

---

浙江大学

# Object-Oriented Programming

## 中期 project 报告



学生姓名： 张童童      学号： 3160101315

学生姓名： 江如蓝      学号： 3160103777

学生姓名： 王钟毓      学号： 3170105709

学生姓名： 陈宇威      学号： 3170105706

学生姓名： 彭子帆      学号： 3170105860

2018 - 2019 春夏学期    2019 年 4 月

## 一、项目实现的 IDE、compiler、库

### 1.1 IDE 介绍

Microsoft Visual Studio Professional 2017

版本 15.9.4

VisualStudio.15. Release /15.9.4+28307.222

Visual C++ 2017 00369-60000-00001-AA074

Microsoft Visual C++ 2017

### 1.2 Compiler 介绍

Microsoft (R) Windows (R) Resource Compiler

Version 10.0.10011.16384

### 1.3 库使用

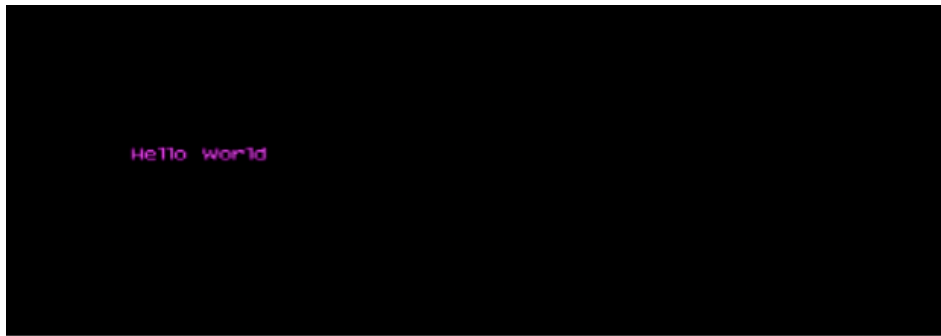
考虑到经济成本以及实现功能的需求，基本的开源图像处理库已经能够基本满足我们的需求，因此我们选择了一下几个开源的图像处理库

#### 1.3.1 Cimg

Cimg 是为 C++ 程序员设计的一个图像处理库。它为加载/保存、显示和处理多种类型的图像提供了有用的类和函数。Cimg 库具有较好的移植性 (Unix/X11、Windows、MacOS...) 并且 cimg 库只包含了一个名为 cimg.h 的头文件。头文件 cimg.h 包含了构成 cimg 库的全部类和函数。这是 cimg 库的一个新颖之处。这意味着：

1) 不需要对库的提前编译：因为 cimg 函数是和自己的 C++ 代码同时编译的。

- 2) 不用处理复杂的依赖关系：只要包含文件 `cimg.h`，就可以使用 C++ 图像处理工具箱。
- 3) 编译是按需完成的：只要程序中用到的 `cimg` 函数才被编译，并出现在编译后的可执行文件中。这样可以产生没有任何无用东西的，非常紧凑的代码。
- 4) 类成员和函数都是内联的，这样程序在执行的时候会有更好的表现。



Graph 1. cimg 版的 “Hello world”

#### 1.3.1.1 需要使用的函数：

绘图函数	<code>draw_point</code> 、 <code>draw_line</code> 、 <code>draw_point</code> 、 <code>draw_triangle</code>  <code>draw_spline</code> 、 <code>draw_arrow</code> 、 <code>draw_rectangle</code> 、 <code>draw_polygon</code>  <code>draw_ellipse</code> 、 <code>draw_circle</code> 、 <code>draw_image</code> 、 <code>draw_text</code>  <code>draw_quiver</code> 、 <code>draw_axis</code>
图像调整	<code>rotate</code> 、 <code>resize_halfXY</code> 、 <code>resize_doubleXY</code> 、 <code>resize_tripleXY</code>  <code>translate</code> 、 <code>resize</code>
相关参数	-1 = 不插值：图像直接按大小剪切  0 = 不插值：多余空间依据 <code>border_condition</code> 决定  1 = 临近点插值

