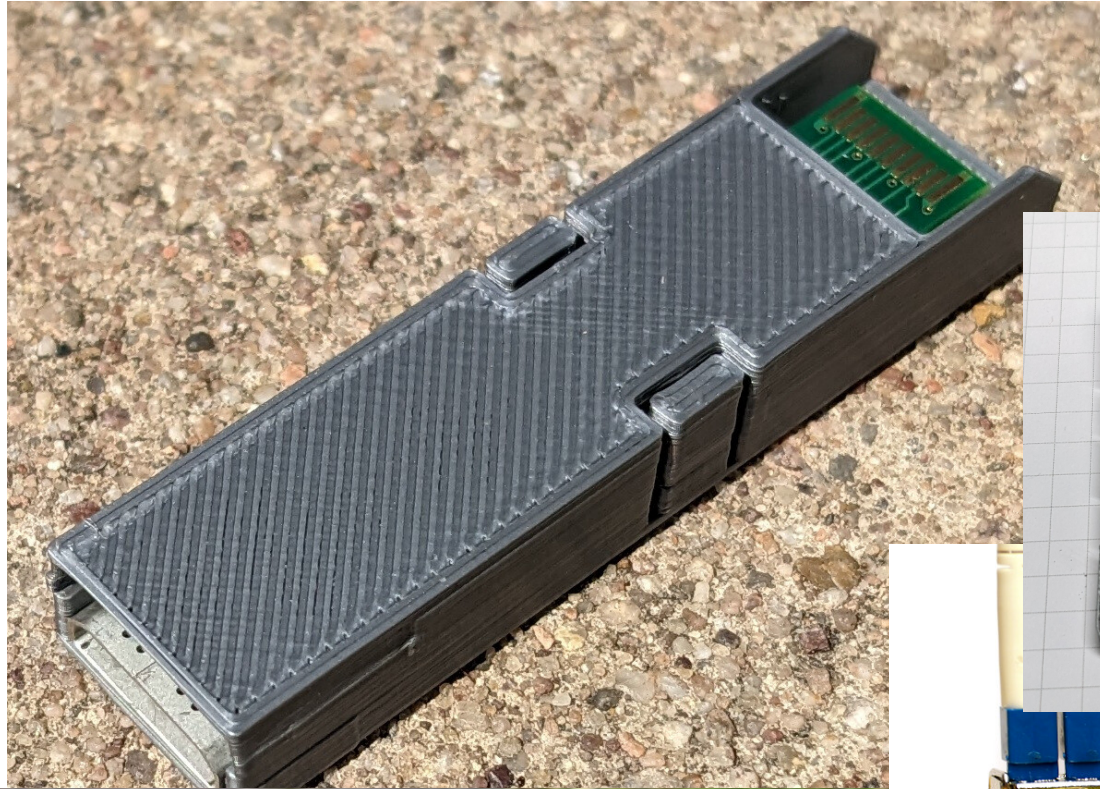
# Dune 3D

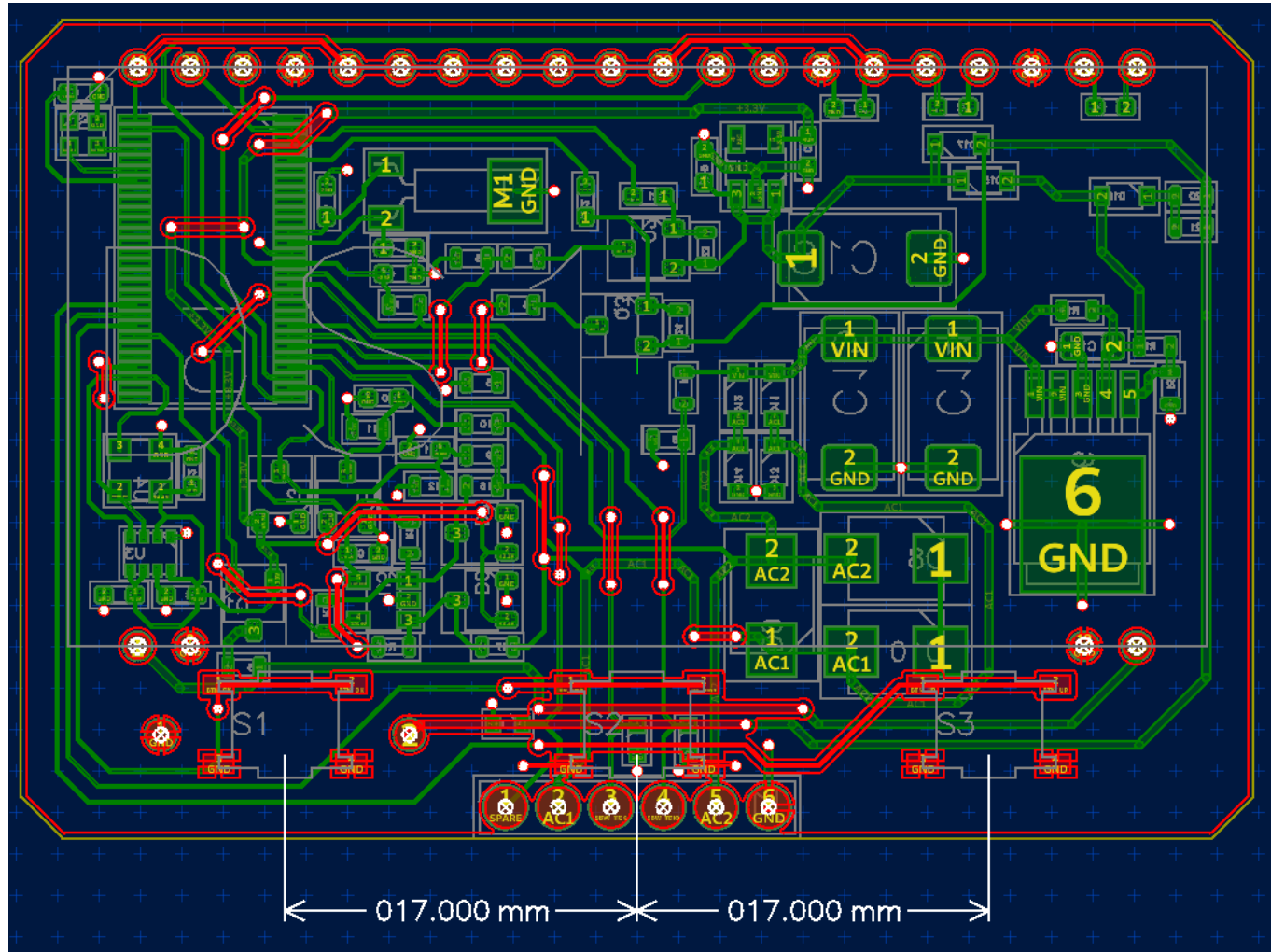
The making of a maker's tool

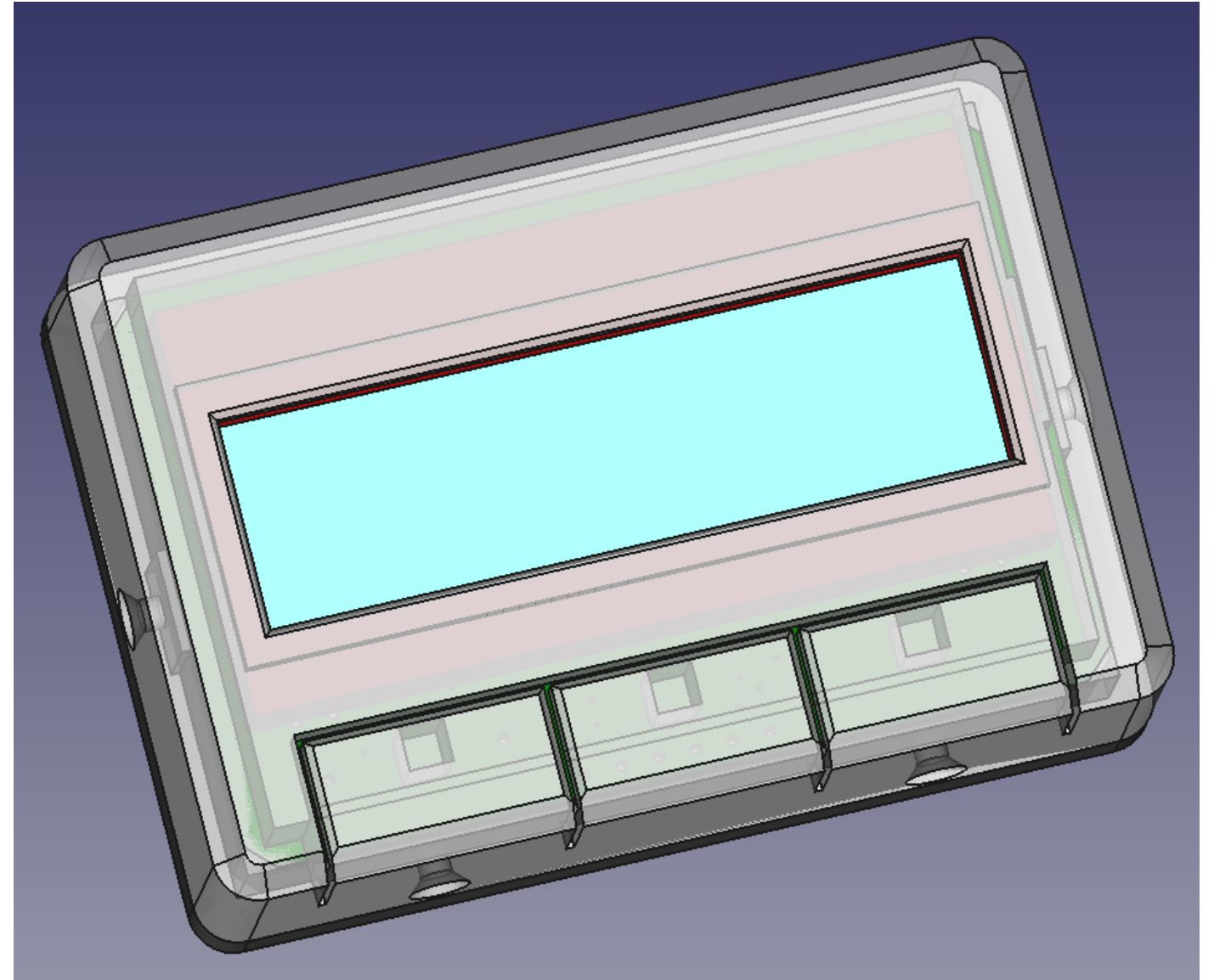
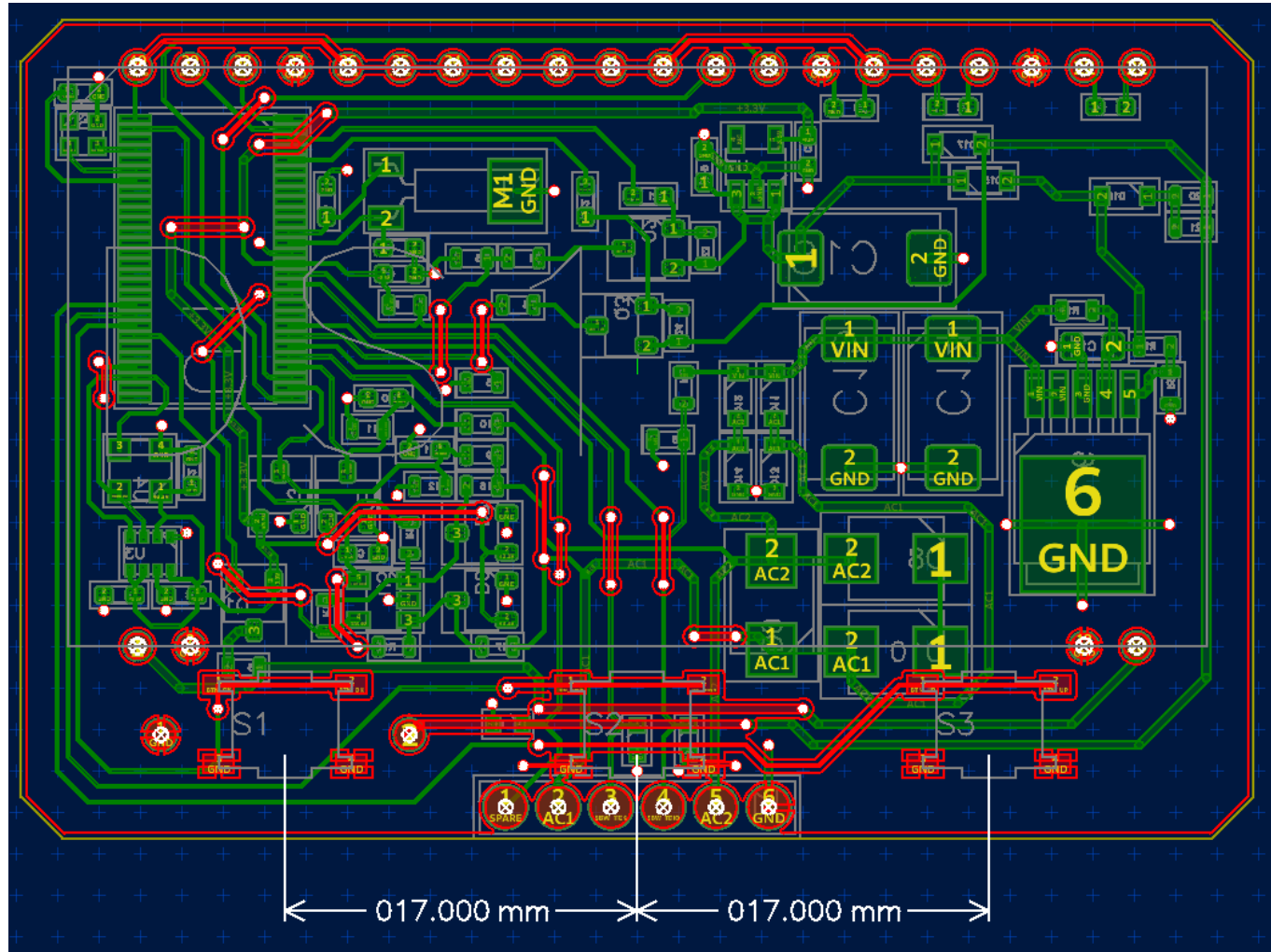
Lukas

