

# CLiTBot 项目说明

2022-2023 学年程序设计小组作业

## 写在前面

---

这是清华大学 2022-2023 秋季学期程序设计基础课程大作业 CLiTBot 的项目说明。本次大作业项目旨在通过尝试复现一个经典的小游戏 LightBot，玩家需要在地图中通过一系列的指令操作一个机器人点亮场景中所有的灯以达成游戏目标。

在项目的实现过程中，我们在 GitHub 建有[项目仓库](#)，并应用 git 进行团队协作。鉴于本项目较为简单，并未启用更多相关功能。本项目采用 CMake 进行构建，具体编译环境配置方面可参考 README.md 中说明。在本项目的 release 页面也有预编译的游戏可供直接下载运行。关于项目实现过程中的更多细节还请参见下文。

## 小组人员情况

---

**黄砚星**：数学科学系数 11，学号：2021010277

**于展**：车辆与运载学院机械 209，学号：2022010343

**汪钰涵**：致理书院数 21，学号：2022012241

## 基本功能实现情况

---

### 公共部分

主要负责人：黄砚星

我们首先采用了面向对象的方法编写了整个项目中所需要用到的数据结构 (struct.h 文件)，如此更加契合 C++ 语言的风格，也更有利于项目的维护。我们项目所需的大多数常量都写到了 const.h 文件中，方便查看以及统一修改。游戏中类的定义也包含了所有可能用到的类方法的声明，加上相当充分的注释，struct.h 文件也同时起到文档的作用，方便查询与配合。

公共部分整体起到服务作用，为后续各模块的实现打下基础。

### 绘制模块

主要负责人：黄砚星

绘制模块的作用是读取当前地图中的信息，并通过文件操作将当前地图输出为 BMP 格式的图像并作为文件存储。这里我们没有依赖外部库实现，而是完全通过自行手动实现了相应函数的功能。我们通过数据结构体 pix 存储每个图像的 RGB 色彩数值信息，并使用 pix 的二维数组在内存中存储整个地图的图像信息。我们会在内存中对这个数组中的数据进行操作，此后整体写入硬盘到当中。我们的图像最终输出在 output/文件夹下，并且方格大小、配色等与示例有所不同，不过不必担心，所有功能都已正确完成。详细代码可以参见 src/imgdraw.cpp 文件中的内容。

### 执行模块

主要负责人：于展

执行模块的主要功能是执行游戏中的各项指令。其中包括 MOV, LIT, TL, TR, JMP 五项简单指令，也有玩家建立的 MAIN, P1... 这些指令序列。我们直接使用系统的调用栈，并通过字符串 string 处理指令的输入与比对。这些功能的实现基本是放在 runtime.cpp 文件当中的，这一模块的实现并不复杂，制作起来也相当快。

## 交互模块

主要负责人：汪钰涵

交互模块主要负责从命令行中接收用户输入并进行处理输出，实现交互。这里我们也是直接通过一个字符串变量 `m_cmd` 接收输入指令，并根据指令进行不同的操作。这个模块的代码分散在 `runtime.cpp` 以及 `interaction.cpp` 两个文件当中，包括后者的全部和前者的部分代码。交互模块的基本功能也全部完成实现。需要注意的是我们对于地图文件格式进行了一些适应性调整，不过指令序列格式并没有变动，对于具体的游戏过程不会有实质性影响。除去要求的指令外，我们还额外做了一个 `HELP` 指令用于输出提示，不过这一点影响不大。

上面我们完全完成了大作业文档中所要求的基本功能，后面是一些拓展功能的实现情况。

## 扩展功能实现情况

---

### 基于raylib引擎制作图形化游戏界面

`raylib`是目前较为流行的轻量级游戏引擎，我们在项目中将其引入，并以此实现图形化的游戏界面绘制。由于 `raylib` 引擎原本使用 C 语言编写，我们实际上使用的是其 C++ 封装`raylib-cpp`，以契合 C++ 语言面向对象为主的编程风格。

我们的整个图形化交互界面也采用 `raylib` 库编写，由于其底层通过 OpenGL 调用 GPU 进行绘制，整个界面的渲染效率也有所保障。我们通过 `raylib` 手动实现了文本标签、输入框、帮助按钮等交互界面元素，这使得整个游戏界面更加和谐。

最终，我们游戏的 GUI 版本可以通过左侧文本框输入指令并实时地观察运行结果，这一过程背后的执行仍是通过基础版本的执行模块进行的。我们也适应图形化游戏重写了绘制和执行模块，以实现动画效果等。相关代码写在 `imgdraw.cpp` 和 `runtime.cpp` 两文件当中，但是目前两块代码还未经过细致的检验，默认情况下并不启用。

本游戏中所使用的资源都放置在 `resources` 文件夹当中，除去大作业所提供的资源之外，游戏素材全部使用 Adobe Illustrator 绘制而成。

## 组内分工情况简介

---

本小组中由黄砚星担任队长，负责组织项目组内分工以及编写主体部分代码。在最初的分工中原定负责编写游戏的绘制模块。

在已完成的项目模块中：

黄砚星负责：

编写项目文件：`main.cpp`, `main_bs.cpp`, `main_ex.cpp`, `struct.h`, `constructors.cpp`, `imgdraw.cpp`, `const.h`, 多数基础部分代码的编写、整合调试，全部拓展部分代码；

引入 CMake 实现项目构建；引入 `raylib` 实现拓展功能；召开四次小组会议讲解项目架构；建立项目仓库并维护项目代码；建立并编写 Overleaf 文档；绘制游戏中所使用到的插图等。

于展负责：

编写项目文件：`runtime.cpp` 中的 `run` 函数和 `open_proc` 函数，`interaction.cpp` 中的 `op_info` 函数。

王钰涵负责：

编写项目文件：runtime.cpp 中的部分代码，interaction.cpp 中的大部分代码。

## 杂项

---

- 最终提交材料当中所附带的演示视频经过剪辑压缩，删节了不少内容，同时画面较模糊，很多细节如终端文字等都没法看清。我们在[清华云盘](#)当中存有原视频备份，如有需要可以前往查看。

## 总结

---

总结而言，本小组完全实现了大作业文档中所要求的基础功能。除此以外，我们通过引入外部库 raylib 制作完成了图形化的游戏界面作为拓展功能。最终我们的整体效果差强人意，主要问题是拓展模块中的部分代码没有来得及进行充分测试，最终选择了调用基础部分代码实现。不过最终我们仍然在完整实现基础功能的基础上实现了一定程度上的拓展……反正交差是够了（