	2 = 移动平均数插值
	3 = 线性插值
	4 = 删格插值
	5 = 双三次插值

### 1.3.2 OpenCV

OpenCV 是一个基于 BSD 许可(开源)发行的跨平台计算机视觉库,可以运行在 Linux、Windows、Android 和 Mac OS 操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成,同时提供了 Python、Ruby、MATLAB 等语言的接口,实现了图像处理 and 计算机视觉方面的很多通用算法。

OpenCV 用 C++ 语言编写,它的主要接口也是 C++ 语言,但是依然保留了大量的 C 语言接口。该库也有大量的 Python、Java and MATLAB/OCTAVE (版本 2.5) 的接口。这些语言的 API 接口函数可以通过在线文档获得。如今也提供对于 C#、Ch、Ruby,GO 的支持。所有新的开发和算法都是用 C++ 接口。一个使用 CUDA 的 GPU 接口也于 2010 年 9 月开始实现。

OpenCV 的安装与使用可参看 openCV 教程:

<http://www.opencv.org.cn/opencvdoc/2.3.2/html/doc/tutorials/tutorials.html>

### 1.3.2.1 需要使用的函数:

图像读取	imread
颜色转换	cvtColor
图像保存	imwrite

### 1.3.2.2 程序使用

```
1. #include <cv.h>
2. #include <highgui.h>
3.
4. using namespace cv;
5.
6. int main( int argc, char** argv )
7. {
8.     if(argc != 2)
9.     {
10.         printf("usage: %s <imagefile>\n ", argv[0]);
11.         return -1;
12.     }
13.     char* imageName = argv[1];
14.
15.     Mat image;
16.     image = imread( imageName, CV_LOAD_IMAGE_COLOR);
17.
18.     if( !image.data )
19.     {
20.         printf( " No image data \n " );
21.         return -1;
22.     }
23.
24.     Mat gray_image;
25.     cvtColor( image, gray_image, CV_BGR2GRAY );
26.
27.     imwrite( "../images/Gray_Image.jpg", gray_image );
28.
29.     namedWindow( imageName, CV_WINDOW_AUTOSIZE );
30.     namedWindow( "Gray image", CV_WINDOW_AUTOSIZE );
```

```

31.
32. imshow( imageName, image );
33. imshow( "Gray image", gray_image );
34.
35. waitKey(0);
36.
37. return 0;
38. }

```

Code 1. imread、cvtColor、imwrite 使用

### 1.3.2.3 改变图像的对比度和亮度:

在这一类图像处理变换中，仅仅根据输入像素值（有时可加上某些全局信息或参数）计算相应的输出像素值。这类算子包括 *亮度和对比度调整*，以及颜色校正和变换。亮度和对比度调整两种常用的点过程（即点算子），是用常数对点进行 *乘法* 和 *加法* 运算：

```

▪ #include <opencv2/core/core.hpp>
▪ #include <opencv2/highgui/highgui.hpp>
▪ #include <iostream>
▪
▪ using namespace std;
▪ using namespace cv;
▪
▪ double alpha; /**< 控制对比度 */
▪ int beta; /**< 控制亮度 */
▪
▪ int main( int argc, char** argv )
▪ {
▪     /// 读入用户提供的图像
▪     Mat image = imread( argv[1] );
▪     Mat new_image = Mat::zeros( image.size(), image.type() );
▪
▪     /// 初始化
▪     cout << " Basic Linear Transforms " << endl;
▪     cout << "-----" << endl;
▪     cout << "* Enter the alpha value [1.0-3.0]: ";
▪     cin >> alpha;

```

```

▪      cout << "Enter the beta value [0-100]: ";
▪      cin >> beta;
▪
▪      /// 执行运算 new_image(i,j) = alpha*image(i,j) + beta
▪      for( int y = 0; y < image.rows; y++ )
▪      {
▪          for( int x = 0; x < image.cols; x++ )
▪          {
▪              for( int c = 0; c < 3; c++ )
▪              {
▪                  new_image.at<Vec3b>(y,x)[c] = saturate_cast<uchar>( alpha*(
image.at<Vec3b>(y,x)[c] ) + beta );
▪              }
▪          }
▪      }
▪
▪      /// 创建窗口
▪      namedWindow("Original Image", 1);
▪      namedWindow("New Image", 1);
▪
▪      /// 显示图像
▪      imshow("Original Image", image);
▪      imshow("New Image", new_image);
▪
▪      /// 等待用户按键
▪      waitKey();
▪      return 0;
▪  }

