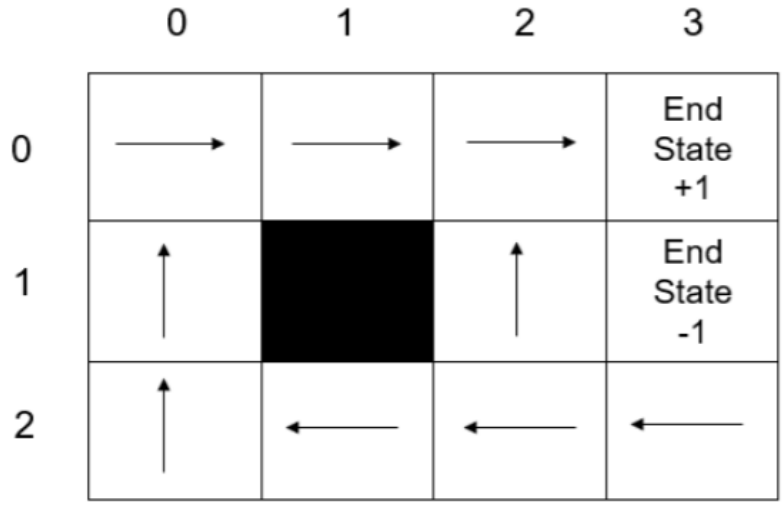
Assignment 2 Part B

Name: David Fodor Student code: B00796884 Email: fodor-d@ulster.ac.uk

Reinforcement Learning in GridWorld



Policy Evaluation

The above diagram specifies a policy in GridWorld, i.e. directions for what action to perform in each state. In this submission, the Temporal Difference Learning approach is going to be used instead of the Monte Carlo approach used in the 'Week_11_RL_GridWorld.ipynb' file from Blackboard.

This code is a modified version of code available at [https://github.com/MJeremy2017/Reinforcement-Learning-Implementation (https://github.com/MJeremy2017/Reinforcement-Learning-Implementation)](https://github.com/MJeremy2017/Reinforcement-Learning-Implementation) The MIT License for the original code has been include with this version on Blackboard.

In [450]:

```
import numpy as np
```

In [451]:

```
BOARD_ROWS = 3
BOARD_COLS = 4
WIN_STATE = (0, 3)
LOSE_STATE = (1, 3)
START = (2, 3)
DETERMINISTIC = False
```

Defining GridWorld

The State class below describes the GridWorld environment, including the transition probabilities (see the functions `nxtPosition` and `_chooseActionProb`) and the rewards in each state (see the function `giveReward`). Note that the environment is not deterministic. If the agent chooses direction "up" there is a 10% chance that actual direction will be "left" and 10% that it will be "right". The same goes for the other directions.

In [452]:

```
class State:
    def __init__(self, state=START):
        self.board = np.zeros([BOARD_ROWS, BOARD_COLS])
        self.board[1, 1] = -1
        self.state = state
        self.isEnd = False
        self.determine = DETERMINISTIC

    def giveReward(self):
        if self.state == WIN_STATE:
            return 1
        elif self.state == LOSE_STATE:
            return -1
        else:
            return -0.04

    def isEndFunc(self):
        if (self.state == WIN_STATE) or (self.state == LOSE_STATE):
            self.isEnd = True

    def _chooseActionProb(self, action):
        if action == "up":
            return np.random.choice(["up", "left", "right"], p=[0.8, 0.1, 0.1])
        if action == "down":
            return np.random.choice(["down", "left", "right"], p=[0.8, 0.1, 0.1])
        if action == "left":
            return np.random.choice(["left", "up", "down"], p=[0.8, 0.1, 0.1])
        if action == "right":
            return np.random.choice(["right", "up", "down"], p=[0.8, 0.1, 0.1])

    def nxtPosition(self, action):
        """
        action: up, down, left, right
        -------------
        0 | 1 | 2| 3|
        1 |
        2 |
        return next position on board
        """
        if self.determine:
            if action == "up":
                nxtState = (self.state[0]-1, self.state[1])
            elif action == "down":
                nxtState = (self.state[0]+1, self.state[1])
            elif action == "left":
                nxtState = (self.state[0], self.state[1]-1)
            else:
                nxtState = (self.state[0], self.state[1]+1)
            self.determine = False
        else:
            # non-deterministic
            action = self._chooseActionProb(action)
            self.determine = True
            nxtState = self.nxtPosition(action)

        # if next state is legal
        if (nxtState[0] >= 0) and (nxtState[0] <= 2):
            if (nxtState[1] >= 0) and (nxtState[1] <= 3):
                if nxtState != (1, 1):
```

```
                return nxtState
        return self.state

    def showBoard(self):
        self.board[self.state] = 1
        for i in range(0, BOARD_ROWS):
            print('-----------------')
            out = '| '
            for j in range(0, BOARD_COLS):
                if self.board[i, j] == 1:
                    token = '*'
                if self.board[i, j] == -1:
                    token = 'z'
                if self.board[i, j] == 0:
                    token = '0'
                out += token + ' | '
            print(out)
        print('-----------------')
```

