1. Strategy GUI改写

1.1 目录结构

使用 **JavaFX** 进行GUI编写，项目结构如下：

> └─src
>
> └─sample
>
> App.java ——启动类
>
> Controller.java ——控制器类
>
> Hand.java ——出手
>
> Main.java ——启动方式类
>
> Player.java ——玩家类
>
> ProbStrategy.java ——Strategy类1
>
> sample.fxml ——GUI界面文件
>
> Strategy.java ——Strategy接口
>
> WinningStrategy.java ——Strategy类2

1.2 关键部分代码

实验报告中仅展示与PPT例不同部分代码

sample.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="447.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="sample.Controller">
   <children>
      <TextField fx:id="player1NameTextField" layoutX="113.0" layoutY="42.0" prefHeight="30.0" prefWidth="93.0" promptText="请输入名称" />
      <TextField fx:id="player2NameTextField" layoutX="113.0" layoutY="100.0" prefHeight="30.0" prefWidth="93.0" promptText="请输入名称" />
      <Button fx:id="playOneTimeButton" layoutX="394.0" layoutY="42.0" mnemonicParsing="false" onAction="#playOneTimeButtonPressed" text="玩一局" />
      <Button fx:id="playTenTimesButton" layoutX="394.0" layoutY="100.0" mnemonicParsing="false" onAction="#playTenTimesButtonPressed" text="玩十局" />
      <Button fx:id="resetButton" layoutX="504.0" layoutY="42.0" mnemonicParsing="false" onAction="#resetButtonPressed" text="重置" />
      <Label layoutX="52.0" layoutY="47.0" text="玩家1：" />
      <Label layoutX="52.0" layoutY="105.0" text="玩家2：" />
      <TextArea fx:id="resultShowTextArea" editable="false" layoutX="50.0" layoutY="164.0" prefHeight="262.0" prefWidth="501.0" />
      <TextField fx:id="seed1TextField" layoutX="239.0" layoutY="42.0" prefHeight="30.0" prefWidth="118.0" promptText="输入随机种子1" />
      <TextField fx:id="seed2TextField" layoutX="239.0" layoutY="100.0" prefHeight="30.0" prefWidth="118.0" promptText="输入随机种子2" />
   </children>
</Pane>
```

Controller.java

```
package sample;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;

public class Controller {
    @FXML private TextField player1NameTextField;
    @FXML private TextField player2NameTextField;
    @FXML private TextField seed1TextField;
    @FXML private TextField seed2TextField;
    @FXML private Button playOneTimeButton;
    @FXML private Button playTenTimesButton;
    @FXML private Button resetButton;
    @FXML private TextArea resultShowTextArea;

    // 判断所有输入是否有空
    private boolean isTextFieldEmpty() {
        String input1 = player1NameTextField.getText().trim();
        String input2 = player2NameTextField.getText().trim();
        String input3 = seed1TextField.getText().trim();
        String input4 = seed2TextField.getText().trim();
        if (input1.isEmpty() || input2.isEmpty() || input3.isEmpty() || input4.isEmpty()) {
            return true;
        } else {
            return false;
        }
    }

    // 判断输入的Seed值是否合法
    private int isSeedInputValid(String input) {
        int tmp = 0;
        try {
            tmp = Integer.parseInt(input);
        } catch (NumberFormatException e) {
            return -1;
        }
        return tmp;
    }

    // 玩一次
    public void playOneTimeButtonPressed(ActionEvent e) {
        if (isTextFieldEmpty()) {
            Alert alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("输入值不能有空");
            alert.setHeaderText(null);
            alert.setContentText("请确保玩家名、种子值不为空！");
            alert.showAndWait();
        } else {
            String player1Name = player1NameTextField.getText();
            String player2Name = player2NameTextField.getText();
            Integer seed1 = isSeedInputValid(seed1TextField.getText().trim());
            Integer seed2 = isSeedInputValid(seed2TextField.getText().trim());
            if (seed1 == -1 || seed2 == -1) {
                Alert alert = new Alert(Alert.AlertType.ERROR);
                alert.setTitle("输入格式错误");
                alert.setHeaderText(null);
                alert.setContentText("非法的Seed值，请重新输入！");
                alert.showAndWait();
                return;
            }
            StringBuilder result = new StringBuilder();
            result.append(String.format("玩家1：%s\n玩家2：%s\n", player1Name, player2Name));
            Player player1 = new Player(player1Name, new WinningStrategy(seed1));
            Player player2 = new Player(player2Name, new ProbStrategy(seed2));
            Hand nextHand1 = player1.nextHand();
            Hand nextHand2 = player2.nextHand();
            if (nextHand1.isStrongerThan(nextHand2)) {
                result.append("Winner：").append(player1).append("\n");
                player1.win();
                player2.lose();
            } else if (nextHand2.isStrongerThan(nextHand1)) {
                result.append("Winner：").append(player2).append("\n");
                player1.lose();
                player2.win();
            } else {
                result.append("双方平手\n");
                player1.even();
                player2.even();
            }
            result.append("总结果：\n");
            result.append(player1).append("\n");
            result.append(player2).append("\n");
            resultShowTextArea.setText(result.toString());
        }
    }

    // 玩十次
    public void playTenTimesButtonPressed(ActionEvent e) {
        if (isTextFieldEmpty()) {
            Alert alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("输入值不能有空");
            alert.setHeaderText(null);
            alert.setContentText("请确保玩家名、种子值不为空！");
            alert.showAndWait();
        } else {
            String player1Name = player1NameTextField.getText();
            String player2Name = player2NameTextField.getText();
            Integer seed1 = isSeedInputValid(seed1TextField.getText().trim());
            Integer seed2 = isSeedInputValid(seed2TextField.getText().trim());
            if (seed1 == -1 || seed2 == -1) {
                Alert alert = new Alert(Alert.AlertType.ERROR);
                alert.setTitle("输入格式错误");
                alert.setHeaderText(null);
                alert.setContentText("非法的Seed值，请重新输入！");
                alert.showAndWait();
                return;
            }
            StringBuilder result = new StringBuilder();
            result.append(String.format("玩家1：%s\n玩家2：%s\n", player1Name, player2Name));
            Player player1 = new Player(player1Name, new WinningStrategy(seed1));
            Player player2 = new Player(player2Name, new ProbStrategy(seed2));
            for (int i = 0; i < 10; i++) {
                Hand nextHand1 = player1.nextHand();
                Hand nextHand2 = player2.nextHand();
                if (nextHand1.isStrongerThan(nextHand2)) {
                    result.append("Winner：").append(player1).append("\n");
                    player1.win();
                    player2.lose();
                } else if (nextHand2.isStrongerThan(nextHand1)) {
                    result.append("Winner：").append(player2).append("\n");
                    player1.lose();
                    player2.win();
                } else {
                    result.append("双方平手\n");
                    player1.even();
                    player2.even();
                }
            }

            result.append("总结果：\n");
            result.append(player1).append("\n");
            result.append(player2).append("\n");
            resultShowTextArea.setText(result.toString());
        }
    }

