

**A 2D world
with a time-axis**

**Anfängerpraktikum
Von Wouter Zeelen**

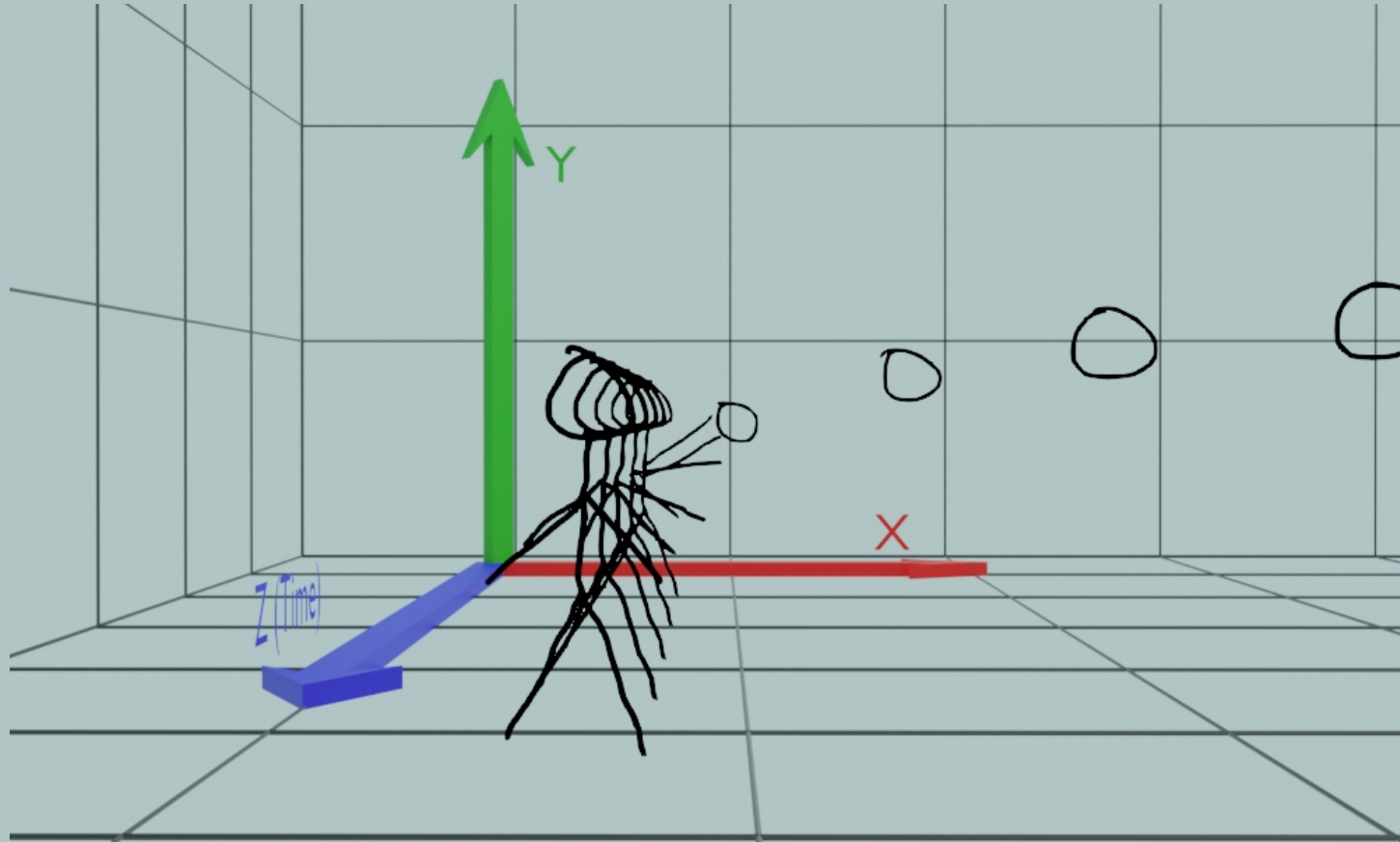
Time on the 3rd dimension



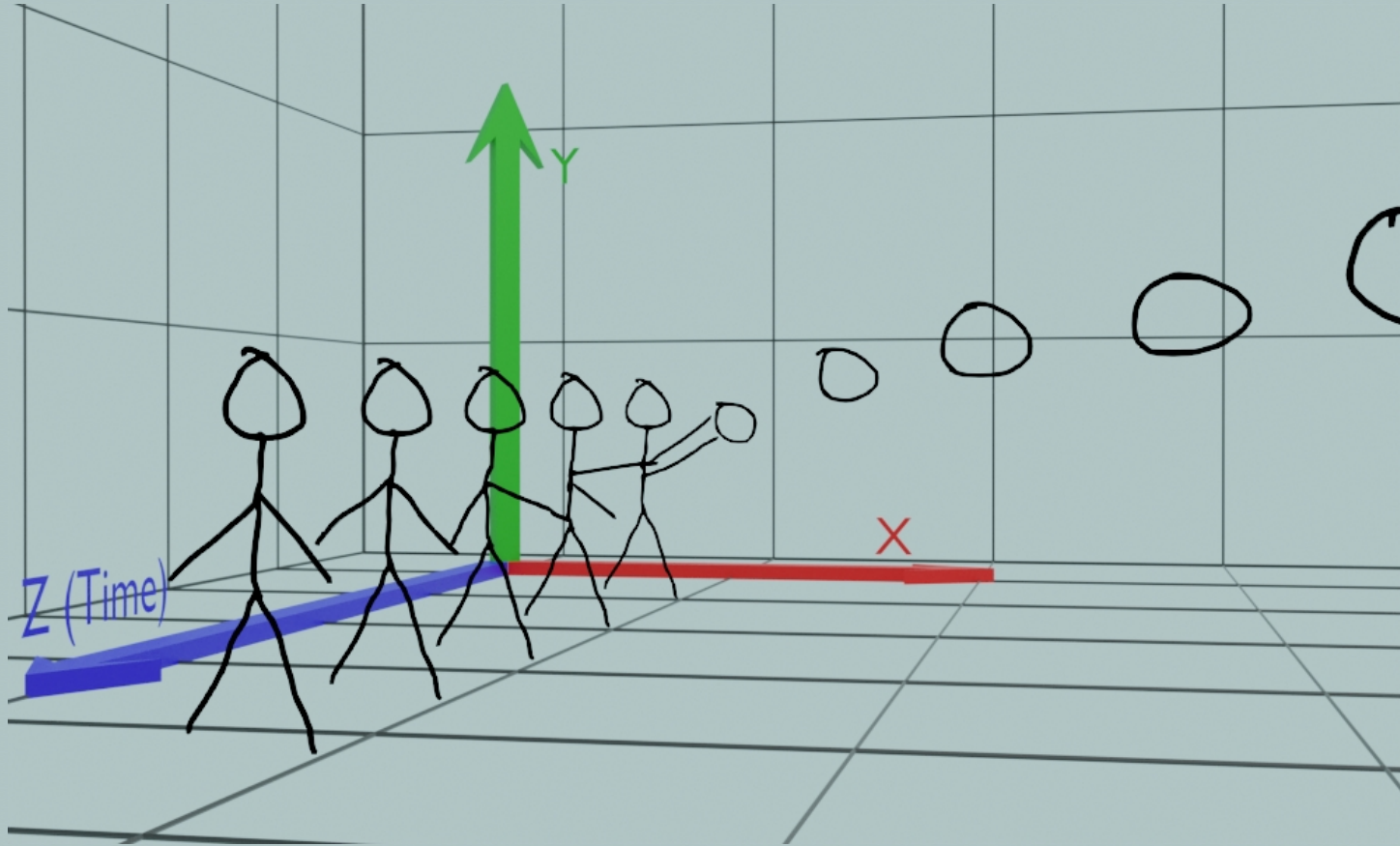
Time on the 3rd dimension



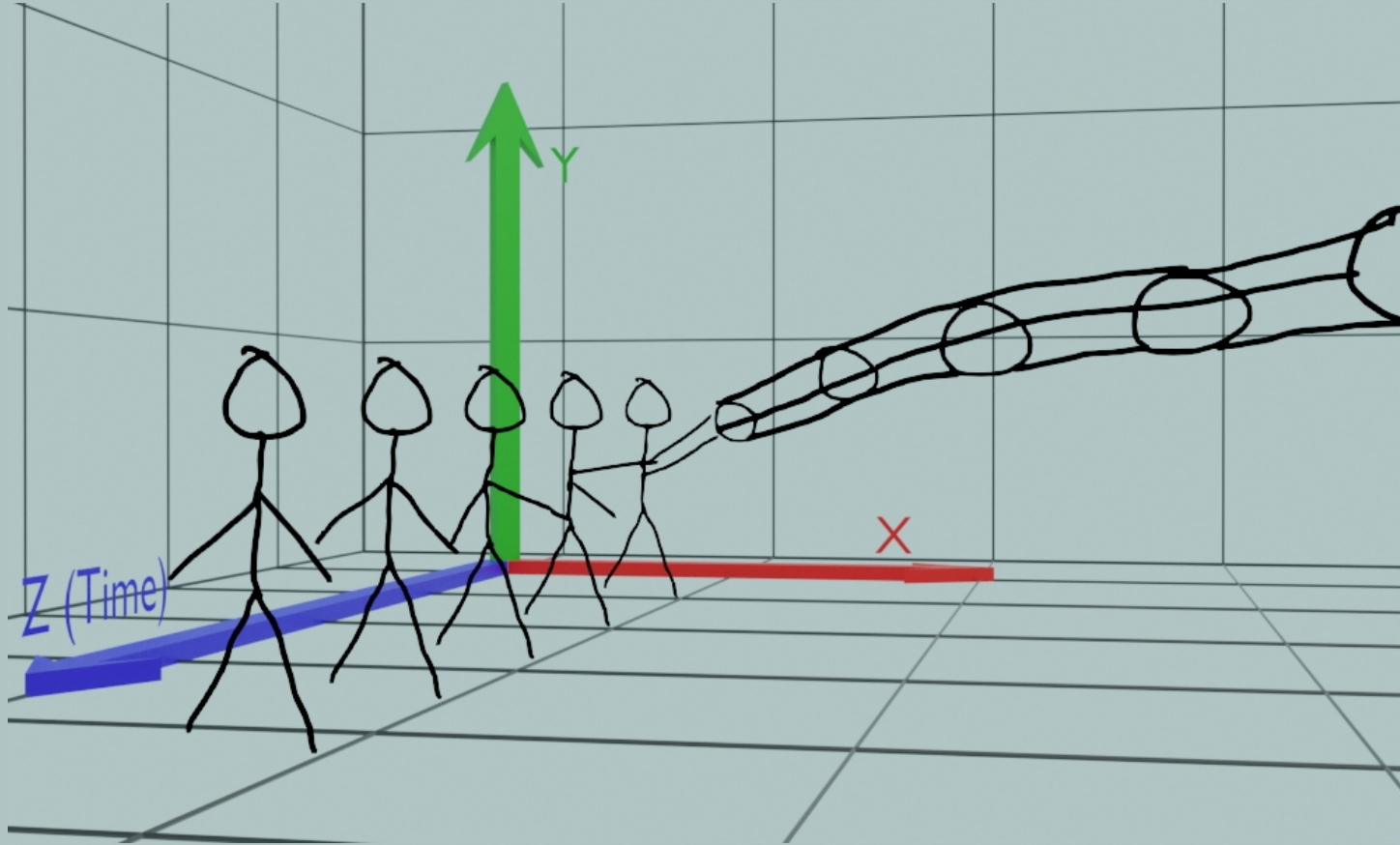
Time on the 3rd dimension



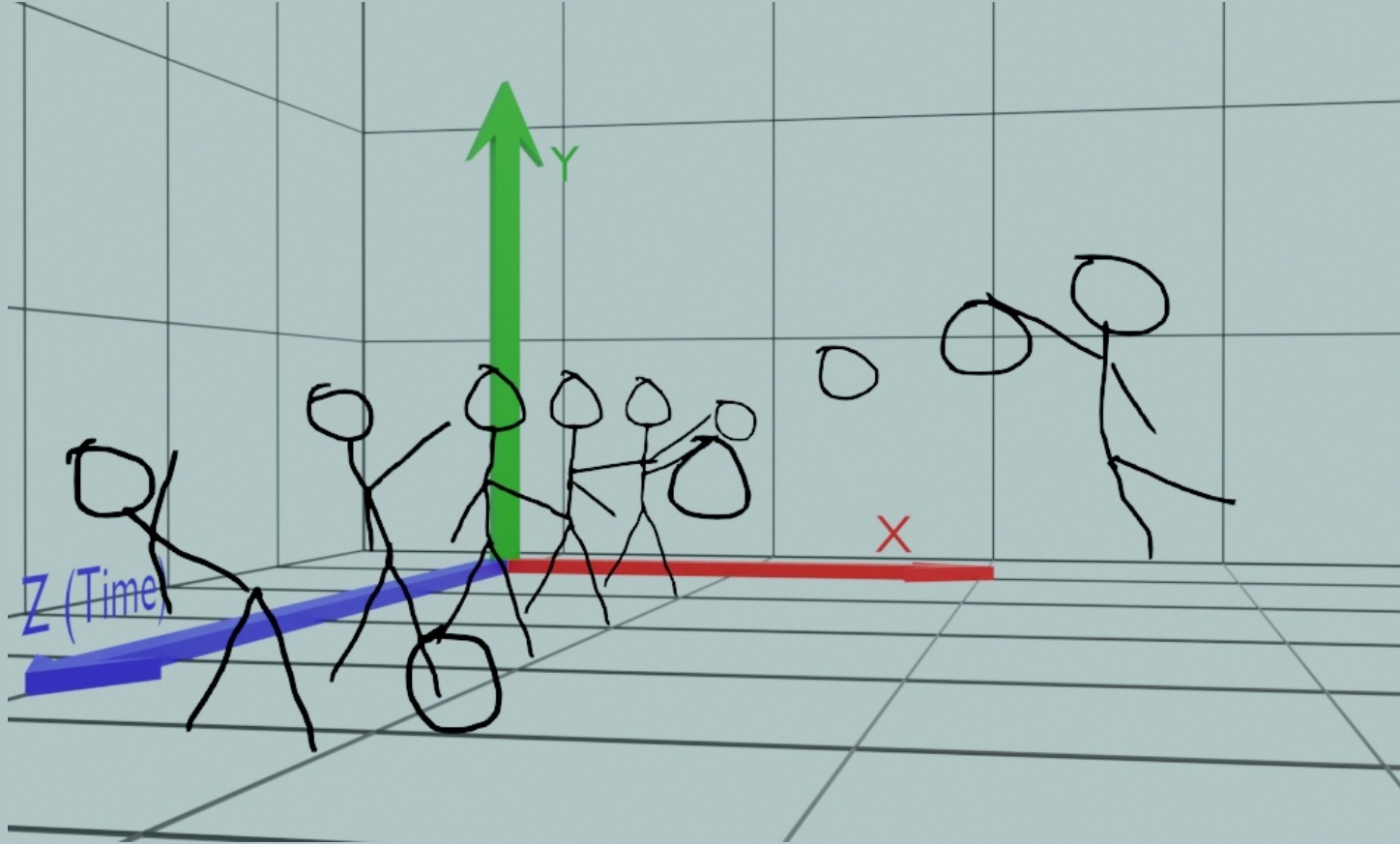
Time on the 3rd dimension



