3D Computer Graphics Lab 3 code quality A

系級：資工4B

學號：105502042

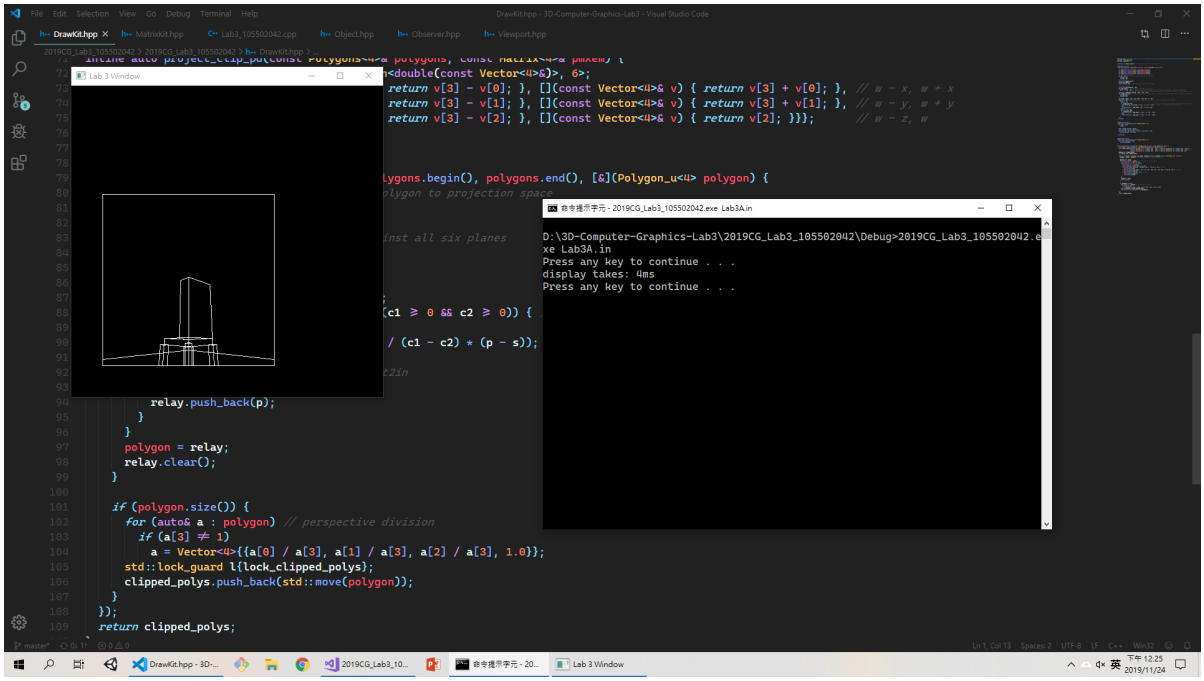
姓名：張博堯

STDOUT 輸出

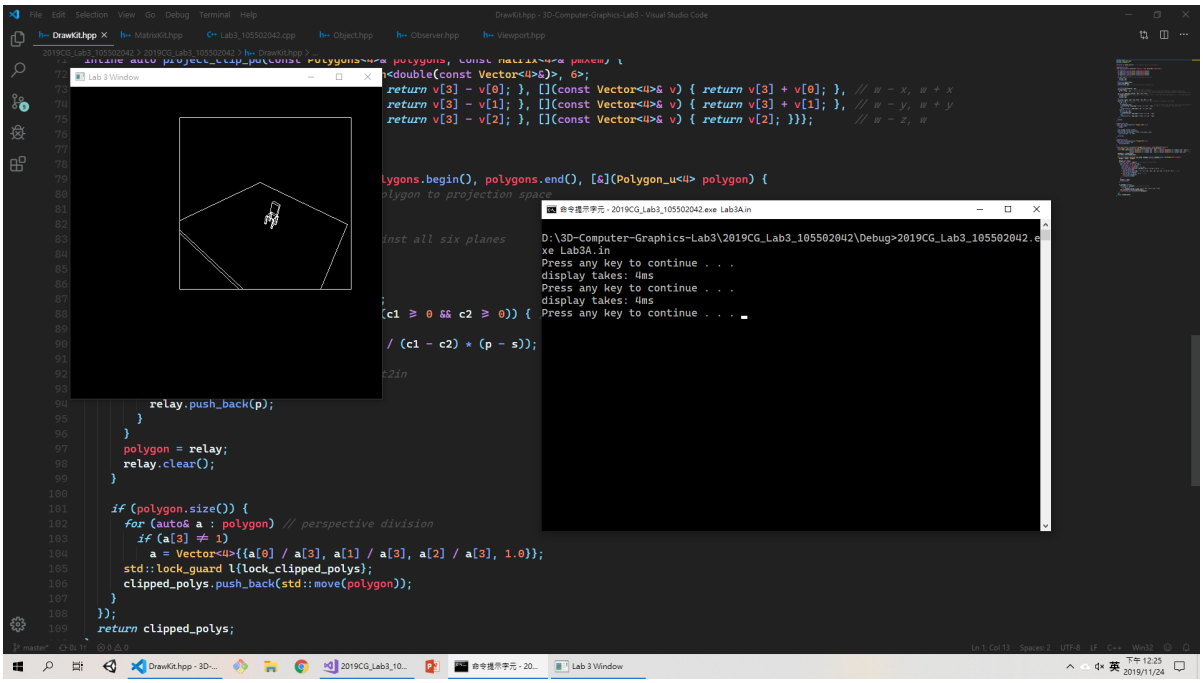
輸出處理 `display` 指令花的時間，以毫秒(ms)計算。

截圖

Lab3A.in

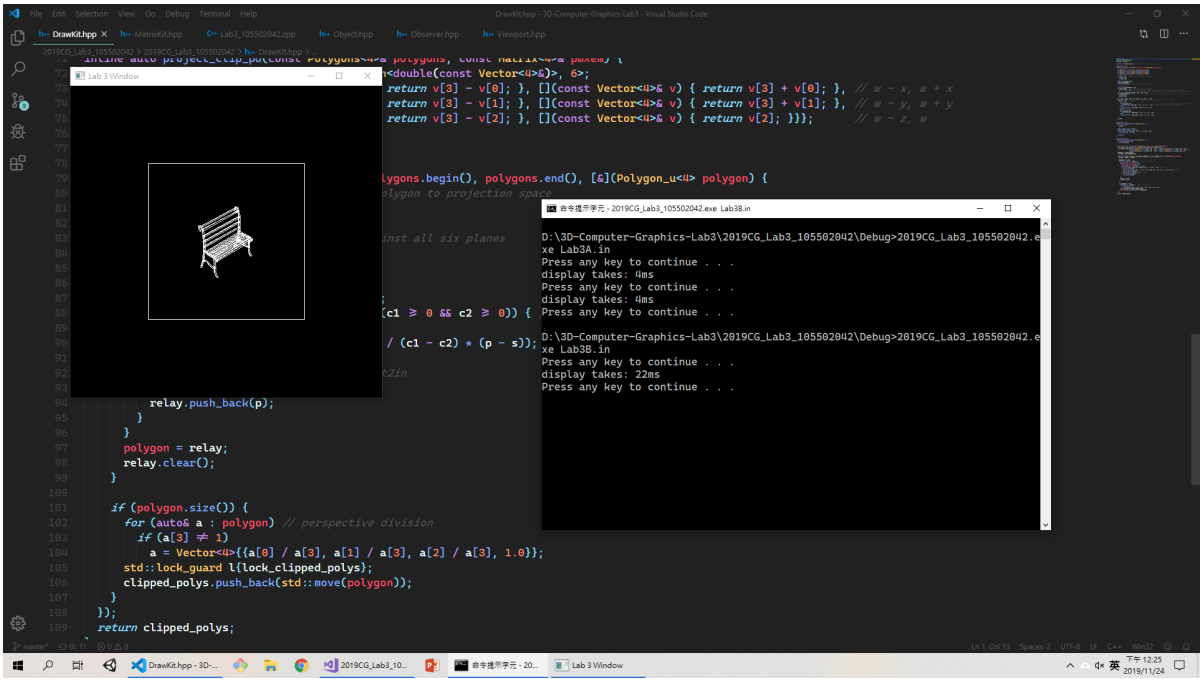
