

Introduction to Computer Science

Final Project

0316081 王威斌

0316218 董乃嘉

0316083 陳逸群

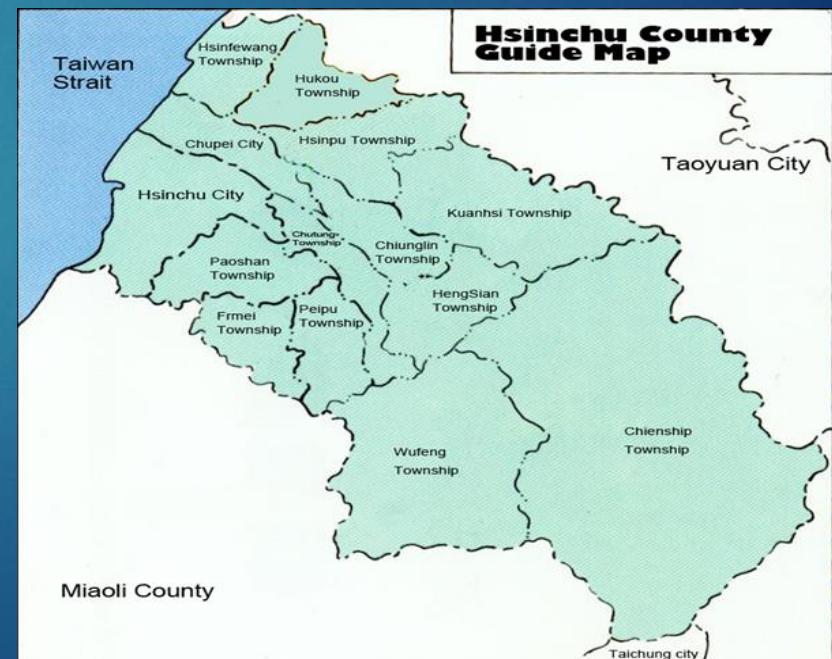
0316095 蔡健怡

0316098 李宗霖

0316327 吳俊賢

0316339 陳霆峰

0316094 甘敬軒



Work Assignment

- ▶ To complete this final project, we divide our group into different team , including Team Query , Team Binary ,Team Database , Team API and Team GUI .

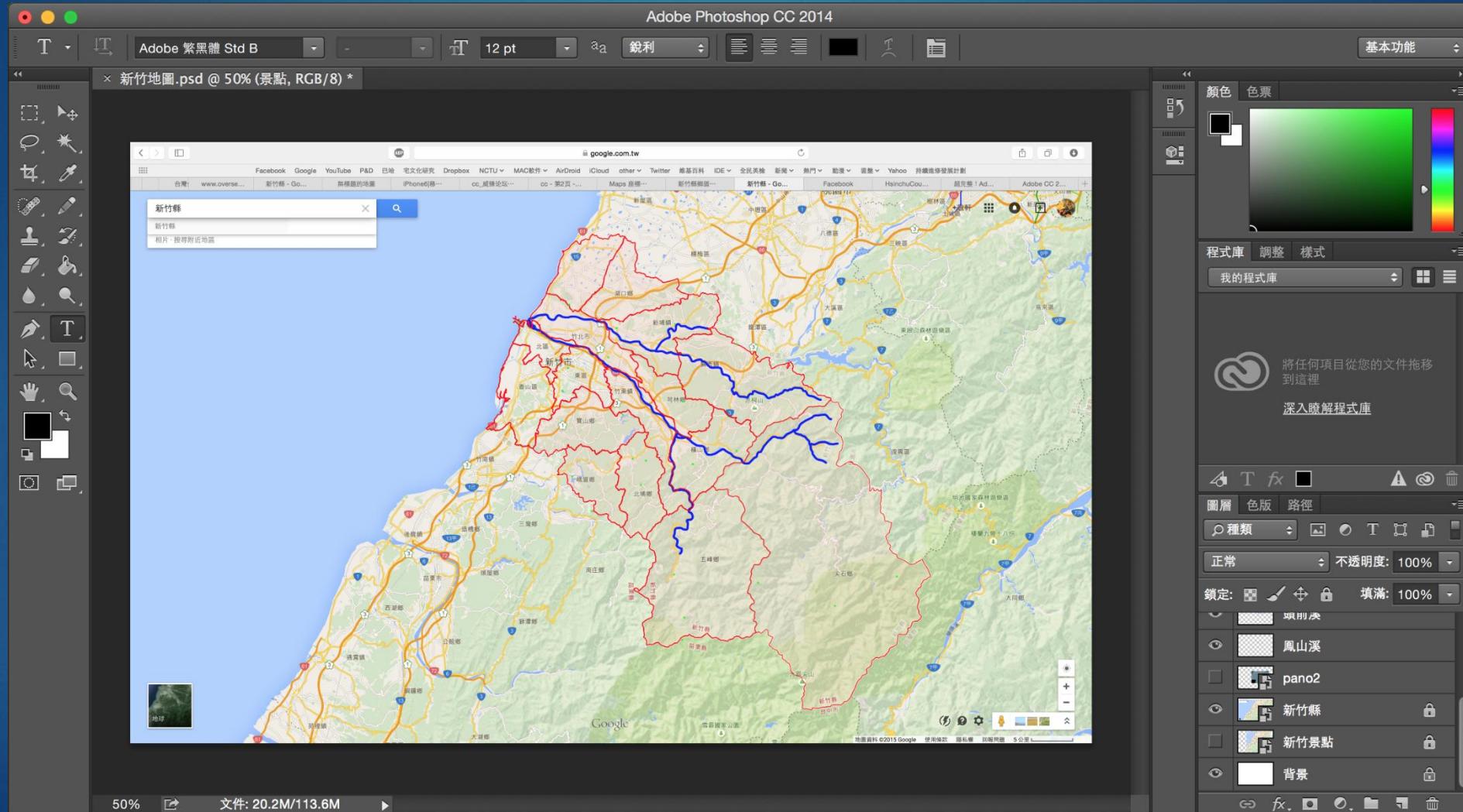
Team Binary

(0316094 甘敬軒)

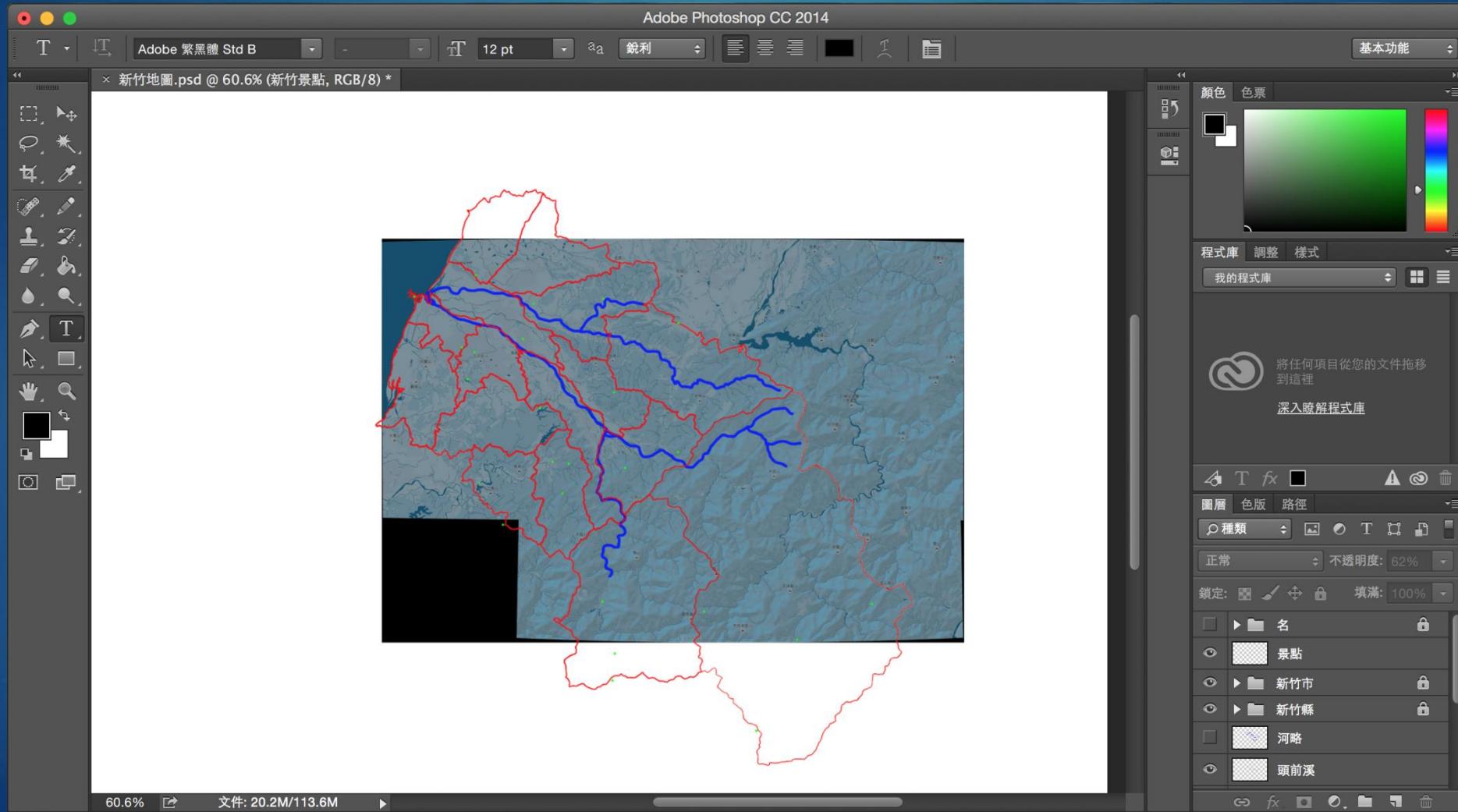
(0316095 蔡健怡)

- The main purpose is to find maps and convert it to the coordinates, which we can use in the latter time, into a file.

