# Minerunner

組員: 許瑋哲 、林揚森、林穎沛、陳宥翔

## Introduction

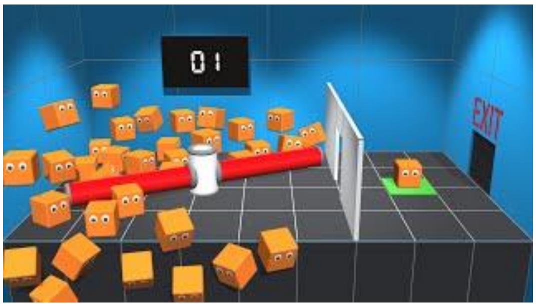
- 3D-world of Minecraft to simulate  real world.
- AI exploring the specific map in Minecraft.
- Use QL & DQN & CNN to train

# Introduction-Why this problem is important?

1. Generalize exploration system .
2. Pushing it to the real world .
3. Manually efficient.

# Related work

- Malmo
- MineDojo
- MineRL
- AI learns to escape



Malmo

# Platform

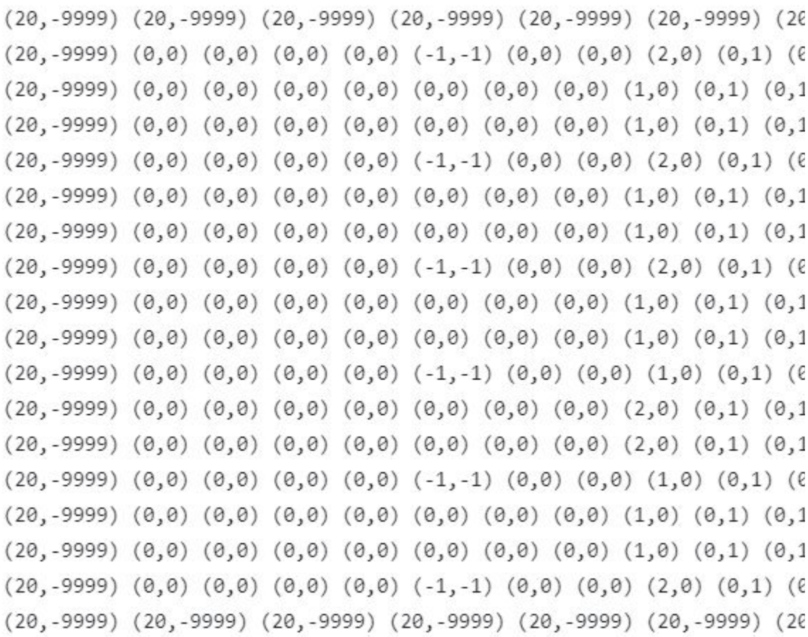
- Platform
  - Minecraft
  - Malmo

# Dataset

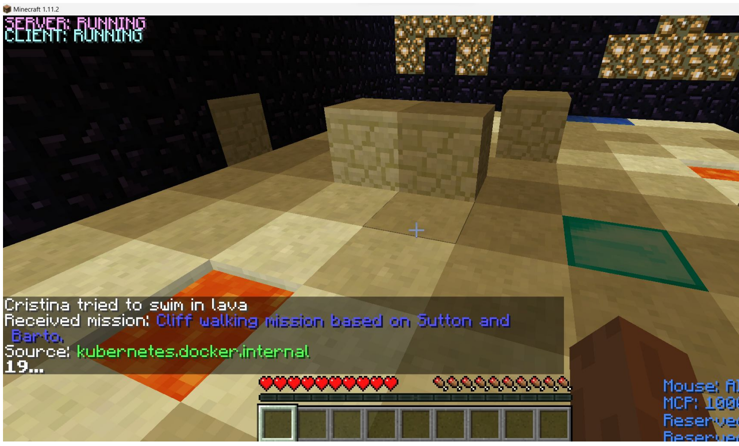
- Store the information about maps in a metrix
    - We take 9 blocks around us as observation.
    - Each block has 2 feature: (h_d, block_type)
- h_d: height difference with block and initial spawn point(height = 0)
- block_type: Serial number of each block

| obsidian | sandStone | diamond | lapis_block |
|---|---|---|---|
| -1 | 0 | 0 | -1 |