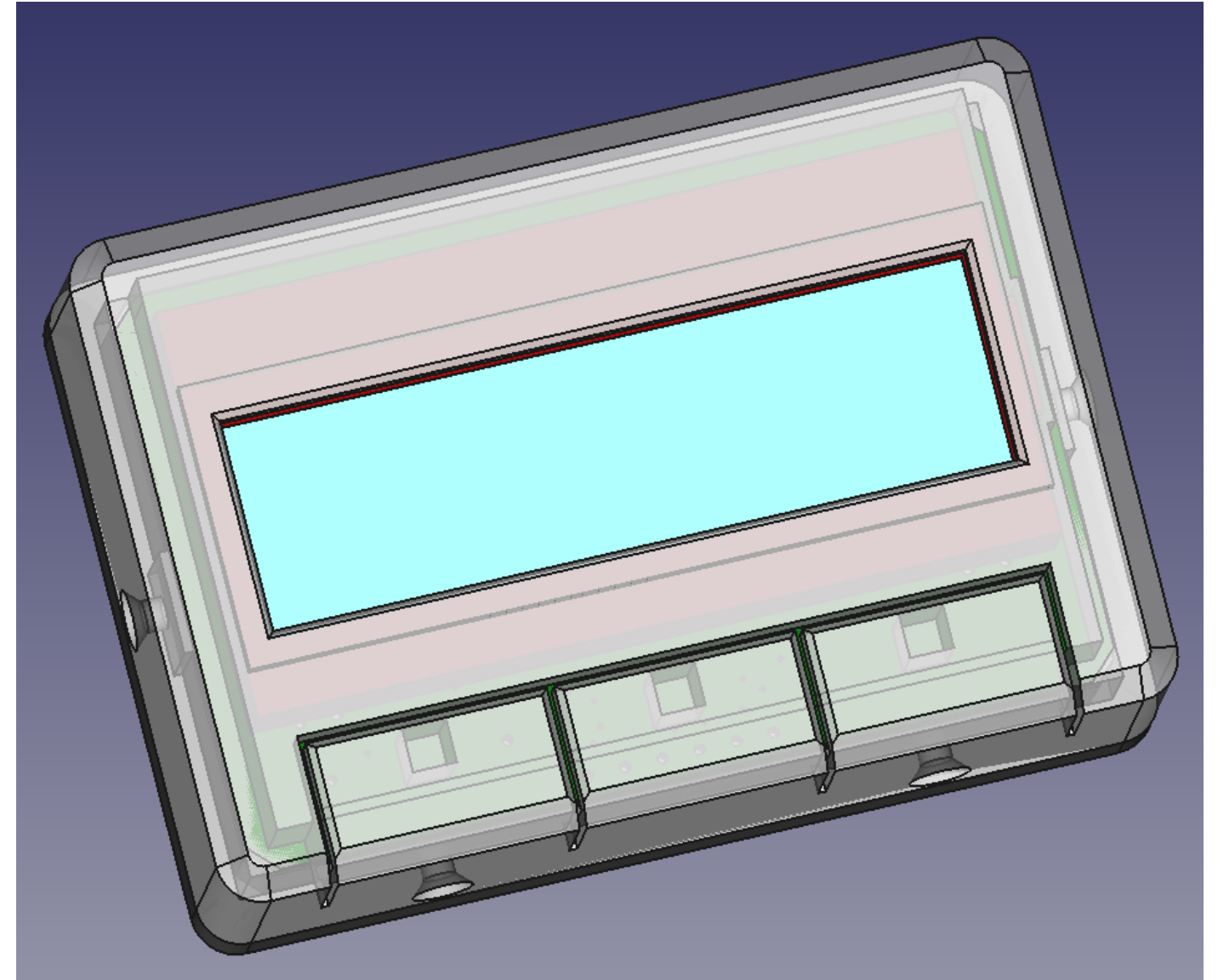
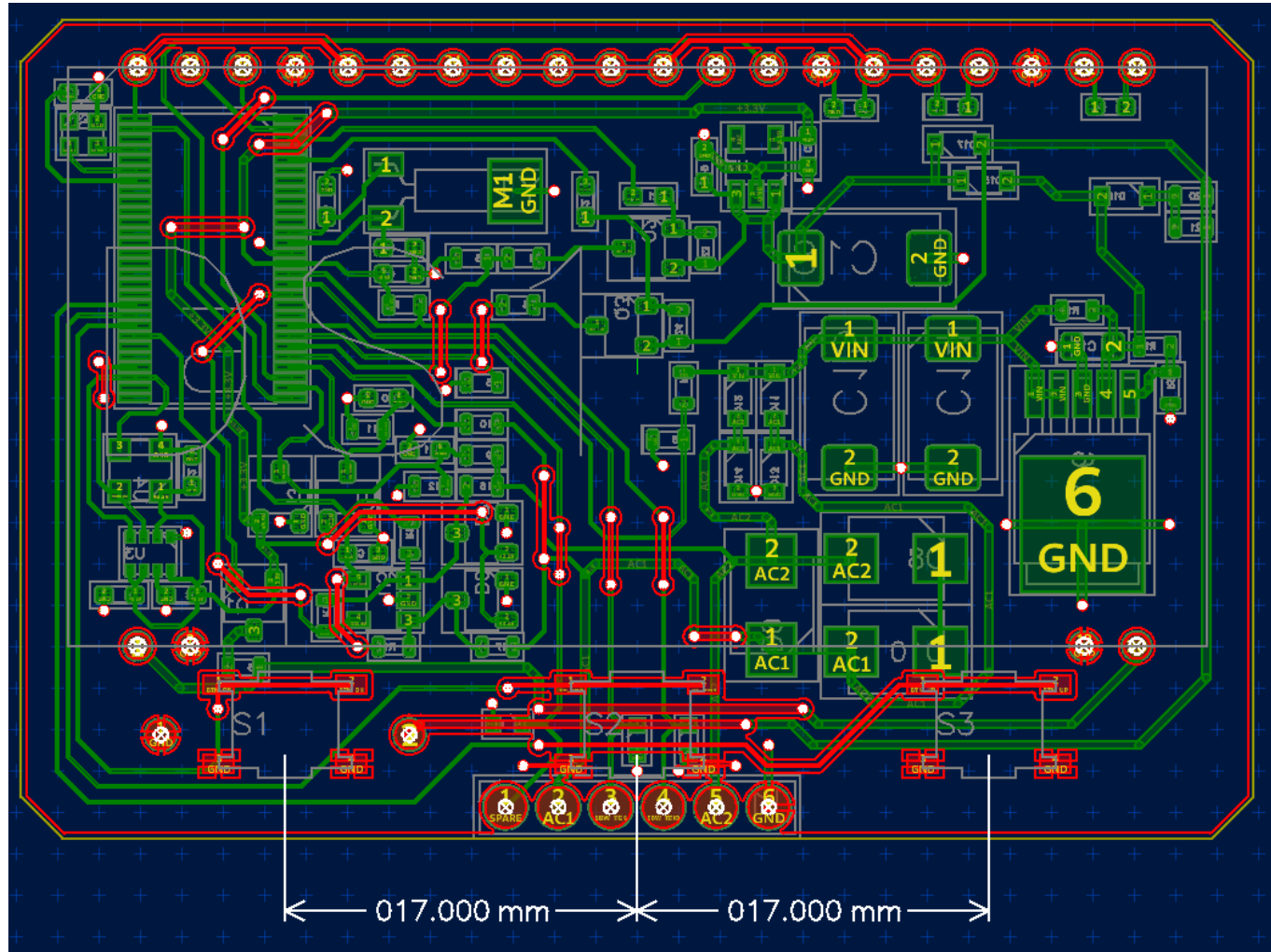
FOSDEM 2024

