

蛇和梯子

在这个项目中，您需要使用Java编程语言实现一个游戏Snakes and Ladders。

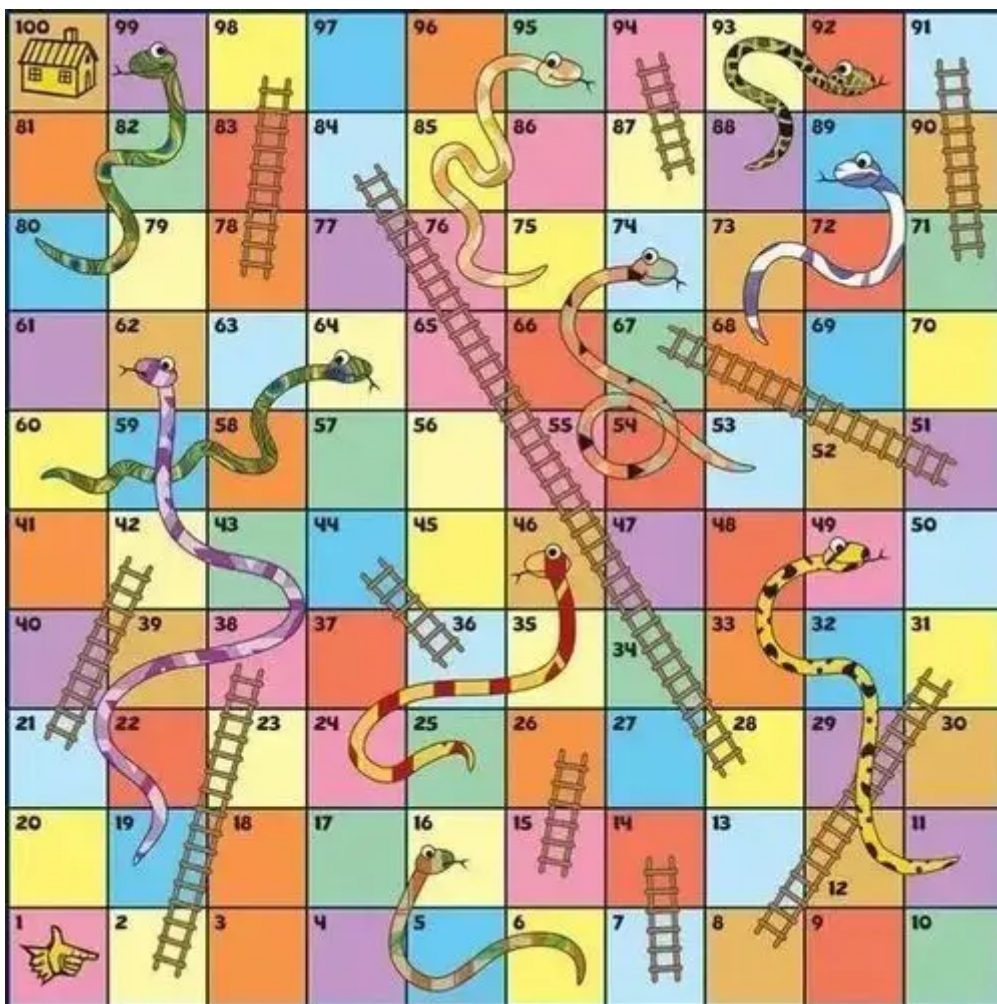
I. 游戏介绍

蛇梯棋是一种经典的双人或多人棋盘游戏，起源于古印度。该游戏由一个编号的网格组成（通常是10x10），玩家通过掷骰子来竞速到达最后一个方格。

如何玩：

- 玩家轮流掷骰子，然后根据骰子的结果向前移动自己的标记。
- 在梯子底部着陆可以让你爬到更高的方块上，加快进度。
- 落在蛇的头上会让你滑到它的尾巴上，让你倒退。
- 第一个到达最后一格的玩家（需要精确的掷骰）获胜！游戏将运气与

简单的策略结合在一起，让所有年龄段的人都能玩得开心。



二、游戏界面与基本元素

游戏界面包括：

- 一个10×10的游戏板，每个单元格的序列号标记在左上角。

- 每个玩家控制一个游戏棋子，必须显示在棋盘上。
- 每个玩家的控制按钮，交替使用掷骰子、生成骰子数字和移动标记。
- 触发奖励（例如，额外的步骤）或惩罚（例如，向后移动）的特殊细胞。为了简化项目，我们使用“+步”代替梯子和“-步”代替蛇。
- 玩家只有当他们的棋子正好落在第100个格子上时才能获胜。

三、比赛细则

1. 比赛顺序：

- 玩家一个接一个地掷骰子，并在所有回合中保持游戏顺序。

2. 移动您的代币：

- 掷骰子，然后将棋子向前移动所示的格数。
- 如果您登陆到：
 - 梯子（+级台阶）：向前移动几步。
 - 蛇形/跌落（-步数）：向后移动几步。

3. 获胜的

- 你必须掷出精确的数字才能落在第100格。如果掷出的数字超过100，你的棋子就会弹回（例如，如果你在97格上掷出5，就移动到98、99、100、99、98，落在98格）。

四、项目要求

在这个项目中，你需要实现多个基本任务，涵盖蛇梯游戏的基本功能，同时还可以实现一些高级任务，使你的程序更加强大。当我们评估你的程序并发现某个任务已实现且功能基本可用时，你将获得相应的分数。