Time on the 3rd dimension

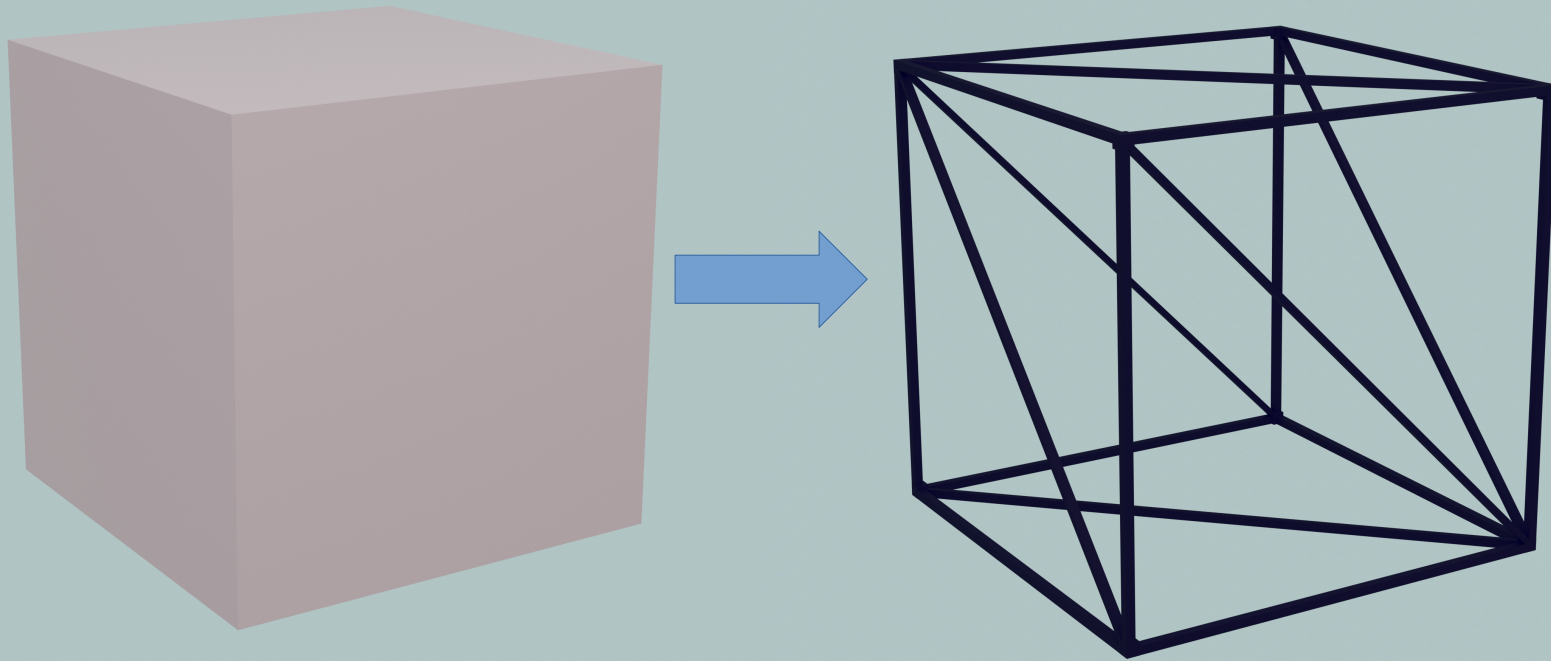


Time on the 3rd dimension

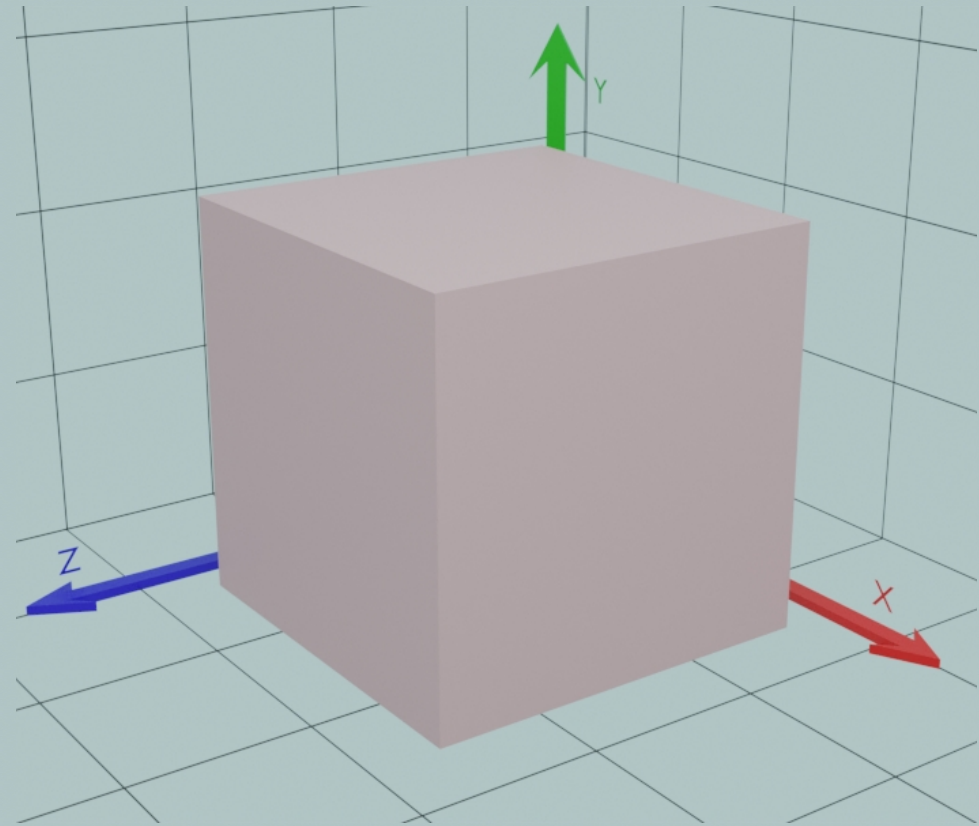


Demonstration

Mesh vs Collision Shape



How 3D objects work



How 3D objects work

Vertecies

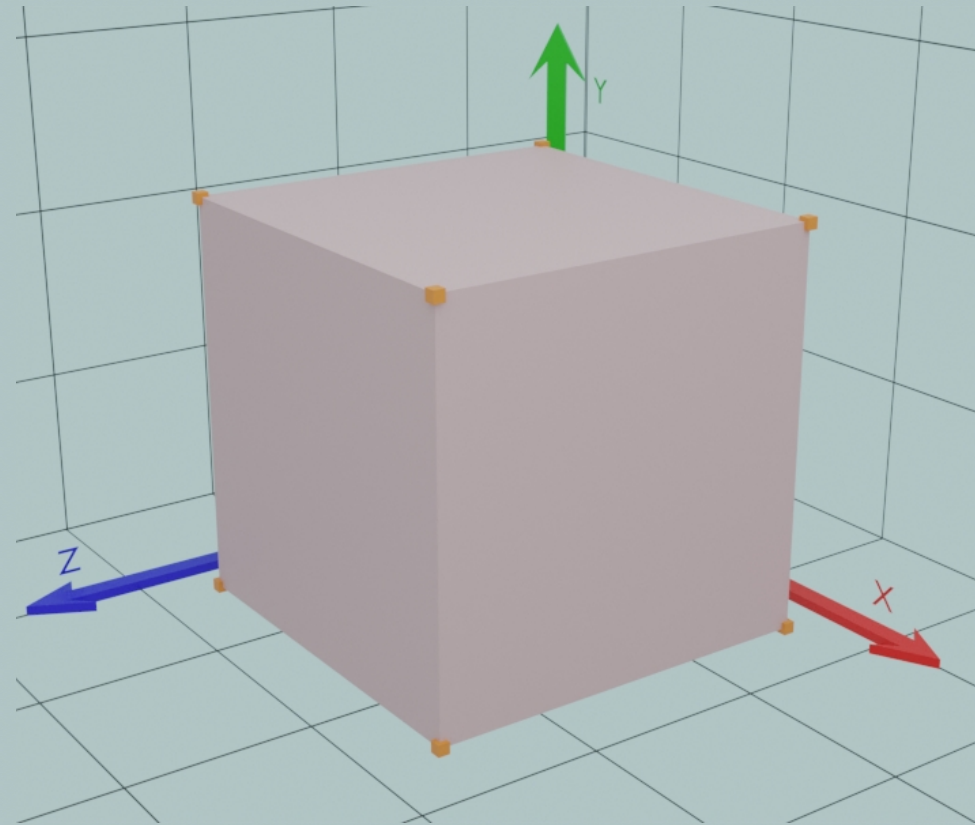
	X	Y	Z
Vertex0	1	1	1
Vertex1	1	0	1
Vertex2	2	0	1

...

