

# 布兰妮-v1



一个基于Live2D技术的智能桌面虚拟助手，支持聊天交互、语音识别、表情控制等丰富功能。

## 重要声明：

**客户端状态：**本项目当前仅为客户端程序，完整的AI服务端功能仍在开发构建中。部分智能交互功能需要等待服务端完成后才能正常使用。

**使用须知：**使用本项目前请务必仔细阅读下方的版权声明和许可证条款，确保合规使用。



## 项目架构说明

### 当前状态：客户端版本

本项目采用客户端-服务端分离架构设计：

- ☒ **客户端：** Live2D渲染、用户界面、本地交互功能已完成
- ☐ **服务端：** AI对话、语音识别、智能响应等功能正在开发中
- ☐ **通信机制：** 通过文件系统进行数据交换（`contact/` 目录）

**当前可用功能：** 模型显示、表情控制、音乐播放、基础交互界面

**待完善功能：** AI智能对话、语音识别处理、自动情感响应等

## 主要特性

### Live2D 动态展示

- 高质量Live2D模型渲染
- 丰富的表情动画系统
- 支持热键快速切换表情
- 透明背景窗口，完美融入桌面

### 智能交互系统

- 聊天模式：** 文字对话交互
- 语音模式：** 语音录制和识别
- 指令模式：** 执行系统指令和脚本
- 支持AI后端响应（需配置外部AI服务）

## 多媒体功能

- 音频文件播放支持
- 背景音乐播放（支持循环、随机等模式）
- 语音录制和处理
- 音效反馈系统

## 用户界面

- 无边框透明窗口
- 可切换的有边框模式
- 响应式UI布局
- 好感度进度条显示
- 对话气泡系统

## 快捷操作

- 表情热键映射
- 窗口模式切换（右键点击）
- ESC键随机表情
- F12调试模式切换

## 快速开始

---

### 系统要求

- Windows 10/11
- Qt 6.9.0 或更高版本
- MinGW 13.1.0 编译器
- 至少 4GB RAM
- 支持OpenGL的显卡

### 编译环境设置

#### 1. 安装Qt开发环境

```
# 下载并安装Qt 6.9.0
# 确保安装MinGW 13.1.0编译器
# 设置环境变量指向Qt安装目录
```

#### 2. 克隆项目代码

```
git clone <repository-url>
cd DesktopGirlClient
```

#### 3. 配置CMake

```
cmake -DCMAKE_BUILD_TYPE=Debug \
      -DCMAKE_MAKE_PROGRAM=C:/Qt/Tools/mingw1310_64/bin/mingw32-make.exe \
      -DCMAKE_C_COMPILER=C:/Qt/Tools/mingw1310_64/bin/gcc.exe \
      -DCMAKE_CXX_COMPILER=C:/Qt/Tools/mingw1310_64/bin/g++.exe \
      -G "MinGW Makefiles" \
      -S . -B cmake-build-debug
```

## 4. 编译项目

```
cd cmake-build-debug
mingw32-make
```

## 运行应用

1. 确保以下目录结构存在：

```
DesktopGirlClient/
├─ Resources/          # Live2D模型资源
├─ contact/            # 配置和通信文件
├─ asset/              # 图标和UI资源
├─ UserData/           # 用户数据
│   └─ music/          # 音乐文件
│   └─ script/         # 脚本文件
└─ cmake-build-debug/ # 编译输出
```

2. 将deploy.bat里面的windeployqt.exe的路径替换为你的Qt安装路径下的windeployqt.exe。
3. 运行可执行文件：

```
./DesktopGirlClient.exe
```

## 使用指南

### 基本操作

- **鼠标左键拖拽**：移动窗口位置
- **鼠标右键点击**：切换窗口边框显示/隐藏
- **ESC键**：触发随机表情
- **F12键**：切换调试模式

### 界面模式

#### 无边框模式（默认）



- 完全透明背景
- 仅显示Live2D模型
- 适合作为桌面宠物使用

#### 有边框模式




- 显示完整UI界面
- 包含控制按钮和交互组件
- 支持窗口大小调整

### 功能按钮


## 按钮1（显示/隐藏）

-  隐藏模式：隐藏所有UI组件
-  显示模式：显示所有UI组件

## 按钮2（模式切换）

-  聊天模式：文字输入对话
-  语音模式：语音录制交互
-  指令模式：执行系统指令

## 按钮3（指南）

-  打开热键映射文件
- 查看当前可用的表情热键

## 指令系统

在指令模式下支持以下命令：

### 音乐播放指令

```
# 播放默认主题音乐
playmusic

# 播放指定音乐文件
playmusic filename.mp3

# 循环播放音乐
playmusic -r filename.mp3

# 随机播放音乐
playmusic random

# 停止所有音乐
playmusic stop
```

### 脚本执行指令

```
# 执行指定批处理脚本
execute scriptname.bat
```

## 配置文件

### contact/config.json

```
{
  "favorability": 50,
  "isReacted": true,
  "modelName": "Nahida_1080"
}
```

- `favorability`: 好感度值 (0-100)
- `isReacted`: 是否已响应AI消息
- `modelName`: 当前使用的Live2D模型名称

## 热键配置

- 系统会自动加载模型对应的 `.vtube.json` 文件
- 支持A-Z和0-9数字键的表情映射
- 热键映射会保存到 `contact/keymap.txt`