Defining the Agent

The Agent class below defines the policy adopted by the agent (see the function `chooseAction`) as well as the Temporal Difference Learning approach used for policy evaluation in the function `play`. A 'round' corresponds to one episode in GridWorld, i.e. from the start state to one of the end states. Many rounds need to be played to evaluate the policy. Note also that a learning rate has been defined `self.lr`. This is the α value used in learning. Recall that the Temporal Difference Learning approach can be expressed as:

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma * V(S_{t+1}) - V(S_t)]$$

where $V(S_t)$ is the estimate of the value function v_π in state S_t, $V(S_{t+1})$ is the estimate of the value function v_π in state S_{t+1} and R_{t+1} is the reward at the next step.

In [453]:

```
class Agent:

    def __init__(self):
        self.states = []  # record position and action taken at the position
        self.actions = ["up", "down", "left", "right"]
        self.State = State()
        self.isEnd = self.State.isEnd
        self.lr = 0.1
        self.exp_rate = 0.3
        self.decay_gamma = 1

        # initial state values
        self.state_values = {}
        for i in range(BOARD_ROWS):
            for j in range(BOARD_COLS):
                self.state_values[(i, j)] = np.random.random()

        # state values are set to zero for the end states and (1,1) since the
        # agent can never go there
        self.state_values[WIN_STATE] = 0
        self.state_values[LOSE_STATE] = 0
        self.state_values[(1,1)] = 0

        # initial state counts
        self.state_counts = {}
        for i in range(BOARD_ROWS):
            for j in range(BOARD_COLS):
                self.state_counts[(i, j)] = 0

        self.state_counts[self.State.state] = 1

    def chooseAction(self):
        # Choose action based on specified policy
        if self.State.state == (2,0) or self.State.state == (1,0) or self.State.state ==
= (1,2):
            action = "up"
        elif self.State.state == (0,0) or self.State.state == (0,1) or self.State.state
== (0,2):
            action = "right"
        else:
            action = "left"

        return action

    def takeAction(self, action):
        position = self.State.nxtPosition(action)
        # update State
        return State(state=position)

    def reset(self):
        self.states = []
        self.State = State()
        self.state_counts[self.State.state] = self.state_counts[self.State.state] + 1
        self.isEnd = self.State.isEnd

    def play(self, rounds=10):
        i = 0
        while i < rounds:
            if self.State.isEnd:
```

```
                # If the agent has reached an end state then apply the Monte Carlo appr
oach
                # The commented out code below is a better way to do it, but the code b
elow
                # that might be more helpful for the exercise.
                '''
                gt = self.State.giveReward()
                for s in reversed(self.states):
                    self.state_values[s.state] = self.state_values[s.state] + self.lr*
(gt - self.state_values[s.state])
                    gt = gt + s.giveReward()
                '''
                # Implementation of Monte Carlo approach
                # We need to calculate Gt for each step. The first step requires adding
all the rewards after
                # the first action. So for each action we need the reward at the next s
tep.
                gt = 0
                for j in range(len(self.states)): # self.states includes all the states
in this round
                    if j == len(self.states)-1:
                        snext = self.State        # self.State is the final state
                    else:
                        snext = self.states[j+1]
#                   gt = gt + snext.giveReward()  # add all the rewards

                    #The five lines below implement the Temporal Difference Learning fo
rmula
                    #We update our estimates of the value function inside the loop in t
his case, instead of after it
                    #Since this formula replaces the Monte Carlo approach, the lines ti
ed to the latter are commented out
                    RtPlus1 = snext.giveReward()
                    VStPlus1 = self.state_values[snext.state]
                    s = self.states[j]
                    #The line below is the key line for implementing the Temporal Diffe
rence Learning formula
                    self.state_values[s.state] = self.state_values[s.state] + self.lr*(
RtPlus1 + self.decay_gamma * VStPlus1 - self.state_values[s.state])

#                # Now use the rewards to update our estimates of the value function
#                 for s in self.states:
#                     self.state_values[s.state] = self.state_values[s.state] + self.lr*
(gt - self.state_values[s.state])
#                     gt = gt - s.giveReward()      # the first reward will not apply on
the next step so remove it

                # This just lets us see what the final reward is for this state -1 or +
1
                reward = self.State.giveReward()
                print("Game End Reward", reward)

                self.reset()
                i += 1
            else:
                action = self.chooseAction()
                # append trace
                self.states.append(self.State)
                print("current position {} action {}".format(self.State.state, action))
                # by taking the action, it reaches the next state
                self.State = self.takeAction(action)
```

```
                # mark is end
                self.State.isEndFunc()
                print("nxt state", self.State.state)
                print("----------------------")
                self.isEnd = self.State.isEnd
                # Increment count
                self.state_counts[self.State.state] = self.state_counts[self.State.stat
e] + 1
```

Create Agent

Display initial value function estimates (set randomly).