...

...

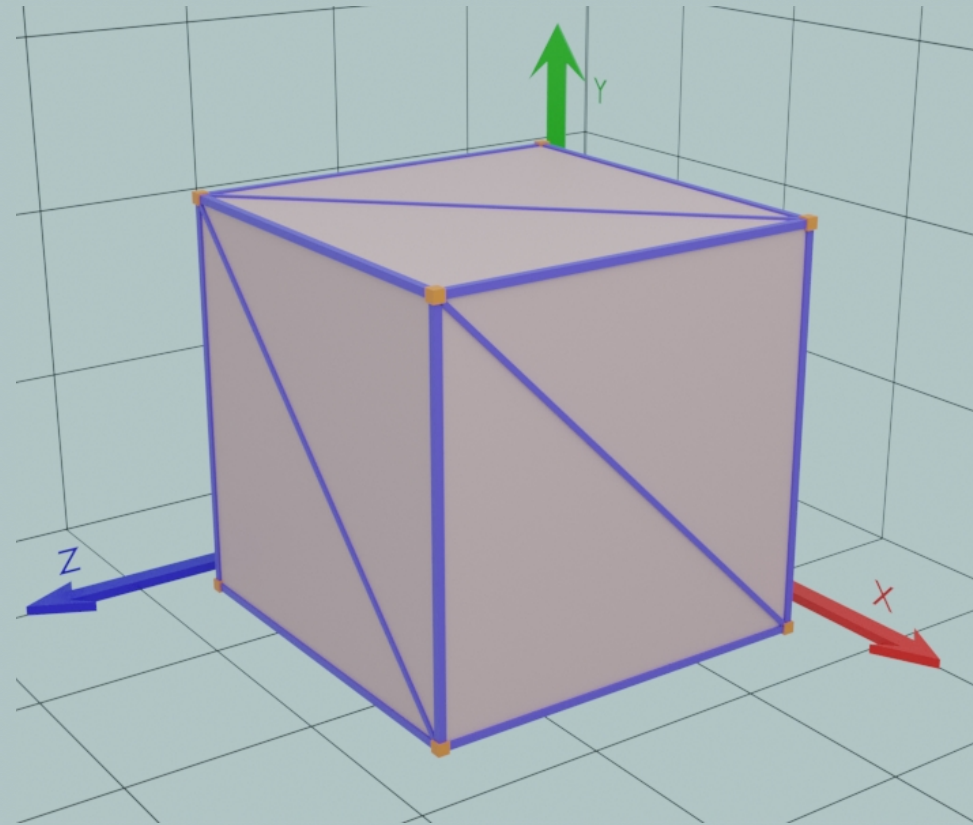
...



How 3D objects work

Edges / Faces

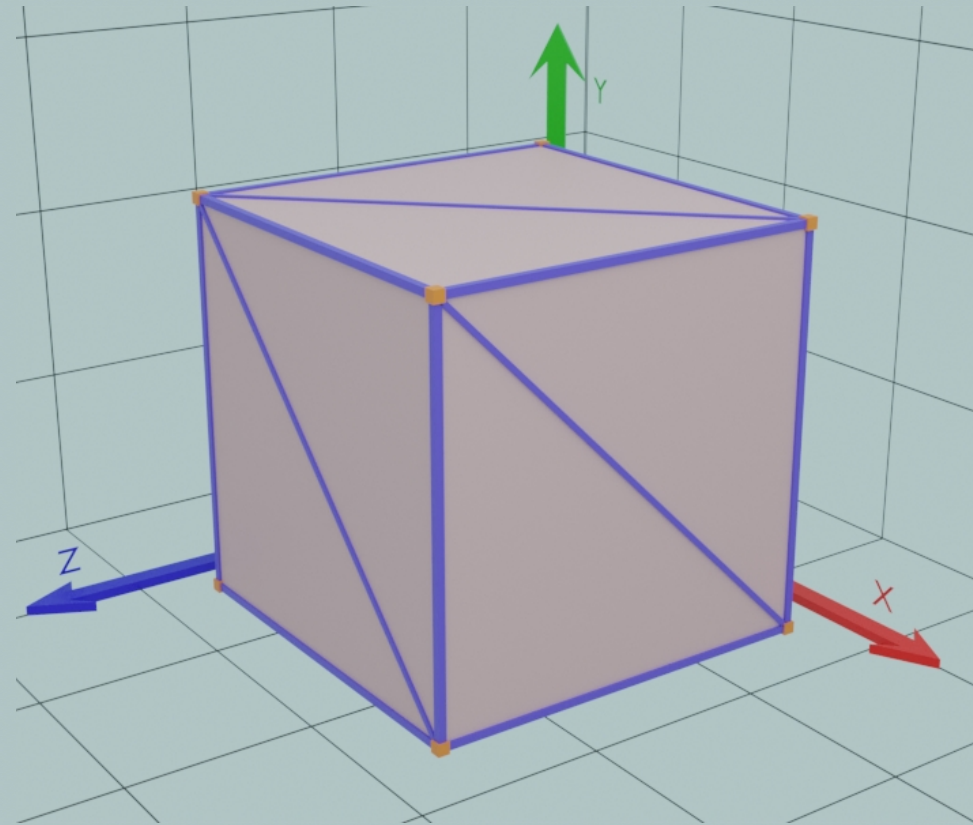
	X	Y	Z
Vertex0	0	1	1
Vertex1	1	0	1
Vertex2	0	0	1
Vertex3	0	1	1
Vertex4	1	1	1
Vertex5	1	0	1
...



How 3D objects work

Edges / Faces

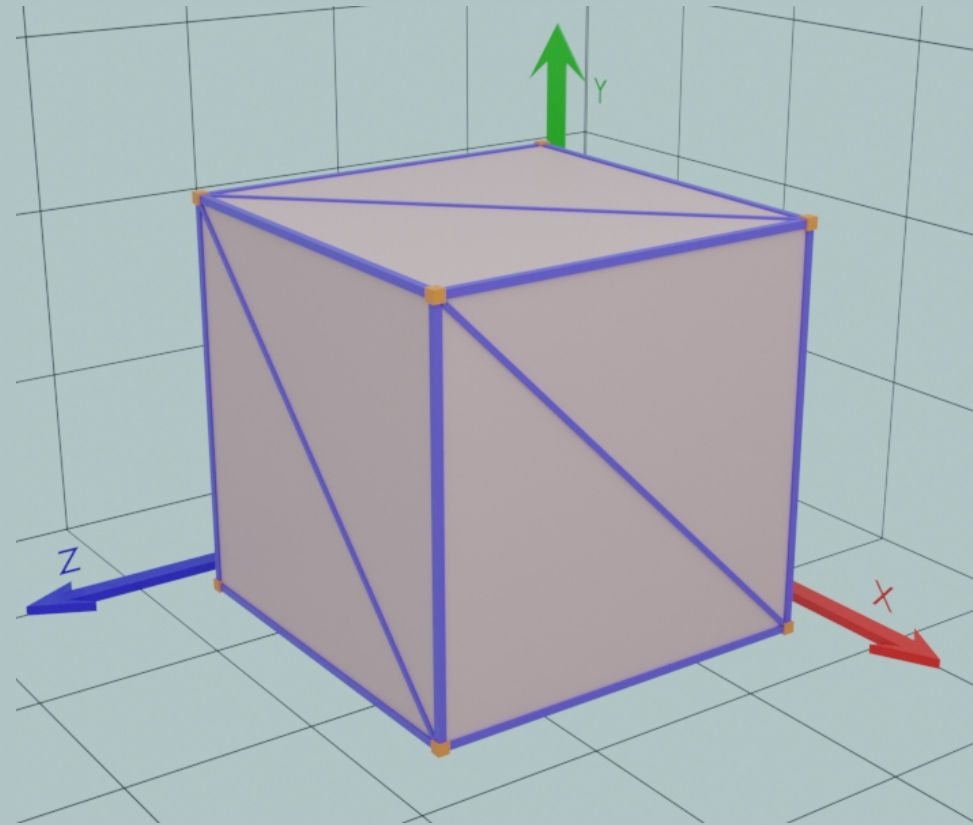
	X	Y	Z
Vertex0	0	1	1
Vertex1	1	0	1
Vertex2	0	0	1
Vertex3	0	1	1
Vertex4	1	1	1
Vertex5	1	0	1
...



How 3D objects work

Edges / Faces

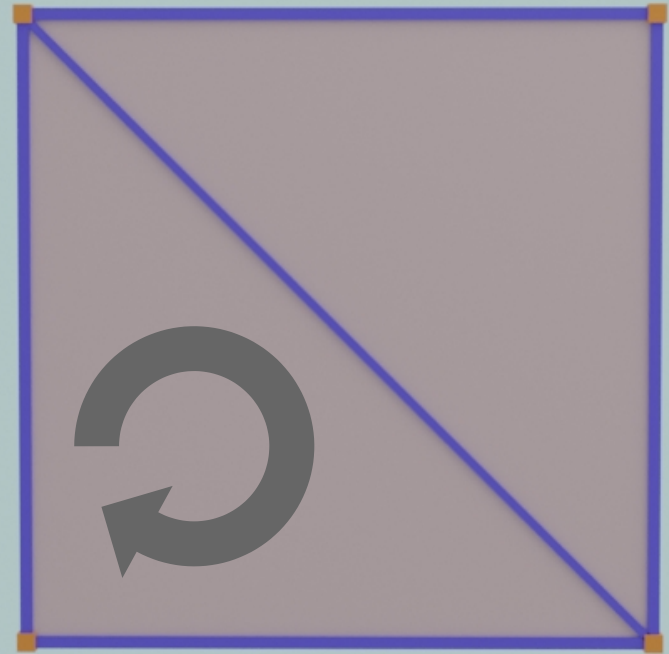
	X	Y	Z
Vertex0	0	1	1
Vertex1	1	0	1
Vertex2	0	0	1
Vertex3	0	1	1
Vertex4	1	1	1
Vertex5	1	0	1
...



How 3D objects work

Edges / Faces

	X	Y	Z
Vertex0	0	1	1
Vertex1	1	0	1
Vertex2	0	0	1
Vertex3	0	1	1
Vertex4	1	1	1
Vertex5	1	0	1
...



How 3D objects work

Edges / Faces

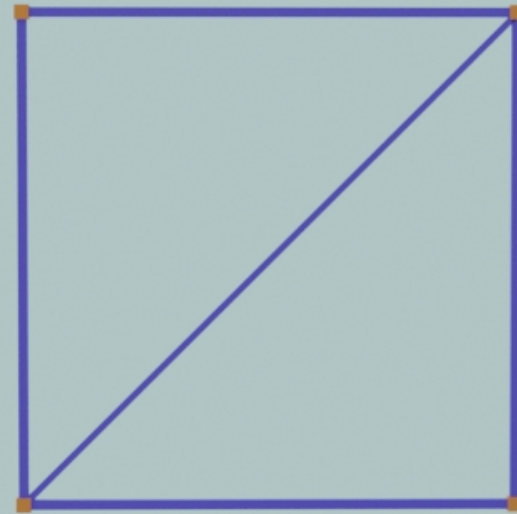
	X	Y	Z
Vertex0	0	1	1
Vertex1	1	0	1
Vertex2	0	0	1
Vertex3	0	1	1
Vertex4	1	1	1
Vertex5	1	0	1
...