1. 

2.

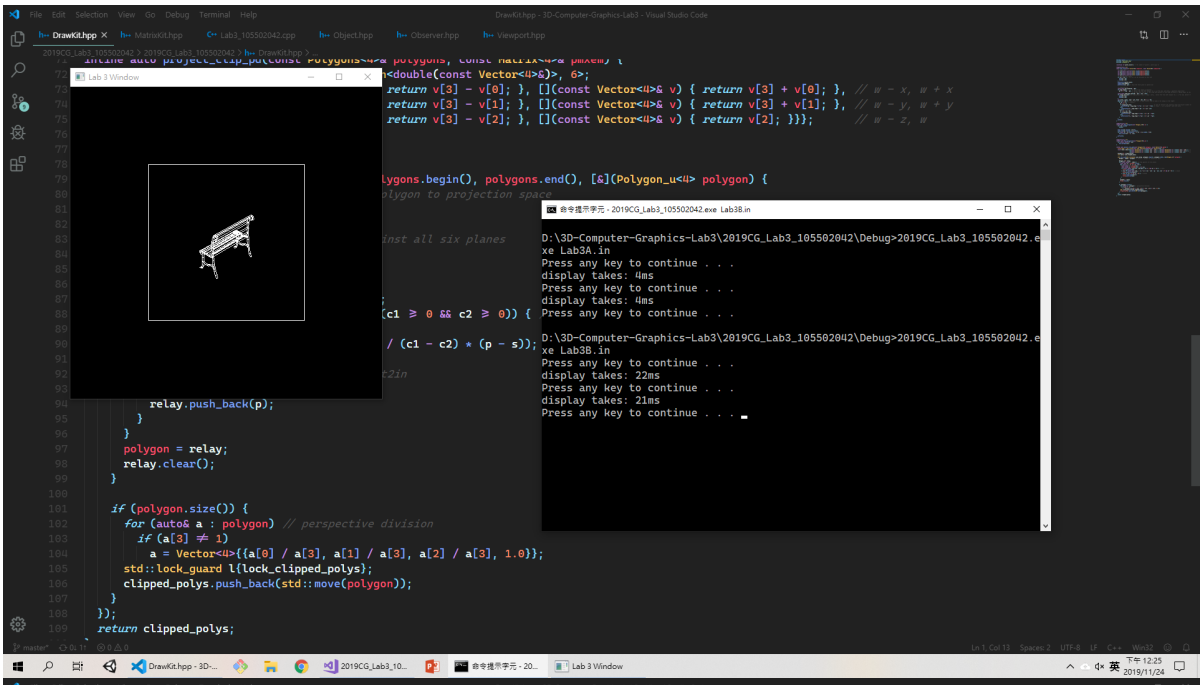

Lab 3 Window
命令提示字元 - 2019CG_Lab3_105502042.exe Lab3A.in
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3A.in
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .

Lab3B.in

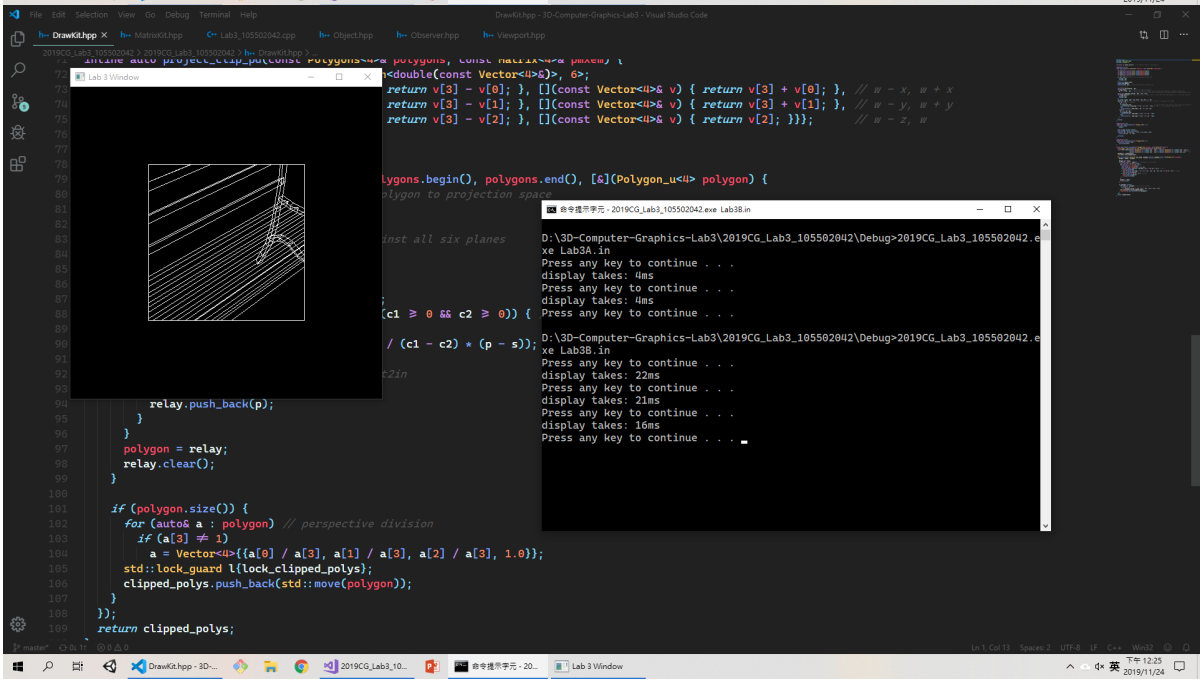
1.


Lab 3 Window
命令提示字元 - 2019CG_Lab3_105502042.exe Lab3B.in
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3A.in
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3B.in
Press any key to continue . . .
display takes: 22ms
Press any key to continue . . .

2.

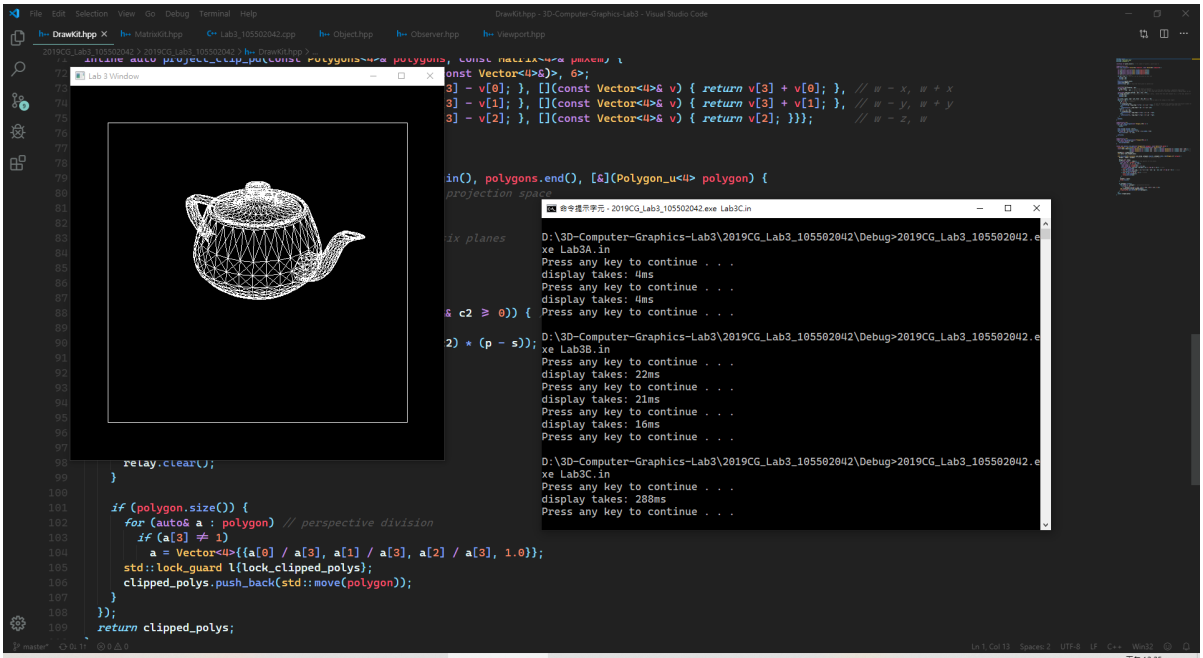

Lab 3 Window
命令提示字元 - 2019CG_Lab3_105502042.exe Lab3B.in
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3A.in
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3B.in
Press any key to continue . . .
display takes: 22ms
Press any key to continue . . .
display takes: 21ms
Press any key to continue . . .

