



## 一、项目分工

学号	名字	角色	班级	职责	贡献
14331358	张皓天	组长	早上班	负责 MVVM 的编写和图灵机器人网络 api 调用	25%
15331154	李景飞	组员	早上班	负责实现语音唤起和语音识别	25%
15331131	解晶璇	组员	早上班	负责实现分享功能和自适应界面	25%
15331117	黄楠绚	组员	早上班	负责实现密码加密、修改头像和磁贴功能	25%

## 二、开发环境

Windows 10 , Visual Studio 2015

## 三、项目阐述

1. 项目名称：OurChat

2. 简介：OurChat 是一款基于 Win10 系统开发的实用的 UWP 应用。用户可以向机器人查询天气、新闻等信息，是一个基于语音识别、人工智能的机器人。

3. 功能：

- 支持从“小娜”语音唤醒 OurChat 应用
- 支持注册时使用本地图片更换头像，注册信息保存到本地，下次登陆时可以保持头像不变
- 支持语音快速方便输入对话信息
- 支持和“图灵机器人”的对话
- 支持分享聊天的信息
- 支持添加磁贴

4. 亮点：

- 使用了以下知识点：

AdaptiveUI, DataBinding, DataBase, AppToAppCommunication,



NetWorkAccessing, FileManagement, LiveTiles

