

# 《拼图游戏》设计及实现

## —《面向对象程序设计基础》作业总结

李堃秀 2022012110 likx22@mails.tsinghua.edu.cn

**【摘要】**使用 C++ 语言实现拼图游戏，基本功能为图片的拆分及拼图块的平移、放置和拼接，玩家使用鼠标左键点击并拖拽拼图块至拼图区，当位置足够准确时会自动放置并拼接，否则图块将回到池中。为提高游戏的可玩性和娱乐性，设计了图片选择、难度选择、计时、背景音乐、最快记录等功能，为增加游戏难度，设计了拼图块的旋转功能，同时，提出了下一步优化提升的方向。

**【关键词】**面向对象程序设计、C++、拼图游戏

### 一、完成情况总览

要求	完成情况
实现拼图游戏	已实现
能够使用鼠标拖动池中的拼图块到某一位置，如果该位置足够准确则拼接正确。	已实现
接接正确时该块会留在正确位置，否则回到池中。	已实现
接图块只能平移无法旋转。	已实现
至少要做 2 组拼图模版，每个模版至少含有 10 个拼图块	已实现，并增加难度选择功能，可拆分不同数量的拼图块（难度越高，块数越多）
进阶版（选做）：拼图块既能够进行平移也能够进行旋转，旋转可以使用滚轮或者键盘。	已实现，并增加了背景音乐、计时、最快记录等娱乐性功能

### 二、《拼图游戏》设计及实现

#### 1. 拼图游戏总体介绍

《拼图》是一款基于 Windows 平台开发的益智类游戏，游戏复现了传统实物拼图的功能，并进行了适当的提升。

在游戏中，玩家可以从所给的三幅图片中选择喜爱的图片开始游戏，游戏共分为三个难度，不同难度对应着不同的拼图块数。游戏中，玩家通过按下鼠标左键并拖拽图案实现拼图功能，同时部分拼图块在初始时是倒置的，玩家需要通过右键单击将其旋转到正确的方向。游戏中提供了背景音乐，玩家也可以通过按钮关闭。

相较于实物拼图，本游戏加入了计时功能和排行功能，在玩家打破最快纪录时会给出提示文字，为游戏增加了乐趣与挑战性。

## 2. 游戏界面及交互设计

### 2.1. 界面设计



上图为本游戏的初始界面，背景图片主要采用神秘梦幻的主题风格，交互按钮使用半透明位图，使得同样的按钮能够在三种不同色系的背景图中都不突兀。



上图为本游戏的游戏中界面，左上方为初始图片的完整图案，以供玩家在游戏中参考，图片被按难度等级拆分为拼图块后，随机放置于右上角的拼图块池中，玩家可以使用鼠标左键点击选择图块，拖拽至左下角的拼图区中，系统判定位置足够准确时会自动放置并接接，直至完成全部拼图。

