

此次编程作业的目标是针对2048游戏设计智能体算法。README中给出了详细的运行指导、目标和需要使用到的函数接口，在仔细阅读之后，实现本次编程作业的基础要求应该是不难的。本次作业的代码框架也比较简单，感兴趣的同学可以深入学习一下。

补充讲解说明

🔗 你可以阅读game的代码设计来验证下面的讲解内容

框架说明

- 在本次作业中，2048游戏被设计为了**你的智能体与方块随机添加者**之间的博弈过程——在游玩2048时，每一次移动-合并操作之后，都会出现一些新的方块，这个过程在此次作业中被分解为两个玩家的博弈：智能体进行滑动合并，而另一个参与者进行方块的随机添加。
- 在此视角下，我们可以更加明确地理解接口设计
 - `set_action()` 在 `game` 类的 `action` 成员中设置行动变量，需要结合 `forward` 方法才能真正发生游戏状态的更新
 - `forward()` 方法是 `forward_player_only()` 和 `add_random_tile()` 的合并，前者仅仅进行滑动-合并操作，后者则在空白区域随机放置新方块
 - `get_valid_actions()`：获取**你的智能体**的可选行动
 - `get_valid_successors()`：针对**滑动-合并后，随机添加方块前的状态**，获取接下来随机添加方块后的状态分布
- 你的任务是使用Expectimax方法，基于**随机添加方块的概率分布**计算收益期望，并确定行动策略

作业要求

基础部分（100）

一、正确实现单步Expectimax方法（70）

- 在单步Expectimax方法中，针对你的智能体每一个行动结果的状态模拟（阅读代码实现中的行动模拟是怎样实现的）
- 进一步获取其**随机添加方块后的状态概率分布**，使用启发式函数作为每一个状态的效用估计
- 给出完整的实现、实验结果以及对实验结果的简单分析

二、实现更深的Expectimax搜索（20）

- 对于模拟的随机添加方块后的状态，进一步使用Expectimax方法对它进行效用估计
- 基础部分的要求是实现**两步的**Expectimax方法，即，对模拟状态使用单步Expectimax方法
 - 你也可以在实验中尝试不同的搜索深度进行对比
- 给出完整的实现、实验结果以及对实验结果的简单分析

三、按时提交你的作业（10）

- 在2025.5.26 23:59之前提交作业的压缩包
- 压缩包中应该包含Lab3的必要代码文件以及你的实验报告

🔗 你可以直接实现目标二的要求，并在实验结果和分析中一并分析单步的情形。

Bonus 部分（30）

一、设计并分析不同启发式函数的效果（10）

二、本次作业中提到的其它智能体算法的实现 (20)

- Rollout智能体：通过**多次模拟实验**来估计某一个行动或者状态的效用值
- MCTS智能体：[Monte Carlo Tree Search - About](#)
- 你需要给出完整的实现、实验结果以及分析和说明

Just For Fun

作业分数在期末的最终结算时会在总和的上限处**截断**

Bonus部分的用意是弥补在实现基础部分时可能出现的一些小瑕疵，让大家都能有满意的分数
不要太卷