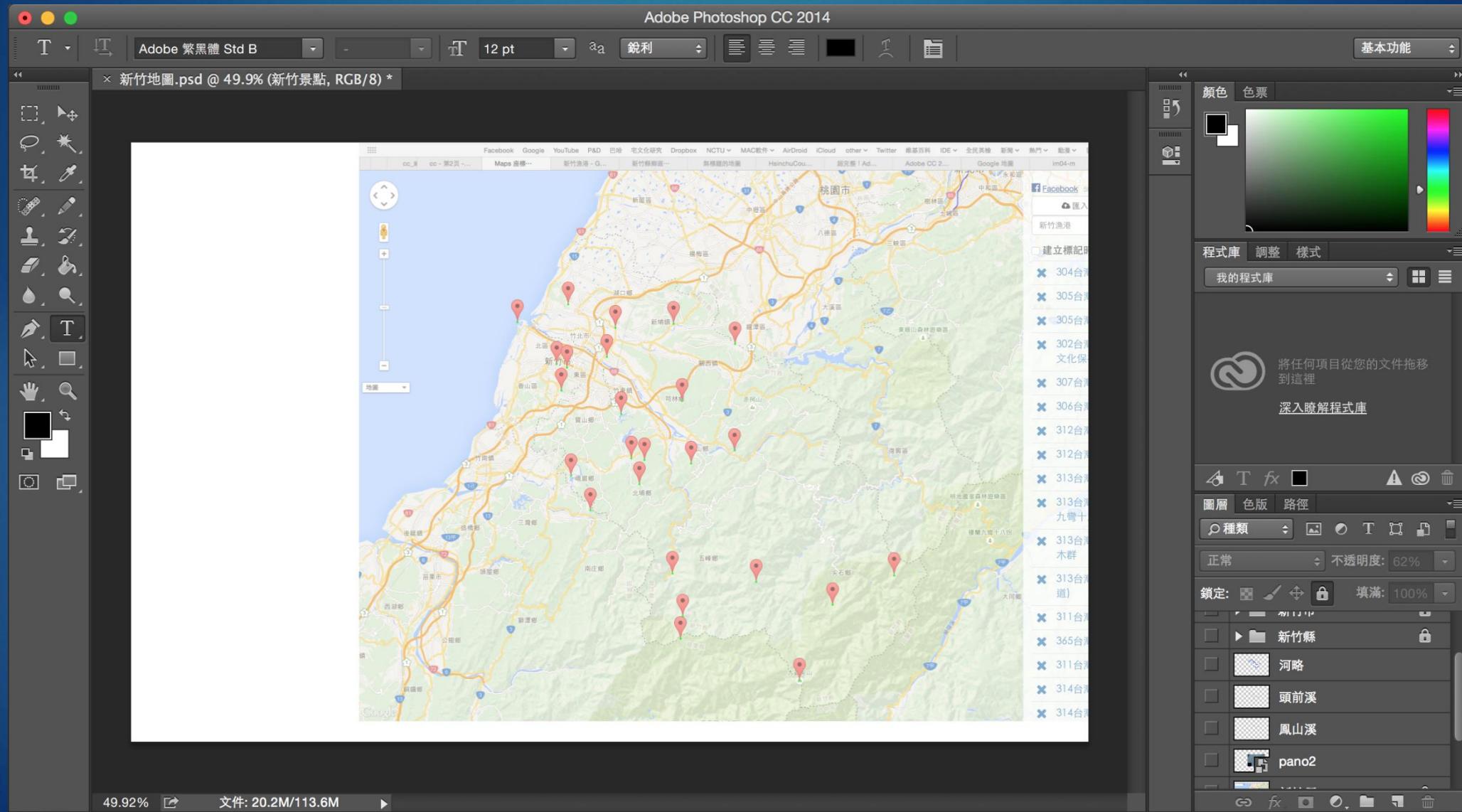
The First Step → Photoshop



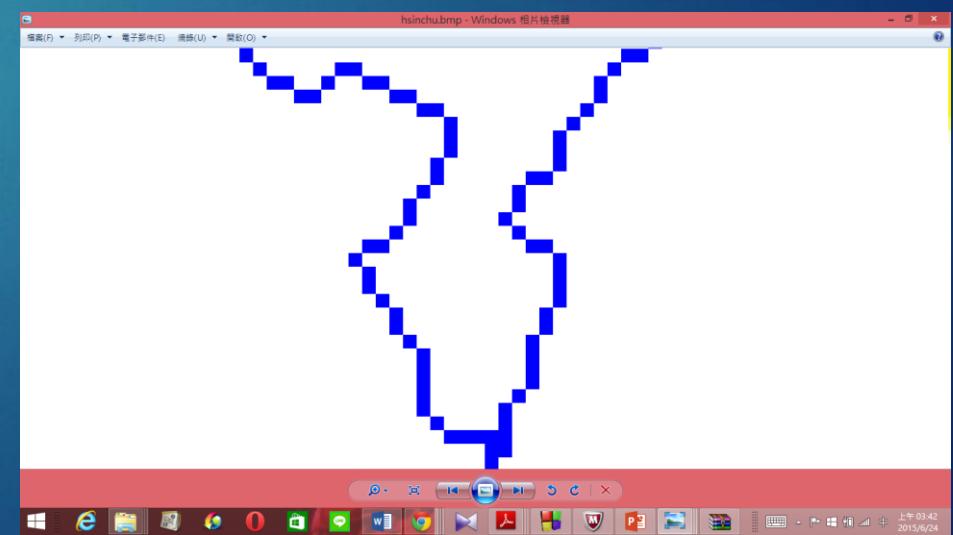
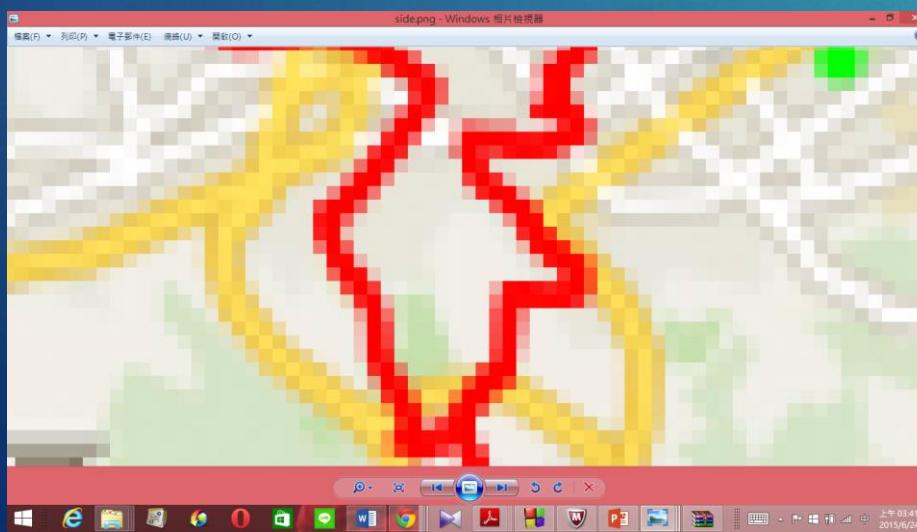
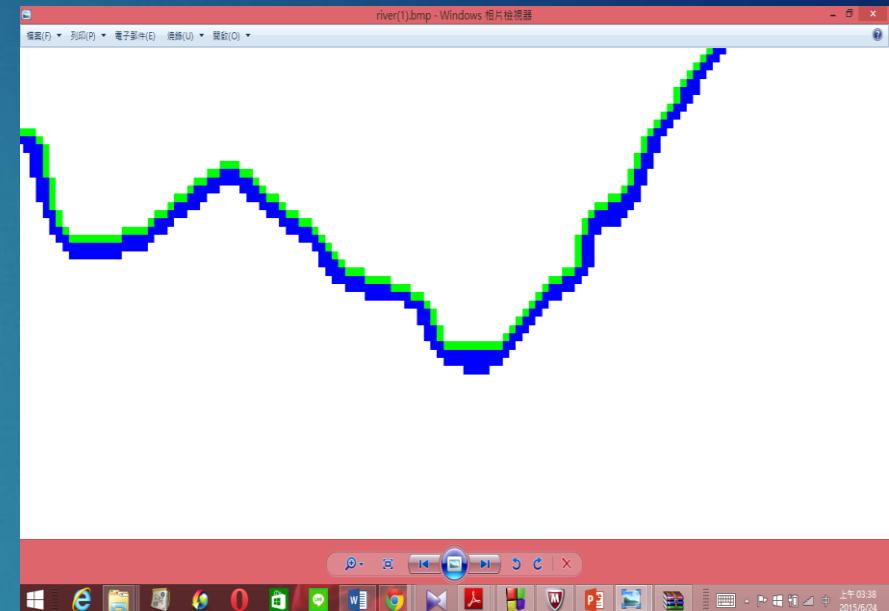
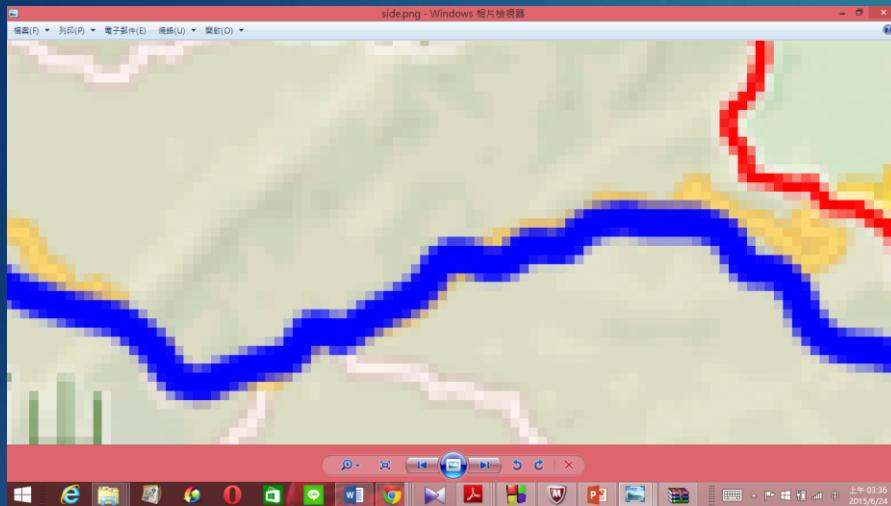
The First Step → Photoshop



