

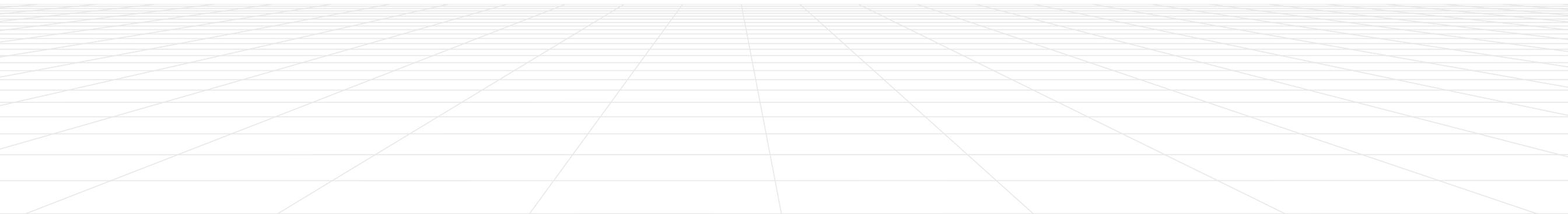
SketchVR

FUNCTIONALITIES DESIGN

TEAM 5FLAGS
CENTRE FOR DIGITAL MEDIA

CONTENTS

SECTION	PAGE	OVERVIEW
User Interface	1	<p>The basic purpose of this FDD is to provide information on the concepts picked for the MVP development of SketchVR. It includes an overview of functionalities which were successfully implemented. Functions to be implemented are also stated in this document.</p> <p>It also consist of the Technical Guide which covers the programming structure and implementation done for SketchVR development.</p>
Toolbox Interface	2	
View Controls	3	
Dependencies	4	
References	5	
Functions	7	
Controls	20	
Technical Guide	21	



FUNCTIONALITIES

USER INTERFACE

Here's a basic breakdown of the interface elements that forms a virtual reality visual base in SketchVR. These segments combined creates immersive and interaction in virtual space.

Environment	A 360 degrees VR space
Grid	Floor section
Skybox	Sky section
Direction	Visual clue for direction facing
Light	Default light source
Controls	Left and Right hand controls
Toolbox	Create UI panel

FUNCTIONALITIES

CREATE TOOLBOX

Minimal tools required to create any low-high fidelity modal in SketchVR. These are the very basic tools that allow a user to have control over using the very basic functions in any creation program.

File	Load, Save options
Pointer	Select and Deselect tool
Objects	Plane, Cube, Cylinder, Sphere shapes
Line	Pen tool
Extrude	Push or Pull tool
Transform	Move, Rotate, Scale tools
Duplicate	Copy and Paste tool
Delete	Remove tool
Paint	Color tool

FUNCTIONALITIES

VIEW CONTROLS

The ability to observe the environment from different view. It give the user 360 degree freedom on top of walking in real for faster access to different angles.

- Navigate** Allow user to move on XYZ axis
- Zoom** Zoom-in, Zoom-out, Reset options
- Orbit** Allow user to Revolve around the environment

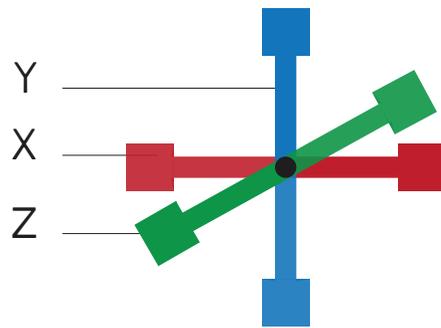
DEPENDENCIES

Here's a list of functions dependences; it states the accessibility of each tool in relation to other tools. The basic purpose of this table is to show tools hierarchy over different levels.

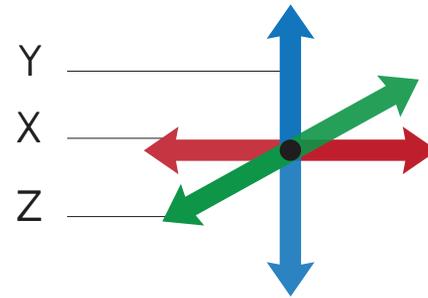
LEVEL	CASE	TOOL	DEPENDENT	ACCESSIBILITY
Primary	A	File	No	Anytime
		Shapes	No	Anytime
		Line	No	Anytime
Secondary	B	Pointer	Yes	Anytime after A
Tertiary	C	Extrude	Yes	Anytime after B applied to a face
		Transform		
		Move	Yes	Anytime after B applied to an edge, a line, a face or an object
		Rotate	Yes	Anytime after B applied to a line, a face or an object
		Scale	Yes	Anytime after B applied to a face or an object
		Duplicate	Yes	Anytime after B applied to a line, a face or an object
		Delete	Yes	Anytime after B applied to an edge, a line, a face or an object
		Paint	Yes	Anytime after B applied to a face or an object

