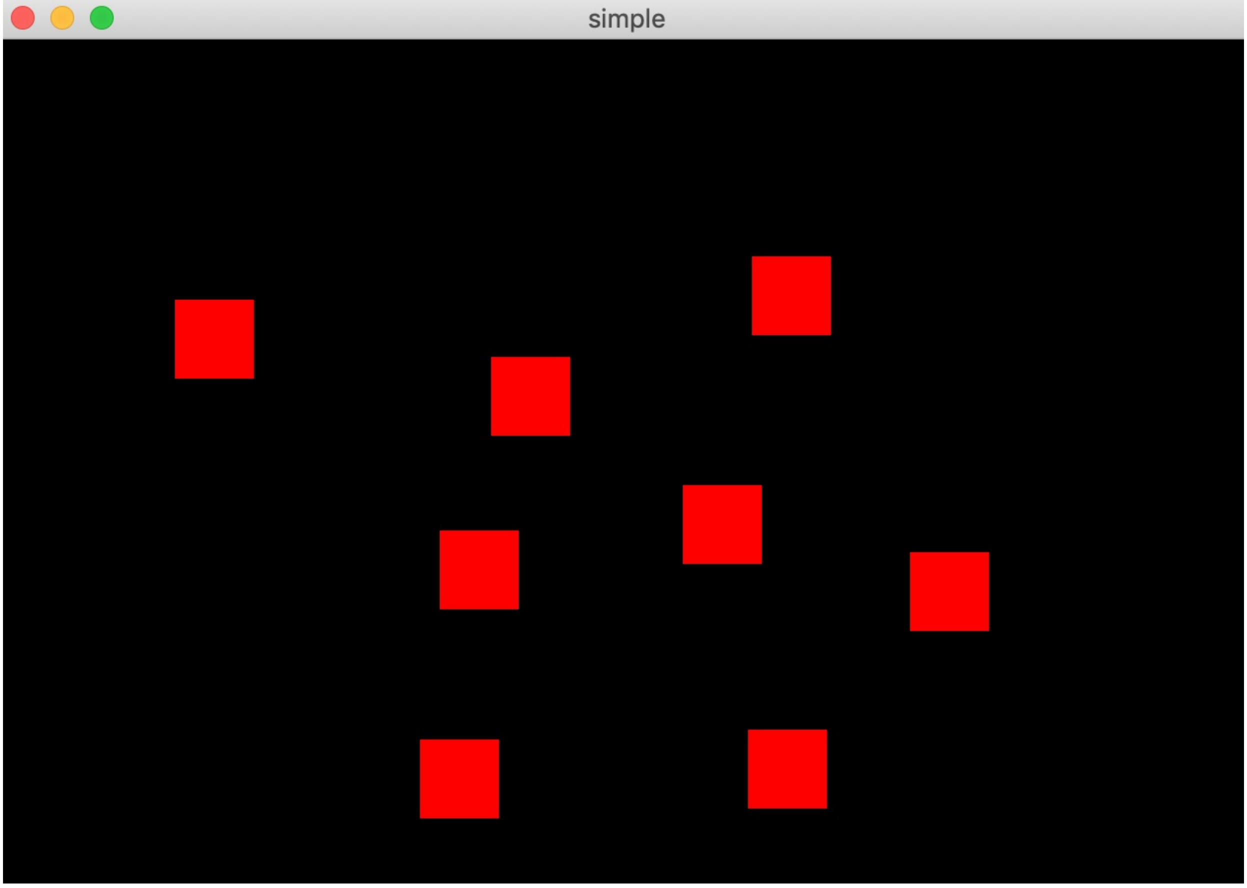
Felipe Costa - Assignment 2A Report

I used CMake[1] in order to build each project in this assignment. Each part has a separate CMakeFile, however they are all implemented the same manner.

```
cmake_minimum_required(VERSION 3.5)

project(Part-B)
set(CMAKE_CXX_FLAGS "-std=c++11")

set(APPLE TRUE)

# Suppress warnings of the deprecation of glut functions on macOS.
add_definitions(-Wno-deprecated-declarations)

# Find OpenGL
find_package(OpenGL REQUIRED)
find_package(GLUT REQUIRED)

# Find include and library for GLFW and GLEW
set(GLFW_INCLUDE "/usr/local/Cellar/glfw/3.3.2/include/")
set(GLEW_INCLUDE "/usr/local/Cellar/glew/2.1.0_1/include/")
set(GLFW_LIB "/usr/local/Cellar/glfw/3.3.2/lib/libglfw.3.3.dylib")
set(GLEW_LIB "/usr/local/Cellar/glew/2.1.0_1/lib/libGLEW.2.1.0.dylib")

set(INCLUDE_DIRS include ${OPENGL_INCLUDE_DIR} ${GLEW_INCLUDE} ${GLFW_INCLUDE})
set(LIBRARIES ${OPENGL_LIBRARIES} ${GLUT_LIBRARIES} ${GLEW_LIB} ${GLFW_LIB})

# Add the list of include paths to be used to search for include files.
include_directories(${INCLUDE_DIRS})

# Search all the .cpp files in the directory where CMakeLists lies
#and set them to ${SOURCE_FILES}.
file(GLOB SOURCE_FILES ${CMAKE_CURRENT_SOURCE_DIR}/*.cpp)

# Search all the .h files in the directory where CMakeLists lies
# and set them to ${INCLUDE_FILES}.
file(GLOB INCLUDE_FILES ${CMAKE_CURRENT_SOURCE_DIR}/include/*.h)

# Add the executable Part-A to be built from the source files.
add_executable(${PROJECT_NAME} ${SOURCE_FILES} ${INCLUDE_FILES})

# Link the executable to the libraries.
target_link_libraries(${PROJECT_NAME} ${LIBRARIES})
```