```

Code 2. 改变图像对比度

### 1.3.2.4 图像添加水印

使用 openCV 对两幅图像进行求和（实现添加水印的功能）：*线性混合操作*也是一种典型的二元（两个输入）的 *像素操作*：通过在范围  $0 \rightarrow 1$  内改变  $\alpha$ ，这个操可以用来对两幅图像或两段视频产生时间上的 *画面叠化*（cross-dissolve）效果，就像在幻灯片放映和电影制作中那样。

```
1. #include <cv.h>
2. #include <highgui.h>
3. #include <iostream>
4.
5. using namespace cv;
6.
7. int main( int argc, char** argv )
8. {
9.     double alpha = 0.5; double beta; double input;
10.
11.     Mat src1, src2, dst;
12.
13.     /// Ask the user enter alpha
14.     std::cout<<" Simple Linear Blender "<<std::endl;
15.     std::cout<<"-----"<<std::endl;
16.     std::cout<<"* Enter alpha [0-1]: ";
17.     std::cin>>input;
18.
19.     /// We use the alpha provided by the user iff it is between 0 and 1
20.     if( alpha >= 0 && alpha <= 1 )
21.     { alpha = input; }
22.
23.     /// Read image ( same size, same type )
24.     src1 = imread("../images/LinuxLogo.jpg");
25.     src2 = imread("../images/WindowsLogo.jpg");
26.
27.     if( !src1.data ) { printf("Error loading src1 \n"); return -1; }
28.     if( !src2.data ) { printf("Error loading src2 \n"); return -1; }
29.
30.     /// Create Windows
31.     namedWindow("Linear Blend", 1);
32.
33.     beta = ( 1.0 - alpha );
34.     addWeighted( src1, alpha, src2, beta, 0.0, dst);
35.
36.     imshow( "Linear Blend", dst );
37.
38.     waitKey(0);
39.     return 0;
40. }
```

Code 3. 图像添加水印



### 1.3.2.4 图像分割

读取一副图片，如果是图片颜色类型是 RGB3 色类型，则转换成灰度类型的图像。转换颜色类型可以运用 OpenCV 中的 `cvtColor` 函数。然后创建一个窗口来显示该图片可以检验转换。结果接着该程序创建两个滚动条来等待用户的输入。在这里等到用户拖动滚动条来输入阈值类型以及阈值的大小，或者是用户键入 ESC 键退出程序。无论何时拖动滚动条，用户自定义的阈值函数都将会被调用。

```
1. #include "opencv2/imgproc/imgproc.hpp"
2. #include "opencv2/highgui/highgui.hpp"
3. #include <stdlib.h>
4. #include <stdio.h>
5.
6. using namespace cv;
7.
8. /// 全局变量定义及赋值
9.
10. int threshold_value = 0;
11. int threshold_type = 3;;
12. int const max_value = 255;
13. int const max_type = 4;
14. int const max_BINARY_value = 255;
15.
16. Mat src, src_gray, dst;
17. char* window_name = "Threshold Demo";
18.
19. char* trackbar_type = "Type: \n 0: Binary \n 1: Binary Inverted \n 2: Trunc
    ate \n 3: To Zero \n 4: To Zero Inverted";
20. char* trackbar_value = "Value";
21.
22. /// 自定义函数声明
23. void Threshold_Demo( int, void* );
24.
25. /**
26.  * @主函数
27.  */
28. int main( int argc, char** argv )
```

```
29. {
30.     /// 读取一副图片, 不改变图片本身的颜色类型 (该读取方式为 DOS 运行模式)
31.     src = imread( argv[1], 1 );
32.
33.     /// 将图片转换成灰度图片
34.     cvtColor( src, src_gray, CV_RGB2GRAY );
35.
36.     /// 创建一个窗口显示图片
37.     namedWindow( window_name, CV_WINDOW_AUTOSIZE );
38.
39.     /// 创建滑动条来控制阈值
40.     createTrackbar( trackbar_type,
41.                    window_name, &threshold_type,
42.                    max_type, Threshold_Demo );
43.
44.     createTrackbar( trackbar_value,
45.                    window_name, &threshold_value,
46.                    max_value, Threshold_Demo );
47.
48.     /// 初始化自定义的阈值函数
49.     Threshold_Demo( 0, 0 );
50.
51.     /// 等待用户按键。如果是 ESC 键则退出等待过程。
52.     while(true)
53.     {
54.         int c;
55.         c = waitKey( 20 );
56.         if( (char)c == 27 )
57.             { break; }
58.     }
59.
60. }
61.
62.
63. /**
64.  * @自定义的阈值函数
65.  */
66. void Threshold_Demo( int, void* )
67. {
68.     /* 0: 二进制阈值
69.        1: 反二进制阈值
70.        2: 截断阈值
```

```

71.     3: 0 阈值
72.     4: 反 0 阈值
73.     */
74.
75.     threshold( src_gray, dst, threshold_value, max_BINARY_value,threshold_type
76.               );
77.     imshow( window_name, dst );
78. }