REFERENCE

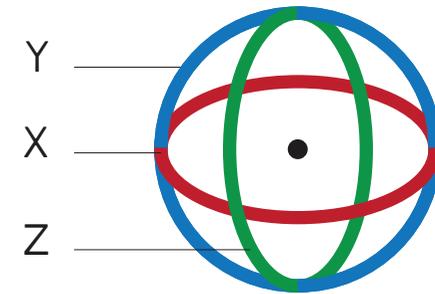
SCALE



MOVE



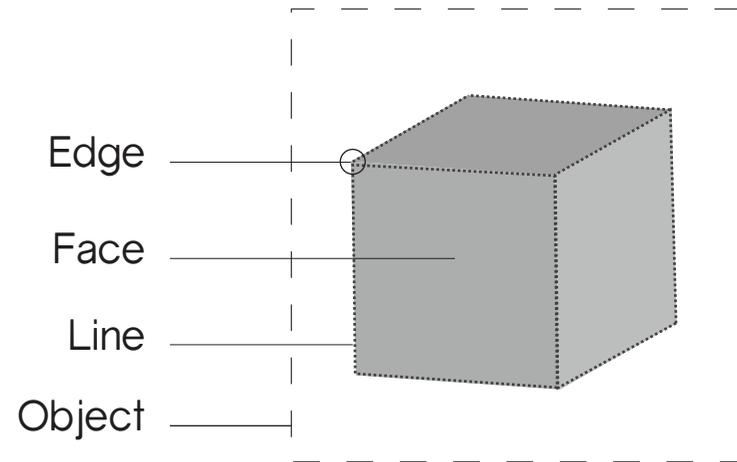
ROTATE



Scaling, Moving and Rotating are core functions that can be axis dependent or not. Once they are axis dependent then the user can only manipulate on a certain axis. The user can also manipulate with free transformation with no fixed dependency on any axis. These symbols and styles can be used for transformation gizmos.

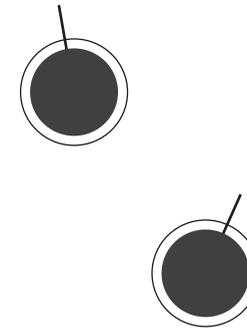
REFERENCE

OBJECT



Every shape is a combination
of one or more of an object
sub-sections

CONTROLS



Left and Right hands in VR
can be seen in form of lasers

FUNCTIONS

This section covers all the basic functions which are implemented and are yet to be implemented. It acts as a guide for future development of functions which will add to the basic creation controls for user while adding on to the overall user experience of the product. What you see in these functions section are low fidelity functional concepts used for logic explanation purpose.

