**Mô Tả Mã Nguồn Mẫu: Huấn Luyện và Thi đấu**

1. **Huấn luyện - Miner-Training-Local-CodeSample**

Đây là mã nguồn (source code) mẫu được sử dụng cho quá trình huấn luyện tại máy của các đội chơi. Mã nguồn bao gồm 02 phần chính: môi trường trò chơi đào vàng (Miner Game Environment) và thuật toán học tăng cường (Deep-Q learning -DQN). Chi tiết của hai phần như sau:

1. ***Môi trường trò chơi đào vàng (Miner Game Environment).***

Mã nguồn của môi trường được lấy từ mã nguồn gốc của trò chơi đào vàng (Miner Game) trên hệ thống Codelearn. Mã nguồn bao gồm: GameSocketDummy.py, MinerEnv.py, MinerState.py, Maps, và 03 Bots (bot1.py, bot2.py, bot3.py).

* 1. **GAME\_SOCKET\_DUMMY.py:** Chương trình mô phỏng lại trò chơi đào vàng bao gồm cả quá trình truyền nhận dữ liệu (message) tới máy chủ (server).

Các đối tượng/hàm chính:

|  |  |  |
| --- | --- | --- |
| No | Tên | Mô tả |
| 1 | \_\_init\_\_ (host, port) | Khởi tạo host, port cho server |
| 2 | init\_bots | Hỗ trợ người chơi huấn luyện agent với bot.  Để chỉ định bot tham gia vào huấn luyện, sử dụng dòng lệnh: self.bots = [Bot1(2), Bot2(3), Bot3(4)] |
| 3 | Bot | Bot là 1 class được định nghĩa với 3 hàm chính:  new\_game(data) -> thực hiện reset data  new\_state(data) -> update state nhận được từ server – data  next\_action -> trả về action cho step tiếp theo |
| 4 | connect | Hàm này mô phỏng hành động connect từ client đến server |
| 5 | receive | Hàm này mô phỏng hành động client nhận message từ server |
| 6 | send | Hàm này mô phỏng hành động client gửi message lên server.  Ở phiên bản Training, sẽ có 2 kiểu message từ client:  - Action: là action cho step tiếp theo trong game, là kiểu số  - Request: là yêu cầu các thông số cho việc khởi tạo game. Các thông số bao gồm: map, init\_x, init\_y, init\_energe, max\_step.  VD: request = map0;1;2;100;150  Có nghĩa là: server sẽ sử dụng thông tin map (vàng, vật cản) theo file map0, khởi tạo vị trí ban đầu cho người chơi tại ô x = 1, y = 2, với năng lượng ban đầu là 100, và trận đấu có tối đa 150 lượt chơi. |

Như vậy, luồng xử lý từ client đến server sẽ như sau:

**Với phiên bản Training:**

GameSocket(host, port) -> connect -> send (request) -> receive (nhận game info) -> send (action) -> receive (nhận step state)

**Với phiên bản Testing:**

GameSocket(host, port) -> connect -> receive (nhận game info) -> send (action) -> receive (nhận step state)

* 1. **MINER\_STATE.py:** Là code mẫu cho việc lưu trữ data nhận được từ server.

Các đối tượng/hàm chính:

|  |  |  |
| --- | --- | --- |
| No | Tên | Mô tả |
| 1 | MapInfo | Là class dùng cho việc lưu trữ toàn bộ thông tin của bản đồ. Bao gồm:  max\_x  max\_y  maxStep  numberOfPlayer  golds: số vàng còn lại trên map tại thời điểm hiện tại  obstacles: thông tin các vật cản hiện tại trên bản đồ |
| 2 | MapInfo.update (golds, changedObstacles) | Cập nhật lại thông tin bản đồ sau mỗi step |
| 3 | State | Là class chứa trạng thái của trò chơi (bao gồm trạng thái của người chơi và bản đồ) |
| 4 | State.init\_state(data) | Khởi tạo thông tin của bản đồ và trạng thái của người chơi tại thời điểm bắt đầu trận đấu |
| 5 | State.update\_state(data) | Cập nhật trạng thái của trò chơi sau mỗi lượt chơi |

