# Unity ↔ GAMA: Truyền dữ liệu hai chiều (WebSocket)

## 1) Mục tiêu

- Unity và GAMA nói chuyện 2 chiều bằng WebSocket.

- Unity gửi \*\*tọa độ\*\* và \*\*lệnh\*\* (`pause`, `resume`) sang GAMA.

- GAMA cập nhật agent theo tọa độ và có thể gửi \*\*phản hồi/nhịp\*\* (`ack`, `cycle`) ngược về Unity.

## 2) Sơ đồ nhanh

```

Unity (WebSocket Server) ──JSON──> GAMA (WebSocket Client)

^ |

| v

└────────────── JSON <──────────────┘

```

- Unity mở cổng ws://127.0.0.1:3001, chờ GAMA kết nối.

- GAMA kết nối với `protocol: "websocket\_client"` và dùng `send` / `fetch\_message` để trao đổi chuỗi JSON.

## 3) Định dạng gói JSON

- \*\*pos\*\*: cập nhật vị trí agent trong GAMA

```json

{ "cmd": "pos", "id": "player1", "x": 1.23, "y": 4.56, "z": 0.78 }

```

Quy ước: GAMA(x,y) = Unity(x,z); trục \*\*y\*\* của Unity được lưu tạm vào trường `z`.

- \*\*pause\*\*: dừng logic mô phỏng trong GAMA

```json

{ "cmd": "pause" }

```

- \*\*resume\*\*: tiếp tục mô phỏng trong GAMA

```json

{ "cmd": "resume" }

```

- \*\*ack\*\* (tùy chọn, Unity phản hồi lại mọi gói nhận được)

```json

{ "ack": "ok" }

```

- \*\*cycle\*\* (tùy chọn, GAMA gửi nhịp để Unity thấy chiều ngược hoạt động)

```json

{ "cmd": "cycle", "cycle": 42 }

```

## 4) Luồng Unity → GAMA (gửi tọa độ và lệnh)

\*\*Ở Unity\*\*

- `UnityGamaBridge.Update()` phát lệnh khi bấm phím: \*\*P = pause\*\*, \*\*R = resume\*\*.

- Định kỳ `posInterval`, hàm `BroadcastPosition()` trích `Transform.position` của đối tượng theo quy ước (x,z) → (x,y) và phát JSON qua `server.Broadcast(...)`.

- Tất cả gói đi qua `UnityWsServer.Broadcast()` để gửi tới mọi client WebSocket đã kết nối.

\*\*Ở GAMA\*\*

- `reflex rx` gọi `fetch\_message\_from\_network` rồi lặp `fetch\_message()` tối đa 64 lần mỗi tick.

- Chuyển `m.contents` sang chuỗi, parse `from\_json(...)`.

- Nếu `cmd = "pos"` → tạo hoặc cập nhật `marker` có `id` tương ứng và gán `location <- {x, y}`.

- Nếu `cmd = "pause"`/`"resume"` → đặt/nhả cờ `paused` (có thể dùng cho `pause\_condition` hoặc chặn các `reflex` khác).

## 5) Luồng GAMA → Unity (phản hồi/nhịp)

\*\*Ở GAMA\*\*

- Dùng `do send to: "ws" contents: "<chuỗi JSON>"` để gửi sang Unity (ví dụ ack/cycle).

\*\*Ở Unity\*\*

- `UnityWsServer` xếp mọi gói nhận được vào hàng đợi (\_inbox).

- `UnityGamaBridge.Update()` gọi `server.DrainInbox(OnGamaMessage, ...)` để rút gói và xử lý trong `OnGamaMessage(string s)`.

- Ví dụ: trích `cycle` và đổi `scale` của `probeVisual` để xác nhận chiều ngược hoạt động.

## 6) Hàm gửi/nhận — giải thích nhanh cho người không rành code

### Unity

- \*\*Gửi\*\*: `server.Broadcast(string json)`

Ném chuỗi JSON cho tất cả client đang kết nối (ở đây là GAMA).

Nên gửi chuỗi hợp lệ dạng JSON để GAMA `from\_json` được.

- \*\*Nhận\*\*: `server.DrainInbox(Action<string> handler, int maxPerFrame)`

Rút từng gói từ hàng đợi; với mỗi gói, Unity gọi `OnGamaMessage(s)`.

Trong `OnGamaMessage`, có thể: in log, đọc trường `cycle`, trả ACK, v.v.

- \*\*Phím tắt\*\*: P → `{"cmd":"pause"}`, R → `{"cmd":"resume"}`.

### GAMA

- \*\*Gửi\*\*: `do send to: "ws" contents: "<json>"`

Ví dụ: `"{\"cmd\":\"cycle\",\"cycle\":42}"`.

- \*\*Nhận\*\*:

1) `do fetch\_message\_from\_network;` (kéo gói mới về hộp thư)

2) `message m <- fetch\_message();` (lấy 1 gói)

3) `string s <- string(m.contents);` (đổi sang chuỗi)

4) `map js <- from\_json(s);` → đọc `js["cmd"]`, `js["id"]`, `js["x"]`, `js["y"]`…

5) Cập nhật agent/biến cờ theo nhu cầu.

## 7) Kiểm thử tối thiểu

1. Chạy Unity (Play). Console Unity phải in dòng gửi `{"cmd":"pos"...}` liên tục.

2. Chạy GAMA. Console GAMA in `rx: {...}` và `pulled N msg(s)`.

3. Nhấn \*\*P\*\* trong Unity → GAMA dừng.

4. Nhấn \*\*R\*\* trong Unity → GAMA tiếp tục.

5. GAMA gửi `{"cmd":"cycle",...}` định kỳ → Console Unity in gói nhận được; `probeVisual` đổi scale (nếu gán).

## 8) Lỗi thường gặp

- Không cùng cổng/địa chỉ WS → GAMA không kết nối được.

- Tắt `raw` ở kết nối GAMA → chuỗi nhận về bị bọc; JSON hỏng.

- Sai định dạng JSON (thiếu ngoặc kép, sai dấu phẩy) → `from\_json` trả `nil`.

- Không rút inbox ở Unity (`DrainInbox`) → Unity không thấy gói từ GAMA.

- Quên map trục: dùng Unity.y cho GAMA.y sẽ khiến marker nhảy sai.

## 9) File tham chiếu

- Unity: `UnityGamaBridge.cs` (gửi pos/pause/resume, nhận cycle/ack), `UnityWsServer.cs` (WS server, Broadcast, hàng đợi).

- GAMA: `ws\_pos\_bridge.gaml` (connect, nhận gói, cập nhật marker, gửi ack/cycle).