Multitouch Extension Module for Google SketchUp

March 29

2009

An Extension Module that would allow Google SketchUp to receive TUIO and hence allow manipulations via Multitouch gestures.

NUI Group

Abstract

Lack of a CAD software that could be used via Multitouch has been the motivating factor behind this proposal. Eversince I used the Sketches application on iPhone –which enables 2D drawing using Multitouch, I felt like doing something similar on my FTIR setup. Adding multitouch ability to an easy to use CAD/modeling software - Google SketchUp would not only make the design process intuitive but also add a lot to the usability factor.

The reason behind choosing Google SketchUp and not start developing something from scratch is due to its easy to start UI. By experience I've felt that Google SketchUp is one of the easiest and widely used surface modeling tools around- what's impressive about it - its patented Push-Pull technique, being free to use and support for both Windows and OSX environments. Hence, the idea is simple - *to develop an extension with new gesture-sets for Google SketchUp that'd ease the modeling procedure via Multitouch.* An additional Bimanual Gesture support that'd allow an IR Pen and hand simultaneously has also been proposed as an add-on.

Previous Work

Autodesk is working on the next version of AutoCAD that would enable CAD using Multitouch. Although **AutoCAD** being a proprietary software shall remain closed source and expensive for most of the Multitouch community that works with DIY setups.

Introduction to Development

There is a Ruby TUIO Client implementation by Collin Harris to start up with. *Google SketchUp* comes with a powerful **Ruby API and a C++ SDK**. The API makes it possible to create macros, such as automated component generators and additional tools that could be included within the SketchUp menu. The plug-ins for Google SketchUp are written as Ruby scripts(.rb) that are interpreted by Google SketchUp. The aim is to create a compiled library that can talk to SketchUp via the Ruby API, and hence implement the OSC messages.

- From the Multitouch hardware Touchlib/Tbeta could be used to receive touch information and interpret it into events such as a click, drag, pinch, rotation, zoom etc.

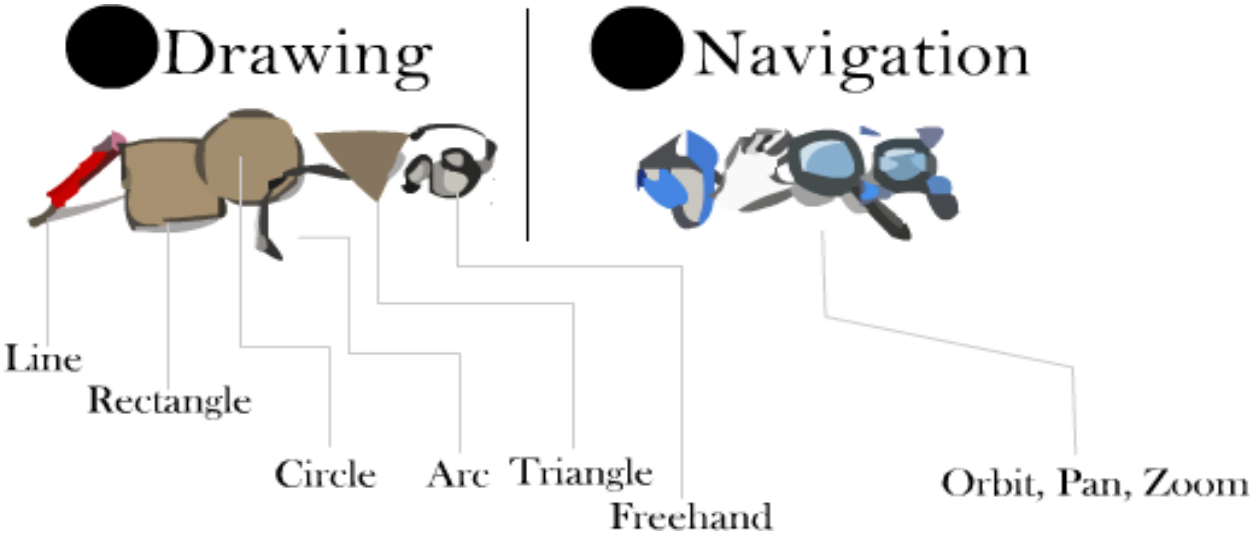
- TouchLib shall send the touch data to other applications by sending Tangible User Interface Object (TUIO) network messages- transmission of the state of gestures.