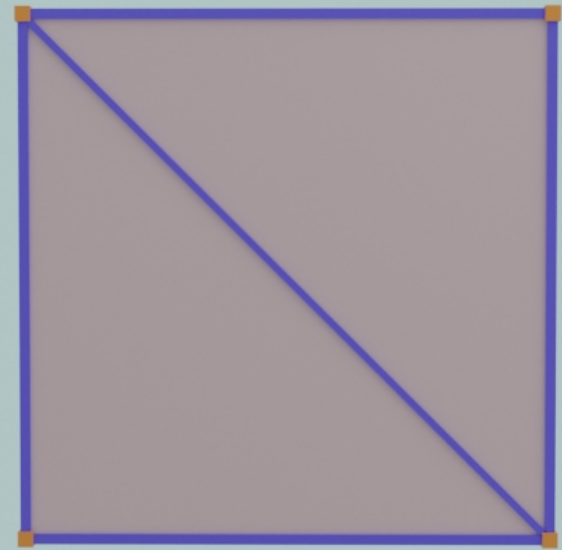
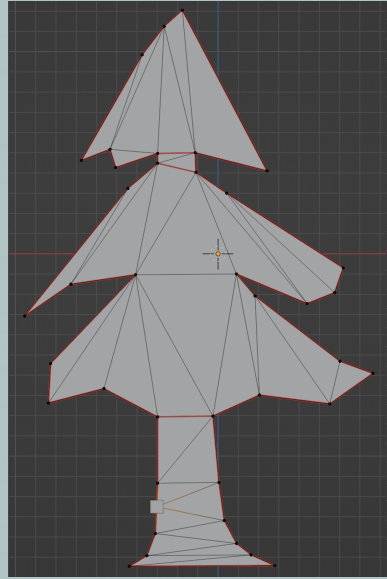
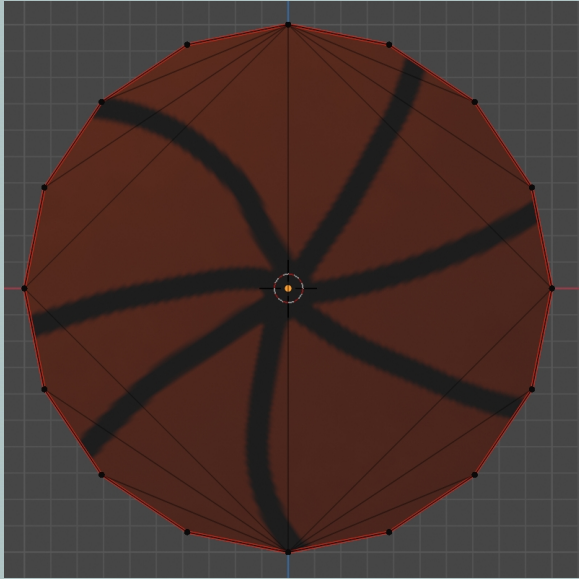
How 3D objects work

Edges / Faces

	X	Y	Z
Vertex0	0	1	1
Vertex1	1	0	1
Vertex2	0	0	1
Vertex3	0	1	1
Vertex4	1	1	1
Vertex5	1	0	1
...

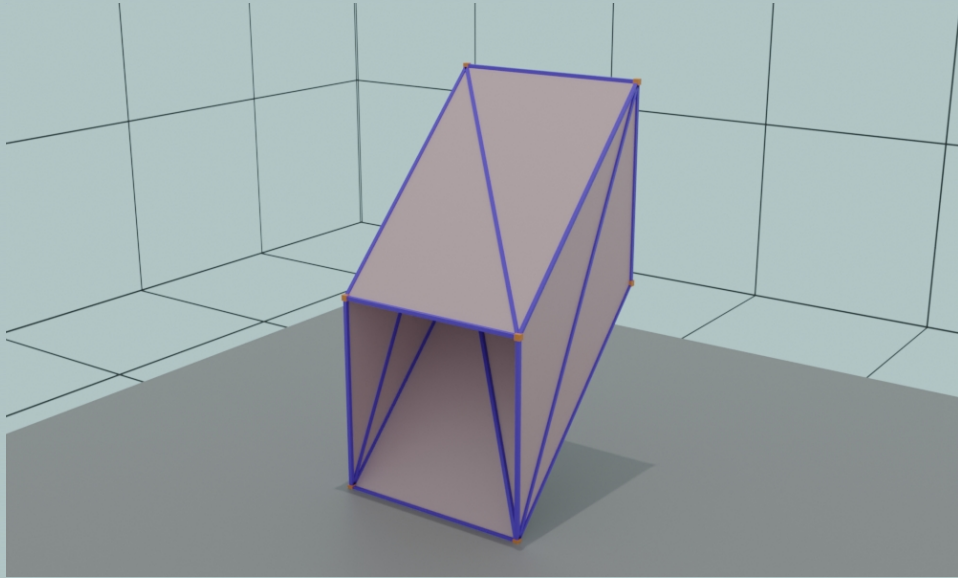


Exporting planes from Blender



`_vertsBase` = vertices of plane at position (0,0,0)

_ObjectVerts

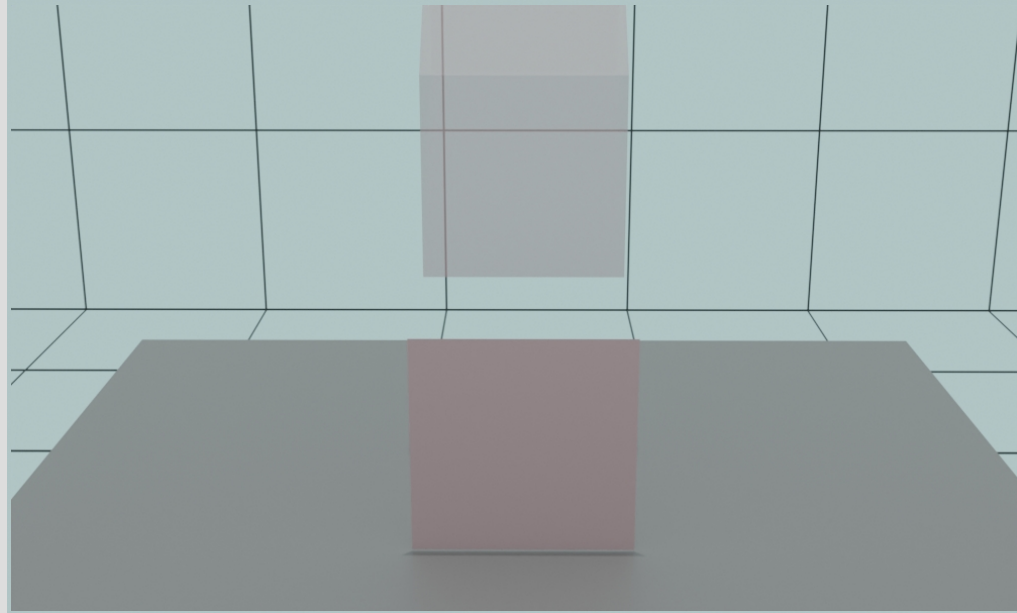


_objectVerts = Only middle parts

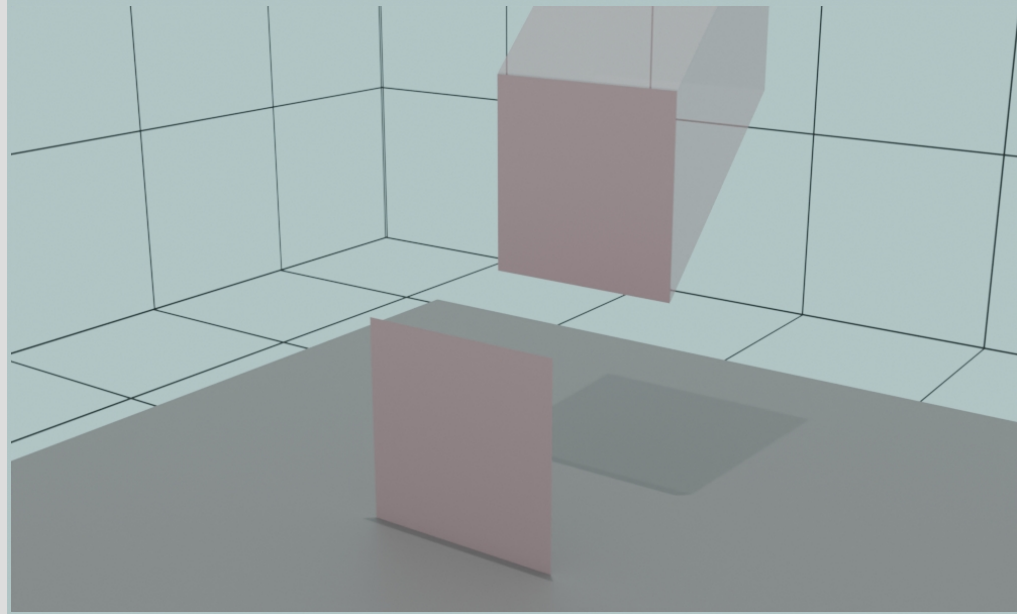
Extending planes in 3D space



Extending planes in 3D space



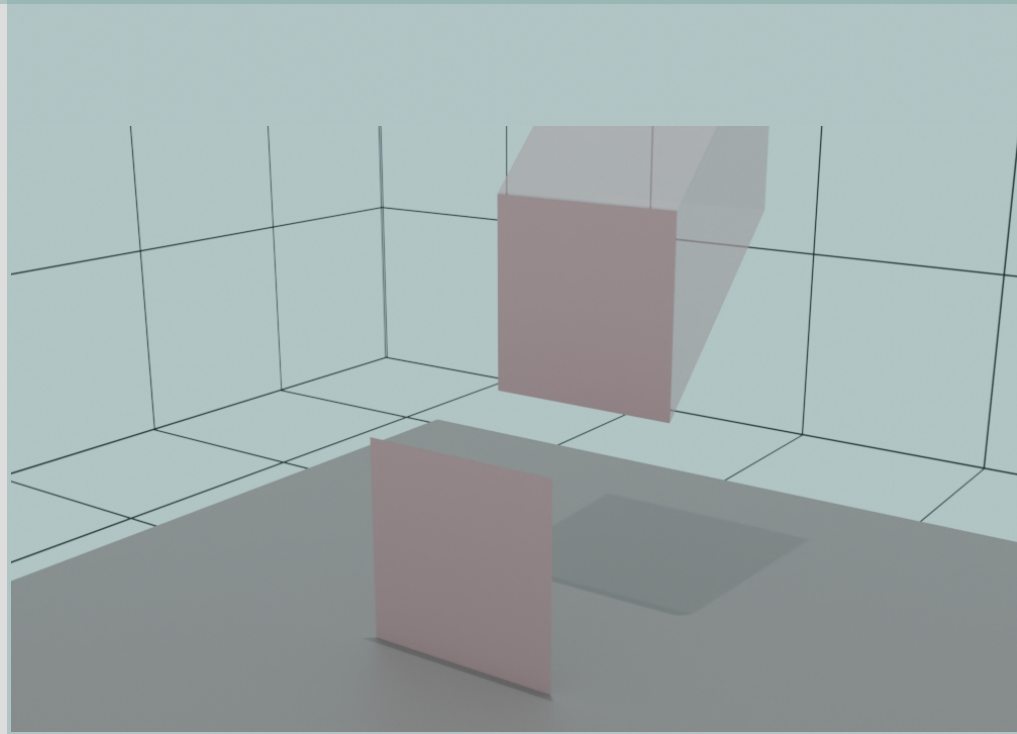
Extending planes in 3D space



Extending planes in 3D space

Known

- `_vertsBase`
- Destination position
(`toPos`)
- `_prevToPos`



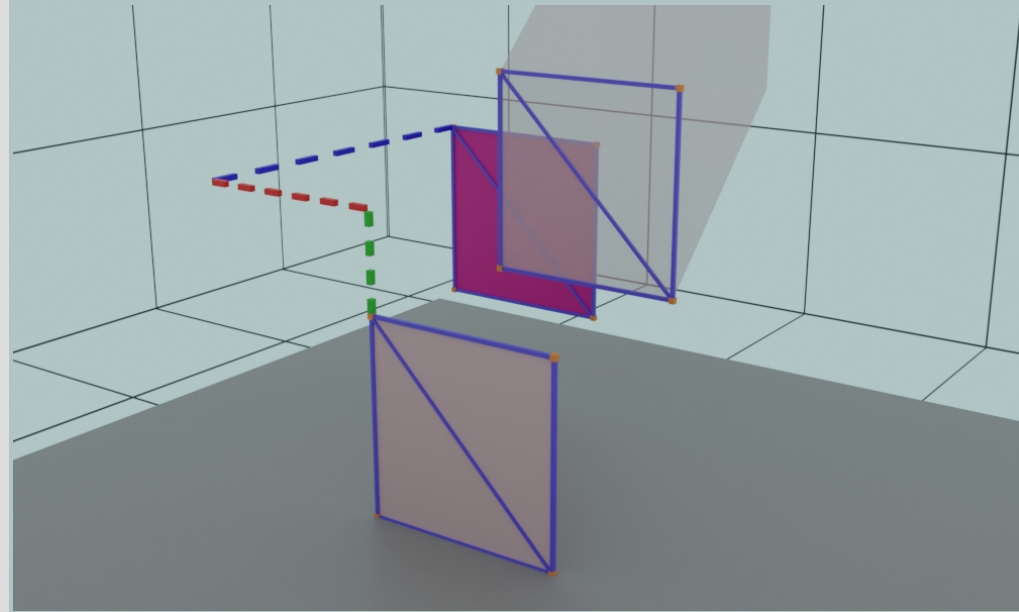
Extending planes in 3D space

`_vertsBase`

	X	Y	Time
Vector(0)	1	1	0
Vector(1)	-1	1	0
...

