

《移动应用软件开发》实验报告

年级、专业、班级		姓名		学号	
实验题目	数字华容道游戏 APP 开发				
实验时间	2023. 11. 5	实验地点	DS1401		
学年学期	2023-2024(1)	实验性质	<input type="checkbox"/> 验证性 <input checked="" type="checkbox"/> 设计性 <input type="checkbox"/> 综合性		
一、实验目的 1. 本次实验的目的是掌握华为 DevEco Studio 开发工具的基本操作，了解所开发 APP 项目的基本结构。掌握 Ability 框架和 UI 框架的使用，掌握常用组件和布局的使用，尤其图像组件的使用，位图的处理等。 2. 确定开发平台为 HarmonyOS，开发工具华为 DevEco Studio，建议选择最新版。可以使用 Java 语言，也可以使用 eTS 语言开发。 3. 设计完成一个数字华容道游戏 APP。					
二、实验项目内容 从多张图片中，选择一张，分割为 4 块以上的小图片，打乱后分布，可以拼成原图，数字正确排列。 评分点：（1）可以从多张图片中选择一张图片，然后分割成小图片，每个图片上面 标记上数字 ；（2）通过开始按钮开始游戏，把分隔后的图片打乱分布，显示在界面上，并开始游戏；（3）通过点击图片进行移动。 （4）游戏过程中，自动判断是否完成游戏。（5）设置结束按钮提前结束游戏，显示正确结果；（6）APP 界面自行设计，不少于两个。（7）可以自行增加游戏计时、计分、关卡、动画、历史记录、从相册选择图片、游戏指南等功能，（8）复杂度、美观度、自定义功能、多设备协同等都是加分项。 提交：（1）本实验报告，（2）源代码压缩文件 zip，注意源代码加注释。（3）软件演示的 MP4 视频，视频大小不超过 20M。能够在搜狗浏览器正常播放。注意文件名称的规范性。文件名：学号姓名 2.doc，学号姓名 2.zip，学号姓名 2.mp4。三个文件分别提交。					
三、实验过程或算法（写明设计思想、程序的结构、功能关系图、类的说明和类之间的关系图、程序主要执行流程图，最后是核心源代码，截图等） 1 设计思想： 1.1 算法设计： 本次数字华容道实验在算法实现方面难度并不是很高，主要涉及到					

下面几个部分：

1.1.1 图片处理方面：

本次实验在图像处理方面主要需要实现图片的等分切割和在图片上进行一些简单绘制（为子图标上编号）。在这里我们主要使用了这样一个方法来完成：

```
/**
 * 等距切割图片
 * @param img 原图
 * @param rows 要切割的行数
 * @param cols 要切割的列数
 * @param withIndex 切割后图片是否标记编号
 * @returns 切割后的图片数组
 */
public static async cutImage(img: Image.PixelMap, rows: number, cols: number, withIndex: boolean = true): Promise<CroppedImage[]>
```

通过传入的 `PixelMap` 对象可以获取到图像的基本长宽信息，再结合给定的要切割的行数和列数可以确定切割的份数和每个子图的长宽信息，再通过简单的循环嵌套，计算得到该次迭代中子图的起始坐标便可切得子图。在这里我并没有真的对原图进行切割，而是使用了 `ArkTS` 提供的 `OffscreenCanvasRenderingContext2D` 对象的 `drawImage` 方法通过离屏绘制原图的某一部分来实现切割图片的效果，离屏绘制可以将需要绘制的内容先绘制在缓存区，然后将其转换成图片，一次性绘制到 `canvas` 上，加快了绘制速度。`drawImage` 方法签名如下

```
/**
 * Draw an image on a canvas
 * @param image Picture objects drawn to the canvas.
 * @param sx x coordinate of the upper left corner of the rectangle (cropping) selection box of the image.
 * @param sy y coordinate of the upper left corner of the rectangle (cropping) selection box of the image.
 * @param sw Width of the rectangle (cropping) selection box of the image.
 * @param sh Height of the rectangle (cropping) selection box of the image.
 * @param dx x-axis coordinate of the upper left corner of the image on the target canvas.
 * @param dy y-axis coordinate of the upper left corner of the image on the target canvas.
 * @param dw Specifies the drawing width of the image on the target canvas. The width of the drawn image will be scaled.
 * @param dh Specifies the drawing height of the image on the target canvas. The height of the drawn image will be scaled.
 * @form
 * @since 9
 */
drawImage(image: ImageBitmap | PixelMap, sx: number, sy: number, sw: number, sh: number, dx: number, dy: number, dw: number, dh: number): void;
```