```

Code 4. 图像分割

#### 1.4.QT(实现图形界面)

Qt 是一个 1991 年由 Qt Company 开发的跨平台 C++ 图形用户界面应用程序开发框架。它既可以开发 GUI 程序，也可用于开发非 GUI 程序，比如控制台工具和服务器。Qt 是面向对象的框架，使用特殊的代码生成扩展（称为元对象编译器(Meta Object Compiler, moc)) 以及一些宏，Qt 很容易扩展，并且允许真正地组件编程 Qt 支持下述平台：

MS/Windows - 95、98、NT4.0、ME、2000、XP 、 Vista、Win7、win8、win2008、win10、Unix/X11 -Linux、SunSolaris、HP-UX、CompaqTru64 UNIX、IBMAIX、SGI IRIX、FreeBSD、BSD/OS 和其它很多 X11 平台 Macintosh -Mac OS X

Embedded - 有帧缓冲(framebuffer)支持的嵌入式 Linux 平台, Windows CE、Symbian、Symbian^3、Symbian Anna、Symbian Belle、MeeGo、haiku-os。

```

1. #include "mainwindow.h"
2. #include <QApplication>
3. #include <QPushButton>
4. int main(int argc, char *argv[])
5. {
6.     QApplication a(argc, argv);
7.     MainWindow w;
8.
9.     QPushButton b;
10.

```

```
11.     b.setText("Hello world"); //给按钮设置内容
12.     b.setParent(&w); //指定父对象，注意是取地址
13.     b.move(10, 10); //移动坐标
14.
15.     w.show();
16.
17.     return a.exec();
18. }
```

Code 5. QT 实现 hello world

## 二、项目需要完成的功能目标

### 2.1 基本功能

#### 2.1.1 图像大小调整、平移、缩放、旋转、分割功能;

对于图像的基本简单操作，也是图像处理工具软件最基础的基本功能。包括大小的缩放和分割等编辑功能和平移、旋转等对于位置的控制。

#### 2.1.2 能够显示直方图;

图像直方图由于其计算代价较小，且具有图像平移、旋转、缩放不变性等众多优点，广泛地应用于图像处理的各个领域，特别是灰度图像的阈值分割、基于颜色的图像检索以及图像分类。

#### 2.1.3 具有图像（圆、三角形、长方形）编辑功能

#### 2.1.4 可将图像文件保存、打开;

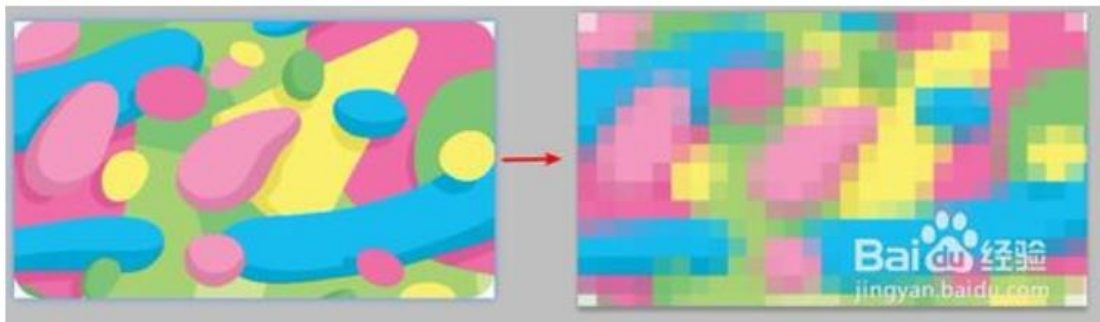
图像处理工具软件需要导入要编辑的图像并输出处理完毕的图像，这就需要图像文件的保存和打开功能。

