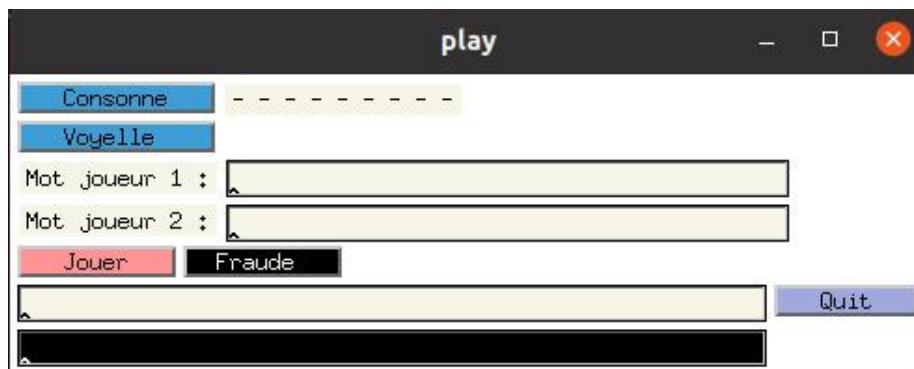


Documentation de Projet C: Mot le plus long! (GUI)

1. *libsX* 库构建的 GUI 界面

最
终
效
果



关于颜色的选择，我选择了几个我自己很喜欢的颜色，使用 Ps 软件提取了这几个颜色的 RGB 数值，并在程序中利用相关函数展现出来。

GUI 界面的构建和函数的选择：

<1>. **"MakeButton"**: 使用该函数以创建按钮部件。

<2>. **"MakeLabel"** & **"MakeStringEntry"**: 这两个函数分别创建了两种能打印文本的部件，同时第二个函数允许输入并读取。

<3>. **"GetRGBColor"** & **"SetBgColor"** & **"SetFgColor"**: 这三个函数用于产生固定 RGB 值的色彩，并给指定的 **"Widget"** 赋予相对的背景色和前景色（前景色在这里指文本的色彩）。

<4>. **"SetWidgetPos"**: 用于确定某个 **"Widget"** 的相对位置以确保 GUI 界面的整洁有序。

2. 游戏流程

① 在终端中输入 `./play` 以运行游戏程式。

② 第一步：玩家 1、2 轮流按下按钮 **Consonne** / **Voyelle** 以生成一个辅音字母或元音字母。由玩家 1 开始，交替进行，一共进行 9 次，得到 9 个字母。

③ 玩家 1、2 轮流在指定的文本框内输入一个由第一步得到的 9 个字母组成的最长的单词，请确保你的单词是最长的！

④ 第三步：按下按钮 **Jouer** 程序将自动进行判断玩家输入的单词是否都由第一步的 9 个字母组成，是否存在于字典中，并比较单词的长度；接着程序将自动声明游戏结果，获胜的玩家是谁（或者没有赢家）。

按钮 **Fraude**（作弊）如果您实在找不到一个单词由给出的 9 个字母组成，这个按钮将给你惊喜！（或者 NULL，没有惊喜）您可以按下 **Quit** 随时结束游戏。

3. 几个函数的实现

下面我会给出几个重要的函数及其说明

- 1) "`void motValide(str *d)`": 在这个函数中我使用 `C1C2C3C4` 代表玩家 1、2 的程序是否能够通过检查，是否存在于字典中。

`C1, C2, C3, C4=0/1`, 0 代表不通过, 1 代表通过。并最终交由判断的 `Jouer` 函数判定胜负情况。

- 2) "`void motValide_best(str *d)`": 在这个程序中代码的基本思路与检查玩家输入单词是否由 9 个给出字母组成的函数非常相似。需要注意的是，需要遍寻整个字典以确保选出的单词是最长的，而不是第一个符合条件的单词，还需要考虑字典中亦不存在符合条件的单词的可能。

4. *Quelques idées* 一些想法

当我编写这个程序的时候遇到了很多困难，首先是要学习 `libsX` 这个库的使用，实际上直到现在即便已经完成了程序要求的内容，这个库里面的有些函数仍然没有搞明白。

为了学习这个库的使用我在 GitHub 中下载了一个基于 `libsX` 库的简单游戏 (`ColorMatch`)，通过对他的源代码的学习，让我的整个编程过程便捷了很多。

在整个编程过程中需要大胆尝试不同的代码，并不断调试以获取最佳状态的代码，最终生成最优的程序。