

QT 大作业报告- Dig Your Money!

一. 程序功能介绍

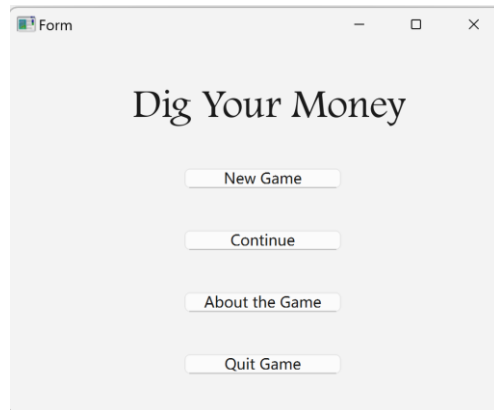
欢迎来到 Dig Your Money 小游戏!

这是一个轻松的放置类小游戏，灵感来源于 4399 里的同类游戏，外观朴实但是绝对好玩！

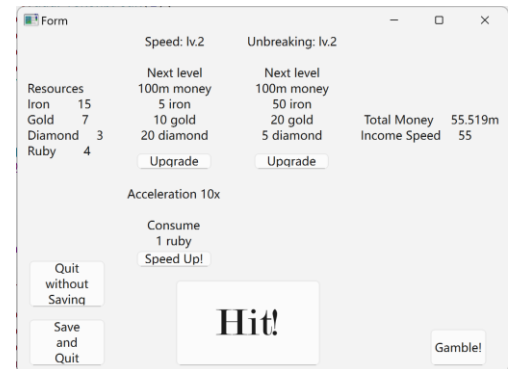
既可以挂机“赚钱”，又可以挖掘宝藏，获取稀有道具！

具体游戏规则与出货率详见[源代码 readme.md](#)

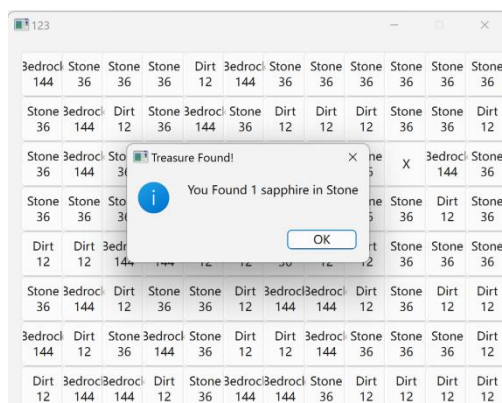
游戏截图：



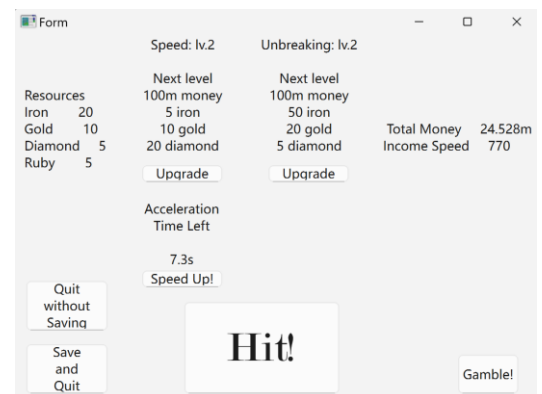
游戏标题界面



游戏主界面



夺宝界面。此处抽到了稀有蓝宝石
(4%概率)



红宝石加速启动

二. 项目设计细节

工程文件结构如右图。

主要类及其实现功能

1. TitleWindow 为游戏标题窗口类

主要实现开启新游戏，以及加载存档接续游戏。

2. PlayerData 为玩家数据类

无界面，以 `private` 形式储存玩家所有资源数据，使用 `inline` 函数进行外部读写，保证数据不被意外修改

3. MainWindow 为游戏主界面类，同时包含一个 PlayerData 对象存储核心游戏数据

主要实现：

(1) 保存游戏。将 `PlayerData` 类转化为数组形式，利用 `QTextStream` 输出储存在 `demo.txt` 中。

(2) 游戏状态实时更新。利用 `QTimer` 与 `connect` 函数，将 `GameTimer` 与 `MainWindow::updateGameEvents` 链接实现每 0.1 秒刷新一次游戏。利用 `PlayerData` 类的 `handleGameEvent` 方法更新资源数据，同时利用 `ui->labelAcc->setText(QString)` 刷新界面文字显示。

(3) 红宝石临时加速。单独设置一个 `AccTimer`，独立于 `GameTimer` 主计时器，以计时红宝石加速。同时利用 `xtimer` 全局变量代表加速启动时间，每一次刷新游戏增加 `xtimer` 值，实现伪倒计时。

