Python程序设计

案例：井字棋游戏程序

张 华

井字棋游戏程序

目的

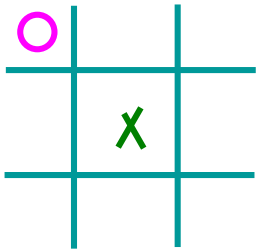
- 通过井字棋游戏程序的设计和实现，了解Python函数的定义和使用。
- 深入了解使用数据结构和算法实现游戏的人工智能。
- 井字棋游戏包括较为复杂的计算机人工智能（AI）落子算法、判断输赢算法等，通过把不同功能定义为独立的函数，可以减少程序的复杂性。

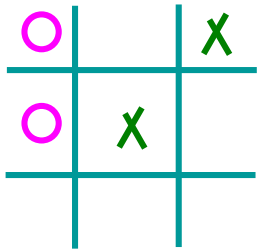
井字棋游戏程序

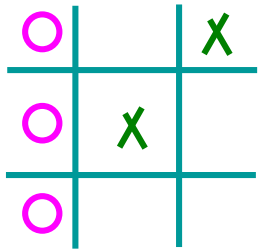
问题

游戏规则：

- 玩家和计算机轮流下棋子。
- 每次一方只能在某一个空格处下一颗棋子。
- 胜负判断：
  - 若棋盘的某一行，或某一列，抑或某一对角线上的三个格子被某一方的棋子占据，则该方胜利；
  - 否则，为平局。