what I do

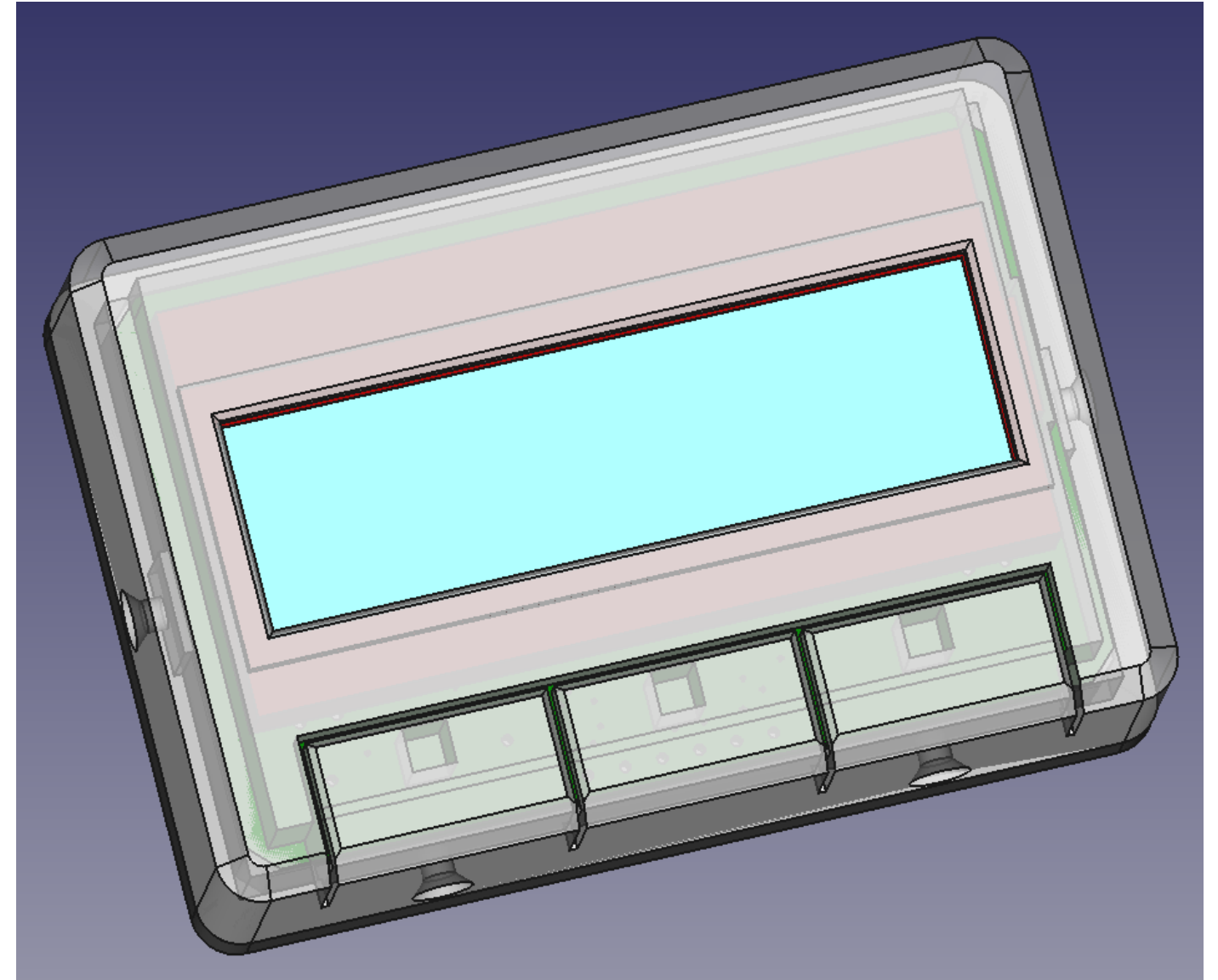
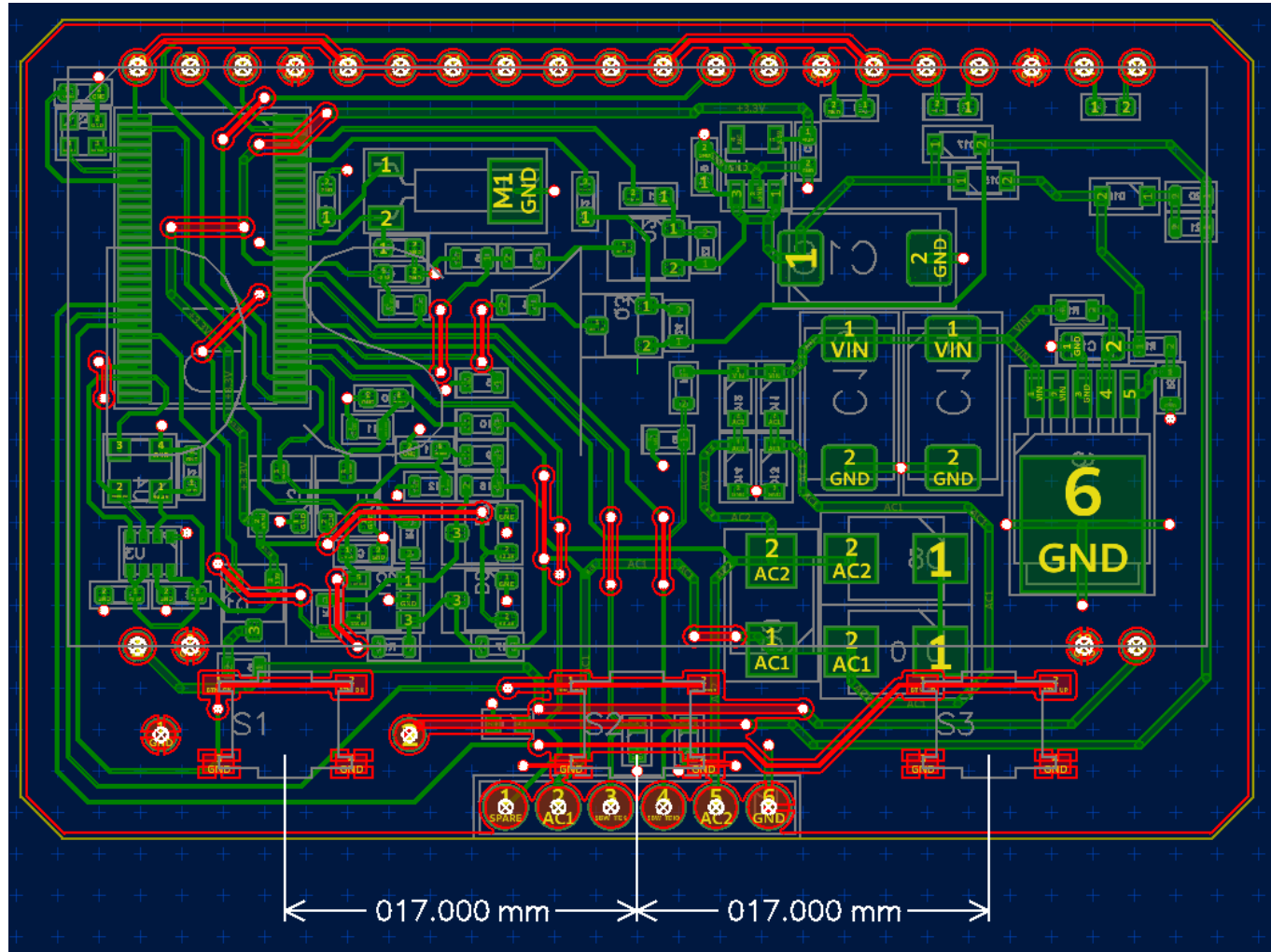













Horizon EDA



Horizon EDA

FreeCAD  
Solvespace



	FreeCAD
STEP import/export	
Chamfer/Flillet	
Easy referencing	
Non-modal sketcher	
Constraints in 3D	

	FreeCAD	Solvespace
STEP import/export	✓	✗
Chamfer/Flillet	✓	✗
Easy referencing	✗	✓
Non-modal sketcher	✗	✓
Constraints in 3D	✗	✓



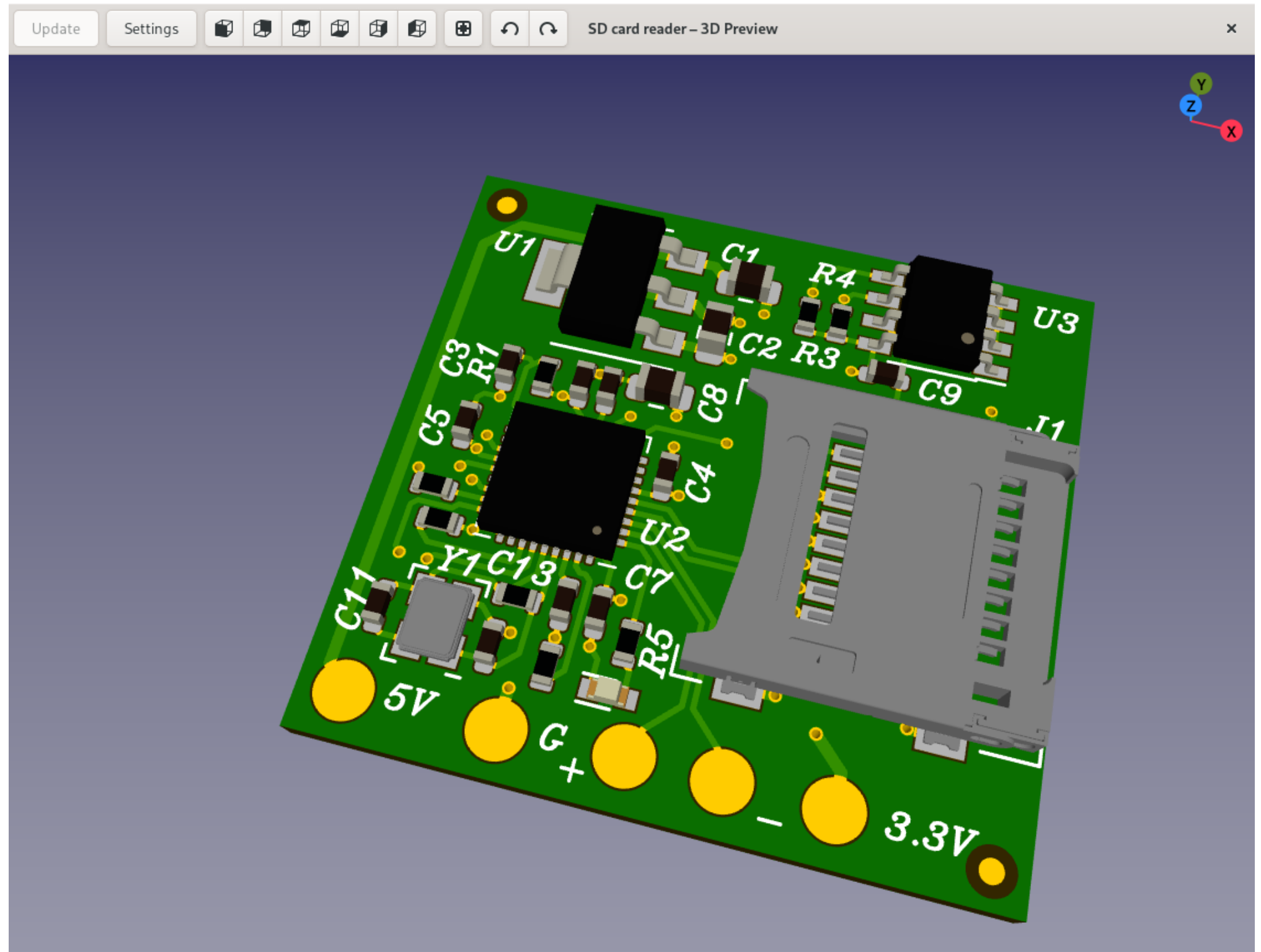
	FreeCAD	Solvespace	???
STEP import/export	✓	✗	✓
Chamfer/Fillet	✓	✗	✓
Easy referencing	✗	✓	✓
Non-modal sketcher	✗	✓	✓
Constraints in 3D	✗	✓	✓

What do we need to  
make a 3D CAD?

3D viewport

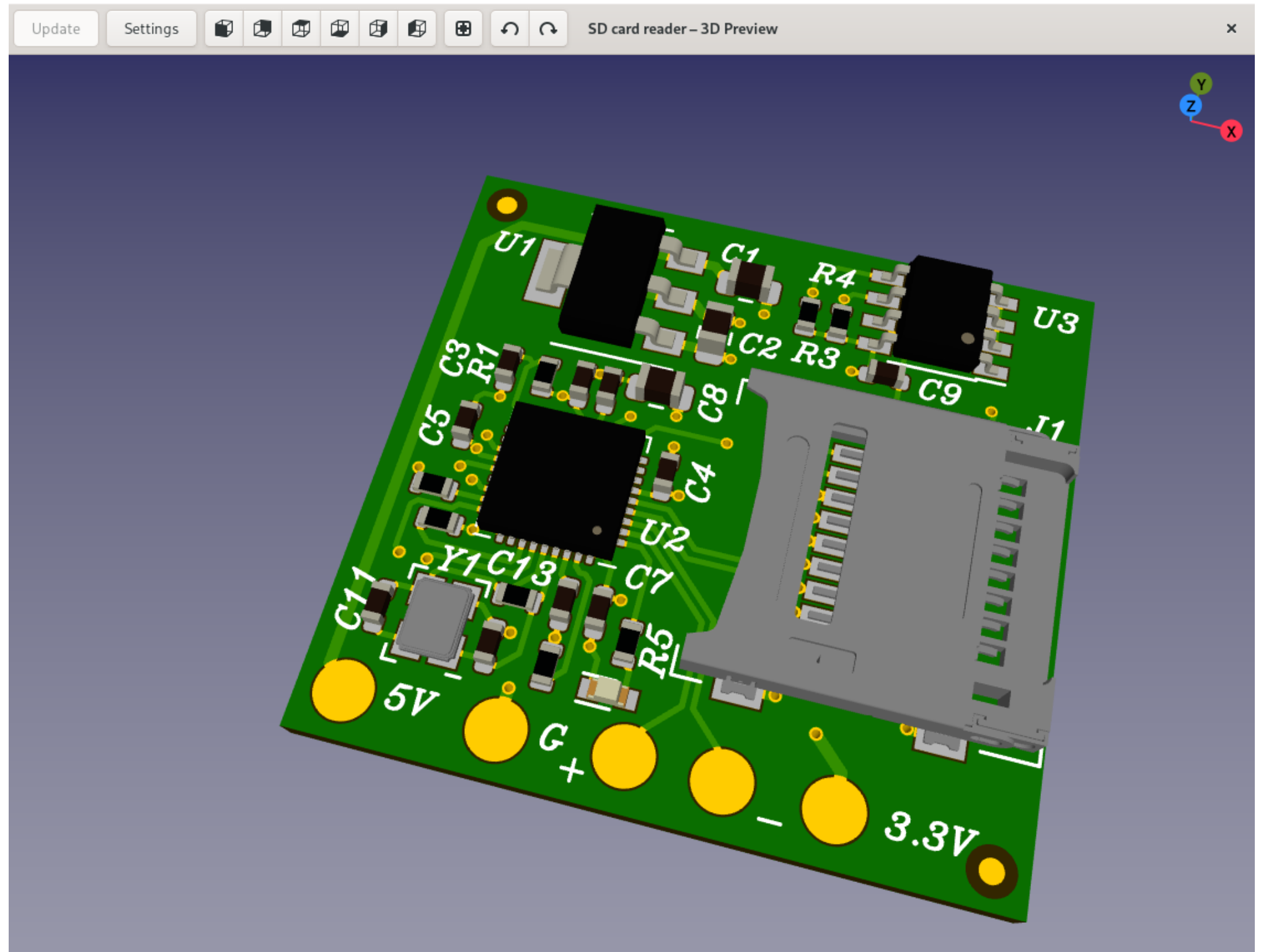
What do we need to  
make a 3D CAD?