3.

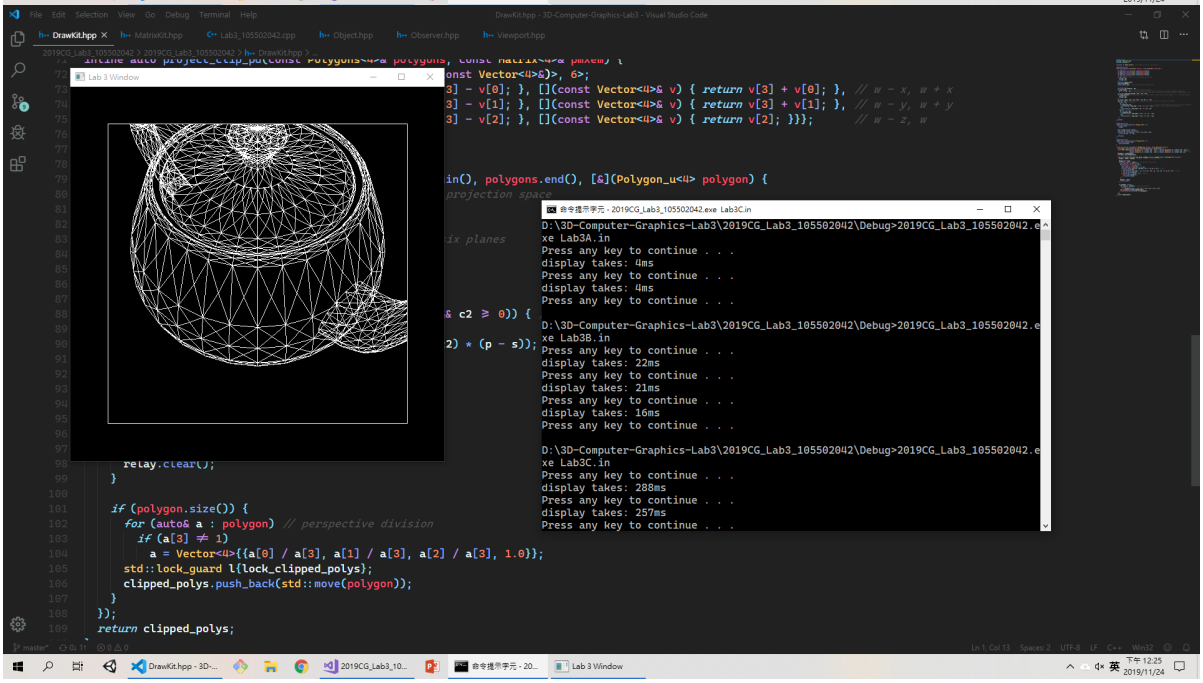

Lab 3 Window
命令提示字元 - 2019CG_Lab3_105502042.exe Lab3B.in
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3A.in
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3B.in
Press any key to continue . . .
display takes: 22ms
Press any key to continue . . .
display takes: 21ms
Press any key to continue . . .
display takes: 16ms
Press any key to continue . . .

Lab3C.in

1.

