**武汉大学**

**计算机学院实训**

**STACK5（自选项目）立项报告**

**项目成员：ZIDANE ABID HASAN**

**撰 写 人：ZIDANE ABID HASAN** 2025 年 06 月 23 日

**评 审 人：**  年 月 日

武汉凡诺软件技术有限公司

2025年06月23日

**修订记录**

| 日期 | 修订版本 | 修改章节 | 修改描述 | 修订人 |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目 录**

自选[项目立项报告 1](#__RefHeading___Toc31083)

[1.项目提出 4](#__RefHeading___Toc11605)

[1.1 项目简介 4](#__RefHeading___Toc19397)

[1.2 类似产品或系统的分析 4](#__RefHeading___Toc9763)

[1.3 产品或系统的特色及预计达到的目标 4](#__RefHeading___Toc2414)

[2.团队组成和计划 5](#__RefHeading___Toc3595)

[2.1项目计划 5](#__RefHeading___Toc9126)

[2.2 项目成员及背景和各项职责 5](#__RefHeading___Toc11180)

[3. 风险评估和规避 6](#__RefHeading___Toc24455)

[3.1 过程中存在的技术风险 6](#__RefHeading___Toc11490)

[3.2 过程中存在的管理风险 6](#__RefHeading___Toc25592)

[3.3 其它可能的风险 6](#__RefHeading___Toc4979)

**1.项目提出**

**1.1 项目简介**

Stack 5 是一款基于 Unity 开发的创新型合并消除类益智游戏，灵感来源于经典游戏 2048，但在核心规则和玩法机制上进行了大胆革新。玩家需要通过上下左右移动方块，将数字合并为目标数字，以获取更高分数。游戏支持特殊“质数”方块逻辑、不同的合并规则（如 2+3=5）、动画反馈以及自动生成提示机制，增强了策略性与趣味性。

**1.2 类似产品或系统的分析**

同类的游戏里有2048和Threes这两款游戏挺经典的也有代表性。下面的表提出它们的一些特点：

| 功能 | 2048 | Threes!（三消） |
| --- | --- | --- |
| 方块数字 | 2 的幂（2, 4, 8, ..., 2048） | 1 和 2（基础），然后是 3、6、12、24...（3 的倍数） |
| 方块生成 | 随机生成 2（90%）或 4（10%） | 战略性地生成 1、2 或 3 的倍数 |
| 合并规则 | 仅相同数字方块可合并（如 2 + 2） | 1 + 2 = 3，且只有相同的 3 的倍数才能合并 |
| 下一个方块预览 | 不可用 | 显示下一个方块 |
| 特殊方块 | 无 | 无明确特殊方块 |
| 强制多样性 | 否 | 否 |
| 游戏目标 | 合成 2048 方块 | 获得最高分（没有具体的终点方块） |
| UI 与动画 | 简单的滑动操作网格 | 平滑动画，卡通风格角色方块 |
| 平台支持 | 网页、移动端、桌面端 | 移动优先（iOS/Android） |
| 美术风格 | 极简数字方块 | 可爱角色，带颜色和表情 |

**1.3 产品或系统的特色及预计达到的目标**

特色机制：跟上面描述的经典的同类游戏，STACK5还是具有一定的区别。

| 功能 | STACK5 |
| --- | --- |
| 方块数字 | 自定义序列，从 2 和 3 开始 → 5, 10, 20, ..., 1280 |
| 方块生成 | 随机生成 2 或 3，并带有防连发机制防止连续出现相同方块 |
| 合并规则 | 允许 2 + 3 = 5，然后仅相同数字才能合并（如 5 + 5 = 10） |
| 下一个方块预览 | UI 中清楚显示（基于文本） |
| 特殊方块 | 稀有出现的素数组成的方块如 2222 和 3333，会触发变换效果 |
| 强制多样性 | 若连续生成两个相同方块，则下一个方块强制不同 |
| 游戏目标 | 无限游戏 — 没有固定目标方块 |
| UI 与动画 | 自定义 UI，包含暂停菜单、主菜单、下一方块预览、游戏结束界面、自定义动画 |
| 平台支持 | 基于 Unity 的 PC 游戏，支持跨平台导出 |
| 美术风格 | 数字方块，支持自定义方块贴图/颜色 |

****目标**：**主要目标就是更加了解设计、开发游戏这方面、熟悉Unity引擎、提高阅读代码能力。根据计划要把那些功能按时加入到游戏里。

**2.团队组成和计划**

**2.1项目计划**

**1. 暂停 /继续游戏scene。**

### 2. 设置菜单

### 3. 从上次进度继续游戏

### 4. 撤销上一步操作

### 5. 代码自己化

6. **动态网格**

### 7.移动建议功能

### 8. 优化动画,音效

**9. 移动平台支持**

**2.2 项目成员及背景和各项职责**

团队只有一个人，我，背景为讨厌写代码，职责分配方面还是会有挺多自由。

**3. 风险评估和规避**

### **3.1 开发过程中存在的技术风险**

****方块合并逻辑错误**：**新的合并逻辑（例如 2 + 3 = 5）可能会打破传统2048的 玩家预期，容易引入逻辑错误。  
   ****规避方式**：**对合并逻辑进行单元测试，尤其是边界情况。

****方块生成公平性问题**：**Unity 自带的随机函数可能不够公平，导致连续生成相 同方块，引发玩家不满。  
   ****规避方式**：**实现反连号机制和平衡生成算法。

### **3.2 开发过程中存在的管理风险**

****时间限制**：**作为一个独立开发者，所有开发进度取决于个人时间安排。某些阶 段可能会卡壳，花费过多时间，从而降低整体效率。  
   ****规避方式**：**制定每周里程碑计划，如有必要，砍掉非核心复杂功能。

****项目范围膨胀**：**尝试过多实验性功能，可能会延迟核心玩法的开发。  
   ****规避方式**：**在早期阶段确定最终设计，并加以锁定，避免中途频繁变更。

****测试阶段不丰富**：**由于是独立开发，测试时间可能较短，且测试角度单一，容 易遗漏部分Bug。  
   ****规避方式**：**尽可能提前安排测试周期，邀请外部玩家试用。

### **3.3 其他可能的风险**

****UI 复杂性**：**需要管理多个 UI 状态（主菜单、暂停、选项、游戏结束），若结 构混乱，容易出现逻辑错误或状态切换不一致。  
 ****规避方式**：**清晰设计 UI 状态流转。

****玩家困惑**：**由于游戏规则创新，与传统 2048 差异较大，可能导致玩家一开始 难以上手。  
   ****规避方式**：**提供简洁明了的新手提示或说明界面，帮助玩家理解玩法。