# Matrix for maps



```
(20,-9999) (20,-9999) (20,-9999) (20,-9999) (20,-9999) (20,-9999) (20
(20,-9999) (0,0) (0,0) (0,0) (0,0) (-1,-1) (0,0) (0,0) (2,0) (0,1) (0
(20,-9999) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (1,0) (0,1) (0,1
(20,-9999) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (1,0) (0,1) (0,1
(20,-9999) (0,0) (0,0) (0,0) (0,0) (-1,-1) (0,0) (0,0) (2,0) (0,1) (0
(20,-9999) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (1,0) (0,1) (0,1
(20,-9999) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (1,0) (0,1) (0,1
(20,-9999) (0,0) (0,0) (0,0) (0,0) (-1,-1) (0,0) (0,0) (2,0) (0,1) (0
(20,-9999) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (1,0) (0,1) (0,1
(20,-9999) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (1,0) (0,1) (0,1
(20,-9999) (0,0) (0,0) (0,0) (0,0) (-1,-1) (0,0) (0,0) (1,0) (0,1) (0
(20,-9999) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (2,0) (0,1) (0,1
(20,-9999) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (2,0) (0,1) (0,1
(20,-9999) (0,0) (0,0) (0,0) (0,0) (-1,-1) (0,0) (0,0) (1,0) (0,1) (0
(20,-9999) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (1,0) (0,1) (0,1
(20,-9999) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (0,0) (1,0) (0,1) (0,1
(20,-9999) (0,0) (0,0) (0,0) (0,0) (-1,-1) (0,0) (0,0) (2,0) (0,1) (0
(20,-9999) (20,-9999) (20,-9999) (20,-9999) (20,-9999) (20,-9999) (20
```

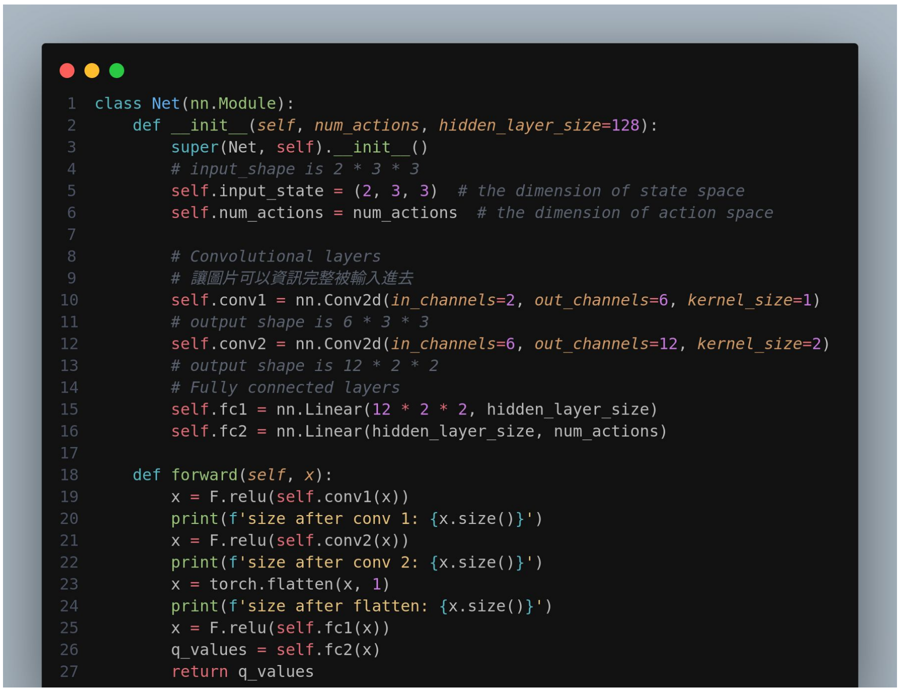


Cristina tried to swim in lava
Received mission: Cliff walking mission based on Sutton and Barto.
Source: kubernetes.docker.internal
19...

# Baseline - Q learning with CNN

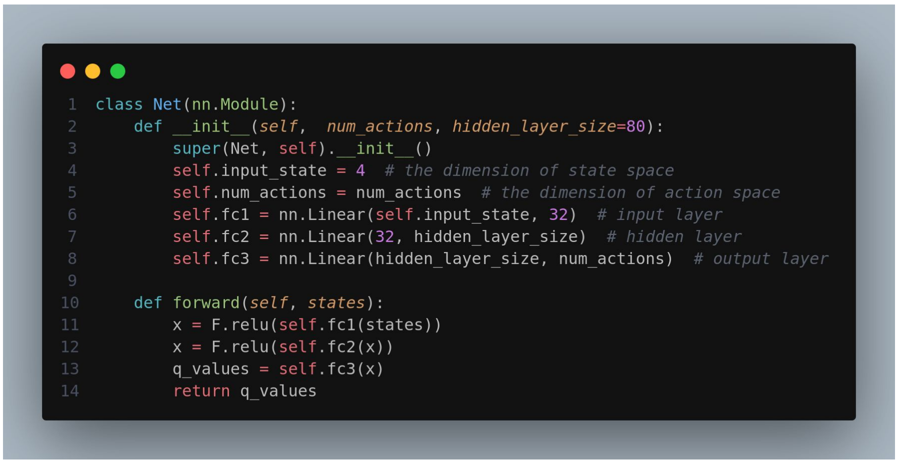
- Input state
- Convolutional layer
  - conv1
- Fully connected layer
  - fc1
  - fc2
- Output q value