3D viewport



What do we need to  
make a 3D CAD?

✓ 3D viewport



What do we need to  
make a 3D CAD?

✓ 3D viewport

Geometry kernel

What do we need to make a 3D CAD?

✓ 3D viewport

Geometry kernel

- Open CASCADE Technology
  - Introduction
  - Tutorials and Samples
  - Build, Debug and Upgrade
  - User Guides
    - Foundation Classes
    - Modeling Data
    - Modeling Algorithms
      - Introduction
      - Geometric Tools
      - Standard Topological Objects
      - Primitives
        - Making Primitives
        - Sweeping: Prism, Revolution and Pipe
    - Boolean Operations
    - Topological Tools
    - The Topology API
    - Planar Fillet
    - Hidden Line Removal
    - Making touching shapes connected
  - Mesh
  - Shape Healing
  - Visualization
  - VTK Integration Services (VIS)
  - IGES Translator
  - STEP Translator
  - Extended Data Exchange (XDE)
  - Data Exchange Wrapper (DE\_Wrapper)
  - OCAF
  - Draw Test Harness
  - Inspector
  - Specifications
  - Contribution
  - License

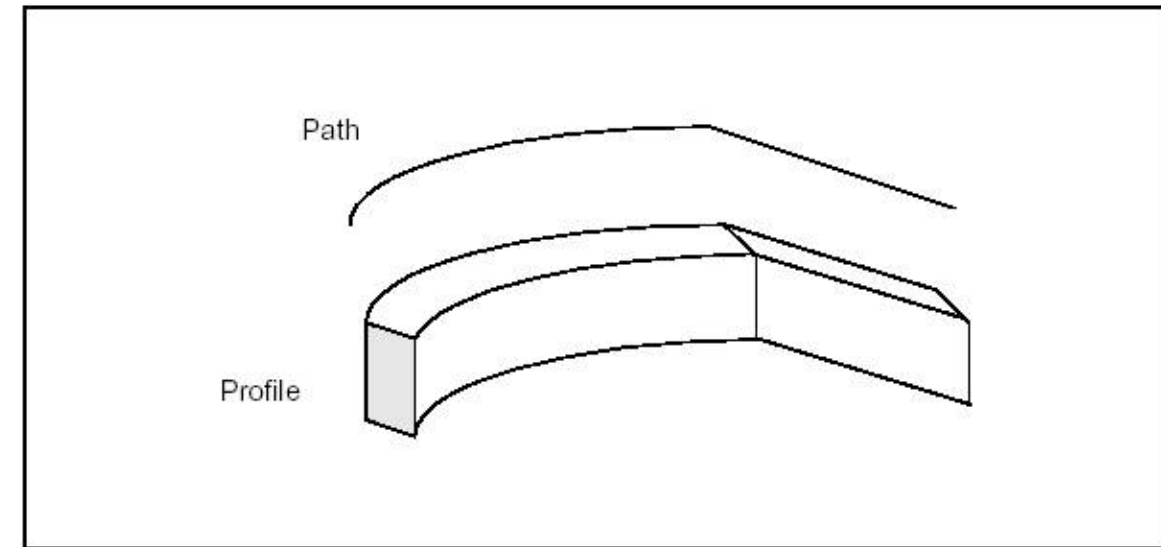
## Sweeping: Prism, Revolution and Pipe

### Sweeping

Sweeps are the objects you obtain by sweeping a **profile** along a **path**. The profile can be of any topology. The path is usually a curve or a wire. The profile generates objects according to the following rules:

- Vertices generate Edges
- Edges generate Faces.
- Wires generate Shells.
- Faces generate Solids.
- Shells generate Composite Solids

It is forbidden to sweep Solids and Composite Solids. A Compound generates a Compound with the sweep of all its elements.



Generating a sweep

*BRepPrimAPI\_MakeSweep* class is a deferred class used as a root of the following sweep classes:

- *BRepPrimAPI\_MakePrism* – produces a linear sweep
- *BRepPrimAPI\_MakeRevol* – produces a rotational sweep
- *BRepPrimAPI\_MakePipe* – produces a general sweep.

What do we need to make a 3D CAD?

✓ 3D viewport

✓ Geometry kernel

- Open CASCADE Technology
  - Introduction
  - Tutorials and Samples
  - Build, Debug and Upgrade
  - User Guides
    - Foundation Classes
    - Modeling Data
    - Modeling Algorithms
      - Introduction
      - Geometric Tools
      - Standard Topological Objects
      - Primitives
        - Making Primitives
        - Sweeping: Prism, Revolution and Pipe**
    - Boolean Operations
    - Topological Tools
    - The Topology API
    - Planar Fillet
    - Hidden Line Removal
    - Making touching shapes connected
  - Mesh
  - Shape Healing
  - Visualization
  - VTK Integration Services (VIS)
  - IGES Translator
  - STEP Translator
  - Extended Data Exchange (XDE)
  - Data Exchange Wrapper (DE\_Wrapper)
  - OCAF
  - Draw Test Harness
  - Inspector
  - Specifications
  - Contribution
  - License

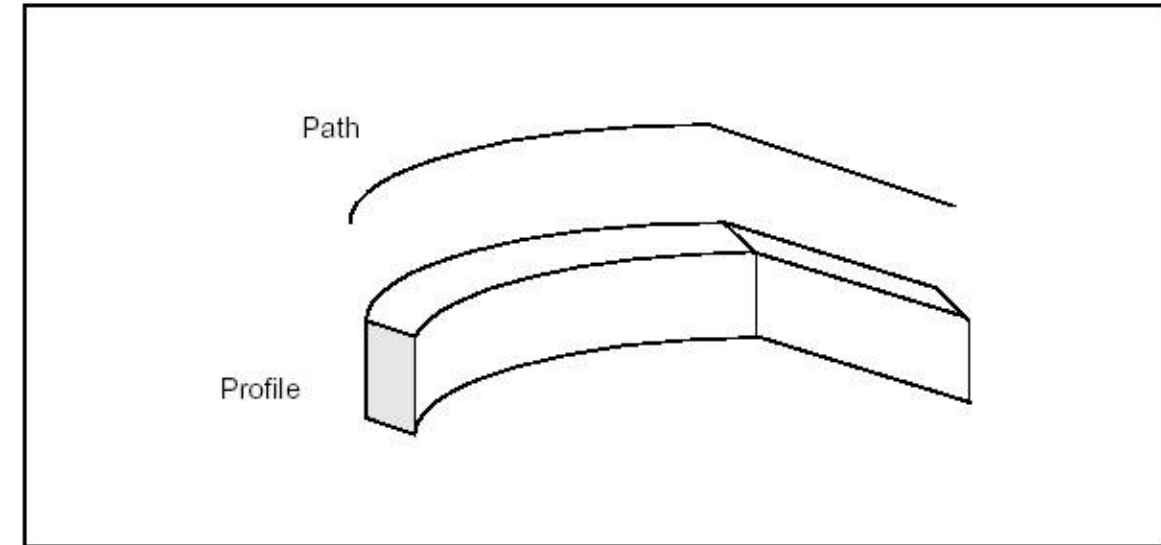
## Sweeping: Prism, Revolution and Pipe

### Sweeping

Sweeps are the objects you obtain by sweeping a **profile** along a **path**. The profile can be of any topology. The path is usually a curve or a wire. The profile generates objects according to the following rules:

- Vertices generate Edges
- Edges generate Faces.
- Wires generate Shells.
- Faces generate Solids.
- Shells generate Composite Solids

It is forbidden to sweep Solids and Composite Solids. A Compound generates a Compound with the sweep of all its elements.



Generating a sweep

*BRepPrimAPI\_MakeSweep* class is a deferred class used as a root of the following sweep classes:

- *BRepPrimAPI\_MakePrism* – produces a linear sweep
- *BRepPrimAPI\_MakeRevol* – produces a rotational sweep
- *BRepPrimAPI\_MakePipe* – produces a general sweep.

What do we need to  
make a 3D CAD?

✓ 3D viewport

✓ Geometry kernel

Solver



# What do we need to make a 3D CAD?

✓ 3D viewport

✓ Geometry kernel

Solver

## SOLVESPACE -- parametric 2d/3d CAD

[Examples](#)

[Tutorials](#)

[Features](#)

[Download](#)

[Reference](#)

[Technology](#)

[Library](#)

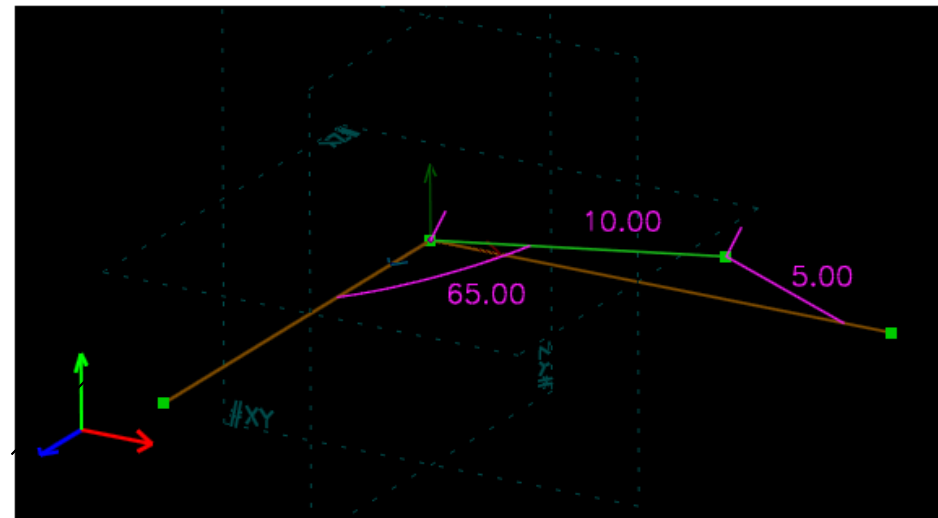
[Forum](#)

[Contact](#)

### TECHNOLOGY: SOLVING CONSTRAINTS

The core of any parametric CAD program is its geometric constraint solver. The solver starts with geometric properties of the sketch, like a line length or an angle or a tangency. Based on these properties, it calculates a simple representation of the sketch's lines, points, and curves.

For a trivial example, we might specify that a point lies 10 mm from the origin, and 5 mm from the x-axis. We also specify that the line from the origin to that point makes a 65° angle with the z-axis. That point turns out to have  $(x, y, z) = (8.66, 2.67, 4.23)$ , after solving three equations in three unknowns. A real sketch may involve hundreds of unknowns.



In SolveSpace, constraints are represented as equations in a symbolic algebra system. In general, these equations are solved numerically, by a modified Newton's method. Some special cases are handled, for any equation that can be solved by forward substitution.

The constraint solver used in SolveSpace is available separately as a [library](#).