## 2.2 扩展功能

### 2.2.1 图片像素化

输入一张图片，产生图片被分解为较大的色块，从而产生一种马赛克的效果，图像实际像素大小不变，图像含义不变，要求依然可辨认图片含义。可实现图像选定区域的像素化，可选择像素化的程度，由像素化图片返回原图片，不要求多次像素化后的图片返回最初图片。

目前暂时没有找到相关的库可支持像素化这以一功能。选定区域像素化这比较简单只要限定好像素点位置的范围。像素化目前的思路是将选定区域所有像素点分块，如四个像素点或九个像素点或十六个像素点作为一个基本块，每一块具有一个统一的 RGB 值，这个统一的 RGB 值由组成这个块的所有像素点共同决定，可以取平均值。边角有无法取到四个或九个或十六个像素点的依然作为独立的一块。



Graph 2. 像素化示意图

### 2.2.2 图片增亮

对于黑白图像和彩色图像，都能改变图片的亮度,既能够增亮也能够使得图片变暗。如果时间和技术允许，增加进一步功能选择区域改变亮度，选择区域和被选择区域之间要有渐变效果。网上有很多图片增强的实现方法，譬如灰色图像可直接改变其灰度，而对于彩色图像有很多修改 RGB 的方法。我们初步采用的方法是将 RGB 空间转到 HSL 修改 L，再将 HSL 转回 RGB 空间，此外 Opencv 库也有相应的辅助工具，我们会对比这两个方法的效果选择视觉效果最佳的方法。



Graph 3. 图片增量示意图

### 2.2.3 图片增加水印

为图片增加水印，要求水印半透明不能将背后图片完全盖住。初步要求是输入图片后可在已有水印模板中选择水印添加，水印位置可自选，高级要求提供输入界面，用户输入文字产生相应水印并添加，提供图片导入界面，用户选择要导入的图片作为水印添加

功能涉及到多个图片的叠加处理，实现方法与前端的用户界面的设计有很大关系。如果需要从库中选择水印，前端界面中就需要有现有水印的展示区域，并且用户选择该水印图形

后可以将该水印图形导入进来。如果输入文字产生水印则更为复杂，输入可能性增大而且将 ASCII 码转化为相应的图片相当于该项目要有自己生成图片而不仅仅是改变图片的能力。

如果仅仅考虑将水印贴到图片上这一过程，网上有现成的方法和工具。



Graph 4. 添加水印示意图

#### 2.2.4 降噪功能

去除图片当中的噪点，同时尽可能保持图片原有的面貌。降噪的方法有很多种，包括古典的线性滤波，基于频波降噪、多图片平均法降噪、变分法、PDE、微分几何的方法等。网上的很多方法都有非常好的效果。我们目前的期望是使用中值滤波，不过也可以提供多种方法，让用户来选择。图片处理过程会需要很多矩阵运算，所以我们需要提前寻找合适的矩阵运算类库来提高效率。



Graph 5. 降噪功能示意图



## 2.3 高级功能

### 2.3.1 图像池化 (深度视觉预处理)

卷积层是对图像的一个领域进行卷积得到图像的领域特征，亚采样层（池化层）就是使用 pooling 技术将小领域内的特征点整合得到新的特征。在完成卷积特征提取之后，对于每一个隐藏单元，它都提取到 $(r-a+1)*(c-b+1)$ 个特征，把它看做一个矩阵，并在这个矩阵上划分出几个不重合的区域，然后在每个区域上计算该区域内特征的均值或最大值，然后用这些均值或最大值参与后续的训练，这个过程就是池化。

