

软件测试说明书

项目名称：MAGIC-SCAN 图片编辑系统

测试文档第一负责人：16020510041 陈桂海

项目成员：16020510041 陈桂海、
16020510068 李慧、
16020510069 陈皖凤、
16020510062 阿尔达合·巴格努尔

文档编写时间：2019.1.1

文档页数：14

目录

- 一、 引言 3
 - 1.1 编写目的 3
 - 1.2 术语 3
 - 1.3 参照标准 3
- 二、 测试内容 3
 - 2.1 合法性检查 3
 - 2.2 软件代码测试 3
 - 2.2.1 源代码一般性检查 3
 - 2.2.2 软件一致性检查 4
 - 2.2.3 软件代码模块测试报告 4
 - 2.3 软件系统测试 10
 - 2.3.1 界面测试 11
 - 2.3.2 功能测试 11
 - 2.3.3 性能测试 12
 - 2.3.4 强度测试 12
 - 2.3.5 容量测试 12
 - 2.3.6 安全性测试 12
 - 2.3.7 安装测试 13
 - 2.3.8 配置测试 13
 - 2.3.9 破坏性测试 13
 - 2.3.10 可用性测试 13
- 三、 测试日志 13
- 四、 测试总结 14

一、引言

1.1 编写目的

为了尽可能的找出 MAGIC-SCAN 软件的不足，提高软件的质量，促进软件的成功验收，专门编写本文档。其主要目的一方面在于为所要进行的测试工作制定各种必要的准则和规范， 另一方面在有关方面协议的基础上对测试工作进行合理组织与管理以及记录下软件测试的结果。

1.2 术语

本文档所提及的术语，其定义遵照 GB/T 11457 标准。

1.3 参照标准

GB9386 — 1988 计算机软件测试文件编制指南。

二、测试内容

2.1 合法性检查

检查开发的软件的合法性（用途是否合法、目的是否合法）；检查开发者在开发本软件时，使用的开发工具是否合法；对在编程中使用的一些非自身开发的，也不是由开发工具提供的控件、组件、函数库等，检查其是否有合法的发布许可。

2.2 软件代码测试

2.2.1 源代码一般性检查

1、命名规范检查

测试目标	检查源代码中的变量、函数、对象、过程等的命名是否符合约定规范，该规范可以由开发方在软件工程文档规范中单方面约定
测试方法和技术	根据软件工程文档的约定，对代码进行检查（黑盒测试、白盒测试）
完成标准	系统中重要部分都按规定命名
需考虑的特殊事项	无

2、注释检查

测试目标	检查程序中的注释是否规范，注释量是否达到约定要求
测试方法和技术	让测试小组人员，对代码进行检查
完成标准	测试人员能根据注释读懂代码并进行后面其他测试
需考虑的特殊事项	无

3、限制性检查

测试目标	在这个系统中，主要检测图片文件选择的格式与数量，检查打开文件不正确会出现什么错误
测试方法和技术	选择不同格式的图片，并打开；选择不同数量的图片来打开，测试是否能正常打开
完成标准	不符合正确打开方式的话会报错（闪退之类的）
需考虑的特殊事项	无

2.2.2 软件一致性检查

1、编译检查

要求提交的源代码在其规定的编译环境中，能够重新编译无错误，并且能够完成相应的功能，从而确定移交的确实是正确的源代码。

2、运行环境检查

在不同的 PC 系统上用交付的 EXE 文件重新打开，并且通过运行软件，测试能否完成相应的功能，从而确定软件是否能在不同环境下正确运行。在运行软件后立即卸载，并且检查是否能够做到彻底卸载。

3、运行模块检查

将源码中的模块与现场运行模块用软件工具抽样比较，确认模块运行正常。

2.2.3 软件代码模块测试报告

黑盒测试：

1、选择、打开

测试用例编号	01	测试需求项	MAGIC-SCAN	设计者	陈桂海
测试需求标识				设计日期	2018-12-25
测试目标状态和测试数据状态	达到预期要求				
序号	测试项	输入说明（操作）		输出说明（预期结果）	
1	文件选择	点击打开按钮，选择文件路径		进入本地文件选择页面	
		按照格式及数量提示选择图片文件		选择成功	
		不按照格式及数量选择图片文件		闪退	

