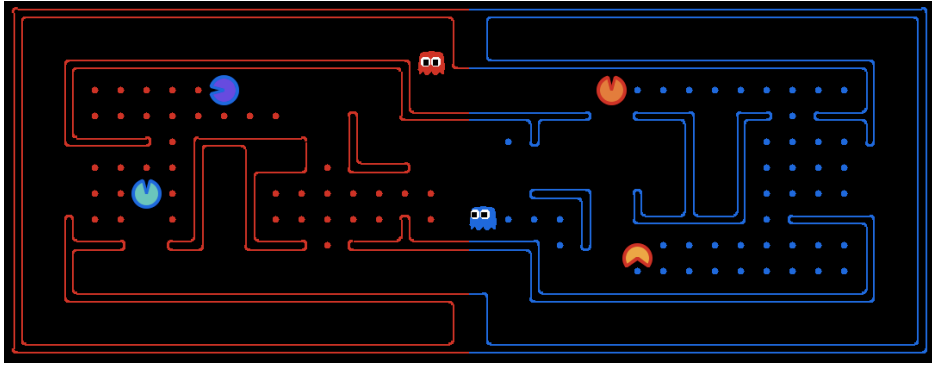


比赛：吃豆人夺旗



足够的防御，
进入敌人的地形。
捕获他们所有的食物！

介绍

课程竞赛涉及 Pacman 的多人夺旗变体，其中特工以协调的团队策略控制 Pacman 和幽灵。你的团队将尝试吃掉地图另一边的食物，同时保卫你家乡的食物。比赛代码以 [zip 存档](#) 的形式提供。

要读取的关键文件：

capture.py	在本地运行游戏的主文件。此文件还介绍了新的捕获标志 GameState 类型和规则。
captureAgents.py	捕获代理的规范和帮助程序方法。
baselineTeam.py	定义两个非常基本的反射代理的示例代码，以帮助您入门。
myTeam.py	在这里，您可以定义自己的代理以包含在夜间锦标赛中。（这是您提交的唯一文件。

支持文件（请勿修改）：

game.py	吃豆人世界运作背后的逻辑。此文件描述了几种支持类型，如 AgentState、Agent、Direction 和 Grid。
util.py	用于实现搜索算法的有用数据结构。
distanceCalculator.py	计算所有迷宫位置之间的最短路径。
graphicsDisplay.py	Pacman 的图形
graphicsUtils.py	支持 Pacman 图形
textDisplay.py	Pacman 的 ASCII 图形
keyboardAgents.py	控制 Pacman 的键盘界面
layout.py	用于读取布局文件并存储其内容的代码

学术不诚实：虽然我们不会对比赛进行评分，但我们仍然希望您不要虚假地陈述您的作品。请不要让我们失望。

吃豆人夺旗规则

布局：吃豆人地图现在分为两半：蓝色（右）和红色（左）。红色特工（都有偶数指数）必须在尝试吃蓝色食物的同时保护红色食物。当在红色一侧时，红色特工是幽灵。当进入敌方领土时，特工会变成吃豆人。

得分：当吃豆人吃到一个食物点时，食物会被永久移除，并为该吃豆人的团队得一分。红队得分为正数，而蓝队得分为负数。

吃吃豆子：当吃豆人被对方的鬼魂吃掉时，吃豆人会回到它的起始位置（作为鬼魂）。吃掉对手不会获得积分。

动力囊：如果吃豆人吃了一个能量胶囊，对方团队的特工会在接下来的 40 步中变得“害怕”，或者直到他们被吃掉并重生，以较早者为准。“害怕”的特工在以幽灵的形式（即在自己团队这边）很容易被吃豆人吃掉。具体来说，如果吃豆人与“害怕”的幽灵相撞，吃豆人不受影响，幽灵在其起始位置重生（不再处于“害怕”状态）。

观察：只有当对手或他们的队友在 5 格（曼哈顿距离）以内时，智能体才能观察对手的配置（位置和方向）。此外，智能体总是会获得棋盘上每个智能体的嘈杂距离读数，这可用于近似定位未观察到的对手。

优胜：当一支球队吃掉对手的两个点时，游戏就结束了。游戏也限制为 1200 次代理移动（四个代理每人 300 次移动）。如果达到这个移动限制，吃得最多的队伍获胜。如果分数为零（即平局），则将其记录为平局。

