JAVA 课程设计

工程日志

张思远 程亚迪

目录

5 月 6 日～5 月 10 日

- 选题：
  - 24 点
- 任务分配
- 设计出题和解题算法：
  - random 随机题目，遍历运算
- 学习图形化界面开发相关内容

5 月 11 日

- 完成开始游戏界面的 demo：WelcomeWindow.java
  - 图片用临时图片代替。

5 月 12 日 15: 01

- 设计管理玩家的 Players 类：
  - 信息储存在 txt 文件中

```
private void checkFile() throws IOException{
    direction = new File(System.getProperty("user.home") + "/Saved Games/TwentyFour");
    file = new File(System.getProperty("user.home") + "/Saved Games/TwentyFour/users.txt");
    if (!direction.exists()){
        direction.mkdirs();
    }
    if (!file.exists()){
        file.createNewFile();
    }
}
```

  - 用户名和密码信息用哈希值的 16 进制字符串储存
  - 由于 16 进制不会产生 h 字符，数据之间用“h”分隔

```
// return -1 for name already exist
// return 0 for succeed
public int newPlayer(String name, char[] password) throws IOException{
    readFile();
    for (String i : lines){
        if (Integer.toHexString(name.hashCode()).equals(i.split("h")[0]))
            return -1;
    }
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {
        String pass = "";
        for (char i: password){
            pass += i;
        }
        for (String i: lines){
            writer.write(i +"\n");
        }
        writer.write(Integer.toHexString(name.hashCode()) + "h" + Integer.toHexString(pass.hashCode())+"h");
    }
    return 0;
}
```

- 设计登陆对话框的组件和行为

- 设计新用户对话框的组件

5 月 13 日 13: 07

- 遇到未知错误：WelcomeWindow.java 在 Code 中不被自动编译

```
siyuan@siyuan-WEI6-15-IML:~/Documents/CourseDesign$  /usr/bin/env /usr/lib/jvm/jdk-24.0.1-oracle-x64/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /home/siyuan/Documents/CourseDesign/bin App
Error: Unable to initialize main class App
Caused by: java.lang.NoClassDefFoundError: guiDesign/WelcomeWindow
```

  - 解决方法：尝试改名等，最终在 Eclipse 中打开后重新在 Code 打开，问题解决
  - 类 WelcomWindow 更名为 Welcome，包 guiDesign 更名为 gui
  - 推测原因：项目文件路径损坏
- 完善新用户对话框的行为
- 完善 Player 类：
  - 设计游客识别方式

```
public void setGuest() throws IOException{
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {
        writer.write("Guesth0");
    }
}
```

```
public void setPlayerHigh(String name, int score) throws Exception{
    readFile();
    ArrayList<String> tempLines = new ArrayList<>();
    if (name.equals("Guest")){
        for (String i : lines){
            if (i.split("h")[0].equals(name)){
              tempLines.add("Guesth"+score);
            }
            else
              tempLines.add(i) ;
        }
```

- 设计 CurrentPlayer 类存储当前玩家信息

```
package gui;
public class CurrentPlayer {
    private String name;
    private int highScore;
    public CurrentPlayer(){}
    public void setName(String name){
        this.name = name;
    }
    public void setHighScore(int highScore){
        this.highScore = highScore;
    }
    public String getName(){
        return name;
    }
    public int getHighScore(){
        return highScore;
    }
}
```

- 新增了游戏窗口的 JFrame：
  - 设计基本属性

5 月 13 日 16: 53

- 完善游戏窗口 JFrame：
  - 添加工具栏
  - 添加状态栏
- 新增关于（空）和首选项（空）

5 月 13 日 21: 09

- 设计两个模式：无限模式、计时模式，在原设计为游戏窗口的 JFrame 中添加新的 JPanel 用于选择游戏模式。

```
private void setSelectModePane(){
    JButton IButton = new JButton("无限模式"), tButton = new JButton("计时模式");
    JLabel iInfo = new JLabel("不限时间◦时间越短，得分越高◦"), tInfo = new JLabel("限时五分钟◦解题越多，得分越高◦");
    IButton.setFont(new Font("Arial", Font.BOLD, 30));
    tButton.setFont(new Font("Arial", Font.BOLD, 30));
    IButton.setBackground(Color.GRAY);
    tButton.setBackground(Color.GRAY);
    selectModePane = new JPanel();
```

