

**软件体系结构作业19**

姓 名 ： 洪伟鑫

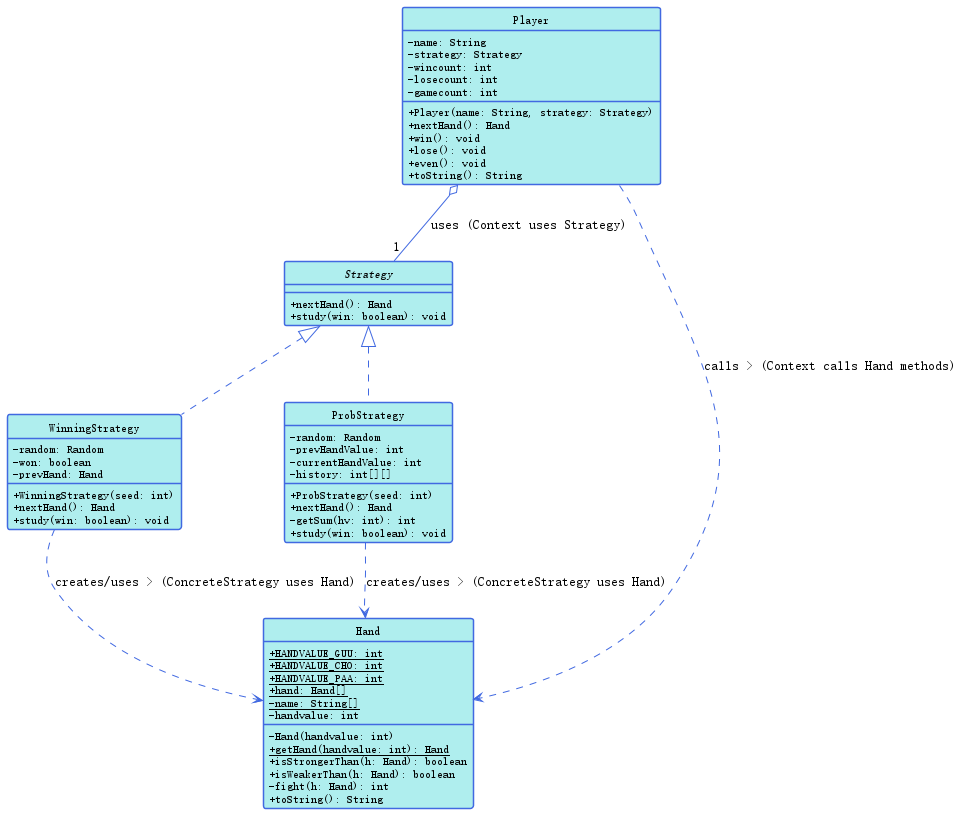
专 业 ： 软件工程

年 级 ： 2022级

学 号 ： 37220222203612

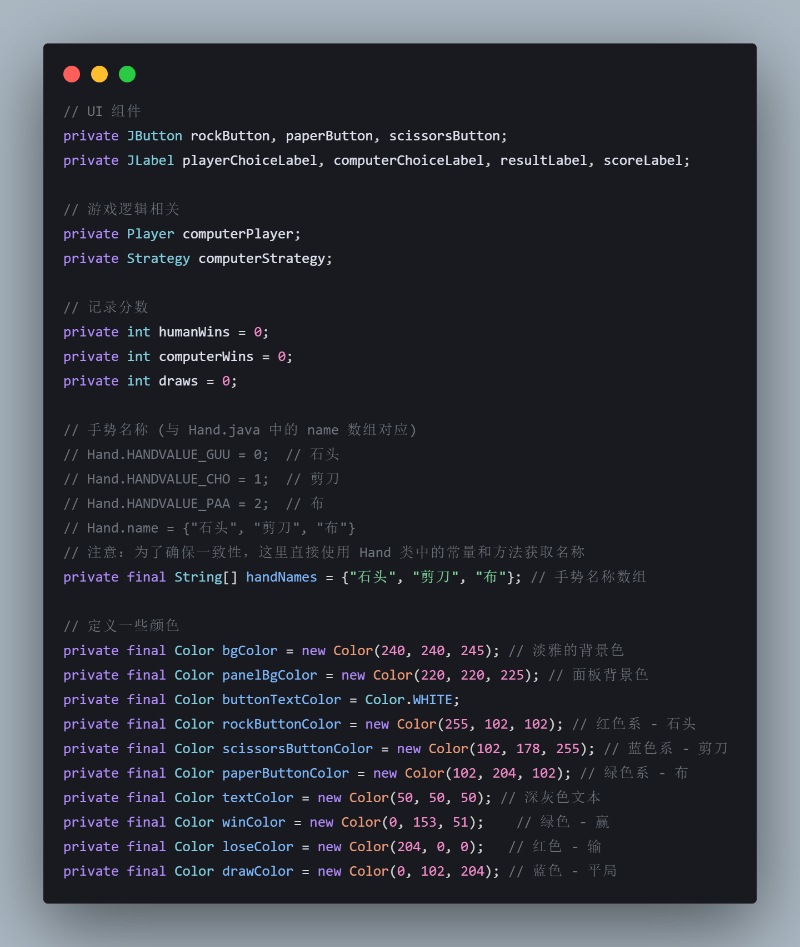
**2025年５月１７日**

**1、用GUI改写策略模式的例子。