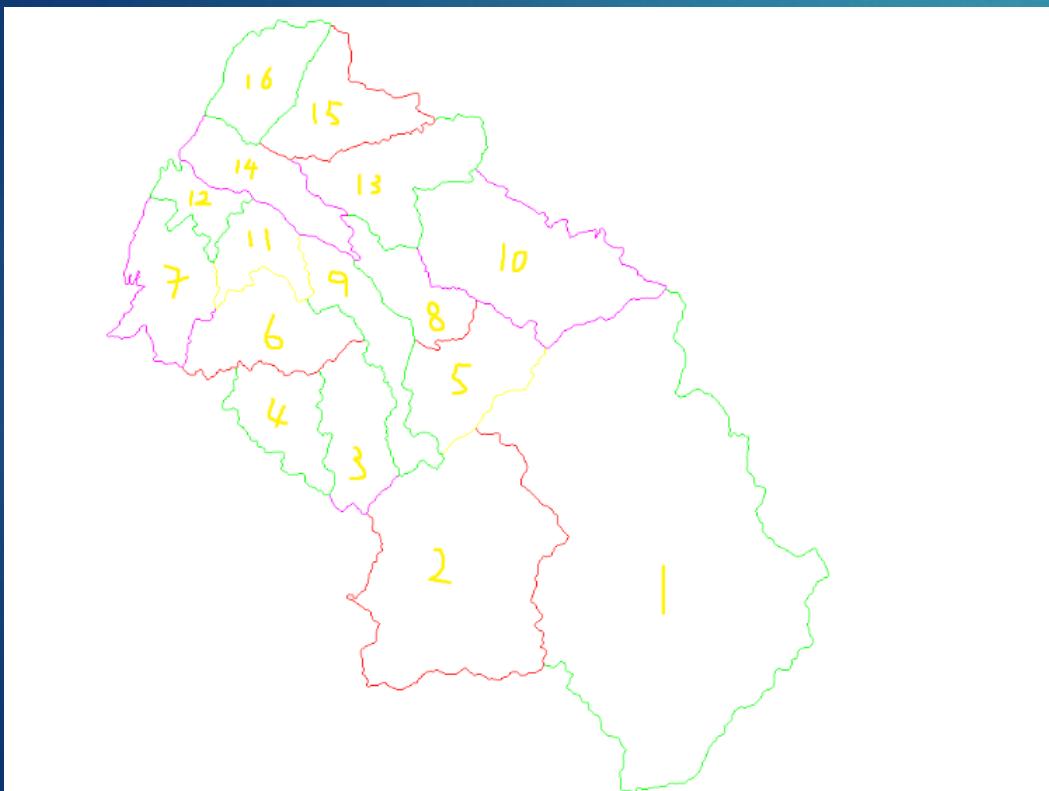
The First Step → Photoshop



The Second Step -----Program



Simply the map and decide the coordinate-----City



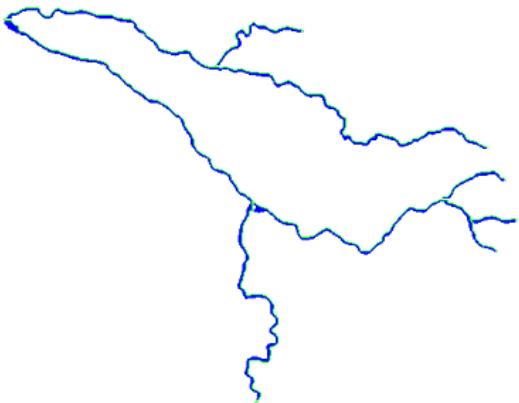
hsinchu.txt - 記事本

```
(x,y)
##1
(611,764)
(612,764)
(613,764)
(614,763)
(614,762)
(614,761)
(615,760)
(616,760)
(617,760)
(618,760)
(619,761)
(620,761)
(621,761)
(622,761)
(623,761)
(624,761)
(625,761)
(626,760)
(627,760)
(628,760)
(629,760)
(630,759)
(631,759)
(632,759)
(633,759)
(634,759)
(635,759)
(636,759)
(637,758)
(638,758)
(639,758)
(640,758)
(641,758)
(642,758)
(643,758)
(644,757)
(645,756)
```

第1列, 第1行

上午 03:48
2015/6/24

Simply the map and decide the coordinate-----River

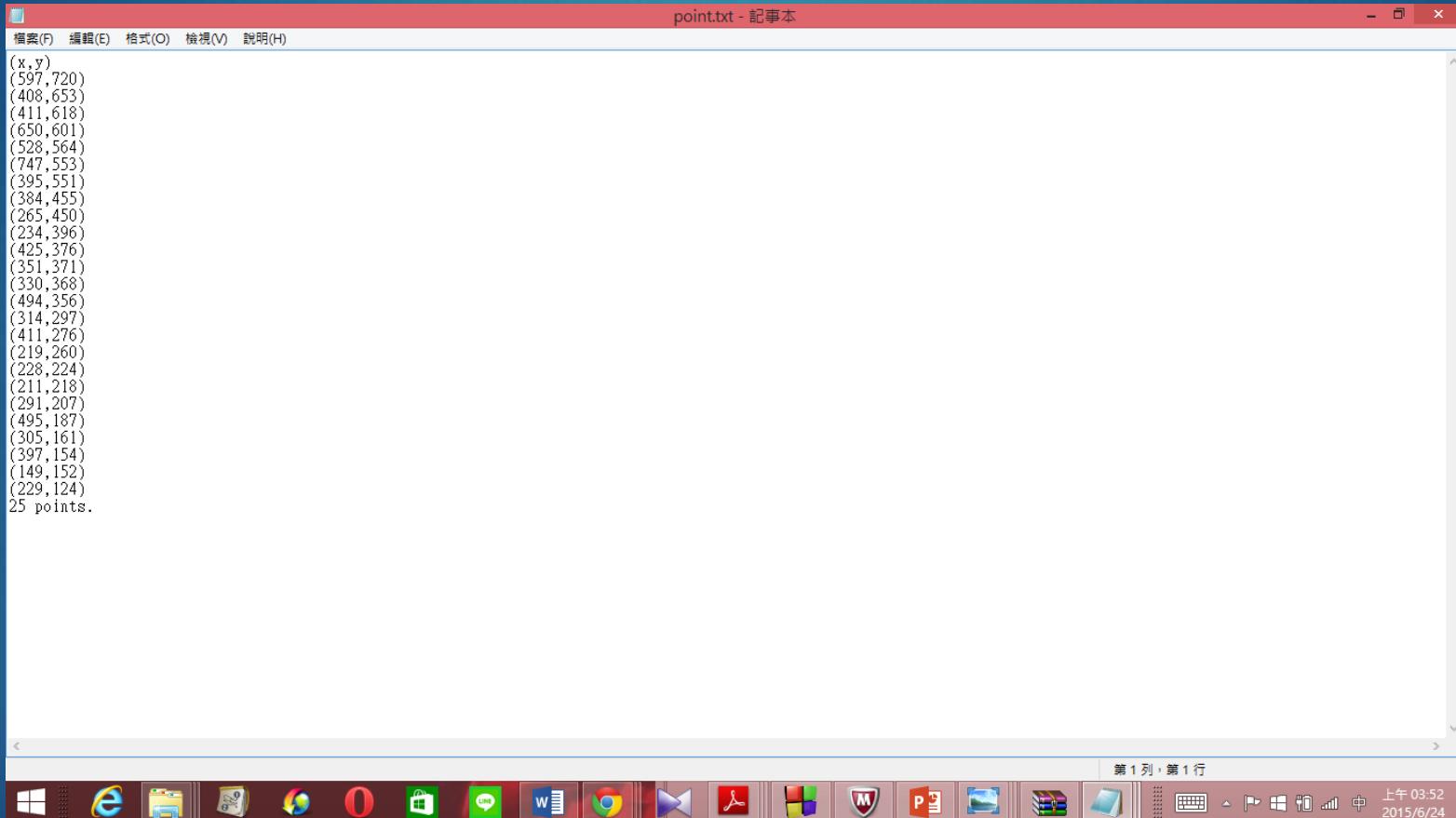


```
river.txt - 記事本
(162,150)
(tturning point up)
(163,150)
(164,149)
(165,148)
(166,147)
(166,146)
(167,146)
(168,145)
(169,144)
(170,144)
(171,144)
(172,144)
(173,144)
(174,144)
(175,143)
(176,143)
(177,142)
(178,141)
(179,141)
(180,140)
(181,140)
(182,140)
(183,139)
(184,139)
(185,139)
(186,139)
(187,139)
(188,139)
(189,139)
(190,139)
(191,139)
(192,139)
(193,139)
(194,139)
(195,139)
(196,140)
(197,141)
(197,142)

第1列,第1行
2015/6/24 上午 03:50
```