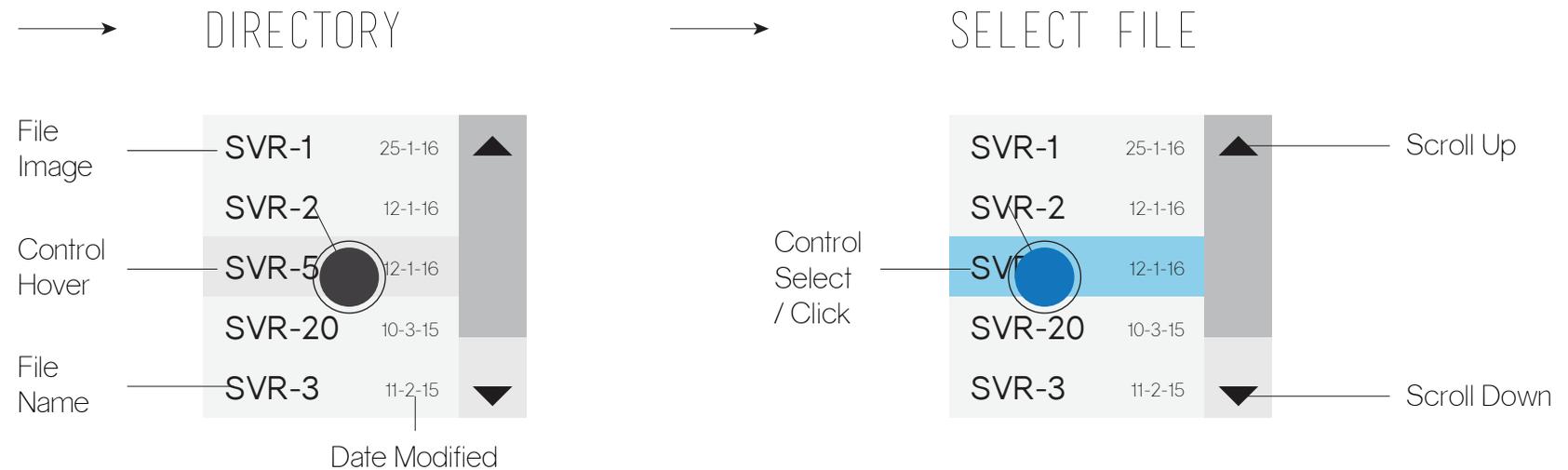
Color reference for functions

-  IMPLEMENTED
-  PARTIALLY IMPLEMENTED
-  TO BE IMPLEMENTED

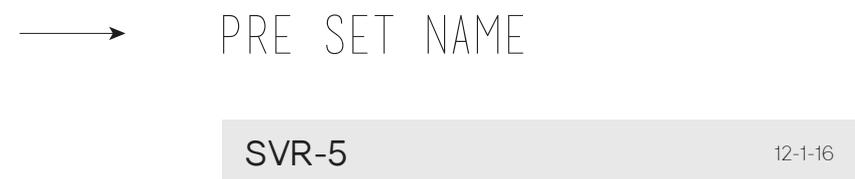
FILE

File system lets the user save their current project or load a previously saved project. There are various ways to implement the file system.

LOAD



SAVE



Screen Notification with File name, date and picture for clear information.

SHAPES

All shapes have been implemented except plane. There are two major integration options for shapes tool (i) the user can click and hold or (ii) click and release the interaction trigger to draw a shape. Measurement visual reference is yet to be implemented.

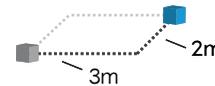
● PLANE



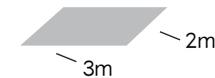
1. Start Point
Start at floor



2. Draw Width
 $z = \text{width}$



3. Complete Plane
 $z = \text{width}$



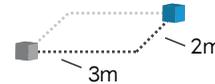
● CUBE



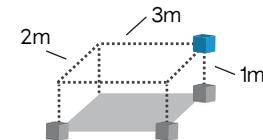
1. Start Point
Start at floor



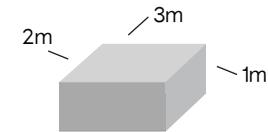
2. Draw Width
 $z = \text{width}$



3. Draw Height
 $z = \text{width}$



4. Complete Cube
 $z = \text{width}$



SHAPES

All shapes have been implemented except plane. There are two major integration options for shapes tool (i) the user can click and hold or (ii) click and release the interaction trigger to draw a shape.

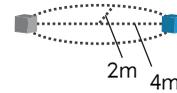
● CYLINDER



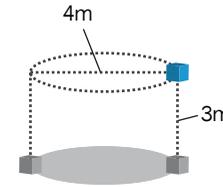
1. Start Point
Start at floor



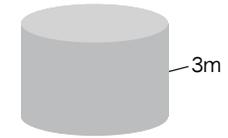
2. Draw Diameter
 $z = \text{radius}$



3. Draw Height
 $x = \text{diameter}$



4. Complete Cylinder
 $y = \text{height}$



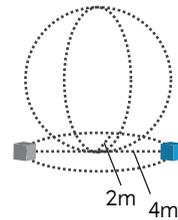
● SPHERE



1. Start Point
Start at floor



2. Draw Diameter
 $x = \text{diameter}$
 $z = \text{radius}$



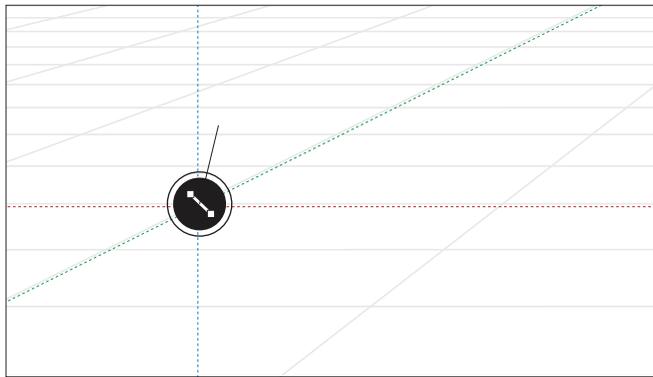
3. Complete Sphere
 $x = \text{diameter}$
 $y = \text{height}$



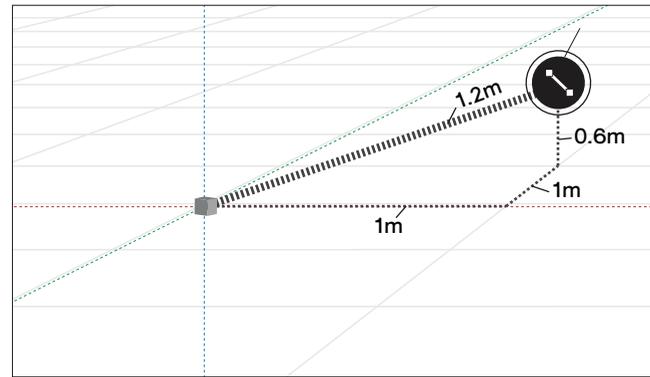
● LINE

Line tools needs to be implemented as it gives the user the ability to create custom shapes. There are a minimum of 4 steps required to create an enclosed shape using the line tool.