- 使用 MD5 对密码进行加密
- 语音启动和语音识别
- 根据内容动态识别列表项的模板
- 解析图灵机器人反馈的信息

## 四、项目展示

### (1) Adaptive UI

在 App.xaml 中为每个模板添加自适应屏幕宽度显示,当屏幕为窄屏时只显示对话信息,而将头像隐藏。宽屏时既显示头像又显示对话信息。



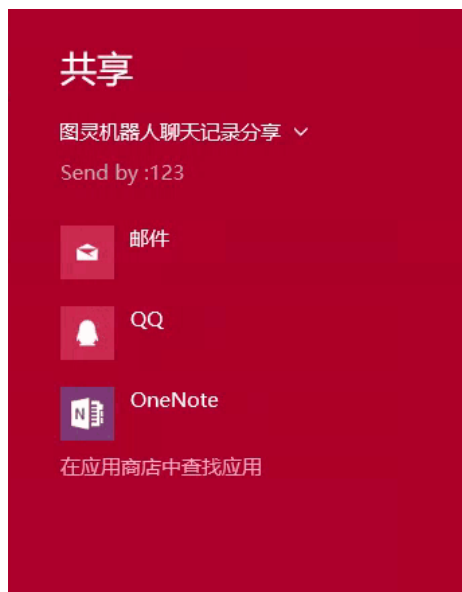


## (2) App to app communication

右键要分享的内容：



桌面右侧显示共享 UI，包含标题和简要信息：



点击邮件，显示要分享的内容信息：

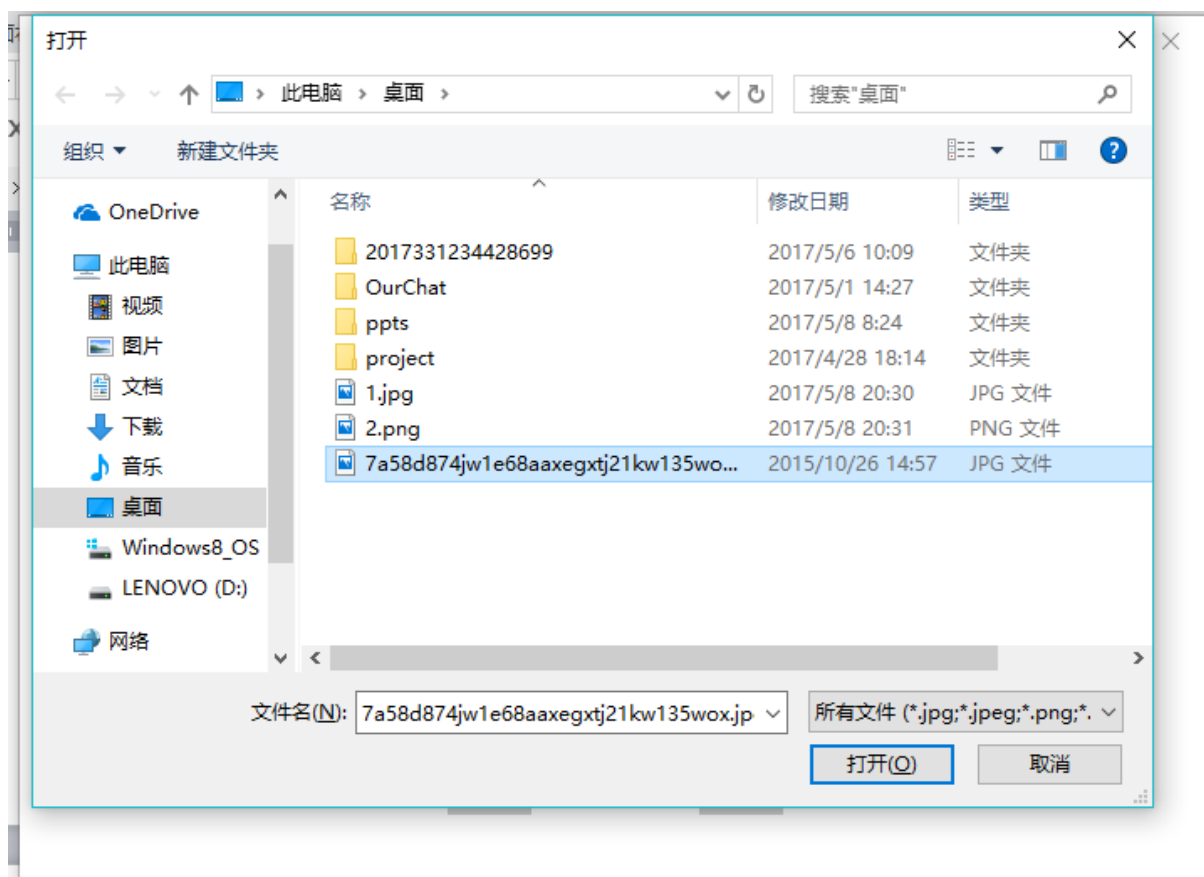


### (3) File management

用户进行双击默认头像后可以自行在本地磁盘中选取图片作为头像：



选取图片后：



可以看到头像更换成功：

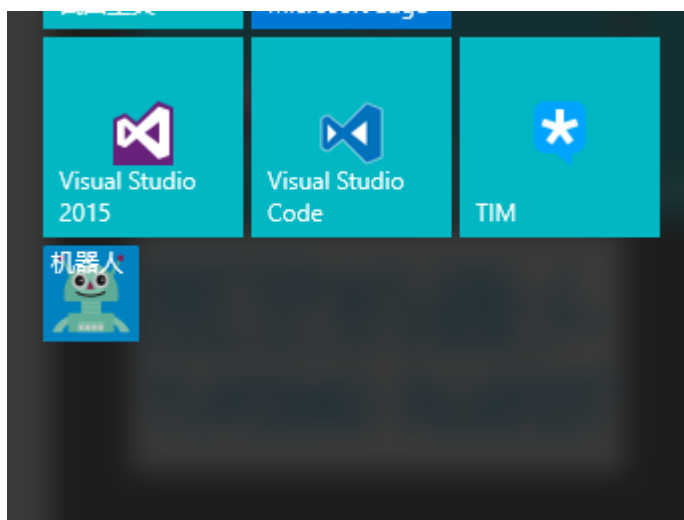


#### (4) Live tiles

磁贴有小中宽三种模式，中磁贴可以实现磁贴队列通知功能，可以滚动出现中英文文字

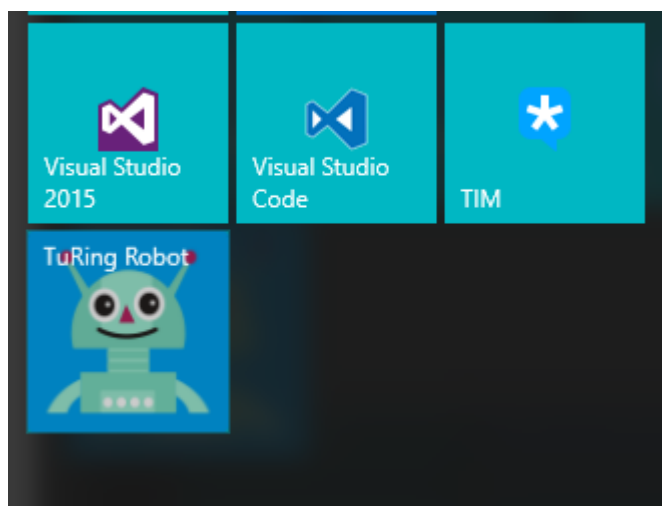
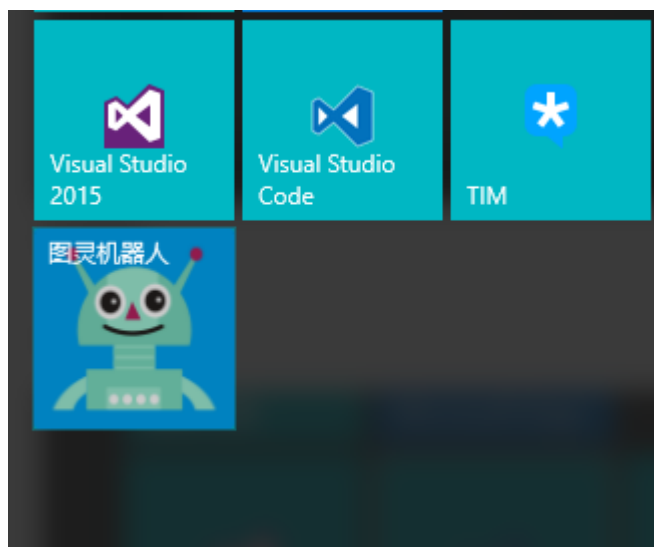
磁贴运行效果：

小磁贴：

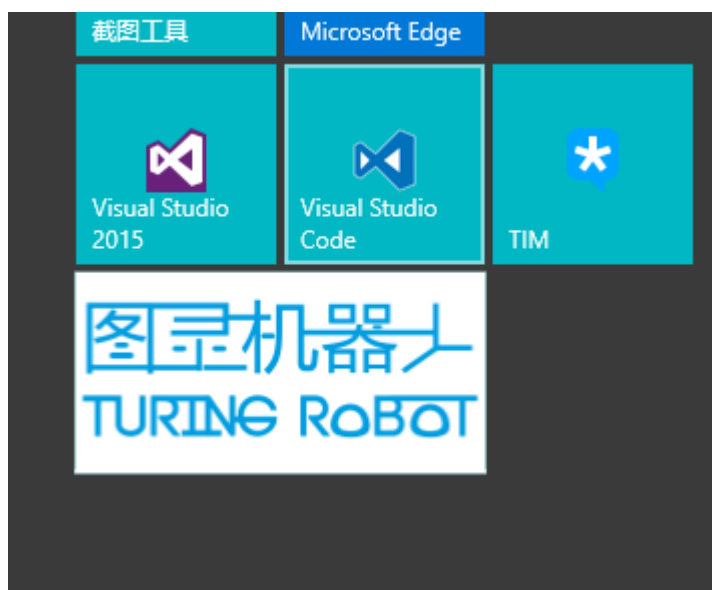




中磁贴：



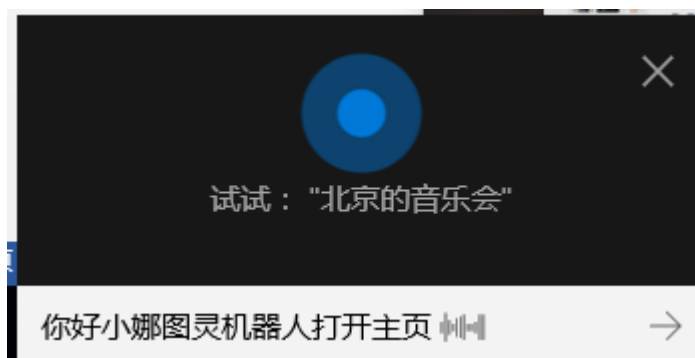
大磁贴：



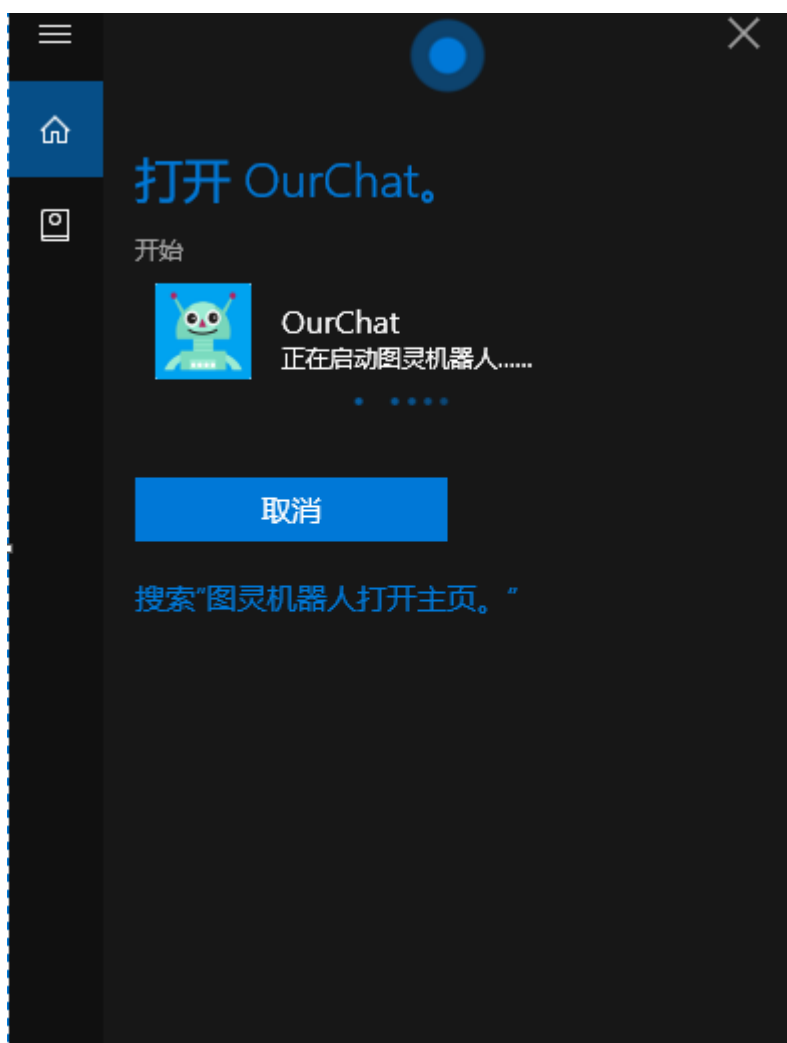
(5) 语音启动和语音输入：



从 cortana 唤醒图灵机器人



识别命令



跳转到 OurChat 主页面





# 图灵机器人 TURING ROBOT

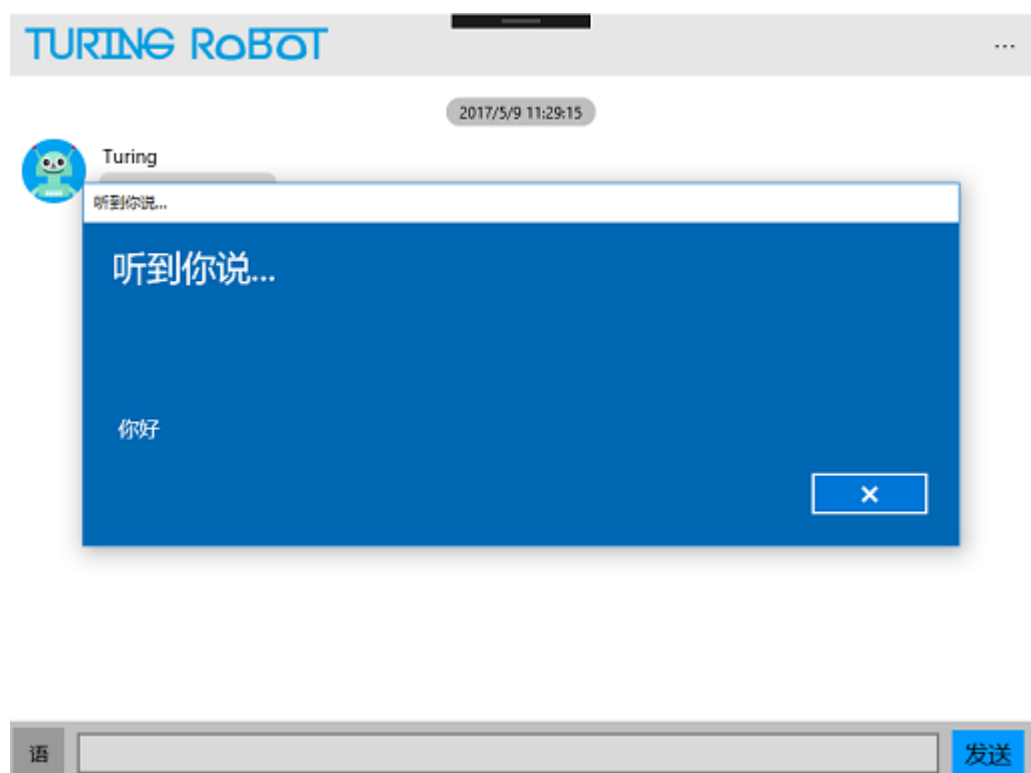
## 登录/注册

注册

登录

### (6) 语音输入

语音输入信息，反馈到文本框内然后等待修改



点击按钮或者 enter 键都会发送信息。如果是空信息 那么将不会发送出去。在按下 enter 键时，模拟点击发送按钮发送信息



TURING RoBOT

2017/5/9 11:31:15



Turing

你好，我是Turing

2017/5/9 11:31:25

lifi

你好



2017/5/9 11:31:25



Turing

我一点都不好呀

语

发送

## (7) 图灵机器人对话

查询天气



查询新闻



## 五、项目难点及解决方案

(1) 难点：针对不同的列表项，如何选择不同的列表项 UI 模版

解决方案：使用 `DataTemplateSelector` 继承的类来帮助选择不同的 UI 模版，我们的项目是根据数据内部不同的 `code` 属性值来判断使用什么列表项 UI 模版

```
class TemplateSelector : DataTemplateSelector {  
  
    protected override DataTemplate SelectTemplateCore(object item, DependencyObject container) {  
        if ((item as Content).name == "right") {  
            return App.Current.Resources["RightDataTemplate"] as DataTemplate;  