2	图片显示	选择文件后点击确定按钮	显示图片
---	------	-------------	------

2、基本工具

测试用例编号		02	测试需求项	MAGIC-SCAN	设计者	陈桂海
测试需求标识					设计日期	2018-12-25
测试目标状态和测试数据状态		达到预期要求				
序号	测试项	输入说明（操作）		输出说明（预期结果）		
1	放大缩小	点击文件，打开图片后点击放大镜图标		图片放大显示		
		点击缩小镜图标		图片缩小显示		
2	添加水印签名	点击文件按钮，打开图片后点击“个性签名”按钮		弹出水印写入对话框		
		写入水印，点击确定按钮		图片显示出水印		
3	上下翻转	点击“上下倒立”图标		图片发生上下翻转		
4	镜像翻转	点击“照镜子”图标		图像发生镜像翻转		
5	图片拼接	点击“图图拼接”图标，选择两张图片		选择的俩图片拼接起来组成一张图片		
6	剪切	点击“剪切”图标				
7	旋转	打开图片后点击“旋转吧”图标		图片发生旋转,		

3、退出系统

测试用例编号		03	测试需求项	MAGIC-SCAN	设计者	陈桂海
测试需求标识					设计日期	2018-12-25
测试目标状态和 测试数据状态		达到预期要求				
序号	测试项	输入说明（操作）		输出说明（预期结果）		
1	直接退出系统	点击右上角退出“×”号		显示 warning 页面		
		点击 warning 页面的确认键		退出成功，页面关闭		
2						

4、“魔法棒”模块

测试用例编号		04	测试需求项	MAGIC-SCAN	设计者	陈桂海
测试需求标识					设计日期	2018-12-25
测试目标状态和测试数据状态		达到预期要求				
序号	测试项	输入说明（操作）		输出说明（预期结果）		
1	模糊化	点击“文件”按钮，点击打开，选择图片路径		显示图片		
		拉动进度条或点击进度条上的位置		图片变模糊		
2	锐化	选择图片路径并打开		显示图片		
		拉动进度条或点击进度条上的位置		图片锐化		
3	油画	选择图片路径并打开		显示图片		
		拉动进度条或点击进度条上的位置		图片变油画风格		
4	七彩	选择图片路径并打开		显示图片		
		拉动进度条或点击进度条上的位置		图片变成由各种颜色显示的状态		
5	保存修改	点击“保存修改”按钮		经过“魔法棒”处理的图片状态被锁定，无法变回原图		

5、“我是逗逼”模块

测试用例编号		05	测试需求项	MAGIC-SCAN	设计者	陈桂海
测试需求标识					设计日期	2018-12-25
测试目标状态和测试数据状态			达到预期要求			
序号	测试项	输入说明（操作）		输出说明（预期结果）		
1	图片融合术	点击“图片融合术”按钮		显示文件选择对话框		
		选择两张图片，点击确定		俩图片以一定的透明度融合成一张图显示		
2	生成字符画	点击“生成字符画”按钮		显示选择字符画文件保存路径的对话框		
		选择路径		以 txt 文件形式保存到具体路径		
3	制作表情包	点击“制作表情包”按钮		显示用于制作表情包的图片文件选择的对话框		
		选择图片后点击打开		图片变为二值图，后边附带着一把卡通剪刀		
4	我爱算脸缘	点击“我爱算脸缘”按钮		显示图片选择文件对话框		
		选择两张图片后点击“打开”		显示相似度比较结果		

6、“潘多拉魔盒”模块

测试用例编号		06	测试需求项	MAGIC-SCAN	设计者	陈桂海
测试需求标识					设计日期	2018-12-25
测试目标状态和测试数据状态		达到预期要求				
序号	测试项	输入说明（操作）		输出说明（预期结果）		
1	黑白空间	点击“文件”按钮，点击“打开”		显示文件选择对话框		
		选择图片后点击“黑白空间”按钮		图片变成二值图		
2	轮廓印象	点击“文件”按钮，点击“打开”		显示文件选择对话框		
		选择图片后点击“轮廓印象”按钮		显示图像的边缘（边缘检测）		
3	浮雕变化	点击“文件”按钮，点击“打开”		显示文件选择对话框		
		选择图片后点击“浮雕变化”按钮		图片以浮雕风格显示		
4	熔岩魔鬼	点击“文件”按钮，点击“打开”		显示文件选择对话框		
		选择图片后点击“熔岩魔鬼”按钮		图片覆盖一层红色后显示		

