AI Teris Game

李玉奇 1710229
李宗昱 1710641
刘雨濠 1711109
张美涵 1710182

2020 年 12 月 15 日

1 AI_Teries

这是一个俄罗斯方块的实现程序，其中添加了基于 Pierre Dellacherie 算法的 AI。
本机环境:python3.7 pygame 1.9.6

2 库安装

需要安装 pygame，安装指令:
pip install pygame==1.9.6
鉴于下载国外资源较慢，可以换成国内的豆瓣源，安装指令:
pip install pygame -i https://pypi.douban.com/simple
安装成功后会显示如下信息：
Installing collected packages: pygame
Successfully installed pygame-1.9.6
安装成功后，在 python 的交互式界面输入 import pygame，系统不会报错。

3 文件说明

font 文件夹里面存的是字体，可以根据自己的需要换;
score.txt 里面存的是每次分数，用来记录最高记录;
tetris.py 是游戏的主程序, 可以 clone 下来这个仓库直接使用, 按 a 进入或者退出 AI 模式;
mp4 文件是程序运行效果的一个例子。

4 程序说明

代码主体为两部分，第一部分是用 python 实现俄罗斯方块，第二部分是基于 Pierre Dellacherie 算法的 AI 功能实现。其中用 python 实现俄罗斯方块主要有三个类，分别是：
Wall: 实现程序的各个界面即消除计分功能；
Brick: 实现板块的各个信息；
HouseWorker：控制器，控制游戏不同状态的进程，包括开始游戏、暂停游戏、重新开始功能。

基于 Pierre Dellacherie 算法的 AI 功能实现主要由 RobotWorker 类构成，其记录板块信息与评分函数各个参数的计算。

4.1 Wall 类函数及变量说明

draw_grids(): 画游戏背景方格
draw_matrix(): 根据二维矩阵绘制已存在的格子
remove_full_line(): 消除一行并计分
show_text()：封装的写文字的函数
draw_score(): 记录分数
showPause(): 暂停界面
show_welcome(): 欢迎界面
drawALL() 封装的函数，用于绘制背景，分数，格子等组件
drawNextBrick(): 绘制下一个方块
getHeightScore(): 获取最高分
writeHeightScore(): 写入最高分
showAD(): 展示我组强烈的征婚内容

4.2 Brick 类的函数及变量说明

SHAPES：所有方块的名字
SHAPE_WITH_DIR：存储所有方块的所有形态
shape：方块的名字
center：方块的中心点
dir：方块的形态
color：方块的颜色
init(): 构造函数，随机生成方块以及它的颜色、形态、中心点
get_al_gridpos(): 根据中心坐标获取方块获取其它点的坐标
conflict(): 碰撞检测，检测是否碰到边和底部
rotate(): 方块变形
left(): 方块左移
right(): 方块右移
draw(): 绘制当前方块
drawNext(): 绘制下一个方块

4.3 HouseWorker 类函数及变量说明

start(): 开始游戏
pause(): 暂停游戏
whenPause(): 暂停时运行的游戏程序
whenNormal(): 正常运行时的游戏程序
whenGameOver(): 结束运行时的游戏程序

4.4 RobotWorker 类函数及变量说明

SHAPES: 所有方块的名字
I、J 等字母：记录字母对应方块的点集
SHAPES_WITH_DIR: 简称与实际图形点集的对应
center: 中心点坐标
shape: 方块形状
color: 方块颜色
station: 当前状态 (方块的那种方式)
matrix: 记录局面的矩阵
get_all_gridpos(self, center,shape,dir): 返回此中心下该方块占据的位置
conflict(self, center,matrix,shape,dir): 检验此方块位置是否合理，主要检测是否超出屏幕之外和占据的位置是否先前已被占据
copyTheMatrix(self,screen_color_matrix): 复制原先的颜色矩阵
getAllPossiblePos(self,thisShape = 'Z'): 获取所有可能的位置 (按照每一列，从上到下检测每一个位置，如果当前位置可以放置，而且下一个位置是不可以放置的，那么当前位置就是一个可能放置方块的位置。for 循环每种形态)
getLandingHeight(self,center): 获取中心点高度
getErodedPieceCellsMetric(self,center,station): 获取消除的行数和自身贡献的方格数 (满一行之后，检测这一行的每一个坐标是否在方块的所有点的坐标矩阵之内。从下网上开始搜索，一但一行里面没有一个实心方格则跳出循环)
getNewMatrix(self,center,station): 把可能的坐标位置放进去颜色矩阵，形成新的颜色矩阵
getBoardRowTransitions(self,theNewmatrix): 获取行变换数
getBoardColTransitions(self,theNewmatrix): 获取列变换数

getBoardBuriedHoles(self,theNewmatrix): 获取空洞数 (按照列开始检测，从上往下，碰到有实心方格把 colHoles 设为 0，继续往下，碰到空心方格则 +1。每一列循环后加入到总的空洞数里)
getBoardWells(self,theNewmatrix): 获取“井”的数量以及返回“井”数的连加数 (每一列从上往下开始检测，碰到空心方格而且两边是墙或者实心方格的时候，“井”深加 1，继续往下检测，再次碰到空心方格则“井”深加 1，若碰到实心方格则重新开始统计“井”深，并把当前“井”加入到总数)
getPrioritySelection(self,point): 计算优先度函数
evaluateFunction(self,point): 根据点的中心位置计算分数