# What do we need to make a 3D CAD?

✓ 3D viewport

✓ Geometry kernel

✓ Solver

## SOLVESPACE -- parametric 2d/3d CAD

[Examples](#)

[Tutorials](#)

[Features](#)

[Download](#)

[Reference](#)

[Technology](#)

[Library](#)

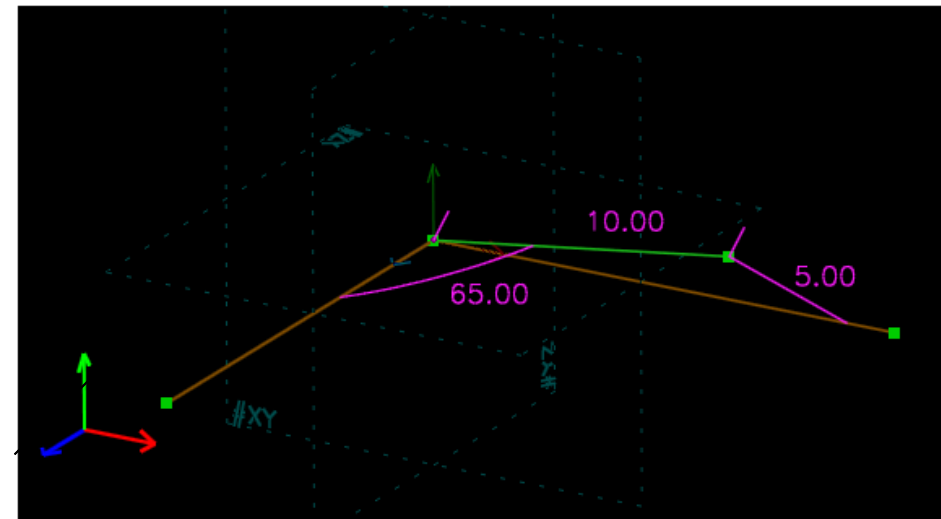
[Forum](#)

[Contact](#)

### TECHNOLOGY: SOLVING CONSTRAINTS

The core of any parametric CAD program is its geometric constraint solver. The solver starts with geometric properties of the sketch, like a line length or an angle or a tangency. Based on these properties, it calculates a simple representation of the sketch's lines, points, and curves.

For a trivial example, we might specify that a point lies 10 mm from the origin, and 5 mm from the x-axis. We also specify that the line from the origin to that point makes a 65° angle with the z-axis. That point turns out to have  $(x, y, z) = (8.66, 2.67, 4.23)$ , after solving three equations in three unknowns. A real sketch may involve hundreds of unknowns.



In SolveSpace, constraints are represented as equations in a symbolic algebra system. In general, these equations are solved numerically, by a modified Newton's method. Some special cases are handled, for any equation that can be solved by forward substitution.

The constraint solver used in SolveSpace is available separately as a [library](#).

What do we need to  
make a 3D CAD?

✓ 3D viewport

✓ Geometry kernel

✓ Solver

User interface

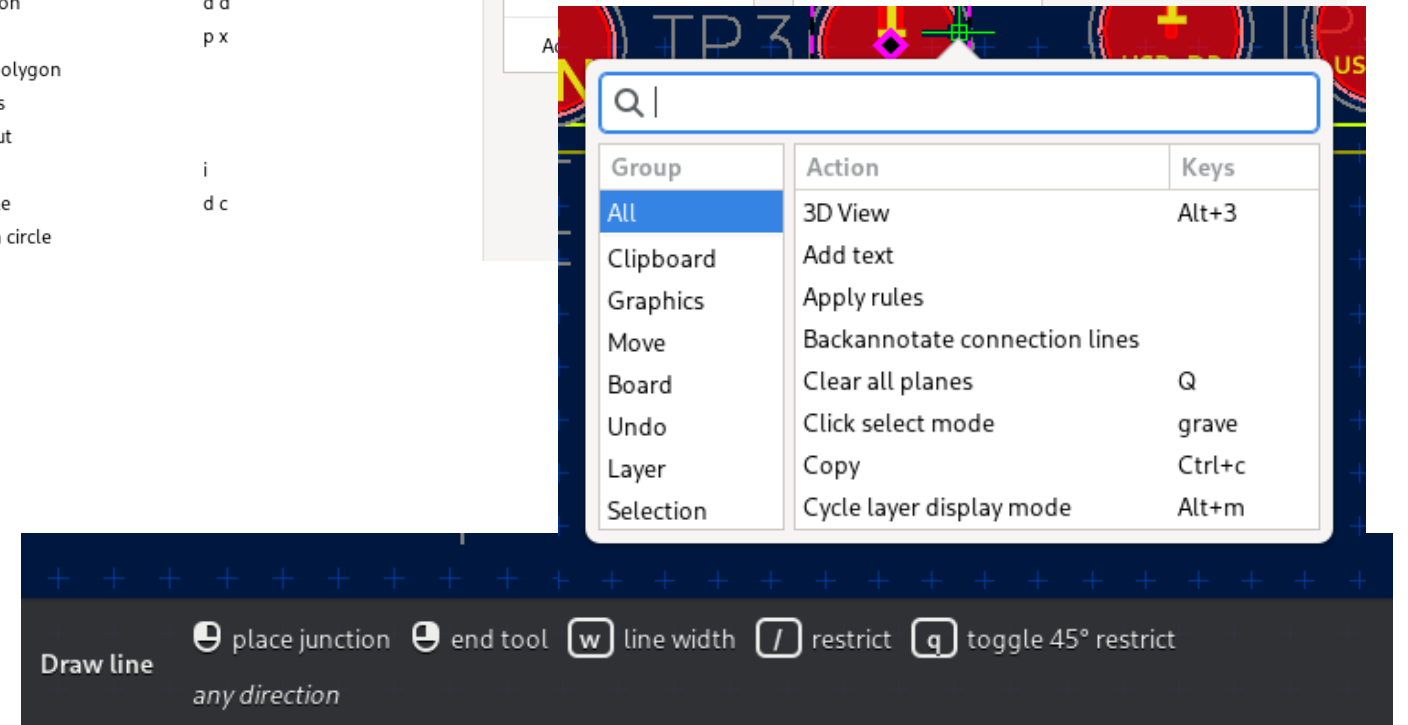
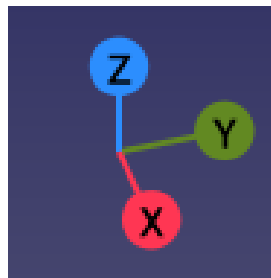
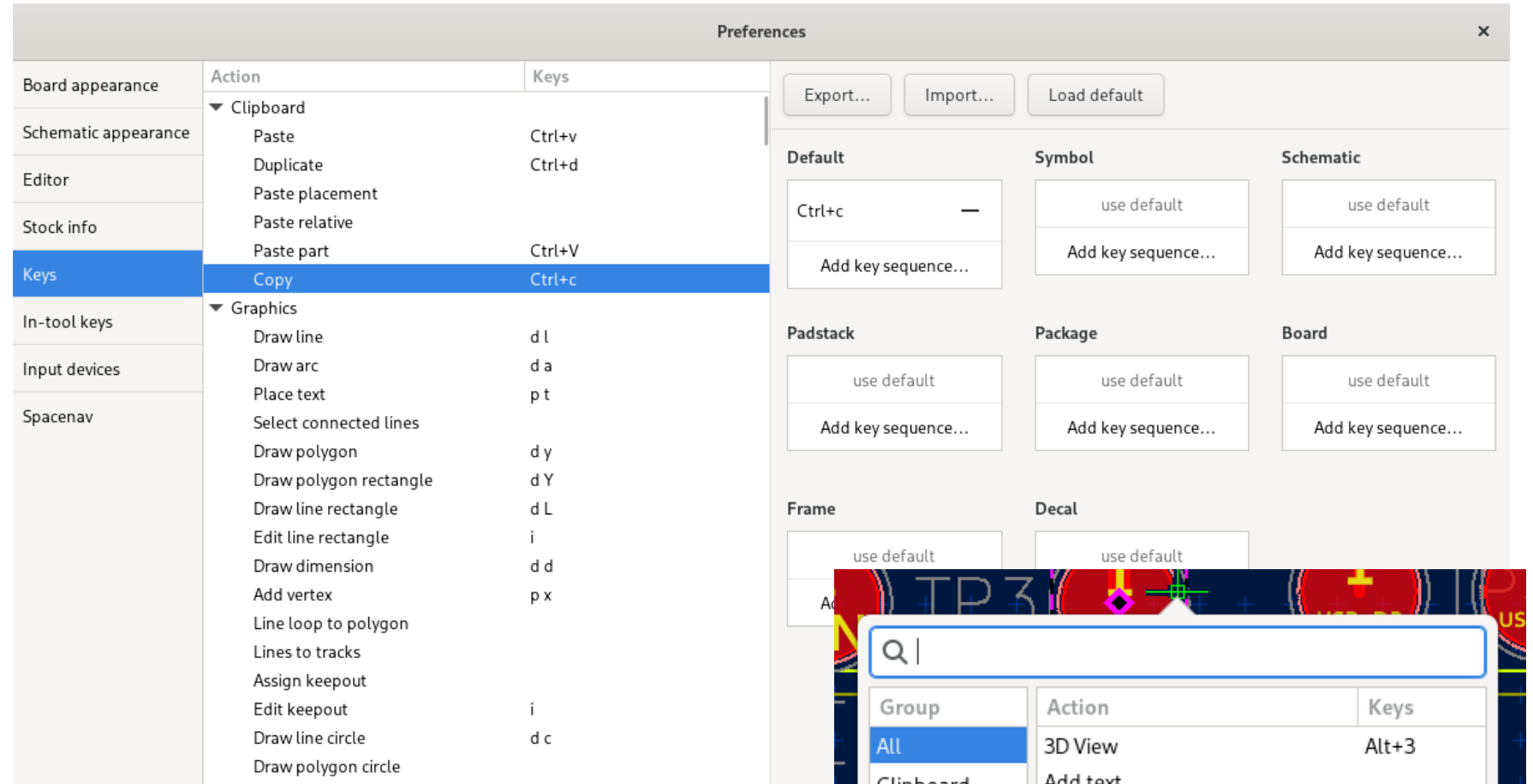
# What do we need to make a 3D CAD?