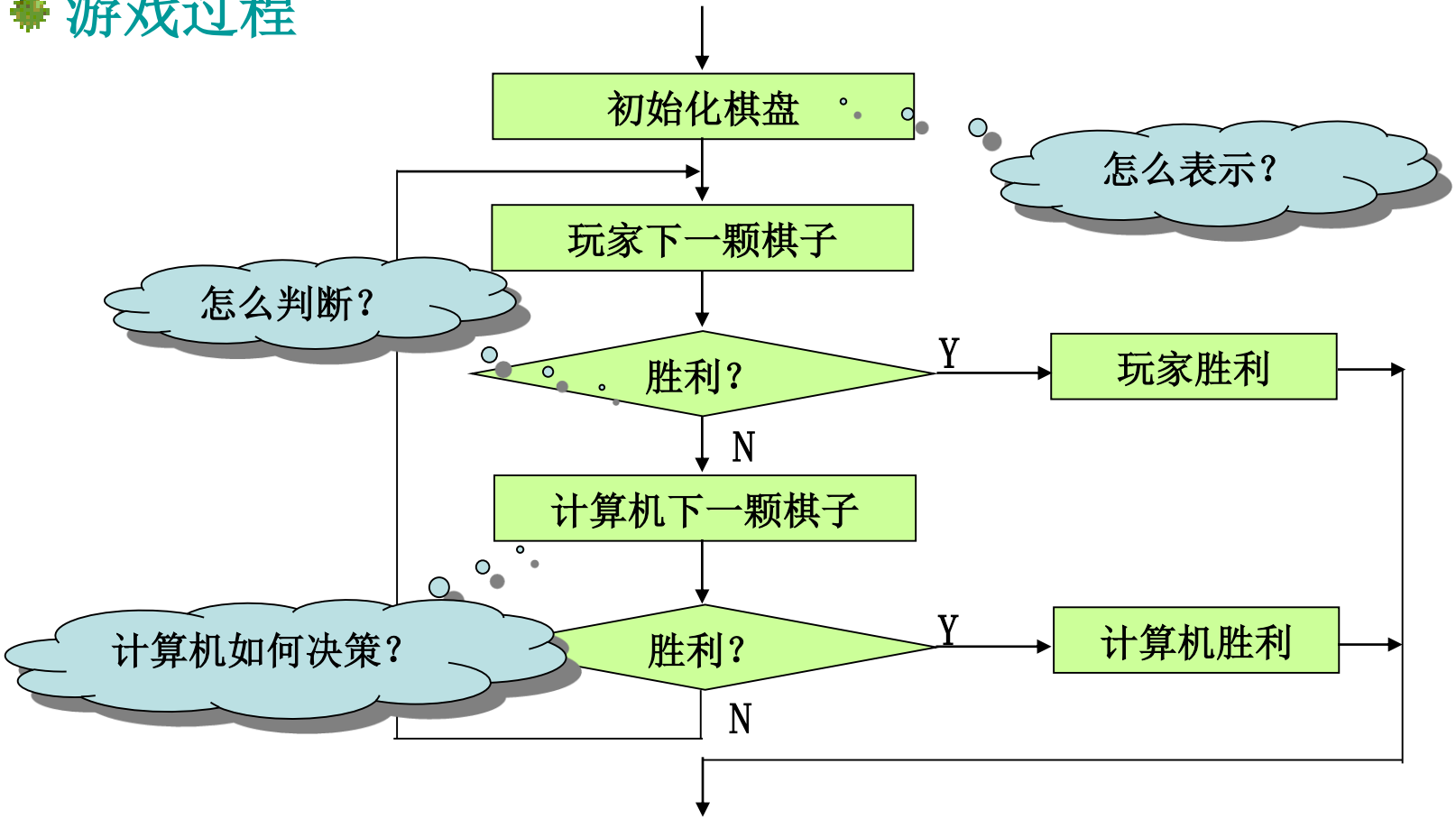




井字棋游戏程序

分析

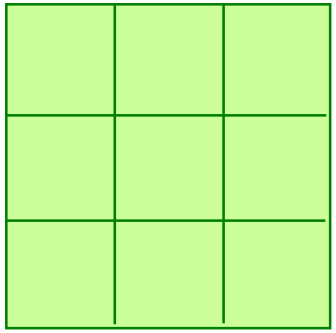
游戏过程



井字棋游戏程序

设计

数据结构



board =

```
['0', '1', '2',
 '3', '4', '5',
 '6', '7', '8']
```

- 先手的棋子用‘X’表示；
- 后手的棋子用‘O’表示。

一步棋

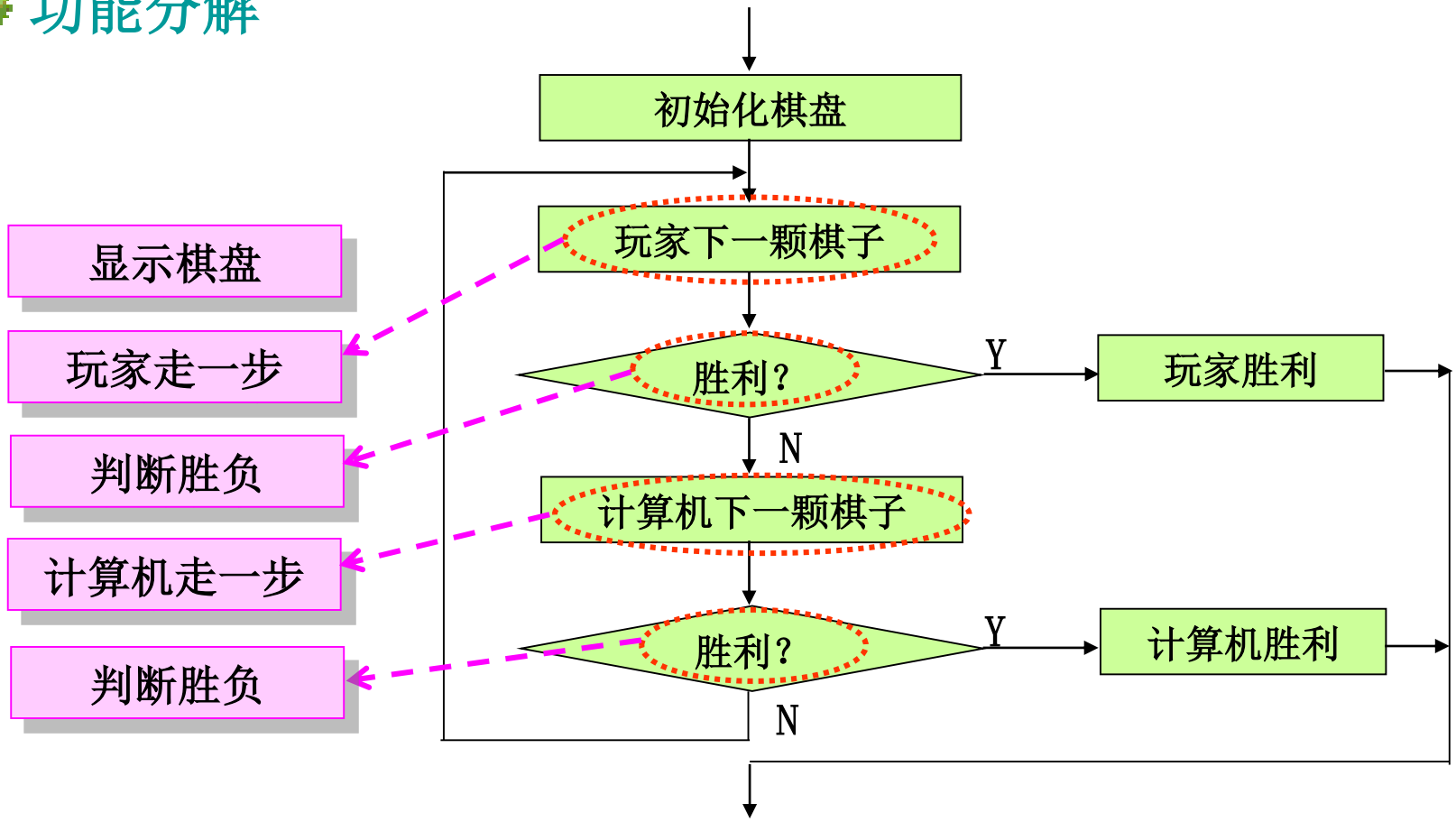
确定棋子的位置：下标

直接修改对应元素的值为棋子字符

井字棋游戏程序

设计

功能分解



井字棋游戏程序

设计

功能模块

井字棋游戏程序

设计思路

棋盘采用包含 9 个元素的列表来表示，board[0]到 board[8]存储代表棋子的字符串，字符串中可以包含"X"、"O"、或者数字 0 到 8（表示未落子）。

程序的流程如下：

（1）初始化棋盘；

（2）询问玩家选择棋子：棋子 X 先走，棋子 O 后走；

（3）显示棋盘及落子布局（调用函数 display_board()）；

（4）循环轮流落子：

（4-1）如果玩家落子，则询问落子位置（调用函数 getPlayerMove()，然后判断玩家是否获胜（调用函数 isWinner()），如果获胜，显示棋盘（调用函数 display_board()），输出信息，break 跳出循环；

