



# *EE369 项目大作业：2048游戏*

杨健程 YANG Jiancheng

Oct, 2018



上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

# demo



## 2048 Game

Use ↑ ↓ ← → to play, any other keys for AUTO by your agent.

Score: 2

			2
			2

Controlled by USER

# How-To-1: Project Overview

duducheng / 2048-api

Unwatch 2 **★ Star 5** Fork 2

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

A 2048 api for training supervised learning (imitation learning) or reinforcement learning agents

Edit

## Code structure

- `game2048/` : the main package.
  - `game.py` : the core 2048 `Game` class.
  - `agents.py` : the `Agent` class with instances.
  - `displays.py` : the `Display` class with instances, to show the `Game` state.
  - `expectimax/` : a powerful ExpectiMax agent by [here](#).
- `explore.ipynb` : introduce how to use the `Agent` , `Display` and `Game` .
- `static/` : frontend assets (based on Vue.js) for web app.
- `webapp.py` : run the web app (backend) demo.
- `evaluate.py` : evaluate your self-defined agent.

<https://github.com/duducheng/2048-api>

# How-To-2: Coding



In [10]: `game.board`

Out[10]: `array([[ 8., 4., 8., 4.],  
[ 16., 128., 16., 2.],  
[ 8., 2., 32., 8.],  
[ 2., 4., 8., 16.]])`

In [11]: `Display().display(game)`

```
State:      8      4      8      4
          16     128     16      2
           8      2     32      8
           2      4      8     16
```

Score: 128

You lose! Score: 128

In [12]: `IPythonDisplay().display(game)`

**Score: 128**

8	4	8	4
16	128	16	2
8	2	32	8
2	4	8	16

You lose! Score: 128

```
print("Running the loop manually...")
```

```
game = Game(4, random=False, enable_rewrite_board=False)
agent = RandomAgent(game)
```

```
for _ in range(10):
    direction = agent.step()
    print("Moving to direction '%s'..."%direction)
    game.move(direction)
    display1.display(game)
    display2.display(game)
```

## To define your own agents

```
from game2048.agents import Agent

class YourOwnAgent(Agent):

    def step(self):
        '''To define the agent's 1-step behavior given the 'game'.
        You can find more instance in ['agents.py'](game2048/agents.py).

        :return direction: 0: left, 1: down, 2: right, 3: up
        ...
        direction = some_function(self.game)
        return direction
```

## To compile the pre-defined ExpectiMax agent

```
cd game2048/expectimax
bash configure
make
```

# How-To-3: Algorithm



- 利用项目代码中提供的ExpectiMaxAgent作为supervision
- 进行监督学习（模仿学习imitation learning）
- 项目要点：
  - 一个完全data-driven的项目（数据要多少有多少）
  - 基于学习的agent，完成训练后其预测时间复杂度是 $O(1)$
  - 如何表示棋盘（提示：one hot encoding）
  - 使用什么模型？（推荐神经网络）
  - 如何使用神经网络，如何设计激活函数和网络结构？
    - 是否需要使用CNN？
    - 是否需要像分类模型一样使用Global Pooling？
  - 如果使用强化学习，如何设计奖励（提示：每活一步就是一个reward）



# How-To-4: Evaluation Script



🔗 For EE369 students from SJTU only

## 给上海交通大学EE369课程的同学

为了方便我公平地评测作业,我雪藏了一些防伪脚本来验证你的代码唯一性.为此,在完成了你的agent之后,你需要做如下几件事情:

1. 获得你的测评分数,并保留程序运行log: 运行 `evaluate.py`

你需要修改该脚本中的相关部分,使其可以运行你自己的agent.

**注意1:** `display`: 你的agent实现必须参考我的Agent实现实现了 `display` 接口,这样你的运行过程的所有棋盘会被保留.该记录是用来防止作弊的一部分,必须实现,否则视为无效提交.

**注意2:** `N_TESTS`: 你不能修改这个测评次数,必须以10次成绩作为平均分数,否则以无效记.该信息将会体现在你的log文件中.

```
python evaluate.py >> EE369_evaluation.log
```

2. 利用我预先给出的棋盘( `board_cases.json` )给出你的agent指纹: 运行 `generate_fingerprint.py` .

同样,你需要修改该脚本中的相关部分,使其可以运行你自己的agent.

```
python generate_fingerprint.py
```

3. 你的代码需要以 Apache-2.0 协议(和该项目一致)开源在GitHub上.

<https://github.com/duducheng/2048-api/blob/master/EE369.md>

# Scoring



- 最终截止时间：18周结束，即2019年1月13日23:59。
- 百分制，最终会占30~40%的成绩，必须使用**基于学习的方法**
- 成绩组成
  - 代码运行平均成绩 $x$ （90分）
    - $if\ x \geq 1024: 90$
    - $elif\ x \geq 64: 15 \times (\log_2 x - 4)$  (30分~90分)
    - $else: 0$
  - 报告（10分）
    - 交大模板，正文10页左右PPT（方法和录制gif/视频），方法不够可添加附录
  - 加分（最多15分），有以下情况之一，进行5min口头报告和提问后可加满
    - 稳定达到2048分，并且在无限制条件下平均成绩在全班排名前3
    - 使用**基于强化学习**的方法，并稳定达到2048分以上
    - **额外**提供一份**基于planning**的方法超过1024分（单独提供不算分），需提供额外报告
- 严禁弄虚作假，包括虚假分数、抄袭代码等，最多直接扣100分。

# Other demo by your classmate



## 2048 Game

Use ↑ ↓ ← → to play, any other keys for AUTO by your agent.

Score: 4

	2		
	4		



Only ~7 days! Bravo!  
Plan your time!

Controlled by USER



# Resource



- 一个在线可玩的2048游戏: <https://gabrielecirulli.github.io/2048/>
- 强化学习: 台大李宏毅的强化学习章节
  - <https://www.csie.ntu.edu.tw/~yvchen/f106-adl/syllabus> (或google个人主页)
- 一些相关的库:
  - 科学计算环境: Anaconda 3.6 (要求使用Python 3, 推荐使用3.6)
  - 科学计算: numpy (矩阵并行), scipy (计算工具包), matplotlib (画图)
  - 机器学习 (SVM, 决策树等): scikit-learn
  - 深度学习 (神经网络): **PyTorch**, **Keras**, TensorFlow (其他不推荐入门使用)
- 一些可用的云计算资源 (需翻墙):
  - 推荐AWS educate: 交大学生可获得\$100, 单卡GPU\$0.9 / 小时
    - <http://cs231n.github.io/aws-tutorial/>
    - 使用AWS Deep Learning AMI, 已经配置好所有环境
  - Google Cloud: 注册送\$200

# Thank You

