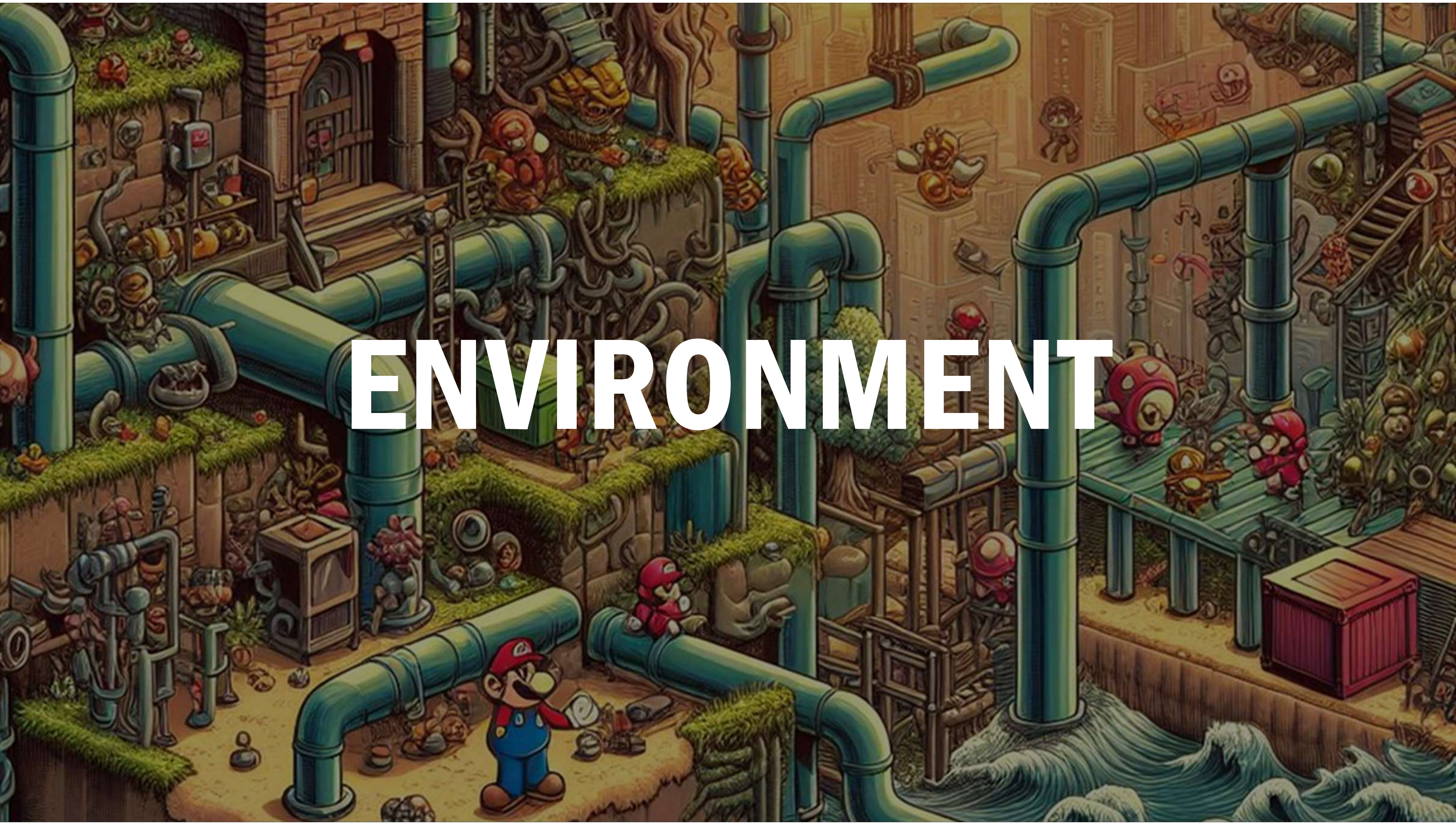
SUPER MARIO AI AGENT

By Kevin Paganini

GOAL

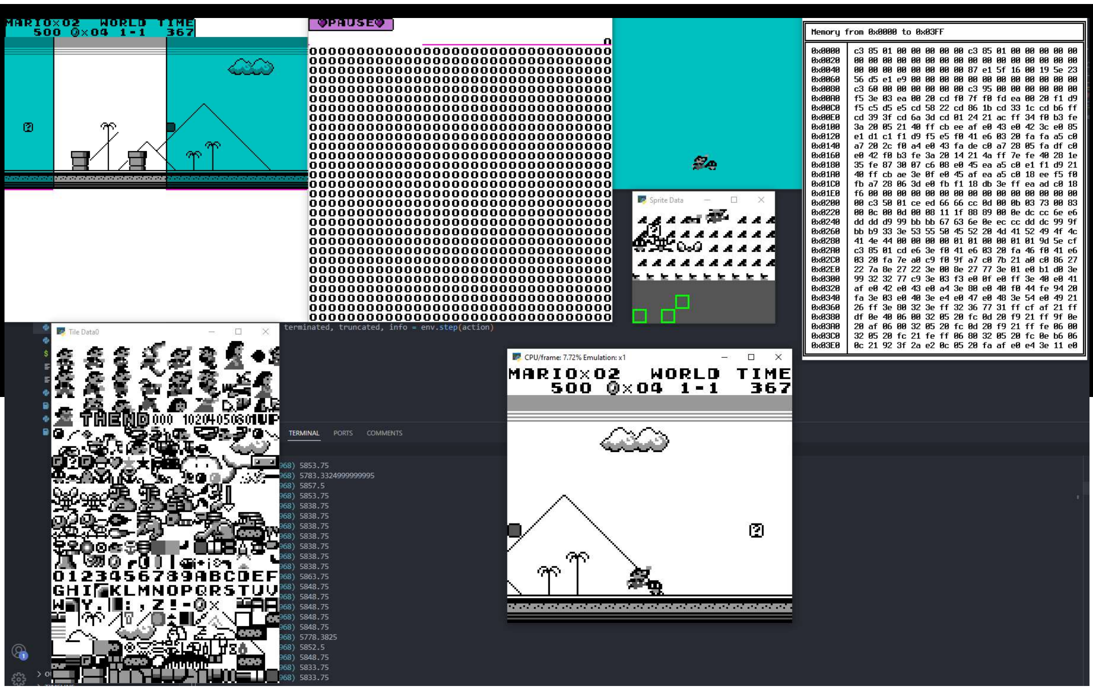


ENVIRONMENT

ENVIRONMENT

- 16x20 Grid of values
- Simplified view of the actual pixels
- 0, 1, 16, 17 is Mario
- Cannot go backwards in environment
- Enemies, Boxes, Pipes all have different values

