無人載具技術與應用
ROS Computer Vision

徐瑋隆

wlhsu304@gmail.com

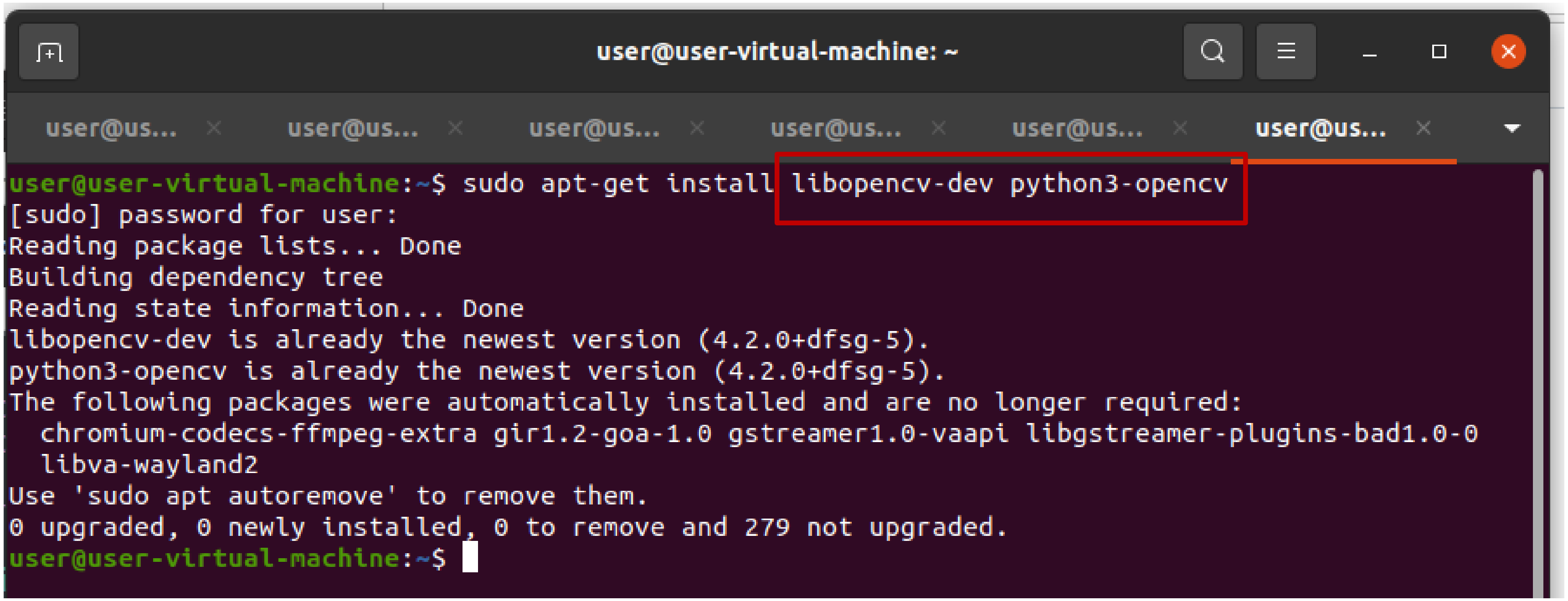
OpenCV

ROS COMPUTER VISION

OpenCV 01

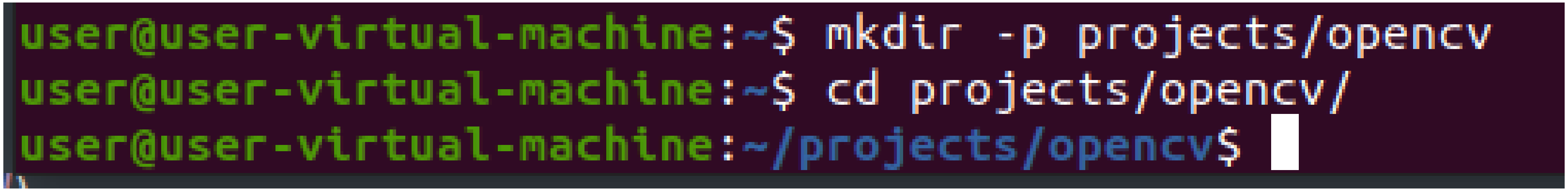
OpenCV -網路安裝

在有網路的環境下 用以下指令安裝
sudo apt install libopencv-dev python3-opencv



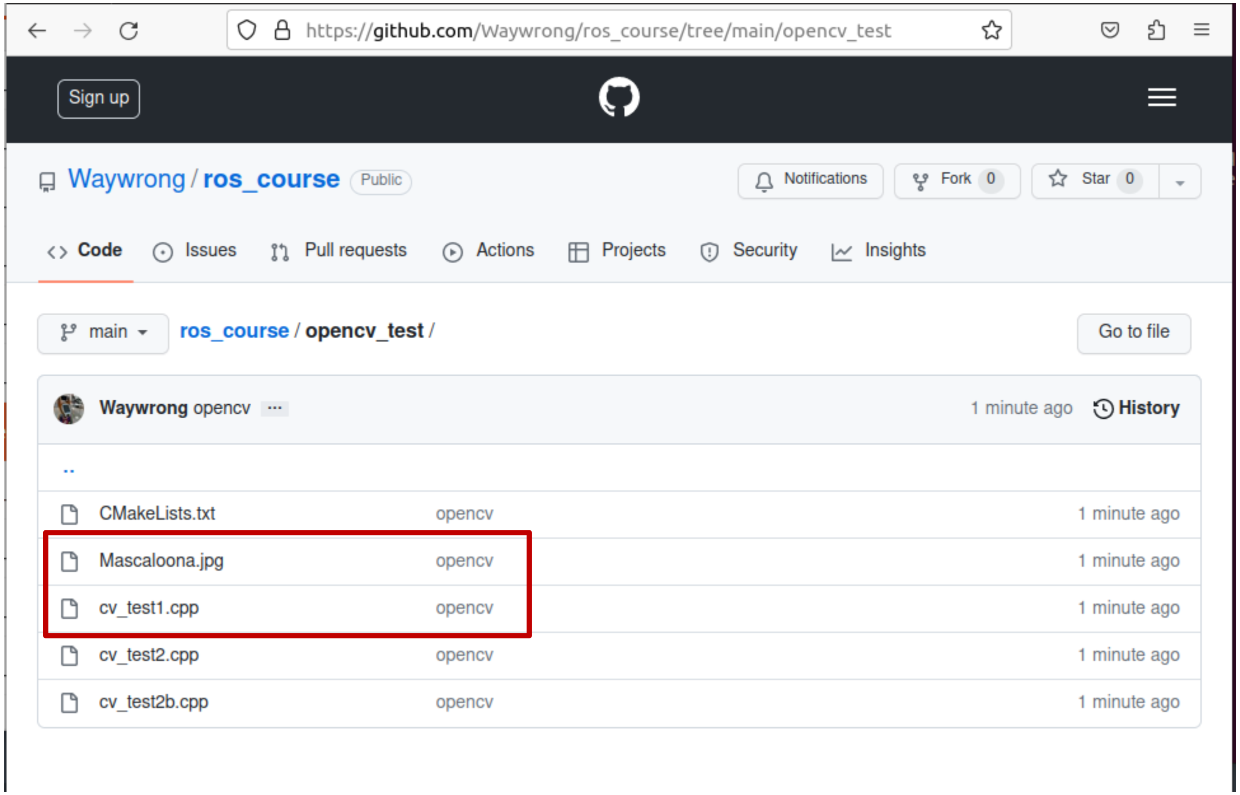
```
user@user-virtual-machine:~$ sudo apt-get install libopencv-dev python3-opencv
[sudo] password for user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
libopencv-dev is already the newest version (4.2.0+dfsg-5).
python3-opencv is already the newest version (4.2.0+dfsg-5).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gir1.2-goa-1.0 gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0
  libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 279 not upgraded.
user@user-virtual-machine:~$
```

OpenCV 01



```
user@user-virtual-machine:~$ mkdir -p projects/opencv
user@user-virtual-machine:~$ cd projects/opencv/
user@user-virtual-machine:~/projects/opencv$ 
```

mkdir -p projects/opencv



https://github.com/Waywrong/ros_course/tree/main/opencv_test

ros_course / opencv_test /

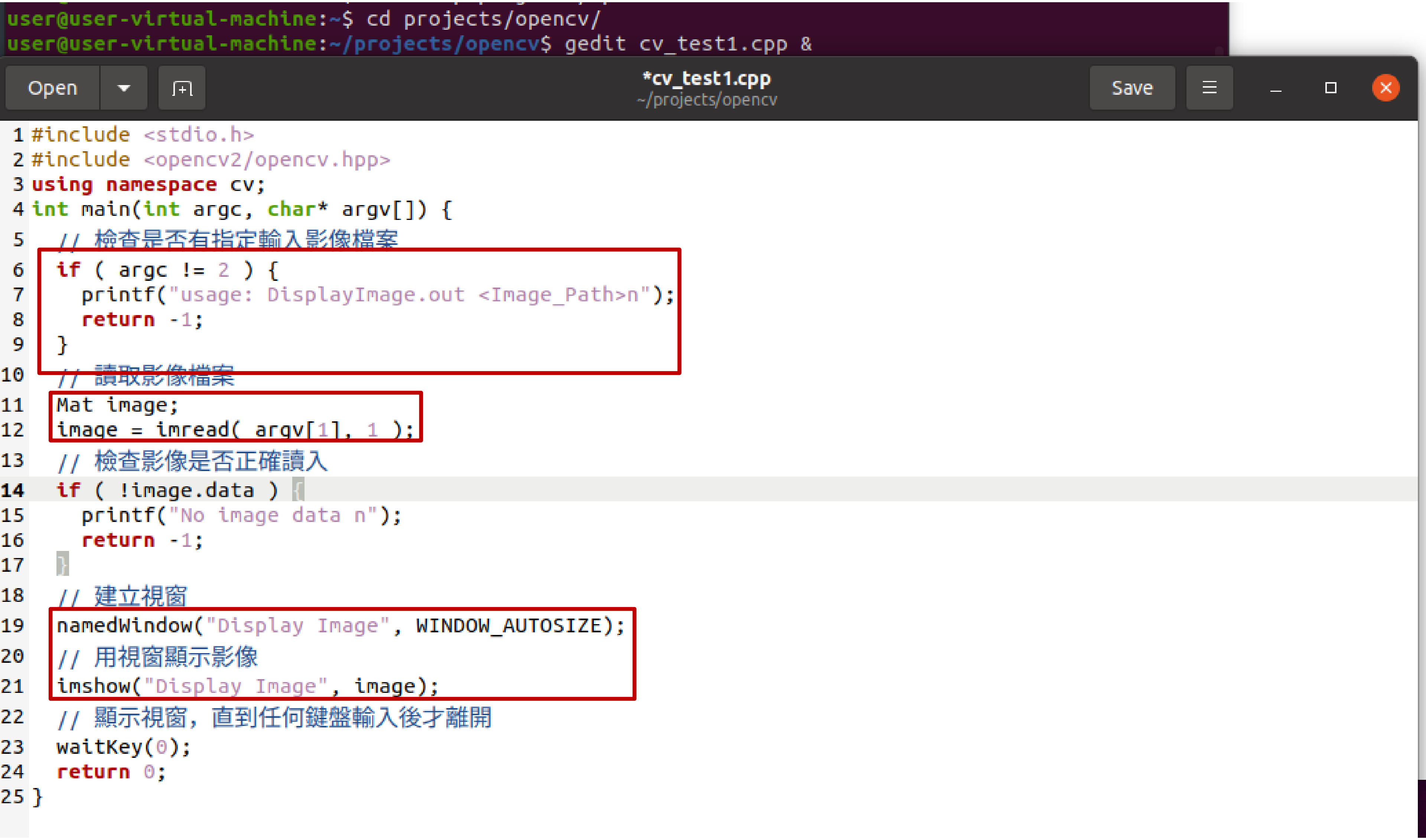
Waywrong opencv ··· 1 minute ago

..