7、图片保存

测试用例编号		07	测试需求项	MAGIC-SCAN	设计者	陈桂海
测试需求标识					设计日期	2018-12-25
测试目标状态和 测试数据状态		达到预期要求				
序	测试项	输入说明（操作）		输出说明（预期结果）		
1	保存图片	点击保存按钮		显示选择保存路径页面		
		选择路径，并点击确定		图片文件成功保存		

白盒测试（测试代码）：

单文件和多文件的选择与文件的保存：

```
#coding:utf-8
from PyQt5 import QtWidgets
from PyQt5.QtWidgets import QFileDialog
from nose.tools import *

class MyWindow(QtWidgets.QWidget):
    def __init__(self):
        super(MyWindow, self).__init__()
        self.myButton = QtWidgets.QPushButton(self)
        self.myButton.setObjectName("myButton")
        self.myButton.setText("Test")
        self.myButton.clicked.connect(self.msg)

    def msg(self):
        directory1 = QFileDialog.getExistingDirectory(self,
                                                    "选取文件夹",
                                                    "C:/") # 起始路径

        a = 'open a folder successfully'
        print(directory1)

        fileName1, filetype = QFileDialog.getOpenFileName(self,
                                                         "选取文件",
                                                         "C:/",
                                                         "All Files (*.*);;Text Files (*.txt)") # 设置文件扩展名过滤,注意用双分
号间隔

        b = 'open a file successfully'
        print(fileName1, filetype)

        files, ok1 = QFileDialog.getOpenFileNames(self,
                                                  "多文件选择",
                                                  "C:/",
                                                  "All Files (*.*);;Text Files (*.txt)")

        c = 'open several files successfully'
        print(files, ok1)

        fileName2, ok2 = QFileDialog.getSaveFileName(self,
                                                    "文件保存",
                                                    "C:/",
                                                    "All Files (*.*);;Text Files (*.txt)")

        d = 'save files successfully'
```

```

        return a, b, c, d

if __name__ == "__main__":
    import sys

    app = QtWidgets.QApplication(sys.argv)
    myshow = MyWindow()
    myshow.show()
    a, b, c, d = myshow.msg()
    assert_equal([a, b, c, d], ['open a folder successfully',
                                'open a file successfully',
                                'open several files successfully',
                                'save files successfully'])

    print('test finished')
    sys.exit(app.exec_())

```

文件的打开：

```

# coding:utf-8
from PIL import Image
import os
from nose.tools import *
import unittest

def open(filename):
    image = Image.open(filename)
    return True

if __name__ == '__main__':
    filename = '../pic/bg.png'
    value = os.path.exists(filename)
    #print('value:', value)
    if not value:
        raise ValueError('No this files')

    #print(open(filename))
    assert_equal(value, True)
    if open(filename)==True:
        print('test finished')

```

2.3 软件系统测试

2.3.1 界面测试

测试目标	对照界面规范，在软件需求规格说明书中规定，检查各界面设计是否规范，包括：界面风格、表现形式、组件用法、字体选择、字号选择、色彩搭配、日期表现、计时方法、时间格式、对齐方式等等，是否符合规范、是否协调一致、是否便于操作
测试方法和技术	让测试人员对现阶段版本进行使用，并提出修改意见
完成标准	所有测试用例都使用到，且系统中的功能全部都测试到
需考虑的特殊事项	无

2.3.2 功能测试

测试目标	确保功能测试需求项以及用例场景能够实现
测试方法和技术	利用有效等价类和无效等价类测试功能用例。 主要核实以下内容： 1.使用有效数据时得到预期的结果。 2.在使用无效数据时显示相应的错误消息或警告消息。
完成标准	所有测试用例都使用到，且系统中的功能全部都测试到
需考虑的特殊事项	无

2.3.3 性能测试

测试目标	针对系统的响应时间、延时、并发性、吞吐量等方面制定测试用例，分析结果确定系统性能。
测试方法和技术	同时让多个测试人员对系统进行使用，对每个不同性能进行测试
完成标准	在所有用例都使用完毕的情况下，无重大问题发生。
需考虑的特殊事项	无