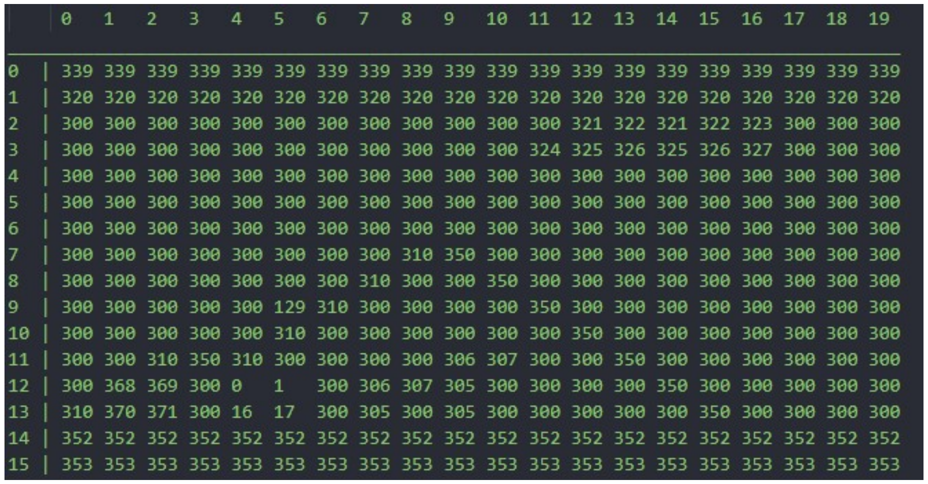
Environment is deterministic, single-agent, stochastic, dynamic and fully observable (for the most part),



```
Memory from 0x0000 to 0x03FF

0x0000  c3 85 01 00 00 00 00 00 c3 85 01 00 00 00 00 00
0x0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0040  00 00 00 00 00 00 00 00 87 e1 5f 16 00 19 5e 23
0x0060  56 d5 e1 e9 00 00 00 00 00 00 00 00 00 00 00 00
0x0080  c3 60 00 00 00 00 00 00 c3 95 00 00 00 00 00 00
0x00A0  f5 3e 03 ea 00 20 cd f0 7f f0 fd ea 00 20 f1 d9
0x00C0  f5 c5 d5 e5 cd 58 22 cd 86 1b cd 33 1c cd b6 ff
0x00E0  cd 39 3f cd 6a 3d cd 01 24 21 ac ff 34 f0 b3 fe
0x0100  3a 20 05 21 40 ff cb ee af e0 43 e0 42 3c e0 85
0x0120  e1 d1 c1 f1 d9 f5 e5 f0 41 e6 03 20 fa fa a5 c0
0x0140  a7 20 2c f0 a4 e0 43 fa de c0 a7 28 05 fa df c0
0x0160  e0 42 f0 b3 fe 3a 20 14 21 4a ff 7e fe 40 28 1e
0x0180  35 fe 87 30 07 c6 08 e0 45 ea a5 c0 e1 f1 d9 21
0x01A0  40 ff cb ae 3e 0f e0 45 af ea a5 c0 18 ee f5 f0
0x01C0  fb a7 28 06 3d e0 fb f1 18 db 3e ff ea ad c0 18
0x01E0  f6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0200  00 c3 50 01 ce ed 66 66 cc 0d 00 0b 03 73 00 83
0x0220  00 0c 00 0d 00 08 11 1f 88 89 00 0e dc cc 6e e6
0x0240  dd dd d9 99 bb bb 67 63 6e 0e ec cc dd dc 99 9f
0x0260  bb b9 33 3e 53 55 50 45 52 20 4d 41 52 49 4f 4c
0x0280  41 4e 44 00 00 00 00 01 01 00 00 01 01 9d 5e cf
0x02A0  c3 85 01 cd e6 3e f0 41 e6 03 20 fa 46 f0 41 e6
0x02C0  03 20 fa 7e a0 c9 f0 9f a7 c0 7b 21 a0 c0 86 27
0x02E0  22 7a 8e 27 22 3e 00 8e 27 77 3e 01 e0 b1 d0 3e
0x0300  99 32 32 77 c9 3e 03 f3 e0 0f e0 ff 3e 40 e0 41
0x0320  af e0 42 e0 43 e0 a4 3e 80 e0 40 f0 44 fe 94 20
0x0340  fa 3e 03 e0 40 3e e4 e0 47 e0 48 3e 54 e0 49 21
0x0360  26 ff 3e 80 32 3e ff 32 36 77 31 ff cf af 21 ff
0x0380  df 0e 40 06 00 32 05 20 fc 0d 20 f9 21 ff 9f 0e
0x03A0  20 af 06 00 32 05 20 fc 0d 20 f9 21 ff fe 06 00
0x03C0  32 05 20 fc 21 fe ff 06 80 32 05 20 fc 0e b6 06