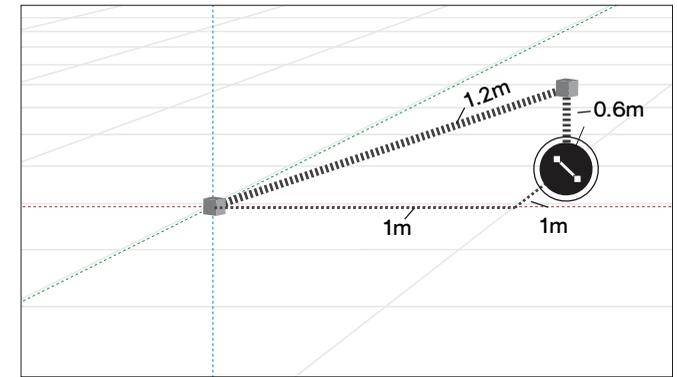
Start Point



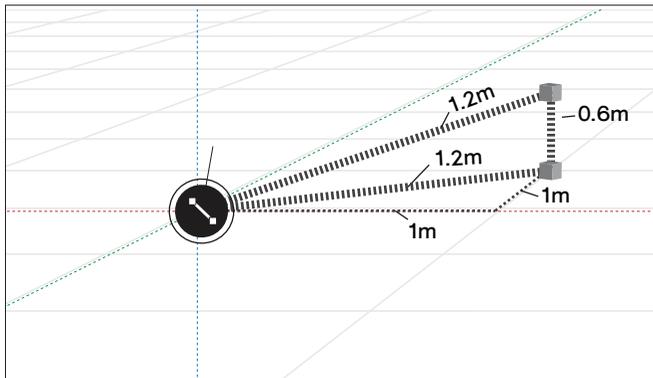
Line Point



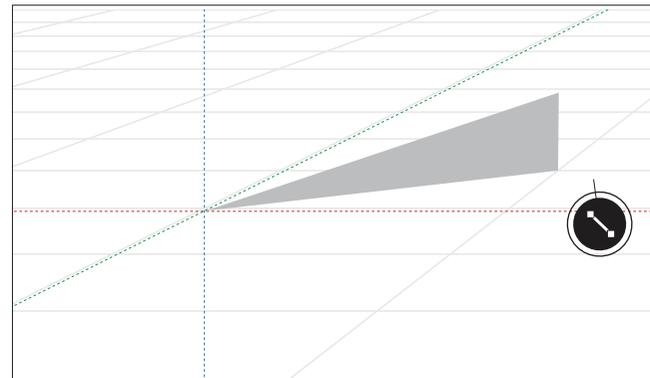
Multi Line Point



Connect Point



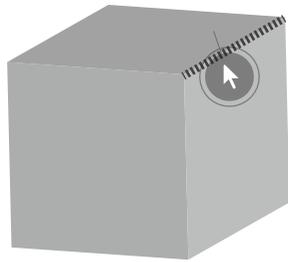
Enclosed Shape



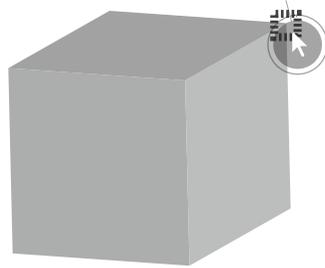
SELECT

Select tool is implemented for a complete object only. For greater control over object selection, line, edge and face cases have to be implemented. Lasers are used as the interaction medium on controls for the select tool.

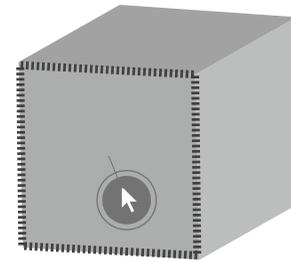
● Line



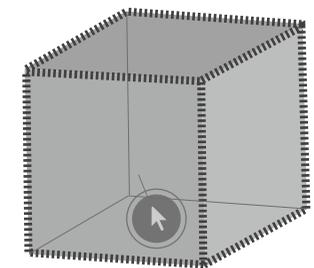
● Edge



● Face



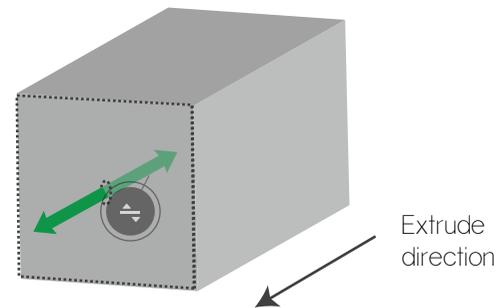
● Object



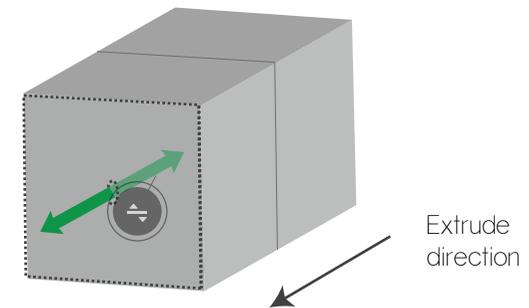
EXTRUDE

Simple extrude is implemented where the user can extrude any face and extend or reduce it. Multiple extrude can be implemented where an object replicates as a face is extruded. Multiple extrude creates a new object instead of continuing on the same