A screenshot of a Windows desktop showing a Notepad window titled "river.txt - 記事本". The window contains a list of coordinate pairs, likely representing the vertices of the river line shown in the image above. The coordinates are listed in a vertical column, starting at (162, 150) and ending at (197, 142). The Notepad window has a standard red title bar and a menu bar with Chinese options: 帮助(F), 插入(I), 格式(O), 检视(V), and 脱机(H). The desktop background is dark blue, and the taskbar at the bottom shows various pinned icons for Microsoft applications like Internet Explorer, File Explorer, and Word.

Simply the map and decide the coordinate-----Attractions



Team Database (0316339 陳霆峰)

- ▶ Create the table according to condition which TA gives.
- ▶ Gets the data and coordinates from Team Binary, and provide them for Team Query and Team GUI.

City Table

City.txt - 記事本

```
福案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
```

Create table `0x0D` (`x-axis` int, `y-axis` int); Insert into `0x0D` values (611,764), (612,764), (613,764), (614,763), (614,762), (614,761), (615,760), (616,760) ^ , (683,724), (684,723), (684,722), (684,721), (685,720), (686,719), (687,718), (688,716), (689,715), (690,715), (691,714), (692,713), (692,712), (692,711), (748,653), (748,652), (748,651), (748,650), (748,649), (749,648), (749,647), (749,646), (750,645), (751,644), (752,644), (753,644), (754,644), (755,645), (756,645), (757 79,582), (778,581), (778,580), (778,579), (778,578), (779,577), (780,576), (780,575), (781,574), (781,573), (782,572), (783,571), (784,571), (785,570), (786,569), (786, 6,526), (755,526), (754,526), (753,525), (752,525), (751,524), (750,524), (749,523), (748,522), (747,521), (747,520), (747,519), (746,518), (745,517), (744,517), (743,5 ,448), (722,447), (722,446), (722,445), (722,444), (722,443), (722,442), (722,441), (722,440), (721,439), (721,438), (721,437), (722,436), (722,435), (722,434), (722,43 376), (682,376), (681,376), (680,376), (679,375), (678,374), (677,373), (676,373), (675,372), (674,372), (673,372), (672,372), (671,373), (670,373), (669,373), (668,374 16), (653,315), (653,314), (653,313), (653,312), (654,311), (654,310), (654,309), (655,308), (655,307), (656,306), (657,305), (658,304), (659,303), (659,302), (659,301) 2), (601,293), (600,294), (599,295), (598,296), (597,297), (596,297), (595,298), (594,299), (593,300), (593,301), (592,302), (591,303), (590,303), (589,303), (588,304),), (519,345), (519,346), (519,347), (518,348), (517,348), (516,349), (515,349), (514,350), (513,351), (512,352), (511,353), (511,354), (511,355), (512,356), (513,356), , (458,410), (459,411), (459,412), (460,413), (461,414), (462,415), (463,416), (464,416), (465,416), (466,416), (467,415), (468,416), (469,416), (470,416), (471,416), (510,471), (511,472), (512,473), (513,474), (514,475), (515,476), (516,477), (517,478), (518,479), (519,480), (520,481), (521,482), (522,484), (523,484), (524 18,546), (518,547), (518,548), (518,549), (518,550), (518,551), (518,552), (517,553), (517,554), (516,555), (516,556), (515,557), (516,558), (517,559), (517,560), (517, 4,633), (524,634), (523,635), (523,636), (523,637), (524,638), (524,639), (525,640), (526,641), (527,641), (528,641), (529,641), (530,641), (531,642), (532,643), (533,6 ,690), (578,691), (578,692), (579,693), (579,694), (579,695), (579,696), (580,697), (580,698), (580,699), (580,700), (580,701), (580,702), (580,703), (581,704), (582,70 0x0F` (`x-axis` int, `y-axis` int); Insert into `0x0F` values (381,664), (382,664), (383,664), (384,664), (385,664), (386,664), (387,664), (388,663), (389,6 464,650), (465,651), (466,651), (467,652), (468,652), (469,653), (470,654), (471,654), (472,654), (473,653), (474,653), (475,654), (476,654), (477,654), (478,655), (479 23,613), (524,612), (524,611), (524,610), (524,609), (523,608), (522,607), (521,607), (520,607), (519,606), (518,605), (517,604), (516,603), (515,602), (515,601), (516, 2,531), (533,530), (534,529), (534,528), (534,527), (535,526), (536,525), (537,524), (538,524), (540,523), (541,523), (542,523), (543,522), (544,522), (545,5 ,451), (507,450), (508,449), (508,448), (508,447), (507,446), (507,445), (506,443), (506,442), (505,441), (505,440), (505,439), (505,438), (504,437), (503,43 423), (435,424), (434,425), (433,426), (433,427), (432,428), (431,429), (430,430), (430,431), (429,432), (428,433), (427,434), (426,435), (426,436), (426,437), (426,438 76), (361,477), (361,478), (360,479), (360,480), (359,481), (358,482), (357,482), (356,483), (355,484), (354,485), (354,486), (354,487), (353,488), (353,489), (353,490 1), (352,562), (352,563), (351,564), (351,565), (351,566), (351,567), (349,568), (348,569), (347,569), (346,570), (345,571), (344,572), (343,573), (342,574), (349,624), (348,624), (347,625), (346,626), (346,627), (346,628), (346,629), (347,631), (347,632), (347,633), (348,634), (348,635), (348,636), (348,637), (360,480), (360,479), (361,478), (361,477), (362,476), (363,475), (364,475), (365,474), (365,473), (366,472), (367,471), (368,470), (369,470), (370,469), (371,46 395), (375,394), (376,393), (377,392), (378,391), (379,390), (380,390), (381,389), (381,388), (381,387), (381,386), (381,385), (381,384), (381,383), (380,382), (379,381 26), (346,326), (345,326), (344,326), (343,326), (342,326), (341,325), (340,325), (339,325), (338,325), (337,326), (336,326), (335,327), (334,328), (333,329), (332,330) 7), (308,398), (308,399), (308,400), (308,401), (308,402), (307,403), (307,404), (306,405), (306,406), (306,407), (306,408), (305,409), (304,410), (304,411), (304,412), (316,476), (316,477), (317,478), (318,479), (318,480), (318,481), (319,482), (319,483), (320,484), (320,485), (321,486), (322,487), (323,488), (324,488), (325,489), (441), (310,440), (311,439), (312,438), (312,437), (312,436), (313,435), (314,434), (314,433), (314,432), (314,431), (314,430), (314,429), (315,428), (315,427), (315,42 362), (310,361), (309,360), (308,359), (307,358), (307,357), (307,356), (306,355), (305,355), (304,355), (303,356), (302,356), (301,356), (300,357), (299,357), (298,357 46), (227,347), (227,348), (227,349), (227,350), (226,351), (225,352), (225,353), (225,354), (225,355), (225,356), (225,357), (225,358), (226,359), (227,360), (227,361) 8), (239,419), (239,420), (239,421), (240,422), (241,422), (242,423), (243,423), (244,422), (245,422), (246,423), (246,424), (246,425), (246,426), (247,427), (247,428), (298,467), (299,467), (300,468), (301,468), (302,468), (303,469), (304,470), (305,471), (306,471), (307,471), (308,471), (310,473), (310,474), (311,475); (379,478), (379,479), (479,378), (480,378), (481,378), (482,378), (483,378), (484,377), (485,376), (486,376), (487,375), (488,375), (489,374), (490,374), (491,374), (492,37 314), (515,313), (515,312), (516,311), (517,310), (517,309), (517,308), (516,307), (515,307), (514,306), (513,306), (512,305), (511,305), (510,306), (509,306), (508,307 10), (451,311), (451,312), (452,313), (452,314), (452,315), (451,316), (450,317), (450,318), (449,319), (448,319), (447,320), (447,321), (446,322), (445,323), (444,323) 6), (391,347), (391,348), (391,349), (390,350), (390,351), (390,352), (389,354), (389,355), (389,356), (389,357), (389,358), (389,359), (389,360), (389,361), (412,415), (413,415), (414,416), (415,417), (416,417), (417,418), (418,418), (419,419), (420,420), (420,421), (420,422), (421,423), (421,424), (422,425), (422,426), (345), (249,344), (250,345), (250,346), (251,346), (251,347), (252,348), (253,347), (254,347), (255,347), (256,348), (257,348), (258,348), (259,348), (260,349), (261,349), (262,34