# Baseline - Q learning with CNN



```python
class Net(nn.Module):
    def __init__(self, num_actions, hidden_layer_size=128):
        super(Net, self).__init__()
        # input_shape is 2 * 3 * 3
        self.input_state = (2, 3, 3)  # the dimension of state space
        self.num_actions = num_actions  # the dimension of action space

        # Convolutional layers
        # 讓圖片可以資訊完整被輸入進去
        self.conv1 = nn.Conv2d(in_channels=2, out_channels=6, kernel_size=1)
        # output shape is 6 * 3 * 3
        self.conv2 = nn.Conv2d(in_channels=6, out_channels=12, kernel_size=2)
        # output shape is 12 * 2 * 2
        # Fully connected layers
        self.fc1 = nn.Linear(12 * 2 * 2, hidden_layer_size)
        self.fc2 = nn.Linear(hidden_layer_size, num_actions)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        print(f'size after conv 1: {x.size()}')
        x = F.relu(self.conv2(x))
        print(f'size after conv 2: {x.size()}')
        x = torch.flatten(x, 1)
        print(f'size after flatten: {x.size()}')
        x = F.relu(self.fc1(x))
        q_values = self.fc2(x)
        return q_values
```

# Baseline - DQN



```
class Net(nn.Module):
    def __init__(self,  num_actions, hidden_layer_size=80):
        super(Net, self).__init__()
        self.input_state = 4  # the dimension of state space
        self.num_actions = num_actions  # the dimension of action space
        self.fc1 = nn.Linear(self.input_state, 32)  # input layer
        self.fc2 = nn.Linear(32, hidden_layer_size)  # hidden layer
        self.fc3 = nn.Linear(hidden_layer_size, num_actions)  # output layer

    def forward(self, states):
        x = F.relu(self.fc1(states))
        x = F.relu(self.fc2(x))
        q_values = self.fc3(x)
        return q_values
```

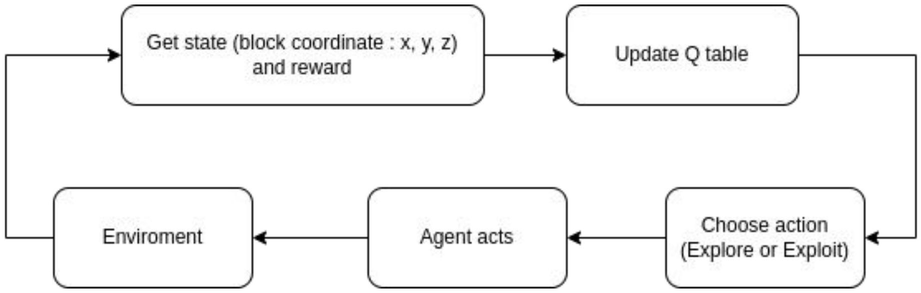
- Input state
- Full connected layer
  - fc1
  - fc2
  - fc3
- Output q value

# Main Approach

- Q-learning
  - Q-table
  - DQN
  - CNN

# Main Approach-Q-Learning

- Q-table

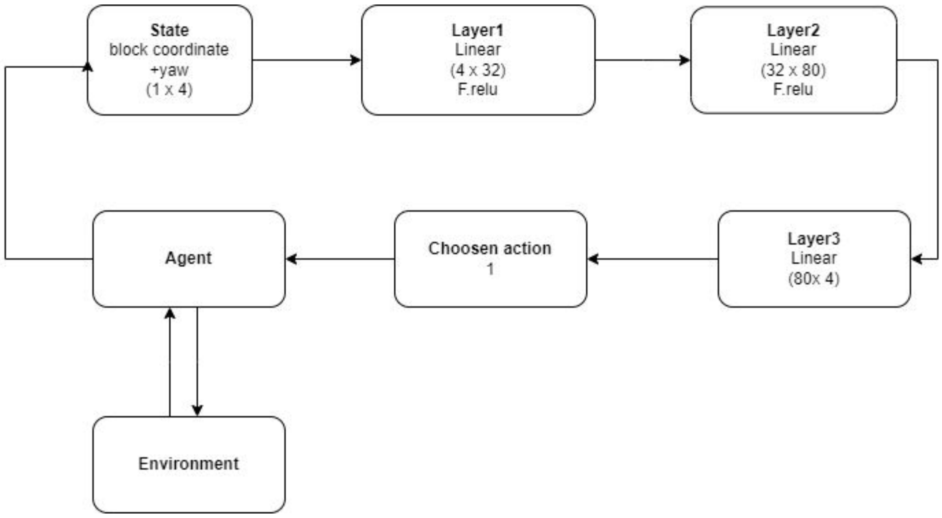

Get state (block coordinate : x, y, z) and reward
Update Q table
Choose action (Explore or Exploit)
Agent acts
Enviroment