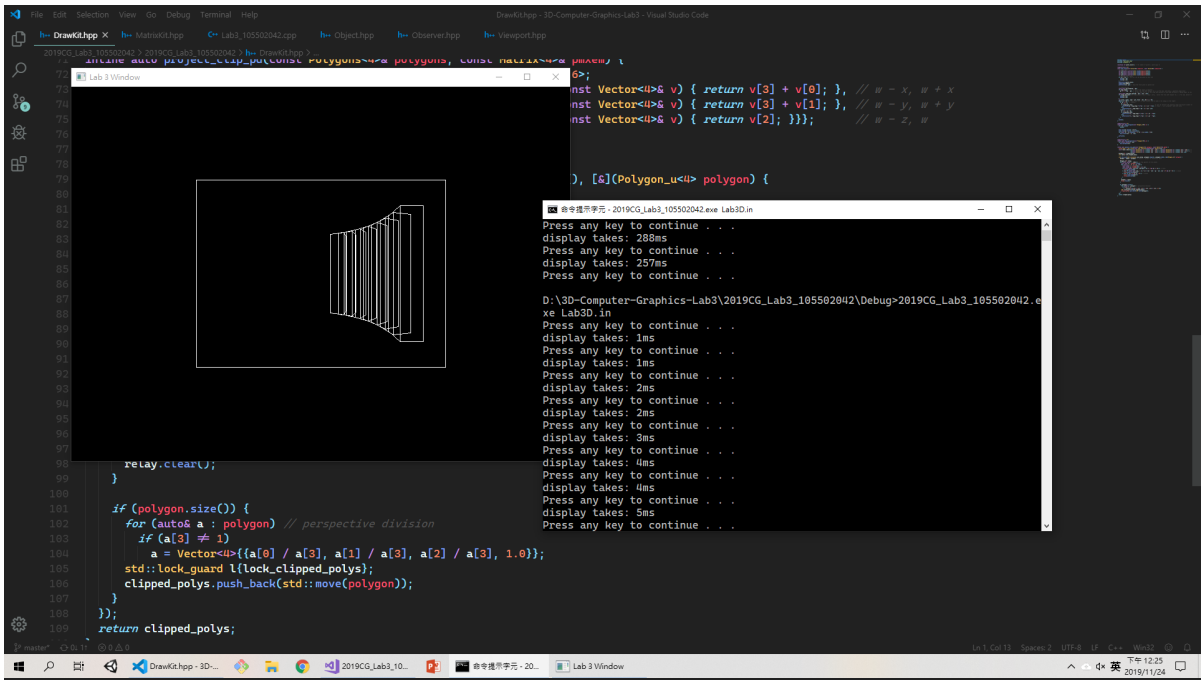

Lab 3 Window
命令提示字元 - 2019CG_Lab3_105502042.exe Lab3C.in
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3A.in
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3B.in
Press any key to continue . . .
display takes: 22ms
Press any key to continue . . .
display takes: 21ms
Press any key to continue . . .
display takes: 16ms
Press any key to continue . . .
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3C.in
Press any key to continue . . .
display takes: 288ms
Press any key to continue . . .

2.

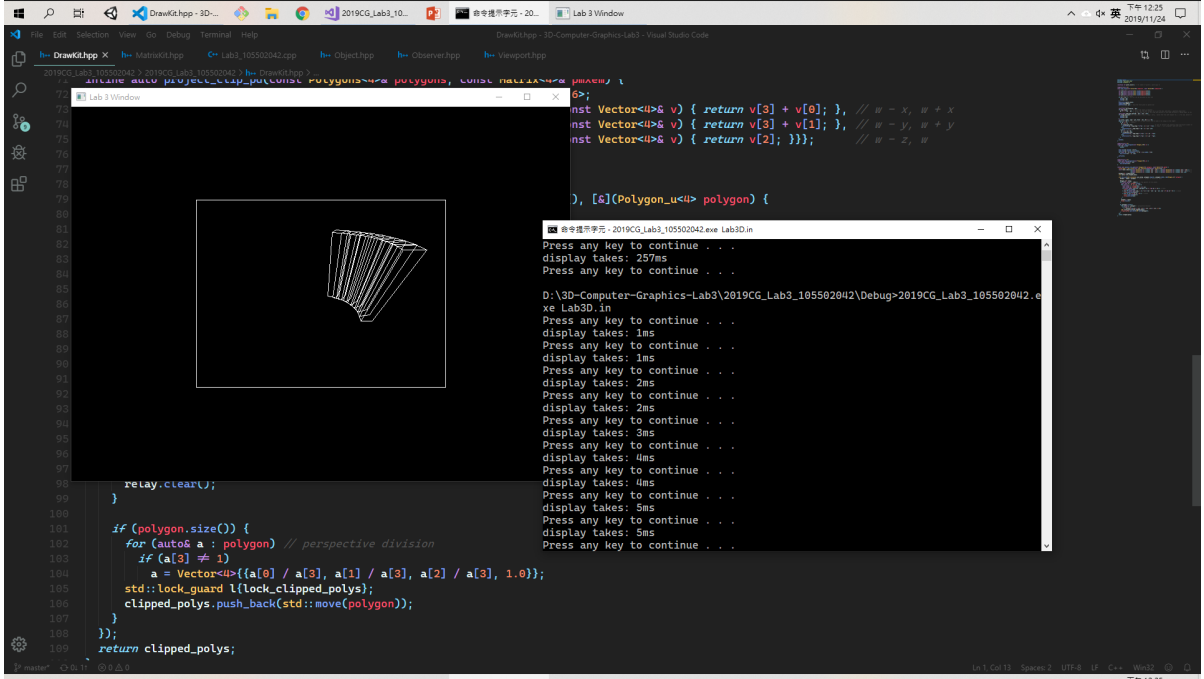

Lab 3 Window
命令提示字元 - 2019CG_Lab3_105502042.exe Lab3C.in
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3A.in
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3B.in
Press any key to continue . . .
display takes: 22ms
Press any key to continue . . .
display takes: 21ms
Press any key to continue . . .
display takes: 16ms
Press any key to continue . . .
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3C.in
Press any key to continue . . .
display takes: 288ms
Press any key to continue . . .
display takes: 257ms
Press any key to continue . . .

Lab3D.in

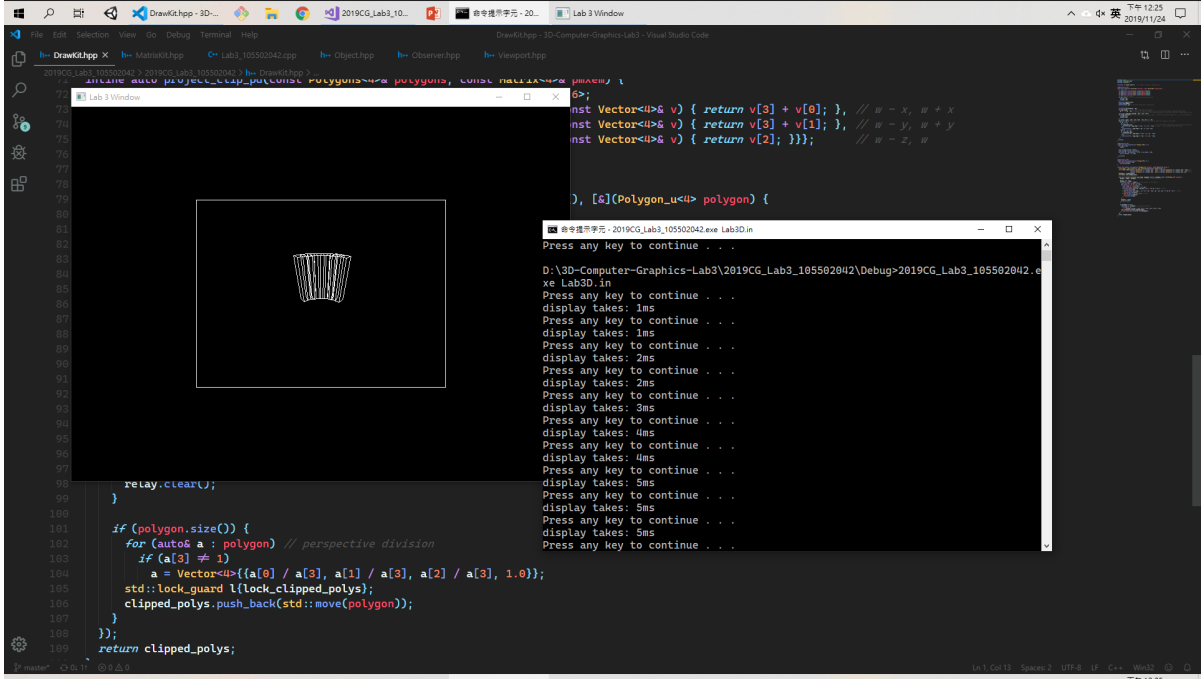
1.