第 1 列, 第 1 行

2015/6/24 上午 04:00



River Table

Table

```
sql.txt - 記事本  
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)  
Create table CITY ( `city-id` varchar(4), `city-name` varchar(10), population int, `area-size` float(8,4) ); Insert into CITY values ('0x00','無',0^  
Township), ('0x11','Zhudong Township'), ('0x11','Zhubei City'), ('0x11','East District'), ('0x11','North District'), ('0x12','Jianshi Township'), ('0x12','Guanxi Towns  
,330,368), ('0x25','獅頭山','峨眉鄉',265,450), ('0x26','峨眉湖','峨眉鄉',234,396), ('0x27','寶山水庫','寶山鄉',314,297), ('0x28','新竹都城隍廟','北區',211,218), ('0x29'  
'Beipu Township','Baoshan Township'), ('Beipu Township','Zhudong Township'), ('Beipu Township','Wufeng Township'), ('Zhudong Township','Wufeng Township'), ('Zhudong  
Xiangshan District','North District'), ('Xiangshan District','East District'), ('Xiangshan District','Baoshan Township'), ('East District','Xiangshan District'), ('East
```

Team Query

0316081 王威斌

0316083 陳逸群

0316098 李宗霖

0316327 吳俊賢

- ▶ Team Query, the main purpose is connecting with Team database and Team GUI, and the tool which Team Query use is C program.

Team Query

► When we make command in GUI,

Team Query should program to pass this command to Database, and Database return data which we want..

Team Query

- ▶ When Database return the data, Team Binary should transform it to the form what Team GUI can use to show the result.

Team Query-----Purpose

- ▶ Store the coordinates in the structure, and then use dynamic array to link every point.
- ▶ We use string to read the Sequel command and link database.

終端機

```
select `city-id` from RiverPass where `river-id`='0x11'
executed!!!
0x0D .....
0x0F .....
0x08 .....
0x0E .....
0x07 .....
0x04 .....
0x01 .....
0x02 .....
*****
Process returned 0 (0x0) execution time : 2.948 s
Press ENTER to continue.
```

m_fields(res);++t)

```
57 // !!!!!!!
58 // !!!!!!!
59 // !!!!!!!
60 // close connection
61 mysql_free_result(res);
62 mysql_close(conn);
63 }
64 }
```

Logs & others

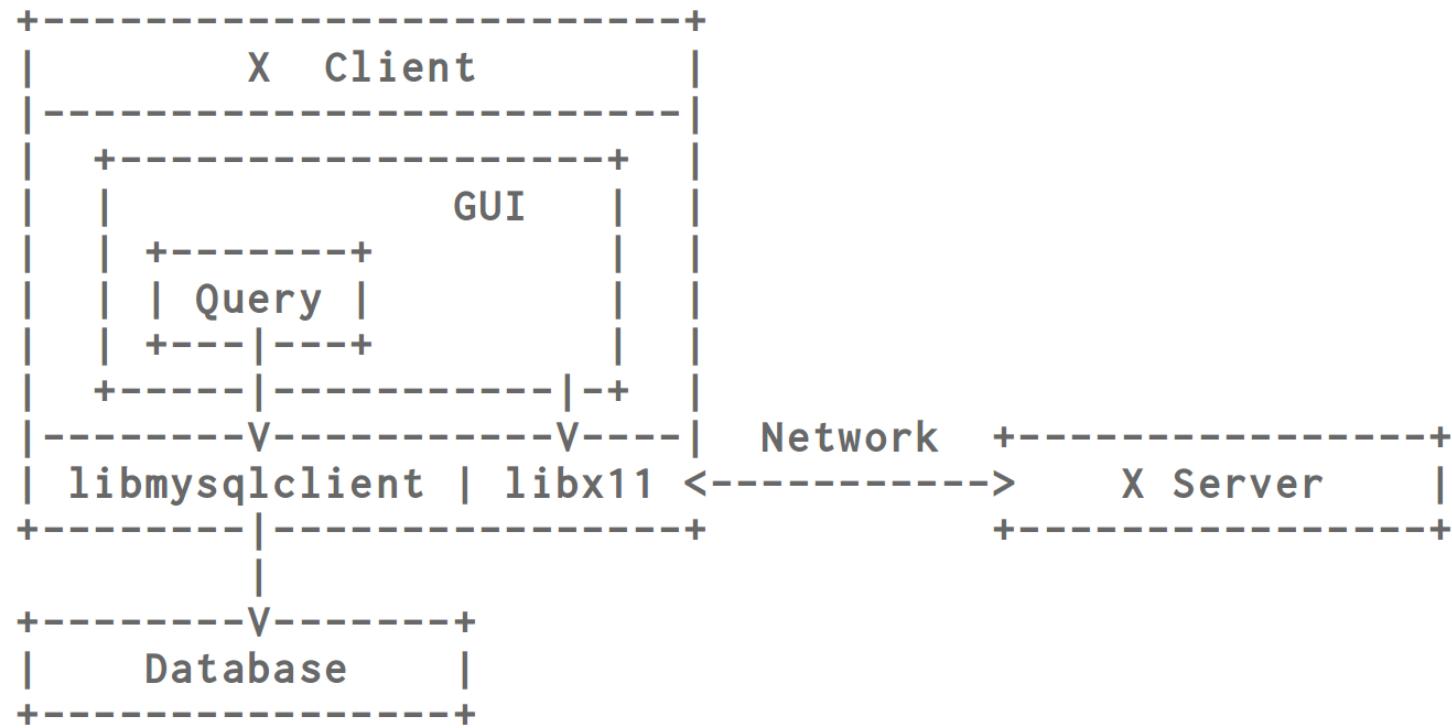
Code::Blocks Search results Debugger Build log Build messages

File	Line	Message
MYSQL_TEST.c	In function `main`:	
MYSQL_TEST.c	47	警告：建議在使用直接的賦值語句前加上括號 (-Parentheses) ***Build finished: 1 errors, 0 warnings***

螢幕擷圖存為 2015-06-09 20:46:25.png
Added Jun 9 at 8:48 PM - 147.09 KB

初期分工

B. LIBRARIES DEPENDENCY



How communicate



► Trello.com

最終分工

- | | | |
|---------------|-------------|---------------------|
| ► 0316081 王威斌 | 0316083 陳逸群 | (Query programming) |
| ► 0316339 陳霆峰 | | (Database) |
| ► 0316095 蔡健怡 | 0316094 甘敬軒 | (Binary Team) |
| ► 0316218 董乃嘉 | | (GUI Team) |
| ► 0316327 吳俊賢 | | (Report Team) |
| ► 0316098 李宗霖 | | (Command (Sequel)) |

Hsinchu-Map Project

Team GUI

0316218 董乃嘉

What is X Window System?



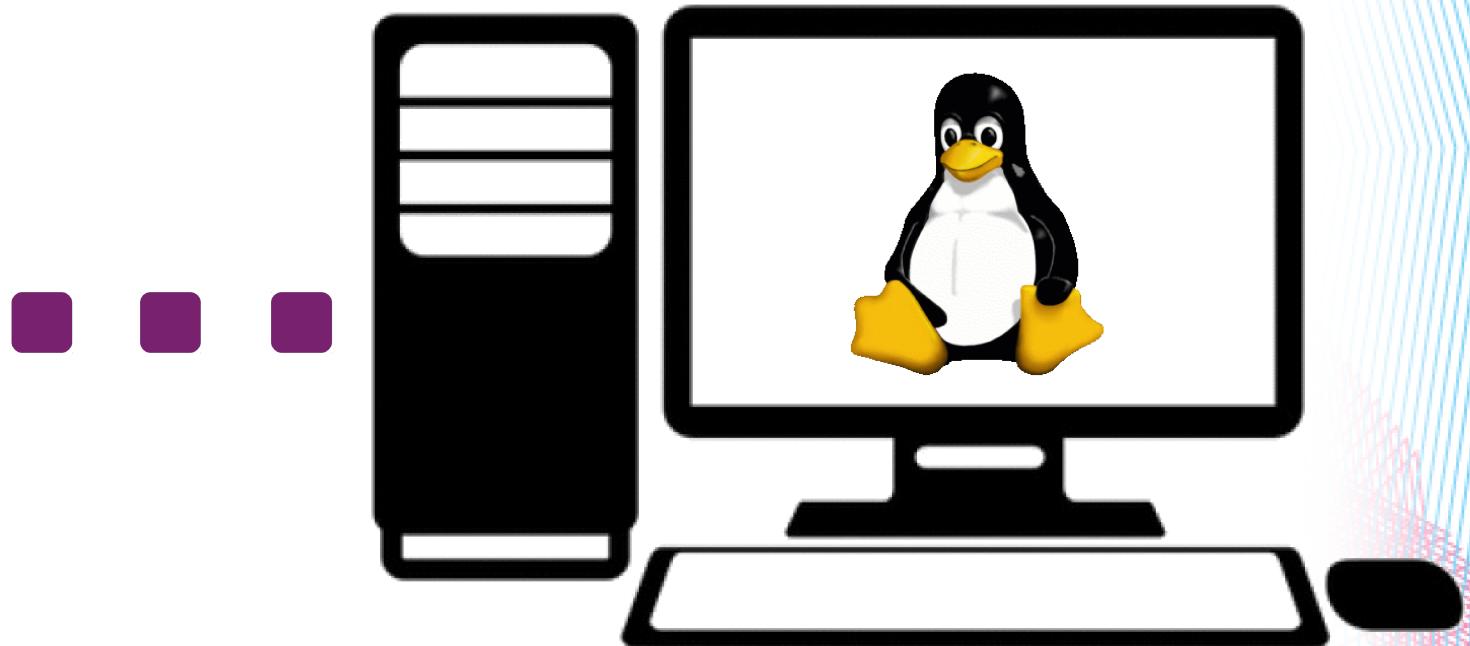
- The X Window System (f.k.a. X) is a windowing system for bitmap displays, common on UNIX-like computer operating systems.
- X provides the basic framework for a GUI environment: editing windows on the display and interacting with a mouse and keyboard.