● Simple Extrude



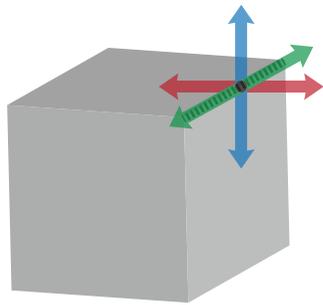
● Multiple Extrude



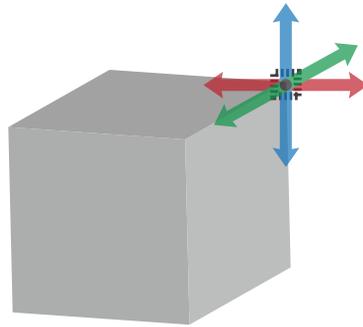
MOVE

Move function is implemented for a complete object. Line, edge and face move function is yet to be implemented. Face move function is similar to simple extrude, but it's only moved on a single axis. Object axis dependent or free-axis move is implemented.

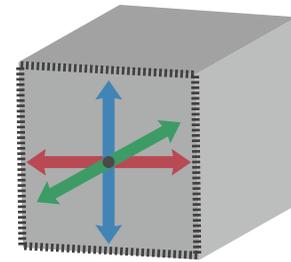
● Line



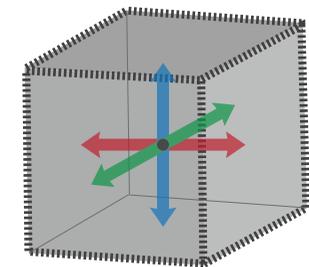
● Edge



● Face

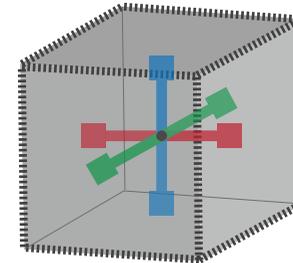
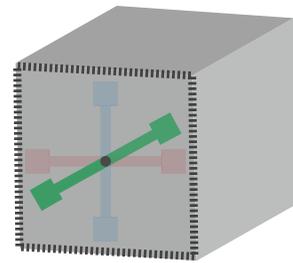


● Object



SCALE

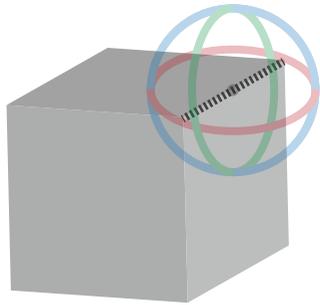
Scale function is implemented for an object's face only. It scales the object face in both directions on the selected axis. Axis with the selected face can be scaled, other axis control is yet to implemented. Universal scaling or an object is yet to be implemented.



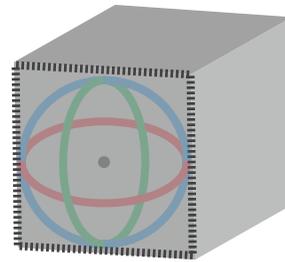
ROTATE

Rotate function is implemented for a complete object. Line and face rotate function is yet to be implemented. Object axis dependent or free-axis rotation is implemented.

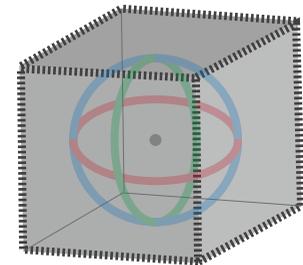
● Line



● Face



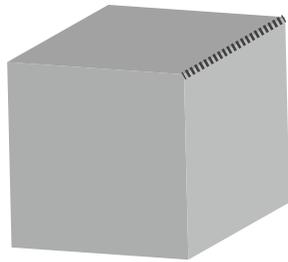
● Object



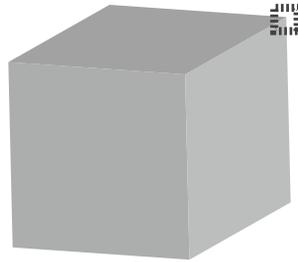
REMOVE

Remove function is implemented for a complete object. Line, edge and face remove function is yet to be implemented.