2.3.4 强度测试

测试目标	在极端情况下出现的错误
测试方法和技术	在不间断重复工作或者处理过程发生断电的情况下发现可能的错误。
完成标准	在极端情况下允许引发一些错误
需考虑的特殊事项	无。

2.3.5 容量测试

测试目标	使用大量图片数据给软件以考验，以确定是否存在使用限制，若存在，是否引发软件错误。
测试方法和技术	通常通过增加处理图片的数量级，即 乘以 10 来确定“大量图片”的值，不断重复处理大量图片
完成标准	在不断处理大量图片的情况下，依然无重大问题发生。
需考虑的特殊事项	无。

2.3.6 安全性测试

测试目标	一方面验证 MAGIC-SCAN 图片编辑系统无法对 PC 机构成安全威胁，另一方面验证 PC 机无法对 MAGIC-SCAN 图片编辑系统构成使用威胁
测试方法和技术	检验 PC 机的杀毒软件是否会排查外来不明身份的安装程序；
完成标准	PC 机与 MAGIC-SCAN 图片编辑系统不会相互影响

2.3.7 安装测试

测试目标	核实在各种情况下，MAGIC-SCAN 图片编辑软件可正确地安装到各种计算机系统中，或从计算机卸载。
测试方法和技术	将系统放到 Windows /Ubuntu 下面并配置环境变量
完成标准	软件能够成功执行，没有出现任何故障。软件完全卸载，没有残余。
需考虑的特殊事项	无

2.3.8 配置测试

测试目标	确保本系统在有其他软件运行的情况下可以不受太大影响
测试方法和技术	在运行本系统的同时打开其他软件，如： Word 等
完成标准	程序运行时，无不可忍受程度的影响
需考虑的特殊事项	无。

2.3.9 破坏性测试

测试目标	输入错误的或非法的图片数据（类型、数量），检查系统报错纠错的能力及稳定性。并测试可连续使用多长时间而系统不崩溃
测试方法和技术	在系统中输入非法数据并提交
完成标准	系统对非法数据报告错误或者提示错误信息
需考虑的特殊事项	无。

2.3.10 可用性测试

测试目标	测试操作是否方便，用户界面是否友好等。测试系统是否有影响操作流程的界面 Bug 和功能 Bug，记录具体 Bug 的数量、出现频率和严重程度
测试方法和技术	让不同的测试人员对系统进行使用，并在系统中处理不同格式的图片
完成标准	系统对非法数据报告错误或者提示错误信息

三、 测试日志

预 定 系 统	测试人员	陈桂海、陈皖凤	所 属 项 目	MAGIC-SCAN 图片编辑系统
---------	------	---------	---------	-------------------

序号	测试项	操作步骤及现象	错误修改及原因简述	回测
1	图片融合术	只选择一张图片时， 3 秒后，系统闪退	原因：选择图片数量错误，应该选择两张 修改：选择两张图片	合格
2	打开图片	点击“文件”按钮， 再点击“打开”按钮， 无图片显示	原因：暂时未找到 修改：点击一下放大镜或缩小镜便能显示	合格
3	界面测试	在主界面底部， 无法显示使用时间	原因：暂时未找到 修改：暂无	不合格
4	剪切模块	点击“剪切”按钮无 反应	原因：未写剪切函数 修改：写上剪切函数并 connect 到信号槽	合格
5	形变模块	点击“形变”按钮无 反应	原因：未写形变函数 修改：写上形变函数并 connect 到信号槽	合格

四、测试总结

本人作为测试任务的负责人，在测试 MAGIC-SCAN 图片编辑系统的过程中发现了一些潜在的问题。但是我们通过黑盒测试与白盒测试后，进行了分析，并将这些测试过程一一记录下来。对于发现的错误，能修改的则修改，暂时不能修改的则保留，待后续进行更新改正。

目前 MAGIC-SCAN 图片编辑系统版本是最新的，已经修复了它的一些已知问题，系统运行比较可靠，功能比较齐全，性能也满足了预期要求。然而，整个系统因为开发的时间不够充足，能够提升与完善的地方还有很多，希望后续能把这个图片编辑系统弄得更好。