✓ 3D viewport

✓ Geometry kernel

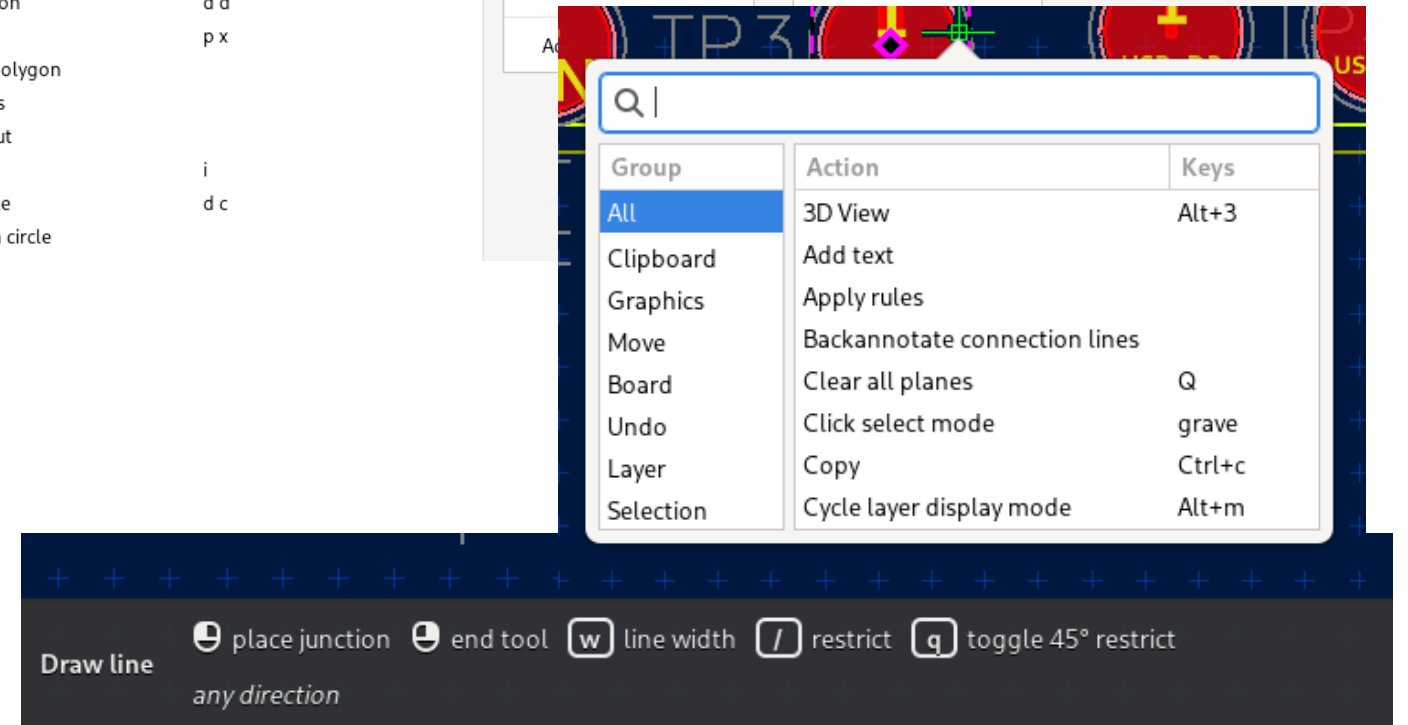
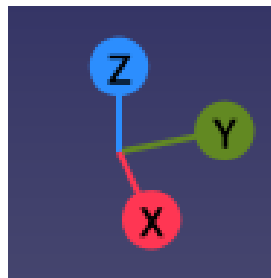
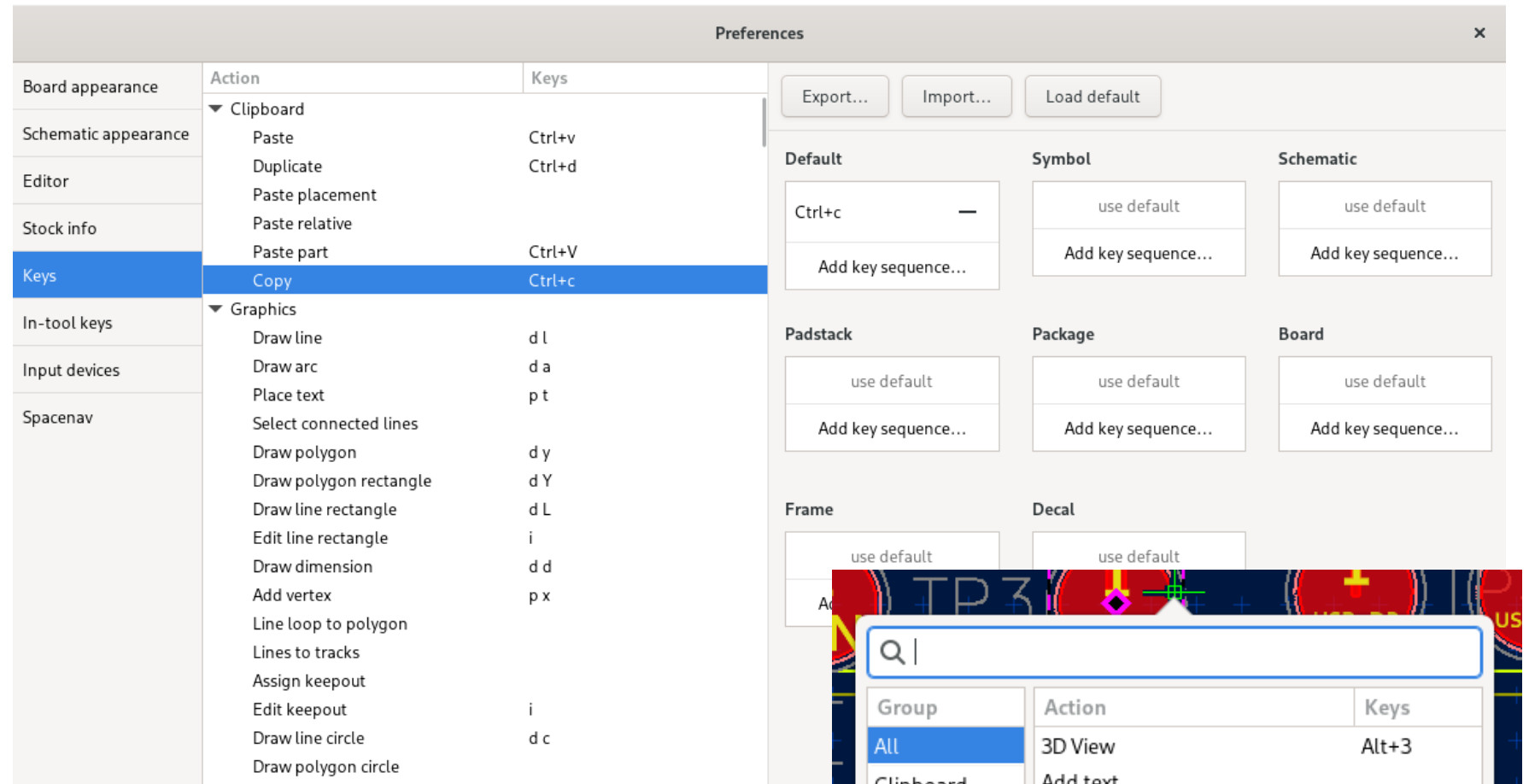
✓ Solver

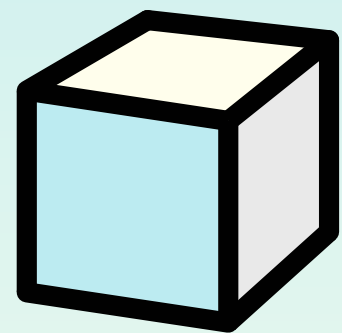
User interface



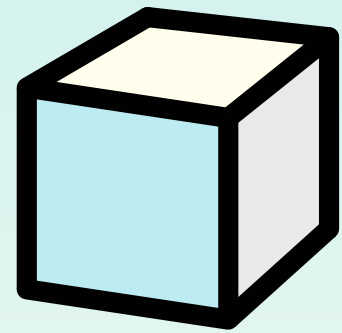
# What do we need to make a 3D CAD?

- ✓ 3D viewport
- ✓ Geometry kernel
- ✓ Solver
- ✓ User interface





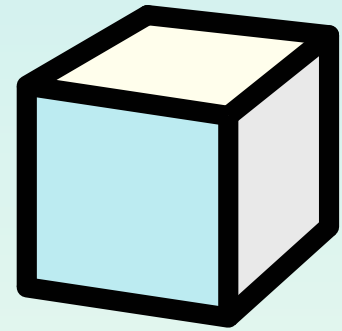
**Dune 3D**  
parametric 3D CAD



**Dune 3D**  
parametric 3D CAD

6 Months

C++20



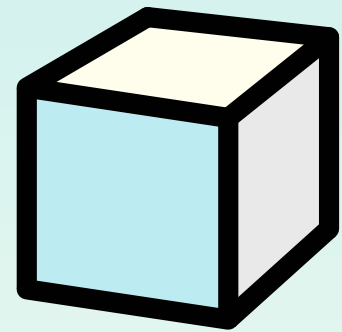
**Dune 3D**  
parametric 3D CAD

6 Months



C++20

33k LOC



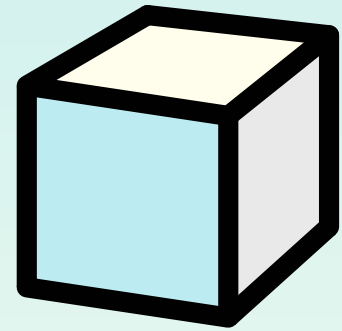
**Dune 3D**  
parametric 3D CAD

6 Months

gtkmm 4

C++20

33k LOC



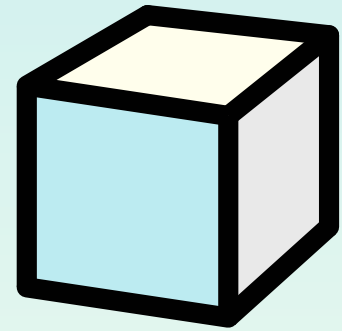
**Dune 3D**  
parametric 3D CAD

6 Months

gtkmm 4

C++20

33k LOC



**Dune 3D**  
parametric 3D CAD

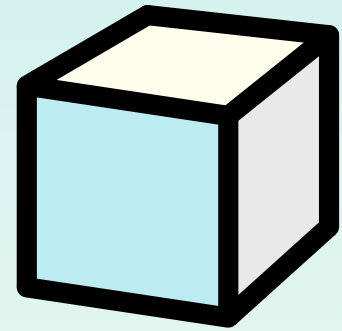
6 Months

0d233a6e-9ea3-4ba1-90e2-2a3f96f3f63e

gtkmm 4

C++20

33k LOC



**Dune 3D**  
parametric 3D CAD

JSON

6 Months

0d233a6e-9ea3-4ba1-90e2-2a3f96f3f63e

# Dune 3D Version 1.0.0 "Ångström"

Latest

Compare



 carrotIndustries released this 3 days ago · [1 commit](#) to main since this release  v1.0.0  ae9fdb4

## This is the first versioned release of Dune 3D

See [the docs](#) for installation instructions.