CMakeLists.txt	opencv	1 minute ago
Mascaloona.jpg	opencv	1 minute ago
cv_test1.cpp	opencv	1 minute ago
cv_test2.cpp	opencv	1 minute ago
cv_test2b.cpp	opencv	1 minute ago

https://github.com/Waywrong/ros_course/tree/main/opencv_test

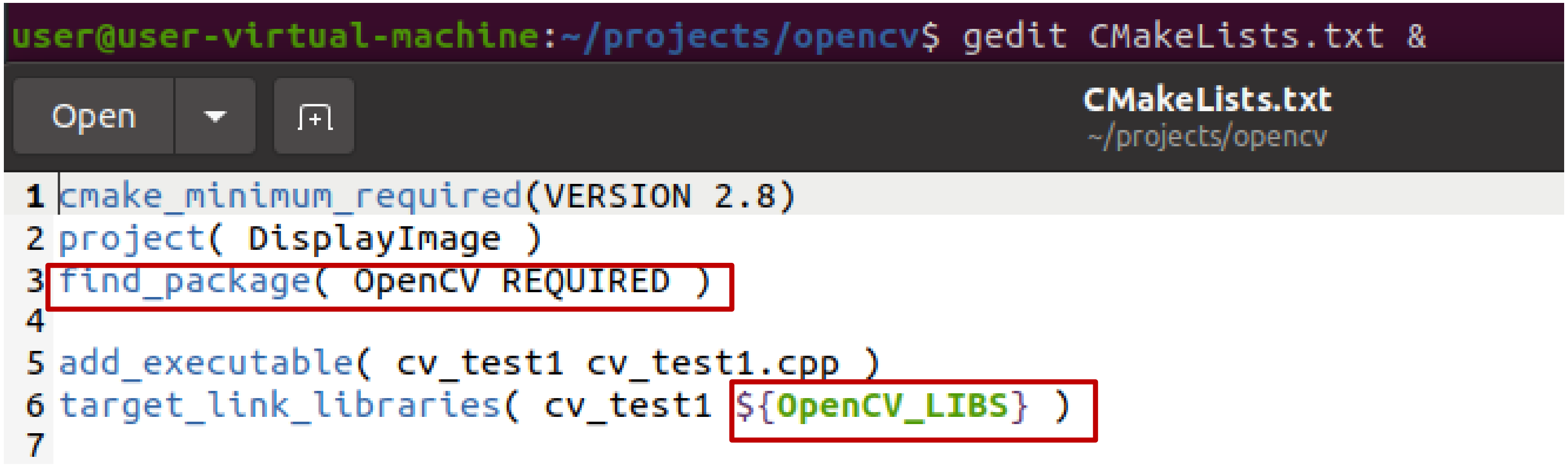
OpenCV 02



```
user@user-virtual-machine:~$ cd projects/opencv/
user@user-virtual-machine:~/projects/opencv$ gedit cv_test1.cpp &
```

```cpp
#include <stdio.h>
#include <opencv2/opencv.hpp>
using namespace cv;
int main(int argc, char* argv[]) {
  // 檢查是否有指定輸入影像檔案
  if ( argc != 2 ) {
    printf("usage: DisplayImage.out <Image_Path>n");
    return -1;
  }
  // 讀取影像檔案
  Mat image;
  image = imread( argv[1], 1 );
  // 檢查影像是否正確讀入
  if ( !image.data ) {
    printf("No image data n");
    return -1;
  }
  // 建立視窗
  namedWindow("Display Image", WINDOW_AUTOSIZE);
  // 用視窗顯示影像
  imshow("Display Image", image);
  // 顯示視窗，直到任何鍵盤輸入後才離開
  waitKey(0);
  return 0;
}
```

OpenCV 03



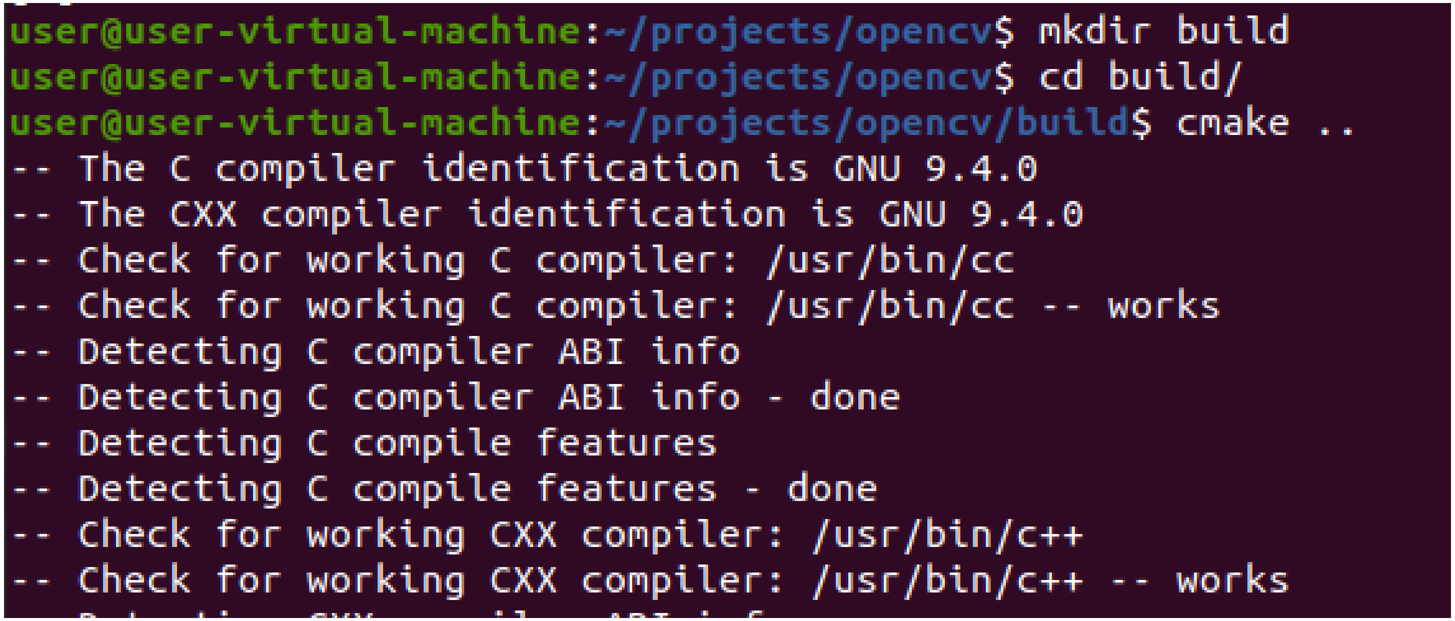
```
user@user-virtual-machine:~/projects/opencv$ gedit CMakeLists.txt &
```

```
cmake_minimum_required(VERSION 2.8)
project( DisplayImage )
find_package( OpenCV REQUIRED )

add_executable( cv_test1 cv_test1.cpp )
target_link_libraries( cv_test1 ${OpenCV_LIBS} )

```

```
cmake_minimum_required(VERSION 2.8)
project( DisplayImage )
find_package( OpenCV REQUIRED )

add_executable( cv_test1 cv_test1.cpp )
target_link_libraries( cv_test1 ${OpenCV_LIBS} )
```

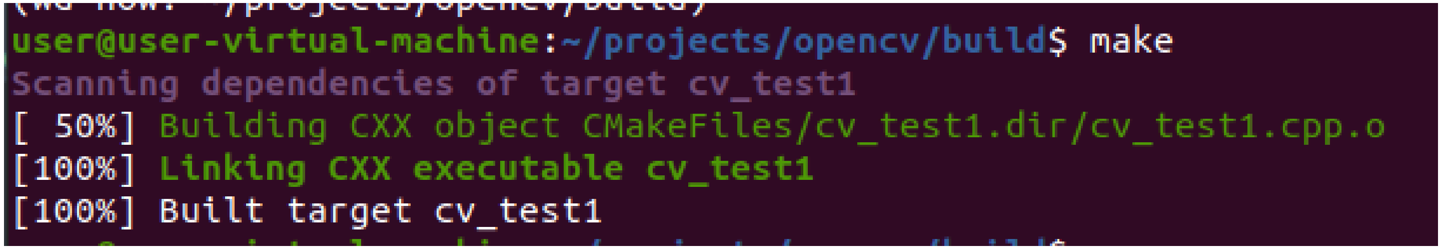
OpenCV 04

mkdir build
cd build/
cmake ..



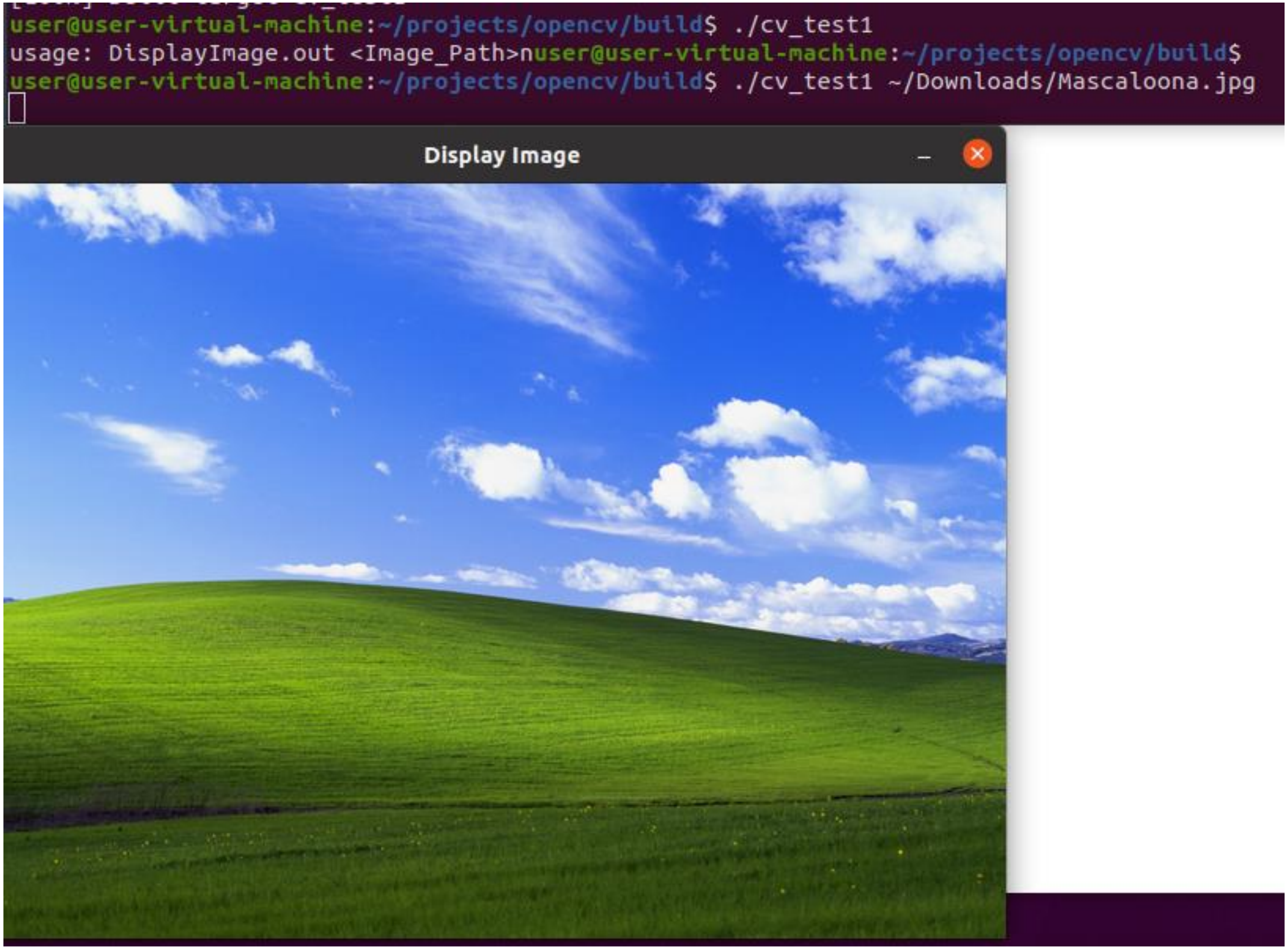
```
user@user-virtual-machine:~/projects/opencv$ mkdir build
user@user-virtual-machine:~/projects/opencv$ cd build/
user@user-virtual-machine:~/projects/opencv/build$ cmake ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
```

make



```
(wd now: ~/projects/opencv/build)
user@user-virtual-machine:~/projects/opencv/build$ make
Scanning dependencies of target cv_test1
[ 50%] Building CXX object CMakeFiles/cv_test1.dir/cv_test1.cpp.o
[100%] Linking CXX executable cv_test1
[100%] Built target cv_test1
user@user-virtual-machine:~/projects/opencv/build$
```