What is Xlib?

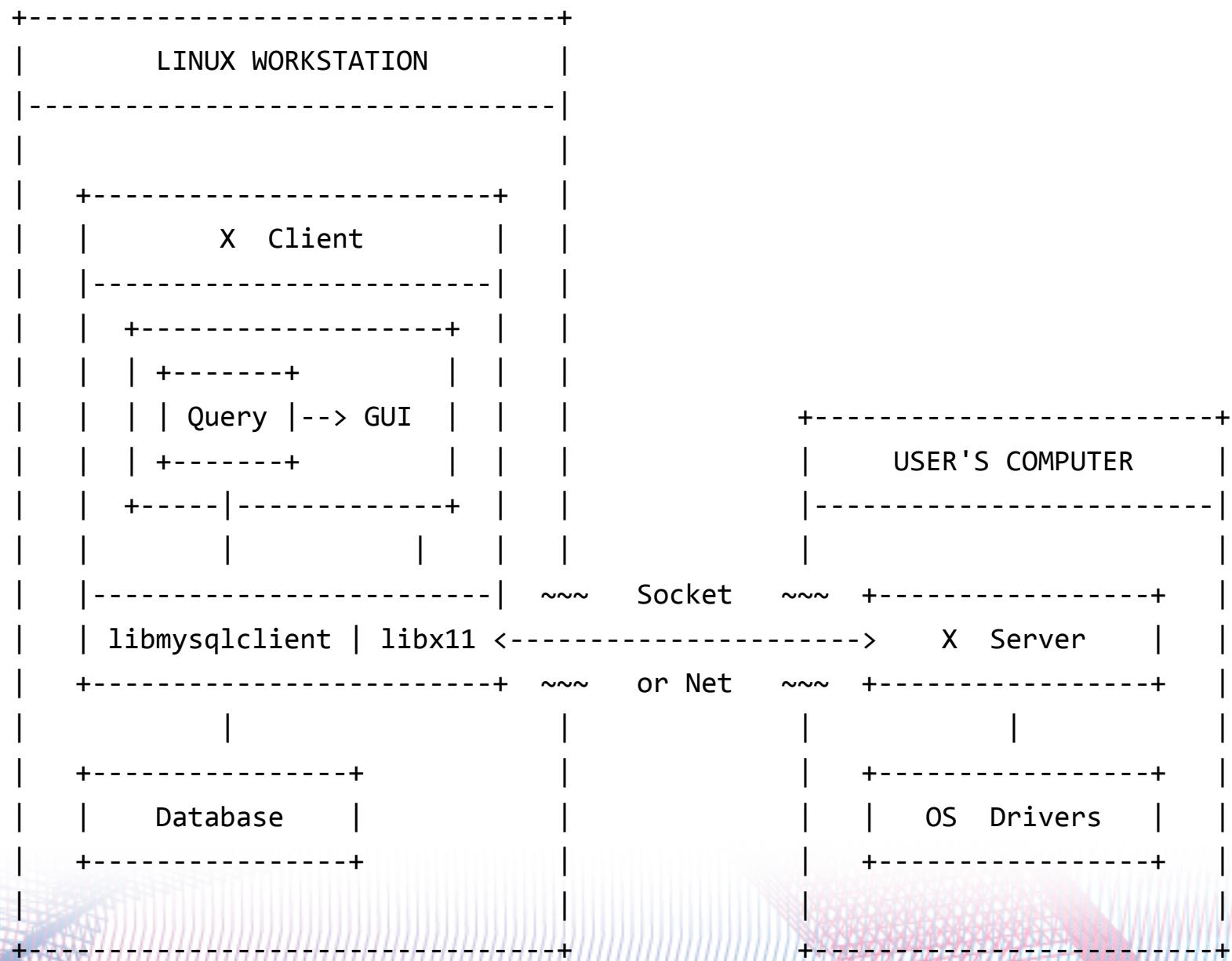


- Xlib (a.k.a. libX11) is an X Window System protocol client library that is written in the C programming language.
- Xlib appeared around 1985 (~30 years ago), and it has developed and maintained by X.org Foundation since Jan 22, 2004.

Project Architecture

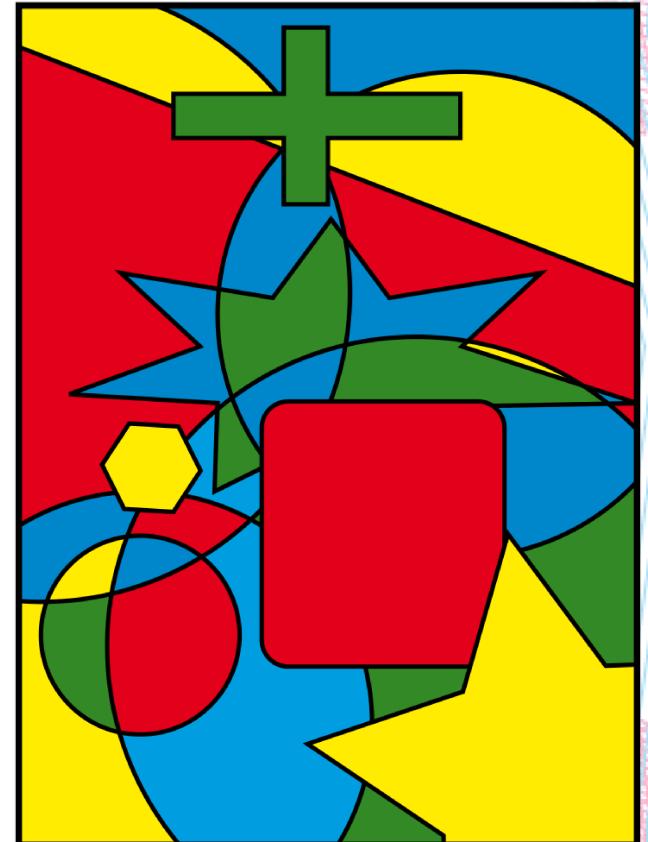


Project Architecture



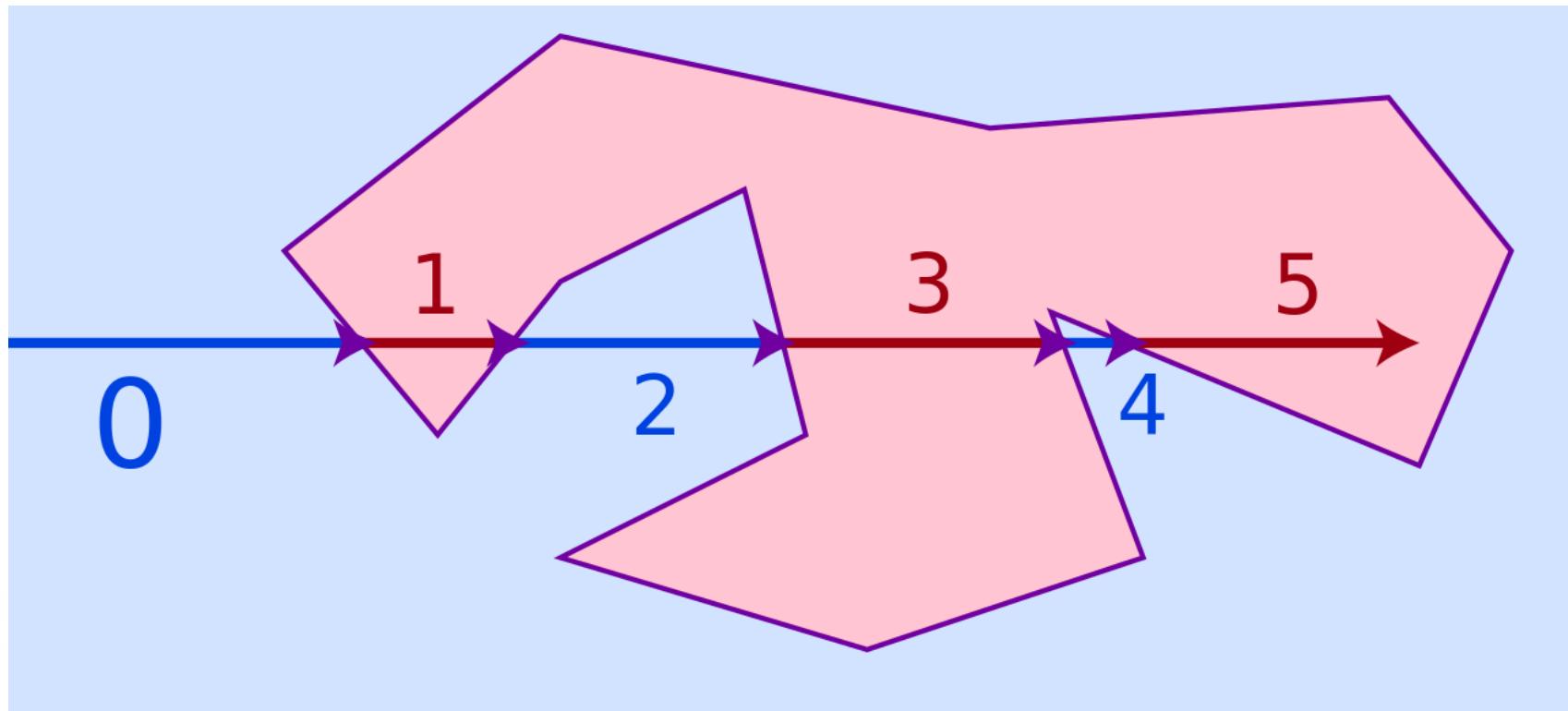
Four Color Theorem

- How many colors are required to draw every district such that the two adjacent ones are drawn in different colors?
- To decrease the stack size allocated for graphic contexts, less colors used is preferred.