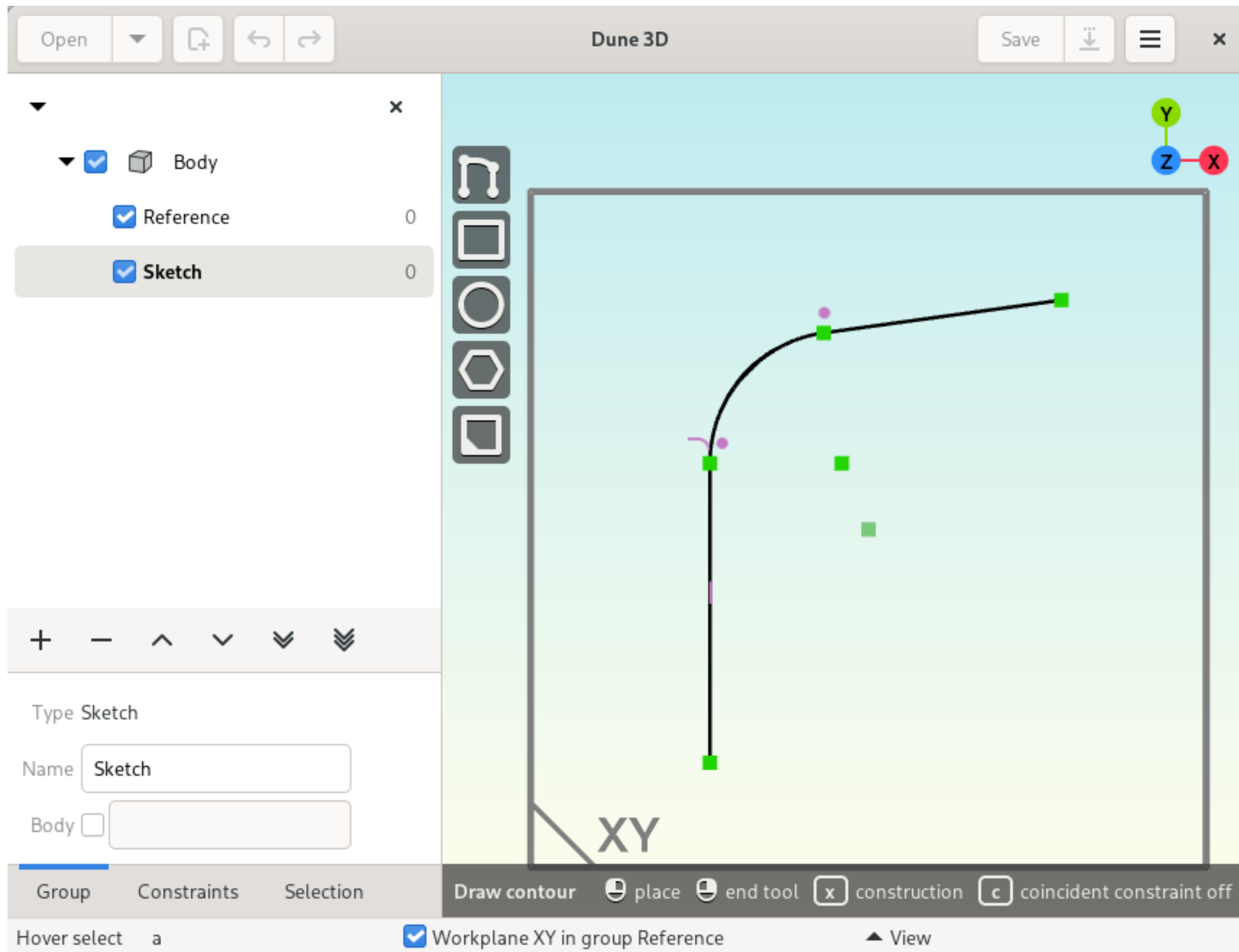
## Changelog

since this is the first versioned release, there is no changelog yet

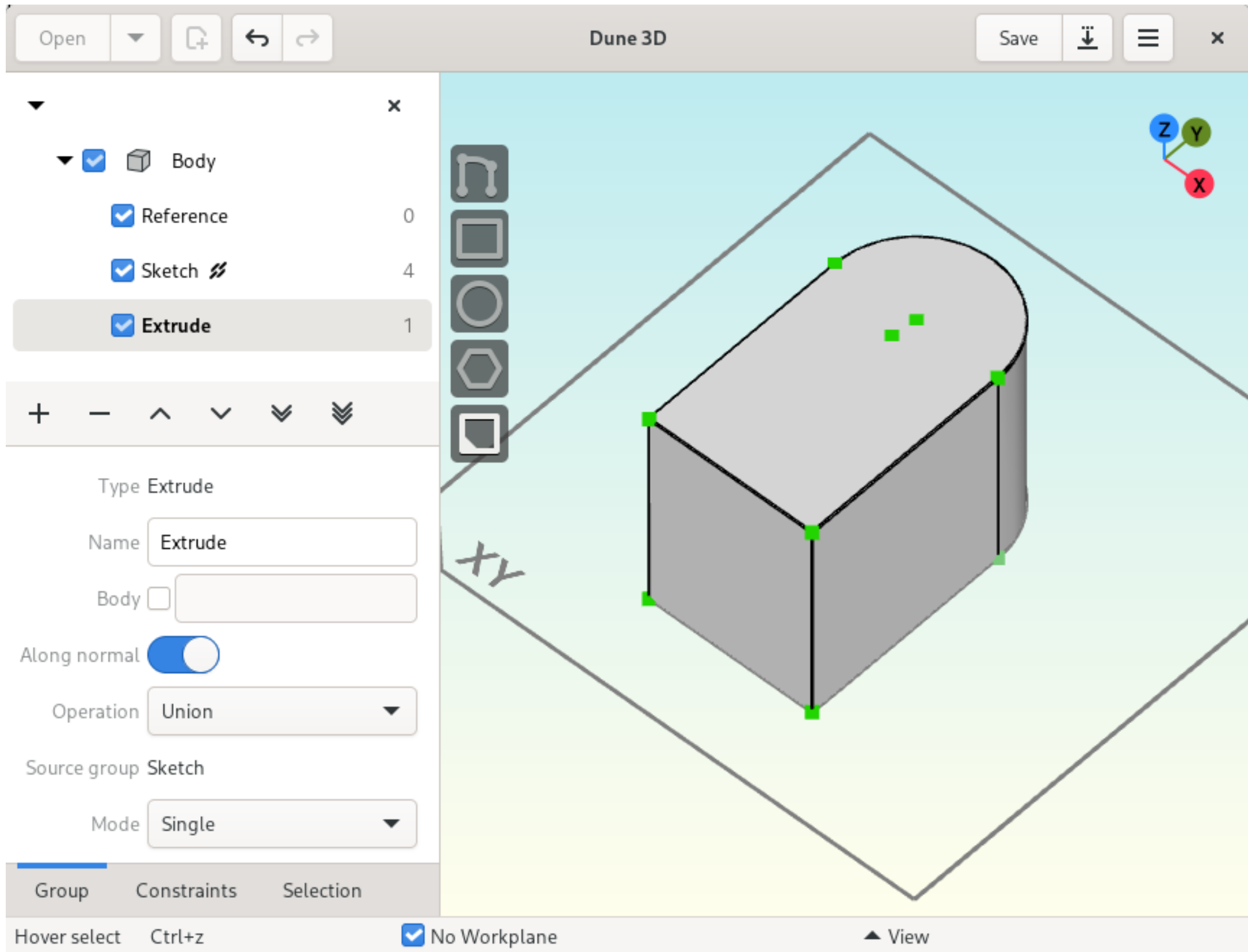
### ▼ Assets 3

 <a href="#">dune3d-1.0.0-x64.msi</a>	59.9 MB	3 days ago
 <a href="#">Source code (zip)</a>		3 days ago
 <a href="#">Source code (tar.gz)</a>		3 days ago

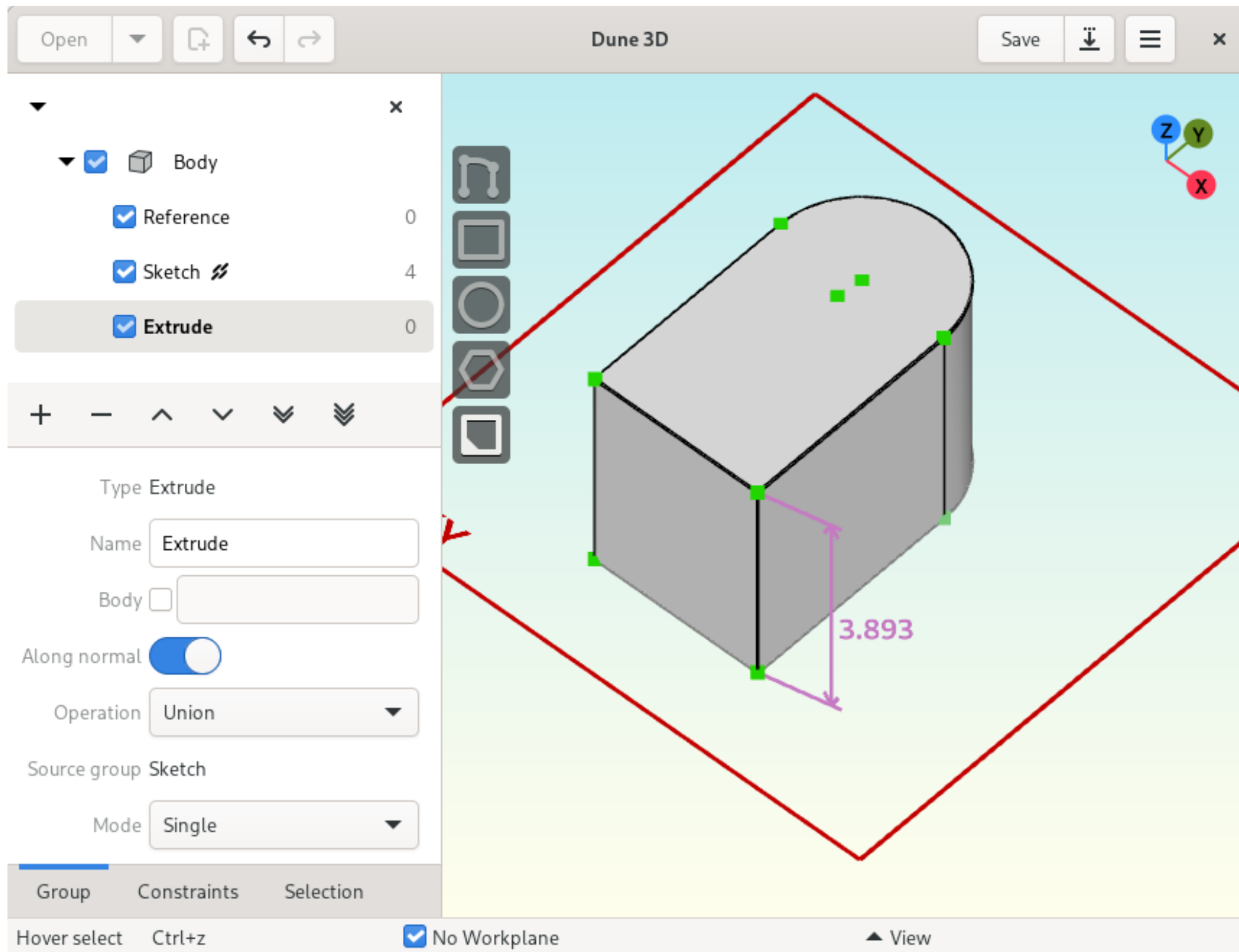
  2  7  2  2 10 people reacted



- Sketch
- Lines
- Arcs
- Circles
- Constraints
- All-in-one tool
- Regular polygon
- Rectangle