- TUIO Client Implementation in Ruby for parsing tuio network messages
  http://github.com/aberant/tuio-ruby/
  http://hans.fugal.net/src/rosc/doc/

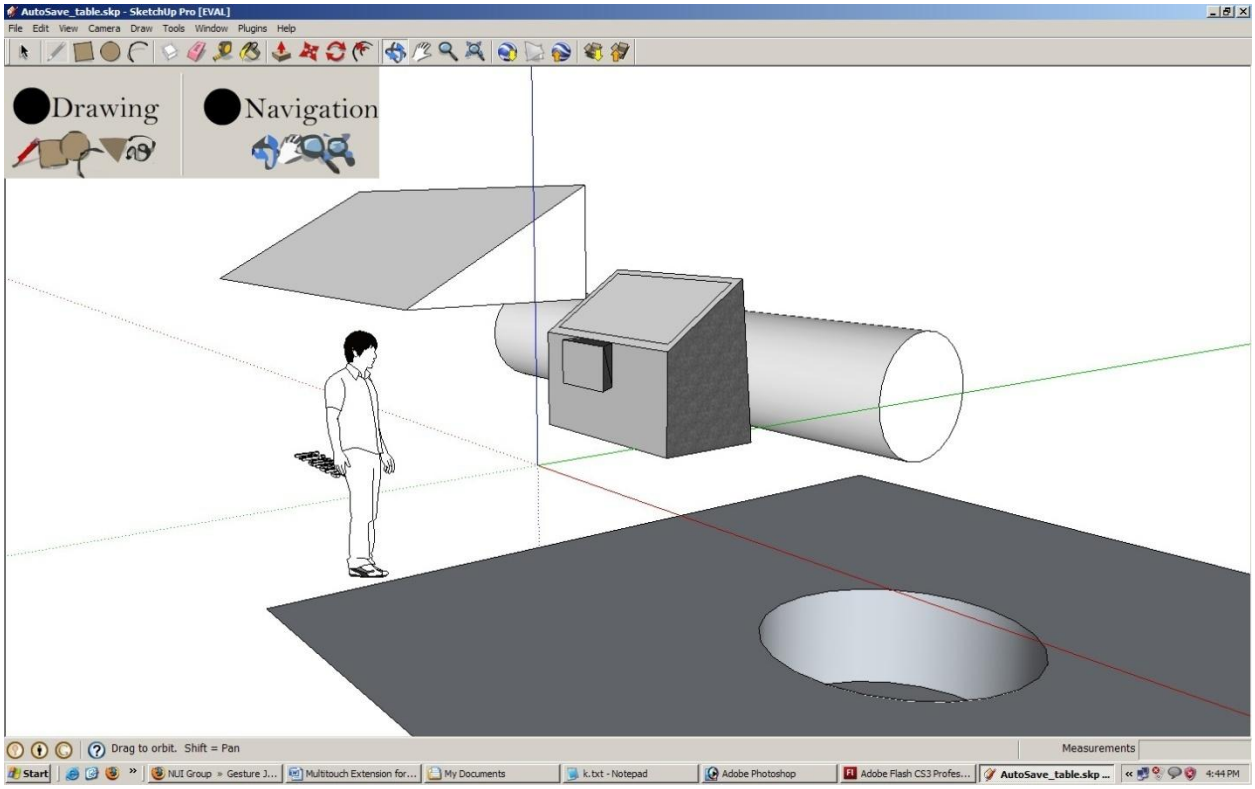
- Event Handling in SketchUp. Gestures mapped to specific tasks

The compiled library- that'll be the installable Multitouch Extension Module could be easily loaded/unloaded using the *Google SketchUp's Extension manager* (similar to NASA WorldWind Plugin by Laurence Muller)

The procedure would be to create ruby scripts with gestures defined for specific tasks that would map to two primary tasks –**Drawing and Workspace Navigation.**



The **Drawing Mode** would allow drawing via Multitouch. On selecting the sub-buttons the Module , and map the gestures to various functions in SketchUp.