Lab 3 Window
命令提示字元 - 2019CG_Lab3_105502042.exe Lab3D.in
Press any key to continue . . .
display takes: 288ms
Press any key to continue . . .
display takes: 257ms
Press any key to continue . . .
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3D.in
Press any key to continue . . .
display takes: 1ms
Press any key to continue . . .
display takes: 1ms
Press any key to continue . . .
display takes: 2ms
Press any key to continue . . .
display takes: 2ms
Press any key to continue . . .
display takes: 3ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 5ms
Press any key to continue . . .

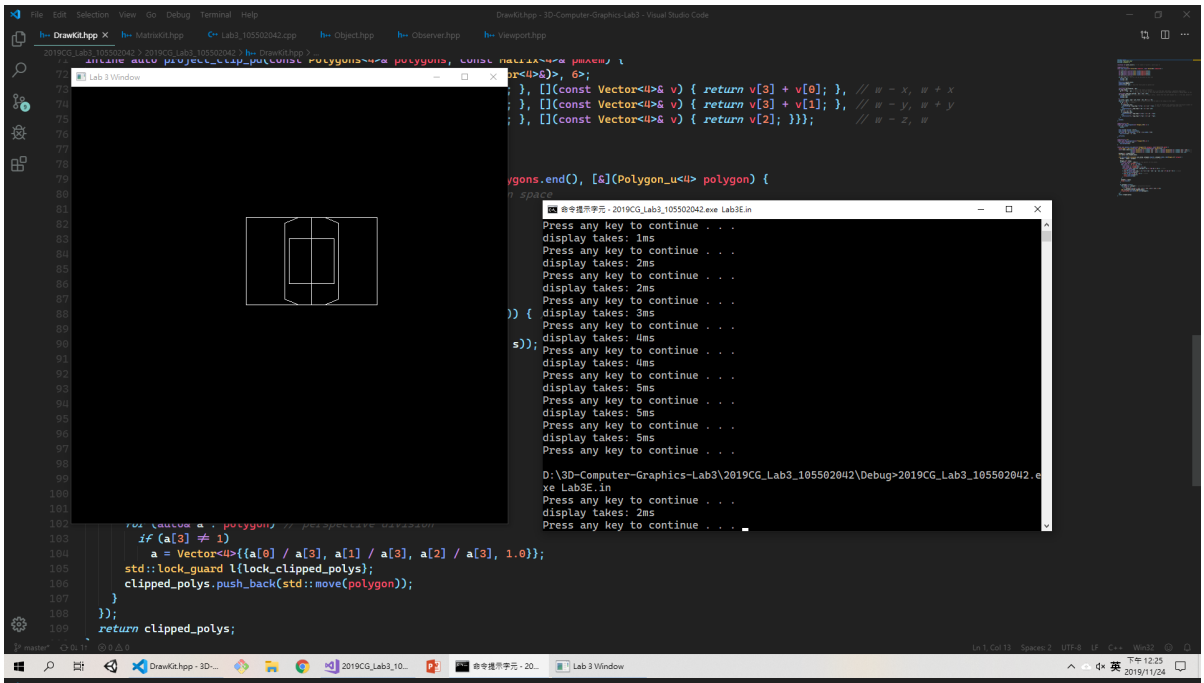
2.