于是切割图片的核心代码如下所示：

```
// 计算切割的起始坐标
const startX: number = j * eachWidth;
const startY: number = i * eachHeight;

// 利用 OffscreenCanvasRenderingContext2D 进行图像切割
this.offC2d.drawImage(img, startX, startY, eachWidth, eachHeight, 0, 0, eachWidth, eachHeight)
```

并且由于我们是使用画布绘制的方式来进行图片处理，所以我们也能很简单的实现在图片上绘制编号：

```
// 如果 withIndex 为真，将图像编号添加到子图像中
if (withIndex) {
  this.offC2d.fillText(index + 1 + '', eachWidth / 2, eachHeight / 2)
}
```

1.1.2 打乱方法设计:

由于开始游戏后需要对切割的子图进行随机打乱，所以我们需要设计一个随机打乱算法，一开始我使用了 Knuth-Durstenfeld shuffle 算法，如下：

```
public static shuffle(array: Array<CroppedImage>): void {
    let n = array.length, random;
    while (0 != n) {
        random = (Math.random() * n--) >>> 0; // 无符号右移位运算符向下取整
        [array[n], array[random]] = [array[random], array[n]] // ES6的结构赋值实现变量互换
    }
}
```

这个算法确实能够实现随机打乱，但是测试之后发现存在一个问题，也就是打乱之后的华容道不一定有解，一个解决办法是打乱之后检查是否有解，无解则继续打乱直至有解，但这个办法的缺点在于需要额外实现一个检查算法，并且随机打乱得到的结果很有可能无解，会导致多次执行打乱和检查算法，程序效率低下，于是我们考虑换一种随机打乱方法。

为了保证打乱后的结果是有解的，我们可以使用一种随机+模拟的方法来进行打乱。华容道的游戏本质就是玩家点击空白格上下左右四个方向的其他单元格使其与空白格换位来完成游戏，于是我们可以模拟多次玩家随机点击空白格周围单元格的行为来实现打乱，具体实现如下：

```
/**
 * 打乱函数
 * @param array 分割后的子图数组
 * @param cols 分割的列数
 */
public static shuffle(array: Array<CroppedImage>, cols: number): void {
    // 获取空白图块的索引
    let blankIndex = this.getBlankIndex(array);
    // 定义移动方向，包括左、右、上、下
    const n = array.length, direction = [-1, 1, -cols, cols];
    // 循环 n*n 次，打乱图块的位置
    for (let i = 0; i < n * n; i++) {
        let nextBlankIndex;
        // 随机选择一个方向，直到找到有效的下一个空白图块的索引
        do {
            nextBlankIndex = blankIndex + direction[Math.floor(Math.random() * 4)];
            // 随机得到的坐标不合法或者在二维化后的子图数组中并不相邻
        } while (!this.areIndexesAdjacent(blankIndex, nextBlankIndex, n / cols, cols));
        // 交换当前图块和下一个空白图块的位置
        [array[blankIndex], array[nextBlankIndex]] = [array[nextBlankIndex], array[blankIndex]];
        blankIndex = nextBlankIndex;
    }
}
```

其中 areIndexesAdjacent 用于判断给定两个一维数组的索引在二维化后的数组中是否相邻，因为当空白格处于二维数组边缘时并不同时拥有上下左右四个方向，所以需要对该判断：

```
/**
 * 检查两个索引在二维化后的子图数组是否相邻
 * @param index1 索引1
 * @param index2 索引2
 * @param rows 分割的行数
 * @param cols 分割的列数
 * @returns 是否相邻
 */
public static areIndexesAdjacent(index1: number, index2: number, rows: number, cols: number): boolean {
    // 检查给定索引是否合法
    if (index1 < 0 || index1 >= rows * cols || index2 < 0 || index2 >= rows * cols) {
        // 不合法, 返回false
        return false
    }
    // 计算 index1 和 index2 在矩阵中的行和列
    const row1 = Math.floor(index1 / cols);
    const col1 = index1 % cols;
    const row2 = Math.floor(index2 / cols);
    const col2 = index2 % cols;

    // 判断两个索引是否在同一行或同一列, 并且相邻
    if ((Math.abs(row1 - row2) === 1 && col1 === col2) || (Math.abs(col1 - col2) === 1 && row1 === row2)) {
        return true;
    }
    return false;
}
```