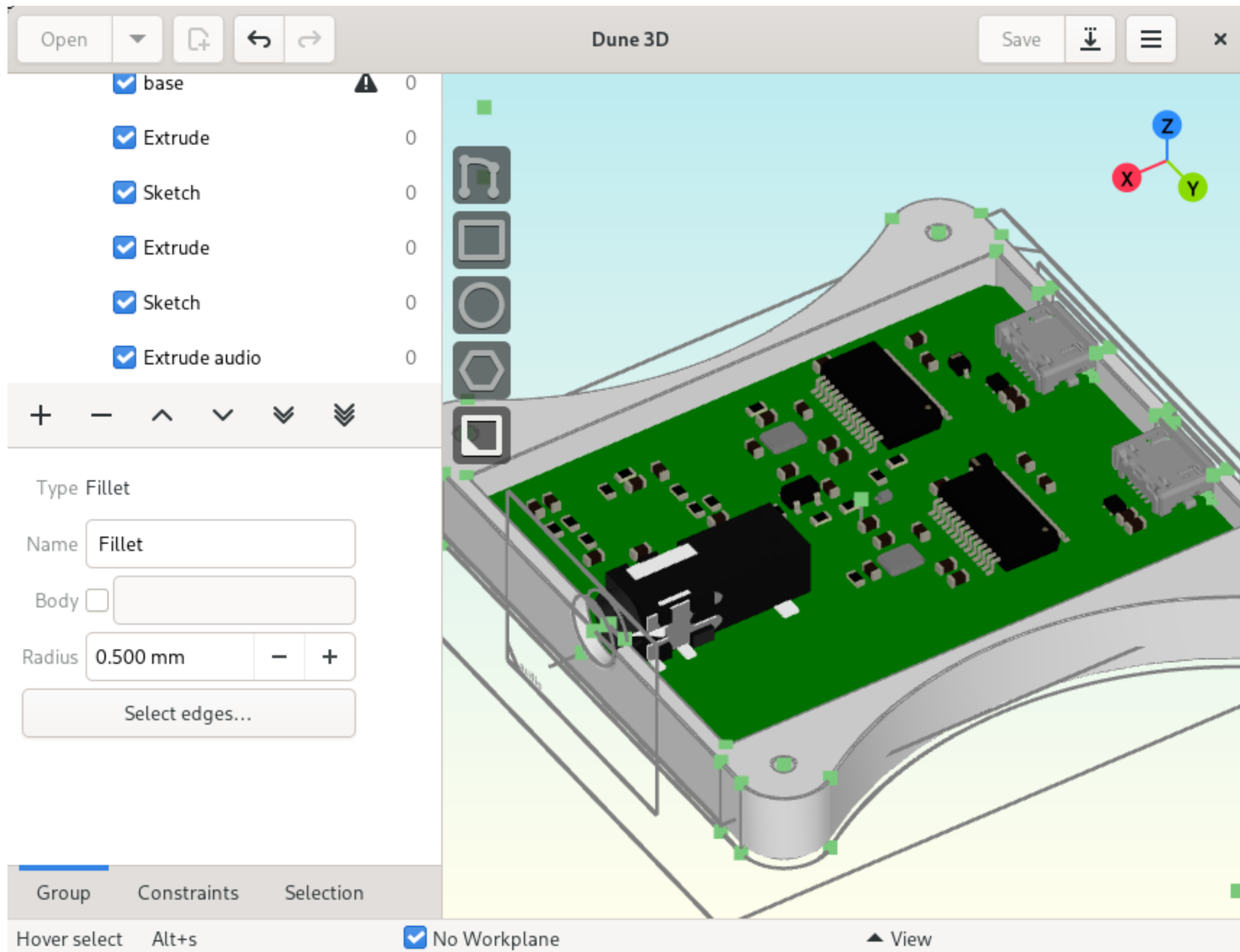


- Extrude
- Lathe
- Linear array
- Polar array

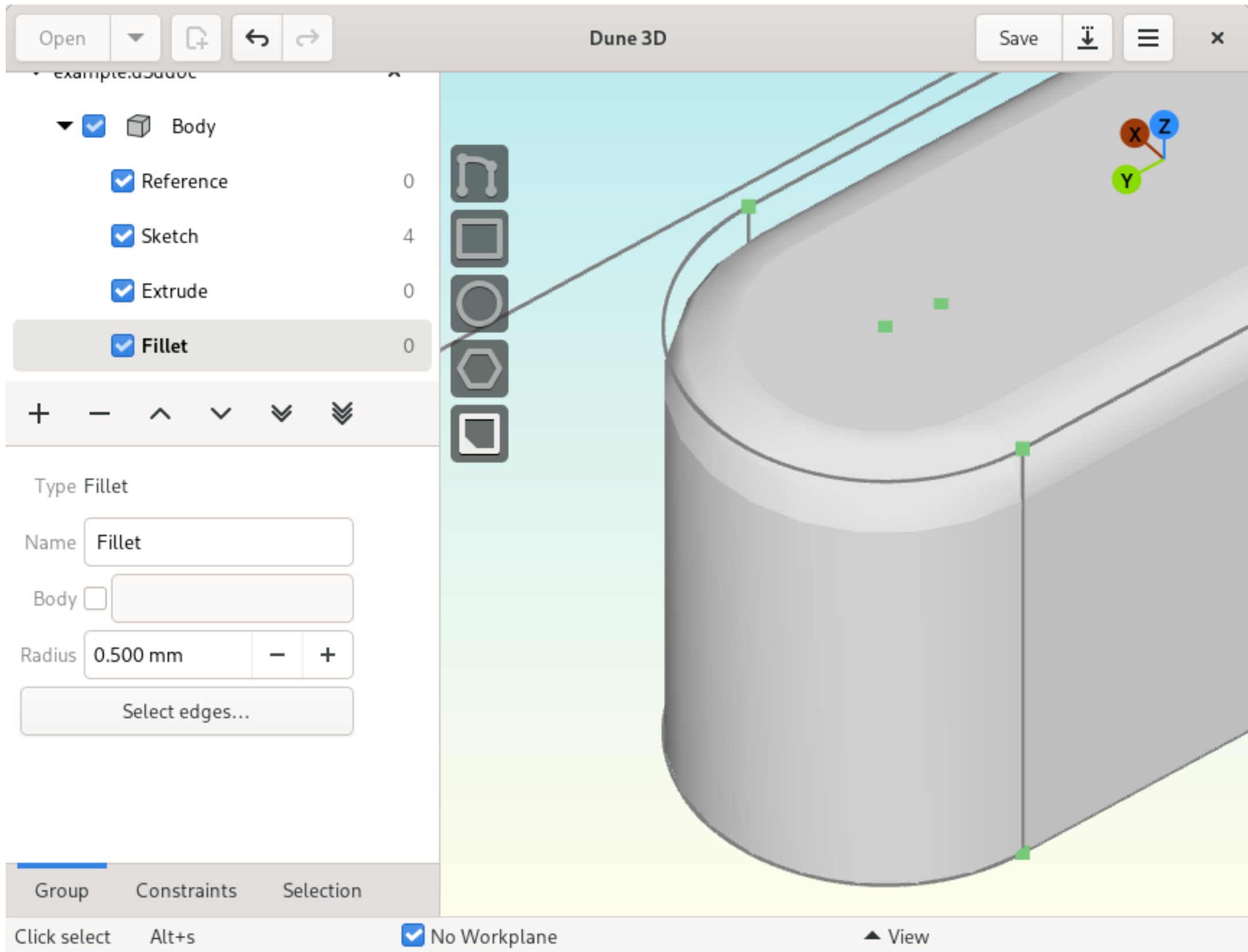


- 3D Constraints
  - Distance
  - Angle
  - Point in plane
  - Point/plane distance



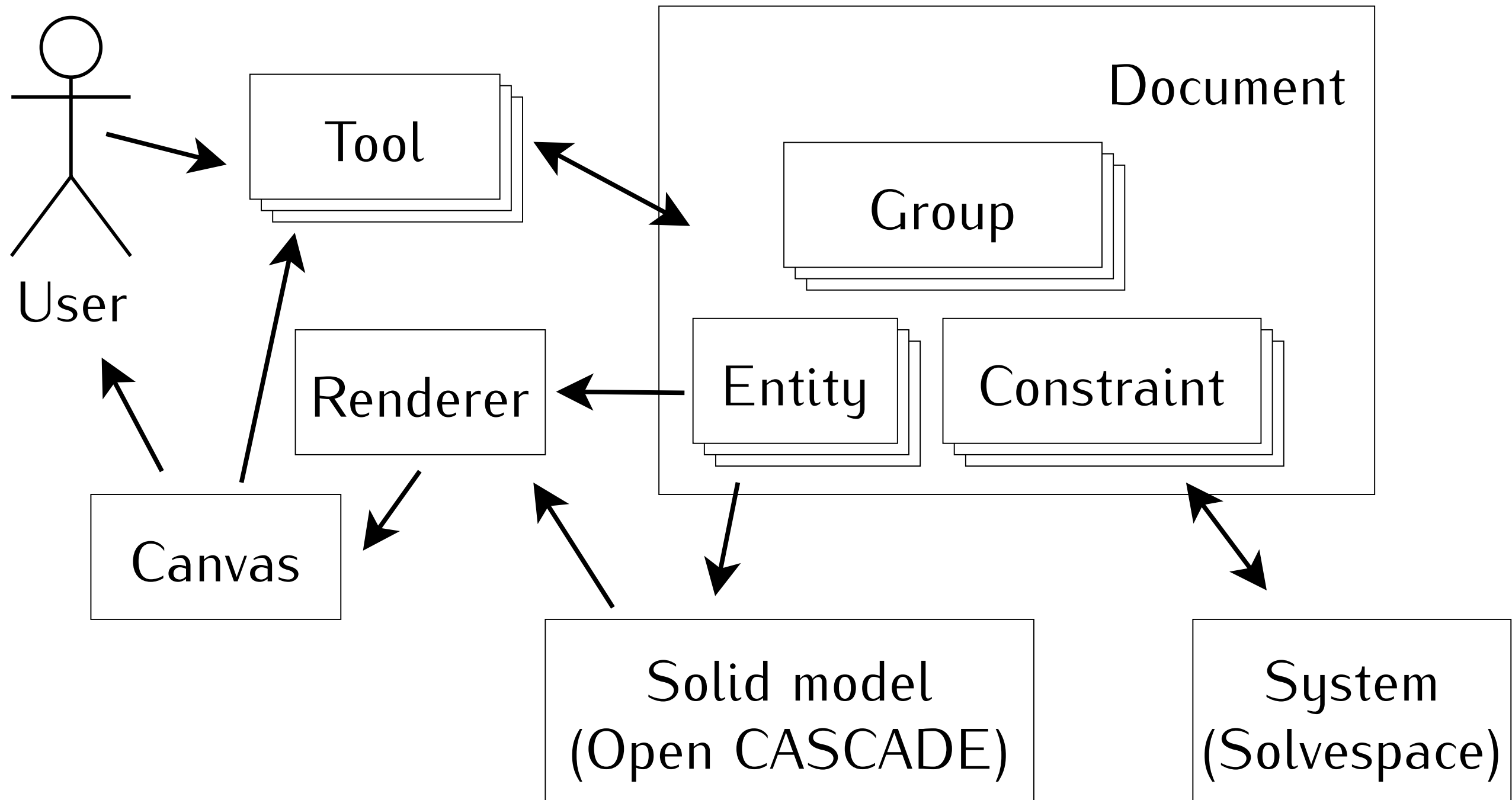


- STEP import
- Extract reference points from model

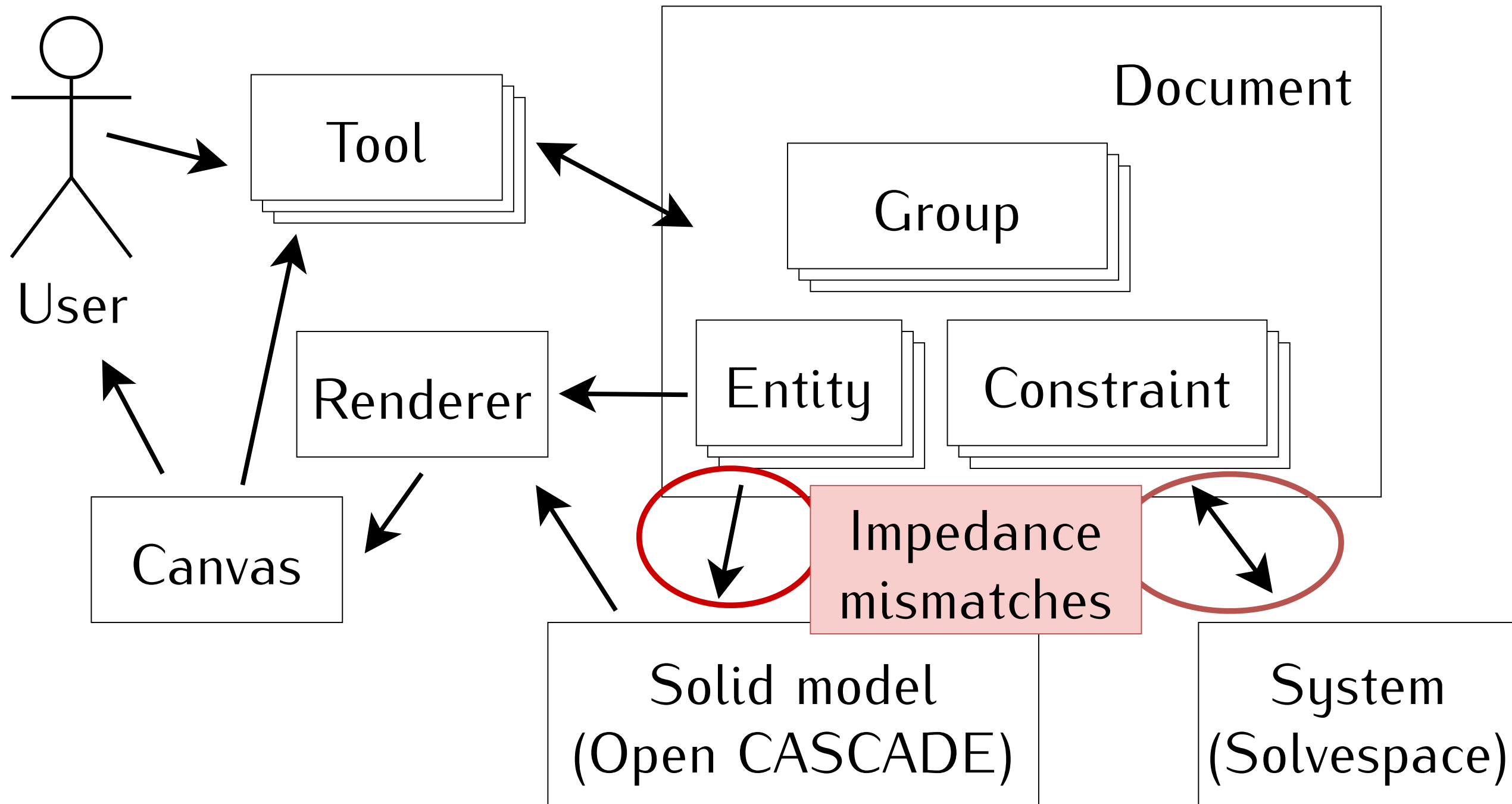


- Chamfers
- Fillets
- Topological naming problem 😞

So how does it work?



So how does it work?



# What's next?

- Measurements
- Revolution
- Copy/paste
- Better solver integration
- Various UI enhancements

*That's it*

**dune3d.org**