



# Gold Hunter Agent

Escaping with the help of DQN Agent

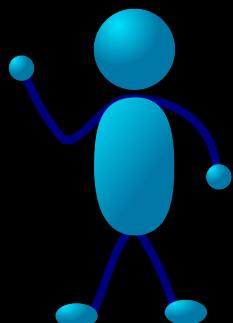
Swee Loke

---

Dec 2021

# About the Game

- The goal is to escape from the room
- Agent starts on the left bottom corner
- Door is at the top right corner
- Dragon starts at the door
- Gold coins are dropping
- Gold pot is at the bottom right corner



```
Time: 0  Action: None, Pot:0
dragon_speed:3, consume_speed:0.9, consumed_in_succession:0
      |     |     | [D]
      | G   | GG  | GGG
G    | GGG |     | GG
      | GG  |
A    |       |     | (θ)
      |     |     |
is_terminated:False, Reward:0.0
Agent:(0, 4), agent_gold:0, alive:True, fainted:False, numFainted:0
last_reward: 0, total:0
```



# The Dragon

- always guards the door
- consumes gold coins at the rate of 0.9 per timestep
- moves immediately to near the gold pot when there is gold
- moves back to the door immediately after finished eating
- moves up and down only
- moves at the speed of 3 cells per timestep



# The Agent



- Maximum of 3 gold coins
- If more than 3 coins, all the previously collected coins at hand will fall and agent will faint.
  - The frozen state lasted 3 timestep
  - In frozen state, no movement can be made, and no additional coins will be collected or cause further fainting
  - The agent can faint at most 5 times, shown by the red bar
- If the agent lost all the lives (due to fainting) or touched by a dragon, the game is terminated

# Impossible Game

- The dragon consumes the gold too fast
- The dragon moves in such speed that it will always reach back the door before the agent

Make it possible by:

- If dragon consumes 10 coins in successions, speed will slow down
- If dragon consumes 15 coins in successions, consumption speed will slow down

# 14 Features

- Agent's current and previous location
- Dragon's current and previous locations
- Falling gold's current and previous locations (with weight)
- Gold # on agent's hands
- Gold # in gold pot
- Number of coins consumed in succession
- Dragon's speed (moving and eating)
- Agent fainted?
- Frozen countdown
- Number of lives agent has left

# Rewards

Reward Type	Value
NORMAL_STEP	-1
ESCAPE	5000
ESCAPE_WITH_MAX_GOLD	8000
CATCH_GOLD	1 * num of coins
LOAD_GOLD	10 * (2 **num of coins)
LOAD_GOLD_BEFORE_EMPTY	50
FAINTED	-200
DEAD	-1000
REACH_MAX_GOLD	300

# Models

Started with 25 filters -> 10 -> 7 filters

Reduced dense layers and neurons

(256->128->128->64->32->5)

to

(128->128->64->32->5)

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 5, 5, 7)	105
flatten (Flatten)	(None, 175)	0
dense (Dense)	(None, 128)	22528
dense_1 (Dense)	(None, 128)	16512
dense_2 (Dense)	(None, 64)	8256
dense_3 (Dense)	(None, 32)	2080
dense_4 (Dense)	(None, 5)	165

Total params: 49,646

Trainable params: 49,646

Non-trainable params: 0

# Early Failures

- When train with default environment, no wins even after 4000 epochs
- Reward alone doesn't help - agent never discovers the goal of the game
- Fixing epsilon at high value early
- Setting max\_step too low
- Use complex Model that consumes too much resources

# Need to train with easy scenario first

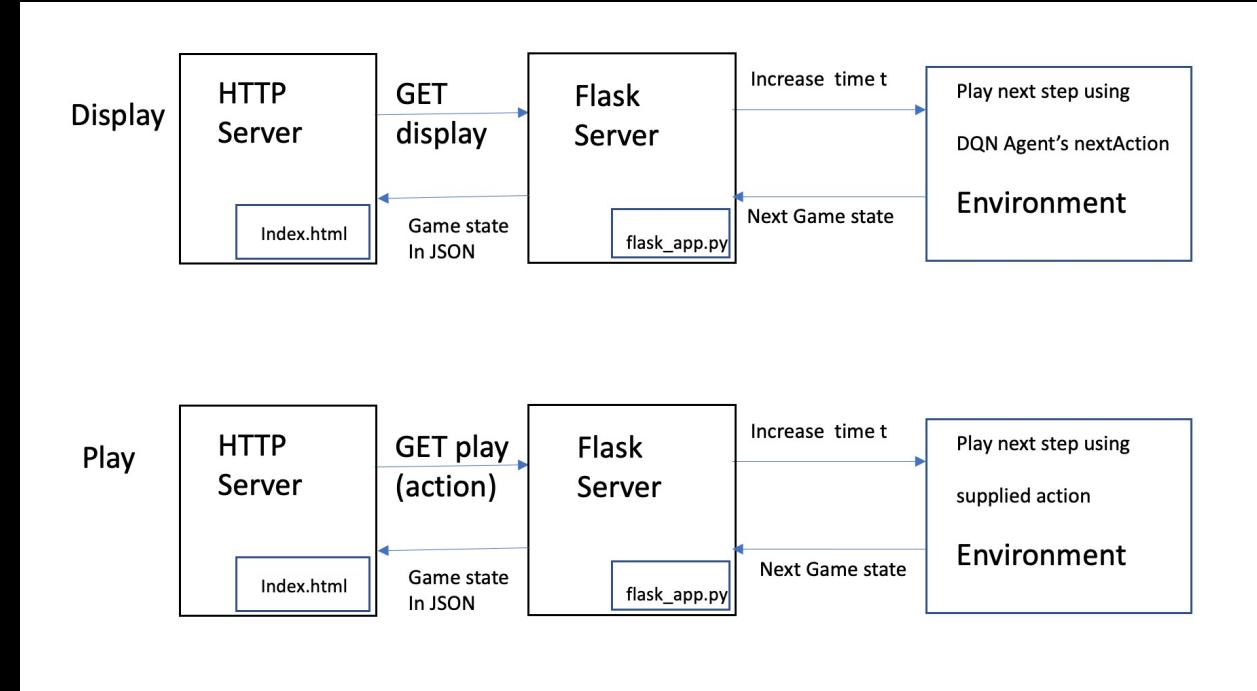
- Train for the end goal first  
Instead of training agent to collect gold first, we train it to escape first
- First 1000 episodes, set the initial pot gold count to 20  
keep the dragon away from the door and it will move back at a slower speed
- Second 1000 episodes, set the initial pot gold count 0 to 15  
Increase the difficulty gradually
- On 2001+ episodes, train with default environments

# Performance

<b>After Epoch</b>	<b>% Wins of 1000 games</b>	<b>Win with NO max gold</b>	<b>Win with max gold</b>
4000	0.9050	0.61	0.39
5000	0.9830	0.57	0.43
6000	0.9730	0.58	0.42
7000	0.9760	0.53	0.47

# Game Implementation

- Run in a docker container
- GUI implementation in JS and PIXI
- HTTP server has no game logic
- Flask server has no knowledge of GUI



- HTTP server makes REST API call to Flask Server
- Flask Server interacts with Environment

Thank you!

Question?