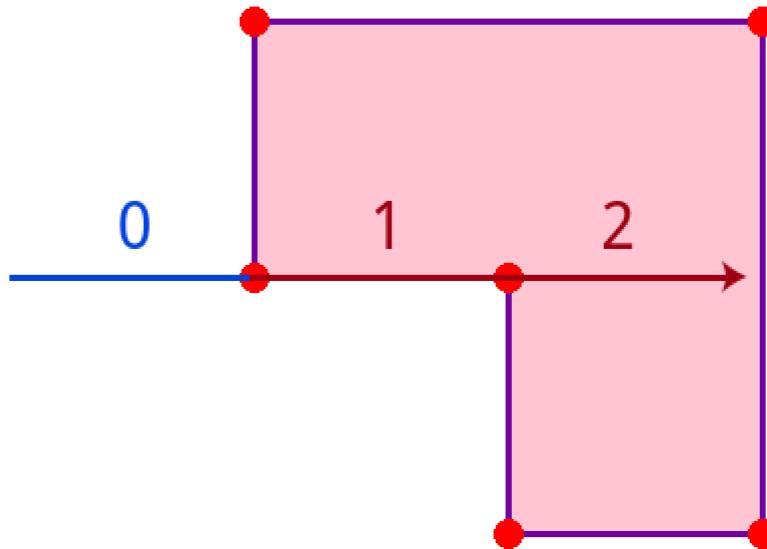
Point in Polygon Algorithm

- Ray casting algorithm:



Point in Polygon Algorithm

- The concept above does not exactly fit the computer algorithm.
- For example:



Point in Polygon Algorithm

- To avoid this problem, I tried to change the slope of the ray into an irrational number, π .
- Thus, this guarantee the ray passes through none of the sampling pixels if every coordinate is an integer.

$$y - y_0 = m(x - x_0)$$

$$m \in \overline{\mathbb{Q}}, \quad x_0, y_0 \in \mathbb{Z}$$

$$s.t. \quad x \in \mathbb{Z} \setminus \{x_0\} \Rightarrow y \notin \mathbb{Z}$$

Point in Polygon Algorithm

```
for (int cnt_1 = DISTRICT_START; cnt_1 <= DISTRICT_END; cnt_1++)
{
    int cnt_pos = 0, cnt_neg = 0;

    for (int cnt_2 = 0; coord[cnt_1][cnt_2 + 1] != NULL; cnt_2++)
    {
        double x_0 = x,
               y_0 = y,
               x_1 = coord[cnt_1][cnt_2]->x,
               y_1 = coord[cnt_1][cnt_2]->y,
               x_2 = coord[cnt_1][cnt_2 + 1]->x,
               y_2 = coord[cnt_1][cnt_2 + 1]->y;
        double b_1 = (x_1 != x_2) ? 1.0 : 0.0;
        double m_0 = MATH_CONST_PI,
               m_1 = (b_1 == 1.0) ? ((y_2 - y_1) / (x_2 - x_1)) : 1.0;

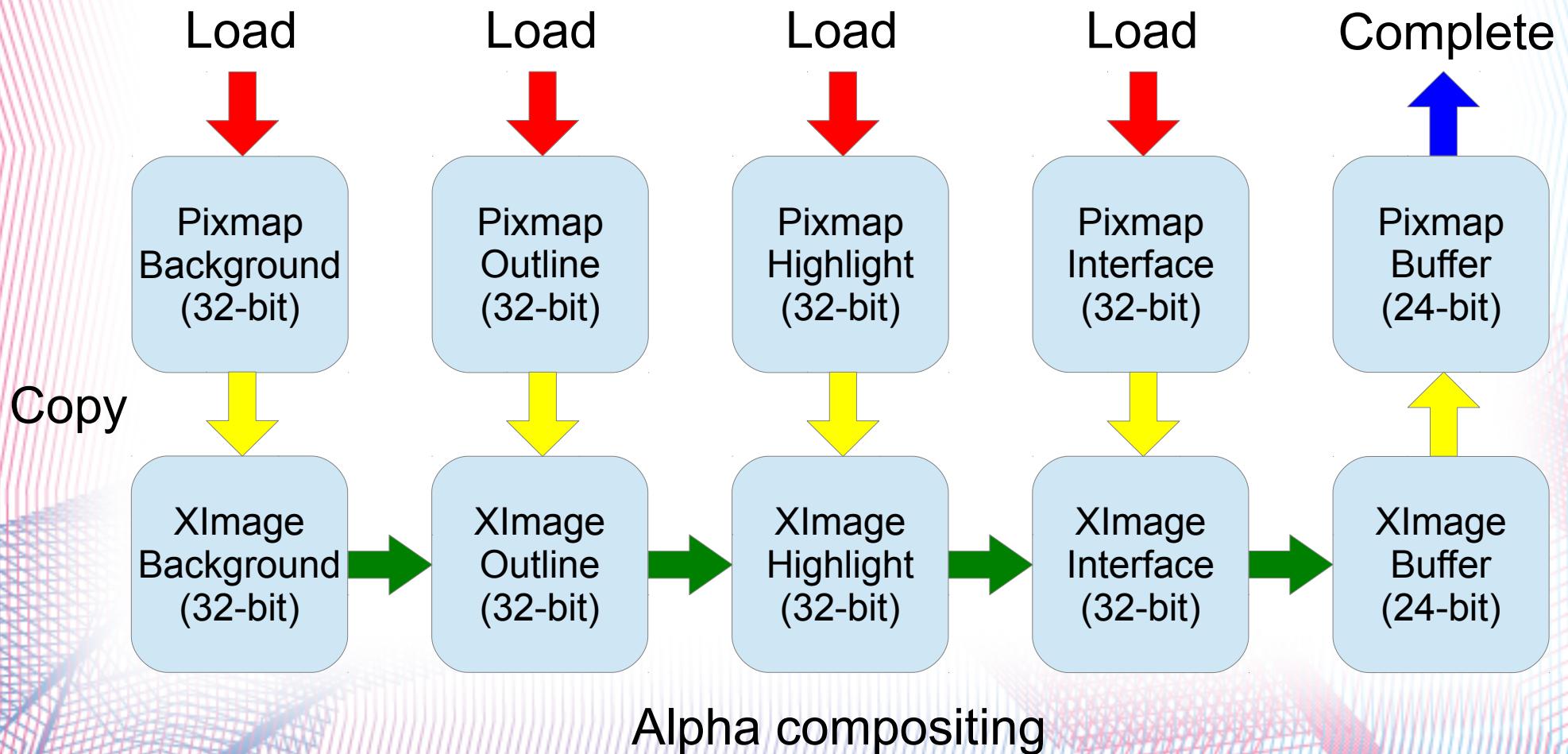
        double u = ((y_0 - y_1) - m_0 * (x_0 - x_1)) / (m_1 - m_0 * b_1),
               v = x_1 + b_1 * u - x_0;

        if (((b_1 == 1.0) && (u * (u - x_2 + x_1) <= 0)) ||
            ((b_1 == 0.0) && (u * (u - y_2 + y_1) <= 0)))
            (v >= 0.0) ? cnt_pos++ : cnt_neg++;
    }

    if (cnt_neg != 0 && cnt_pos % 2 == 1)
        (*id)[0] = cnt_1;
}
```

Frame Buffering

- If you draw everything directly on the window, then it will be very inefficient in X.



Alpha Compositing

- Assume that the background image is opaque, and every image on it contains alpha channel.
- The original formula to perform **over** operation:

$$C_o = C_a\alpha_a + C_b\alpha_b(1 - \alpha_a)$$

- Regard every newly blended layer as opaque:

$$\Rightarrow C_o = C_a\alpha_a + C_b(1 - \alpha_a)$$

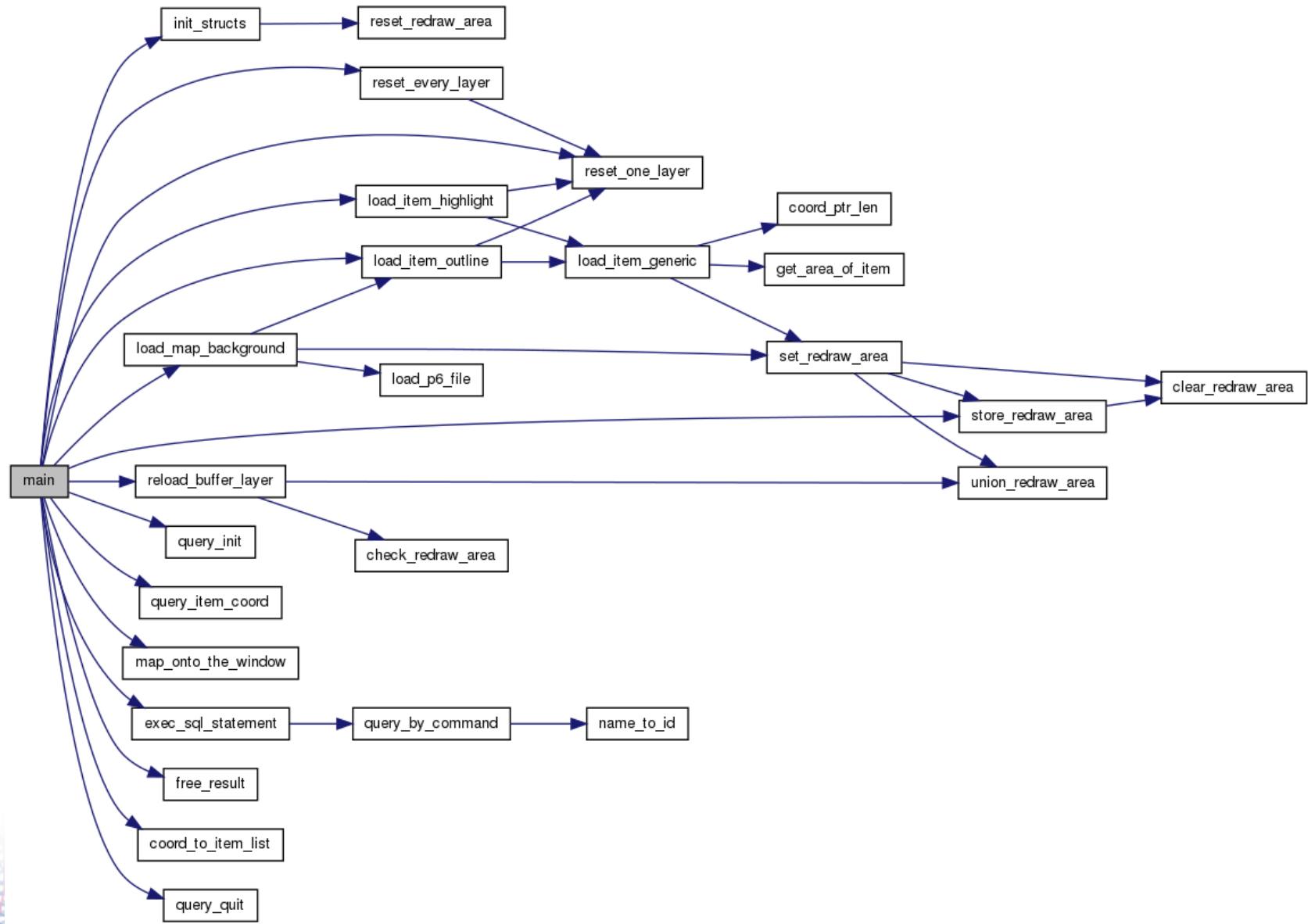
```
/* Alpha blending for 32-bit XImage */
for (unsigned int cnt_x = 0; cnt_x < width; cnt_x++)
    for (unsigned int cnt_y = 0; cnt_y < height; cnt_y++)
{
    for (int pxl_ofs = 0; pxl_ofs < WIN_BYTES_PER_PIXEL; pxl_ofs++)
    {
        int rel_ofs = pxl_ofs + XPM_BYTES_PER_PIXEL *
                      (cnt_x + cnt_y * width);
        int abs_ofs = pxl_ofs + XPM_BYTES_PER_PIXEL *
                      ((x + cnt_x) + (y + cnt_y) * XPM_MAX_W);

        layers->xmg[LAYER_BUFFER]->data[abs_ofs] =
        layers->xmg[LAYER_BOTTOM]->data[abs_ofs];

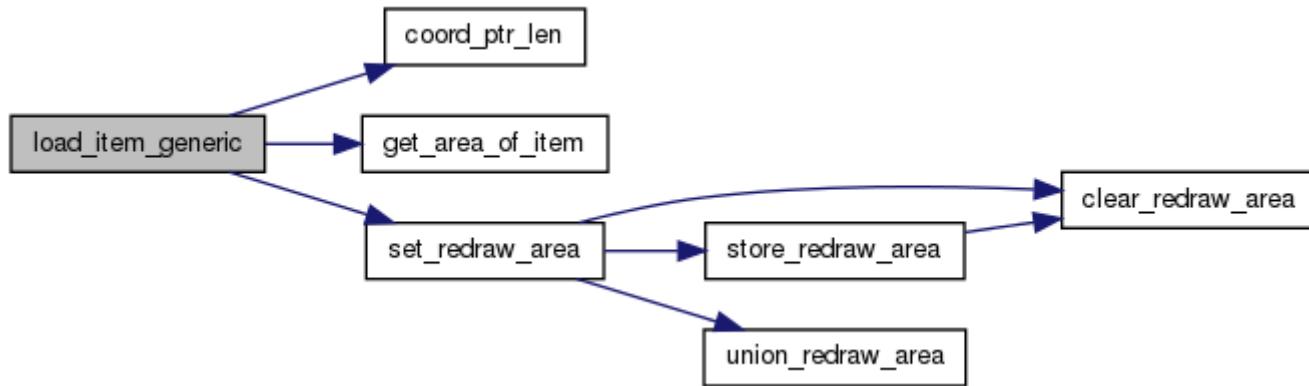
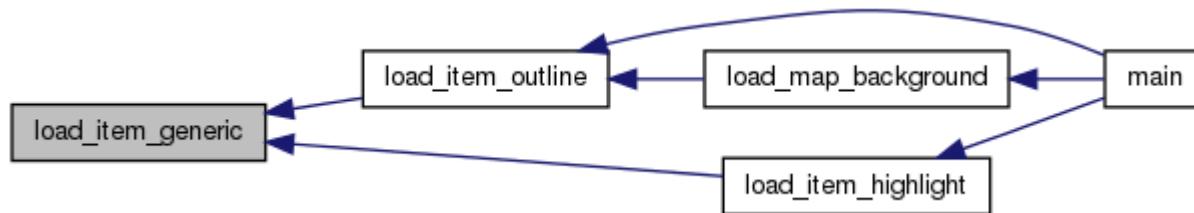
        for (int cnt_l = LAYER_BOTTOM; cnt_l < LAYER_TOP; cnt_l++)
        {
            char alpha = layers->xmg[cnt_l + 1]->
                         data[rel_ofs - pxl_ofs + 3];

            layers->xmg[LAYER_BUFFER]->data[abs_ofs] =
                (unsigned char)layers->xmg[cnt_l + 1]->data[rel_ofs]
                * (unsigned char)alpha / MAP_COLORS
                + (unsigned char)layers->xmg[LAYER_BUFFER]->data[abs_ofs]
                * (0xFF - (unsigned char)alpha) / MAP_COLORS;
        }
    }
}
```

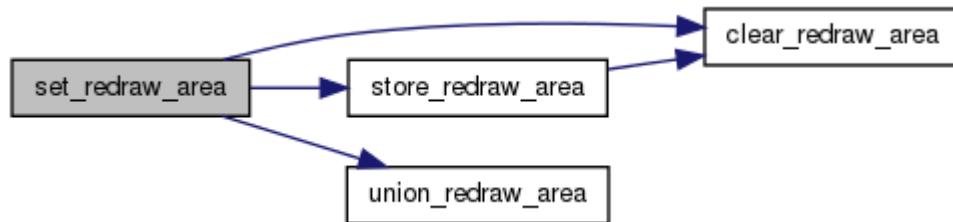
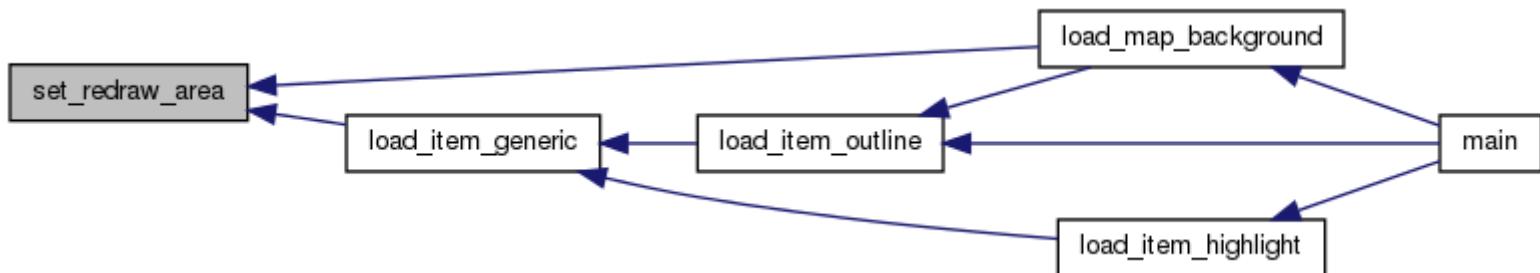
Code Visualization



Code Reuse



Code Reuse



Thanks for your listening