**1. 虚拟键盘类**

* **VirtualKeyBoard** (virtualkeyboard.h, virtualkeyboard.cpp)
  + 单例模式，负责虚拟键盘的全局管理和显示
  + 包含一个 LanguageLayout 对象，用于管理键盘布局
  + 提供 InstallKeyBoard 方法，用于安装虚拟键盘并监听输入焦点变化

**2. 键盘布局管理类**

* **LanguageLayout** (languagelayout.h, languagelayout.cpp)
  + 负责虚拟键盘的具体布局和按键管理
  + 包含一个 CPrivate 类，用于封装私有成员和方法
  + 管理按键的状态（如 Shift、Caps Lock）和语言切换
  + 与 CharKeyButton 和 KeyBoardButton 类进行交互，处理按键点击事件

**3. 按键基类**

* **KeyBoardButton** (keyboardbutton.h, keyboardbutton.cpp)
  + 继承自 QPushButton
  + 提供基本的按键功能，如设置按键码、处理按键按下和释放事件
  + 包含 CPrivate 类，用于封装私有成员和方法
  + 提供 CapsSwitch 和 ShiftSwitch 方法，用于处理大小写和字符切换

**4. 字符按键类**

* **CharKeyButton** (charkeybutton.h, charkeybutton.cpp)
  + 继承自 KeyBoardButton
  + 专门处理字符按键，如字母和数字
  + 提供 CapsSwitch、ShiftSwitch、ResetKey 和 LanguageSwitch 方法，用于处理按键的大小写、字符切换和语言切换
  + 使用 KeyBoard 命名空间中的字符映射表进行字符转换

**5. 字母按键类**

* **LetterKeyButton** (letterkeybutton.h)
  + 继承自 KeyBoardButton
  + 专门处理字母按键
  + 提供 CapsSwitch、ShiftSwitch 和 ResetKey 方法，用于处理字母按键的大小写切换

**6. 输入法管理**

* **KeyBoard** 命名空间 (inputmethod.h, inputmethod.cpp)
  + 提供输入法相关的功能，如模拟按键、获取和设置输入法状态
  + 包含 ClickKey、PressKey、ReleaseKey、GetKeyOpenState、IsCharKey、IsLetterKey 等方法
  + 提供 GetActiveIMMContext、GetConversionMode、SetConversionMode、GetLanguageState、IsIMEOpen、SetIMEOpenStatus 和 MicroSoftSwitch 等方法，用于管理输入法的状态和切换