**

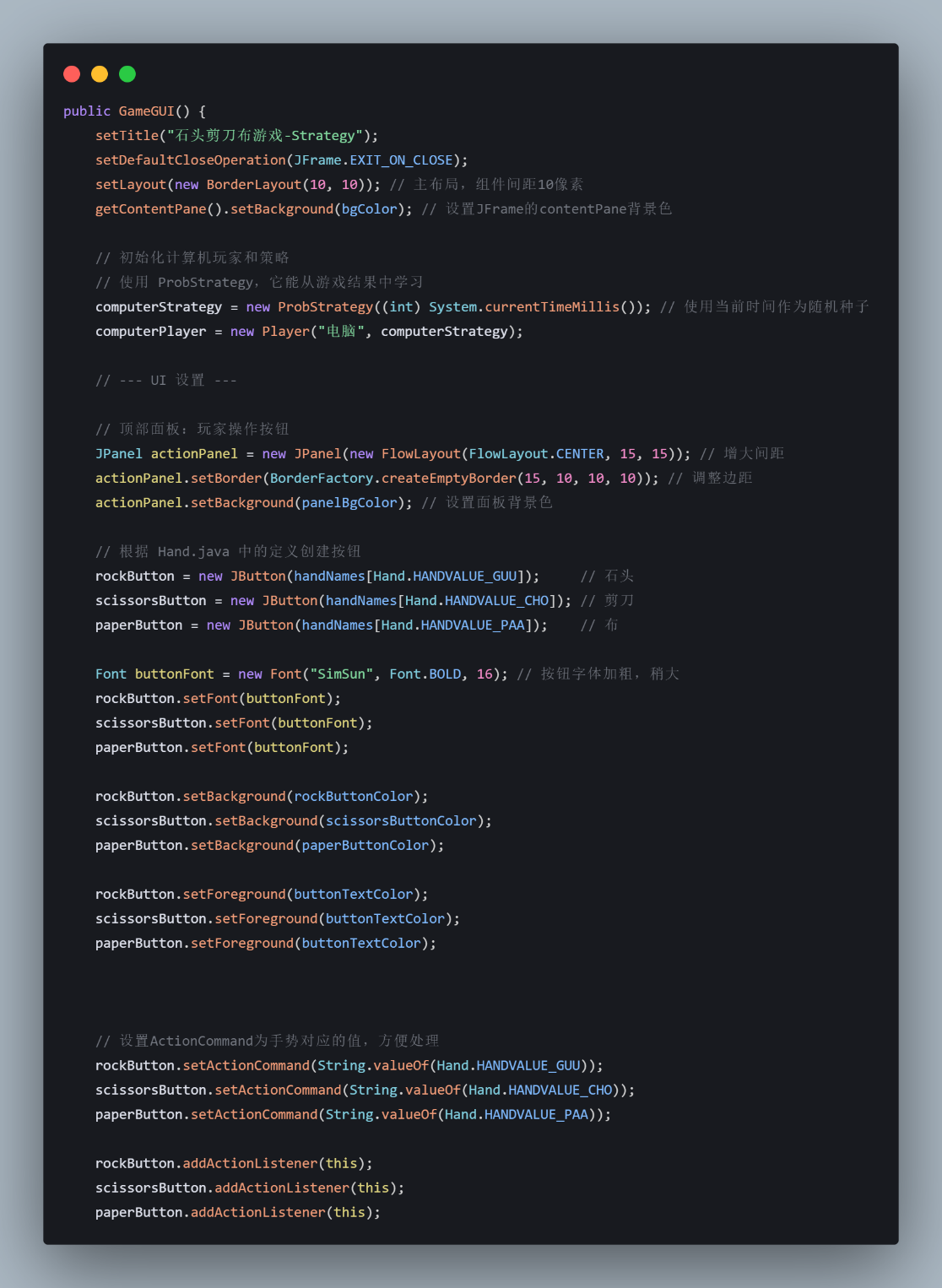


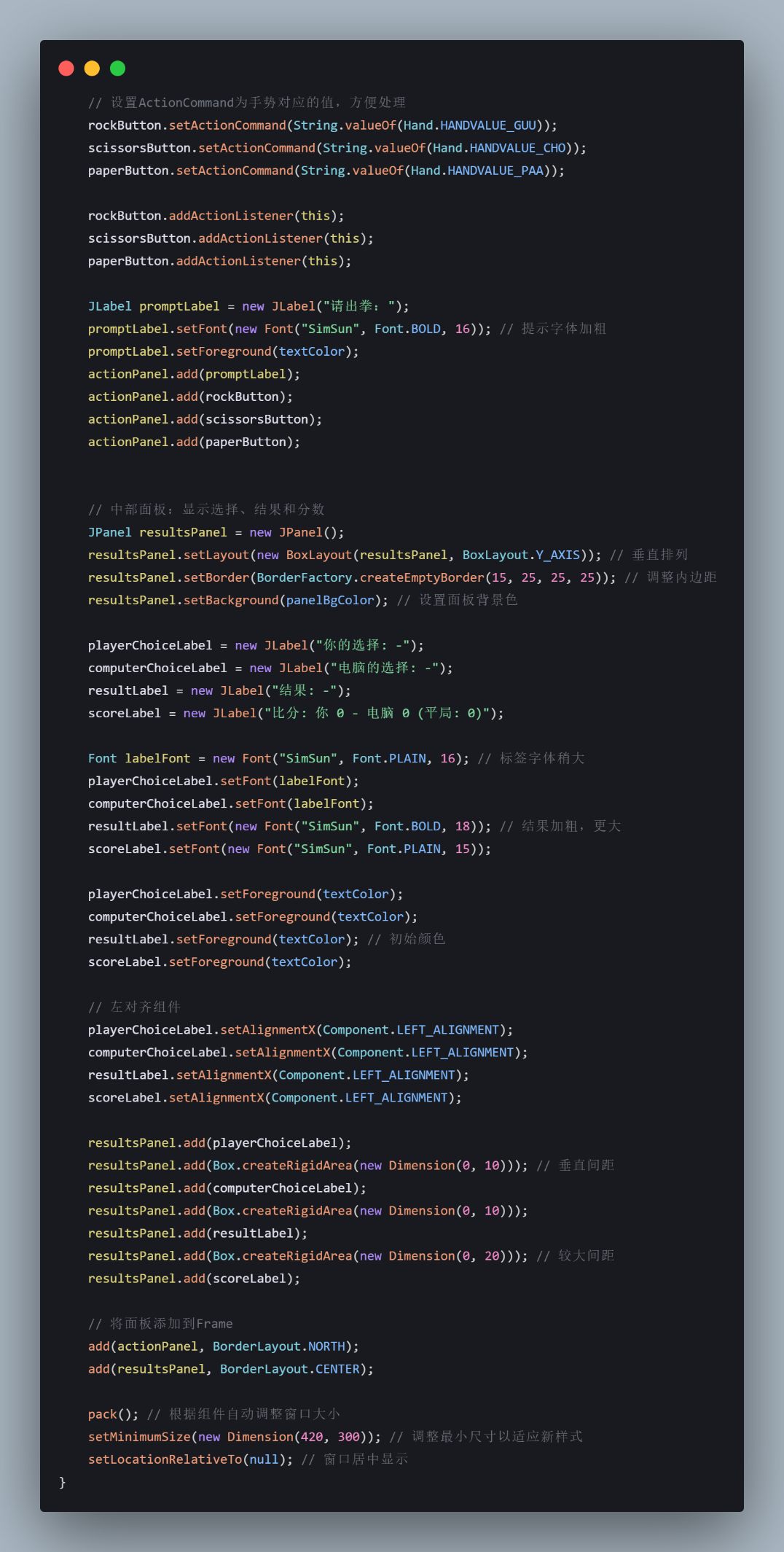
**策略（Strategy）设计模式的核心结构，应用于一个石头剪刀布游戏的场景。**

* **图中定义了一个 Strategy 接口，它规定了策略类必须实现的行为，即选择下一个手势（ nextHand ）和根据游戏结果进行学习（ study ）。**
* **WinningStrategy 和 ProbStrategy 是两个具体的策略实现类。**
* **Hand 类封装了游戏中的手势（如石头、剪刀、布）及其相互之间的克制关系。**
* **Player 类作为上下文（Context），持有一个 Strategy 对象的引用，并通过该策略来决定其出拳行为。**