计算时间：我们将在 Amazon EC2 小型实例上运行您的提交，该实例具有 1.2 Ghz Xeon / Opteron 等效处理器和 1.7GB RAM。每个代理有 1 秒的时间返回每个操作。每一步在一秒钟内没有返回，就会招致警告。在三次警告后，或任何一次移动超过 3 秒，游戏将被没收。初始启动余量为 15 秒（使用该功能）。如果代理超时或以其他方式引发异常，则日志文件中将显示一条错误消息，您可以从结果页下载该文件（见下文）。registerInitialState

投稿须知

要参加夜间锦标赛，您的团队必须在 中定义。由于锦标赛的运行方式，您的代码不得依赖于我们未提供的任何其他文件（提交系统可能允许您提交其他文件，但当您的代码在锦标赛中运行时，竞赛框架不会包含它们）。您不得修改我们提供的代码。myTeam.py

还必须在 中指定唯一的团队名称。此文件应仅包含 ASCII 字母和数字（任何其他字符（包括空格）将被忽略）。只有您的团队名称将显示给班级的其他成员。一旦您以特定名称提交，只有您被允许以该名称提交，直到您放弃该名称，以不同的团队名称提交或以空白团队名称提交，这会将您从锦标赛中删除。像往常一样，您还应该包括一个文件。name.txtpartners.txt

若要实际提交，请使用作业名称下方的 。[比赛提交](#)网页显示所有团队的团队名称和最近提交时间，您可以使用它们来验证您的提交是否已正确收到。本网站应在您提交后两分钟内更新。submitcontest

开始

默认情况下，您可以使用工作人员提供的简单方法运行游戏：baselineTeam

```
python capture.py
```

您有多种选择：

```
python capture.py --help
```

代理有四个插槽，其中代理 0 和 2 始终在红队，1 和 3 在蓝队。代理由代理工厂创建（一个用于红色，一个用于蓝色）。有关上面调用的代理的说明，请参阅有关设计代理的部分。我们提供的唯一团队是默认情况下，它同时被选为红队和蓝队，但作为如何选择队的一个示例：它指定红队和蓝队都是从 创建的。要用键盘控制四个特工之一，请传递相应的选项：箭头键控制你的角色，当越过中心线时，这个角色会从幽灵变为吃豆人。baselineTeam

```
python capture.py -r baselineTeam -b baselineTeam
```

```
-r-bbaselineTeam.py
```

```
python capture.py --keys0
```

布局

默认情况下，所有游戏都在布局上运行。若要在其他布局上测试代理，请使用该选项。具体而言，您可以通过指定 例如，将使用种子 13 随机生成的地图。defaultcapture-lRANDOM[seed]-l RANDOM13

游戏类型

您可以通过两种方式玩游戏：本地游戏和夜间锦标赛。

本地游戏（如上所述）允许您根据我们提供的基线团队测试您的代理，这些团队旨在用于开发。

官方锦标赛

实际比赛将在 Amazon EC2 集群（1.2 Ghz Xeon / 1.7GB RAM 机器）上使用夜间自动锦标赛进行，决赛决定最终比赛结果。有关如何让团队参加锦标赛的详细信息，请参阅提交说明。比赛每天晚上 11 点左右举行，包括截至比赛开始时已提交的所有球队（当天早些时候或前一天）。目前，每支球队与其他球队交手 9 次（以减少随机性），但这可能会在学期晚些时候发生变化。比赛中使用的布局将从 zip 文件中包含的默认布局以及每晚随机生成的布局中提取。所有布局都是对称的，最先移动的团队是随机选择的。比赛结果每晚都会在网上更新 - 在这里您可以查看每场比赛的总排名和比分。您还可以下载重播、使用的布局以及每个代理的 stdout/stderr 日志。

与项目 2 不同的是，特工现在有更复杂的工作，即在进攻与防守之间进行权衡，并在团队环境中有效地充当幽灵和吃豆人。此外，提供给您的代理的有限信息可能需要一些概率跟踪（如项目 4）。最后，计算时间限制的增加带来了新的挑战。

基线团队：为了启动您的代理设计，我们为您提供了一个由两个基线代理组成的团队，定义在 `baselineTeam.py` 中。他们俩都很糟糕。向对面最近的食物移动。它自己四处游荡，试图追赶它碰巧看到的入侵者。

文件命名：为了在本地测试或运行游戏，您可以在任意命名的 python 文件中定义代理团队。但是，在提交参加夜间锦标赛时，您必须在其中定义您的代理（并且您还必须创建一个指定您的团队名称的文件）。`myTeam.pyname.txt`

接口：`in` 应该看起来很熟悉，但包含新的方法，例如 `getFood`，它在红色一侧获得食物网格（请注意，网格是板的大小，但仅适用于红色一侧有食物的单元格）。另请注意，您可以使用 `getFoodYouAreDefending` 列出团队的索引，或使用 `getOpponents` 测试成员资格。