The formula for updating the Q table:

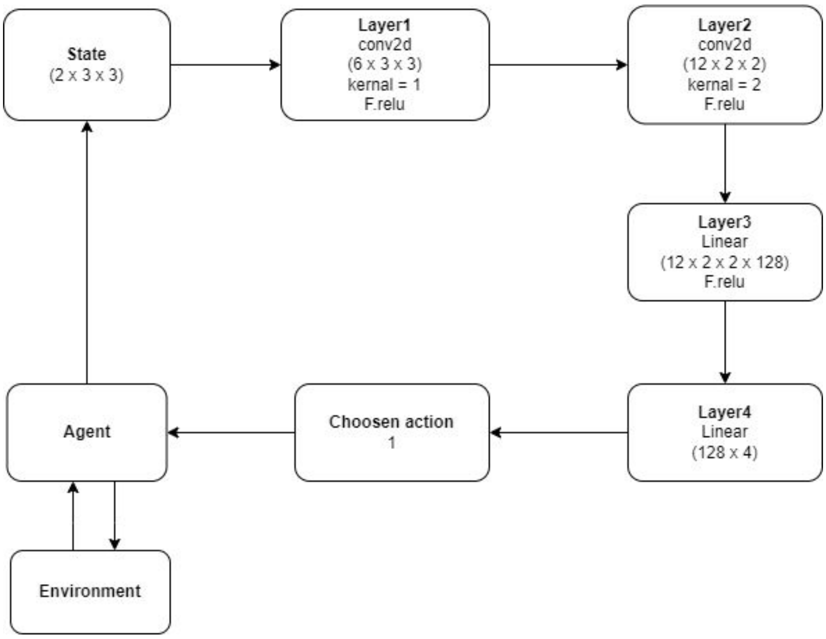
new_q = old_q + learning_rate * ( reward - old_q )

