1. How to install
    A. Team-AllyHyeseongKim/iphone-realdepth-streammer
        i. Install Xcode
        ii. Clone repo.
        iii. Open Project
        iv. Open:

            iphone-realdepth-streammer/TrueDepthStreamer/Sender/Common/Common.swift
        v. Change server address (BundelFusion or nodeJs server)
        vi. Connect to iPhone, Build & Run
    B. Team-AllyHyeseongKim/real-depth-streamer-server
        i. Install nodeJs
        ii. Clone repo
        iii. node index.js to test
    C. Team-AllyHyeseongKim/http-image-server-for-bundle-fusion-scanner
        i. Install VisualStudio
        ii. Install cpprest
        iii. Download http://kaldir.vc.in.tum.de/mLib/mLibExternal.zip
        iv. Download https://github.com/niessner/mLib
        v. File structure /c/scan $ ls BundleFusion-master/ mLib/ mLibExternal/
        vi. Check CustomSensor.cpp
            1. Change server address
            2. Change camera param
        vii. Run
        viii. if want to save press 9
        ix. if want to exit press esc

D. MeshLab

i. Install MeshLab

ii. Use Align function

iii. File-Export mesh as -> export obj file

E. Team-AllyHyeseongKim/unity-game

i. Install unity

ii. Open project

iii. Import obj file

iv. Make user & set property – server ip

F. Team-AllyHyeseongKim/rasp-socket-trigger-and-image-sender

i. Prepare rasp pi

ii. Install python

iii. Change server ip

iv. Trigger.py – python2

v. Client.py – python3

G. Team-AllyHyeseongKim/orb_slam2-for-socket-streaming

i. Prepare Ubuntu 16.04

ii. Check Examples/heebin/custom.cc and change server

iii. Check Examples/heebin/heebin.yaml

iv. apt-get install -y unzip sudo apt install -y git

v. mkdir OpenCV

vi. cd OpenCV

vii. wget https://gist.githubusercontent.com/eungbean/0880de7604472219c7e3f6ddb7cebde5/raw/443d629bf83b65bb59e34564626f872ab1124b3f/opencv-3.4.0-install-script.sh

viii. sudo bash opencv-3.4.0-install-script.sh

ix. cd ..

x. sudo apt-get install openjdk-8-jre openjdk-8-jdk

xi. sudo apt -y install g++ cmake cmake-gui doxygen mpi-default-dev openmpi-bin openmpi-common libusb-1.0-0-dev libqhull* libusb-dev libgtest-dev sudo apt -y install git-core freeglut3-dev pkg-config build-essential libxmu-dev libxi-dev libphonon-dev libphonon-dev phonon-backend-gstreamer sudo apt -y install phonon-backend-vlc graphviz mono-complete qt-sdk libflann-dev libflann1.8 libboost1.58-all-dev

xii. sudo apt install libgl1-mesa-dev libglew-dev cmake libpython2.7-dev ffmpeg libavcodec-dev libavutil-dev libavformat-dev libswscale-dev libavdevice-dev libdc1394-22-dev libraw1394-dev libjpeg-dev libpng12-dev libtiff5-dev libopenexr-dev libproj-dev

xiii. wget https://gitlab.com/libeigen/eigen/-/archive/3.3.8/eigen-3.3.8.tar.gz tar -xvf eigen-3.3.8.tar.gz

xiv. cd eigen-3.3.8/

xv. mkdir build

xvi. cd build cmake ../

xvii. make -j6 cd ..

xviii. Pangolin

xix. git clone https://github.com/stevenlovegrove/Pangolin.git

xx. cd Pangolin

xxi. mkdir build

xxii. cd build

xxiii. cmake ..

xxiv. cmake -build .

xxv. make -j32

xxvi. PCL&VTK
https://lsdtopotools.github.io/LSDTT_documentation/LSDTT_installation.html#_inst

all_the_point_cloud_library

xxvii. wget http://www.vtk.org/files/release/7.1/VTK-7.1.0.tar.gz

xxviii. tar -xf VTK-7.1.0.tar.gz

xxix. cd VTK-7.1.0 && mkdir build && cd build

xxx. cmake ..

xxxi. make

xxxii. sudo make install

xxxiii. Now install the Point Cloud Library v1.8 (use more than 15G ram,make without -j keyword(too heavy))

xxxiv. cd ~/Downloads

xxxv. wget https://github.com/PointCloudLibrary/pcl/archive/pcl-1.8.0.tar.gz

xxxvi. tar -xf pcl-1.8.0.tar.gz

xxxvii. cd pcl-pcl-1.8.0 && mkdir build && cd build

xxxviii. cmake ..

xxxix. make

xl. sudo make install

xli. git clone https://github.com/yanyan-li/Structure-SLAM-PointLine.git

xlii. cd Thirdparty/

xliii. cd DBoW2/

xliv. rm -rf build/

xlv. cd ..

xlvi. cd g2o/

xlvii. rm -rf build/

xlviii. cd ..

xlix. sudo ./build.sh

l. run Examples/heebin/custom 1, 2 .... As you want

2. run unity
3. run slam per gun
4. run rasp. Program

Play.