最后，您可以通过 `getMazeDistance` 访问嘈杂的距离观测列表。这些距离在真值的 6 以内，噪声是从 `[-6, 6]` 范围内随机均匀选择的（例如，如果真实距离为 6，则 `{0, 1, ..., 12}` 中的每一个都以 `1/13` 的概率选择）。您可以使用 `getAgentDistances` 获得嘈杂读数的可能性。

距离计算：为了方便智能体开发，我们提供了代码以提供最短的路径迷宫距离。`distanceCalculator.py`

若要开始设计自己的代理，建议对类进行子类化。这提供了对几种便利方法的访问。一些有用的方法是：`CaptureAgent`

```
def getFood(self, gameState):
    """
    Returns the food you're meant to eat. This is in the form
    of a matrix where m[x][y]=true if there is food you can
    eat (based on your team) in that square.
    """

def getFoodYouAreDefending(self, gameState):
    """
    Returns the food you're meant to protect (i.e., that your
    opponent is supposed to eat). This is in the form of a
    matrix where m[x][y]=true if there is food at (x,y) that
    your opponent can eat.
    """

def getOpponents(self, gameState):
    """
    Returns agent indices of your opponents. This is the list
    of the numbers of the agents (e.g., red might be "1,3,5")
    """

def getTeam(self, gameState):
    """
    Returns agent indices of your team. This is the list of
    the numbers of the agents (e.g., red might be "1,3,5")
    """

def getScore(self, gameState):
    """
    Returns how much you are beating the other team by in the
    form of a number that is the difference between your score
    and the opponents score. This number is negative if you're
    losing.
    """

def getMazeDistance(self, pos1, pos2):
    """
    Returns the distance between two points; These are calculated usir
    distancer object.

    If distancer.getMazeDistances() has been called, then maze distanc
    Otherwise, this just returns Manhattan distance.
    """

def getPreviousObservation(self):
    """
    Returns the GameState object corresponding to the last
    state this agent saw (the observed state of the game last
    time this agent moved - this may not include all of your
    opponent's agent locations exactly).
    """

def getCurrentObservation(self):
    """
```

```

Returns the GameState object corresponding this agent's
current observation (the observed state of the game - this
may not include all of your opponent's agent locations
exactly).
"""

def debugDraw(self, cells, color, clear=False):
    """
    Draws a colored box on each of the cells you specify. If clear is
    will clear all old drawings before drawing on the specified cells.
    This is useful for debugging the locations that your code works wi

    color: list of RGB values between 0 and 1 (i.e. [1,0,0] for red)
    cells: list of game positions to draw on (i.e. [(20,5), (3,22)])
    """

```

限制：您可以自由设计任何您想要的代理。但是，如果您想参加比赛，则需要遵守提供的 API。在对手回合内计算的特工将被取消资格。特别是，任何形式的多线程都是不允许的，因为我们发现很难确保在对手的回合中不进行计算。

警告：如果您的某个代理在夜间锦标赛的比赛中产生任何 stdout/stderr 输出，则该输出将包含在网站上发布的比赛结果中。此外，在某些情况下，如果其中一个代理引发异常，则此输出中可能会显示堆栈跟踪。在设计代码时，应确保不会公开任何要保密的信息。

比赛详情

团队：您可以在最多 4 人的团队中工作。

奖品：排名是根据在夜间循环赛中获得的积分确定的，其中获胜值 4 分，平局值 1 分，输球值 0（平局不值太多以阻止僵局）。要参加夜间锦标赛，您的提交必须在当晚 11:00 之前提交。

额外积分将根据每晚锦标赛的排名授予，具体如下：我们跟踪每晚锦标赛中排名前 20、10、3 和 1 的球队。我们将在比赛开始后的不同时间开始跟踪排名前 20、10、3 和 1 的球队：请参阅下面的“重要日期”部分。如果您的**团队在跟踪开始后**排名前 20 名，您将在期末考试中获得 1 个额外学分。如果您的**团队在跟踪开始后**排名前 10 名，您将获得 1 分。如果您的**团队在跟踪开始后**排名前 3，您将获得 1 分。最后，如果您的团队在最后的夜间锦标赛中获胜，您将获得 1 分的额外积分。

此外，卑鄙的工作人员带着自己狡猾的经纪人进入了比赛，寻求名声和荣耀。这些代理的团队名称以 Staff- 开头。除了夜间锦标赛的额外积分外，在夜间锦标赛中，您在 9 局两胜制**中击败的**每名工作人员代理，您将获得 0.5 的额外积分。