Cách sử dụng MINER\_STATE theo luồng xử lý:

receive (nhận game info) & state.init\_state -> send (action) -> receive (nhận step state) & state.update\_state

* 1. **MinerEnv.py**: Chương trình được thiết kế theo cấu trúc chung của môi trường học tăng cường (Reinforcement learning environment) cho phép các đội chơi truy cập tới chương trình chính (GameSocketDummy**.**py) đơn giản và thuận tiện. Cấu trúc bên trong bao gồm 04 hàm chính: *reset(), step(action), get\_state(), và get\_reward()*. Các đội chơi tùy vào chiến lược huấn luyện riêng có thể viết lại các hàm *get\_state() và get\_reward()* cho phù hợp.
  2. **Maps**: Thư mục Maps chứa 05 bản đồ (map) mẫu phục vụ cho việc huấn luyện. Thông tin về bẫy trong 05 bản đồ là được giữ nguyên, và chỉ thay đổi vị trí vàng và số lượng vàng trong vòng sơ loại. Đội chơi có thể thiết kế lại những bản đồ này cho phù hợp với chiến lược huấn luyện riêng. Một số nội dung cần chú ý khi các đội chơi làm việc với Maps như sau:
* Mỗi file trong thư mục Maps được xem là một bản đồ, tên của file (filename) được xem là tên của bản đồ (map name).
* Mỗi bản đồ là một ma trận các số nguyên với ý nghĩa như sau:

|  |  |
| --- | --- |
| 0 | Đất |
| -1 | Rừng |
| -2 | Bẫy |
| -3 | Đầm lầy |
| > 0 | Vàng |

* 1. Bots: 03 bots (bot1.py, bot2.py, và bot3.py) được cung cấp cho các đội chơi. Các đội chơi có thể tạo những bot theo chiến lược huấn luyện riêng. Các bots sẽ được đưa vào trong môi trường trò chơi trong file GameSocketDummy**.**py**.** Bao gồm 02 bước: khai báo bots (*import Bot1*) và khởi tạo bot (*init\_bots()*).

1. **Thuật toán học tăng cường (Deep Q-learning)**

Trong phần này, mã nguồn được viết theo thuật toán học tăng cường sâu (Deep Q-learning - DQN). Nguồn thuật toán được giới thiệu trong nghiên cứu của Mnih et al ("Human-level control through deep reinforcement learning." *Nature* 518.7540 (2015): 529-533).

Mã nguồn thuật toán bao gồm những file chương trình sau: TrainingClient.py, DQNModel.py, và Memory.py.

1. **TrainingClient.py**: đây là mã nguồn của thuật toán DQN cho phép giao tiếp với phần môi trường trò chơi.
2. **DQNModel.py**: mã nguồn này được thiết kế cho phép tạo những mô hình học sâu (deep learning models) và những hàm huấn luyện mô hình.
3. **Memory.py**: mã nguồn được sử dụng để lưu dữ liệu (experiences) cho việc huấn luyện

* ***Note***: *Mã nguồn trên được dùng cho việc huấn luyện nên trận đấu kết thúc chỉ khi hết vàng trên bản đồ (map) hoặc người chơi bị loại (khi hết vàng và ra khỏi bản đồ).*

1. **Miner-Testing-CodeSample**

* Mã nguồn được thiết kế cho các đội chơi sử dụng để tham gia cuộc thi chính thức.
* Điểm khác với mã nguồn cung cấp cho phần huấn luyện (**Miner-Training-Local-CodeSample**) đó là sử dụng GameSocket.py thay cho DummyGameSocket.py. Chương trình GameSocket.py cho phép truyền nhận dữ liệu tới máy chủ (server).
* HOST và PORT của server được lấy từ biến môi trường khi run TestingAgent.py
* Những mã nguồn còn lại (MinerState.py và MinerEnv.py) được giữ nguyên như trong mã nguồn được cung cấp cho huấn luyện (Miner-Testing-CodeSample).