## 高级配置

### 添加新的Live2D模型

1. 在 `Resources/` 目录下创建新的模型文件夹
2. 放入模型文件 (`.moc3`, `.physics3.json`, `.cdi3.json`等)
3. 创建对应的 `.vtube.json` 热键配置文件
4. 修改 `contact/config.json` 中的 `modelName`

### 热键配置示例

```
{
  "Hotkeys": [
    {
      "Action": "ToggleExpression",
      "File": "Happy1.exp3.json",
      "Triggers": {
        "Trigger1": "A",
        "Trigger2": "",
        "Trigger3": ""
      },
      "IsActive": true
    }
  ]
}
```

## AI后端集成

1. 将AI响应写入 `contact/in.txt`
2. AI处理后将结果写入相应文件
3. 系统会自动检测并响应

## 自定义

### UI主题

- 修改GLCore.cpp中的样式代码
- 自定义颜色、字体、布局等

### 音效替换

- 替换 `asset/Buttons/click_effect.mp3`
- 添加自定义音效文件

## 图标更换

- 替换 `asset/icon/app.ico`
- 更新按钮图标文件

## 故障排除

---

### 常见问题

#### Q: 编译时出现Qt相关错误

A: 检查Qt安装路径和CMakeLists.txt中的路径配置

#### Q: Live2D模型不显示

A: 确保模型文件完整，检查Resources目录结构

#### Q: 音频播放无声音

A: 检查音频文件格式，确保系统音频设备正常

#### Q: 热键不响应

A: 检查.vtube.json配置文件格式，确保表情文件存在

## 调试模式

- 按F12启用调试模式
- 查看控制台输出获取详细信息
- 检查 `contact/keymap.txt` 了解热键映射状态

## 版权声明与许可证

---

### 开源许可证

本项目基于 **MIT 许可证** 开源发布。详细条款请参阅项目根目录下的 LICENSE 文件。

### 重要法律声明

请在使用本项目前仔细阅读以下条款：

#### 1. 第三方库许可证

- **Qt 框架**：本项目使用了Qt动态链接库，使用时请遵守Qt相关开源协议（LGPL v3）
- **FFmpeg 库**：本项目使用了FFmpeg动态链接库，使用时请遵守FFmpeg相关开源协议（LGPL v2.1+）
- **Live2D Cubism SDK**：请遵守Live2D官方许可协议

#### 2. 模型版权

本项目中自带的Live2D模型仅用于展示项目功能效果，**版权归原作者所有**。商业使用或二次分发前请：

- 获得原作者明确授权
- 遵守相关版权法律法规
- 建议替换为自有或已获授权的模型

### 3. 🎧 多媒体资源版权

项目中包含的以下内容版权归项目作者本人所有，受版权法严格保护，**严禁任何形式的复制、转载、使用或分发**：

- 🎵 背景音乐和音效文件（包括但不限于 `asset/Buttons/click_effect.mp3` 等）
- 🖼️ UI设计和图标资源（包括但不限于 `asset/icon/`、`asset/Buttons/` 目录下的所有文件）
- 😊 美术设计元素（包括但不限于字体文件、图像素材等）
- 📄 文档和说明材料

**特别声明：**

- 这些资源**仅限于本项目内部使用**
- **禁止在任何其他项目中使用**，无论是商业用途还是非商业用途
- **禁止在开源项目中使用**，即使标注来源也不被允许
- 如需使用这些资源，必须事先获得版权所有者的明确书面授权
- 违反上述规定的行为将承担相应的法律责任

### 4. 🚫 使用限制

**严禁将本项目用于以下用途：**

- 任何违法违规活动
- 侵犯他人知识产权
- 传播不当内容
- 商业欺诈行为
- 其他损害社会公共利益的行为

### 5. 📄 免责声明

- 本项目按"现状"提供，不提供任何明示或暗示的担保
- 使用本项目所产生的任何风险由使用者自行承担
- 开发者不对使用本项目造成的任何损失承担责任
- 使用者应确保遵守所在地区的法律法规

### 6. 📧 版权投诉

如果您认为本项目侵犯了您的合法权益，请通过以下方式联系我们：

- 提供详细的权益说明和证明材料
- 我们将在收到有效投诉后及时处理

**使用本项目即表示您已阅读、理解并同意遵守上述所有条款。**

## 🤝 贡献

欢迎提交Issue和Pull Request来改进项目！

### 贡献指南

1. Fork本项目
2. 创建feature分支
3. 提交更改
4. 发起Pull Request

## 📞 联系方式

- 项目地址：[GitHub Repository]

- 问题反馈: [Issues页面]
- 开发者: Desktop Girl Studio



## 致谢与声明

---

### 开源项目致谢

- **Live2D Cubism SDK** - 提供Live2D模型渲染核心功能
- **Qt Framework** - 跨平台应用程序开发框架
- **FFmpeg** - 多媒体处理库支持
- **MinGW** - Windows下的GNU工具链

### 社区贡献

- 所有项目贡献者和用户的支持
- B站UP主見崎音羽对本项目构建提供的大量灵感和指导
- Live2D和Qt开发社区的技术支持

### 特别说明

**本项目中使用的Live2D模型、音效、UI素材等仅用于技术展示和学习交流，如有版权问题请及时联系处理。**

---

享受与虚拟桌宠的互动时光! 🐾🌸