（4-2）如果计算机人工智能（AI）落子，则根据计算机人工智能（AI）落子算法计算落子位置，然后判断 AI 是否获胜（调用函数 isWinner()），如果 AI 获胜，则显示棋盘（调用函数 display_board()），输出信息，break 跳出循环；

（4-3）判断是否平局（调用函数 isTie()），如果平局，则显示棋盘（调用函数 display_board()），输出信息，break 跳出循环；否则继续轮流落子。

井字棋游戏程序

设计思路

计算机人工智能（AI）落子算法如下：

（1）如果某位置落子可以获胜，则选择该位置；

（2）否则，如果某个位置玩家下一步落子可以获胜，则选择该位置；

（3）否则，按中心（4）、角（0、2、6、8）、边（1、3、5、7）顺序选择空的位置。

判断输赢规则如下：如果三条横线（(0,1,2), (3,4,5), (6,7,8)）、三条竖线（(0,3,6), (1,4,7), (2,5,8)）、两条对角线（(0,4,8), (2,4,6)）共八种情况的三个位置的棋子相同，则该棋子方赢棋。如果全部位置落子，则平局。

“井”字棋游戏程序

实现

- 看源代码：tic_tac_toe.py

```
def displayBoard(b):
    """显示棋盘"""
    print("\t{0}|{1}|{2}".format(b[0],b[1],b[2]))
    print("\t-|-|-")
    print("\t{0}|{1}|{2}".format(b[3],b[4],b[5]))
    print("\t-|-|-")
    print("\t{0}|{1}|{2}".format(b[6],b[7],b[8]))
```