OpenCV 05

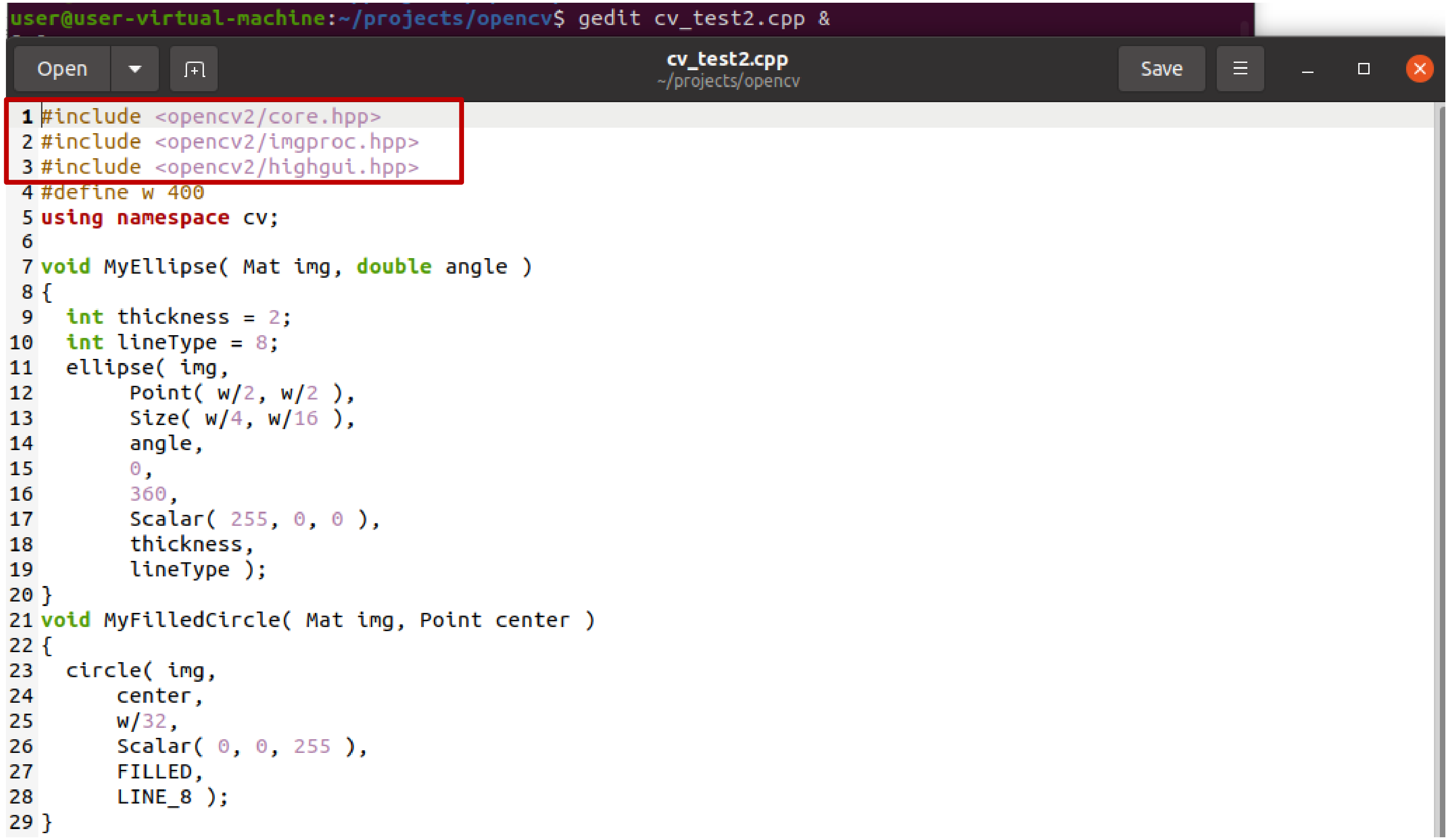


```
user@user-virtual-machine:~/projects/opencv/build$ ./cv_test1
usage: DisplayImage.out <Image_Path>nuser@user-virtual-machine:~/projects/opencv/build$
user@user-virtual-machine:~/projects/opencv/build$ ./cv_test1 ~/Downloads/Mascaloona.jpg
```

OpenCV II

ROS COMPUTER VISION

OpenCV 06



```
user@user-virtual-machine:~/projects/opencv$ gedit cv_test2.cpp &
```

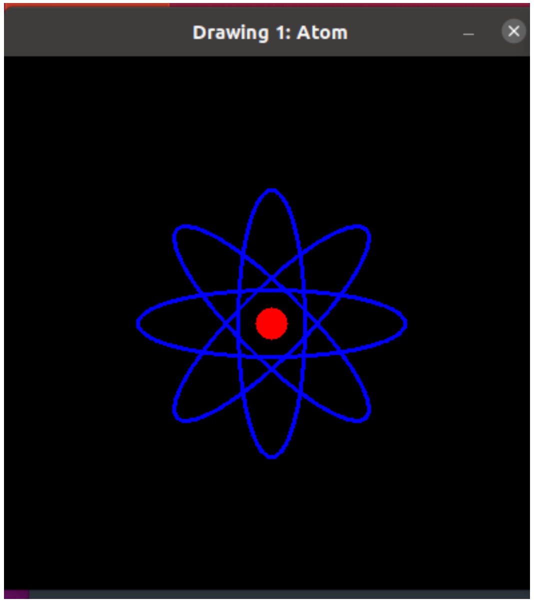
```cpp
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui.hpp>
#define w 400
using namespace cv;

void MyEllipse( Mat img, double angle )
{
  int thickness = 2;
  int lineType = 8;
  ellipse( img,
       Point( w/2, w/2 ),
       Size( w/4, w/16 ),
       angle,
       0,
       360,
       Scalar( 255, 0, 0 ),
       thickness,
       lineType );
}
void MyFilledCircle( Mat img, Point center )
{
  circle( img,
      center,
      w/32,
      Scalar( 0, 0, 255 ),
      FILLED,
      LINE_8 );
}
```

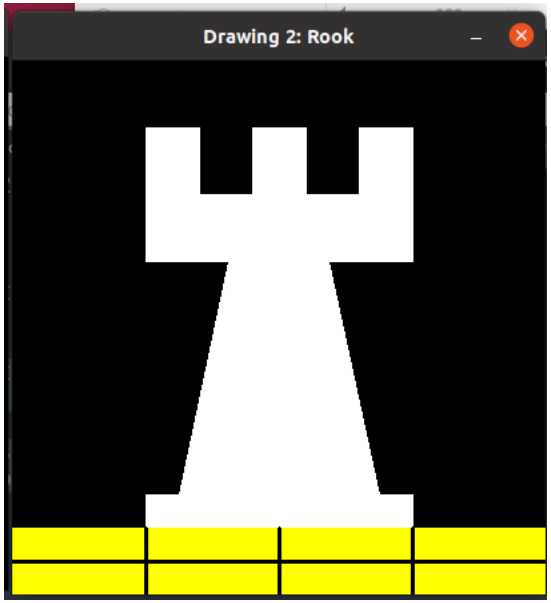
```
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui.hpp>
#define w 400
using namespace cv;
```

```
void MyEllipse( Mat img, double angle )
{
  int thickness = 2;
  int lineType = 8;
  ellipse( img,
       Point( w/2, w/2 ),
       Size( w/4, w/16 ),
       angle,
       0,
       360,
       Scalar( 255, 0, 0 ),
       thickness,
       lineType );
}
void MyFilledCircle( Mat img, Point center )
{
  circle( img,
      center,
      w/32,
      Scalar( 0, 0, 255 ),
      FILLED,
      LINE_8 );
}
```

```
void MyPolygon( Mat img )
{
  int lineType = LINE_8;
  Point rook_points[1][20];
  rook_points[0][0]  = Point(    w/4,   7*w/8 );
  rook_points[0][1]  = Point(  3*w/4,   7*w/8 );
  rook_points[0][2]  = Point(  3*w/4,  13*w/16 );
  rook_points[0][3]  = Point( 11*w/16, 13*w/16 );
  rook_points[0][4]  = Point( 19*w/32,  3*w/8 );
  rook_points[0][5]  = Point(  3*w/4,   3*w/8 );
  rook_points[0][6]  = Point(  3*w/4,     w/8 );
  rook_points[0][7]  = Point( 26*w/40,    w/8 );
  rook_points[0][8]  = Point( 26*w/40,    w/4 );
  rook_points[0][9]  = Point( 22*w/40,    w/4 );
  rook_points[0][10] = Point( 22*w/40,    w/8 );
  rook_points[0][11] = Point( 18*w/40,    w/8 );
  rook_points[0][12] = Point( 18*w/40,    w/4 );
  rook_points[0][13] = Point( 14*w/40,    w/4 );
  rook_points[0][14] = Point( 14*w/40,    w/8 );
  rook_points[0][15] = Point(    w/4,     w/8 );
  rook_points[0][16] = Point(    w/4,   3*w/8 );
  rook_points[0][17] = Point( 13*w/32,  3*w/8 );
  rook_points[0][18] = Point(  5*w/16, 13*w/16 );
  rook_points[0][19] = Point(    w/4,  13*w/16 );
  const Point* ppt[1] = { rook_points[0] };
  int npt[] = { 20 };
  fillPoly( img,
        ppt,
        npt,
        1,
        Scalar( 255, 255, 255 ),
        lineType );
}
void MyLine( Mat img, Point start, Point end )
{
  int thickness = 2;
  int lineType = LINE_8;
  line( img,
    start,
    end,
    Scalar( 0, 0, 0 ),
    thickness,
    lineType );
}
```

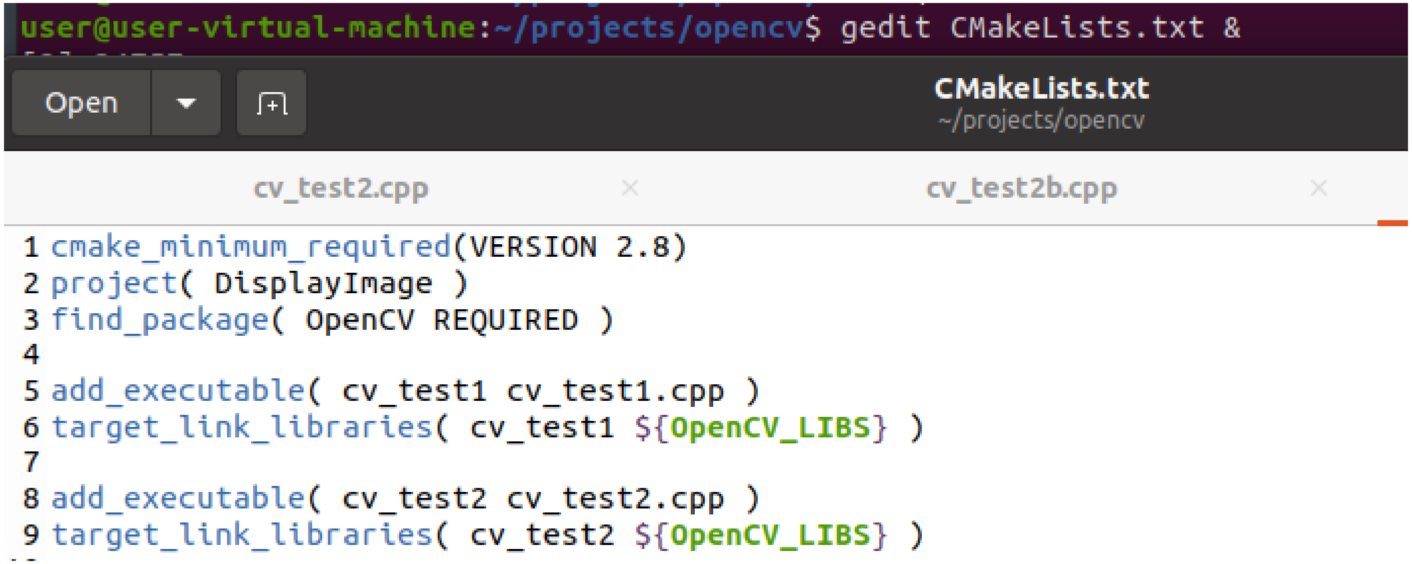
```
int main( void ){
 char atom_window[] = "Drawing 1: Atom";
 Mat atom_image = Mat::zeros( w, w, CV_8UC3 );
 MyEllipse( atom_image, 90 );
 MyEllipse( atom_image, 0 );
 MyEllipse( atom_image, 45 );
 MyEllipse( atom_image, -45 );
 MyFilledCircle( atom_image, Point( w/2, w/2) );
 imshow( atom_window, atom_image );
 moveWindow( atom_window, 0, 200 );
 waitKey( 0 );
 return(0);
}
```





```
int main( void ){
 char rook_window[] = "Drawing 2: Rook";
 Mat rook_image = Mat::zeros( w, w, CV_8UC3 );
 MyPolygon( rook_image );
 rectangle( rook_image,
       Point( 0, 7*w/8 ),
       Point( w, w),
       Scalar( 0, 255, 255 ),
       FILLED,
       LINE_8 );
 MyLine( rook_image, Point( 0, 15*w/16 ), Point( w, 15*w/16 ) );
 MyLine( rook_image, Point( w/4, 7*w/8 ), Point( w/4, w ) );
 MyLine( rook_image, Point( w/2, 7*w/8 ), Point( w/2, w ) );
 MyLine( rook_image, Point( 3*w/4, 7*w/8 ), Point( 3*w/4, w ) );
 imshow( rook_window, rook_image );
 moveWindow( rook_window, w, 200 );
 waitKey( 0 );
 return(0);
}
```

OpenCV 07

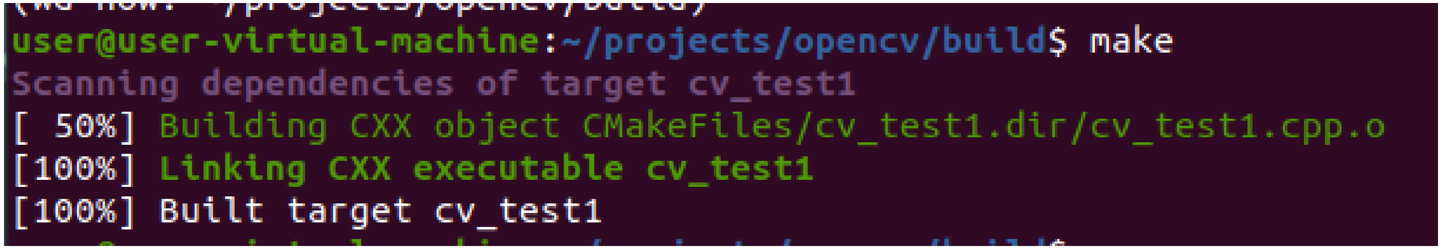


```
user@user-virtual-machine:~/projects/opencv$ gedit CMakeLists.txt &
```

```
cmake_minimum_required(VERSION 2.8)
project( DisplayImage )
find_package( OpenCV REQUIRED )

add_executable( cv_test1 cv_test1.cpp )
target_link_libraries( cv_test1 ${OpenCV_LIBS} )

add_executable( cv_test2 cv_test2.cpp )
target_link_libraries( cv_test2 ${OpenCV_LIBS} )
```

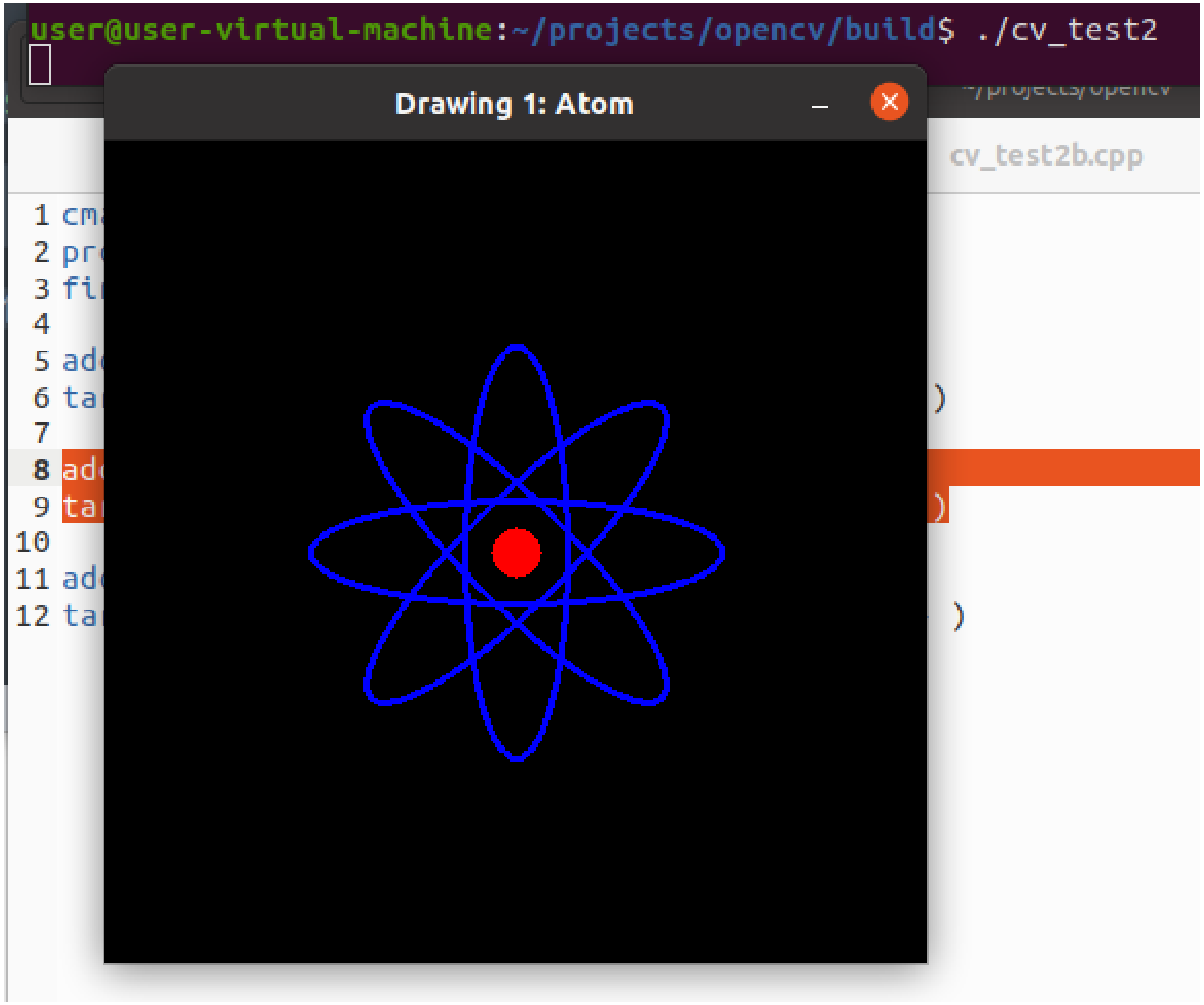
add_executable(cv_test2 cv_test2.cpp)
target_link_libraries(cv_test2 ${OpenCV_LIBS})

make



```
user@user-virtual-machine:~/projects/opencv/build$ make
Scanning dependencies of target cv_test1
[ 50%] Building CXX object CMakeFiles/cv_test1.dir/cv_test1.cpp.o
[100%] Linking CXX executable cv_test1
[100%] Built target cv_test1
```

OpenCV 08

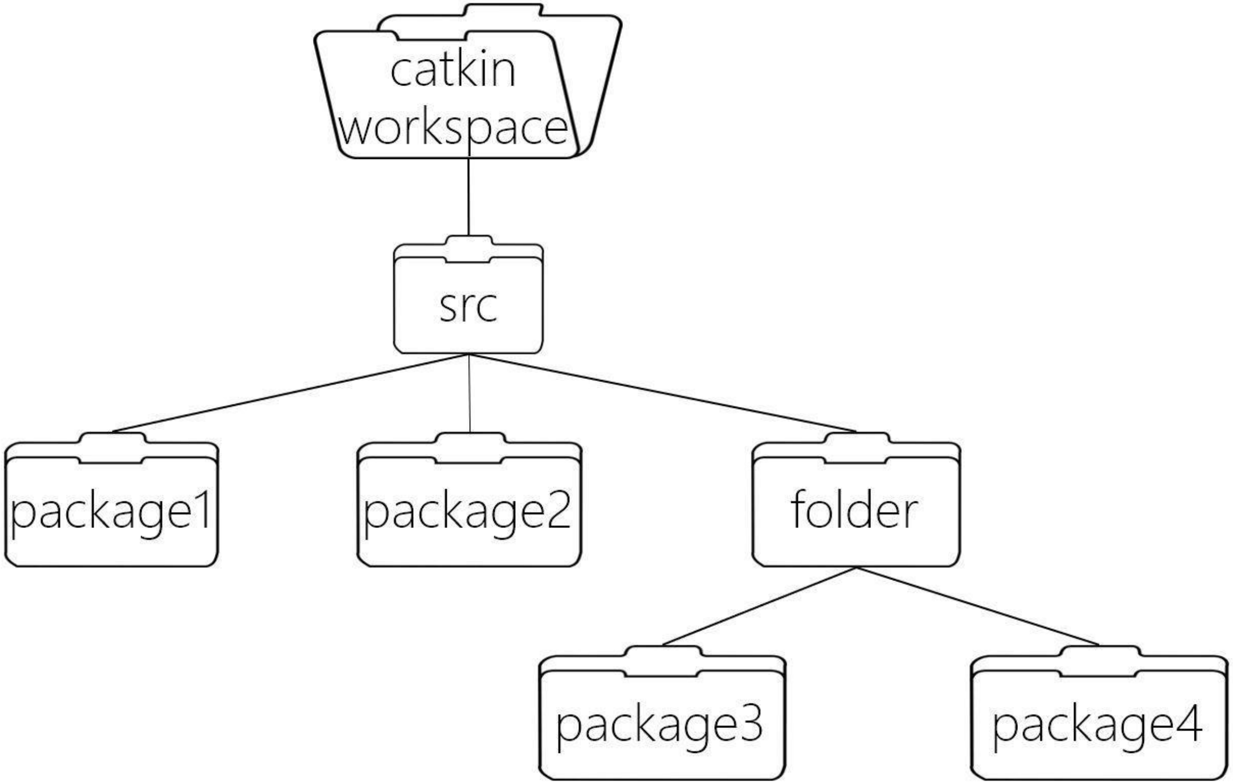


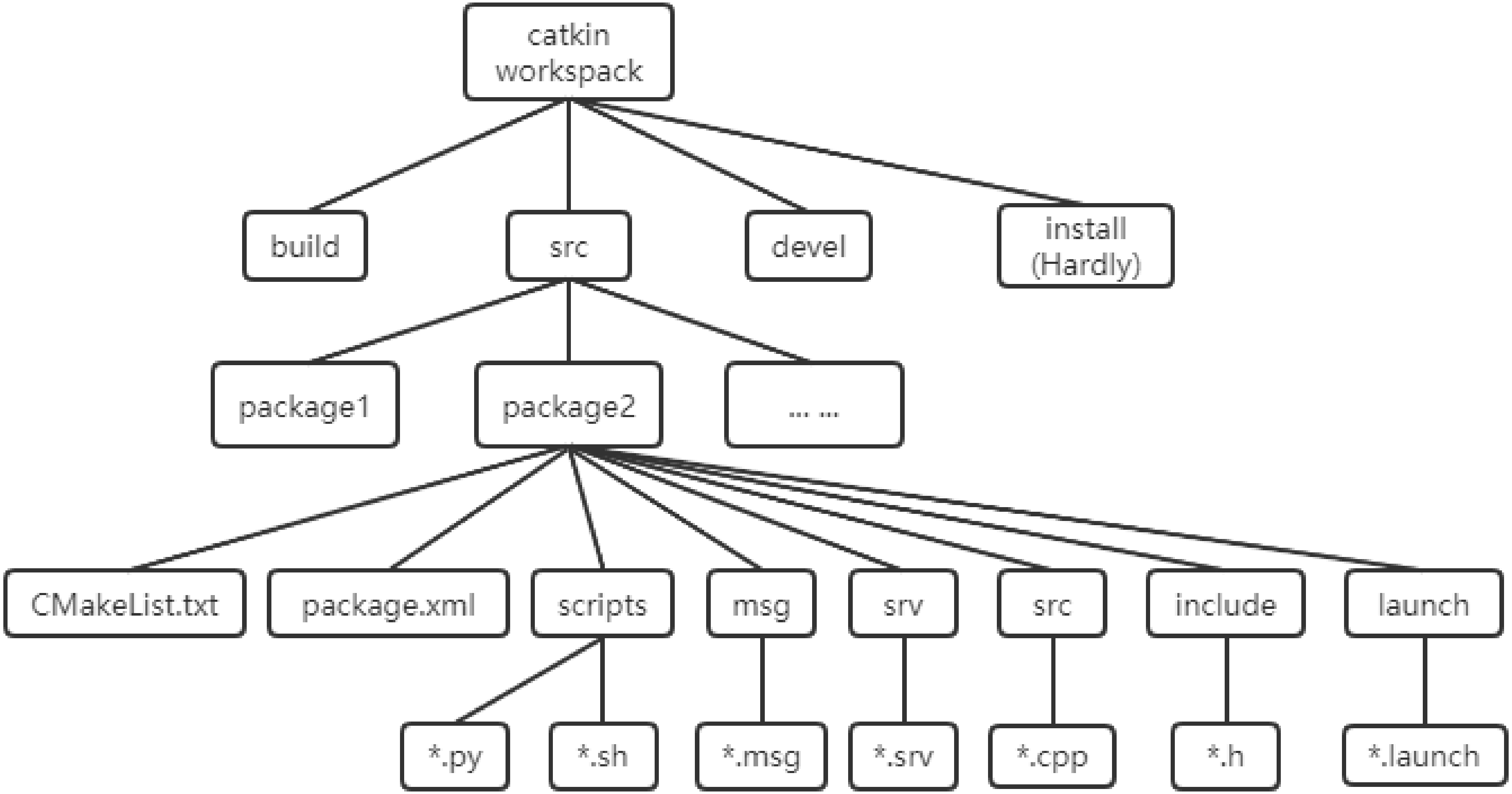
```
user@user-virtual-machine:~/projects/opencv/build$ ./cv_test2
```

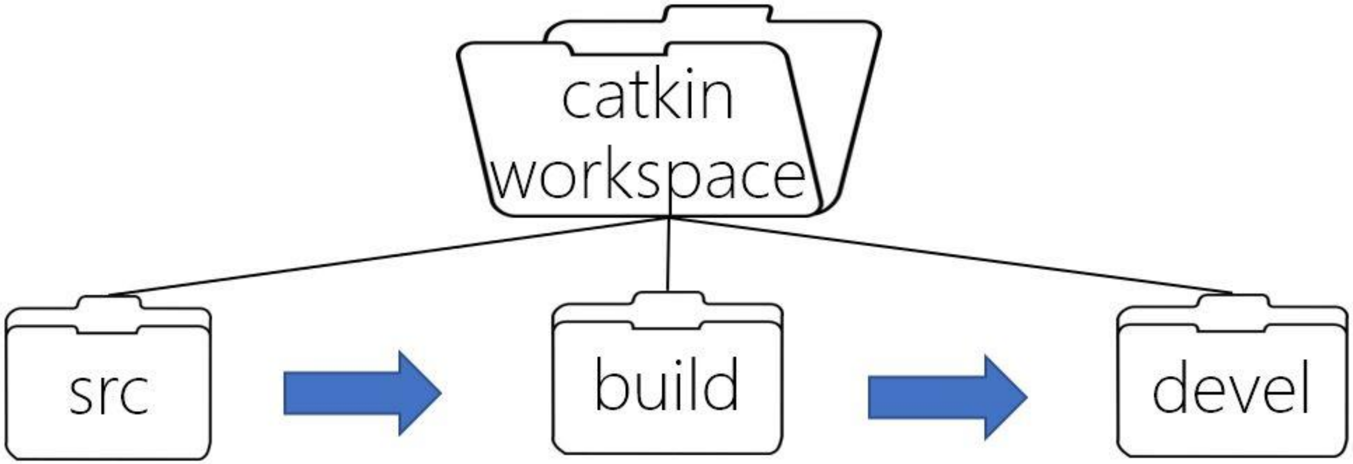
```
cma
pr
fi

ad
ta                                    )

ad
ta                                    )

ad
ta                                     )
```

ros node coding

ROS + OPENCV


catkin
workspace
src
package1
package2
folder
package3
package4


catkin workspack
build
src
devel
install (Hardly)
package1
package2
... ...
CMakeList.txt
package.xml
scripts
msg
srv
src
include
launch
*.py
*.sh
*.msg
*.srv
*.cpp
*.h
*.launch



catkin workspace

src

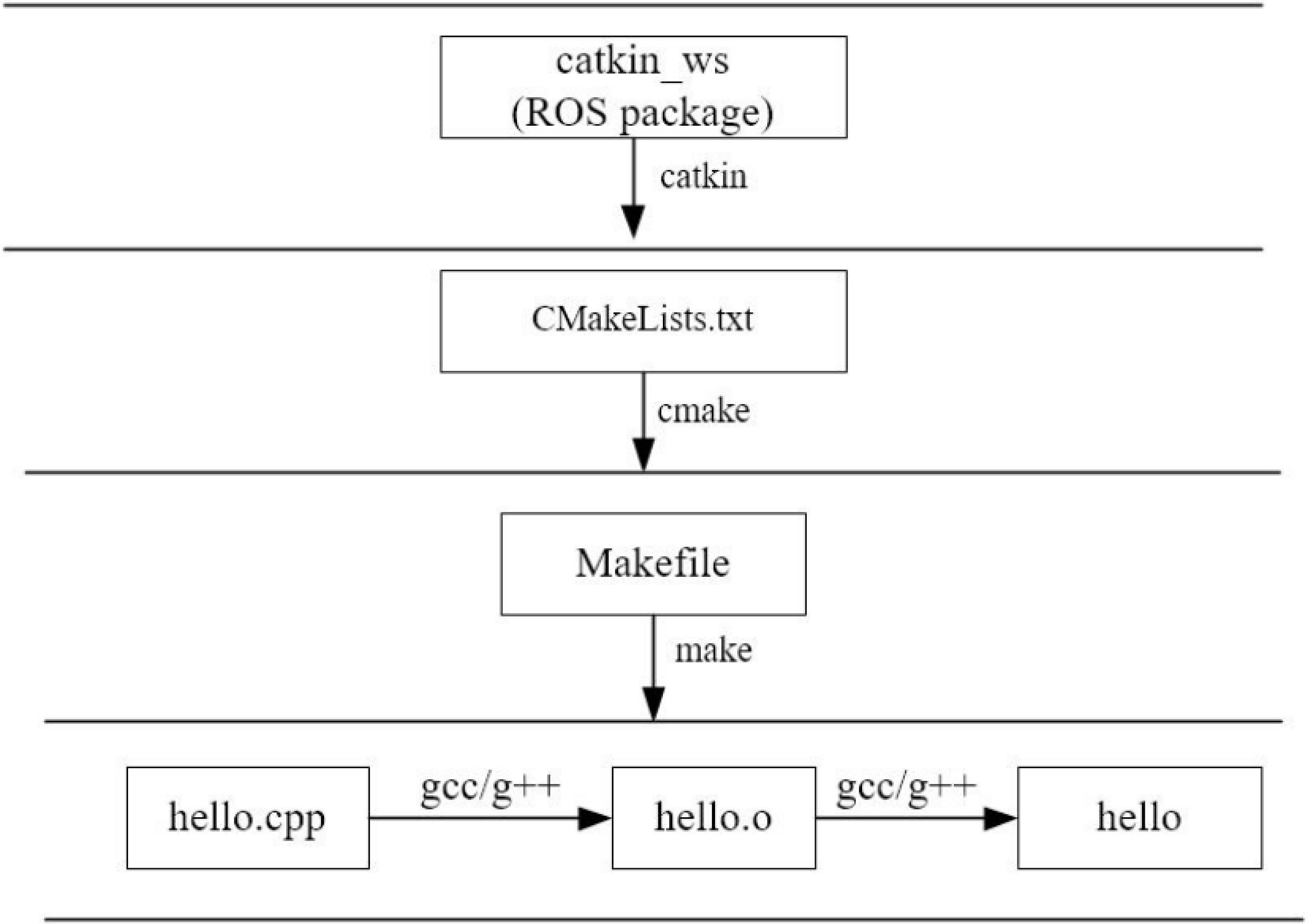
build

devel

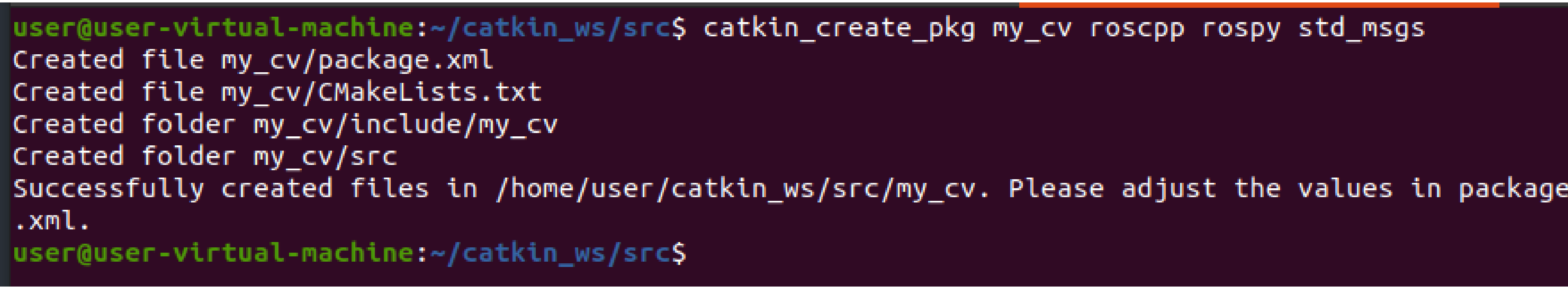
package源代码包

cmake&catkin缓存和中间文件

目标文件


catkin_ws
(ROS package)
catkin
CMakeLists.txt
cmake
Makefile
make
hello.cpp
gcc/g++
hello.o
gcc/g++
hello

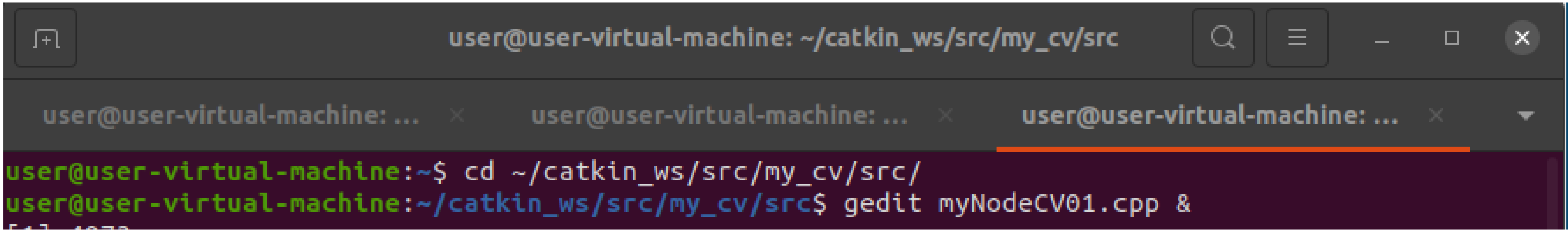
ros node 01



```
user@user-virtual-machine:~/catkin_ws/src$ catkin_create_pkg my_cv roscpp rospy std_msgs
Created file my_cv/package.xml
Created file my_cv/CMakeLists.txt
Created folder my_cv/include/my_cv
Created folder my_cv/src
Successfully created files in /home/user/catkin_ws/src/my_cv. Please adjust the values in package
.xml.
user@user-virtual-machine:~/catkin_ws/src$
```

cd catkin_ws/src/
catkin_create_pkg my_cv roscpp rospy std_msgs

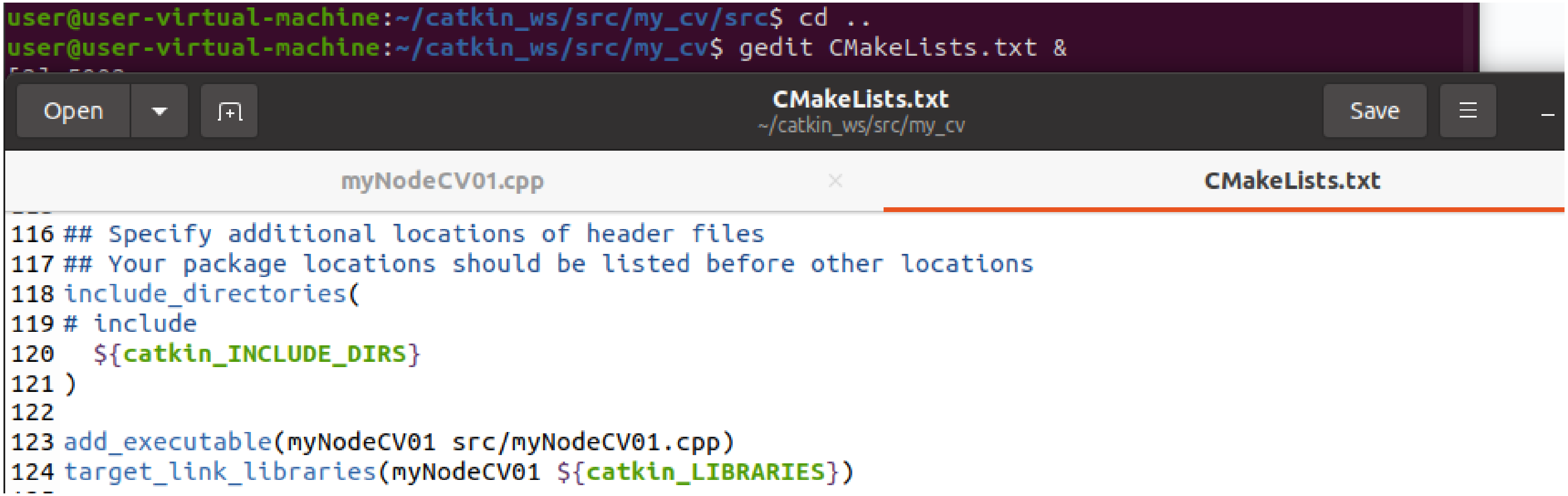
ros node 02


user@user-virtual-machine: ~/catkin_ws/src/my_cv/src
user@user-virtual-machine: ...
user@user-virtual-machine: ...
user@user-virtual-machine: ...

```
user@user-virtual-machine:~$ cd ~/catkin_ws/src/my_cv/src/
user@user-virtual-machine:~/catkin_ws/src/my_cv/src$ gedit myNodeCV01.cpp &
```

```
#include "ros/ros.h"

int main(int argc, char **argv)
{
  ros::init(argc, argv, "myNodeCV01");
  ros::NodeHandle n;

  ROS_INFO("myNodeCV01: hi");
  ros::Rate r(30);
  while(ros::ok())
  {
    r.sleep();
    ros::spinOnce();
  }
  return 0;
}
```

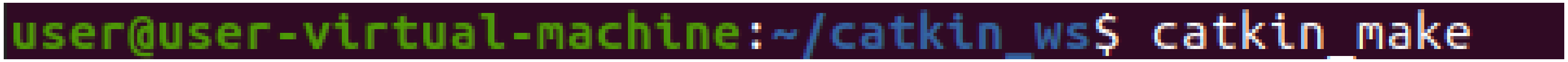
ros node 03



```
user@user-virtual-machine:~/catkin_ws/src/my_cv/src$ cd ..
user@user-virtual-machine:~/catkin_ws/src/my_cv$ gedit CMakeLists.txt &
```

```
## Specify additional locations of header files
## Your package locations should be listed before other locations
include_directories(
# include
  ${catkin_INCLUDE_DIRS}
)

add_executable(myNodeCV01 src/myNodeCV01.cpp)
target_link_libraries(myNodeCV01 ${catkin_LIBRARIES})
```

```
add_executable(myNodeCV01 src/myNodeCV01.cpp)
target_link_libraries(myNodeCV01 ${catkin_LIBRARIES})
```

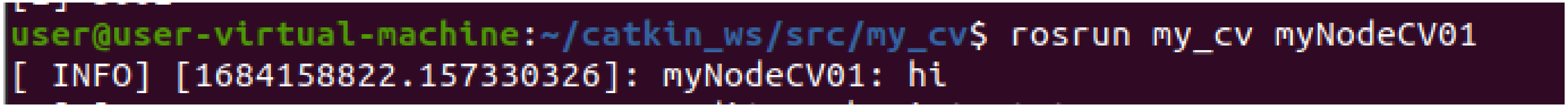


```
user@user-virtual-machine:~/catkin_ws$ catkin_make
```

```
cd ~/catkin_ws
catkin_make
```

編譯程式

ros node 04



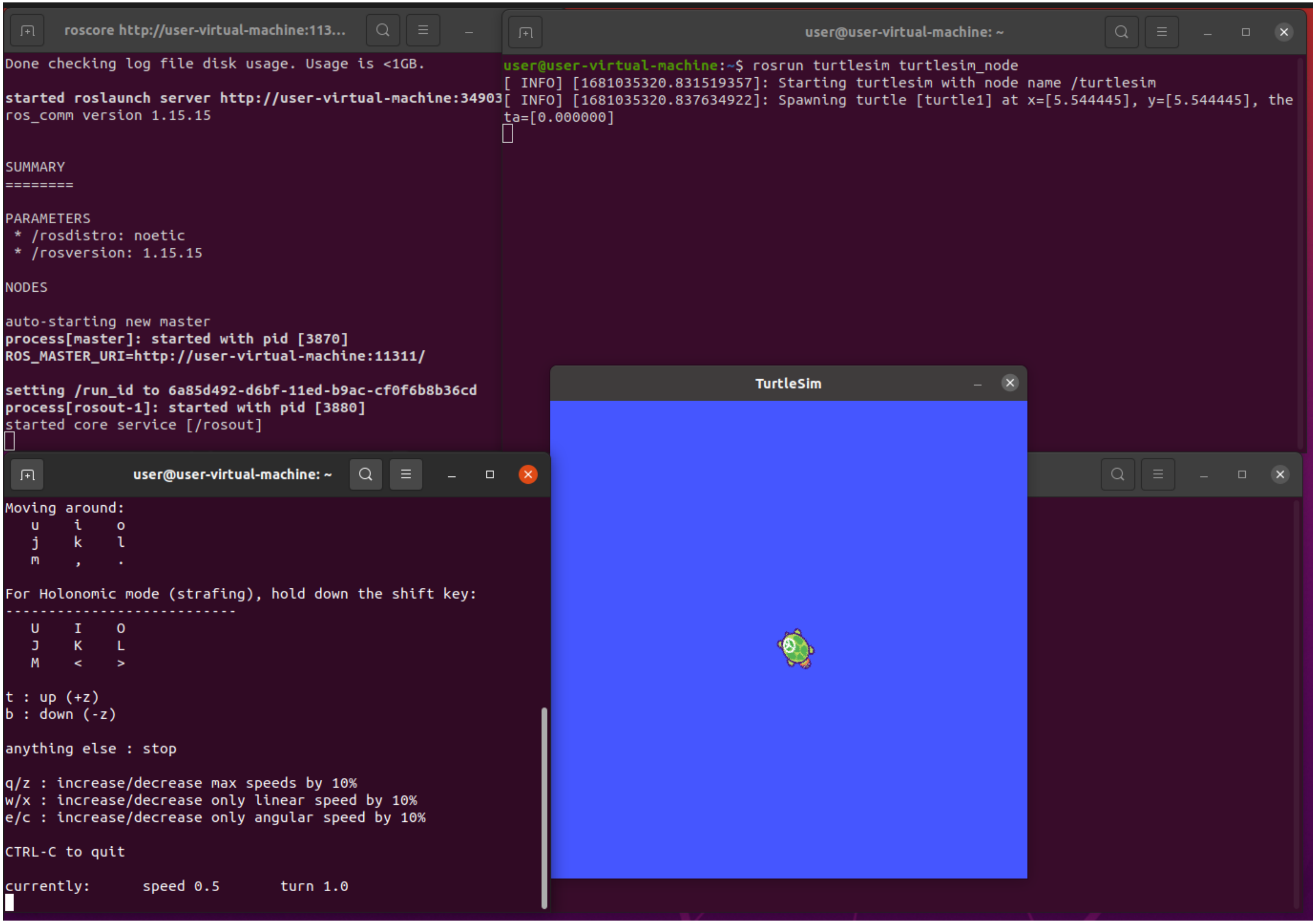
```
user@user-virtual-machine:~/catkin_ws/src/my_cv$ rosrun my_cv myNodeCV01
[ INFO] [1684158822.157330326]: myNodeCV01: hi
```

roscore
rosrun my_cv myNodeCV01

ros node + opencv

ROS + OPENCV

ros cv 01


roscore http://user-virtual-machine:113...
user@user-virtual-machine: ~
user@user-virtual-machine: ~
TurtleSim

```
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://user-virtual-machine:34903
ros_comm version 1.15.15


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.15

NODES

auto-starting new master
process[master]: started with pid [3870]
ROS_MASTER_URI=http://user-virtual-machine:11311/

setting /run_id to 6a85d492-d6bf-11ed-b9ac-cf0f6b8b36cd
process[rosout-1]: started with pid [3880]
started core service [/rosout]
```

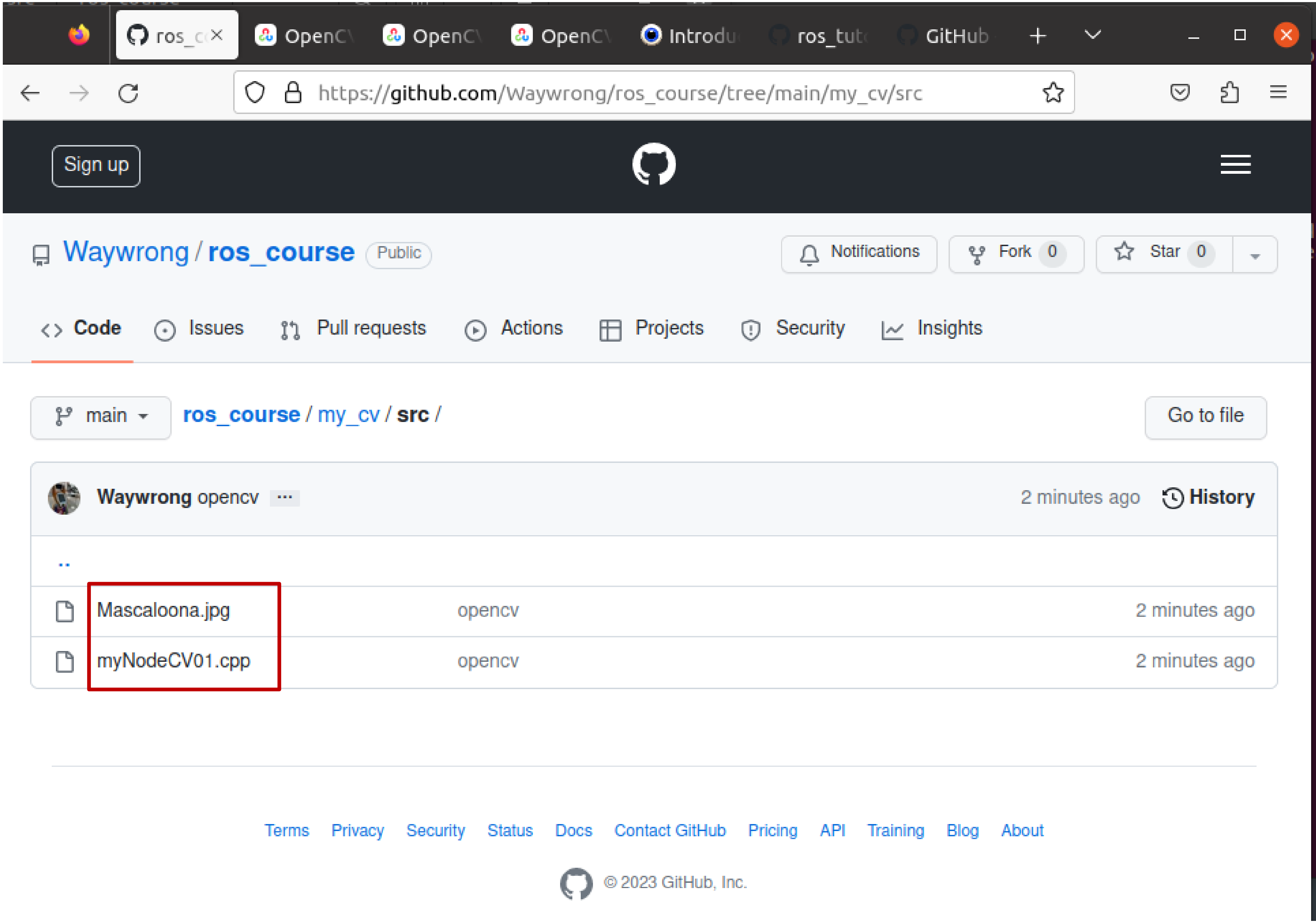
```
user@user-virtual-machine:~$ rosrun turtlesim turtlesim_node
[ INFO] [1681035320.831519357]: Starting turtlesim with node name /turtlesim
[ INFO] [1681035320.837634922]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], the
ta=[0.000000]
```

```
Moving around:
   u    i    o
   j    k    l
   m    ,    .

For Holonomic mode (strafing), hold down the shift key:
---------------------------
   U    I    O
   J    K    L
   M    <    >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:      speed 0.5       turn 1.0
```