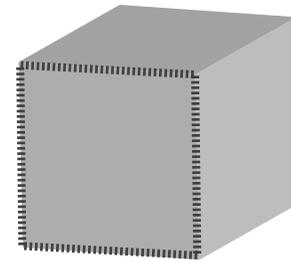
● Line



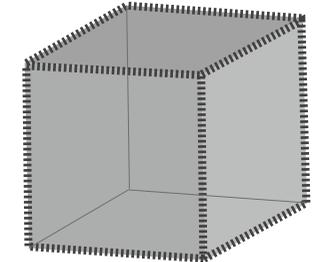
● Edge



● Face



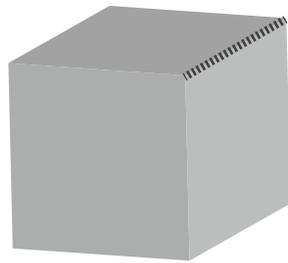
● Object



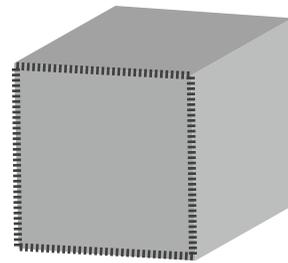
DUPLICATE

Duplicate function is implemented for a complete object. Line and face duplicate function is yet to be implemented.

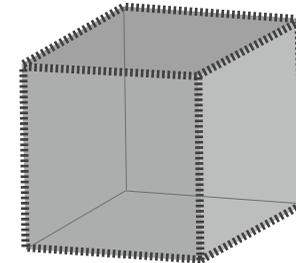
● Line



● Face



● Object



● GIZMOS

Gizmos has been implemented for transformation tools; move, scale and rotate. Transformations are either dependent on axis or not. There is universal transformation for move and rotate -- they can be used through a free transform control -- except scale.

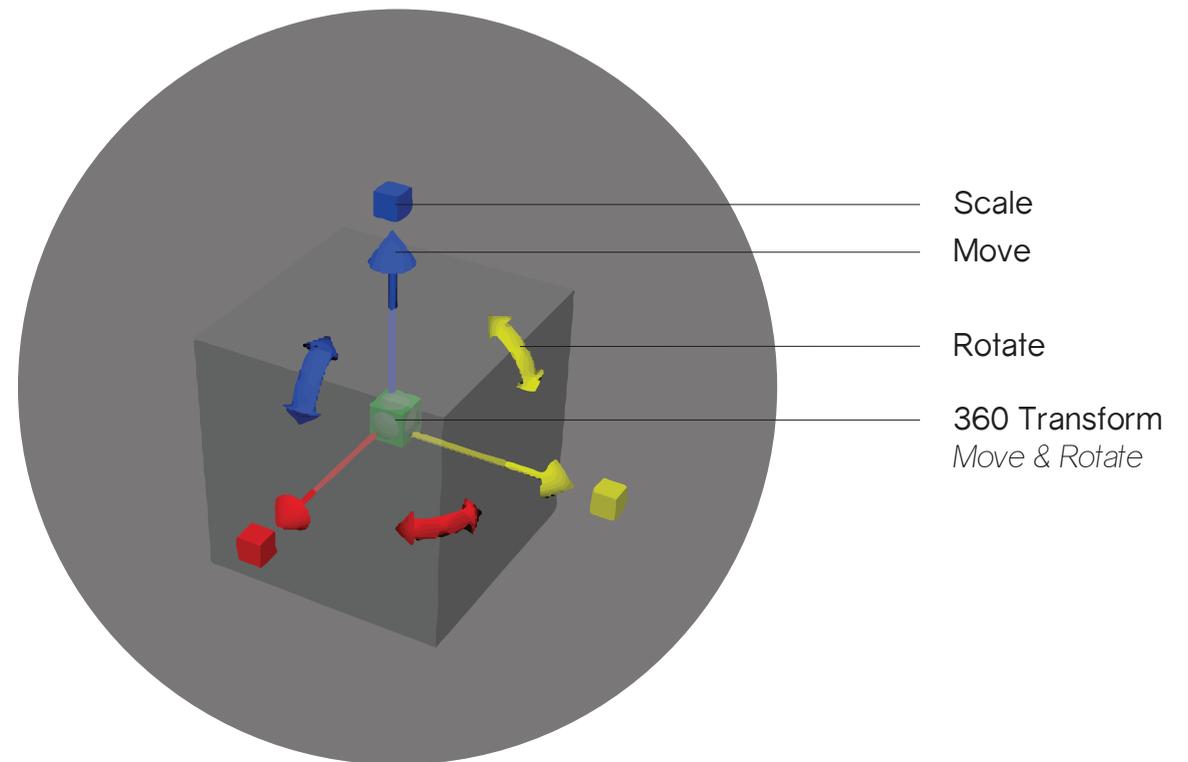
Gizmos concept remains the same but UI visual feedback design can change.