`cap`

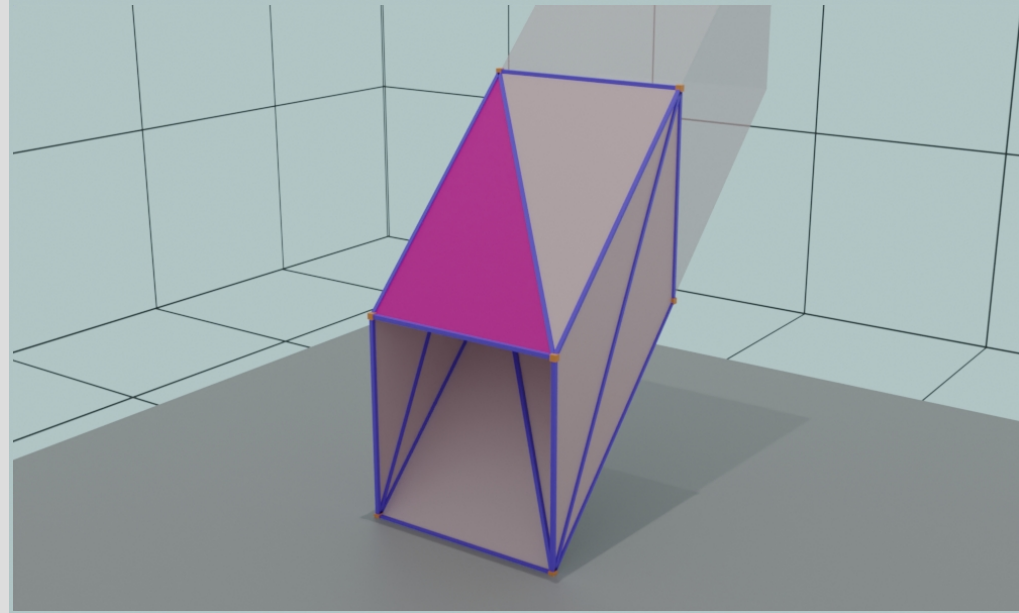
	<code>_vertsBase[i] + toPos</code>
Vector(0)	0
	1



Extending planes in 3D space

Middle (`_objectVerts`)

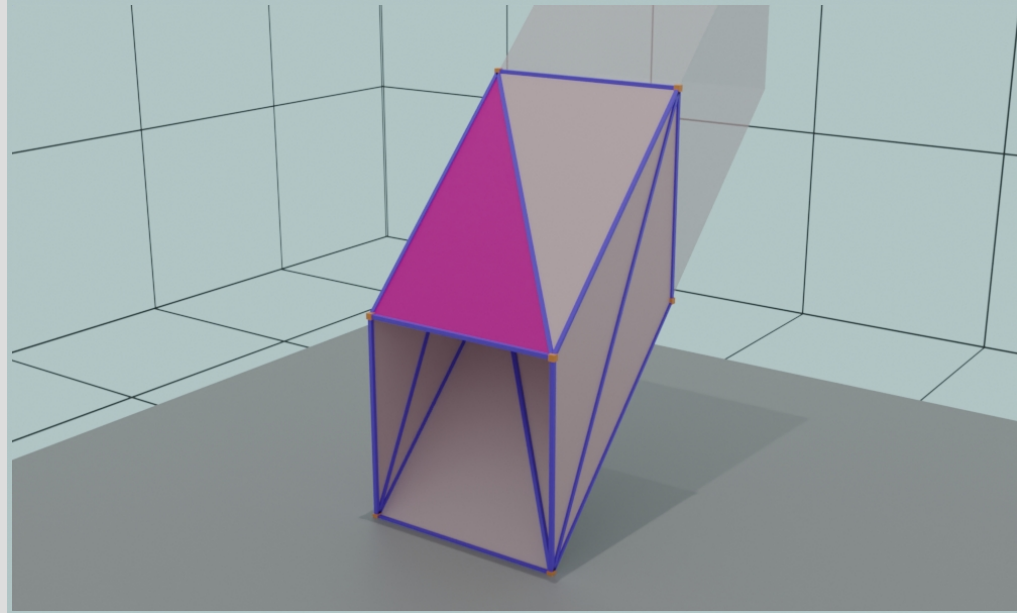
	X	Y	Time
...
Vector(j+0)		?	
Vector(j+1)		?	
Vector(j+2)		?	
...



Extending planes in 3D space

Middle (`_objectVerts`)

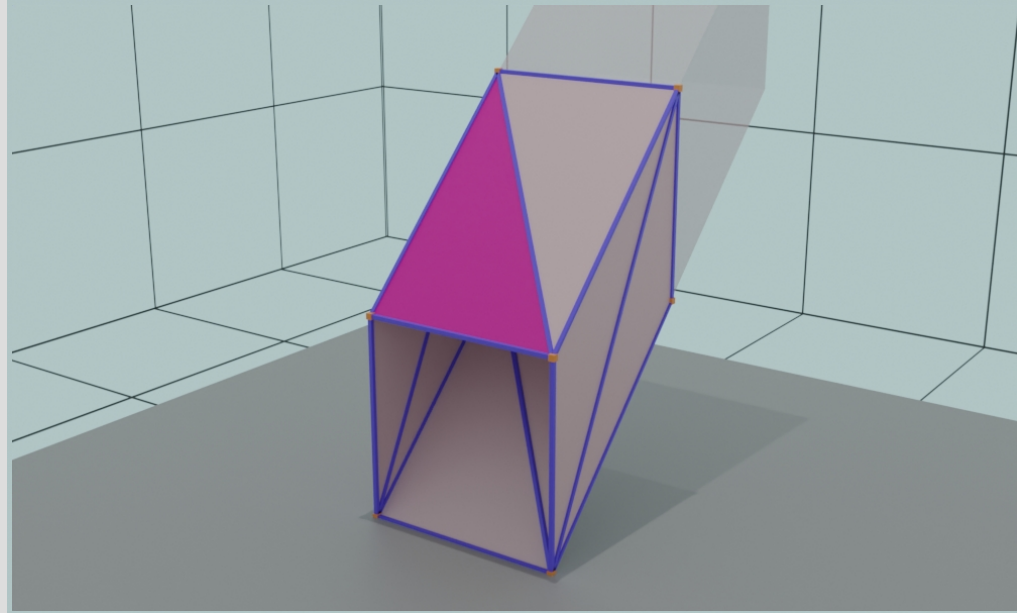
	X	Y	Time
...
Vector(j+0)		?	
Vector(j+1)		?	
Vector(j+2)	<code>_vertsBase[0]</code>	<code>+toPos</code>	
...



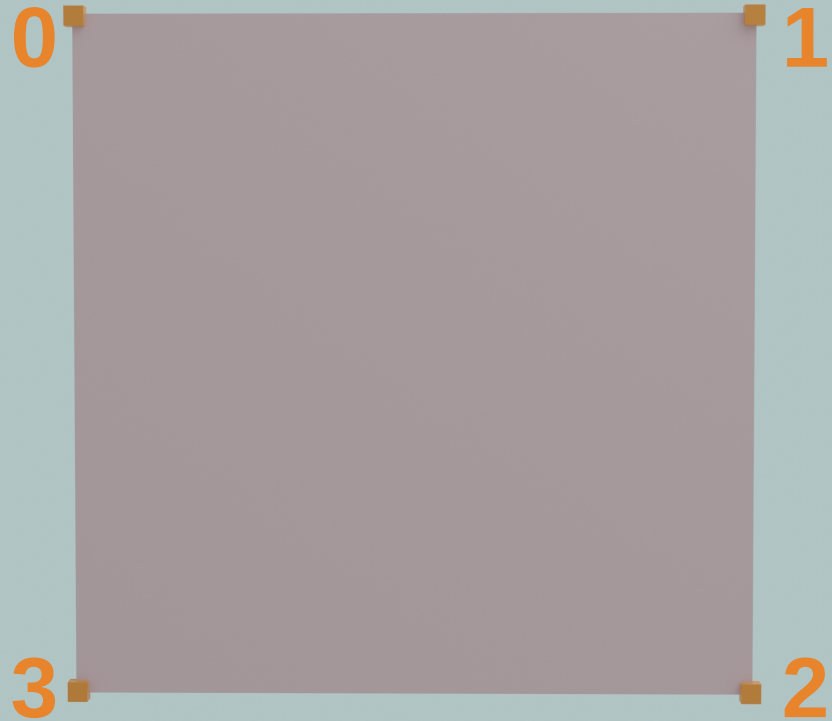
Extending planes in 3D space

Middle (`_objectVerts`)

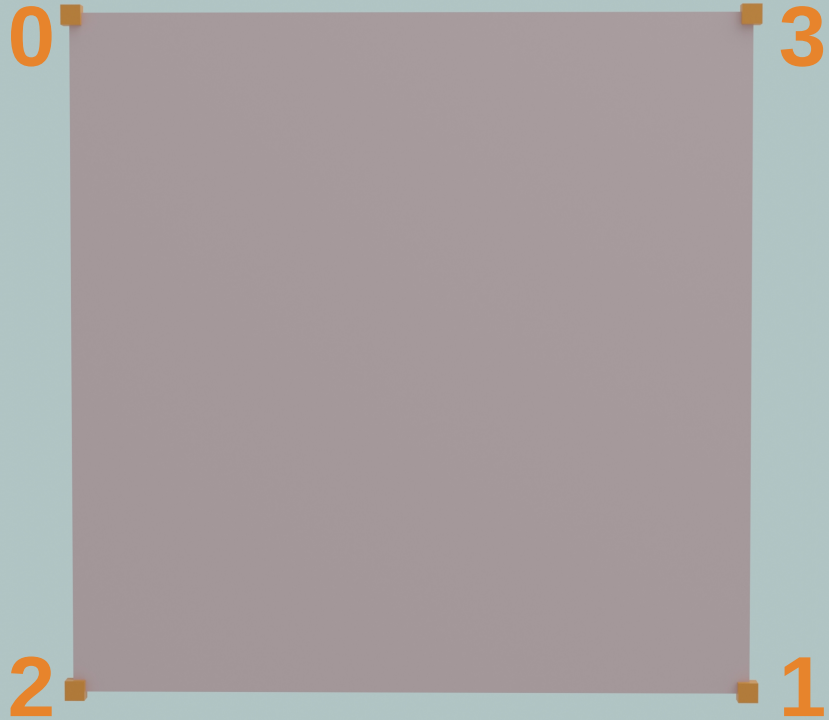
	X	Y	Time
...
Vector(j+0)	<code>_vertsBase[0] + prevToPos</code>		
Vector(j+1)	?		
Vector(j+2)	<code>_vertsBase[0] + toPos</code>		
...