roscore

rosrun turtlesim turtlesim_node

rosrun teleop_twist_keyboard teleop_twist_keyboard.py /cmd_vel:=/turtle1/cmd_vel



https://github.com/Waywrong/ros_course/tree/main/my_cv/src

Waywrong / ros_course Public

ros_course / my_cv / src /

Waywrong opencv ··· 2 minutes ago

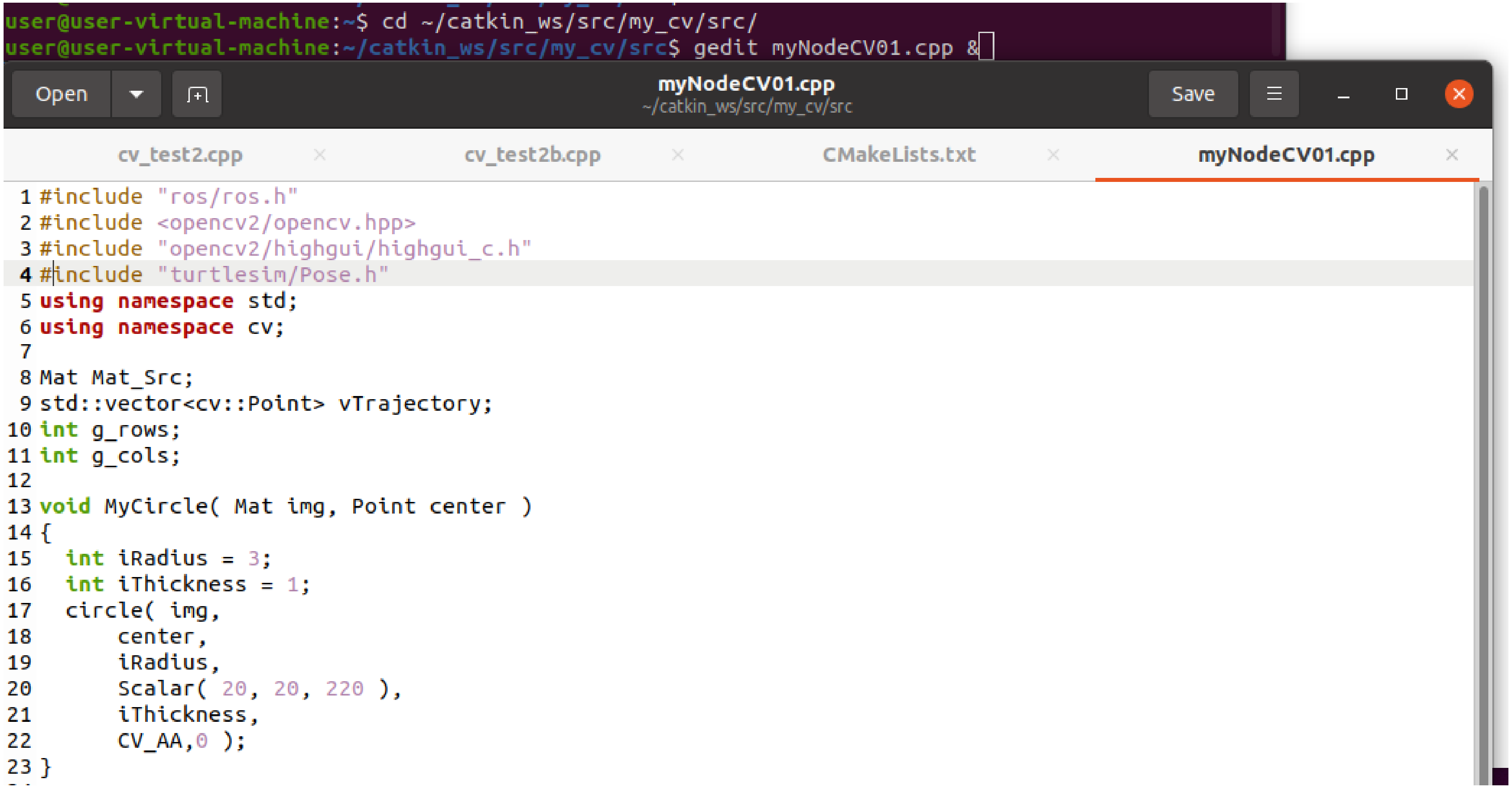
..

Mascaloona.jpg opencv 2 minutes ago

myNodeCV01.cpp opencv 2 minutes ago

© 2023 GitHub, Inc.

ros cv 02



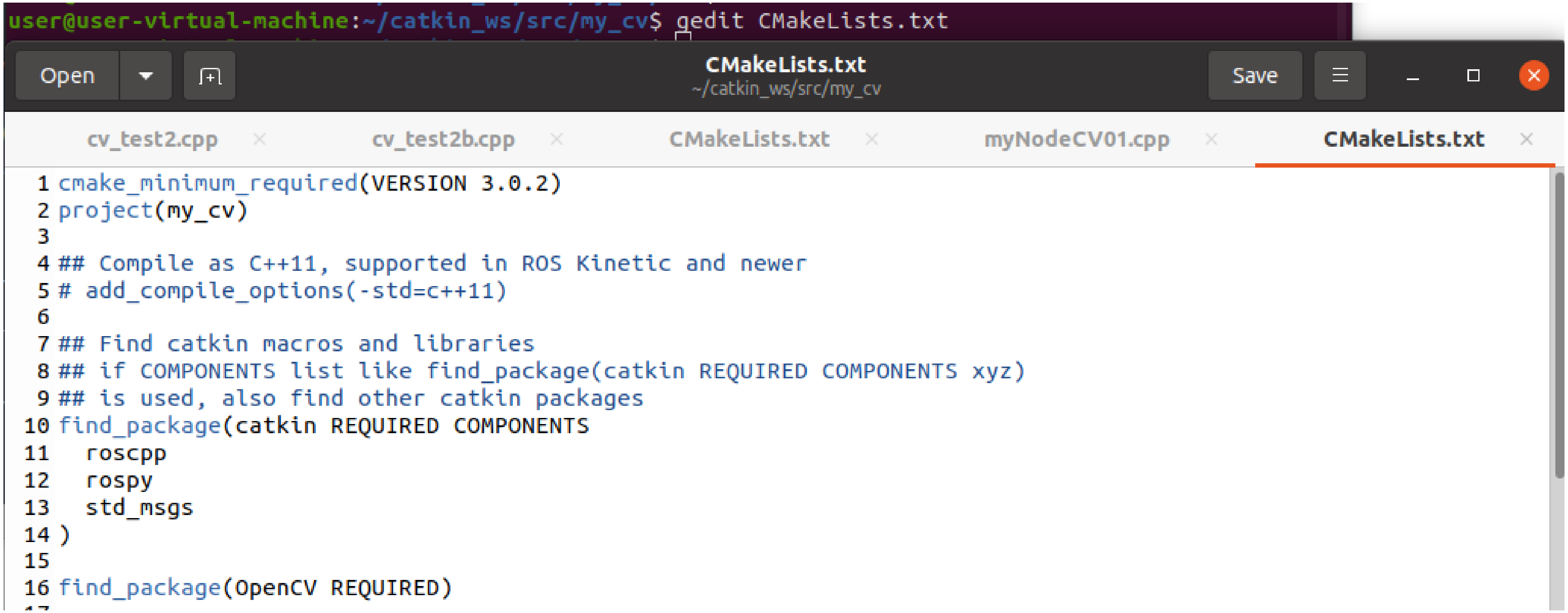
```
user@user-virtual-machine:~$ cd ~/catkin_ws/src/my_cv/src/
user@user-virtual-machine:~/catkin_ws/src/my_cv/src$ gedit myNodeCV01.cpp &
```

```cpp
#include "ros/ros.h"
#include <opencv2/opencv.hpp>
#include "opencv2/highgui/highgui_c.h"
#include "turtlesim/Pose.h"
using namespace std;
using namespace cv;

Mat Mat_Src;
std::vector<cv::Point> vTrajectory;
int g_rows;
int g_cols;

void MyCircle( Mat img, Point center )
{
  int iRadius = 3;
  int iThickness = 1;
  circle( img,
      center,
      iRadius,
      Scalar( 20, 20, 220 ),
      iThickness,
      CV_AA,0 );
}
```

```
#include "ros/ros.h"
#include <opencv2/opencv.hpp>
#include "opencv2/highgui/highgui_c.h"
#include "turtlesim/Pose.h"
using namespace std;
using namespace cv;

Mat Mat_Src;
std::vector<cv::Point> vTrajectory;
int g_rows;
int g_cols;

void MyCircle( Mat img, Point center )
{
  int iRadius = 3;
  int iThickness = 1;
  circle( img,
      center,
      iRadius,
      Scalar( 20, 20, 220 ),
      iThickness,
      CV_AA,0 );
}
```

```
void img_Draw(void)
{
 float fScale = 40;
 if(vTrajectory.size()>1)
 {
  for(auto it:vTrajectory)
  {
   Point2f pp(it.x*fScale, g_rows-it.y*fScale);
   MyCircle(Mat_Src,pp);
  }
 }
 imshow("GUI", Mat_Src);
 cvWaitKey(30);
}

void cb_get_pose(const turtlesim::Pose& msg)
{
 Point2f _pose(msg.x, msg.y);
 static Point2f _pose_k_1(0,0);
 if(_pose!=_pose_k_1)
 {
  vTrajectory.push_back(_pose);
  _pose_k_1 = _pose;
 }
 ROS_INFO("x = %f, y = %f, theta = %f [%d]", msg.x, msg.y,
msg.theta, vTrajectory.size());
}
```

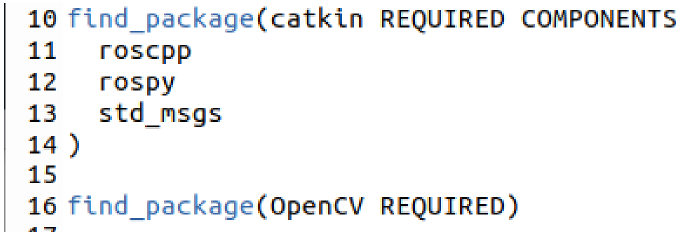
```
int main(int argc, char **argv)
{
 ros::init(argc, argv, "myNodeCV01");
 ros::NodeHandle n;
 ros::Subscriber sub_turtle_pose = n.subscribe("/turtle1/pose",
10, cb_get_pose);
 cvNamedWindow("GUI",0);
 Mat_Src =
imread("/home/user/catkin_ws/src/my_cv/src/Mascaloona.jpg
");
 g_rows = Mat_Src.rows;
 g_cols = Mat_Src.cols;

 ROS_INFO("myNodeCV01: hi");
 ros::Rate r(30);
 while(ros::ok())
 {
  img_Draw();
  r.sleep();
  ros::spinOnce();
 }
 return 0;
}
```



```
user@user-virtual-machine:~/catkin_ws/src/my_cv$ gedit CMakeLists.txt
```