# Main Approach-DQN

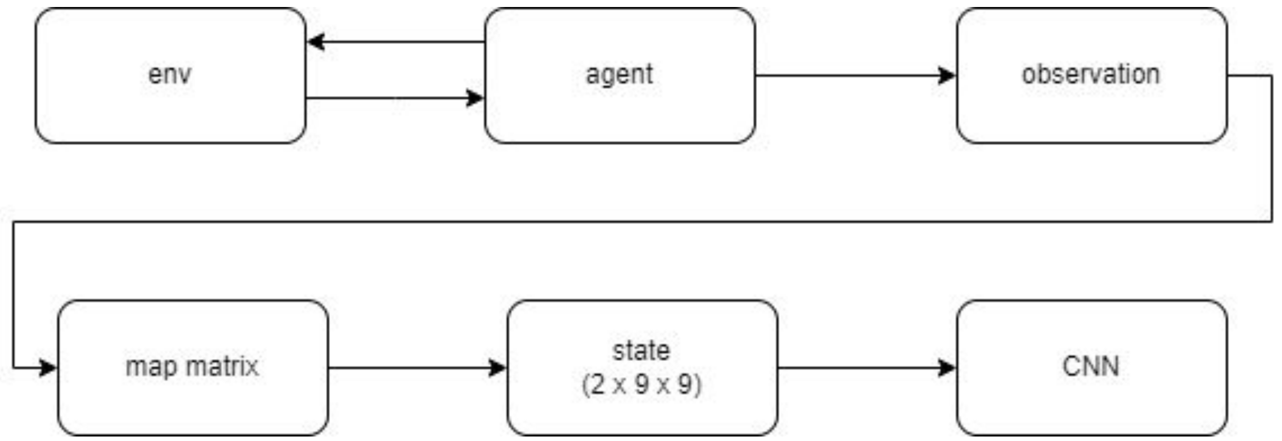
- DQN



# Main Approach-CNN

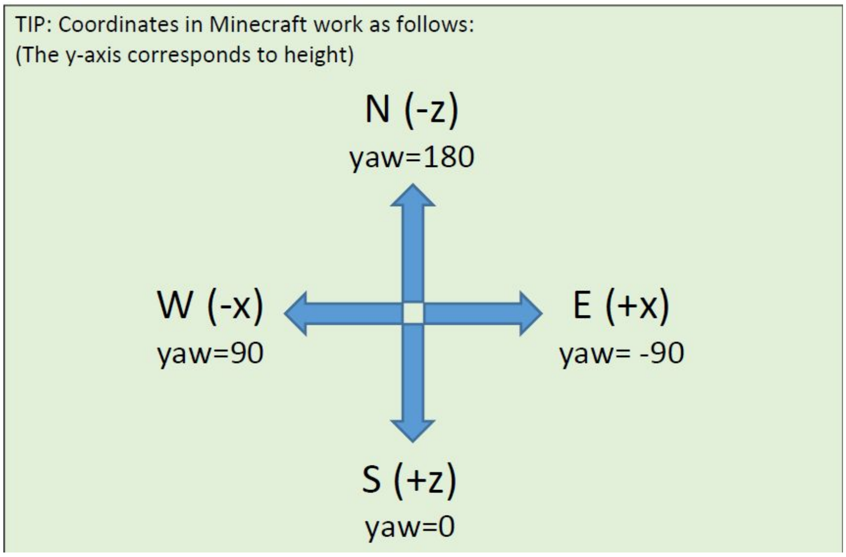


# Main Approach-How to get state



# Main Approach-How to get state

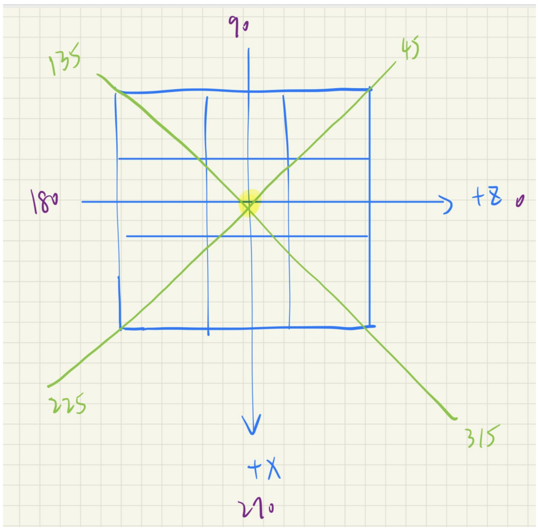
- Coordination system in malmo



TIP: Coordinates in Minecraft work as follows:
(The y-axis corresponds to height)

# Main Approach-How to get state

- state transform from map matrix



# Main Approach-How to get state

- Block order change based on different yaw

map matrix

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

45<=yaw<=135

| | | |
|---|---|---|
| 7 | 8 | 9 |
| 4 | 5 | 6 |
| 1 | 2 | 3 |

135<=yaw<=225

| | | |
|---|---|---|
| 9 | 6 | 3 |
| 8 | 5 | 2 |
| 7 | 4 | 1 |

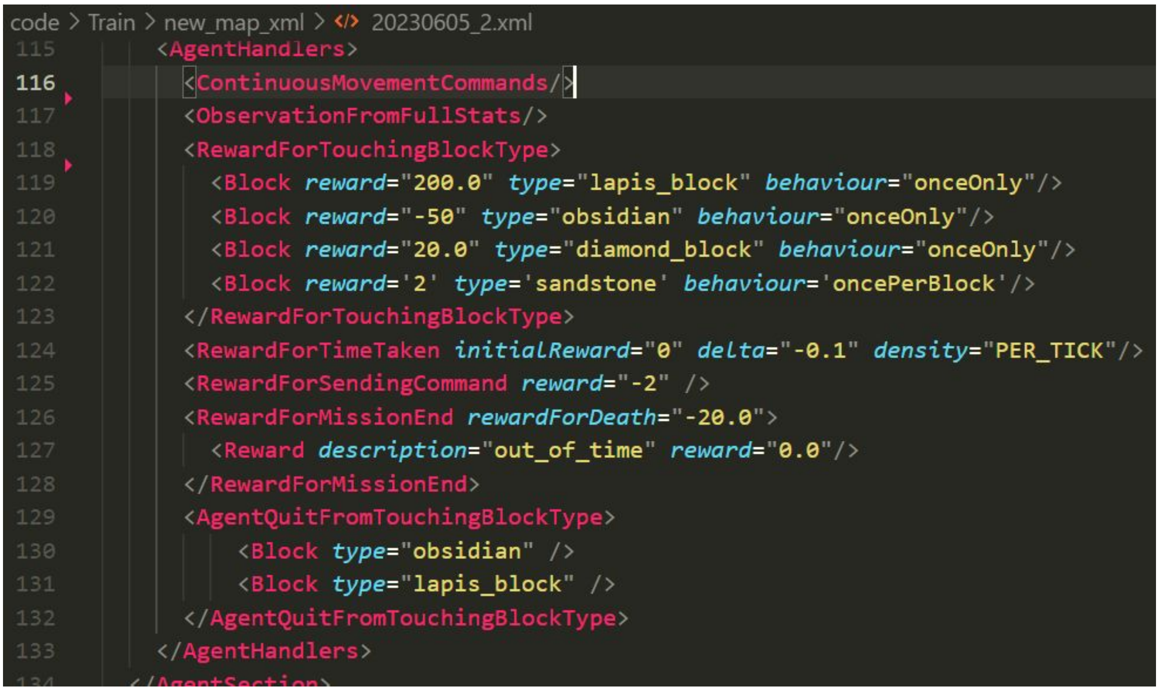
225<=yaw<=315

| | | |
|---|---|---|
| 3 | 2 | 1 |
| 6 | 5 | 4 |
| 3 | 2 | 1 |

0<=yaw<=45
or 315<=yaw<=360(0)