5 月 14 日 0: 11

- 新增 GameResult 对话框类展示游戏结果（空）

5 月 14 日 13: 02

- 更新了 Players、CurrentPlayer 和状态栏来适配两个模式：
  - 设置两个最高分

```
public void setGuest() throws IOException{
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {
        writer.write("Guesth0h0");
    }
}
```

  - 将读取最高分方法的返回值设置成 int[]型

```
    public int[] getPlayerHigh(String name) throws IOException{
```

- 新增了检测数据损坏的方法，并添加了提示对话框和重设文件方法

```
private void resetFile(){
    var reset = new JDialog();
    initUI(reset);
    reset.setVisible(true);
}

private void initUI(JDialog reset){
    reset.setModal(true);
    reset.setTitle("错误");
    reset.setSize(280,190);
    reset.setLocationRelativeTo(null);
    reset.setResizable(false);
    reset.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
    var info = new JLabel("用户数据文件已损坏。");
    var button = new JButton("重设");
```

5 月 14 日 13: 39

- 更新了存储最高分的模式：
  - 需要在存储前和最高分比较。由于有 CurrentPlayer 中的存储高分，存储的方法中删去了比较的部分以减少多次读取文件的麻烦。

```
//change high in any case. must check high.
public void setPlayerHigh(String name, int score, int mode) throws Exception{
    readFile();
    ArrayList<String> tempLines = new ArrayList<>();
    if (name.equals("Guest")){
        for (String i : lines){
            if (i.split("h")[0].equals(name)){
              String[] temp = i.split("h");
                temp[mode + 1] = Integer.toString(score);
                tempLines.add(temp[0]+"h"+temp[1]+"h"+temp[2]);
            }
            else
              tempLines.add(i) ;
        }
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {
            for (String i: tempLines){
                writer.write(i+"\n");
            }
        }
        return;
    }
```

5 月 15 日 0: 47

- 根据原出题算法设计了新的算法：（遗留问题：类似方法是否可以归纳）
  - 讨论了性能上的可行性
  - 补充了原算法缺少的可能性，分五种结合方式，增强了出题的多样性

```
String way;
way = chanceOne(nums[i],nums[j],nums[k],nums[m]);
if (!way.equals("-1")) return way;
way = chanceTwo(nums[i],nums[j],nums[k],nums[m]);
if (!way.equals("-1")) return way;
way = chanceThree(nums[i],nums[j],nums[k],nums[m]);
if (!way.equals("-1")) return way;
way = chanceFour(nums[i],nums[j],nums[k],nums[m]);
if (!way.equals("-1")) return way;
way = chanceFive(nums[i],nums[j],nums[k],nums[m]);
if (!way.equals("-1")) return way;
```

  - 增加了去掉不必要的括号的算法

- 着手设计游戏界面:
  - 作为新的 JPanel 设计
  - 通过新 JLabel 设计计时器

```
public class Game extends JPanel{
    JLabel timer, cardStack;
```

  - 通过继承 JLabel 的类设计卡牌放置的框架

```
class CardContainer extends JLabel{
    public CardContainer(){
        super();
    }
    private void init(){

    }
}
```

5 月 15 日 14: 26

- 修改了计时器的设计思路:
  - 单独写一个继承 JLabel，实现 Runnable 的类（下），使用 javax.swing.Timer（右）来计算时间，防止精确度降低或线程阻塞

```
class Time extends JLabel implements Runnable{
```

```
@Override
public void run(){
    Timer timer = new Timer(1000, e -> {
        switch (mode) {
            case COUNT_UP:
                if (--currentSec == 60){
                    currentSec = 0;
                    currentMin ++;
                }
                setUI();
                break;
            case COUNT_DOWN:
                if (--currentSec < 0){
                    currentSec = 59;
                    if (--currentMin < 0){
                        currentMin = 0;
                        currentSec = 0;
                        setText("Time Up");
                        ((Timer)e.getSource()).stop();
                        break;
                    }
                }
                setUI();
                break;
        }
    });
    timer.start();
}
```

  - 使用 String.format 来格式化时间的显示方式

```
setText(timeMin + String.format(":%02d", timeSec));
```

- 增加游戏内容组件
  - 运算符号用 JComboBox<String>提供给玩家选择

```
for (JComboBox<String> i: ops){
    i = new JComboBox<>(new String[] {"","+","-","*","/"});
}
```