导入相关的包，定义GUI要用到的各种颜色



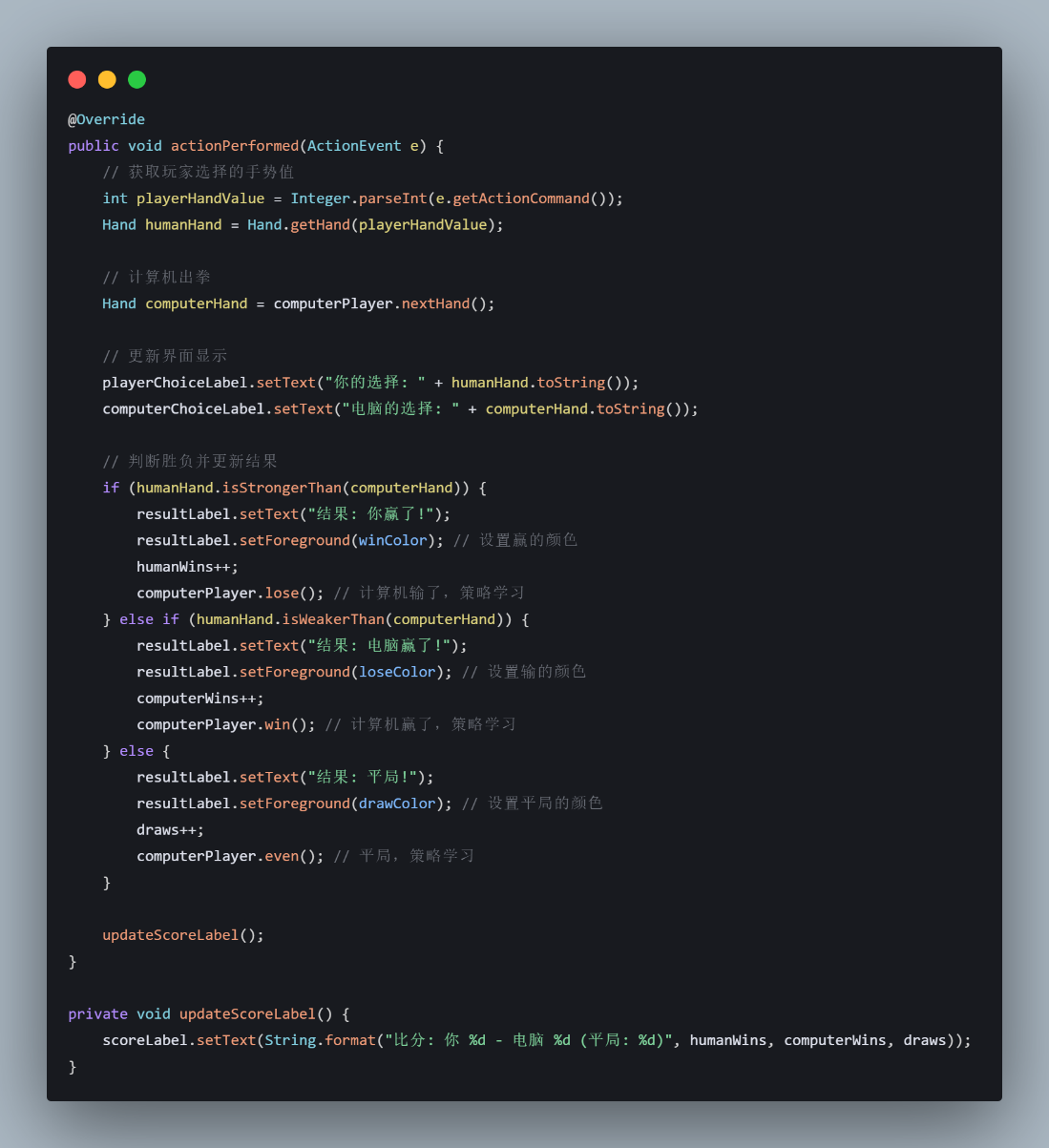


**这部分代码是 GameGUI 类的构造函数，它负责初始化整个游戏的用户界面。**

**首先，它设置了窗口的基本属性，如标题、关闭操作、布局和背景色。接着，它初始化了计算机玩家，为其分配了一个基于概率学习的策略（ ProbStrategy ），并使用当前时间作为随机种子。**

**然后，它开始构建界面元素：创建了一个顶部面板用于放置玩家的操作按钮（石头、剪刀、布），并为这些按钮设置了文字、字体、颜色、以及点击时触发的动作指令；同时，它还创建了一个中部面板，用于显示玩家的选择、电脑的选择、单局结果以及累计得分，并对这些显示标签的字体和颜色进行了设定。**