### 2.3.2 图像滤镜-亮度-色温-曝光度

提供图像效果的渲染功能。除此之外，还提供基本的图像的亮度、色温、曝光度等基本属性的调整功能。

## 三、基本实现思路

### 3.1 Project 基本实现思路

程序功能总体分为三个部分进行。基本功能可通过调用 OpenCV 库中的函数实现，如平移、缩放、分割、旋转等操作。其中，显示直方图则是通过遍历像素点，并对每一个颜色通道的像素点值进行统计，调用绘图功能，则可完成。而可视化操作界面将通过 QT 完成，通过信号槽完成参数的传递。

扩展功能则涉及对图像底层的实现。图像像素化，可通过灰度算法结合每一点及相邻点的 RGB 值实现；图像增量则涉及图像的中值滤波计算实现，找出其周围与其最接近且亮度最高的元素；图像添加水印功能可通过 OpenCV 中，DrawText 函数实现；降噪功能则可



通过较为经典的变分法算法，取图像梯度的平方，将图像降噪视作为求解优化问题；图像滤镜则需要分情况，如进行浮雕效果，只需要将图像的一个像素和左上方的像素进行求差运算，并添加表示背景颜色的灰度。而扩展功能的图像清晰度可拆分为较为基本的增加黑场、提高对比度、锐化进行提高。

高级功能则涉及到与其他计算机领域的联动，如图像风格化与图像池化是与深度视觉的结合。目前，业界已有相关模型提供，可直接用于生产，只需要相应的接口即可。而图像数据库、撤销操作，则是保存用户的每一步操作所生成的图像，用户可以自由撤销到指定步骤。



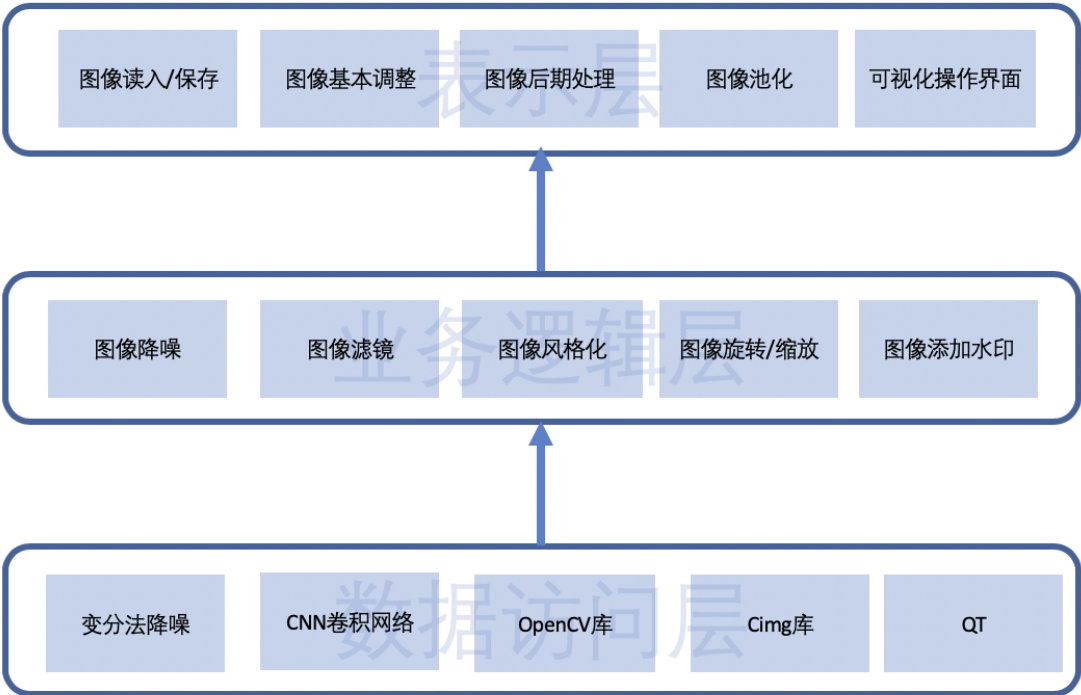
Graph 6. 功能基本实现思路

### 3.2 Project 总体架构设计

整个程序架构将分为三个部分进行。数据访问层提供了底层支持，变分法提供了图像降噪功能，而 CNN 卷积神经网络提供了风格化支持，OpenCV 和 Cimg 是图像处理中两个常用的库，其内置了图像滤镜、旋转/缩放等函数，而 QT 将作为图像界面开发语言。