1.2 用户界面设计:

整个 APP 主要包含两个页面, 主页面和游戏页面:

1.2.1 主页面:

主页面主要包含三部分:

1.2.1.1 难度选择区:

在难度选择区, 用户可以根据下拉框选择华容道的行数和列数, 也可通过 checkBox 选择是否开启图像编号显示。

1.2.1.2 图片选择区:

图片选择区主要有三部分:

1.2.1.2.1 当前选择图片展示

1.2.1.2.2 可选图片列表:

用户可滑动可选图片列表选择要选择的图片

1.2.1.2.3 从相册选择图片:

用户可从相册选择图片加入到可选图片列表

1.2.1.3 开始游戏按钮:

用户选择好难度和图片后可点击开始游戏按钮跳转到游戏页面, 路由跳转时会携带上用户选择的游戏设置。

1.2.2 游戏页面:

游戏页面主要包含五部分:

1.2.2.1 功能按钮区:

1.2.2.1.1 返回按钮: 用户可通过点击返回按钮返回主页面

1.2.2.1.2 重置按钮: 用户可通过点击重置按钮重新打乱华容道并重置计时器和计步器

1.2.2.2 信息统计区:

1.2.2.2.1 计时器: 用户开始游戏后便会开始计时, 点击重置按钮后会重置计时

1.2.2.2.2 计步器: 用户点击华容道完成一次交换后计步器加一, 点击重置按钮后会重置计步

1.2.2.3 游戏区:

游戏区以 Grid 布局展示以用户选择的设置分割得到的华

容道游戏主体,用户通过点击空白格周围单元格以交换子图,最终完成游戏

1.2.2.4 原图查看按钮:

考虑到游戏难度问题,特别是在用户选择不显示子图编号时,我们设计了查看原图功能,在用户感到困难时可以点击查看原图按钮查看原图。

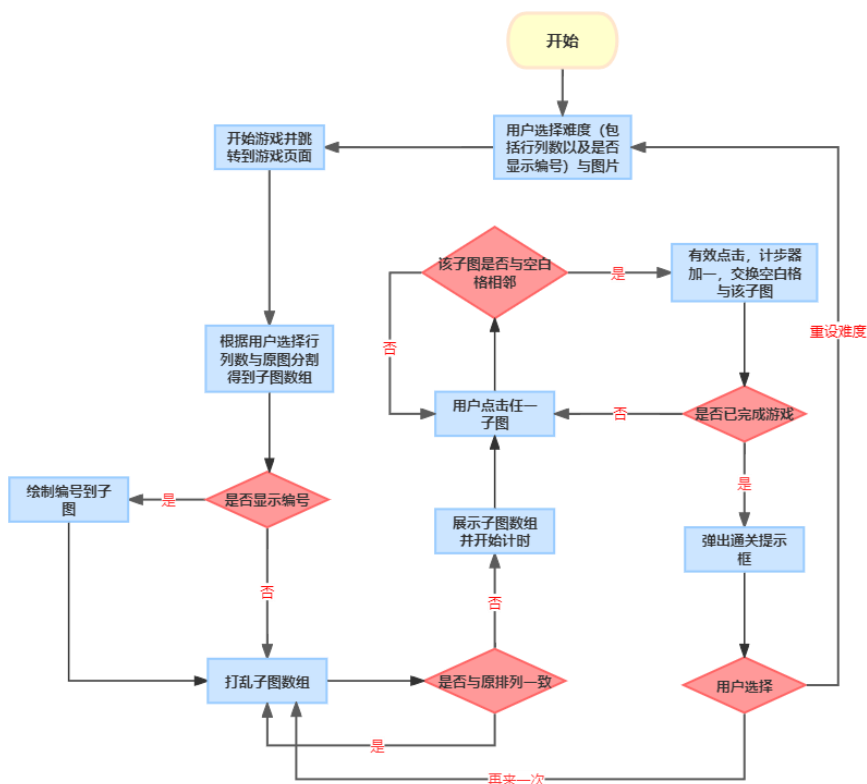
1.2.2.5 游戏通过对话框:

该部分不会直接显示,会在用户完成华容道后自动弹出,主要会展示本次挑战用时、步数,并且提供给用户再来一次和重设难度的功能。

2 程序结构



3 程序主要执行流程图



4 核心源代码

出去上面的算法设计部分的代码外，程序还有以下核心源代码：

4.1 检查是否已通关

只需简单将子图原本编号与当前排列的编号进行一一检查即可：

```
// 检查是否已成功还原
checkForSuccess(): boolean {
  for (let i = 0; i < this.croppedImages.length; i++) {
    if (this.croppedImages[i].index !== i) {
      return false
    }
  }
  return true
}
```

4.2 处理用户点击任一子图：

用户在点击任一子图后会触发 tryMoveImage 方法：

```
// 尝试移动子图
tryMoveImage(index: number): boolean {
  // 取得空白格坐标
  const blankIndex = GameUtils.getBlankIndex(this.croppedImages)
  // 点击图片与空白格不相等，无法移动
  if (!GameUtils.areIndexesAdjacent(index, blankIndex, this.rows, this.cols)) {
    return false
  }
  // 成功移动，步数加一
  this.stepCounter++
  // 点击图片与空白格换位
  [this.croppedImages[index], this.croppedImages[blankIndex]] = [this.croppedImages[blankIndex], this.croppedImages[index]]
  if (this.checkForSuccess()) {
    // 已经成功还原，弹出游戏通关对话框
    this.onSuccess()
  }
  return true
}
```

四、实验结果及分析和（或）源程序调试过程（界面截图和文字）、实验总结与体会

1 实验结果与分析： 1.1 主页面效果：



1.2 游戏页面效果：



1.3 游戏通关提示框:



2 总结与体会:

这次移动应用开发实验我选择了基于 HarmonyOS 的 Stage 模型和 ArkTS 声明式开发范式,也是目前最新的 API 9 主要支持的技术选择,旨在掌握华为 DevEco Studio 开发工具的基本操作,了解 APP 项目的基本结构,掌握 Ability 框架和 ArkTS 的 UI 框架的使用,以及常用组件、布局和图像处理等技能。以下是我对这次实验的总结与体会:

2.1 工具与平台选择:

选择了最新版本的华为 DevEco Studio,这确保了我们能够利用最新的功能和工具来进行开发。

使用 HarmonyOS 平台为我们提供了多设备的支持,使我们的应用在不同设备上都能够运行,并享受 HarmonyOS 的生态系统优势。

2.2 学习和理解基本结构:

通过实验，我深入了解了 APP 项目的基本结构，包括 Ability 的概念、布局文件、资源文件等。这为我以后的 HarmonyOS 开发奠定了坚实的基础。

2.3 掌握 Ability 框架和 UI 框架：

在实验中，我学会了如何创建 Ability，配置 Ability 的生命周期，以及如何在 Ability 中构建 UI 界面。

通过使用 ArkTS 的 UI 框架，我能够轻松地创建应用的用户界面，包括各种组件和布局，从而提高了用户体验。

2.4 图像处理：

实验中，我学会了如何处理位图和图像组件，这对于创建各种视觉元素和图形效果非常重要。这包括图像的加载、显示和处理。

2.5 ArkTS 声明式开发范式：

ArkTS 的声明式开发范式让我更加高效地开发应用，减少了繁琐的手动编码，提高了代码的可读性和可维护性。

总的来说，这次实验让我对 HarmonyOS 应用开发有了更深入的了解，并掌握了许多关键的技能和工具。我对 HarmonyOS 的灵活性和多设备支持感到印象深刻，期待将来继续深入研究和开发基于 HarmonyOS 的应用。此外，ArkTS 的声明式开发范式让开发变得更加高效和愉快。

实验报告填写说明：

- 1、第一、二部分由老师提供；
- 2、第三部分填写源程序或者算法，清单文件，资源文件等。源程序要符合程序编写风格（缩进、注释等）；
- 3、第四部分主要填写程序调试运行过程、结果（截图）、解决问题的方法、总结和体会等；
- 4、报告规范：包含报告页眉、报告的排版、内容是否填写，命名是否规范等。