Functions

Applied to

Move	● ● ● ● ●
Rotate	● ● ● ● ●
Scale	● ● ● ● ●
Extrude	● ● ● ● ●

KEY ● Object ● Face ● Line ● Edge/Vertex



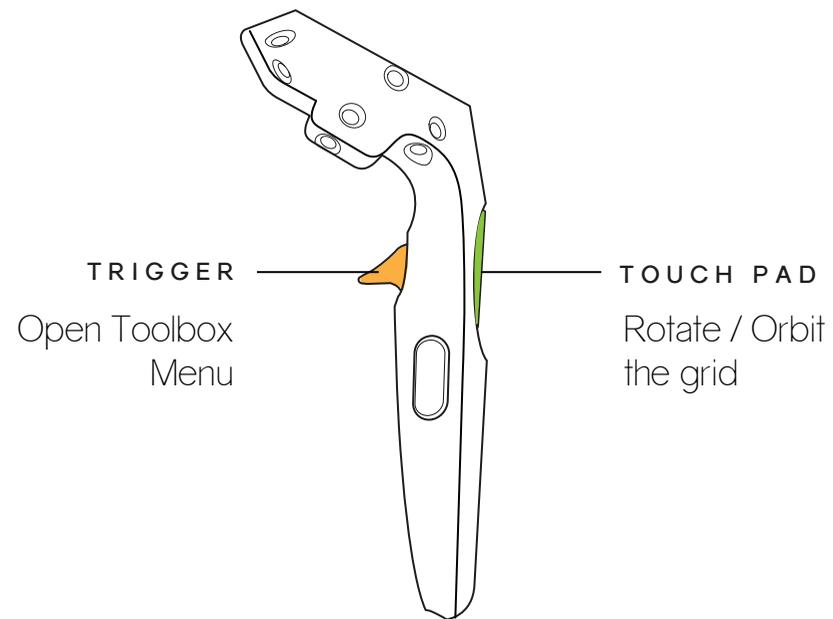
Scale
Move
Rotate
360 Transform
Move & Rotate

● CONTROLS

This is how HTV Vive Pre controller mapping for MVP functions is done for SketchVR. Mapping will change additional functionalities are implemented.

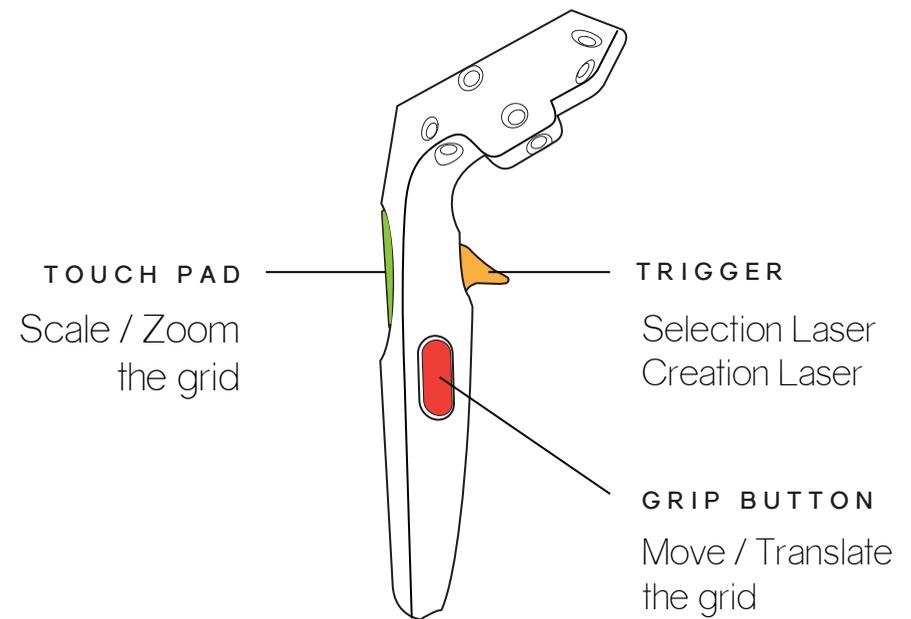
SECONDARY CONTROLLER

LEFT HAND



PRIMARY CONTROLLER

RIGHT HAND



TECHNICAL GUIDE

SHAPES

Shapes are what the users sees and creates. They are entities in space, with some arbitrary shape, which the user can create, destroy, and manipulate with the tools.

The shapes consists of three separate items (see diagram). There is [**Shape**], [**ShapeBehaviour**] and [**Selectable**]. The Shape holds all the data of a shape, i.e. all of the vertices, their indices and it's located in space. The [**ShapeBehaviour**] is a unity component, which takes the data from [**Shape**] and brings it onto the screen as visible geometry. In addition, [**Shape**] is serializable. It means that all shapes which inherit [**Shape**] can be saved and loaded.

Then there is the [**Selectable**], a unity component that in cooperation with [**Selection**], allows users to select different shapes, and work with them.

TOOLS

The tools are what the user interacts with in order to manipulate their creation. Whenever the user wants to move, destroy or create new geometry they use a tool. All tools define their own behavior, and therefore works very differently.

