

# birdbot

Train your own flappy bird robot easily

## 介绍

此项目提供了一个简易的框架来训练flappy bird机器人，可以很方便在此框架上学习或测试agent的一些算法。

## 特点

- 对游戏进行了简单的封装，可以很方便得到游戏的状态来辅助算法实现。
- 可以显示游戏界面方便调试，能够看到算法实现的效果。也可以选择关闭游戏界面以及声音，这样游戏仍然能正常运行，一般用于训练阶段，可以减少CPU的占用。

## 开始训练自己的机器人

### 准备

确认安装了python2.7.x和pyglet。pyglet的安装Window可从[这里](#)直接下载二进制exe文件安装。Linux可从各个发行版的软件源安装，例如ubuntu:

```
sudo apt-get install python-pyglet
```

若软件源中找不到，可以按[这里](#)下载源代码编译安装

### 实现智能算法

算法代码写在Bot.plan()方法里，Bot.plan()方法每0.05秒会被调用一次，可以根据每次传入的state信息来辅助此次的决策。

state是一个dict，存放的是当前的游戏状态（也就是上一次执行动作后的状态）。具体信息如下：

```
state = { 'bird': (x, y, bird_state) ,  
          'pipes': [(x0, y0), (x1, y1), ... ] }
```

其中bird是鸟相关信息，x, y是鸟的横纵坐标，bird\_state为string，若值为'alive'表示鸟活着，'dead'表示鸟死了。

pipes是各个管道的位置，上管道的位置用左下角的点表示，下管道的位置用左上角的点表示，管道存放的顺序按“上下上下上下”，**见最后面的图**。红点表示管道位置，数字表示顺序。**另外注意坐标系是从左到右，从下到上，即原点在左下角**

由于游戏比较简单，可以执行的动作只有点击屏幕。可以通过调用self.tap()指令来点击屏幕。

### 数据的保存及其他

需要在整个程序执行期间更新的数据可以放在Bot.\_init\_()方法里。训练得到的数据可以通过\_init\_()方法里的do\_at\_exit()保存，这样训练时间过长或程序出错时可以把数据存下来，避免从头开始训练。

是否显示窗口和播放声音可以在main的两个标志show\_window和enable\_sound设置。

## 例子程序

代码里有一个例子sample\_bot.py，实现了Q-learning算法，参考的是这篇文章[Flappy Bird hack using Reinforcement Learning](#)。保存数据等也可以参考例子。可以考虑结合一些规则（例如低于下根管道时尽量飞高，高于时则鸟降低高度等）来加快训练。

## 参考资料

wikipedia的[强化学习介绍](#)

《AI: a modern approach》的第21章