    // 重置
    public void resetButtonPressed(ActionEvent e) {
        player1NameTextField.clear();
        player2NameTextField.clear();
        resultShowTextArea.clear();
        seed1TextField.clear();
        seed2TextField.clear();
    }
}
```

启动方式类Main.java

```
package sample;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));
        Scene scene = new Scene(root);
        primaryStage.setTitle("Hello World");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    // 此main函数无法使用，请使用App.java进行启动
    public static void main(String[] args) {
        launch(args);
    }
}
```

启动类App.java

```
package sample;

import javafx.application.Application;

public class App {
    public static void main(String[] args) {
        Application.launch(Main.class);
    }
}
```

1.3 运行演示

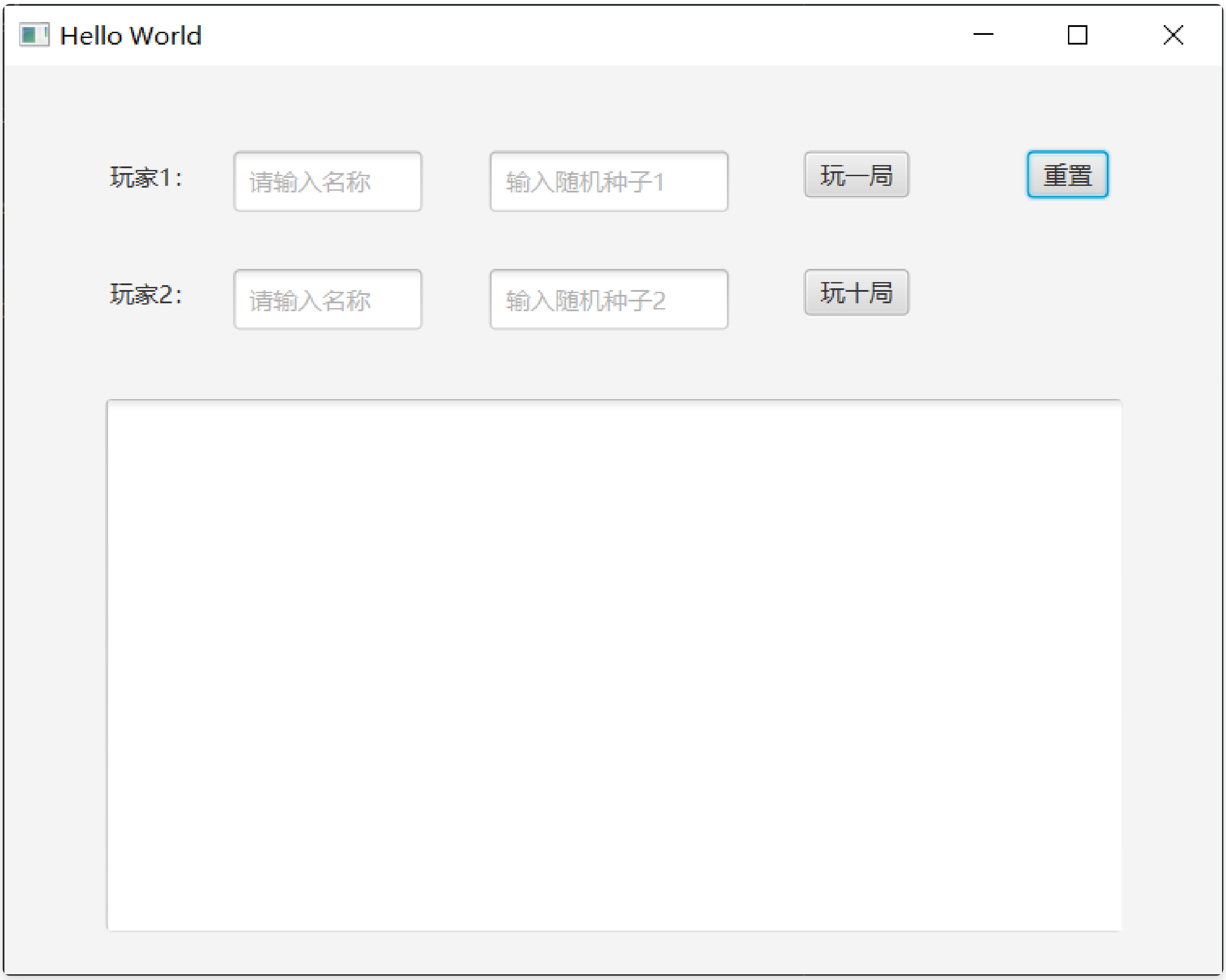
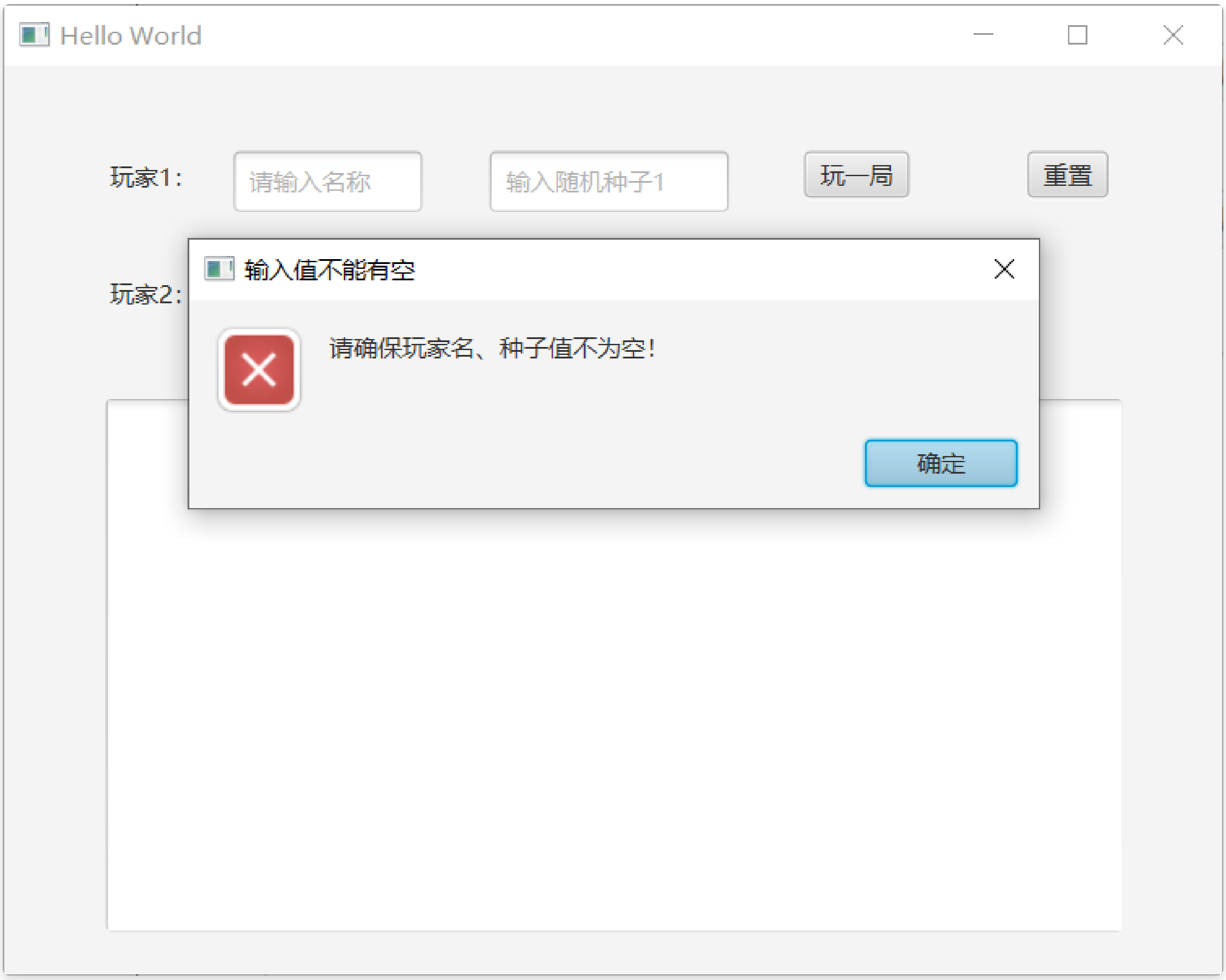


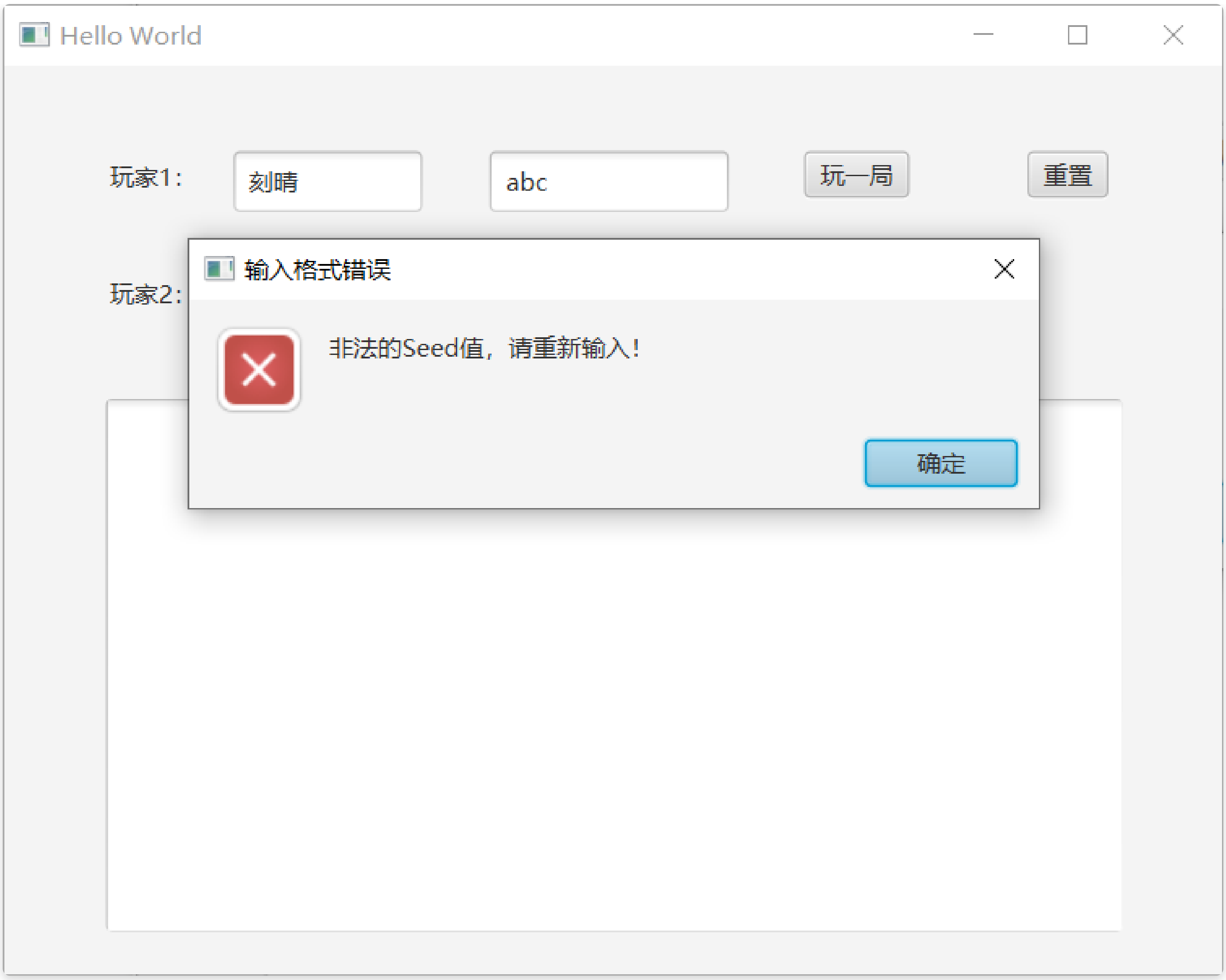
图1-3-1主界面截图



输入值不能有空

请确保玩家名、种子值不为空！

图1-3-2 输入控制判断提示



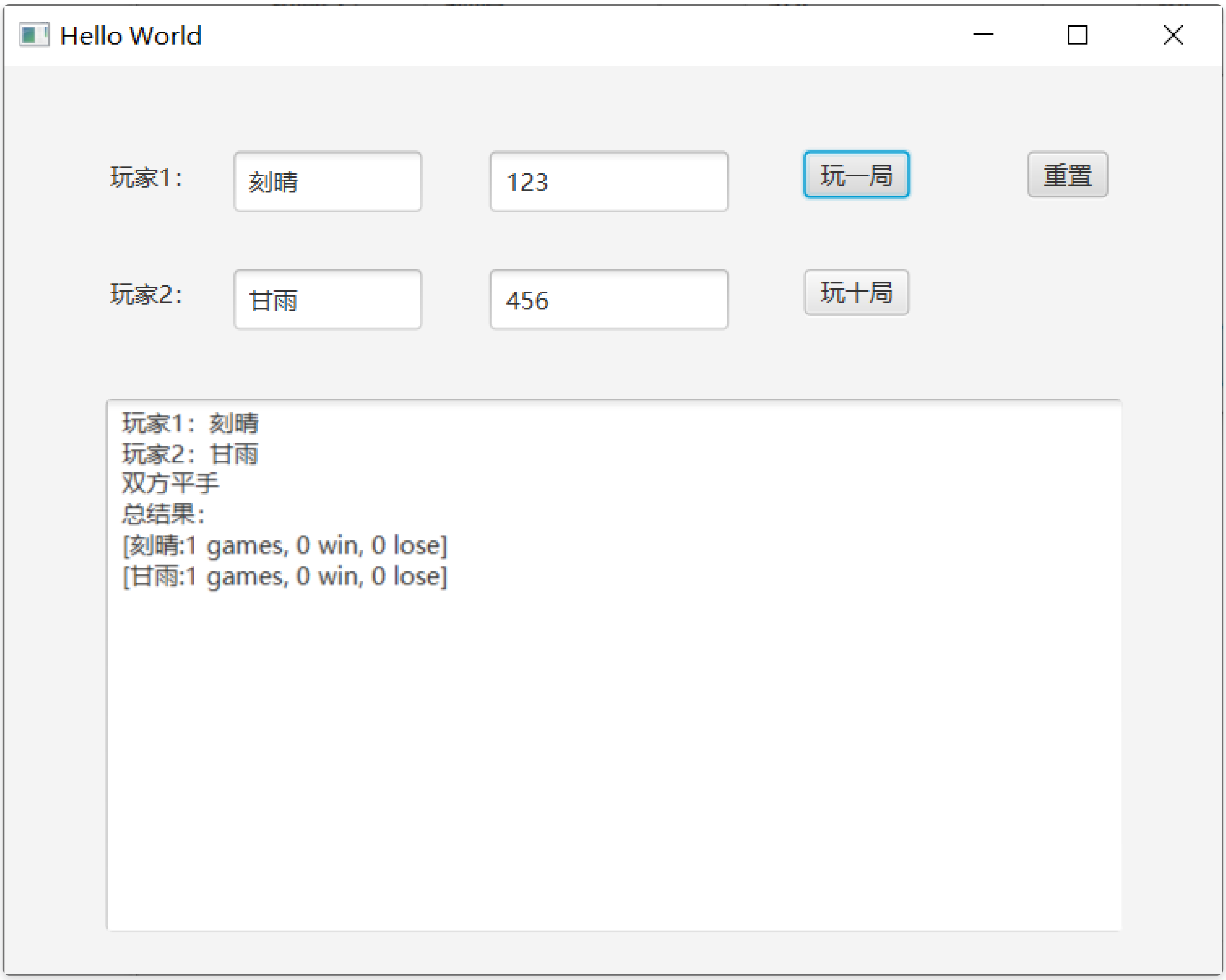
Hello World

玩家1：刻晴 abc

输入格式错误

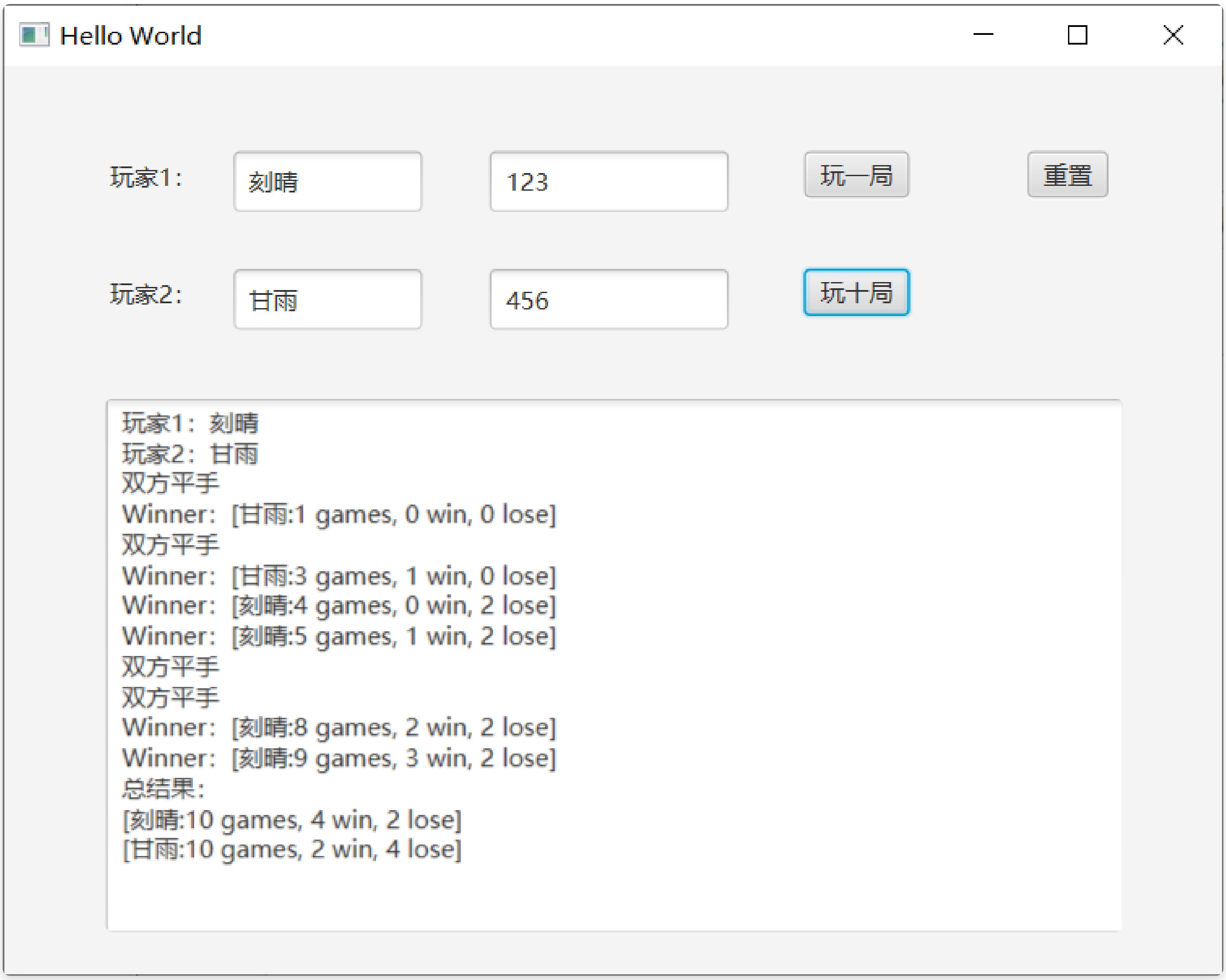
非法的Seed值，请重新输入！

图1-3-3 非法Seed值提示



```
玩家1：刻晴
玩家2：甘雨
双方平手
总结果：
[刻晴:1 games, 0 win, 0 lose]
[甘雨:1 games, 0 win, 0 lose]
```

图1-3-4 玩一局运行结果



玩家1：刻晴 123

玩家2：甘雨 456

```
玩家1：刻晴
玩家2：甘雨
双方平手
Winner：[甘雨:1 games, 0 win, 0 lose]
双方平手
Winner：[甘雨:3 games, 1 win, 0 lose]
Winner：[刻晴:4 games, 0 win, 2 lose]
Winner：[刻晴:5 games, 1 win, 2 lose]
双方平手
双方平手
Winner：[刻晴:8 games, 2 win, 2 lose]
Winner：[刻晴:9 games, 3 win, 2 lose]
总结果：
[刻晴:10 games, 4 win, 2 lose]
[甘雨:10 games, 2 win, 4 lose]
```

图1-3-5 玩十局运行结果

2. Composite GUI改写

2.1 目录结构

> └─sample
> App.java ——启动类
> Controller.java ——控制器类
> Directory.java ——表示目录的类
> Entry.java ——对File和Directory的抽象类
> File.java ——表示文件的类
> FileTreatmentException.java ——异常类
> Global.java ——全局类，用于结果输出的字符串拼接
> Main.java ——启动方式定义类
> sample.fxml ——界面文件

2.2 关键部分代码

sample.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="sample.Controller">
   <children>
      <Button fx:id="chooseFileButton" layoutX="23.0" layoutY="25.0" mnemonicParsing="false"
onAction="#chooseFileButtonPressed" text="选择文件夹" />
      <TextArea fx:id="resultShowTextArea" editable="false" layoutX="23.0" layoutY="109.0"
prefHeight="269.0" prefWidth="554.0" />
      <Label layoutX="23.0" layoutY="78.0" text="文件目录遍历结果: " />
   </children>
</Pane>
```

Global.java

```
package sample;

public class Global {
    public StringBuilder stringBuilder = new StringBuilder();
}
```

File.java（修改printList方法，修改size为long类型）

```
package sample;

public class File  extends Entry{
    private String name;
    private long size;
    public File(String name, long size) {
        this.name = name;
        this.size = size;
    }
    public String getName() {
        return name;
    }
    public int getSize() {
        return size;
    }
    protected void printList(String prefix) {
        Global global = Controller.global;
        global.stringBuilder.append(prefix + "/" + this).append("\n");
    }
}
```

Directory.java（修改printList方法，修改size为long类型）

```
package sample;

import java.util.Iterator;
import java.util.Vector;

public class Directory extends Entry {
    private String name;
    private Vector directory = new Vector();
    public Directory(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public long getSize() {
        long size = 0;
        Iterator it = directory.iterator();
        while (it.hasNext()) {
            Entry entry = (Entry)it.next();
            size += entry.getSize();
        }
        return size;
    }
    public Entry add(Entry entry) {
        directory.add(entry);
        return this;
    }
    protected void printList(String prefix) {
        Global global = Controller.global;
        global.stringBuilder.append(prefix + "/" + this).append("\n");
        Iterator it = directory.iterator();
        while (it.hasNext()) {
            Entry entry = (Entry)it.next();
            entry.printList(prefix + "/" + name);
        }
    }
}
```

Entry.java（修改size为long类型，size输出格式修改为MB）

```
package sample;

public abstract class Entry {
    public abstract String getName();
    public abstract long getSize();
    public Entry add(Entry entry) throws FileTreatmentException {
        throw new FileTreatmentException();
    }
    public void printList() {
        printList("");
    }
    protected abstract void printList(String prefix);
    // 输出总览
    public String toString() {
        return getName() + " (" + String.format("%.4f", getSize()/1024.0/1024.0) + ") MB";
    }
}
```

Controller.java

```java
package sample;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.Button;
import javafx.scene.control.TextArea;
import javafx.stage.DirectoryChooser;
import javafx.stage.Stage;

import java.io.File;

public class Controller {
    @FXML private Button chooseFileButton;
    @FXML private TextArea resultShowTextArea;
    public static Global global = new Global();

    // 递归遍历文件夹
    public static void showFile(File file, Directory dir) {
        String [] list = file.list();
        for (int i = 0; i < list.length; i++) {
            File tmp = new File(file.getAbsolutePath() + "\\" + list[i]);
            if (tmp.isDirectory()) {
                Directory tmpDir = new Directory(tmp.getName());
                dir.add(tmpDir);
                showFile(tmp, tmpDir);
            } else {
                dir.add(new sample.File(tmp.getName(), (int) tmp.length()));
            }
        }
    }

    public void chooseFileButtonPressed(ActionEvent e) {
        global.stringBuilder.delete(0, global.stringBuilder.length());
        DirectoryChooser directoryChooser = new DirectoryChooser();
        File file = directoryChooser.showDialog(new Stage());
        Directory beginDir = new Directory(file.getName());
        showFile(file, beginDir);
        beginDir.printList();
        resultShowTextArea.setText(global.stringBuilder.toString());
    }
}
```

Main.java

```java
package sample;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;
```

```
public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));
        Scene scene = new Scene(root);
        primaryStage.setTitle("Composite改写");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    // 此main函数不可用，请使用App.java启动
    public static void main(String[] args) {
        launch(args);
    }
}
```

App.java

```
package sample;

import javafx.application.Application;

public class App {
    public static void main(String[] args) {
        Application.launch(Main.class);
    }
}
```

2.3 运行演示

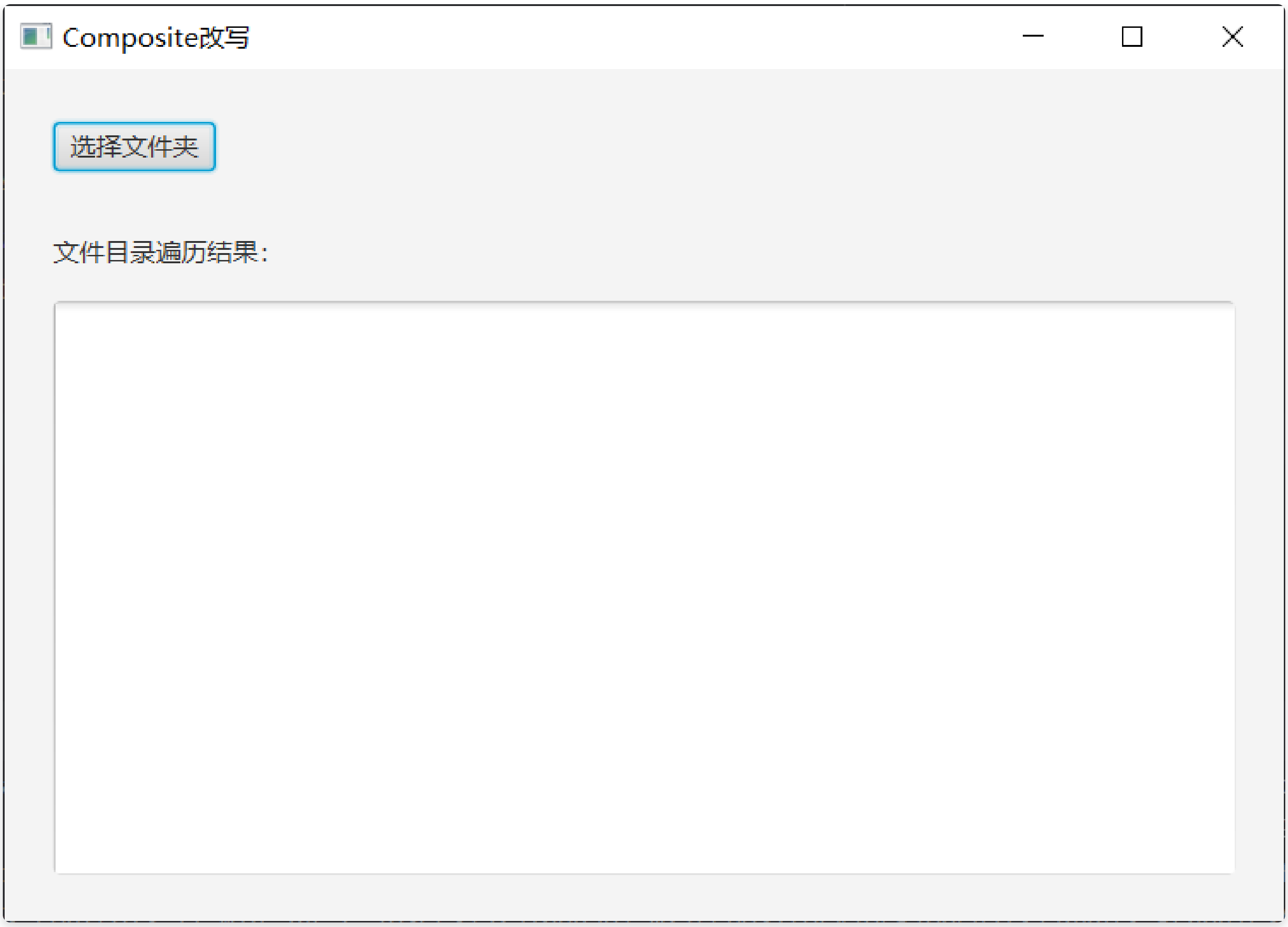
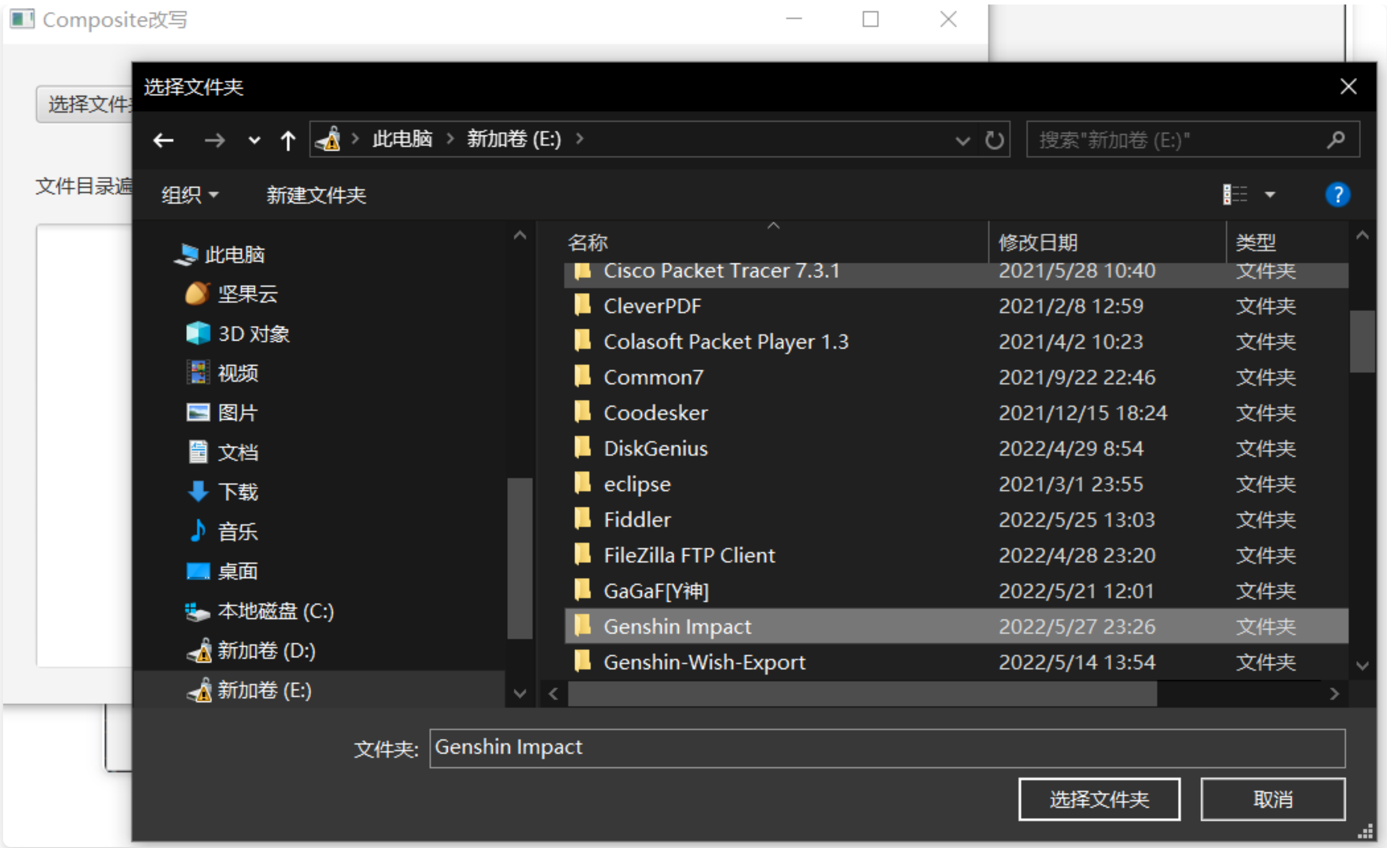


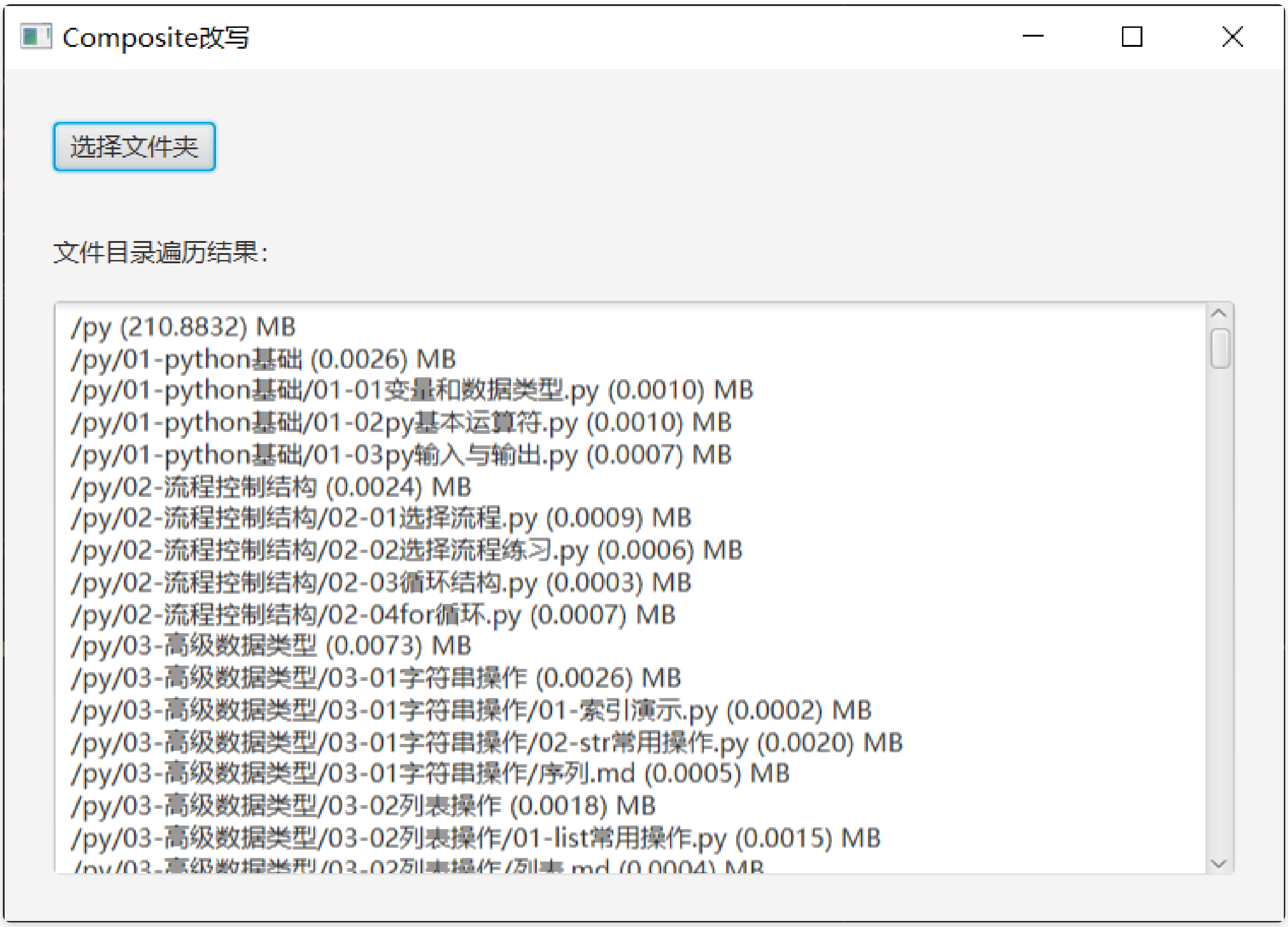
图2-3-1 主界面



名称	修改日期	类型
Cisco Packet Tracer 7.3.1	2021/5/28 10:40	文件夹
CleverPDF	2021/2/8 12:59	文件夹
Colasoft Packet Player 1.3	2021/4/2 10:23	文件夹
Common7	2021/9/22 22:46	文件夹
Coodesker	2021/12/15 18:24	文件夹
DiskGenius	2022/4/29 8:54	文件夹
eclipse	2021/3/1 23:55	文件夹
Fiddler	2022/5/25 13:03	文件夹
FileZilla FTP Client	2022/4/28 23:20	文件夹
GaGaF[Y神]	2022/5/21 12:01	文件夹
Genshin Impact	2022/5/27 23:26	文件夹
Genshin-Wish-Export	2022/5/14 13:54	文件夹

文件夹: Genshin Impact

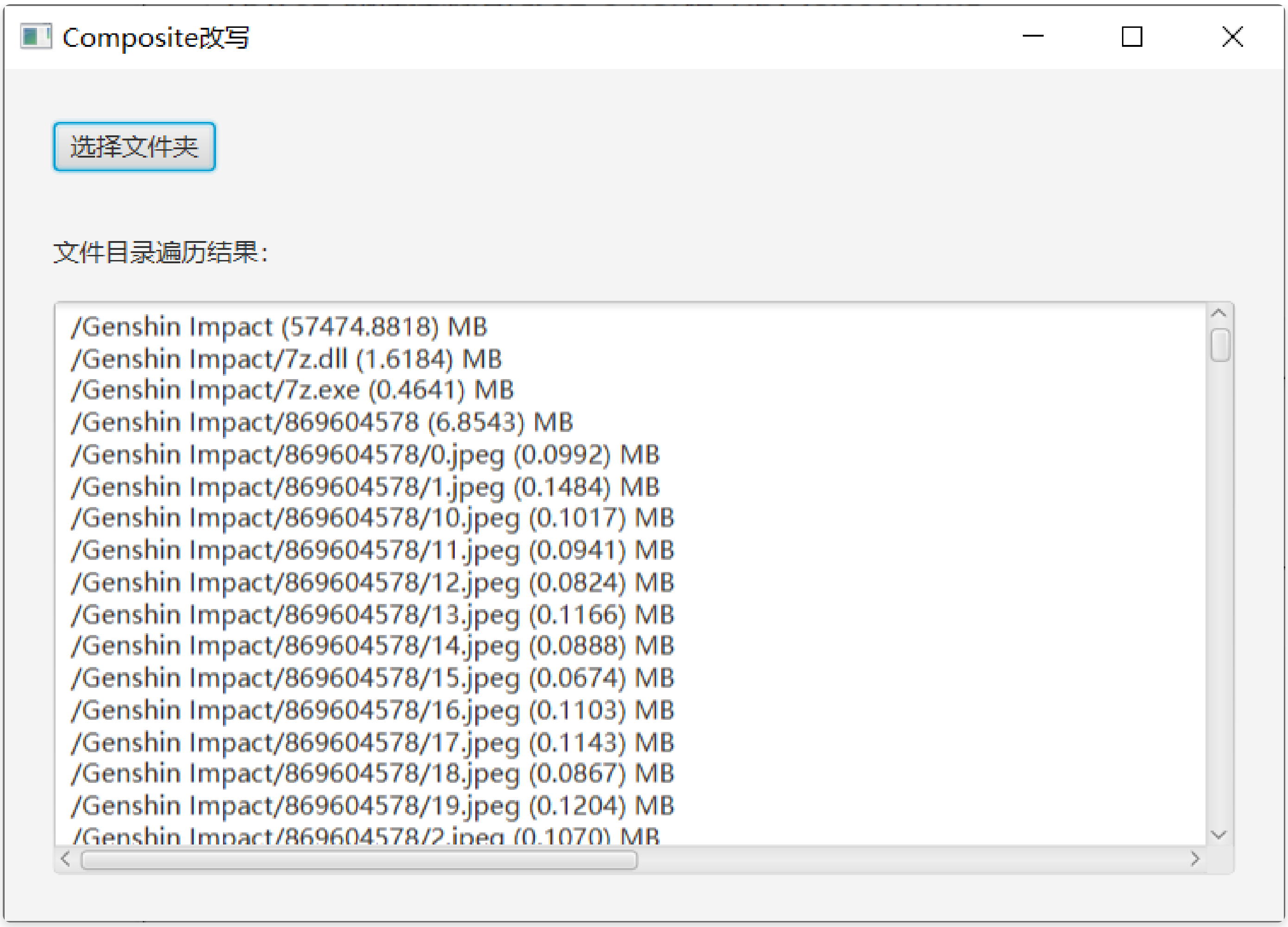
图2-3-2 点击选择文件夹按钮的选择器



文件目录遍历结果：

```
/py (210.8832) MB
/py/01-python基础 (0.0026) MB
/py/01-python基础/01-01变量和数据类型.py (0.0010) MB
/py/01-python基础/01-02py基本运算符.py (0.0010) MB
/py/01-python基础/01-03py输入与输出.py (0.0007) MB
/py/02-流程控制结构 (0.0024) MB
/py/02-流程控制结构/02-01选择流程.py (0.0009) MB
/py/02-流程控制结构/02-02选择流程练习.py (0.0006) MB
/py/02-流程控制结构/02-03循环结构.py (0.0003) MB
/py/02-流程控制结构/02-04for循环.py (0.0007) MB
/py/03-高级数据类型 (0.0073) MB
/py/03-高级数据类型/03-01字符串操作 (0.0026) MB
/py/03-高级数据类型/03-01字符串操作/01-索引演示.py (0.0002) MB
/py/03-高级数据类型/03-01字符串操作/02-str常用操作.py (0.0020) MB
/py/03-高级数据类型/03-01字符串操作/序列.md (0.0005) MB
/py/03-高级数据类型/03-02列表操作 (0.0018) MB
/py/03-高级数据类型/03-02列表操作/01-list常用操作.py (0.0015) MB
/py/03-高级数据类型/03-02列表操作/列表.md (0.0004) MB
```

图2-3-3 选择之后的遍历结果



```
/Genshin Impact (57474.8818) MB
/Genshin Impact/7z.dll (1.6184) MB
/Genshin Impact/7z.exe (0.4641) MB
/Genshin Impact/869604578 (6.8543) MB
/Genshin Impact/869604578/0.jpeg (0.0992) MB
/Genshin Impact/869604578/1.jpeg (0.1484) MB
/Genshin Impact/869604578/10.jpeg (0.1017) MB
/Genshin Impact/869604578/11.jpeg (0.0941) MB
/Genshin Impact/869604578/12.jpeg (0.0824) MB
/Genshin Impact/869604578/13.jpeg (0.1166) MB
/Genshin Impact/869604578/14.jpeg (0.0888) MB
/Genshin Impact/869604578/15.jpeg (0.0674) MB
/Genshin Impact/869604578/16.jpeg (0.1103) MB
/Genshin Impact/869604578/17.jpeg (0.1143) MB
/Genshin Impact/869604578/18.jpeg (0.0867) MB
/Genshin Impact/869604578/19.jpeg (0.1204) MB
/Genshin Impact/869604578/2.jpeg (0.1070) MB
```

图2-3-4 另一个文件夹遍历