| | | |
|---|---|---|
| 3 | 6 | 9 |
| 2 | 5 | 8 |
| 1 | 4 | 7 |

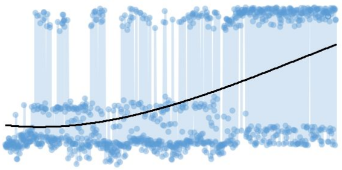
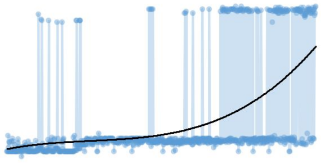
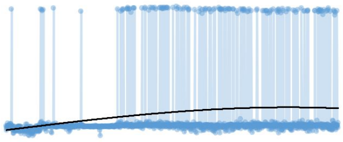
# Main Approach-XML file



code > Train > new_map_xml > </> 20230605_2.xml

```xml
  <AgentHandlers>
    <ContinuousMovementCommands/>
    <ObservationFromFullStats/>
    <RewardForTouchingBlockType>
      <Block reward="200.0" type="lapis_block" behaviour="onceOnly"/>
      <Block reward="-50" type="obsidian" behaviour="onceOnly"/>
      <Block reward="20.0" type="diamond_block" behaviour="onceOnly"/>
      <Block reward='2' type='sandstone' behaviour='oncePerBlock'/>
    </RewardForTouchingBlockType>
    <RewardForTimeTaken initialReward="0" delta="-0.1" density="PER_TICK"/>
    <RewardForSendingCommand reward="-2" />
    <RewardForMissionEnd rewardForDeath="-20.0">
      <Reward description="out_of_time" reward="0.0"/>
    </RewardForMissionEnd>
    <AgentQuitFromTouchingBlockType>
        <Block type="obsidian" />
        <Block type="lapis_block" />
    </AgentQuitFromTouchingBlockType>
  </AgentHandlers>
</AgentSection>
```

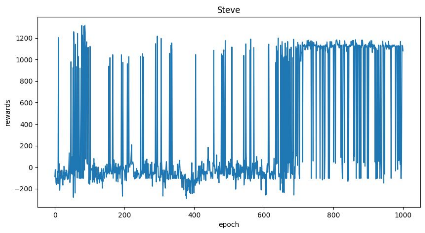
# Evaluation metric

| | Success Rate | Learning Curve |
|---|---|---|
| Q-table | 0.372 |  |
| DQN | 0.1916 |  |
| CNN | 0.0752 |  |