“井”字棋游戏程序

实现

- 看源代码：tic_tac_toe.py

```
def legalMoves(board):
    """返回可落子的位置列表"""
    moves = []
    for i in range(9):
        if board[i] in list("012345678"):
            moves.append(i)
    return moves

def getPlayerMove(board):
    """询问并确定玩家(player)选择落子位置，无效位置时重复询问"""
    move = 9 # 初始值9为错误的位置
    while move not in legalMoves(board):
        move = int(input("请选择落子位置(0-8):"))
    return move
```

“井”字棋游戏程序

```
def getComputerMove(board, computerLetter, playerLetter):
    """计算人工智能AI的落子位置, Tic Tac Toe AI核心算法"""
    boardcopy = board.copy() #拷贝棋盘，不影响原来的值

    # 规则1：判断如果某位置落子可获胜，则选择该位置
    for move in legalMoves(boardcopy):
        boardcopy[move] = computerLetter
        if isWinner(boardcopy, computerLetter): #判断是否获胜
            return move
        boardcopy[move] = str(move)

    # 规则2：某个位置玩家下一步落子可获胜，则选择该位置
    for move in legalMoves(boardcopy):
        boardcopy[move] = playerLetter
        if isWinner(boardcopy, playerLetter): #判断是否获胜
            return move
        boardcopy[move] = str(move)

    # 规则3：中心（4）、角（0、2、6、8）、边（1、3、5、7）顺序选择空的位置
    for move in (4,0,2,6,8,1,3,5,7):
        if move in legalMoves(board):
            return move
```

“井”字棋游戏程序

实现

- 看源代码：tic_tac_toe.py

```
def isWinner(board, letter):
     """判断所给的棋子是否获胜"""
    WAYS_TO_WIN = {(0,1,2),(3,4,5),(6,7,8),(0,3,6),(1,4,7),(2,5,8),
    (0,4,8), (2,4,6)}
    for r in WAYS_TO_WIN:
        if board[r[0]]==board[r[1]]==board[r[2]]:
            return True
    return False

def isTie(board):
    """判断是否平局"""
    for i in list("012345678"):
        if i in board:
            return False
    return True
```

“井”字棋游戏程序

实现

看源代码：tic_tac_toe.py

```
def tic_tac_toe():
    """井字棋"""
    #初始化棋盘为['0', '1', '2', '3', '4', '5', '6', '7', '8']
    board = list("012345678")

    #询问玩家选择棋子：棋子X先走，棋子O后走
    playerLetter = input("请选择棋子X或O（X先走，O后走）：")
    if playerLetter in ("X", "x"):
        turn = "player"    #玩家先走
        computerLetter = "O"
    else:
        turn = "computer"
        computerLetter = "X"
        playerLetter = "O"
    print("{}先走!".format(turn))
```

“井”字棋游戏程序

实现

- 看源代码：tic_tac_toe.py

```
while True: #循环轮流落子
      displayBoard(board)
      if turn == 'player':  #玩家落子
          move = getPlayerMove(board) #询问落子位置
          board[move] = playerLetter  #落子
          if isWinner(board, playerLetter): #判断是否获胜
              display_board(board)
              print('恭喜玩家获胜!')
              break
          else:
              turn = "computer"
```

“井”字棋游戏程序

实现

看源代码：tic_tac_toe.py

```
else:   #计算机人工智能AI落子
        # 计算人工智能计算AI落子位置
        move = getComputerMove(board, computerLetter, playerLetter)
        print("计算机人工智能AI落子位置: ", move)
        board[move] = computerLetter   #落子
        if isWinner(board, computerLetter): #判断是否获胜
            displayBoard(board)
            print('计算机人工智能AI获胜!')
            break
        else:
            turn = "player"
```

“井”字棋游戏程序

实现

看源代码：tic_tac_toe.py

```
            #判断是否平局
            if isTie(board):
                displayBoard(board)
                 print('平局!')
                break

if __name__ == '__main__':
    tic_tac_toe()
```