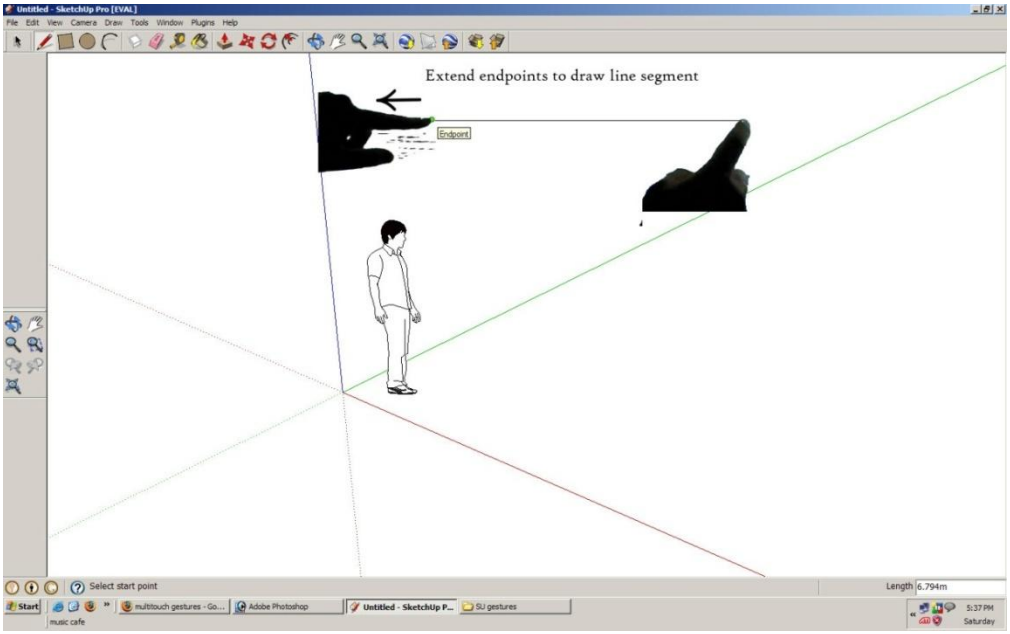
To keep the interface intuitive a new toolbar menu set will be created which shall contain **two sets**. This has a certain advantage over the traditional mouse system since it'd minimize the need to access the toolbar again and again for commonly used tasks:

TOUCH SET 1 - Camera Movements/Workspace Navigation- This includes Gestures to ZOOM-IN , ZOOM-OUT, PAN , AND ORBIT and taking measurements- without having the need to select any of the sub-options.

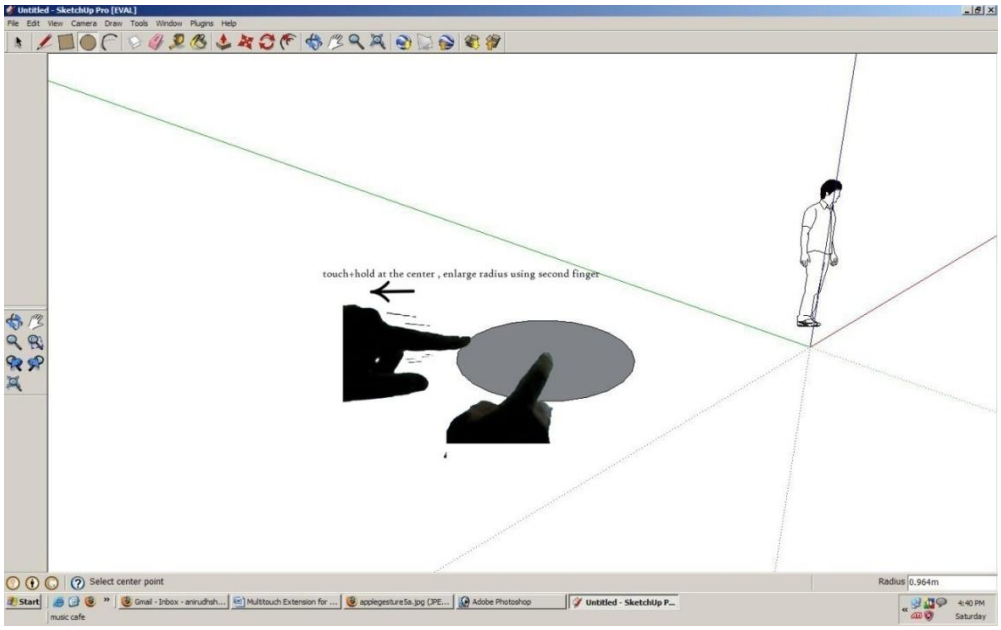
- Rotation of workspace/model in 3D – Orbit Tool
- Panning – Single touch drag
- Tapping initial and final point to make measurements