奖品摘要：

- 击败员工 - 代理正面交锋：每个代理在决赛中额外获得 0.5 个学分。
- 2010 年 11 月 11 日之后的前 20 名：决赛额外获得 1 个额外学分。
- 2010 年 11 月 18 日之后的前 10 名：额外 1 分。
- 2010 年 11 月 29 日之后的前 3 名：额外 1 分。
- 2010 年 12 月 1 日排名第一的球队：1 分。

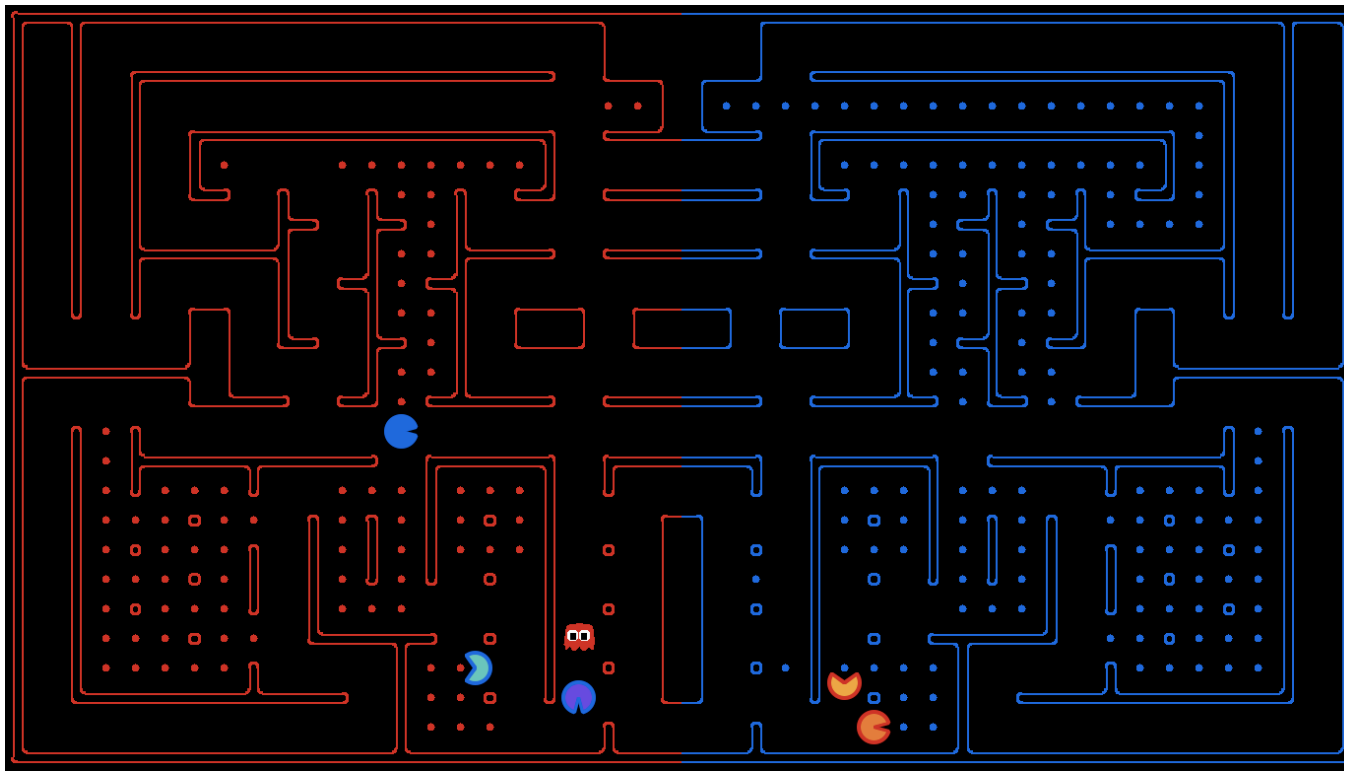
你越早提交你的经纪人，你获得高排名的机会就越大，你击败工作人员经纪人的机会就越大。

重要日期：

星期二	11/9/2010	比赛公布并发布
星期四	11/11/2010	开始跟踪前 20 名团队
星期四	11/18/2010	开始跟踪前 10 名团队
星期二	11/23/2010	最终比赛细节公布。
星期一	11/29/2010	开始跟踪前 3 名团队
星期三	12/1/2010	晚上 11:00 之前最终提交比赛，开始跟踪顶级团队。
星期四	12/2/2010	课堂上公布的成绩

确认

感谢 Barak Michener 和 Ed Karuna 提供改进的图形和调试帮助。



玩得愉快！如果您发现任何问题，请引起我们的注意。