Assignment 1: Programming Task

Svetlin Penkov
School of Informatics
Feb 14, 2017

Setup

- Repository: https://github.com/ipab-rad/rl-cw1
- The README file provides lots of information
- On a DICE machine:

```
> git clone https://github.com/ipab-rad/rl-cw1
> cd rl-cw1
> python keyboard_agent.py
```

Setup on personal machine

- Install OpenCV with graphical user interface support.
  - This usually involves installing a windowing system such as GTK+, including the dev files for it, and compiling OpenCV from source.

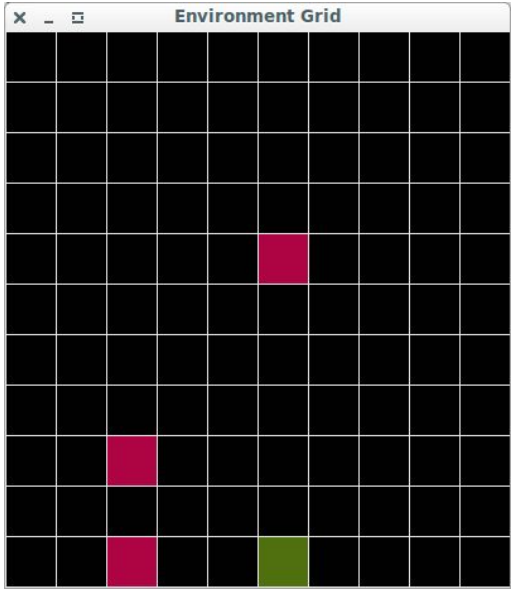
- Install ALE by following the instructions
  - https://github.com/mgbellemare/Arcade-Learning-Environment

SSH into DICE from personal machine

```
# Remember to change to your student number
> ssh -X sNNNNNNN@student.ssh.inf.ed.ac.uk
# Enter Dice password and SSH again
> ssh -X student.login

> git clone https://github.com/ipab-rad/rl-cw1
> cd rl-cw1/
> python keyboard_agent.py
```

Running the Keyboard Agent



The Enduro game

Game dynamics

Actions:

Accelerate, Brake, Left, Right

Reward:

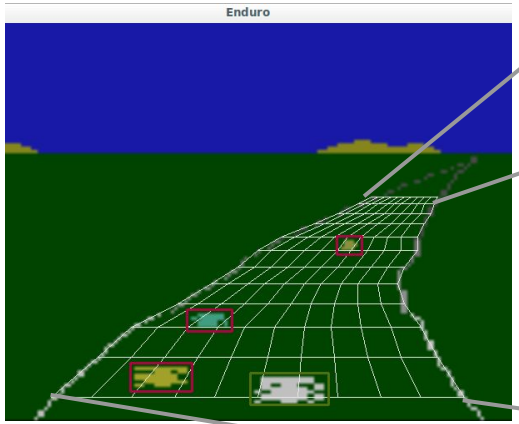
+1 if agent passes by a car

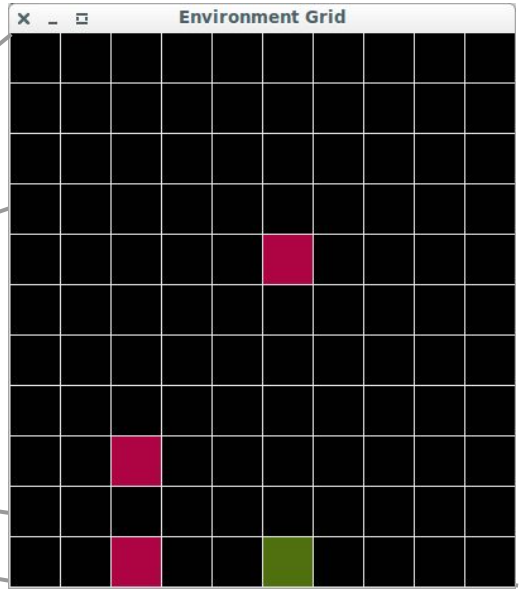
-1 if opponent passes by the agent

Collisions:

When a collision occurs the agent's velocity is decreased which may lead to negative reward.

Game discretisation

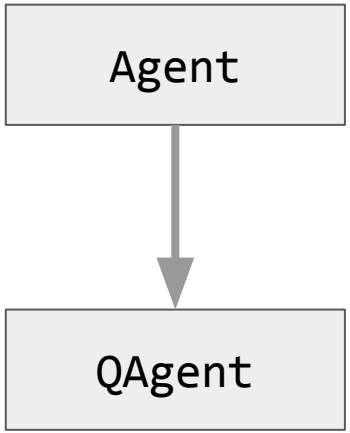




Col 0 Col 9

Row 10 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 2 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 2 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
Row 0 0 0 2 0 0 1 0 0 0 0

Class Hierarchy



- Any agent that you write should be derived from the `Agent` class
- The `Agent` class implements key functions and provides an interface for the implementation of specific agents
- Specific agents that you will use are the `KeyboardAgent`, `RandomAgent`, and `QAgent`.

The `Agent` class

```
def run(self, learn, episodes)-> None

def getActionsSet(self)-> [Action.ACCELERATE,
                           Action.BREAK,
                           Action.RIGHT,
                           Action.LEFT]

def move(self, action)-> reward
```

The `Agent` subclasses

```
def initialise(self, grid)

def act(self)

def sense(self, grid)

def learn(self)

def callback(self, learn, episode, iteration)
```

Let’s explore the source code...