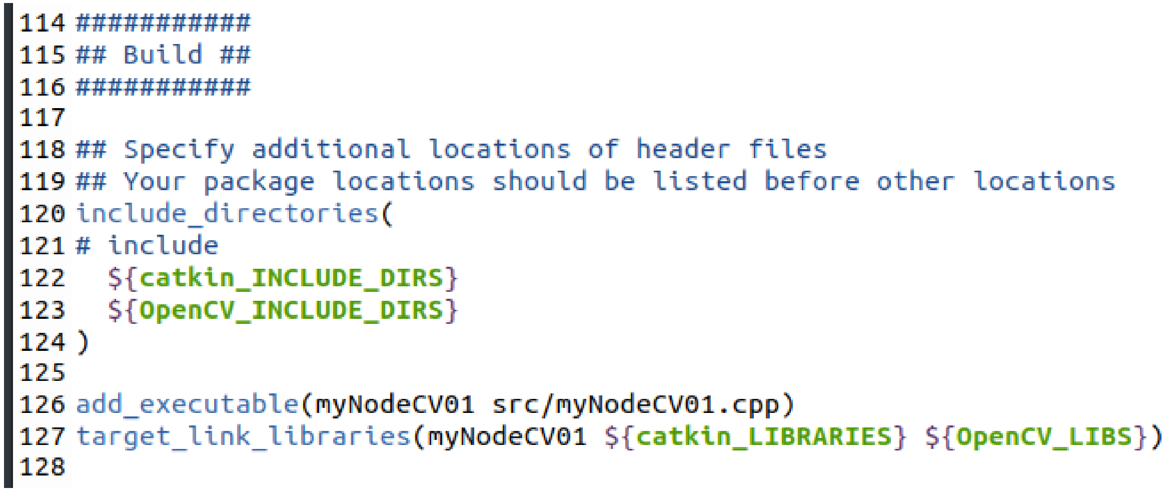
```
cmake_minimum_required(VERSION 3.0.2)
project(my_cv)

## Compile as C++11, supported in ROS Kinetic and newer
# add_compile_options(-std=c++11)

## Find catkin macros and libraries
## if COMPONENTS list like find_package(catkin REQUIRED COMPONENTS xyz)
## is used, also find other catkin packages
find_package(catkin REQUIRED COMPONENTS
  roscpp
  rospy
  std_msgs
)

find_package(OpenCV REQUIRED)
```

find_package(OpenCV REQUIRED)



```
find_package(catkin REQUIRED COMPONENTS
  roscpp
  rospy
  std_msgs
)

find_package(OpenCV REQUIRED)
```

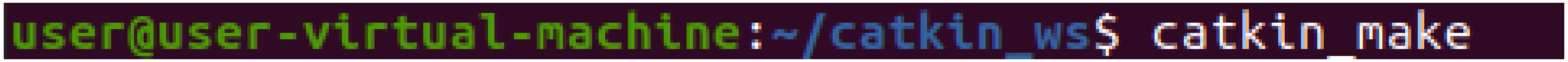
```
find_package(OpenCV REQUIRED)
```



```
###########
## Build ##
###########

## Specify additional locations of header files
## Your package locations should be listed before other locations
include_directories(
# include
  ${catkin_INCLUDE_DIRS}
  ${OpenCV_INCLUDE_DIRS}
)

add_executable(myNodeCV01 src/myNodeCV01.cpp)
target_link_libraries(myNodeCV01 ${catkin_LIBRARIES} ${OpenCV_LIBS})
```

```
include_directories(
  ${catkin_INCLUDE_DIRS}
  ${OpenCV_INCLUDE_DIRS}
)
add_executable(myNodeCV01 src/myNodeCV01.cpp)
target_link_libraries(myNodeCV01 ${catkin_LIBRARIES} ${OpenCV_LIBS})
```

ros cv 03



```
user@user-virtual-machine:~/catkin_ws$ catkin_make
```

```
cd ~/catkin_ws
catkin_make
```

編譯程式

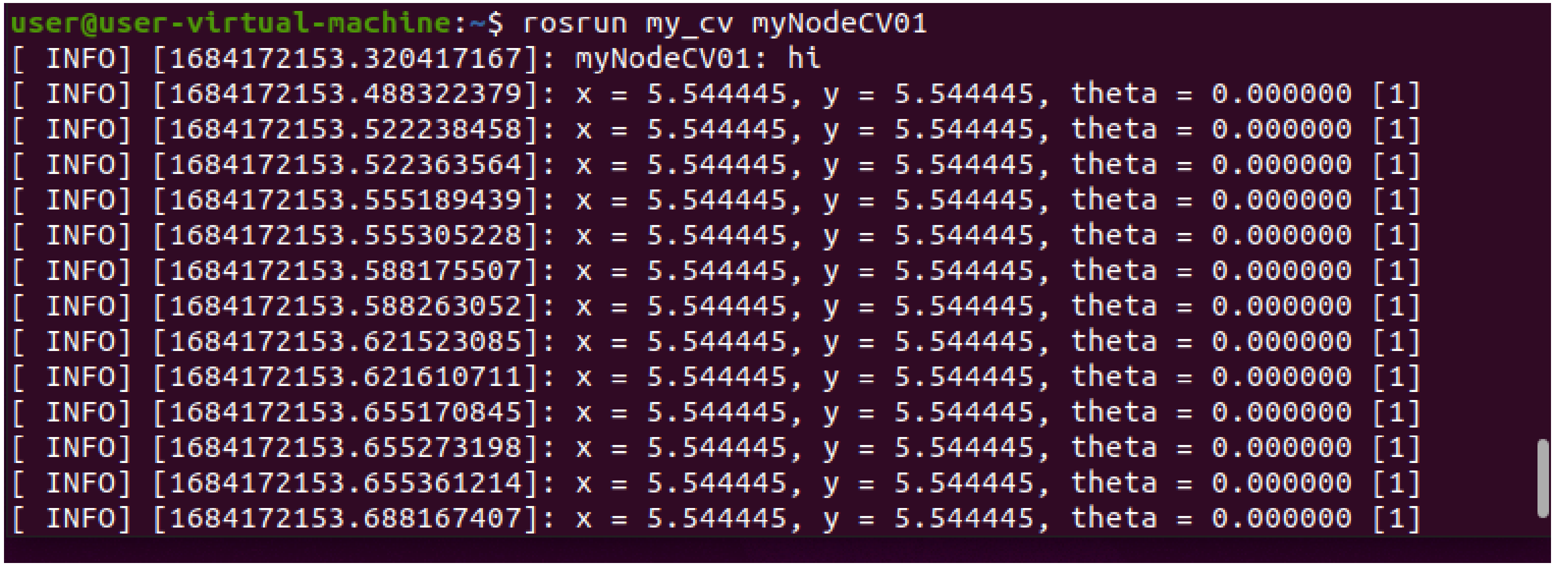
ros cv 04

roscore

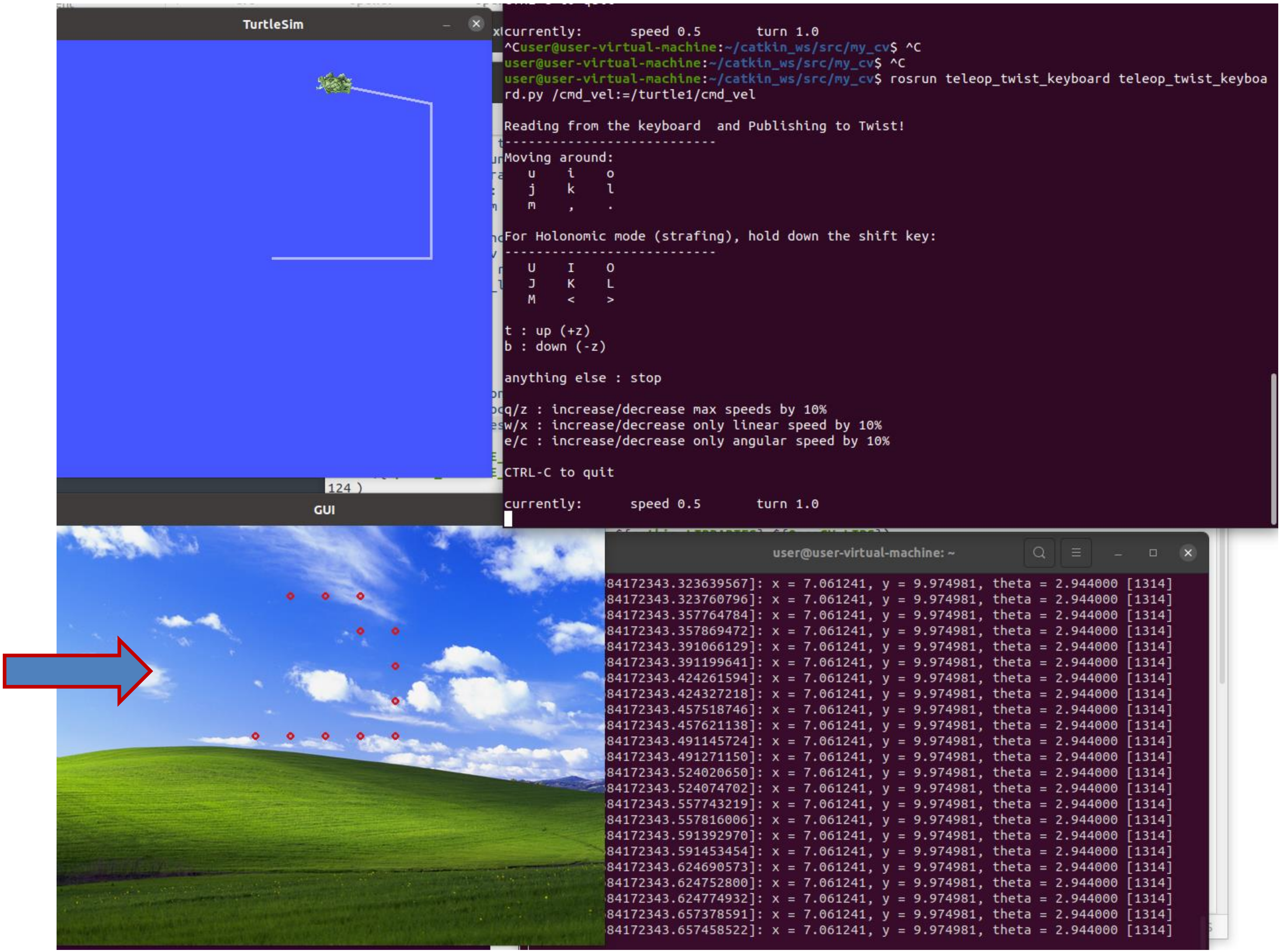
rosrun turtlesim turtlesim_node

rosrun teleop_twist_keyboard teleop_twist_keyboard.py /cmd_vel:=/turtle1/cmd_vel

rosrun my_cv myNodeCV01



```
user@user-virtual-machine:~$ rosrun my_cv myNodeCV01
[ INFO] [1684172153.320417167]: myNodeCV01: hi
[ INFO] [1684172153.488322379]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.522238458]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.522363564]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.555189439]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.555305228]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.588175507]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.588263052]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.621523085]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.621610711]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.655170845]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.655273198]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.655361214]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
[ INFO] [1684172153.688167407]: x = 5.544445, y = 5.544445, theta = 0.000000 [1]
```



```
currently:      speed 0.5       turn 1.0
^Cuser@user-virtual-machine:~/catkin_ws/src/my_cv$ ^C
user@user-virtual-machine:~/catkin_ws/src/my_cv$ ^C
user@user-virtual-machine:~/catkin_ws/src/my_cv$ rosrun teleop_twist_keyboard teleop_twist_keyboa
rd.py /cmd_vel:=/turtle1/cmd_vel

Reading from the keyboard  and Publishing to Twist!
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

For Holonomic mode (strafing), hold down the shift key:
---------------------------
   U    I    O
   J    K    L
   M    <    >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:      speed 0.5       turn 1.0
```

```
84172343.323639567]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.323760796]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.357764784]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.357869472]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.391066129]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.391199641]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.424261594]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.424327218]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.457518746]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.457621138]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.491145724]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.491271150]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.524020650]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.524074702]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.557743219]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.557816006]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.591392970]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.591453454]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.624690573]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.624752800]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.624774932]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.657378591]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
84172343.657458522]: x = 7.061241, y = 9.974981, theta = 2.944000 [1314]
```