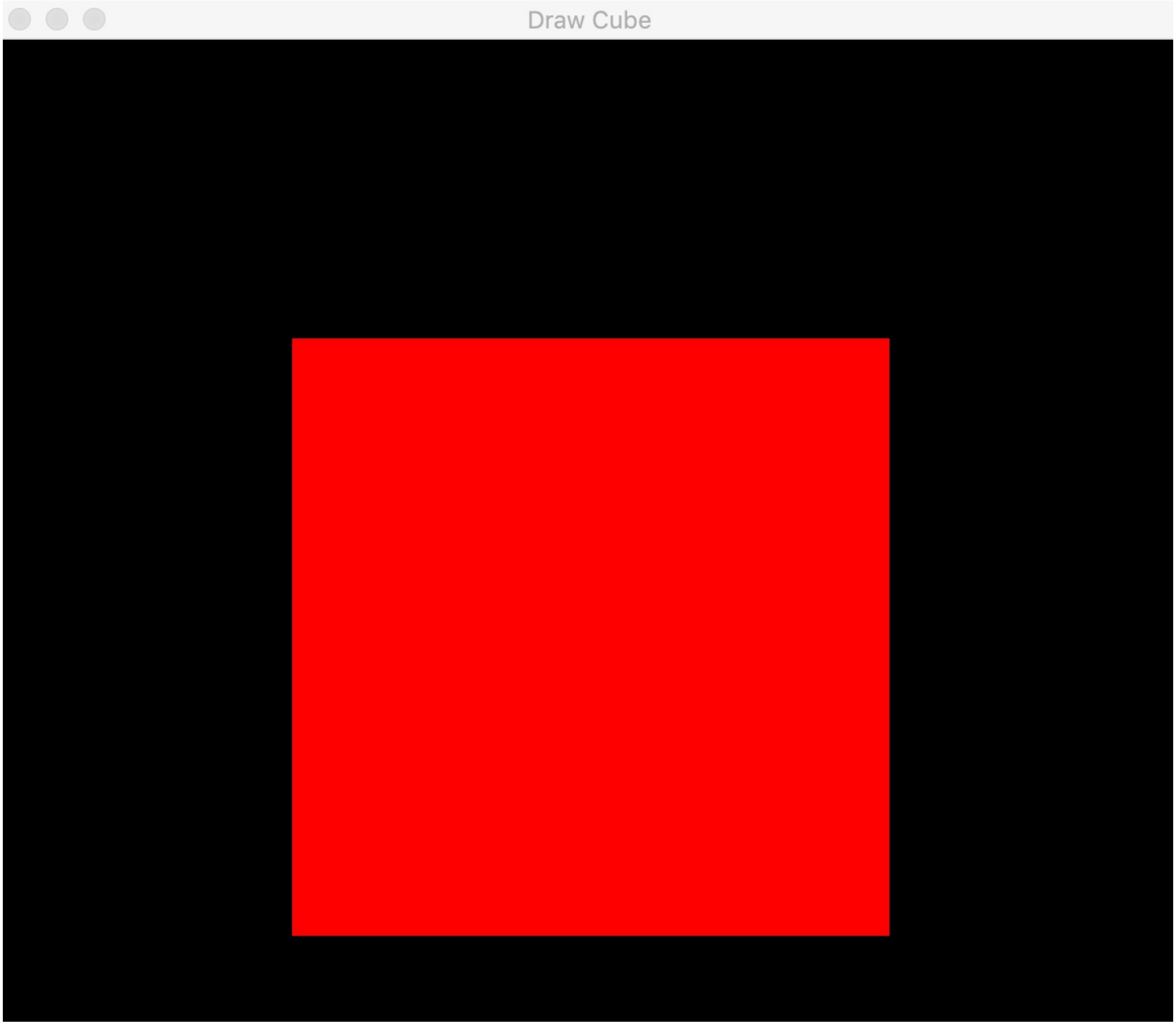
Part A

Download the program DrawSquare.cpp, create a VC++ project, compile and run it.