(4) 小提示。当资源不够时点击升级按钮，会弹出 `QMessageBox` 实现的提示。

4. GameWindow 为夺宝界面类

主要实现：

(1) 方块阵列。该窗口未使用 `ui` 文件设计，而是直接利用 `QGridLayout` 纯代码排布 8*12 按钮阵列，便于批量产生方块。

(2) 单个方块。用 `Block` 类实现定义，具体数据由 `GameWindow` 生成。每个方块构造前会产生 3 个整型随机数 `q,a,b`，取值范围分别为 `[0,5]`, `[0,99]`, `[0,99]`。不同方块的随机数位于 `q,a,b` 取值范围上的位置分别决定了它的方块种类，宝藏种类，宝藏数。利用该方法可以实现不同方块和宝藏的概率不同，同时避免了一个方块有多个方块种类或者宝藏种类，缺点是代码会比较臃肿。

(3) 自动退出。当镐子耐久完全耗尽，会出现 `QMessageBox` 弹窗提示，随后界面自动关闭。

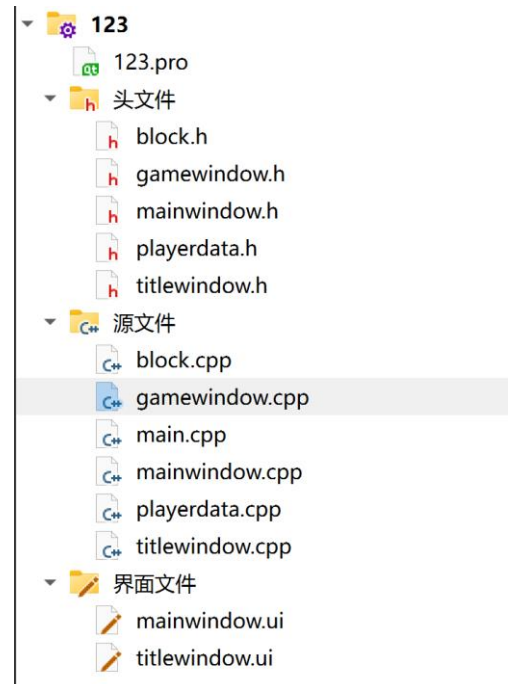
5. Block 为夺宝界面里的方块类

主要实现了方块的数据以及一些与方块有关的操作。

窗口之间的消息传递

1. TitleWindow->MainWindow

(1) 在点击 “New Game” 时，会创建一个全新的默认值 `MainWindow` 对象



(2) 在点击“Continue”时，会读取 demo.txt 的数据，转化为 PlayerData 结构，并将其传递给新建的 MainWindow 对象。

(3) MainWindow 建立后开启窗口并永久关闭 TitleWindow，此后直接在 MainWindow 中保存游戏

2. MainWindow->GameWindow

点击“Gamble!”按钮时，建立新 GameWindow 对象，同时将 PlayerData 里 PickUnbreak 与 PickSpeed（即挖宝界面所需的镐子耐久值及镐子速度）通过构造函数传递该 GameWindow

3. GameWindow->MainWindow

在 MainWindow 建立 int gl[7]数组，GameWindow 里声明 extern int gl[7]。gl[0]作为退出 GameWindow 的标记，防止传值被重复叠加；gl[1]~gl[6]接收来自 GameWindow 里挖到的各种宝藏的数量。这样可以避免退出 GameWindow 时相关数据被一并销毁。

三. 小组成员及分工

CuBr202 - 程序设计 & UI 设计 & 游戏机制设计

本游戏一切细节由我独立完成

（没错，我没找队友，这是我的单人作业）

四. 项目总结与反思

1. 下载 qt 就遇到了不少问题。我原本电脑上就安装有 qt4，但是 qt4 里没有 QGridLayout 类，导致只能重新下载 qt6。Qt5 的一个巨大改动使得 qt 只能在线联网安装，于是尝试从国内镜像源安装。由于不正确的代理设置，第一次用了 10 小时仅安装了 30%的文件。正确设置代理并且换源之后，用了一小时成功安装了新版 qt。

2. 由于时间不充分，最初做的项目是一个类似 Dig to China 的向下挖掘小游戏，但发现个人码力不够，无法实现背景的移动。结合本人曾经所玩的挂机小游戏，我决定将挖宝玩法与挂机玩法相融合，构造一个代码难度适中且有较强游戏性的小游戏。

3. 尝试询问 ChatGPT 构造代码框架，却发现其写出的代码全是用纯代码直接设计 UI，可配置性极差，因而决定按照官方文档以及相关教程，从 0 开始自己写。不过 ChatGPT 仍然给了我用纯代码写 GameWindow 这种需要批量定义按钮的界面的启示。

4. 本来想引用 Terraria 与 Minecraft 的材质作为挖宝界面方块以及资源的图标，可惜时间不够了。

5. 比较大的收获是将类的定义与实现分为 h 和 cpp2 个文件，这样虽然麻烦，但是在写多个类的工程文件时逻辑会更清晰。

6. 编写 PlayerData 类让我了解了类外读写 private 成员变量的方法

7. 编写 GameWindow 类让我熟悉了 private 函数的用法