        } else if ((item as Content).name == "left") {  
            if ((item as Content).code == "100000") {  
                return App.Current.Resources["LeftDataTemplate"] as DataTemplate;  
            } else if ((item as Content).code == "200000") {  
                return App.Current.Resources["urls"] as DataTemplate;  
            } else if ((item as Content).code == "302000") {  
                return App.Current.Resources["news"] as DataTemplate;  
            } else if ((item as Content).code == "308000") {  
                return App.Current.Resources["cookbook"] as DataTemplate;  
            }  
        }  
        return base.SelectTemplateCore(item, container);  
    }  
}
```

(2) 当右击发生时，在当前点击位置创建一个浮动菜单并显示 Share 的文字。用户点击 Share 时，执行点击事件，并从后台 Data.cs 中获取点击对应的 ListView 的



Content 类中的信息。

```
// 获取当前点击的item中textBlock的文本
private void ItemsStackPanel_RightTapped(object sender, RightTappedRoutedEventArgs e)
{
    // 创建一个MenuFlyoutItem
    MenuFlyout myFlyout = new MenuFlyout();
    MenuFlyoutItem firstItem = new MenuFlyoutItem { Text = "Share" };
    myFlyout.Items.Add(firstItem);
    // 添加分享点击事件
    firstItem.Click += share_Click;
    FrameworkElement senderElement = sender as FrameworkElement;
    // 在当前点击位置显示Share选项
    myFlyout.ShowAt(sender as UIElement, e.GetPosition(sender as UIElement));
    // 从后台Data.cs中的Content类中获得其中的文本属性
    selectContent = (e.OriginalSource as TextBlock)?.DataContext as Content;
}
```

(3) 在实现语音唤醒时，由于开始不知道需要良好的网络环境，导致在语音唤醒的时候“小娜”都会跳转到网页搜索，浪费了大量的时间。

## 六、项目总结

我们小组在做本次实验中运用到之前课程上所学的大部分知识，也通过实现一些拓展的功能来学习一些新的技能。

在对用户密码进行储存的时候，学习使用 md5 加密，加强用户密码的安全性。在使用带有 selector 的 listview 的时候，也了解了如何动态选择列表项 UI 模版。

在实现键盘交互时，按下 enter 键的时候，它会先读入空行，所以一开始的时候发送信息时总有一个空行。尝试着在添加 KeyDown 事件，但是 KeyDown 事件必须在按下足够长时间才会触发，所以最后就在发送的时候使用 SubString 去掉了最后一个空行。

在实现代码的过程中，我们也学习到了应该考虑用户的各种操作，不能以自己已知的正确方法去测试它，例如用户在注册的时候可以不用更换头像，直接使用默认头像，确保在这种情况下可以注册成功，发送信息若是空消息则发送无效。解决各式各样的小细节之后，大大增强了程序健壮性。通过对这些细节的思考，我们优化了用户的体验，使项目更加完善了。