In [454]:

```
ag = Agent()
ag.state_values
```

Out[454]:

```
{(0, 0): 0.5529312683262151,
 (0, 1): 0.44607110449534815,
 (0, 2): 0.7427619374922747,
 (0, 3): 0,
 (1, 0): 0.4842800432156288,
 (1, 1): 0,
 (1, 2): 0.2398790178938207,
 (1, 3): 0,
 (2, 0): 0.44268128298990783,
 (2, 1): 0.14873532941969136,
 (2, 2): 0.20507839383578763,
 (2, 3): 0.9003486567037694}
```

Start Learning

Display the sequence of states and actions.

In [455]:

```
ag.play(37)
```

```
current position (2, 3) action left
nxt state (1, 3)
---------------------
Game End Reward -1
current position (2, 3) action left
nxt state (1, 3)
---------------------
Game End Reward -1
current position (2, 3) action left
nxt state (2, 3)
---------------------
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
```

```
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (1, 3)
---------------------
Game End Reward -1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
```

```
nxt state (2, 3)
---------------------
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (1, 2)
---------------------
current position (1, 2) action up
nxt state (0, 2)
---------------------
current position (0, 2) action right
```

```
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 3)
---------------------
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
```

```
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 3)
---------------------
current position (2, 3) action left
nxt state (2, 3)
---------------------
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
```

```
--------------------
current position (1, 0) action up
nxt state (1, 0)
--------------------
current position (1, 0) action up
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (0, 3)
--------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
--------------------
current position (2, 2) action left
nxt state (2, 1)
--------------------
current position (2, 1) action left
nxt state (2, 0)
--------------------
current position (2, 0) action up
nxt state (1, 0)
--------------------
current position (1, 0) action up
nxt state (1, 0)
--------------------
current position (1, 0) action up
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (0, 3)
--------------------
Game End Reward 1
current position (2, 3) action left
nxt state (1, 3)
--------------------
Game End Reward -1
current position (2, 3) action left
nxt state (2, 2)
--------------------
current position (2, 2) action left
nxt state (2, 1)
--------------------
current position (2, 1) action left
nxt state (2, 0)
--------------------
```

```
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (1, 3)
---------------------
Game End Reward -1
current position (2, 3) action left
nxt state (2, 3)
---------------------
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (1, 2)
---------------------
current position (1, 2) action up
```

```
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
```

```
--------------------
current position (2, 2) action left
nxt state (2, 1)
--------------------
current position (2, 1) action left
nxt state (2, 0)
--------------------
current position (2, 0) action up
nxt state (1, 0)
--------------------
current position (1, 0) action up
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (0, 3)
--------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
--------------------
current position (2, 2) action left
nxt state (2, 1)
--------------------
current position (2, 1) action left
nxt state (2, 1)
--------------------
current position (2, 1) action left
nxt state (2, 0)
--------------------
current position (2, 0) action up
nxt state (1, 0)
--------------------
current position (1, 0) action up
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (0, 3)
--------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
--------------------
current position (2, 2) action left
```

```
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
```

```
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (1, 3)
---------------------
Game End Reward -1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
```

```
--------------------
current position (0, 0) action right
nxt state (1, 0)
--------------------
current position (1, 0) action up
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (0, 3)
--------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
--------------------
current position (2, 2) action left
nxt state (2, 2)
--------------------
current position (2, 2) action left
nxt state (2, 1)
--------------------
current position (2, 1) action left
nxt state (2, 1)
--------------------
current position (2, 1) action left
nxt state (2, 0)
--------------------
current position (2, 0) action up
nxt state (1, 0)
--------------------
current position (1, 0) action up
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (1, 2)
--------------------
current position (1, 2) action up
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (0, 3)
--------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
--------------------
current position (2, 2) action left
```

```
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (1, 2)
---------------------
current position (1, 2) action up
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
```

```
current position (0, 0) action right
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (0, 1)
---------------------
current position (0, 1) action right
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (1, 2)
---------------------
current position (1, 2) action up
nxt state (0, 2)
---------------------
current position (0, 2) action right
nxt state (0, 3)
---------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
---------------------
current position (2, 2) action left
nxt state (2, 1)
---------------------
current position (2, 1) action left
nxt state (2, 0)
---------------------
current position (2, 0) action up
nxt state (1, 0)
---------------------
current position (1, 0) action up
nxt state (0, 0)
---------------------
current position (0, 0) action right
nxt state (1, 0)
```

```
--------------------
current position (1, 0) action up
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (0, 3)
--------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
--------------------
current position (2, 2) action left
nxt state (2, 1)
--------------------
current position (2, 1) action left
nxt state (2, 0)
--------------------
current position (2, 0) action up
nxt state (1, 0)
--------------------
current position (1, 0) action up
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (0, 3)
--------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 2)
--------------------
current position (2, 2) action left
nxt state (2, 1)
--------------------
current position (2, 1) action left
nxt state (2, 0)
--------------------
current position (2, 0) action up
nxt state (1, 0)
--------------------
current position (1, 0) action up
nxt state (0, 0)
--------------------
current position (0, 0) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
nxt state (0, 1)
--------------------
current position (0, 1) action right
```

```
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (0, 3)
--------------------
Game End Reward 1
current position (2, 3) action left
nxt state (2, 3)
--------------------
current position (2, 3) action left
nxt state (2, 2)
--------------------
current position (2, 2) action left
nxt state (1, 2)
--------------------
current position (1, 2) action up
nxt state (0, 2)
--------------------
current position (0, 2) action right
nxt state (0, 3)
--------------------
Game End Reward 1
```