TOUCH SET 2 - Drawing- This shall enable drawing using gestures via SketchUp- The tentative plan for the gestures has been described in the text and diagrams below

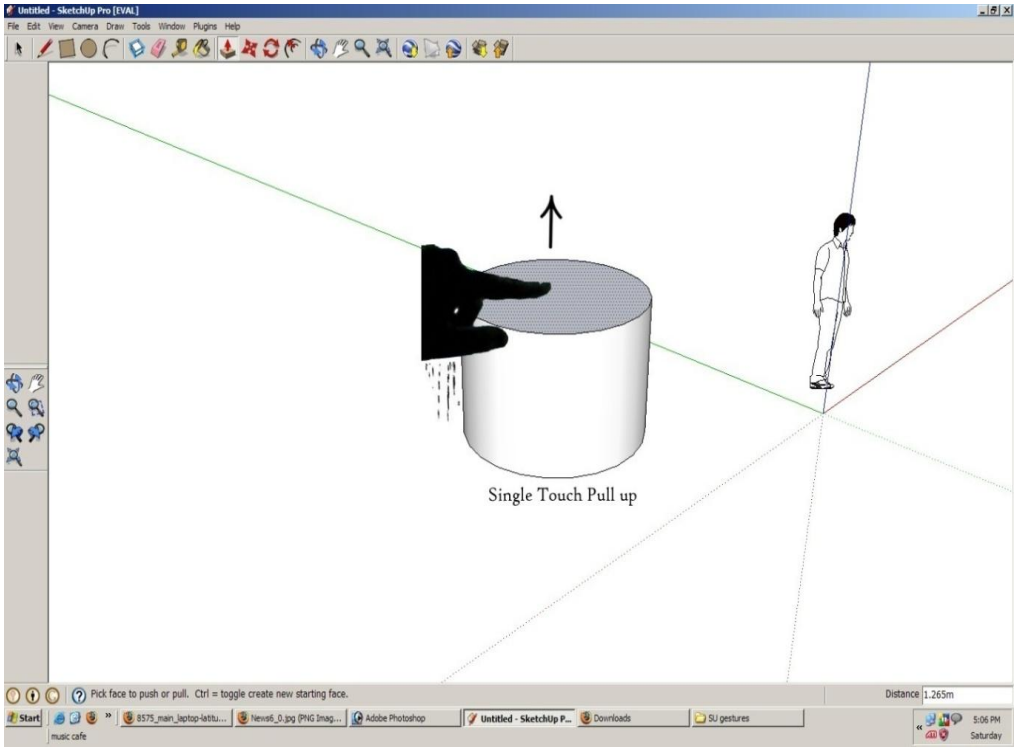
- Select **Square Option** and use four fingers to enlarge rectangle. SketchUp would take care of the approximations when user intends to draw a square. (rounding the measurements of the dimensions)- Default Feature-Could be Disabled.
- Select **Line** Option and tap **two points** on screen to create a line segment in between.
- Select **Circle** Option to draw circle using Multitouch by manipulating **radius** using MT



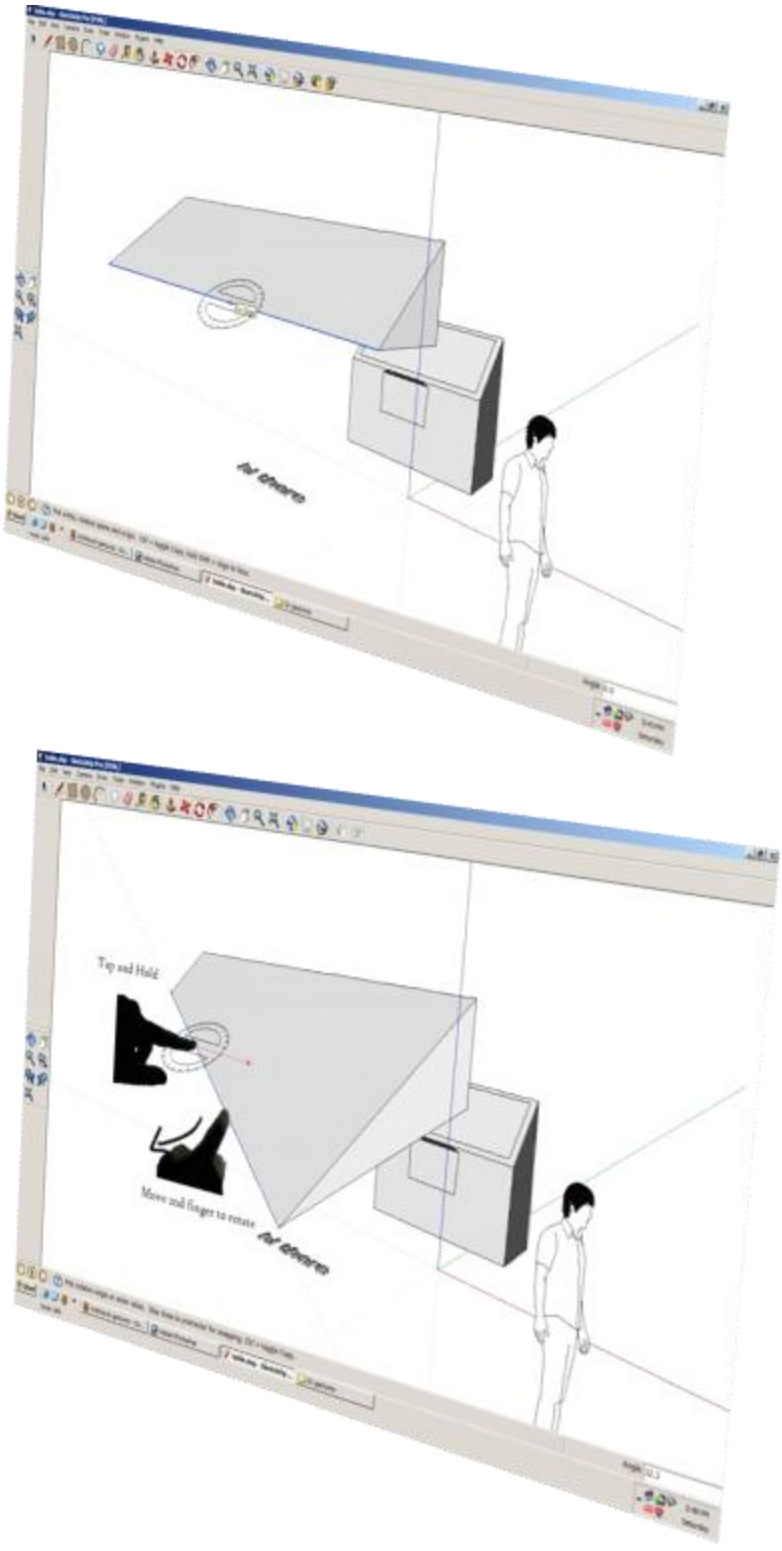
Drawing line using TWO FINGER Gesture (Hold one+ Drag other)



Drawing Circle using Multifinger gesture



Push and Pull up 2-D shape- single touch FACE drag



Complex manipulation – Rotatation- This has a fair advantage over the current traditional Single Touch method

It has to be seen how well one of the above gesture works together with other gestures. Via Experimentation and Testing it'll be figured out what gestures will be used together while general modeling, and then design them with similar features. – usually the chances are that while following a standard modeling procedure, ***scaling(zoom) and rotation would be done together***, so it'd be better to assign a gesture with same number of fingers to each operation and such.

Addon

After implementation of the above features, and community review, support could be added for Bimanual Gestures. Realizing the fact that the blob-size in case of a Pen would always be quite smaller as compared to that of a Hand, the pen and the Hand could be used simultaneously to **draw** and to bring up the **toolbar** menu respectively.