Part B

Download the program DrawCube.cpp that will Render a 3D cube in OpenGL.



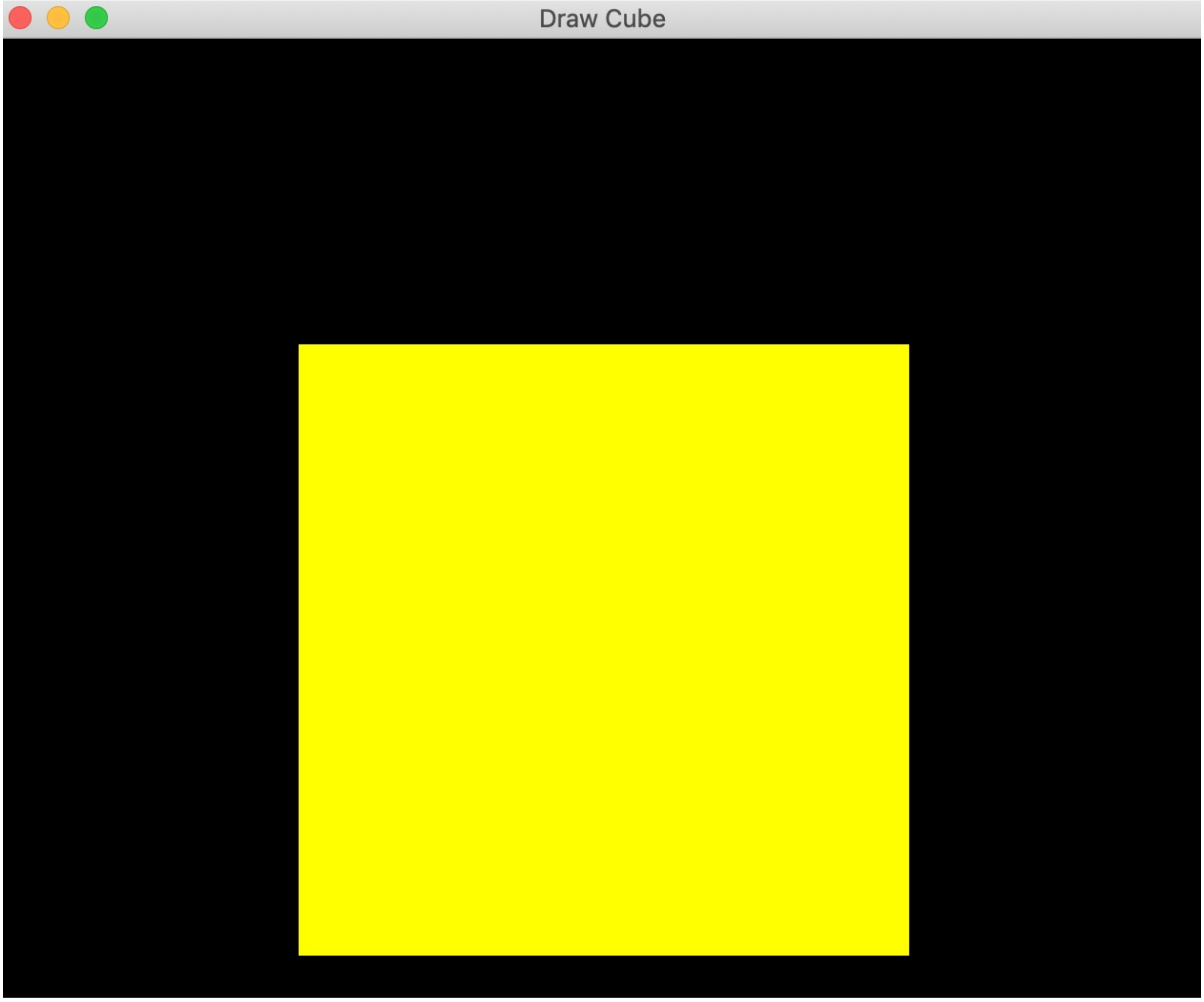
Add the following callback functions into the code