0x03E0  0c 21 92 3f 2a e2 0c 05 20 fa af e0 e4 3e 11 e0
```

```
terminated, truncated, info = env.step(action)
```

```
968) 5853.75
968) 5783.3324999999995
968) 5857.5
968) 5853.75
968) 5838.75
968) 5838.75
968) 5838.75
968) 5838.75
968) 5838.75
968) 5838.75
968) 5838.75
968) 5863.75
968) 5848.75
968) 5848.75
968) 5848.75
968) 5848.75
968) 5848.75
968) 5778.3825
968) 5852.5
968) 5848.75
968) 5833.75
968) 5833.75
```



```
      0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19
───────────────────────────────────────────────────────────────────────────────────────
0   | 339 339 339 339 339 339 339 339 339 339 339 339 339 339 339 339 339 339 339 339
1   | 320 320 320 320 320 320 320 320 320 320 320 320 320 320 320 320 320 320 320 320
2   | 300 300 300 300 300 300 300 300 300 300 300 300 321 322 321 322 323 300 300 300
3   | 300 300 300 300 300 300 300 300 300 300 300 324 325 326 325 326 327 300 300 300
4   | 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300
5   | 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300
6   | 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300
7   | 300 300 300 300 300 300 300 300 310 350 300 300 300 300 300 300 300 300 300 300
8   | 300 300 300 300 300 300 300 310 300 300 350 300 300 300 300 300 300 300 300 300
9   | 300 300 300 300 300 129 310 300 300 300 300 350 300 300 300 300 300 300 300 300
10  | 300 300 300 300 300 310 300 300 300 300 300 300 350 300 300 300 300 300 300 300
11  | 300 300 310 350 310 300 300 300 300 306 307 300 300 350 300 300 300 300 300 300
12  | 300 368 369 300 0   1   300 306 307 305 300 300 300 300 350 300 300 300 300 300
13  | 310 370 371 300 16  17  300 305 300 305 300 300 300 300 300 350 300 300 300 300
14  | 352 352 352 352 352 352 352 352 352 352 352 352 352 352 352 352 352 352 352 352
15  | 353 353 353 353 353 353 353 353 353 353 353 353 353 353 353 353 353 353 353 353
```