Problem - Vertex indexing

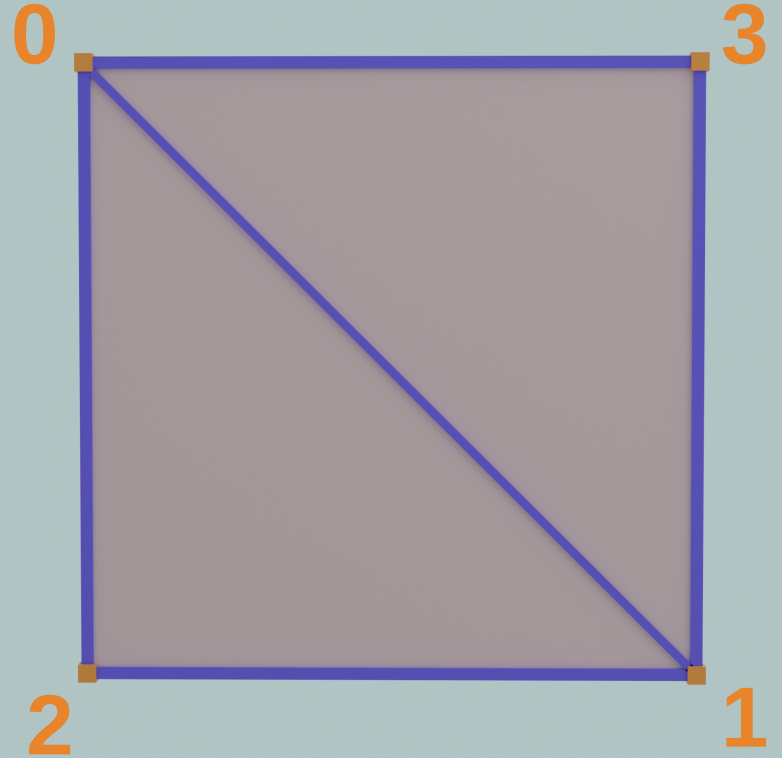


Problem - Vertex indexing



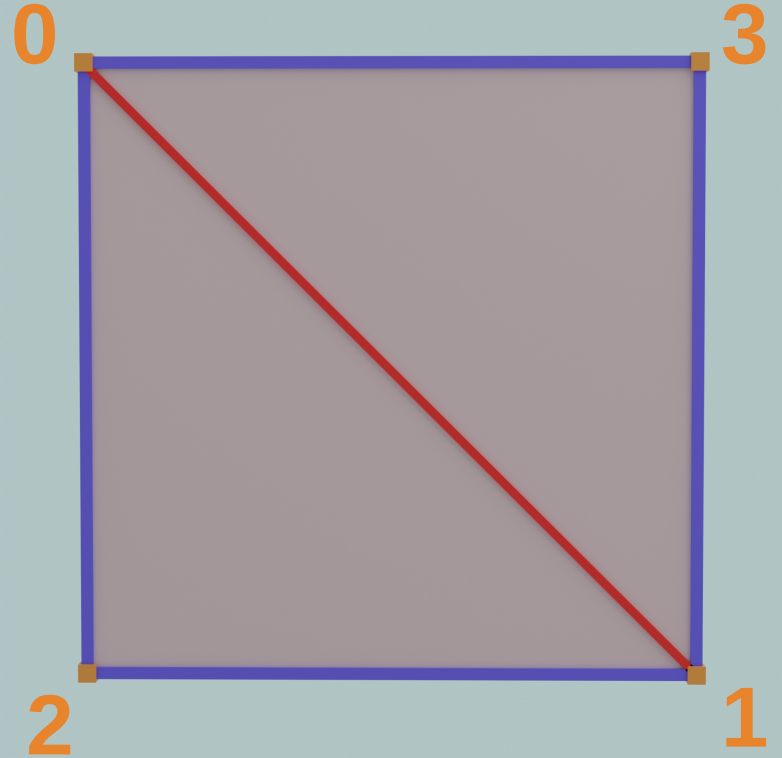
Problem - Vertex indexing

	1 st Vertex	2 nd Vertex
Edge1	0	3
Edge2	3	1
Edge3	1	0
Edge4	0	1
Edge5	1	2
Edge6	2	0



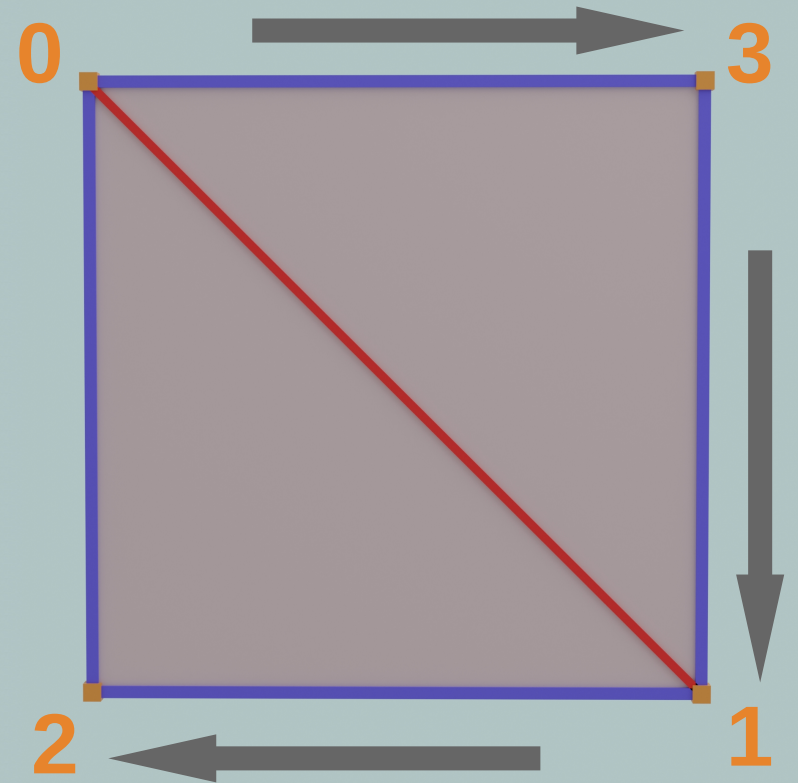
Problem - Vertex indexing

	1 st Vertex	2 nd Vertex
Edge1	0	3
Edge2	3	1
Edge3	1	0
Edge4	0	1
Edge5	1	2
Edge6	2	0

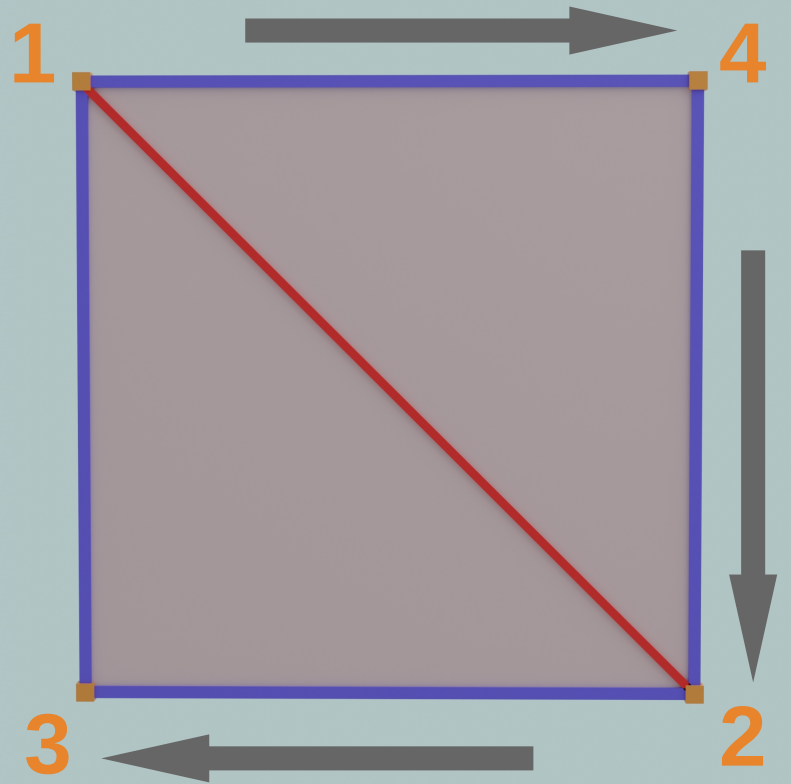
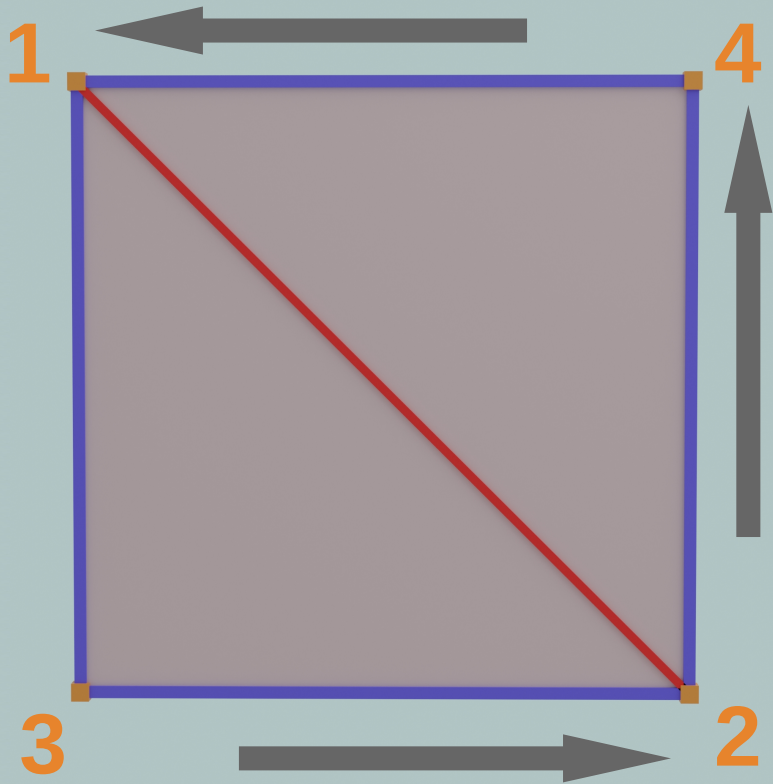


Problem - Vertex indexing

	1 st Vertex	2 nd Vertex
Edge1	0	3
Edge2	3	1
Edge3	1	0
Edge4	0	1
Edge5	1	2
Edge6	2	0



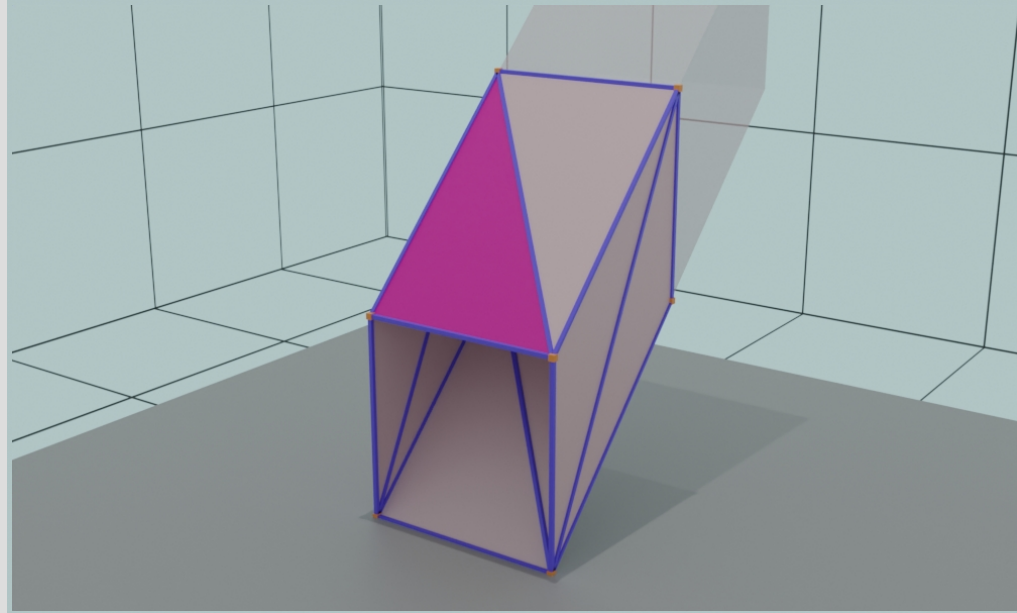
Problem - Winding order



Extending planes in 3D space

Middle (`_objectVerts`)

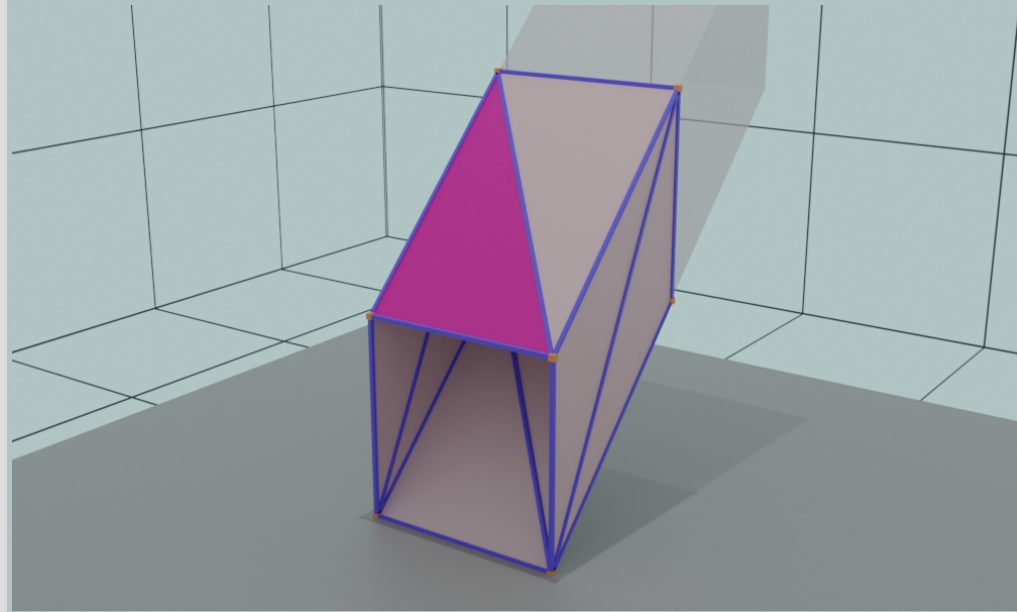
	X	Y	Time
...
Vector(j+0)	<code>_vertsBase[0] + prevToPos</code>		
Vector(j+1)	?		
Vector(j+2)	<code>_vertsBase[0] + toPos</code>		
...



Extending planes in 3D space

Middle (`_objectVerts`)

	X	Y	Time
...
Vector(j+0)	<code>_vertsBase[0] + prevToPos</code>	1	0
Vector(j+1)	<code>_vertsBase[1] +toPos</code>		
Vector(j+2)	<code>_vertsBase[0] +toPos</code>		



Extending Planes in 3D Space

```
var prevToPos
func createExtension(toPos):
    var verts = []
    for i in range(_vertsBase.size()-1):
        verts.append(_vertsBase[i] + toPos)
        verts.append(_vertsBase[i+1] + prevToPos)
        verts.append(_vertsBase[i+1] + toPos)

        verts.append(_vertsBase[i] + toPos)
        verts.append(_vertsBase[i] + prevToPos)
        verts.append(_vertsBase[i+1] + prevToPos)

    ...
```

Extending Planes in 3D Space

...

```
verts.append(_vertsBase[_vertsBase.size()-1] + toPos)
```

```
verts.append(_vertsBase[0] + prevToPos)
```

```
verts.append(_vertsBase[0] + toPos)
```

```
verts.append(_vertsBase[_vertsBase.size()-1] + toPos)
```

```
verts.append(_vertsBase[_vertsBase.size()-1] + prevToPos)
```

```
verts.append(_vertsBase[0] + prevToPos)
```

```
prevToPos = toPos
```

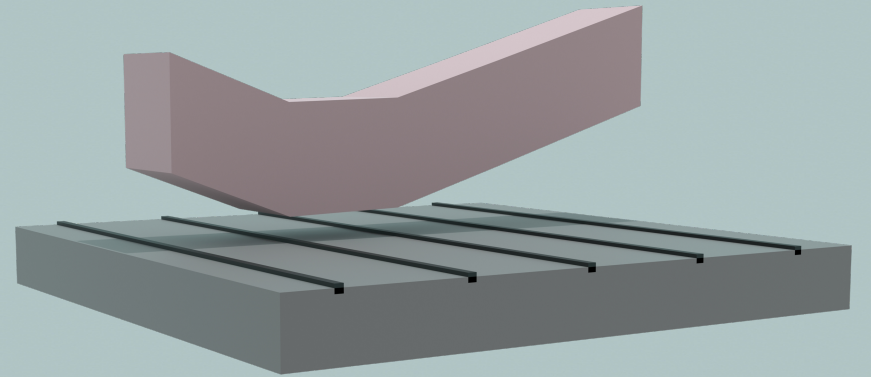
```
return verts
```

Extending Planes in 3D Space

```
func addExtensionToMesh(verts):  
    _objectVerts.append_array(verts)  
    _objectUVs.append_array(_uvcoords)  
    finalizeExtension()
```


When extends happen

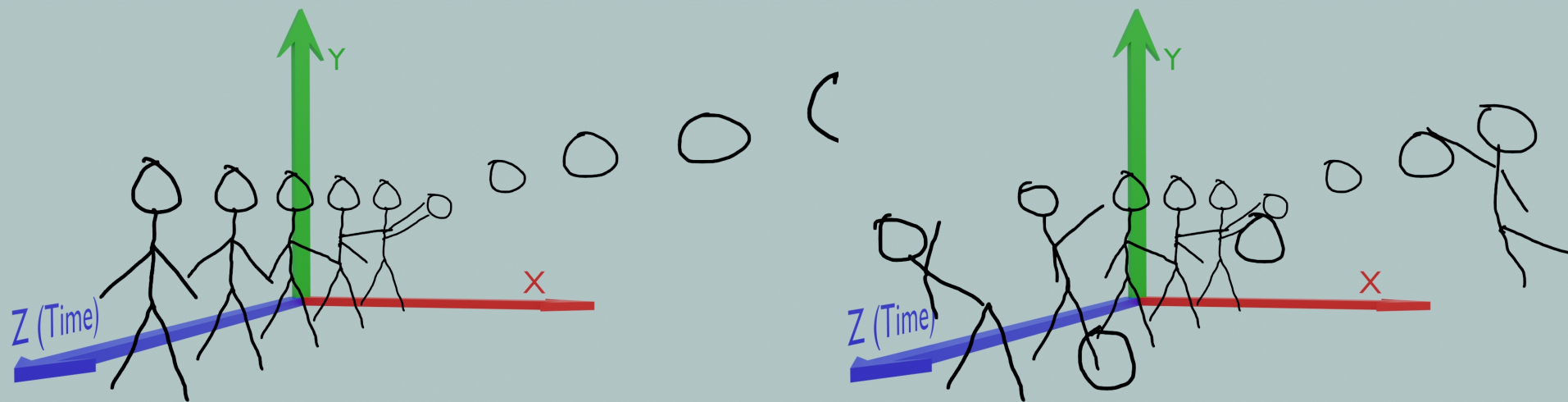
- After x seconds have passed
- On collisions



Physics

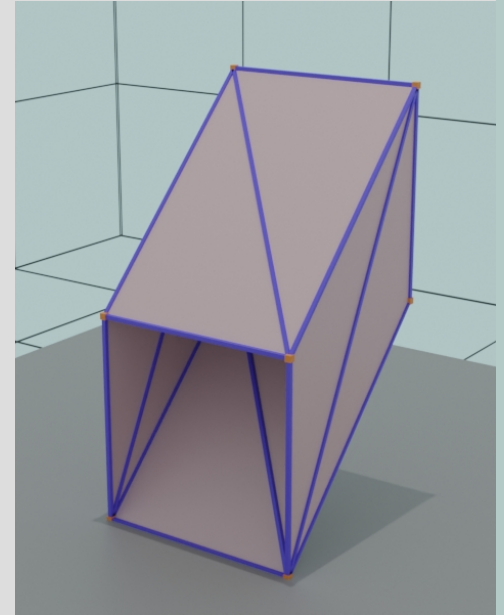
- Using Godot built-in physics and collision detection
- For interactions vertices need to be deleted
- Figure out the exact position we interacted with

Physics



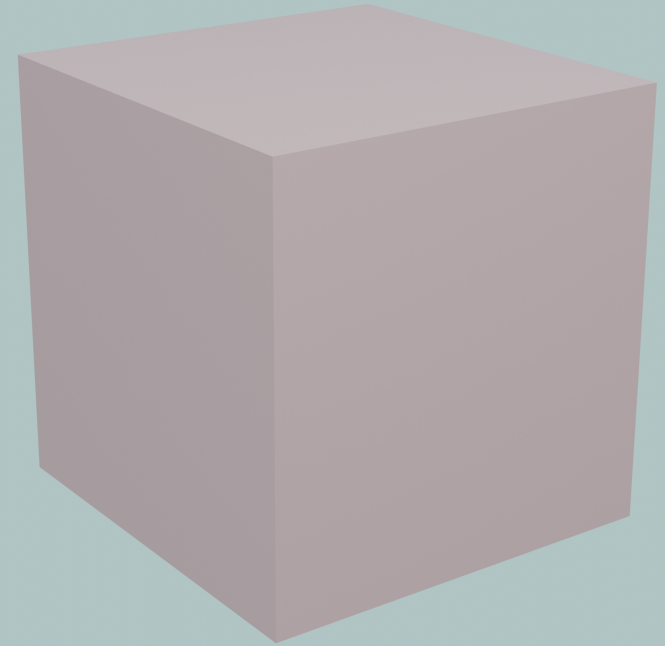
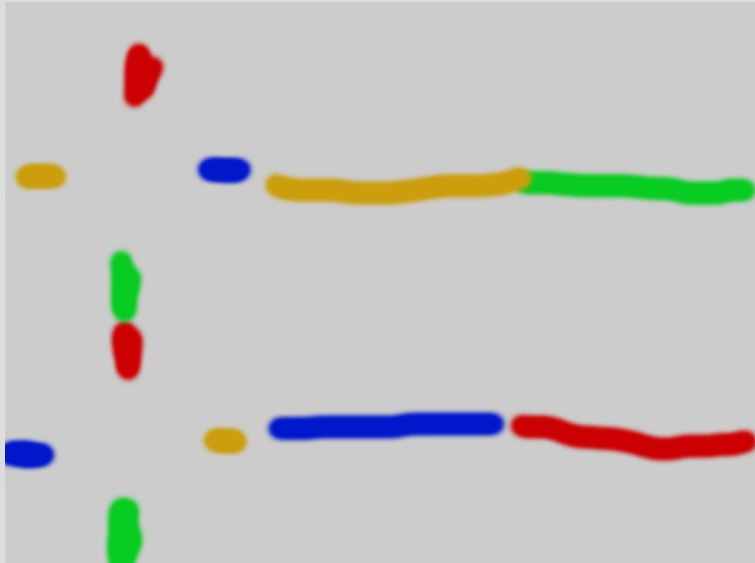
Physics

```
func deleteExtensionsFromMesh(pos):  
    for i in range(_objectVerts.size()/(6*_vertsBase.size()),0,-1):  
        if(_objectVerts[i*6*_vertsBase.size()-3].z>=pos.z):  
            for j in range(_vertsBase.size()*6):  
                _objectVerts.remove(_objectVerts.size()-1)  
                _objectUVs.remove(_objectVerts.size()-1)
```