[1] Click the right mouse button to exit application

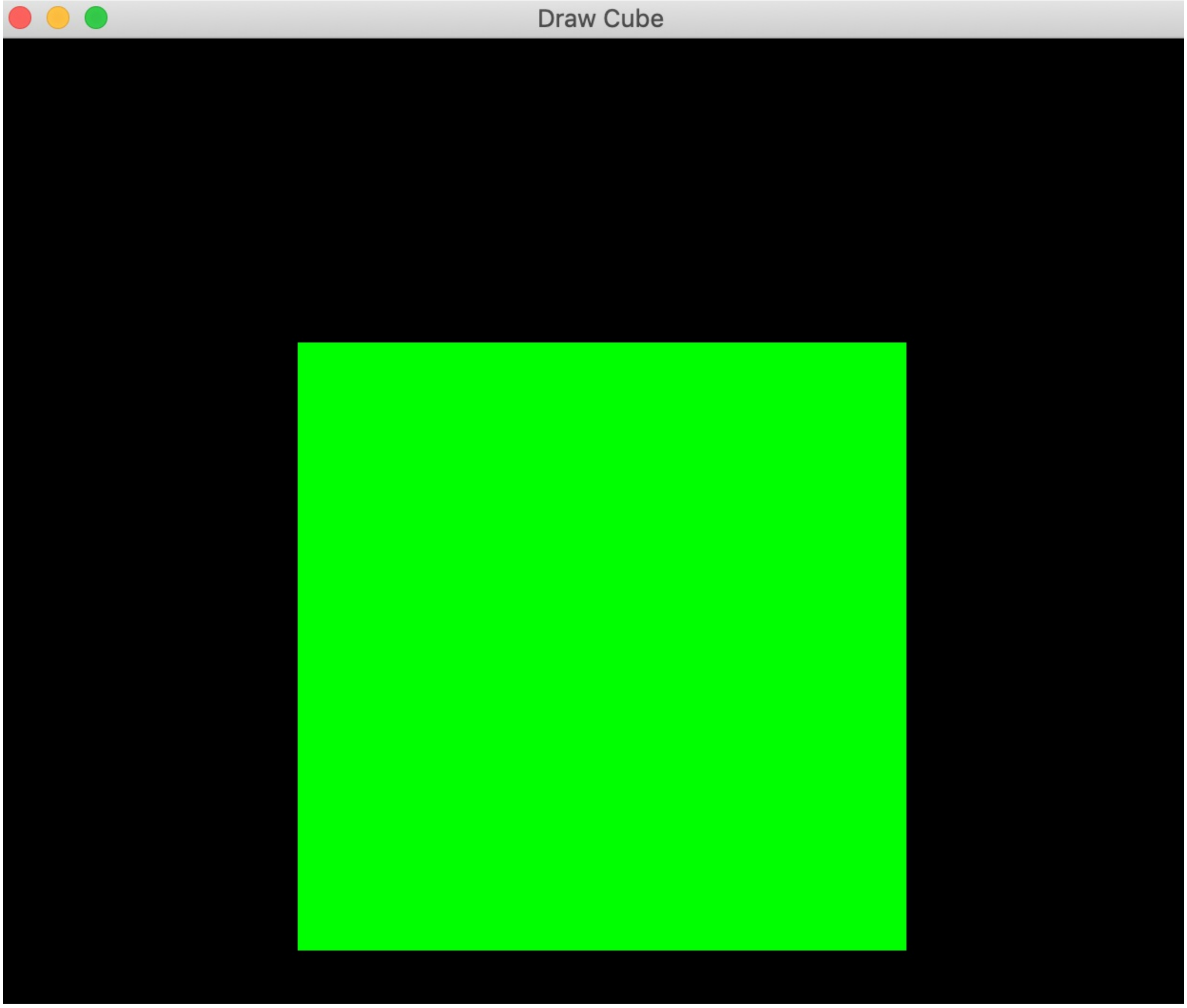
```
void mouse_exit(GLFWwindow* window, int button, int action, int mods) {
    if (button == GLFW_MOUSE_BUTTON_RIGHT && action == GLFW_PRESS){
        // exit the application
        glfwDestroyWindow(window);
        glfwTerminate();
        exit(EXIT_SUCCESS);
    }
}
```

[2] Press a key in the keyboard to change the color of the whole cube

```
//Add initial color
const char *fragment_shader = "#version 410\n"
  "uniform vec4 color = vec4(1.0, 0.0, 0.0, 1.0);"
  "out vec4 FragColor;"
  "void main () {"
  "   /*FragColor = vec4( 1.0, 0.0, 0.0, 1.0 );*/"
  "   FragColor = color;"
  "}";
...
//Function to change Cube color on keyboard press
void keyboard_color(GLFWwindow* window, int key, int scancode, int
                     action, int mods) {
    if (action == GLFW_PRESS) {
        col = glGetUniformLocation(shader_programme "color");
        int r = 0, g = 0, b = 0;
        while (r == 0 && g == 0 && b == 0) {
            r = rand() % 2;
            g = rand() % 2;
            b = rand() % 2;
        }
        vec4 m = vec4(r, g, b, 1.0);
        glUniform4fv(col, 1, m);
    }
}
```



Draw Cube



Draw Cube