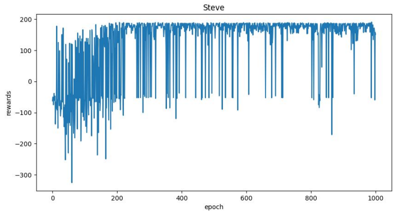
1. 任
2. 學
3. ~~平~~

# Results & analysis & Others // Important

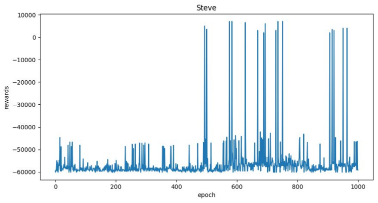
- Change learning rate



Large learning rate  : 0.1



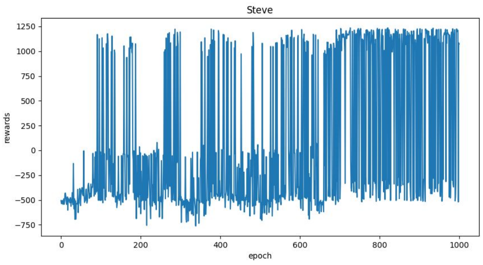
Medium learning rate : 0.01



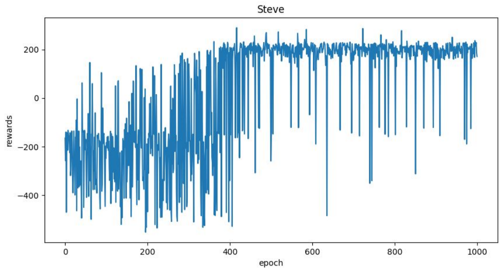
Small learning rate : 0.0001

# Results & analysis & Others

- Change the epilson



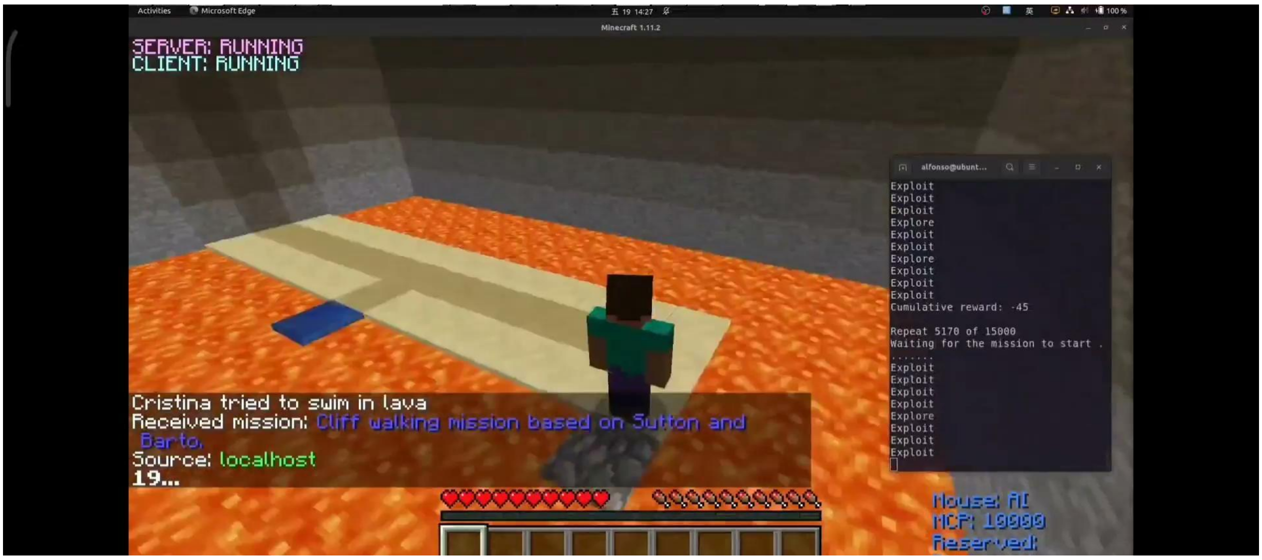
High epilson : 0.99 -> 0.3



Decayed epilson : 0.99->0.01

# Results & analysis & Others // Important

- The reward of the sand is too high



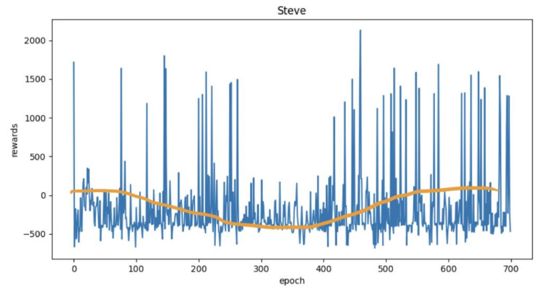
SERVER: RUNNING
CLIENT: RUNNING

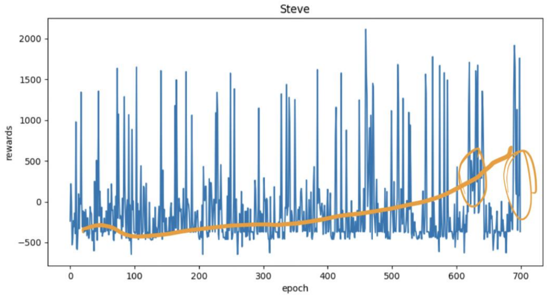
```
Exploit
Exploit
Exploit
Explore
Exploit
Exploit
Explore
Exploit
Exploit
Exploit
Cumulative reward: -45

Repeat 5170 of 15000
Waiting for the mission to start .
.......
Exploit
Exploit
Exploit
Exploit
Explore
Exploit
Exploit
Exploit
```

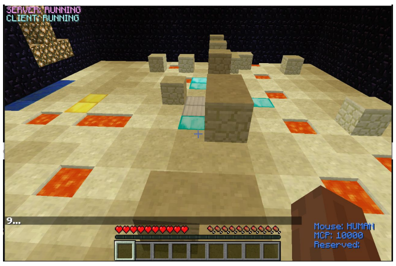
Cristina tried to swim in lava
Received mission: Cliff walking mission based on Sutton and Barto.
Source: localhost
19...