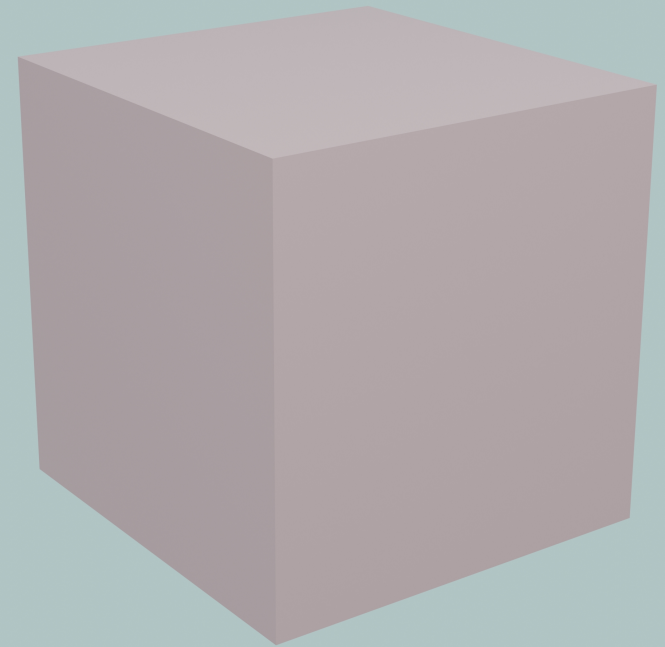
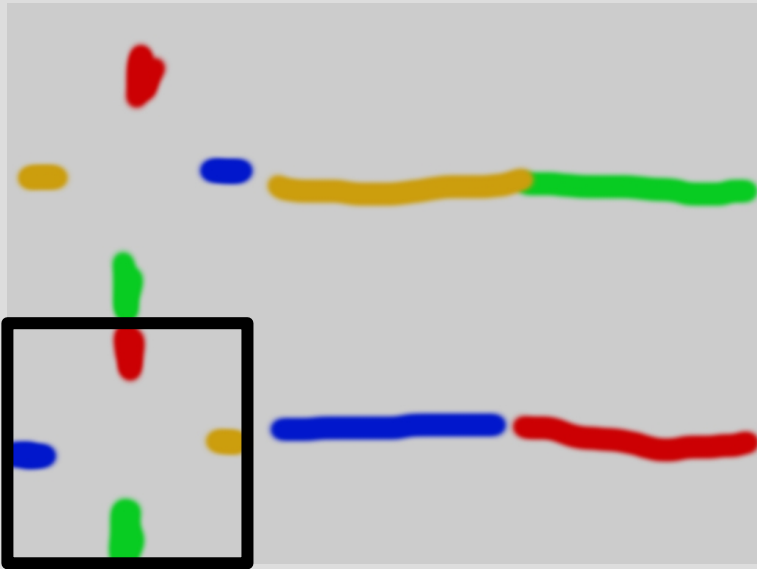


```
_lastToPos = _objectVerts[_objectVerts.size()-3]-_vertsBase[_vertsBase.size()-1]  
addExtensionToMesh(createExtension(pos))
```

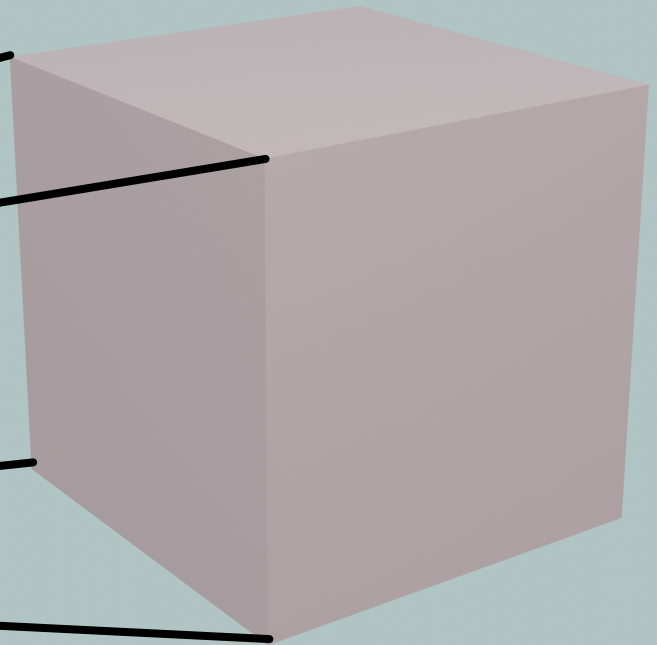
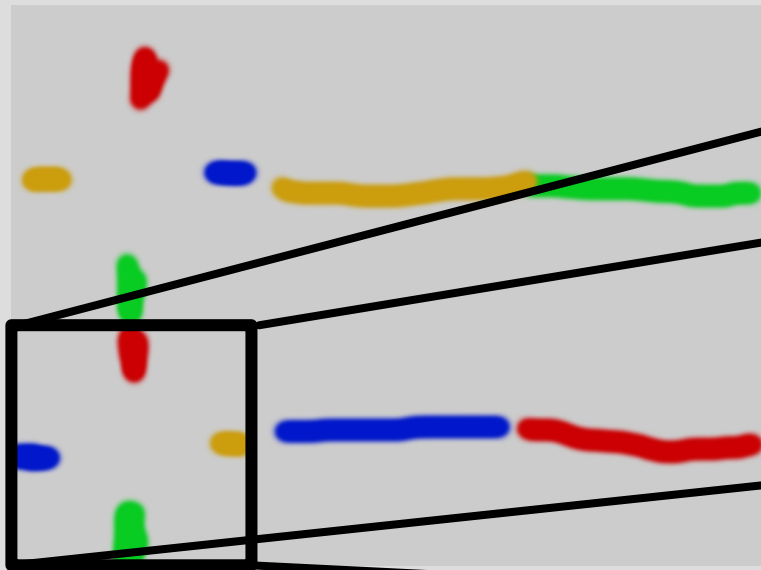
Adding Textures



Adding Textures



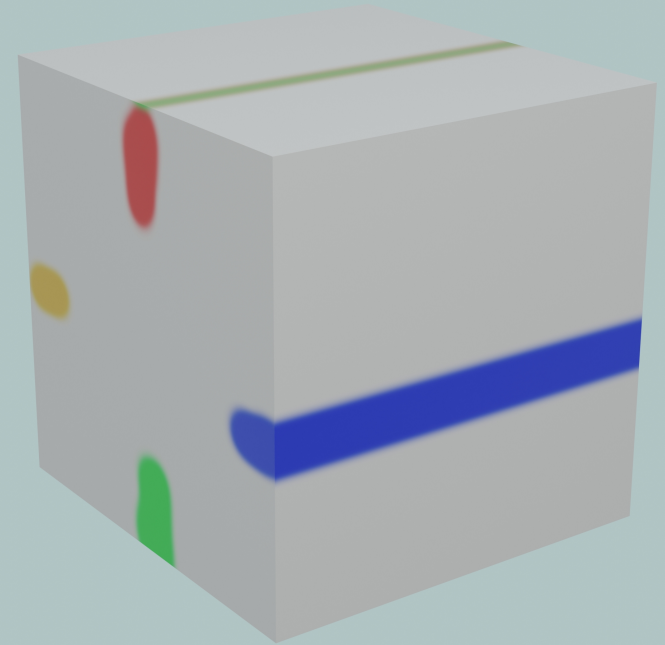
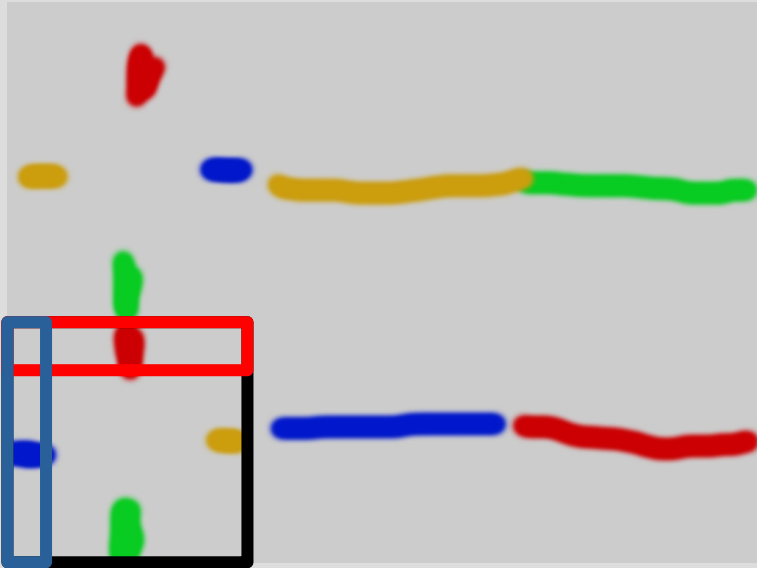
Adding Textures



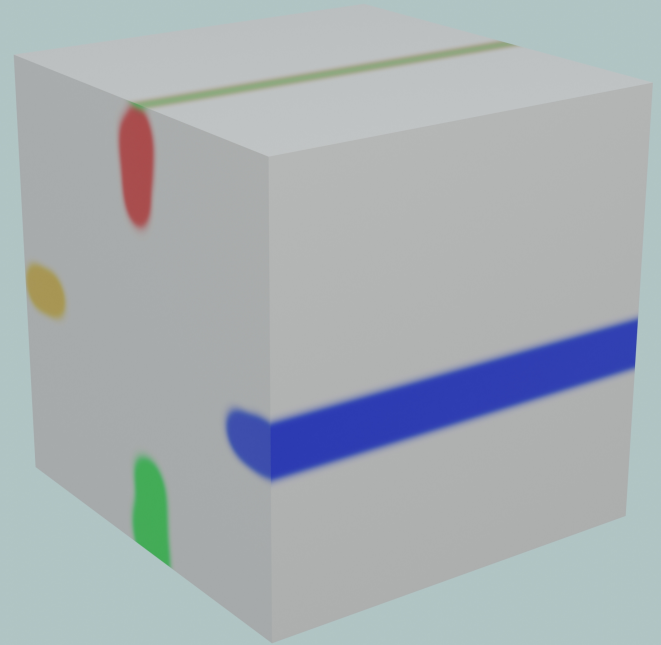
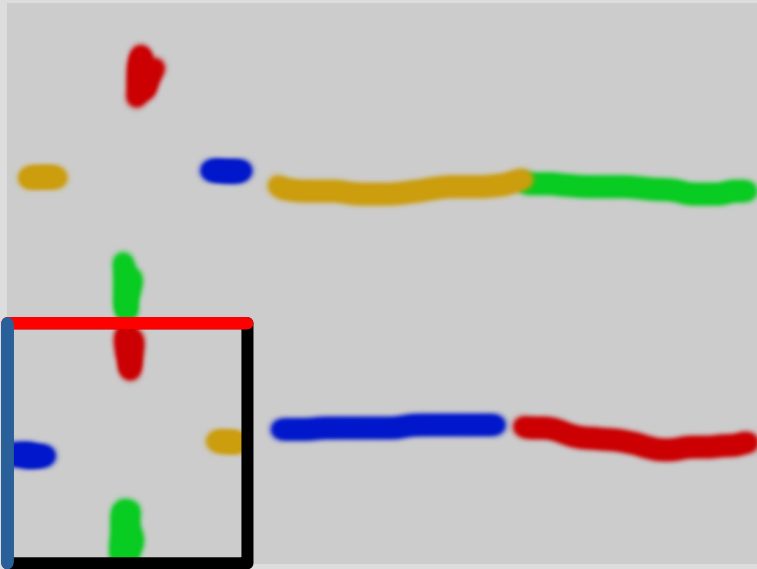
Adding Textures



Adding Textures



Adding Textures



Adding Textures

```
func addExtensionToMesh(verts):  
    _objectVerts.append_array(verts)  
    _objectUVs.append_array(_uvcoords)  
    finalizeExtension()
```

```
var _uvcoords = PoolVector2Array()  
func _getUvCoords(mdt, baseMesh):  
    ...
```

Future

- Adding rotation
- Adding collision between movable objects
- Adding segments in bigger chunks
- Making it smoother

Are there any Questions?