5 月 16 日 13: 24

- 修改了卡牌展示框设计思路:
  - 通过新建继承了 JLabel 的类来设置

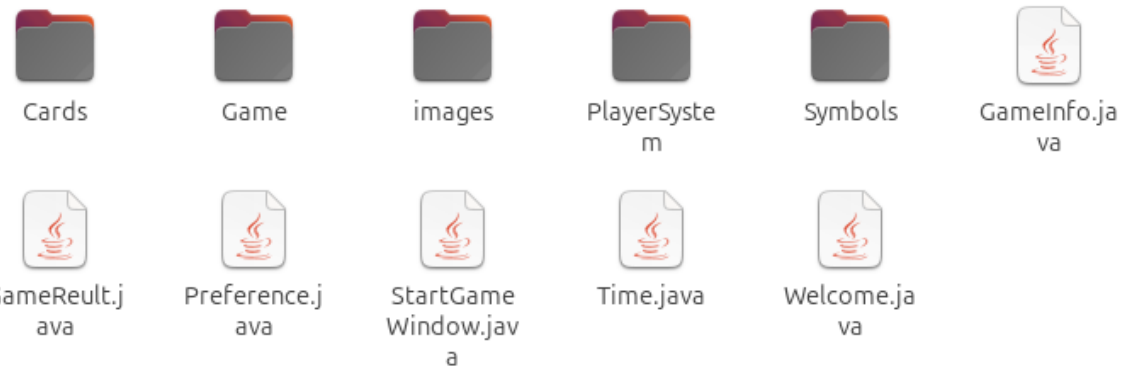
```
public class CardDisplay extends JLabel {
    private ImageIcon filledIcon, emptyIcon;
    //private Card...
```

- 新建了 Cards 类，封装 Card 类来存储图片和值

```
public class Cards {
    public Card[] cardsSet;
    public Cards(){
        cardsSet = new Card[13];
        for (int i = 0; i < 13; i++) {
            cardsSet[i] = new Card(null, i + 1);
        }
    }
    public class Card{ // public or private
        private ImageIcon icon;
        private int value;
        public Card(String filename, int value){
            icon = new ImageIcon(filename);
            this.value = value;
        }
        public int getValue(){
            return value;
        }
        public ImageIcon getIcon(){
            return icon;
        }
    }
}
```

5 月 17 日 1: 42

- 整理了项目结构，在包 gui 下新建了 4 个包



- 修改了运算符的设计思路：
  - 新建了继承 JComboBox<String>的类供用户选择运算符

```
public class Operators extends JComboBox<String>{
    public Operators(){
        super(new String[] {"    ","  +  ","  -  ","  *  ","  /  "});
        setFont(new Font("Arial", Font.BOLD, 24));
    }
}
```

- 新建了继承 JPanel 的 Pars 类供用户选择括号，其中包含两个 JToggleButton 可开关。

```
public class Pars extends JPanel{
    final static public int LEFT = 0, RIGHT = 1;
    public boolean[] selected;
    JToggleButton[] parButtons;
    private int side;
    public Pars(int side){
        this.side = side;
        initUI();
    }
```

- 将 Game 类更改为抽象类，而新建 IGame（后续新增 TGame）继承它

```
public abstract class Game extends JPanel{
```

```
package gui.Game;
import javax.swing.*;
public class IGame extends Game {
    public IGame(JFrame owner){
        super(INFINITE_MODE, 0, owner);
    }

}
```

- 设计了 Game 的布局 demo（GroupLayout）

```
        gl.setHorizontalGroup(gl.createParallelGroup(GroupLayout.Alignment.CENTER,
false).addGroup(sg1).addGroup(sg2).addGroup(sg3).addGroup(sg4).addGroup(sg5));

gl.setVerticalGroup(gl.createSequentialGroup().addGroup(pg1).addGroup(pg2).addGroup(pg3).addPreferredGap(LayoutStyle.Compo-
nentPlacement.UNRELATED).addGroup(pg4).addGroup(pg5));
    }
```

- 设计了拖动卡牌时的动作：（未测试）
  - 使用鼠标和鼠标动作监听器
  - 设置 offset 存储鼠标和 Component 左上角位置的差值
  - 将屏幕位置转换为窗口中的 point，再减去 offset

```
cardFrames[i].addMouseMotionListener(dragAdapter);
cardFrames[i].addMouseListener(releaseAdapter);
cardSelects[i].addMouseMotionListener(dragAdapter);
cardSelects[i].addMouseListener(releaseAdapter);
```

5 月 17 日 19: 08

- 基本完成组件的摆放，完成初始界面的 demo

- 修改了游戏计时器中的设计错误，将正数模式中的—改为++
- 重写了游戏界面的布局管理，使其在窗口放大缩小时正确放大、缩小间隙，并设置了最小窗口大小防止组件无法正常显示
- 整理了代码

5 月 18 日 01: 00

- 在 Windows 上进行了跨平台测试
- 设置了卡牌框架的大小，使其可以比原来缩小更多
    - 设置了卡牌图片随框架大小变化自动改变，防止卡牌图像显示不完整（ComponentResized 方法）

```
boolean[] leftSelected= {
    leftPars[0].getSelected()[0],
    leftPars[0].getSelected()[1],
    leftPars[1].getSelected()[0],
    leftPars[1].getSelected()[1],
    leftPars[2].getSelected()[0],
    leftPars[2].getSelected()[1],
    leftPars[3].getSelected()[0],
    leftPars[3].getSelected()[1],
};
boolean[] rightSelected= {
    rightPars[0].getSelected()[0],
    rightPars[0].getSelected()[1],
    rightPars[1].getSelected()[0],
    rightPars[1].getSelected()[1],
    rightPars[2].getSelected()[0],
    rightPars[2].getSelected()[1],
    rightPars[3].getSelected()[0],
    rightPars[3].getSelected()[1],
};
```

```
String[] opers = {
    (String)ops[0].getSelectedItem(),
    (String)ops[1].getSelectedItem(),
    (String)ops[2].getSelectedItem(),
    (String)ops[3].getSelectedItem()
};
```

```
int[] values = {
    cardSelects[0].card.getValue(),
    cardSelects[1].card.getValue(),
    cardSelects[2].card.getValue(),
    cardSelects[3].card.getValue()
};
```

5 月 19 日 0: 19

- 开始设计检查用户输入答案的方法:
    - 读取括号、运算符和数值

    - 判断括号是否成对，运算符是否都已选择

```
for (String i: opers){
    if (i.equals(anObject:"  "))
        return -28562; // 13 * 13 * 13 * 13 = 28561
}
```

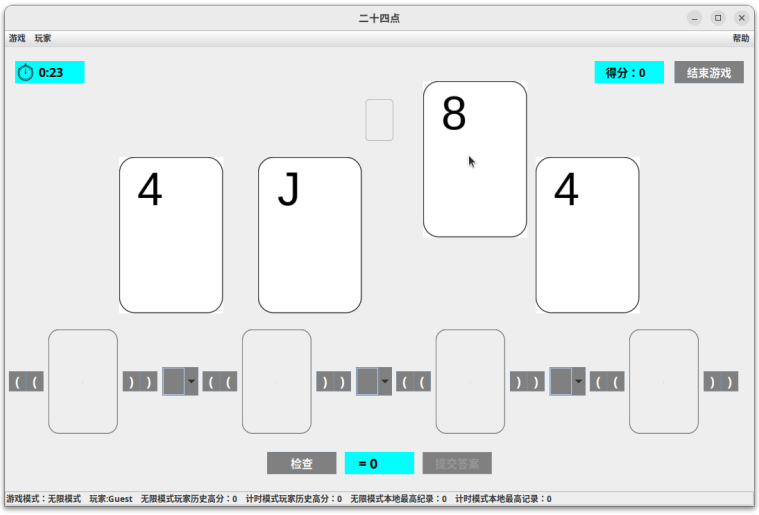
```
int rightIndex = 0;
for (int i = 0; i < 8; i ++){
    if (leftSelected[i] == false)
        continue;
    parPairSelected ++;
    for (int j = rightIndex; j < 8; j ++){
        if (rightSelected[j] == true){
            rightIndex = j;
            break;
        }
        if (j == 7){
            return -28563;
        }
    }
}
```

5 月 19 日 15: 01

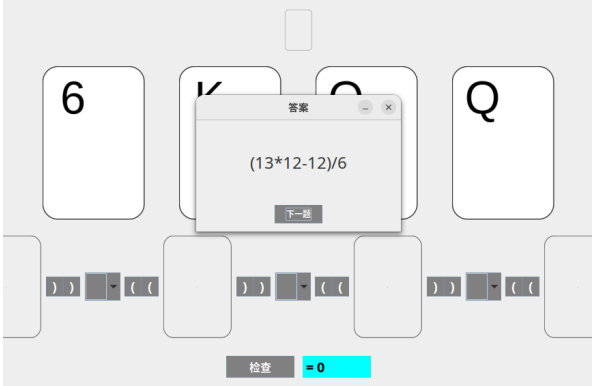
- 完善解析用户选择的方法:
  - 转换成字符串处理/直接处理?
  - 括号提升优先级是对于运算符而不是数值
  - 从运算符出发，左右寻找分别连续的左/右括号。运算次序增加两侧括号数的最大值*10
  - 把运算分成三部分：(A x B) (B x C) (C x D)装在 ArrayList 中
    - 计算优先级最高的运算
    - 把值给邻近的，优先级最高的运算
    - 本次运算结束后删除自身

5 月 19 日 18:33

- 测试解析用户选择的方法
- 新增了卡牌图片 demo，测试拖动和释放:
  - 释放后到回归有 1s 以内的迟滞




```
cardStack.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e){
        showKey();
    }
});
```



- 添加了修改图片大小的步骤以修复图片显示不完整的问题

5 月 19 日 20: 37

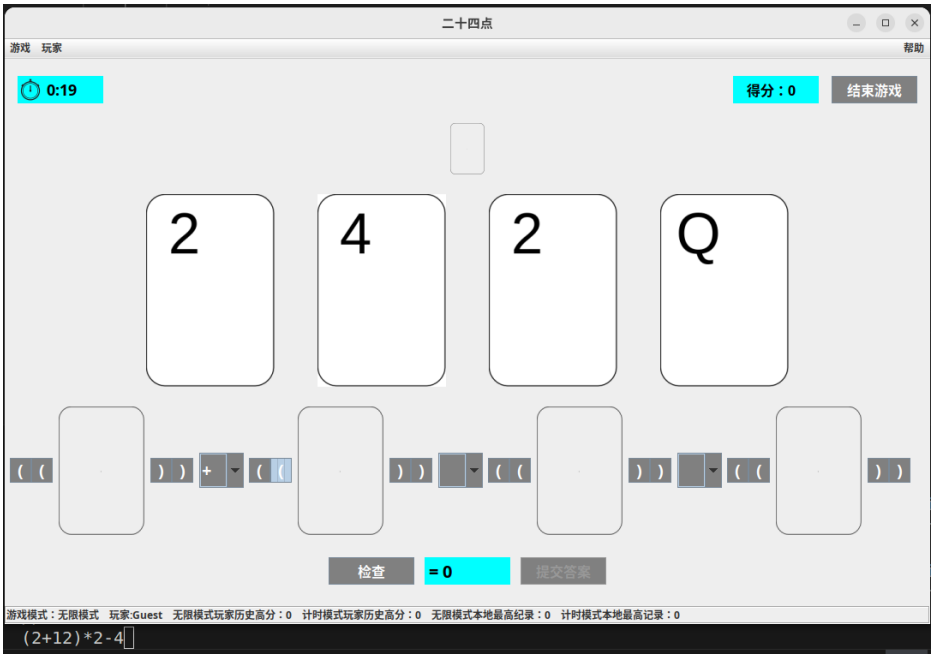
- 修复了总是按照顺序出题的问题:
  - 添加了打乱顺序的步骤

```
int[] res = new int[] {-1, -1, -1, -1};
boolean[] index = new boolean[] {false, false, false, false};
for (int i = 0; i < 4; i ++){
    int tempIndex;
    do {
        tempIndex = (int)(Math.random()*4);
    } while (index[tempIndex]);
    res[i] = nums[tempIndex];
    index[tempIndex] = true;
}
return res[0]+","+res[1]+","+res[2]+","+res[3]+","+solution;
```

- 修复了拖动存在的问题
  - 修改了判定落点的方式：使用 getComponentAt()而非由目标本身的监听器

```
if (dragging){
    Point releasingPoint = e.getLocationOnScreen();
    SwingUtilities.convertPointFromScreen(releasingPoint, Game.this);
    Component target = Game.this.getComponentAt(releasingPoint);
```

  - 添加 revalidate()解决了迟滞的问题。
  - 已经基本可以正常使用



5 月 19 日 21: 31

- 修复了游戏主体设计中出现的问题。

5 月 20 日 13: 24

- 实现了点击卡牌堆显示答案并进入下一题的功能

5 月 20 日 20: 07

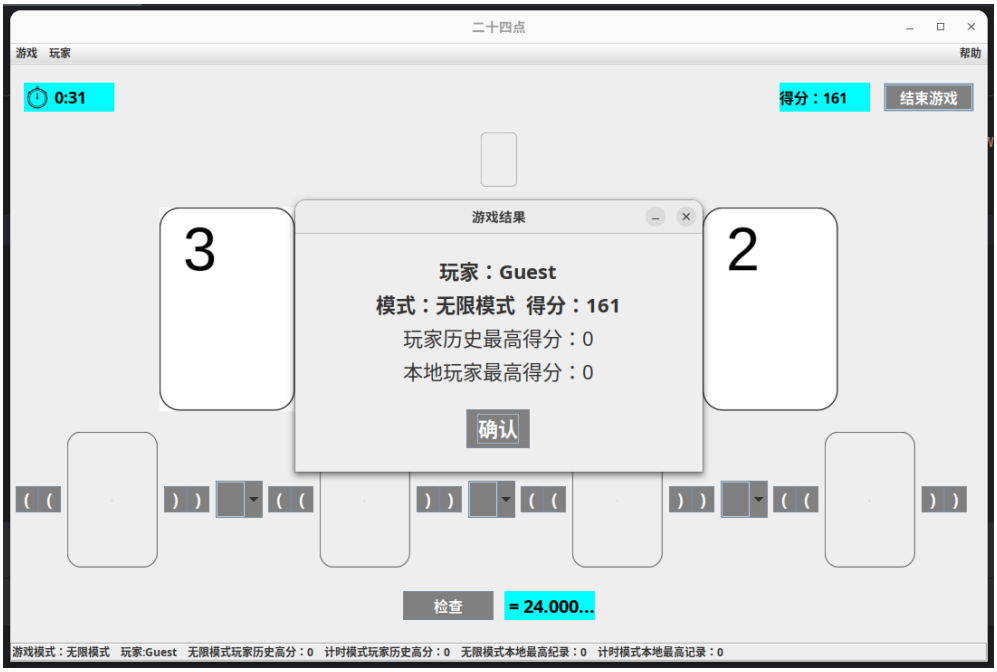
- 新增了计算分数的方法

```java
public void calculateScore(){  3 usages
    score = (gameTimer.getSecondPassed() != 0) ? (int)(1.0 * questionCompleted / gameTimer.getSecondPassed() * 5000) : 0;
    scoreLabel.setText(String.format("得分：%d    ", score));
}
```

5 月 21 日 02: 27

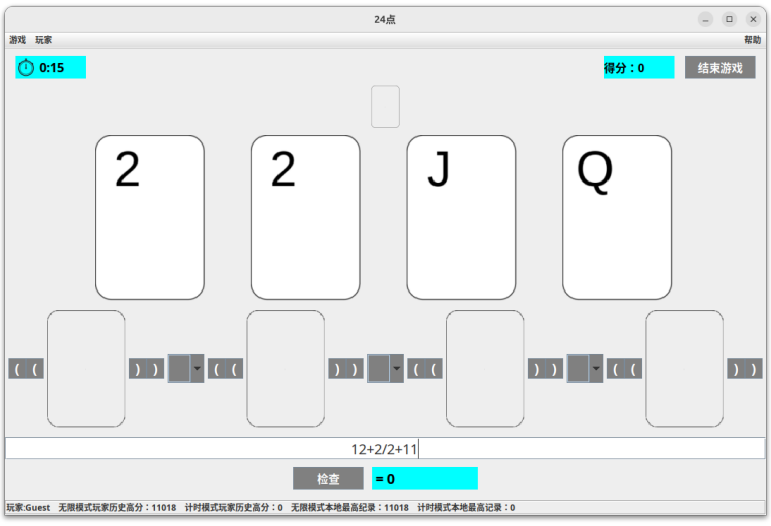
- 新增了游戏结束时的结算画面，以及游戏结束的逻辑

```java
public abstract void endGame(int source);
```



5 月 21 日 15: 25

- 修复了在 owner 被设为不可见时 JDialog 失去焦点后消失的问题：
  - 不把 JFrame 设成不可见
- 完善了游戏结束后回到主界面的步骤，修复了底部状态栏不更新的问题



5 月 22 日 Alpha 0.1

- 完成了课程设计项目的 demo，进入测试和美化阶段

5 月 28 日 Alpha 0.6

- 修改了游戏模式判定方式，减少重复代码

```
gamePane = switch (gameM){
    case 0 -> new IGame( owner: this, player);
    case 1 -> new TGame( owner: this, player);
    default -> throw new RuntimeException("WrongGameMode");
};
```

- 新增了被定义的组件顺序

```
Component[] components = getComponents();
cardFrameZOrder = 0;
otherZOrder = getComponentCount() - 1;
for (Component i : components){
    if (i instanceof CardFrame){
        setComponentZOrder(i, cardFrameZOrder ++);
    } else {
        setComponentZOrder(i, otherZOrder --);
    }
}
```

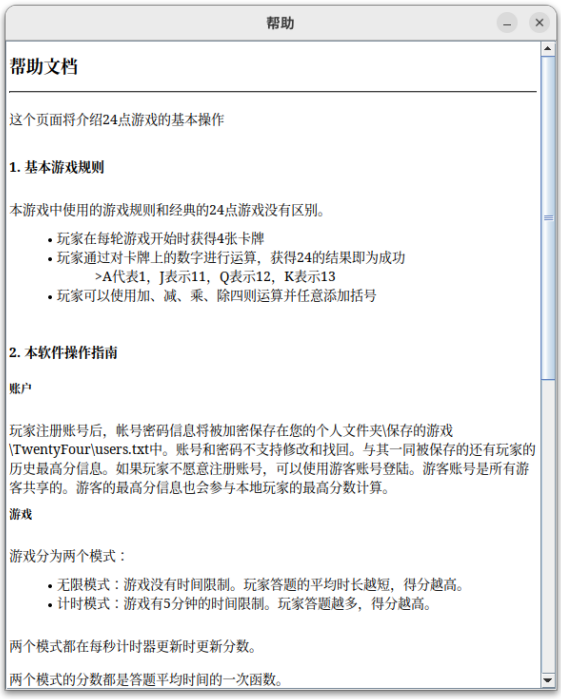
- 将平行组的 resizable 参数设置为 true，修复了卡牌不能正确放大缩小的问题。

```
var pg3 = gl.createParallelGroup(GroupLayout.Alignment.CENTER, resizable: true)
    .addComponent(cardFrames[0])
    .addComponent(cardFrames[1])
    .addComponent(cardFrames[2])
    .addComponent(cardFrames[3]);
var pg4 = gl.createParallelGroup(GroupLayout.Alignment.CENTER, resizable: true)
    .addComponent(leftPars[0])
    .addComponent(cardSelects[0])
    .addComponent(rightPars[0])
    .addComponent(ops[0])
```

- 修复了一定情况下可以凭空复制卡牌的 bug (Alpha 0.4)
- 修复了在登出时的幽灵计时器和残留的游戏面板问题

```
if (gamePane != null){
    finishTimer.stop();
    gamePane.endGame(Game.PLAYER_END);
    remove(gamePane);
    gamePane = null;
    gameM = -1;
}
startTimer.stop();
```

- 添加了帮助窗口



- 添加了用户信息浏览

- 修改了判断左右括号的方式，修复了括号错误判定的问题

5 月 30 日 1: 00

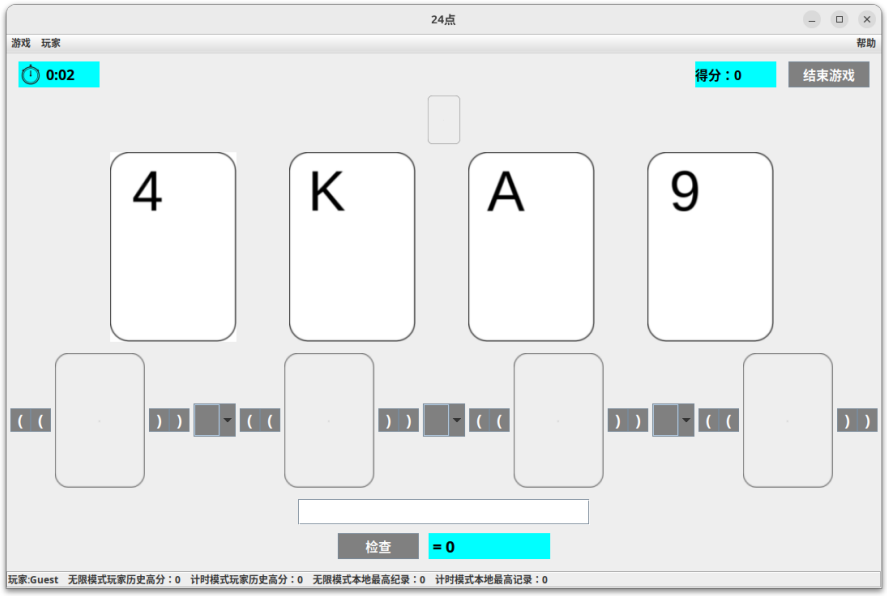
- 添加了输入框来接受答案以丰富操作体验，并添加了基本的支持

5 月 30 日 Alpha 0.7.0

- 稳定了输入框的支持
- 修复了一些已知的问题

5 月 31 日 Alpha 0.7.1

- 修改了输入框的外观，使其周围有一定空间



- 整理了代码，增加代码复用性
- 添加了注释，大幅减少了魔法数字，增加了代码的可读性

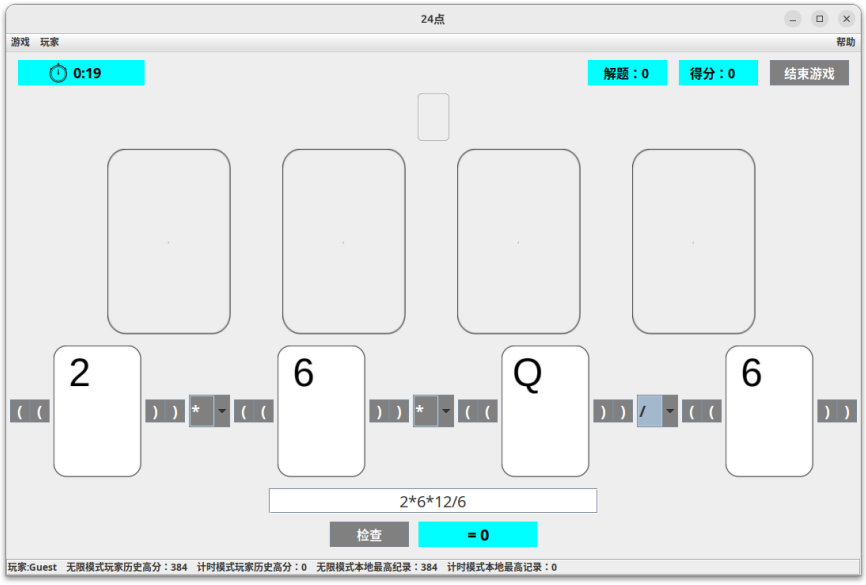
5 月 31 日 Alpha 0.7.2

- 增加了退出游戏界面的 Ctrl + Q 的快捷键（Accelerator）

- 更新了帮助内容

5 月 31 日 0.7.5 – alpha

- 修复了已知的问题
- 听取了玖玖女士的建议，增加的解出题目的数量 JLabel



- 调整资源路径为 getClass().getResource()，以打包成可执行 jar 文件运行


```
image = new ImageIcon(Objects.requireNonNull(getClass().getResource( name: "/gui/images/TitleImage.png")));
```

6 月 1 日 00: 07 0.7.6-alpha 儿童节更新

儿童节快乐！

- 优化了程序过程判定方法，减少 timer 的使用，大幅提升了性能和体验

6 月 1 日 15: 37

- 优化了代码结构
- 撰写了注释

6 月 3 日 21: 56 0.8.0-alpha Multi-thread

- 重构了游戏面板，将 Game 和 GamePane 分离，将过长的代码分担；将 GamePane 的构造器放入新线程中使其在后台获取新题目，大幅提升了性能：进入游戏按钮事件监听器执行时长从约 150 ms 减少到约 15 ms

  - 现：

```
iButton.addActionListener(( ActionEvent ignored) -> {
    long a = System.currentTimeMillis();
    game = new IGame(gamePane, player);
    startNewGame(gamePane);
    System.out.println(System.currentTimeMillis() - a + " ms");
});
```

```
/usr/java/jdk-
11+2+7+4
34 ms
8*(12-7-2)
20 ms
(13-10)*(9-1)
12/9*(11+7)
9 ms
(7+7)*2-4
10 ms
11*2-11+13
10 ms
```

  - 原：

```
iButton.addActionListener((ignored) -> {
    long a = System.currentTimeMillis();
    gamePane = new IGame(this, player);
    selectModePane.setVisible(aFlag:false);
    startNewGame(gamePane);
    System.err.println(System.currentTimeMillis() - a + "ms");
});
```

```
siyuan@siyuan
.1-oracle-x64
Design Applic
(7*2-6)*3
374ms
2*(11-10+11)
147ms
(7-2+1)*4
10/(8-3)*12
137ms
(12-6)/3*12
155ms
```

  - 注：运行一段时间后的效率提升由 JVM 的预热过程导致。结果显示，无论预热前后，更改后的性能都有大幅提升。

6 月 9 日 01: 49 0.8.5-alpha

- 将记录运算符的 double 数组改为 SubExpression 类，大幅提高了代码的可读性