Display Estimated Value Function

In [456]:

```
ag.state_values
```

Out[456]:

```
{(0, 0): 0.6915539243090698,
 (0, 1): 0.8338118353783813,
 (0, 2): 0.926232108835523,
 (0, 3): 0,
 (1, 0): 0.5790349159992736,
 (1, 1): 0,
 (1, 2): 0.5230695567556738,
 (1, 3): 0,
 (2, 0): 0.4721785000127598,
 (2, 1): 0.3858862712455421,
 (2, 2): 0.321888758604263,
 (2, 3): 0.16886866612774665}
```

Conclusion

I have concluded that from 37 plays and onward, the algorithm produces a consistently correct result regarding the order of the 3 most valuable cells (0, 2), (0, 1), (0, 0), which satisfies the defined policy. Anything below 37 plays has an increasing risk of finding an erratic descending order of value estimates for the 3 most valuable cells. (The estimated values of the rest of the cells can still be wrong at 37 plays. More plays result in more accurate estmates for all cells on the grid.)

Alternative policy

A better policy would be to change cell (2, 2) into having an arrow pointing up instead of left. This would result in a shorter route. I was unsuccessful in trying to implement this.

Reinforcement learning from an employability perspective

Nowadays the advancement in AI is apparent to everyone, for it surrounds us in our daily lives. Ranging from voice assistants to the YouTube algorithm, AI has many practical uses and it is becoming ever more widespread. Of course this growing frequency of practical uses inevitably causes higher demand for AI professionals in the job market (Culbertson, 2018), especially people with machine learning skills, to be specific. Some noticeable rising job trends are related to deep learning and natural language processing (Terra, 2020), but every industry is bound to undergo machine learning reformation sooner or later. In contrast with the rise in demand for machine learning experts, the supply seems to be levelling off, causing competition between employers and promising top salary for future employees (Zafarino, 2018). For the reason that the number of machine learning candidates is finite, we can notice tech giants like Google trying to incentivise people to consider this field by making the knowledge widely available. Of course being a machine learning expert is not an easy task, it requires a lot of dedication, time, effort, and lastly, affinity. Regardless, should one choose this career option, it looks like the knowledge they have to learn for it will not be outdated anytime soon.

References

Daniel Culbertson. (2018). Demand for AI Talent on the Rise. [online] Available at: https://www.hiringlab.org/2018/03/01/demand-ai-talent-rise (https://www.hiringlab.org/2018/03/01/demand-ai-talent-rise) [Accessed 11 Dec. 2020]. John Terra. (2020). The Rise of Artificial Intelligence and Machine Learning Job Trends in 2021. [online] Available at: https://www.simplilearn.com/rise-of-ai-and-machine-learning-job-trends-article (https://www.simplilearn.com/rise-of-ai-and-machine-learning-job-trends-article) [Accessed 08 Dec. 2020]. Stephen Zafarino. (2018). The outlook for machine learning in tech: ML and AI skills in high demand. [online] Available at: https://www.cio.com/article/3293019/the-outlook-for-machine-learning-in-tech-ml-and-ai-skills-in-high-demand.html (https://www.cio.com/article/3293019/the-outlook-for-machine-learning-in-tech-ml-and-ai-skills-in-high-demand.html) [Accessed 11 Dec. 2020].