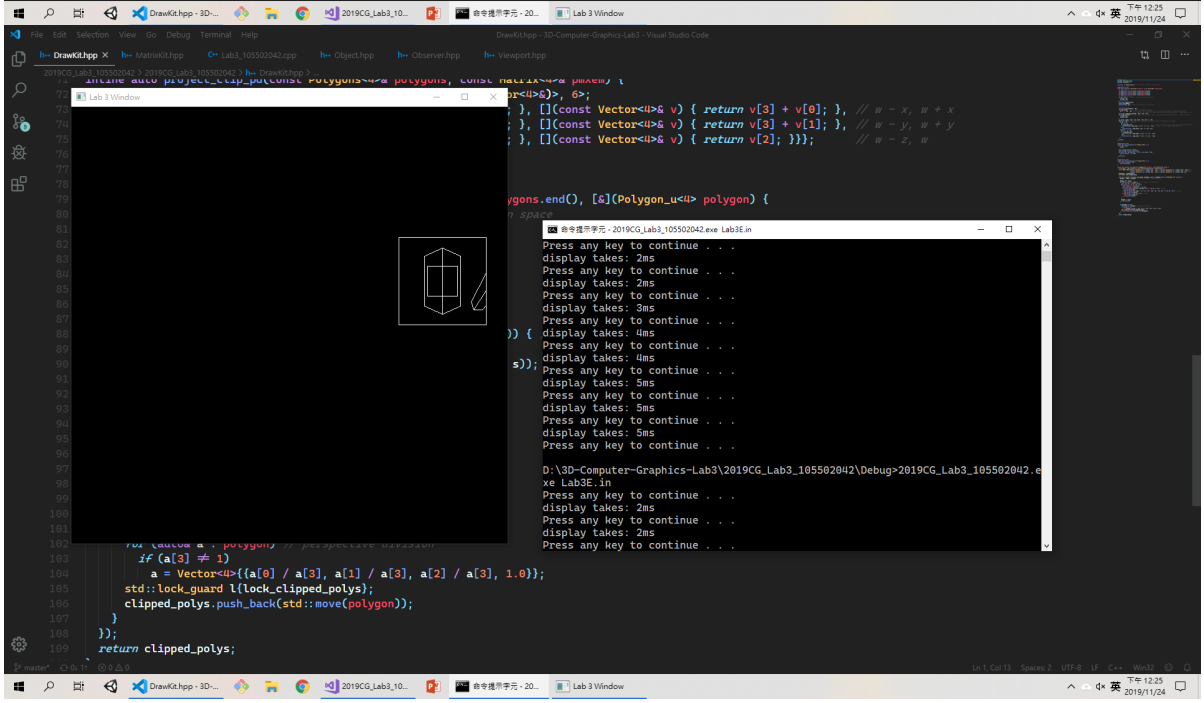

Lab 3 Window
命令提示字元 - 2019CG_Lab3_105502042.exe Lab3D.in
Press any key to continue . . .
display takes: 257ms
Press any key to continue . . .
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3D.in
Press any key to continue . . .
display takes: 1ms
Press any key to continue . . .
display takes: 1ms
Press any key to continue . . .
display takes: 2ms
Press any key to continue . . .
display takes: 2ms
Press any key to continue . . .
display takes: 3ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 5ms
Press any key to continue . . .
display takes: 5ms
Press any key to continue . . .

3.


Lab 3 Window
命令提示字元 - 2019CG_Lab3_105502042.exe Lab3D.in
Press any key to continue . . .
D:\3D-Computer-Graphics-Lab3\2019CG_Lab3_105502042\Debug>2019CG_Lab3_105502042.exe Lab3D.in
Press any key to continue . . .
display takes: 1ms
Press any key to continue . . .
display takes: 1ms
Press any key to continue . . .
display takes: 2ms
Press any key to continue . . .
display takes: 2ms
Press any key to continue . . .
display takes: 3ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 4ms
Press any key to continue . . .
display takes: 5ms
Press any key to continue . . .
display takes: 5ms
Press any key to continue . . .
display takes: 5ms
Press any key to continue . . .

Lab3E.in

1. 

2. 

影片

https://youtu.be/FgfSL_YjRI8

6 Source files

1. Lab3_105502042.cpp ➔ 處理讀檔和每個指令。
2. Viewport.hpp ➔ 定義存放viewport資訊的object，包含了aspect ratio。
3. Object.hpp ➔ 定義了每個 `.asc` 檔讀進來的資料結構，也提供了幾個member function方便讀入 `.asc` 檔使用。在 `set_vertex` 中每個頂點在讀入時即會左乘當時的TM。 `to_polygons()` 則是會回傳多個多邊形 (of type `Polygons<4>` , which is a `std::vector` of `Polygon_u<4>` , which is yet another `std::vector` of `Vector<4>` , which is a `std::array<double, 4>`)。
4. Observer.hpp ➔ 讀到observer指令時會instantiate一個observer物件，提供了一個member function回傳 $projection\ matrix \times eye\ matrix$ 的4×4矩陣。

5. DrawKit.hpp ➔ 定義了畫線、畫多邊形、畫多邊形們和 `project_clip_pd` 這幾個函式。
6. MatrixKit.hpp ➔ 定義了各種和矩陣有關的運算，包含但不限於矩陣乘法、內積、外積、矩陣和向量乘法。有個對多邊形們的每個頂點左乘矩陣的函式也運用了上述的 `std::execution::par_unseq` execution policy。

footnote: A Tour of C++ (Second edition) is a concise, insightful, well-written book on C++17, authored by the creator of C++, Bjarne Stroutrup. A lot of the language constructs that I use come from this book.

More on the rendering pipeline

Indside of `project_clip_pd`

這個函式把多邊形們從world space轉到projection space (line 80)，再clipping (lines 82 to 90)，再做perspective divide (lines 102 to 104)，然後回傳處理好的多邊形們給 `process_display` 留待nobackfaces處理。

這個函式利用了 `std::for_each` (line 79)這個higher-order function，我把我要做的事情寫成一個lambda expression (lines 80 to 108)當成fisrt-class function成為他的參數，參數部分再多加入 `std::execution::par_unseq` (parallel and unsequenced，但MSVC似乎只實作了parallel，也就是multi-threading的部分) ，使得整件事可以平行處理，但如此便要加lock🔒 (lines 77 and 105)否則會crash。

