Unity and Python Communication for Generative Agents

Unity and Python will communicate through WebSockets, facilitating interaction among generative agents. Each command consists of an invoker, functionality, parameters, and output, structured in JSON format.

Command Structure

The communication between Unity (frontend) and Python (backend) occurs via JSON objects sent over WebSockets. Below is a detailed breakdown of commands, including their invokers, functionality, parameters, and expected outputs.

1. `command.auth.Register`
    - **Invoker**: Unity
    - **Functionality**: Registers a new player in the system.
    - **Parameters**: `{"name": str}`
    - **Output**: `{"status": "success"}`
        - Returns an empty JSON object upon successful registration.
2. `command.building.GetBuildingInfo`
    - **Invoker**: Python
    - **Functionality**: Retrieves detailed information about a specific building, including its name and associated NPCs.
    - **Parameters**: `{"buildingID": int}`
    - **Output**: `{"building_name": str, "NPCs": [npc]}`
        - Returns the building's name and a list of assigned NPCs.
3. `command.building.GetBuildings`
    - **Invoker**: Python
    - **Functionality**: Retrieves a list of all buildings present in the game.
    - **Parameters**: None
    - **Output**: `{"buildings": [{"buildingID": int, "building_name": str}]}`
        - Returns a list of all buildings with their respective IDs and names.
4. `command.chat.ChatToNPC`
    - **Invoker**: Unity
    - **Functionality**: Sends a chat message to a specific NPC in the game world.
    - **Parameters**: `{"NPCID": int, "content": str}`
    - **Output**: `{"response": str}`
        - Returns the NPC’s response to the chat message.
5. `command.chat.NPCChatUpdate`
    - **Invoker**: Python
    - **Functionality**: Updates the chat bubble for an NPC.
    - **Parameters**: None
    - **Output**: None
        - Updates the NPC's chat bubble based on the latest information.
6. `command.config.GetBuildingsConfig`
    - **Invoker**: Python
    - **Functionality**: Retrieves configuration data for all building types in the game.
    - **Parameters**: None
    - **Output**: `{"building_types": [{"type_id": int, "name": str}]}`
        - Returns all available building types and their attributes.
7. `command.config.GetEquipmentsConfig`
    - **Invoker**: Python
    - **Functionality**: Retrieves configuration data for all types of equipment in the game.
    - **Parameters**: None
    - **Output**: `{"equipments": [{"equipmentID": int, "name": str, "attributes": {}}]}`
        - Returns a list of equipment and their configurations.
8. `command.config.GetNPCsConfig`
    - **Invoker**: Python
    - **Functionality**: Retrieves configuration data for all types of NPCs in the game.
    - **Parameters**: None
    - **Output**: `{"npc_types": [{"npcID": int, "npc_name": str}]}`
        - Returns a list of NPC configurations and their attributes.
9. `command.map.GetMapScene`
    - **Invoker**: Python
    - **Functionality**: Retrieves data for the current scene of the map being used in the game.
    - **Parameters**: None
    - **Output**: `{"scene_name": str, "details": {}}`
        - Returns the current scene name and its details.
10. `command.map.GetMapTown`
    - **Invoker**: Python
    - **Functionality**: Retrieves data for the town map, including details of NPCs, buildings, etc.
    - **Parameters**: None

- **Output**: `{"town_map": {}}`
    - Returns details of the town's layout, including NPCs, buildings, and more.

11. **`command.map.UserNavigate`**
    - **Invoker**: Unity
    - **Functionality**: Moves a player to a specific location on the map using X and Y coordinates.
    - **Parameters**: `{"x": int, "y": int}`
    - **Output**: None
        - Confirms if the movement was successful or failed.

12. **`command.map.NPCNavigate`**
    - **Invoker**: Python
    - **Functionality**: Directs an NPC to move to a specified location on the map using X and Y coordinates, generally controlled by AI.
    - **Parameters**: `{"npc_id": int, "x": int, "y": int}`
    - **Output**: None
        - Ensures the NPC receives the movement directive and the backend tracks the NPC's movement.

13. **`command.map.NPCNavigateTime`**
    - **Invoker**: Python
    - **Functionality**: Calculates the distance and estimated travel time from the NPC's current location to a specified X and Y position.
    - **Parameters**: `{"npc_id": int, "x": int, "y": int}`
    - **Output**: `{"distance": int, "time": int}`
        - Provides the distance to the target location and the estimated time for the NPC to reach it.

14. **`command.npc.Create`**
    - **Invoker**: Unity
    - **Functionality**: Creates a new NPC in the game.
    - **Parameters**: `{"asset": str, "model": str, "memorySystem": str, "planSystem": str, "nickname": str, "bio": str, "goal": str, "cash": int}`
    - **Output**: `{"NPCID": int, "status": "created"}`
        - Returns the new NPC's ID and confirms the creation was successful.

15. **`command.npc.GetNPCInfo`**
    - **Invoker**: Python
    - **Functionality**: Retrieves detailed information about a specific NPC.
    - **Parameters**: `{"NPCID": int}`
    - **Output**: `{"npc_name": str, "bio": str, "status": {}}`
        - Returns the NPC's name, bio, and current status (health, location, etc.).

16. **`command.npc.GetNPCs`**
    - **Invoker**: Python
    - **Functionality**: Retrieves a list of all NPCs currently in the game.
    - **Parameters**: None
    - **Output**: `{"npcs": [{"NPCID": int, "npc_name": str}]}`
        - Returns a list of all NPCs with their IDs and names.

17. **`command.player.GetPlayerInfo`**
    - **Invoker**: Python
    - **Functionality**: Retrieves information about the current player in the game.
    - **Parameters**: None
    - **Output**: `{"player_name": str, "player_stats": {}}`
        - Returns the player's name and current stats (e.g., health, location, etc.).

18. **`command.timetick.Tick`**
    - **Invoker**: Unity
    - **Functionality**: Sends a tick command to advance in-game time, influencing NPC actions and world events.
    - **Parameters**: None
    - **Output**: `{"status": "ticked"}`
        - Confirms the tick was processed successfully and that game time has advanced.

19. **`command.Event.ProximityEvent`**
    - **Invoker**: Unity
    - **Functionality**: Notifies the backend when NPCs are in close proximity, useful for monitoring interactions or engagement.
    - **Parameters**: `{"npc_a_id": int, "npc_b_id": int}`
    - **Output**: None
        - The backend processes this data to trigger potential actions or events based on NPC proximity.

20. **`command.Event.GlobalEvent`**
    - **Invoker**: Unity
    - **Functionality**: Sends global events affecting all NPCs, triggered by environmental changes or major announcements.
    - **Parameters**: `{"event_type": string, "details": object}`
    - **Output**: None
        - Ensures all NPCs are aware of global changes, influencing their behavior accordingly.

21. **`command.Event.StatusChangeEvent`**
    - **Invoker**: Unity
    - **Functionality**: Sends updates to the backend when an NPC's state changes (health, mood, environmental effects).

- **Parameters**: `{"npc_id": int, "status": string, "details": object}`
- **Output**: None
    - The backend tracks these changes to apply effects and trigger relevant events.