业务逻辑层，整合了数据访问层提供的多个函数，进行汇总，实现了用户交互中涉及到的多个功能。如图像风格化功能，需要 QT、OpenCV 和 CNN 卷积神经网络三者的结合。

而表示层，则展现了用户交互功能，用户通过可视化界面读入图像，完成对图像的处理，并保存图像。



Graph 7. Project 总体架构设计

#### 四、人员职责

角色	姓名 (学号)	职责
组长	彭子帆 (3170105860)	1. 总体策划, 负责协调活动 2. 负责召集组内成员学习讨论 3. IDE 下载、安装、调试; 4. 代码实现; 5. Project 中期报告撰写。
组员	张童童 (3160101315)	1. 代码实现; 2. 库的下载、安装、使用;
组员	江如蓝 (3160103777)	✧ 代码实现; ✧ IDE 下载、安装、调试;
组员	王钟毓 (3170105709)	✧ 代码实现; ✧ Project 总体架构设计;
组员	陈宇威 (3170105706)	✧ Project 总体测试; ✧ 代码实现;


六、附录

6.1 小组会议记录

∞(937427981) 10:23:55  
我们等会  
∞(937427981) 10:23:59  
讨论一下做什么  
∞(937427981) 10:24:01  
分工吧  
∞(937427981) 10:24:07  
10: 45  
王钟毓<wzy0173@qq.com> 10:25:44  
收到  
王钟毓<wzy0173@qq.com> 12:18:37  
那要不我们晚上9点叭  
王钟毓<wzy0173@qq.com> 12:18:40  
@ 277 @奥拉夫  
江如蓝(805034982) 12:18:51  
ok  
张童童(290410607) 12:23:53  
ok  
王钟毓<wzy0173@qq.com> 21:03:52  
大家好了嘛w  
王钟毓<wzy0173@qq.com> 21:04:33  
要不我们语音说一下吧  
∞(937427981) 21:04:56  
我好了  
王钟毓<wzy0173@qq.com> 21:06:26  
@ 277 @奥拉夫  
陈宇威(1051260524) 21:09:08  
三个人面对面语音  
张童童(290410607) 21:11:01

张童童(290410607) 21:11:15  
sorry啊大家我大概不能语音  
王钟毓<wzy0173@qq.com> 21:11:22  
emm那我们打字吧  
张童童(290410607) 21:11:29  
可以的  
∞(937427981) 21:12:08  
那我们先来决定选题吧  
∞(937427981) 21:12:35  
• 题目 1: 企业工资管理系统  
目标: ∞  
1、企业最起码有管理人员、技术人员、工人三类人员，每一类人员有不同的工资计算方法（具体计算方法自行设计）； ∞  
2、可以存储、显示、修改和删除企业人员信息。 ∞  
3、可以查询、统计工资信息。 ∞  
• 题目 2: 文本编辑器  
∞(937427981) 21:12:43  
• 题目 3: 学生成绩管理系统  
目标: ∞  
1、有不同类别的学生（本科生、硕士生和博士生）； ∞  
2、实现学生成绩的录入、显示、查询、修改和删除等功能。  
• 题目 4: 图形编辑器  
目标: ∞  
1、图形文件的读取和保存 ∞  
2、各种基本线条，图形（圆、长方形、三角形）的控制 ∞  
• 题目 5: 邮件系统  
目标: ∞  
1、支持基本的用户注册和管理 ∞


2019/4/9 21:43:39



王钟毓

我们图形界面用QT 好不好呀 大家w


2019/4/9 21:44:09



王钟毓

求各位可怜一下没有双系统的废物口

2019/4/9 22:54:36



张童

行啊

6.2 小组会议记录

江如蓝(805034982) 21:19:11

(想做图形)

∞(937427981) 21:19:19

那就做吧真的

∞(937427981) 21:19:21

我们都行

张童童(290410607) 21:19:42

我也都行

张童童(290410607) 21:19:44



江如蓝(805034982) 21:19:52

图形可以吗?