### 2.2. 交互设计

初始界面：

鼠标左键单击：  
切换成编号减 1 的图片



鼠标左键单击：  
切换成编号加 1 的图片

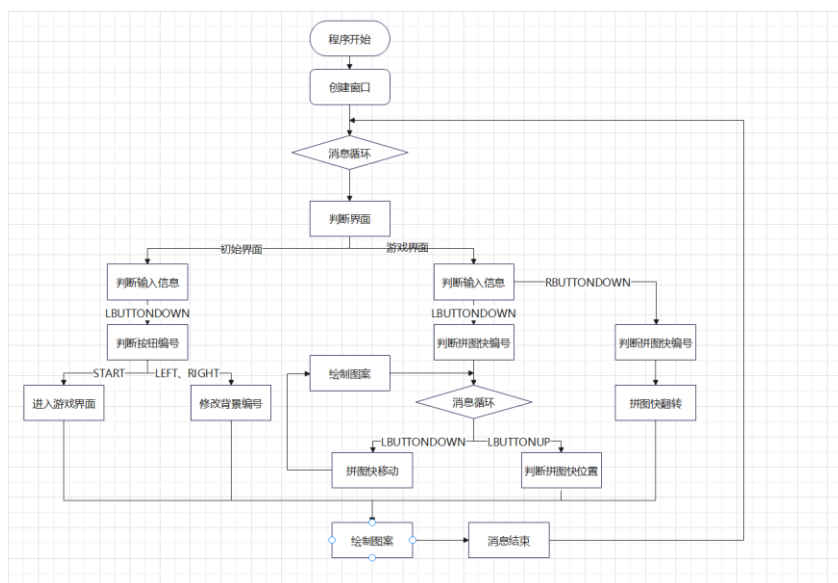
鼠标左键单击：  
选择当前图片并进入游戏



在拼图块池中  
鼠标左键按下并移动鼠标：拖拽并移动图片  
鼠标右键单击：旋转图片

### 3. 代码实现

#### 3.1.流程图



#### 3.2.核心模块代码实现

##### 1) 图案绘制

本游戏使用 EasyX 图形库进行图形编辑,在循环中运用双缓冲技术及 Sleep 函数解决了屏幕闪烁问题,保证了游戏的流畅运行。

##### 2) 背景修改

游戏程序中定义了窗口类、按钮类及游戏管理器类,游戏管理器获取鼠标输入信息后遍历类成员(按钮组),并通过按钮类中的成员(按钮坐标和类型)判断是否需要修改背景编号,游戏管理器获取背景编号后调用类成员函数并绘制相应背景图。

##### 3) 拼图块翻转

初始化游戏管理器时,通过 EasyX 库中的翻转函数向游戏管理器类的图片组中导入了正反两张图片,绘制时游戏管理器现获取拼图块的编号和旋转属性:若不需要选择,则打印正向图片中该编号的区域;反之,则打印反向图片中(难度^2-拼图块编号)的区域。

游戏管理器进入游戏运行进程后获取鼠标信息，若为右键按下，则对比判断鼠标位置和拼图块位置，使二者匹配的拼图块发生旋转。

注：由于本游戏中的拼图块为长宽不等的矩形，设计其他角度的旋转（如  $90^\circ$ ）对游戏操作并无实际意义，故只设置了旋转  $180^\circ$  功能。

#### 4) 拼图块移动

游戏管理器进入游戏运行进程后获取鼠标信息，若为左键按下，且遍历拼图块组后有与该位置匹配的拼图块，则进入“拼图块移动”循环。在该循环中，若左键按下，则更新对应拼图块的位置，若左键弹起，则判断当前鼠标位置，在正确坐标范围内时拼图块自动移动到正确位置，否则拼图块移动回初始位置。

#### 5) 拼图完成判断

游戏运行开始时记录当前现实时间并进入循环。

每次循环结束前，游戏管理器遍历按钮组，判断其当前位置和正确位置是否相同，若全部为 `true`，则游戏结束并记录当前时间。游戏管理器的时间组中加入上述两次时间的差值，并判断该数值是否为组内最小，若为 `true`，则打印“New Record!”字样。随后背景编号被重置为 0，返回初始场景，游戏难度提升。

### 4. 游戏运行结果

#### 4.1.运行环境

Window 10 , 64 位操作系统，visual studio 2022

#### 4.2.运行结果截图

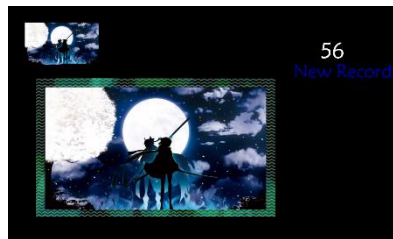
初始场景



游戏进行界面



游戏结束界面



### 三、总结反思与下一步提升

#### 1、总结与反思

拼图游戏的实现难度并不大，我很快就在三月中旬就完成了初代版本，实现了基本拼图功能以及旋转、计时、排行榜、背景音乐等增值功能，调试运行通过。但在后续学习以及完善代码的过程中，我认识到我对于“面向对象”思想的了解还十分浅薄，第一个版本的代码虽然能够基本完成选题要求的功能，但是所定义的类基本仅限于将多个函数打包，不具有面向对象程序设计的可复用性等特点。在认真学习《C++程序设计从入门到精通》的“设计模式”一章后，我深受启发，将原有的代码全部推翻，按照我对“面向对象”思想的学习和理解重写了第二个版本。例如，在按钮类中，我尝试运用“策略模式”，在之后丰富游戏功能的时候可以大大减少后续的工作量，并且避免了修改已成型模块所带来的众多麻烦。

同时，我在使用 easyX 图形库进行开发的过程中，发现在拖拽拼图块时会出现屏幕闪烁的问题，游戏体验很差，经过多方查阅资料，最终通过在循环中运用双缓冲技术及 Sleep 函数，实现了游戏的流畅操作

以上过程虽然耗时颇多，但提升了我的学习能力和编程思想，让我受益匪浅。

#### 2、游戏下一步优化方向

设计更多不同形状的拼图块（模仿真实拼图，设计不规则、嵌入式的图块形状），并实现多种角度的旋转功能，增加游戏的趣味性。

#### 参考资料：

- 1、雍俊海：《C++程序设计从入门到精通》，清华大学出版社
- 2、EasyX 在线文档，<https://docs.easyx.cn/zh-cn/intro>