Ofcourse the gesture sets for this implementation would be different. The Pen Would work via Single Touch (Default Mode of Google SketchUp) and Hand gesture would populate a Menu system where the prime tools would populate around the hand. Interfaces like this have already been proposed in some Multitouch based image manipulation systems http://www.youtube.com/watch?v=JmHNr9EH1iU

Timeline

1st Month

- Further discuss the idea with mentors.
- Prepare Final tentative list of features/gesture implementations.
- Integrate TUIO client implementation of Ruby within the GoogleSU frame. Make SketchUp receive TUIO messages
- Study and implement the socket class Ruby has in its standard library http://www.ruby-doc.org/stdlib/libdoc/socket/rdoc/index.html
  http://hans.fugal.net/src/rosc/doc/

2nd Month

- First part of implementation: Google SU communicating with Single Touch using TUIO
- Finalize base UI button-set.
- Map proposed gestures to common tasks. Apply to Orbit, Zoom , Pan. Refer to **RotatableScalable** Implementation http://wiki.nuigroup.com/RotatableScalable
- **Testing**- test out on FTIR table, find shortcomings, go back to the design stage consult again and try something that'll works better.
- The last step is to document your results.

3rd Month

- Package Ruby Scripts and Client into a installable extension module –could be turned ON-OFF using SketchUp's MenuBar
- Improve upon the usability factor.
- Work on documentation.
- Test on Multitouch setup, Release packaged module to community.
- Fine tune the working related to Multitouching gestures. Reduce confusion.
- Debug , Reduce code complexity.
- ***If plausible implement BiManual Gesture Support as an addon. Using Pen + Hand Support to draw and bring up Toolbar***
- Testing
- Improve documentation, Release

Author

Name: Anirudh Sharma

Email: anirudhsharma.crypto@gmail.com

Timezone: +530 GMT

Blog URL: www.touchaddict.blogspot.com

Location: New Delhi, India

NUI Group Activities: Been a NUI Group member for more than a year now, joined on 19th Mar 08. Our college group will be organizing a Software + Hardware DIY camp "Open-source Multi-touching" during university's annual festival in Sept. 2009.

Forum Nick: ani **IRC nick:** anirudh

I am a 21 year old student from New Delhi, India pursuing Bachelor of Engineering degree in Informatics Technology at Govt. Engineering College Bikaner, Rajasthan Technical University. I have been involved with Open Source communities online for over three years. Last year's Google Summer of Code got me introduced to NUI Group, of which I've been a part since. Thereafter I made a FTIR table in college and began experimenting with real Multitouching and realized how it is going to revolutionize the way we interact with our computers.

At the age of 17 I won an interschool Flash Website Design competition that got further got me interested into UIs, and programming. During my curriculum I've taken prime courses on object-oriented programming in C++, Data Structures and Algorithms, Perl, UML, Computer Graphics, and Architecture. Apart I've been experimenting with Touchlib's AS3 Library apps by extending the premade apps.

I can program in AS3, C, C++, Ruby and PHP. Since I am already well-versed in the community interactions/dynamics of OSS projects, I will have absolutely no trouble in mingling with the NUI community and working with the infrastructure (Forums; Version Systems – I am familiar with CVS, SVN etc.) already in place. I have contributed to a Windows based GUI for manipulating files on the iPhone http://code.google.com/p/iphonebrowser/ . Being already familiar with the community coding could be started right away, which shall give me extra time in hand to try and implement the Bimanual Gesture Support feature, as proposed above.

Recently participation in some intercollegiate technical festivals, fetched my team First Prize at IIT Kanpur and BITS Pilani's technical festival, where I exhibited our table with NUI's apps and demonstrated how to make a Flash App from scratch.

References

- **When It Gets More Difficult, Use Both Hands** – Exploring Bimanual Curve Manipulation Russell Owen, Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, Bill Buxton http://www.billbuxton.com/Bimanual2005.pdf

- **Interaction Techniques for 3D Modeling on Large Displays** -Tovi Grossman, Ravin Balakrishnan, Gordon Kurtenbach, George Fitzmaurice, Azam Khan, Bill Buxton