任务1：游戏初始化(10分)

1. 用户登录或选择访客模式后，游戏将显示棋盘。
2. 游戏应该允许玩家在游戏过程中随时重新开始新游戏。（而不是退出程序，然后再次运行。）
3. 重新启动新游戏时，游戏数据需要与新游戏保持一致。
4. 游戏界面必须包括：
 1. 游戏板
 2. 骰子滚动结果显示
 3. 每个玩家的单个滚轮按钮

董事会必须显示这些基本要素：

- 所有球员的当前位置
- 网格编号（单元格标识符）
- 特殊网格空间（例如，+N/-N步）”

任务2：多用户登录(20分)

1. 为客人和注册用户实现登录选择界面。
2. 客人可以不注册玩游戏，但没有保存游戏进度的功能。
3. 用户登录界面包括注册页面，输入账号凭证后可进行登录。
4. 程序退出并重新运行后，以前注册的用户仍然可以登录。

任务3：保存和加载游戏(20分)

1. 每个用户（除访客外）都可以选择加载他们之前保存的游戏；保存是一个单独的保存文件，再次保存会覆盖之前的保存（覆盖原始保存是基本要求。如果为每个用户实现多个保存槽位，则不会获得额外积分。）
2. 从游戏开始界面，玩家可以选择加载他们上次保存的游戏，其中应该包含关于游戏板的状态和到目前为止的移动次数的信息。
3. 每个用户的保存数据是唯一的。
4. 手动保存是一项基本要求；在定时间隔或退出时自动保存可以赢得高级部分的分数。
5. 保存文件错误检查：如果加载时保存文件的格式或内容损坏，受损的保存文件将不会被加载，游戏仍会继续运行而不是崩溃。（如果您的游戏能够检测到被他人修改过的保存文件，同时仍能保持保存数据的有效性，它将获得高级积分。）

任务4：游戏玩法(30分)

1. 游戏顺序：基本要求指定了一个双人游戏，玩家严格交替地掷骰子。程序必须通过阻止非活跃玩家掷骰来强制执行轮次顺序——当轮到A玩家时，B玩家的掷骰按钮应被禁用，反之亦然。系统需要在整个游戏过程中保持正确的轮次顺序，确保A玩家和B玩家只能在各自的轮次内掷骰。
2. 每回合：每个玩家点击自己的掷骰子按钮，得到一个从1到6的随机数字，这个数字就是玩家需要移动的步数。然后玩家的棋子应该向前移动步数。

如果该片段落在特殊网格上：

- 梯子（+级台阶）：向前移动几步。
- 蛇形/跌落（-步数）：向后移动几步。

玩家必须掷出正确的数字才能落在第100格上，才能赢得游戏。如果掷出的数字超过100，玩家的棋子会弹回（例如，如果在97格上掷出5，移动到98、99、100、99、98，落在98格上）。

3. 游戏胜利：检查玩家是否正好落在100号方格上。如果是，显示该玩家的胜利画面。在第一个玩家获胜后，游戏可以终止，或者继续进行剩余玩家的游戏（跳过获胜者的回合）-这两种实现都可以获得此要求的分数。

任务5：图形用户界面（GUI）(10分)

1. 使用JavaFX、Swing或任何其他Java图形框架实现游戏的图形界面。
2. 您将通过根据课程中提供的演示完成代码来获得本节的积分。
3. 独立创建GUI将计入高级积分。
4. 如果您的程序需要在命令行中输入，则无法获得此任务的全部分数。

任务6：高级功能(30分)

超出上述基本要求的任何附加功能都将获得此高级类别的积分，包括但不限于：

1. 增强的图形和美观性
2. 更多游戏模式设计
3. 添加一些动画效果
4. 在游戏中添加道具
5. 添加您自己设计的地图6....

V. 团队合作和评分

你将与2-3名成员组成一个小组，所有成员都应来自同一个实验室部门。通过完成任务（基础：最高90分+高级：最高30分），你将获得一个基础分数。然后，我们将根据团队规模调整分数：

2名成员：调整后的=基线* 1.0

3名成员：调整后的=基线* 0.9

成员可以在小组内调整自己的贡献比例，该比例应在0.8至1.2之间。所有成员的贡献比例之和应等于成员总数。调整贡献比例需通知并获得所有成员的同意，教师和助教不负责协调这些调整。

此外，本项目将有两个截止日期：

提前截止日期（第15周）：在5月27日23：59之前将代码提交到Blackboard，并在第15周的实验课上展示。为了鼓励提前提交，将奖励5%的奖金。

定期截止日期（第16周）：在6月3日23：59之前将代码提交到Blackboard，并在第16周的实验课上展示。所有剩余的小组必须在截止日期前完成工作。

个人最终得分=最小值（110，基线*团队规模调整*贡献率*截止日期调整（1或1.05））

个人最终得分不得超过110分。

展示您的作品

你需要在第15/16周的实验课上进行课堂演示。所有团队成员都必须到场。

演示文稿应包括对项目设计的简要介绍，例如，如何持久化数据，如何管理CLI的状态等。您可以准备一张幻灯片（PPT）来帮助您展示工作，但不是必需的。

然后，你应当运行你的程序并演示它的功能。请提前准备好自己的流程，以便能够顺利、充分地展示完成的任务。老师和助教可能会在你演示的整个过程中提问，以验证工作的真实性并更准确地评分。