王钟毓<wzy0173@qq.com> 21:19:54

可以挖

∞(937427981) 21:19:57

我们4还是7

王钟毓<wzy0173@qq.com> 21:20:00

奥拉夫学姐 你来决定叭

∞(937427981) 21:20:11

可以的

∞(937427981) 21:20:13

hhh

江如蓝(805034982) 21:20:49

那7?

王钟毓<wzy0173@qq.com> 21:21:00



∞(937427981) 21:24:16

要不就VS怎么样

张童童(290410607) 21:24:32

我也是VS

张童童(290410607) 21:24:34



王钟毓<wzy0173@qq.com> 21:24:36

...xcod 哭了

∞(937427981) 21:25:00

mac走开

张童童(290410607) 21:25:02

其实我也有xcod

∞(937427981) 21:25:03

我们不是一族的了

张童童(290410607) 21:25:04



王钟毓<wzy0173@qq.com> 21:25:06

这个问题不大emm

王钟毓<wzy0173@qq.com> 21:25:08

到时候分开写 我和他们住的比

王钟毓<wzy0173@qq.com> 21:25:10

0-0可以到他们电脑上

∞(937427981) 21:26:17

基本功能目标，扩展功能目标与高级功能目标。❖

> 基本功能目标就是软件必须完成达到主实践任务的最低要求的功能目标。也是软件迭代开发第一轮应该完成的功能。❖

> 扩展功能目标是指软件推荐完成的功能目标。一般是软件迭代开发中第二轮完成的功能。❖

> 高级功能目标是指实践可以努力去完成的包括学生自己设想出来的功能目标。一般是软件迭代开发以后过程中完成的。❖

需要完整列出功能点，对这些功能点的说明详略，组内成员可以有所不同，组长需进行完整的比较详细的说明，其他成员只需对自己分工关系比较密切的功能进行详细的说明，对于详细说明的功能，同时还应该对它们的性能要求进行必要的讨论。❖

∞(937427981) 21:26:24

我们有最终的目标的

∞(937427981) 21:26:30

老师提供的是基本功能

∞(937427981) 21:26:49

我们期中报告是需要列出我们的功能目标

∞(937427981) 21:27:29

也不一定要现在订，但是接下来做的时候我们需要不断提出来

王钟毓<wzy0173@qq.com> 21:28:28




∞(937427981) 21:29:08

这是我们需要进行的分工

∞(937427981) 21:29:08

职责❖
❖ 总体策划，负责协调活动❖



一、IDE介绍\Complier\库\人员职责【文档撰写】 PZF

二、项目需要完成的功能目标（参考群文件word）

库的下载、安装【代码部分】【涉及所有功能的调用库 可用库的数量与编程难度成反比】江如蓝、张童童

Project总体架构设计【代码部分】王钟毓

基本功能【文档部分】陈宇威

扩展功能【文档部分】江如蓝

高级功能【文档部分】陈宇威

基本实现思路【文档部分】王钟毓


三、附录【文档撰写】-张童童

标题【宋体 14号 加粗】正文宋体14号 1.5倍行距

DDL下周五晚上23：59分 - 之后进行中期报告相应的优化




我们这样分工一下吧



王钟毓

收到

2019/4/7 11:06:15



张童童

ok

6.3 小组会议记录

oop team ↗



王钟毓

<https://www.qt.io/offline-installers>



王钟毓

Linux Host

- Qt 5.12.3 for Linux 64-bit (1.4 GB) (info)

macOS Host

- Qt 5.12.3 for macOS (3.2 GB) (info)


Windows Host

- Qt 5.12.3 for Windows (3.7 GB) (info)

Source packages & Other releases

The source code is available:

- For Windows users as a single zip file (792 MB) (info)
- For Linux/macOS users as a tar.xz file (484 MB) (info)



王钟毓


大噶下载windows的这个 3.7G



好大




大大大大



张童童

真滴大23333



王钟毓

嘻嘻嘻 安装完更大



我电脑



张童童





我电脑






还能行



吗



张童童

可以的