ACTIONS

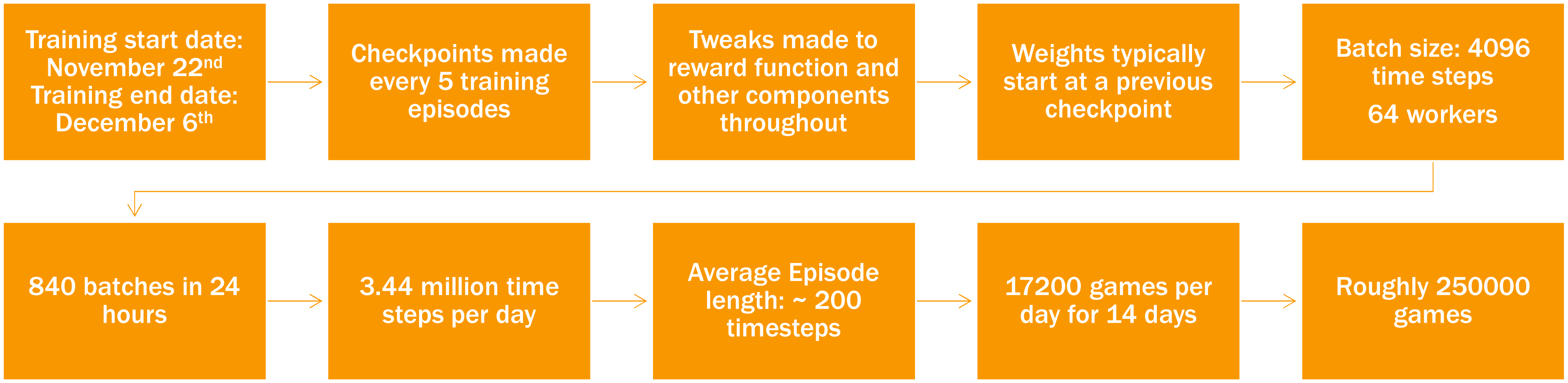
ACTIONS

In the real world: RIGHT, LEFT and JUMP

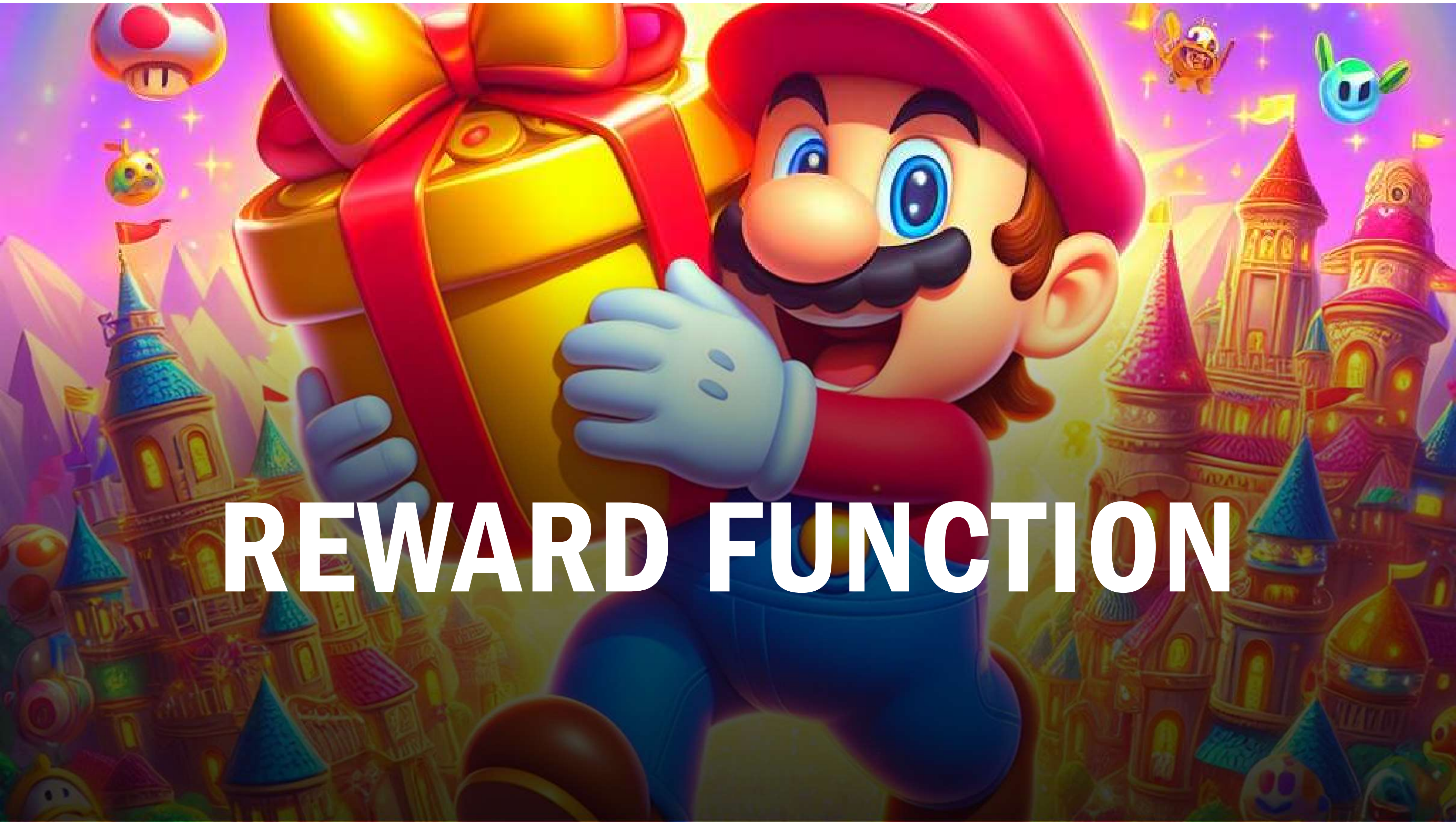
Agent's actions:
- PRESS RIGHT
- PRESS LEFT
- JUMP
- RIGHT + JUMP
- LEFT + JUMP
- RELEASE RIGHT
- RELEASE LEFT
- DO NOTHING

TRAINING

TRAINING DETAILS


Training start date: November 22nd Training end date: December 6th
Checkpoints made every 5 training episodes
Tweaks made to reward function and other components throughout
Weights typically start at a previous checkpoint
Batch size: 4096 time steps
64 workers
840 batches in 24 hours
3.44 million time steps per day
Average Episode length: ~ 200 timesteps
17200 games per day for 14 days
Roughly 250000 games

TRAINING MONTAGE

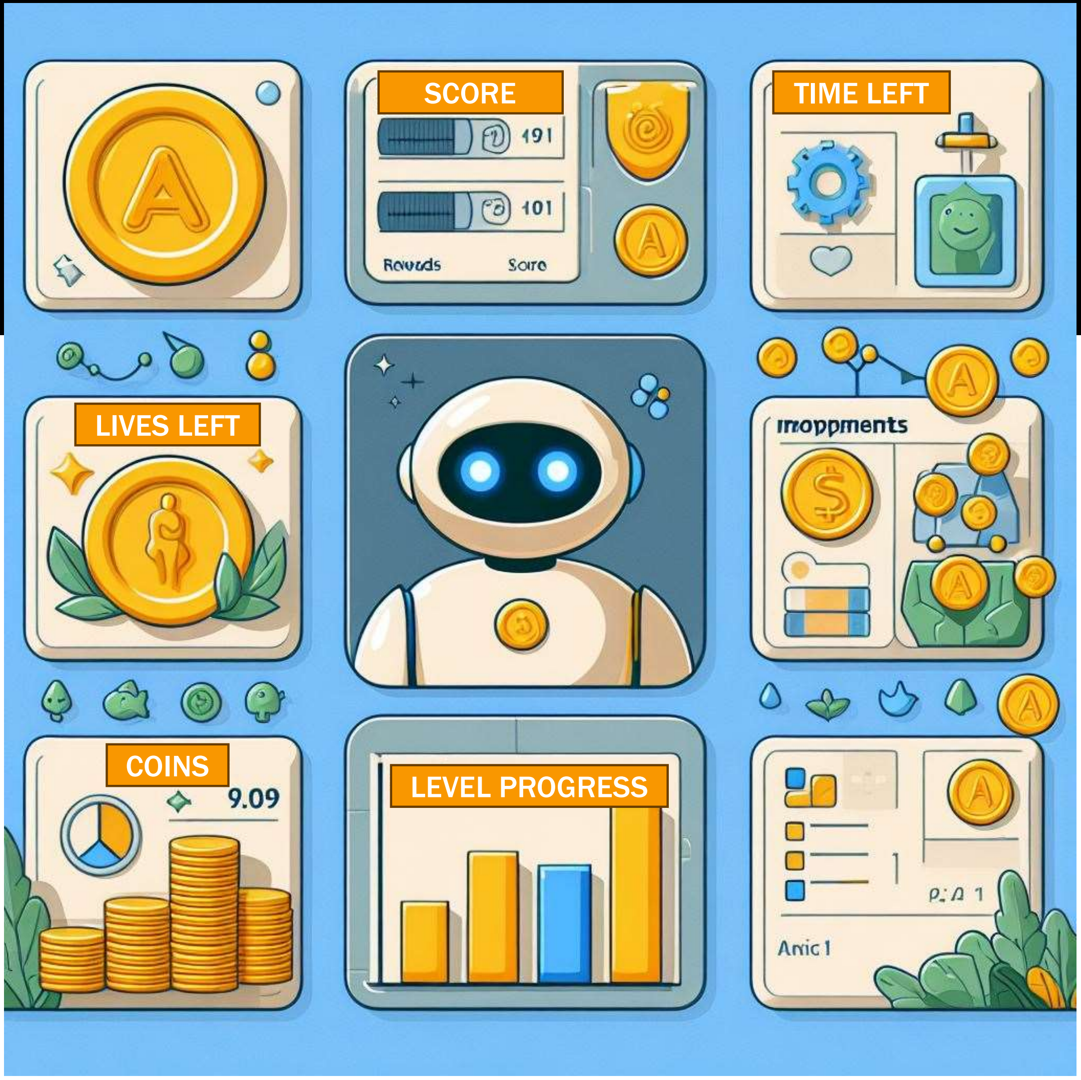


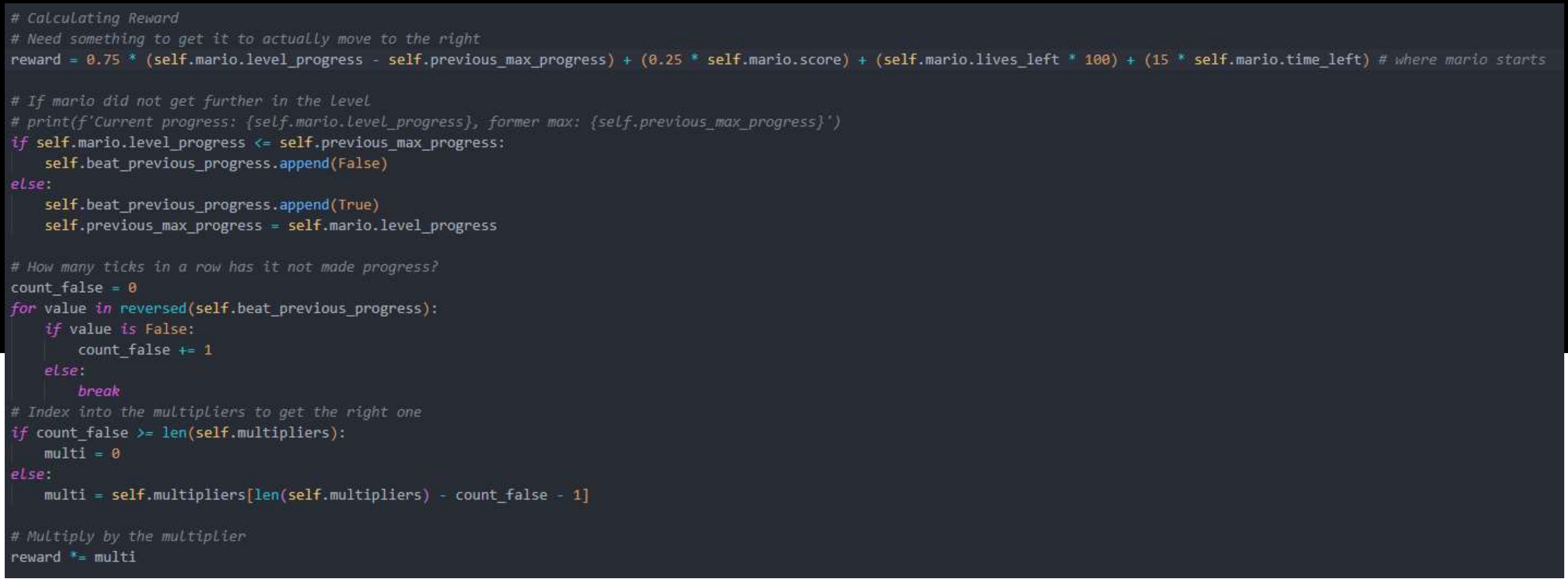
REWARD FUNCTION

REWARD FUNCTION

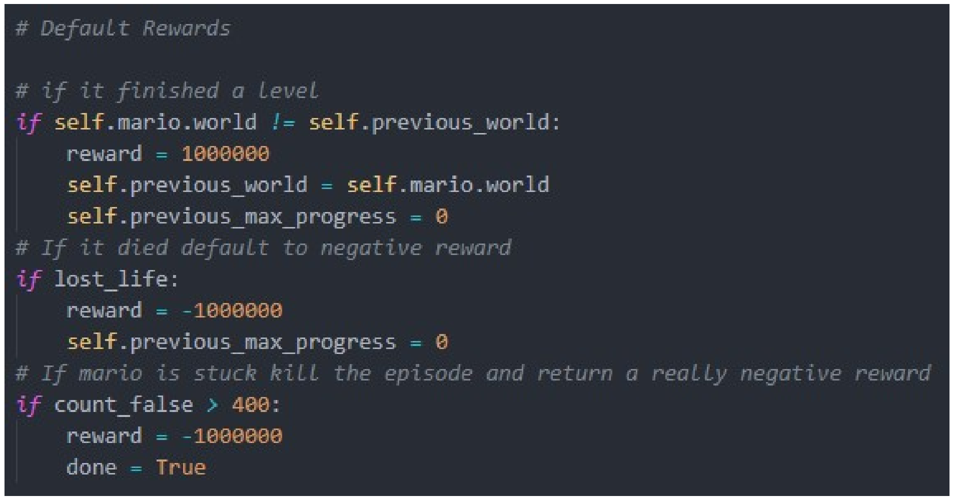
Reward parameter options:
- coins
- lives left
- score
- time left
- level progress

Important add ons:
- Reward decay
- Default Rewards

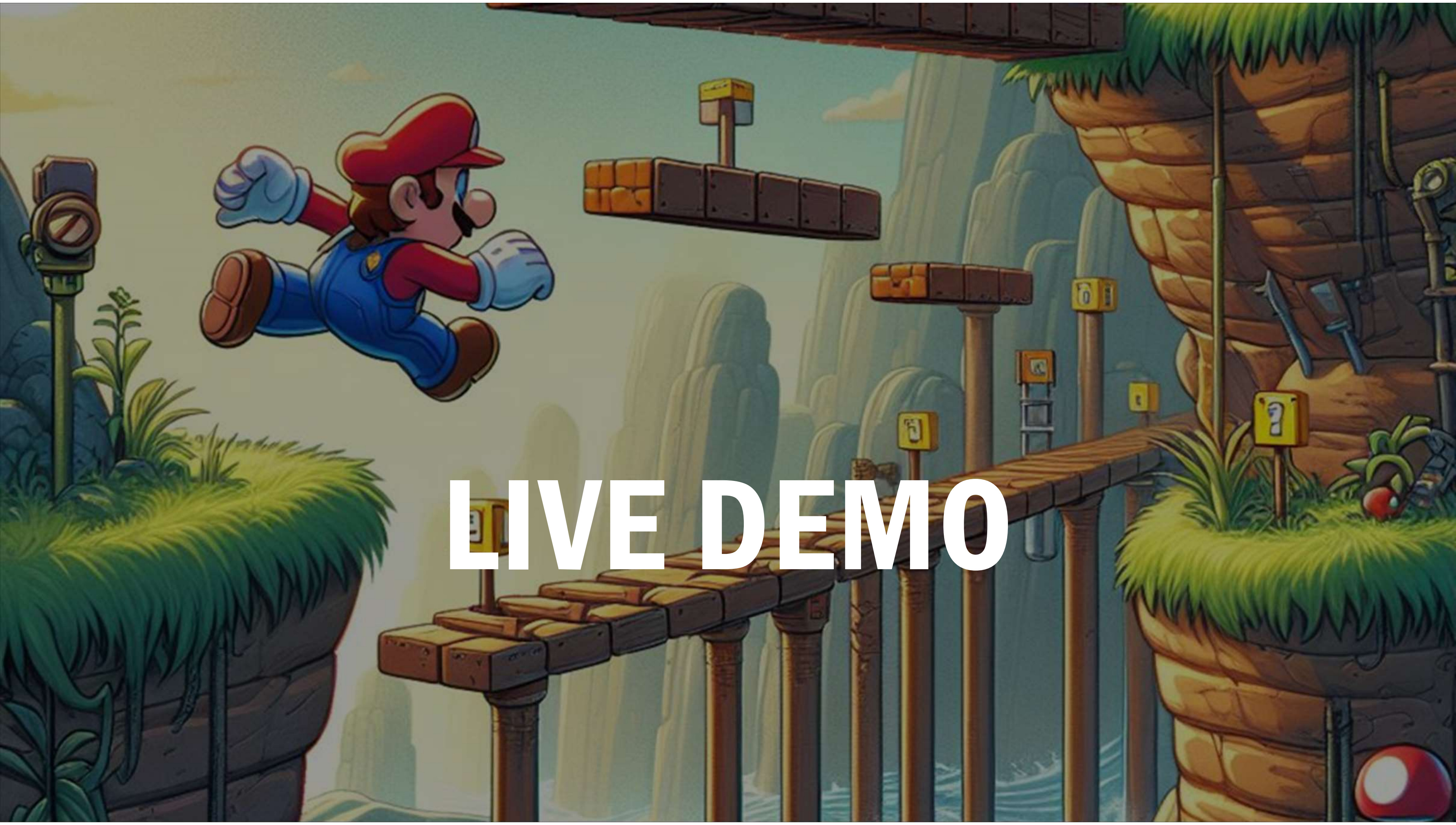




```python
# Calculating Reward
# Need something to get it to actually move to the right
reward = 0.75 * (self.mario.level_progress - self.previous_max_progress) + (0.25 * self.mario.score) + (self.mario.lives_left * 100) + (15 * self.mario.time_left) # where mario starts

# If mario did not get further in the level
# print(f'Current progress: {self.mario.level_progress}, former max: {self.previous_max_progress}')
if self.mario.level_progress <= self.previous_max_progress:
    self.beat_previous_progress.append(False)
else:
    self.beat_previous_progress.append(True)
    self.previous_max_progress = self.mario.level_progress

# How many ticks in a row has it not made progress?
count_false = 0
for value in reversed(self.beat_previous_progress):
    if value is False:
        count_false += 1
    else:
        break
# Index into the multipliers to get the right one
if count_false >= len(self.multipliers):
    multi = 0
else:
    multi = self.multipliers[len(self.multipliers) - count_false - 1]

# Multiply by the multiplier
reward *= multi
```



```python
# Default Rewards

# if it finished a level
if self.mario.world != self.previous_world:
    reward = 1000000
    self.previous_world = self.mario.world
    self.previous_max_progress = 0
# If it died default to negative reward
if lost_life:
    reward = -1000000
    self.previous_max_progress = 0
# If mario is stuck kill the episode and return a really negative reward
if count_false > 400:
    reward = -1000000
    done = True
```



LIVE DEMO

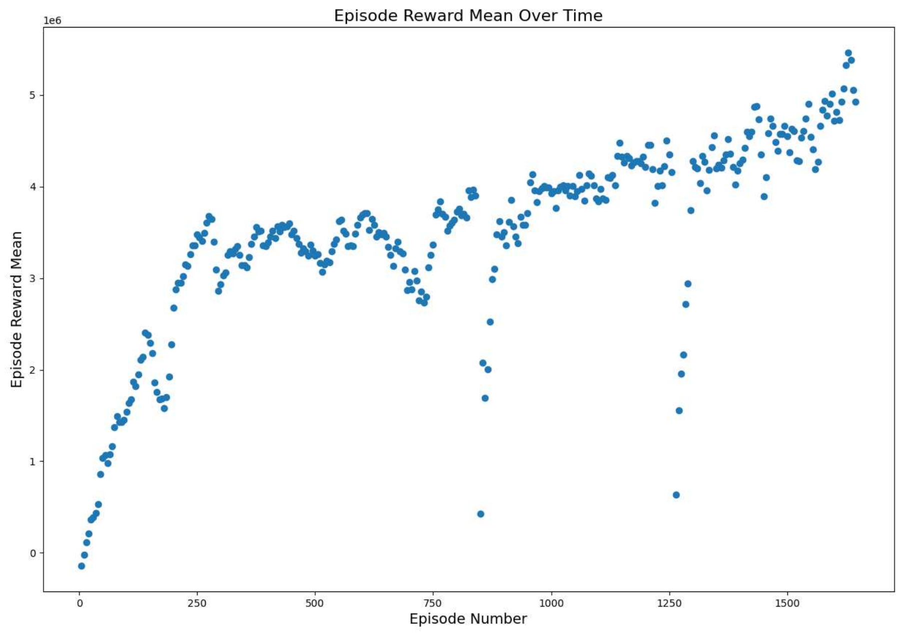
END RESULT

- Mario goes super fast
- Mario does not care about coins
- Mario consistently beats first level
- Given more training I believe he could beat every level

REWARD THROUGHOUT TRAINING

- Initial performance increase is due to previous training

- Drops in the middle of training are due to the job restarting and the optimizer needing to be loaded up



IMPROVEMENTS

- Different game view
- Bigger batch sizes
- Starting on different levels
- Increase Decay Rate of rewards
- Larger Neural Network

QUESTIONS
COMMENTS
CONCERNS