**最后，它将这两个主要面板添加到窗口中，并调用 pack() 方法自动调整窗口大小以适应内容，设置了窗口的最小尺寸，并让窗口在屏幕上居中显示。**



**这段代码主要处理用户在石头剪刀布游戏中点击按钮后的交互逻辑以及更新得分显示。**

**当用户点击代表石头、剪刀或布的按钮时， actionPerformed 方法会被调用。**

**它首先根据按钮的 ActionCommand 获取用户选择的手势，然后调用 computerPlayer.nextHand() 来获取计算机根据其当前策略所出的手势。**

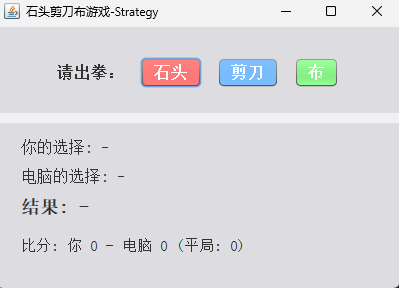
**接着，它会更新界面上显示用户选择和计算机选择的标签。之后，通过比较用户和计算机的手势，判断本局的胜负平关系，并相应地更新结果标签的文本和颜色（赢为绿色，输为红色，平局为蓝色）。**

**同时，它会累加用户赢、计算机赢或平局的次数，并调用计算机玩家的 win() 、 lose() 或 even() 方法，这些方法内部会调用策略对象的 study() 方法，以便策略能够根据游戏结果进行学习和调整。最后，它调用 updateScoreLabel 方法。 updateScoreLabel 方法则负责将最新的得分情况（包括用户赢的次数、计算机赢的次数以及平局次数）格式化并显示在界面上的得分标签中。**

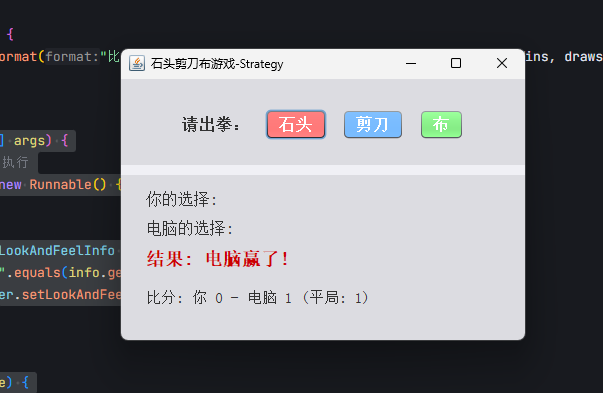


**Ｍａｉｎ函数创建 GameGUI 类的一个新实例，并通过调用 setVisible(true) 方法来显示游戏窗口，从而启动整个应用程序。**

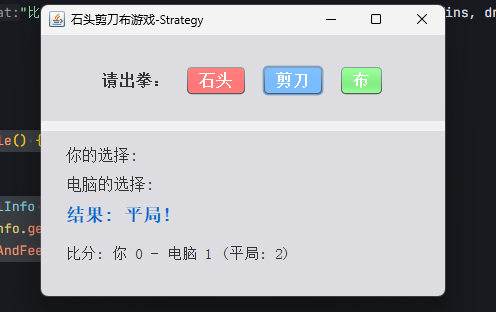
**运行结果：**

****

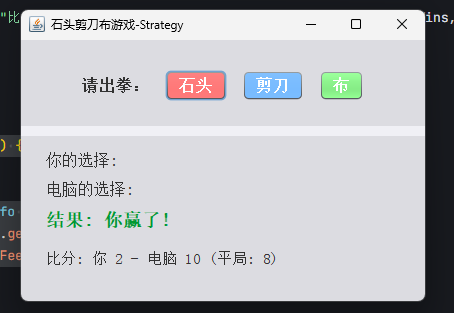
**初始界面**

****

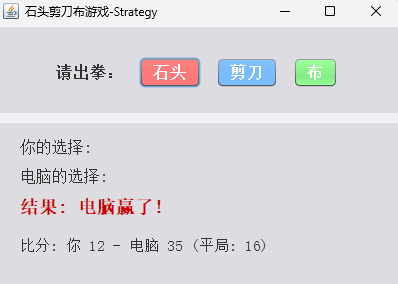
**电脑获胜**

****

**平局**

****

**玩家获胜**

****

**经过若干次出拳后的比分**