最外層的 `std::for_each` 會iterate over每個polygon，內層處理clipping的部分 `for (const auto& c : codes)` 則是iterate through 6 個functors (lines 73 to 75)對應到老師筆記的6個不等式，比如說 `c(S)` 可能就會回傳S這個點的$S_w - S_x$。如此一來便不用寫6個長得差不多的for迴圈了✌️ 。

```
70
71  inline auto project_clip_pd(const Polygons<4>& polygons, const Matrix<4>& pmXem) {
72    using Codes = std::array<const std::function<double(const Vector<4>&)>, 6>;
73    const Codes codes{{[](const Vector<4>& v) { return v[3] - v[0]; }, [](const Vector<4>& v) { return v[3] + v[0]; }, // w - x, w + x
74                       [](const Vector<4>& v) { return v[3] - v[1]; }, [](const Vector<4>& v) { return v[3] + v[1]; }, // w - y, w + y
75                       [](const Vector<4>& v) { return v[3] - v[2]; }, [](const Vector<4>& v) { return v[2]; }}};      // w - z, w
76    Polygons<4> clipped_polys;
77    std::mutex lock_clipped_polys;
78
79    std::for_each(std::execution::par_unseq, polygons.begin(), polygons.end(), [&](Polygon_u<4> polygon) {
80      polygon = pmXem * polygon; // project a polygon to projection space
81
82      Polygon_u<4> relay;
83      for (const auto& c : codes) { // clip against all six planes
84        const auto sz = polygon.size();
85        for (size_t i = 0; i < sz; ++i) {
86          const auto& s = polygon[i];
87          const auto& p = polygon[(i + 1) % sz];
88          if (const double c1{c(s)}, c2{c(p)}; (c1 >= 0 && c2 >= 0)) { // in2in
89            relay.push_back(p);
90          } else if (const auto new_s = (s + c1 / (c1 - c2) * (p - s)); (c1 >= 0 && c2 < 0)) { // in2out
91            relay.push_back(new_s);
92          } else if (c1 < 0 && c2 >= 0) { // out2in
93            relay.push_back(new_s);
94            relay.push_back(p);
95          }
96        }
97        polygon = relay;
98        relay.clear();
99      }
100
101     if (polygon.size()) {
102       for (auto& a : polygon) // perspective division
103         if (a[3] != 1)
104           a = Vector<4>{{a[0] / a[3], a[1] / a[3], a[2] / a[3], 1.0}};
105       std::lock_guard l{lock_clipped_polys};
106       clipped_polys.push_back(std::move(polygon));
107     }
108   });
109   return clipped_polys;
110 }
111
```

Call site of `project_clip_pd` (from world space to screen space)

lines 75–80 ➔ 把所有讀進來過的objects都轉成多邊形們等下一次處理。

line 82 ➔ 所有多邊形從world space到image space

lines 84–88 ➔ nobackfaces處理，移除所有法向量和Z方向夾角 $\theta \leq 90^\circ$ 的多邊形

lines 90–93 ➔ 清除畫面並畫出viewport邊界，其中 `vp.get_borders()` 回傳一個tuple可以透過structured binding 賦值給左側的四個變數

line 94 ➔ 轉到screen space後畫在螢幕上，其中最右邊的 `*` 平行地對多邊形們(`ps`)左乘左側的矩陣

參數中的 `pmXem` 為 $projection\ matrix \times eye\ matrix =$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & A_R & 0 & 0 \\ 0 & 0 & \frac{y}{y-H}\tan\theta & \frac{H\times y}{H-y}\tan\theta \\ 0 & 0 & \tan\theta & 0 \end{pmatrix} \times Rotate_{(-Tilt)} \times Mirror_{(x)} \times GRM \times Translate_{(-E_x,-E_y,-E_z)}$$

```
auto process_display(const Viewport& vp, const std::vector<Object>& objects, const Matrix<4>& pmXem) {
  auto t0 = std::chrono::high_resolution_clock::now();
  // dump all faces of all objects to Polygons<4>
  Polygons<4> ps;
  for (const auto& obj : objects) {
    const auto&& a = obj.to_polygons();
    ps.insert(ps.end(), a.begin(), a.end());
  }

  ps = project_clip_pd(ps, pmXem); // performs projection, clipping, and perspective division in parallel

  // nobackfaces
  if (nobackfaces)
    ps.erase(std::remove_if(std::execution::par_unseq, ps.begin(), ps.end(),
                            [](const auto& a) { return cross(a[1] - a[0], a[2] - a[1])[2] >= 0; }),
             ps.end());

  glClearColor(0.0, 0.0, 0.0, 0.0);
  glClear(GL_COLOR_BUFFER_BIT);
  const auto [vxl, vxr, vyb, vyt] = vp.get_borders();
  draw_polygon(Polygon_u<2>{{{vxl, vyb}, {vxr, vyb}, {vxr, vyt}, {vxl, vyt}}}); // this function will do the flushing
  draw_polygons(translation_m(vxl, vyb) * scaling_m((vxr - vxl) / 2.0, (vyt - vyb) / 2.0) * translation_m(1.0, 1.0) * ps);

  auto t1 = std::chrono::high_resolution_clock::now();
  std::cout << "display takes: " << std::chrono::duration_cast<std::chrono::milliseconds>(t1 - t0).count() << "ms\n";
  system("pause");
}
```

Requirement

1. the latest Visual Studio 2017
2. the latest Windows SDK
3. C++17  (structured binding, `std::string_view`, parallel algorithms)

.clang-format settings

```
BasedOnStyle: LLVM
PointerAlignment: Left
ColumnLimit: 0
SpaceAfterTemplateKeyword: false
```

.editorconfig settings

```
root = true

[*]
indent_style = space
indent_size = 2
end_of_line = lf
charset = utf-8
trim_trailing_whitespace = true
insert_final_newline = true

[*.md]
trim_trailing_whitespace = false
```