For example, the FreeMoveTool works by taking the user's selection and parenting it to the primary controller. This allows the shape to be freely and intuitively be moved around, as its movement is directly dependent on how the user is moving the controller.

The Translate tool works via a similar concept, creating an invisible dummy handle, parenting that to the controller, and then projecting its movement along the normal of whichever gizmos the user is pulling in.

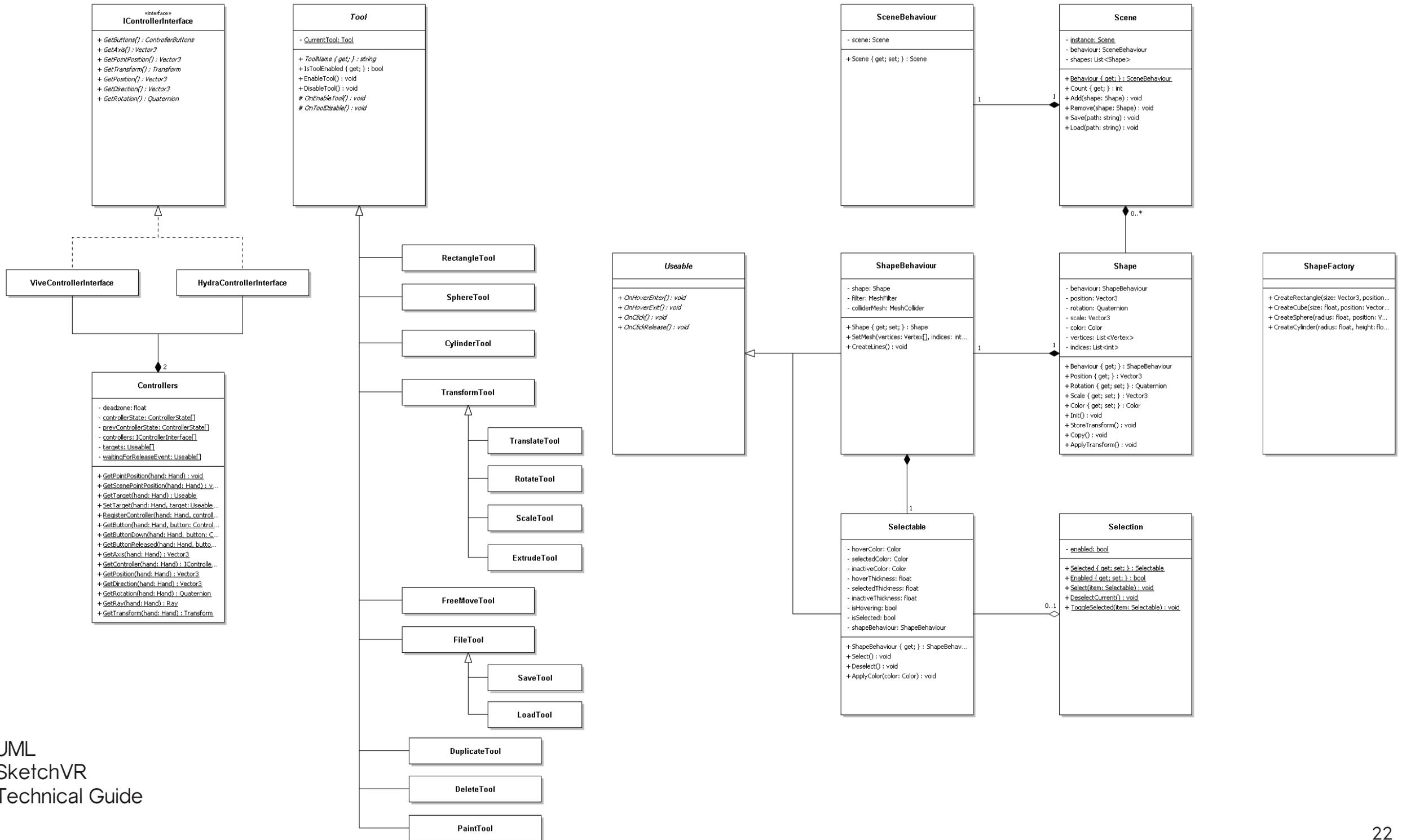
SCENE

The [**Scene**] is the collection of all the shapes the user has created and manipulated for their creation. It allows the user to rotate, zoom and move everything around, so they can get a better overview of their work. It also has the functionality to save and load your projects, so you can come back to them later.

USER INTERFACE

One of their biggest lack in the code of Sketch VR is some sort of common UI functionality. Much of the UI works in very similar manner, and due to lack of good common platform to develop on top of, there are a lot of abundant and repeating UI code across Sketch VR.

Creating a set of common UI functionality, for easy use and expansion would be ideal for continued development.



SketchVR

FUNCTIONALITIES
DESIGN