Mouse: AI
MCP: 10000
Reserved:

# DQN-Map3







- Difference:

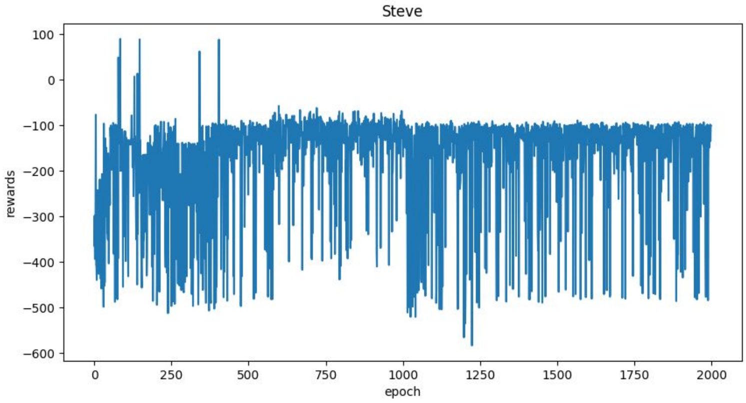
  1. Added: increasing reward block

  2. Added :time penalty & movement reward

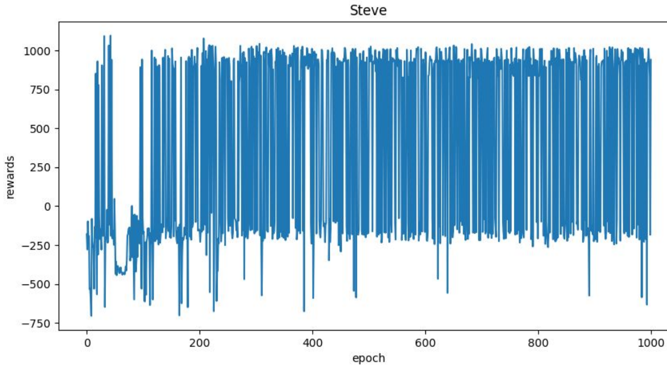
  3.Result: Number of success increased!

# Results & analysis & Others-CNN

- Change gamma rate



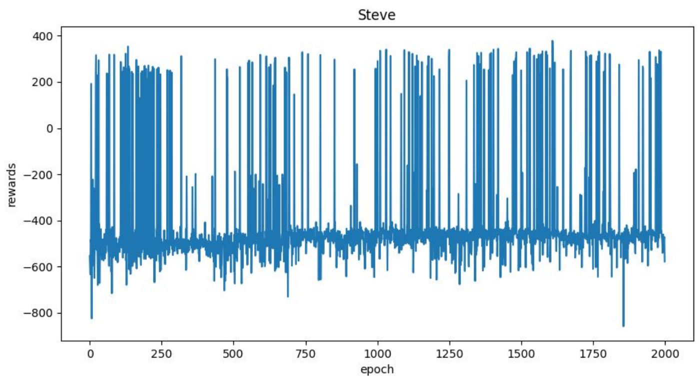
gamma = 0.15



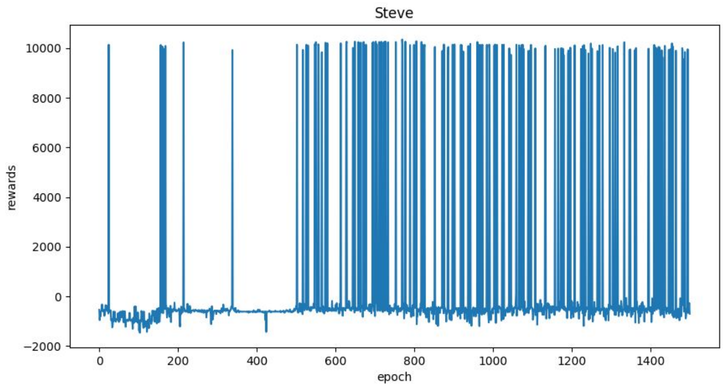
gamma = 0.99

# Results & analysis & Others-CNN

- Change learning rate



learning rate: 0.1



learning rate: 0.001

# Results & analysis & Others-Limitation

- input state
- movement
- q-learning algorithm with previous info and current info

# Github link

https://github.com/zebra314/MineRunner

# Reference

malmo : https://github.com/microsoft/malmo

MineDojo : https://github.com/MineDojo/MineDojo

MineRL:  https://github.com/minerllabs/minerl

AI learn to escape :https://www.youtube.com/watch?v=2tamH76Tjvw&t=20s

# Main Contribution of each member

- 許瑋哲
  - 寫主要CNN演算法、 map file 變換成 input state、處理agent action
- 林穎沛
  - 影片剪輯、寫DQN演算法、Q_table
- 林揚森
  - 調整地圖xml檔、調整reward、數據分析
- 陳宥翔
  - 製作地圖及地圖資訊矩陣、研究xml檔、調整reward
- 共同工作
  - training、報告製作、錄製影片

Me: *uses machine learning*
Machine: *learns*
Me: