Table of Contents

1 Run Module

Sistema Multi-Agent Interactiu per a Minecraft Manté el sistema obert i respon a comandes de xat.

1.1 `main()`

Inicialitza el sistema i manté un bucle esperant comandes.

1.1 `safe_mc_post(mc, lock, message)`

Funció auxiliar per publicar de manera segura al xat utilitzant un bloqueig.

2 Agents

2.1 Agents

Aquesta secció documenta els aggents disponibles a `MyAdventures`.

- Base Agent
- Builder Bot
- Explorer Bot
- Miner Bot

2.2 Base Agent

2.2.1 `BaseAgent`

Bases: `ABC`

Classe base abstracta per a tots els agents.

`act()` `abstractmethod`

execució d'accions

`decide()` `abstractmethod`

presa de decisions

`handle_command(command, args)`

Gestiona les comandes de control (pausa, reprendre, aturar, actualitzar).

`pause()`

Pausa l'agent.

`perceive()` `abstractmethod`

percepció de l'entorn

`restore_checkpoint()`

Restaura l'estat guardat al checkpoint anterior.

`resume()`

Repren l'agent.

`run_once()`

Executa un sol cicle percepció-decisió-acció si l'estat és RUNNING.

`set_state(new_state, reason='')`

Canvia l'estat de l'agent amb registre de transició.

`start_loop(tick_interval=0.2)`

Inicia un fil que executa `tick` periòdicament. (0,2s)

`stop_loop()`

Atura el fil de l'agent i espera la seva finalització.

2.3 Builder Bot

2.3.1 `BuilderBot`

Bases: `BaseAgent`

Agent que construeix qualsevol dels planols generats.

`act()`

Executa la construcció si l'estat és RUNNING.

`cycle_plan()`

Rota al següent pla disponible.

`decide()`

Decisió

`on_message(msg)`

Gestiona missatges rebuts.

`perceive()`

Percepció

`reset()`

Reseteja l'estat del BuilderBot per a un nou workflow.

`start()`

Inicia el bot (en aquest cas, simplement el posa a IDLE esperant un mapa).

`stop()`

Atura el bot.

`switch_plan(plan_name)`

Canvia el pla de construcció actiu.

2.4 Explorer Bot

2.4.1 `ExplorerBot`

Bases: `BaseAgent`

Agent que descobreix zones planes del terreny prop del jugador.

`act()`

Marca la zona i publica el mapa.

`cycle_range()`

Cicle al següent rang d exploració.

`decide()`

Selecciona la zona objectiu si està en mode RUNNING.

`on_message(message)`

Gestiona missatges rebuts pel bus.

`perceive()`

Analitza el terreny prop del jugador real si està en mode RUNNING.

`reset()`

Reseteja l'estat per a un nou workflow.

`set_range(range_val)`

Estableix un rang d'exploració específic.

`start()`

Inicia l'exploració.

`stop()`

Atura l'exploració.

2.5 Miner Bot

2.5.1 `MinerBot`

Bases: `BaseAgent`

Agent que mina blocs de terra i pedra, recolectant recursos.

`act()`

Executa la mineria si l'estat és RUNNING.

`cycle_strategy()`

Ciclem a la següent estratègia.

`decide()`

Decisió

`on_message(msg)`

Gestiona missatges rebuts.

`pause()`

Pausa el MinerBot i la seva estratègia actual.

`perceive()`

Percepció

`reset()`

Reseteja l'estat del MinerBot.

`resume()`

Repren el MinerBot i la seva estratègia actual.

`set_strategy(index)`

S'estableix l'estratègia segons l'índex.

`start()`

Inicia la mineria si hi ha requeriments.

`stop()`

Atura la mineria.

`switch_strategy_by_name(name)`

Canvia l'estratègia buscant-la pel nom de la classe.

3 Strategies

3.1 Estratègies

Aquesta secció documenta les estratègies i els plans de construcció disponibles a `MyAdventures`.

- Estratègia Base
- Grid Search
- Vertical Search
- Plans de construcció

3.2 Strategy Base

3.2.1 `MiningStrategy`

Bases: `ABC`

Classe base per a totes les estratègies de mineria.

Gestiona operacions comunes de mineria amb suport per a gestió d'inventari, seguiment del progrés i comandes de control (pausa, reprendre, aturar).

`__init__()`

Inicialitza l'estratègia amb seguiment d'estat.

`get_name()`

Retorna el nom de l'estratègia.

`get_status()`

Obtenir estatus actual de mineria.

Returns:

Name	Type	Description
`dict`	`Dict`	Informació d'estatus incloent posició, materials col·lectats, blocs minats

`handle_pause()`

Pausar operacions de mineria.

`handle_resume()`

Reprendre operacions de mineria.

`handle_stop()`

Aturar operacions de mineria.

`mine(mc, start_pos, inventory, requirements=None, mc_lock=None)` `abstractmethod`

Executar estratègia de mineria.

Parameters:

Name	Type	Description	Default
`mc`		Instància de Minecraft	*required*
`start_pos`	`Tuple[int, int, int]`	Posició inicial (x, y, z)	*required*
`inventory`	`Dict`	Diccionari d'inventari actual amb requeriments	*required*
`mc_lock`		Lock per sincronitzar accés a mc (opcional)	`None`

Returns:

Name	Type	Description
`dict`	`Dict`	Materials col·lectats {material: quantitat}

`mine_block(mc, position, block_type, inventory, requirements=None, mc_lock=None)`

Minar un bloc única i afegir materials a l'inventari.

Parameters:

Name	Type	Description	Default
`mc`		Instància de Minecraft	*required*
`position`	`Tuple[int, int, int]`	Posició del bloc (x, y, z)	*required*
`block_type`	`str`	Tipus de bloc a minar	*required*
`inventory`	`Dict`	Inventari actual	*required*
`mc_lock`		Threading lock per sincronitzar accés al socket	`None`

Returns:

Name	Type	Description
`dict`	`Dict`	Materials extrets d'aquest bloc

`reset()`

Resetejar estat de l'estratègia.

`update_inventory(inventory, collected)`

Actualitzar inventari amb materials col·lectats.

Parameters:

Name	Type	Description	Default
`inventory`	`Dict`	Inventari actual	*required*
`collected`	`Dict`	Materials col·lectats en aquesta operació	*required*

Returns:

Name	Type	Description
`dict`	`Dict`	Inventari actualitzat

`validate_requirements(inventory, requirements)`

Validar si l'inventari actual compleix amb els requeriments.

Parameters:

Name	Type	Description	Default
`inventory`	`Dict`	Inventari actual	*required*
`requirements`	`Dict`	Diccionari de materials requerits	*required*

Returns:

Name	Type	Description
`bool`	`bool`	Cert si tots els requeriments es compleixen

3.3 Grid Search

3.3.1 `GridSearchStrategy`

Bases: `MiningStrategy`

Explora una regió cúbica seguint un patró de graella estructurat per a cobertura uniforme.

Aquesta estratègia mina blocs de forma sistàtica en una graella cúbica, movent-se per la regió amb un patró d'espaiament regular per assegurar extracció integral de recursos.

`__init__(grid_size=4)`

Inicialitzar estratègia de cerca en graella.

Parameters:

Name	Type	Description	Default
`grid_size`	`int`	Mida de la regió cúbica a minar (per defecte: 4 blocs)	`4`

`mine(mc=None, start_pos=None, inventory=None, requirements=None, mc_lock=None)`

Executar mineria basada en graella a través d'una regió cúbica.

Parameters:

Name	Type	Description	Default
`mc`		Instància de Minecraft	`None`
`start_pos`	`Tuple[int, int, int]`	Posició inicial (x, y, z)	`None`
`inventory`	`Dict`	Diccionari d'inventari actual amb requeriments	`None`
`mc_lock`		Lock per sincronitzar accés a mc (opcional)	`None`

Returns:

Name	Type	Description
`dict`	`Dict`	Materials col·lectats {material: quantitat}

3.4 Vertical Search

3.4.1 `VerticalSearchStrategy`

Bases: `MiningStrategy`

Des del punt d'anchor, mina verticalment cap avall fins a la profunditat màxima.

`__init__()`

Inicialitzar estratègia de mina vertical.

`mine(mc=None, start_pos=None, inventory=None, requirements=None, mc_lock=None)`

Executar mineria de perforació vertical cap avall a través de capes.

Parameters:

Name	Type	Description	Default
`mc`		Instància de Minecraft	`None`
`start_pos`	`Tuple[int, int, int]`	Posició inicial (x, y, z)	`None`
`inventory`	`Dict`	Diccionari d'inventari actual amb requeriments	`None`

Returns:

Name	Type	Description
`dict`	`Dict`	Materials col·lectats {material: quantitat}

3.5 Plans de construcció

Aquesta secció documenta els plans de construcció disponibles per al Builder Bot.

- Base Plan
- Castell
- Chess
- Plataforma

4 Utils

4.1 Utils

Aquesta secció documenta els mòduls `utils` disponibles a `MyAdventures`.

- Comandes de xat
- Comunicació
- Descobriment
- Funcional
- Configuració de registre
- Validadors
- Visuals

4.2 Chat Commands

4.2.1 `ChatCommand`

Representa una ordre analitzada del xat de Minecraft.

4.2.1 `ChatCommandHandler`

Controlador per analitzar i executar ordres del xat.

`handle_command(message)`

Executa una ordre si existeix un controlador per a ella.

`parse_command(message)`

Analitza un missatge i extreu l'ordre i els seus arguments.

`register(command, handler)`

Registra un controlador per a una ordre.

4.2.1 `create_default_handlers(agents_dict, mc, mc_lock=None, system_flags=None)`

Crea els gestors de comandes per defecte.

Parameters:

Name	Type	Description	Default
`agents_dict`		Diccionari d'agents	*required*
`mc`		Instància de Minecraft	*required*
`mc_lock`		Lock per sincronitzar accés al socket de Minecraft	`None`

4.3 Communication

4.3.1 `MessageBus`

Bus de missatges asíncron (Producer-Consumer) amb cues i validació Implementa: - processament asíncron non-blocking (amb Queue) - validació de missatges - traçabilitat completa

`__init__()`

Inicialitza el bus amb una cua i un thread de treball en background.

`publish(msg)`

Envia un missatge a la cua de processament (no bloquejant).

Realitza validació i garanteix traçabilitat abans d'encuar. L'emissor recupera el control immediatament després d'encuar (asíncron).

Parameters:

Name	Type	Description	Default
`msg`	`dict`	Missatge a enviar.	*required*

`stop()`

Atura el bus.

`subscribe(callback)`

Afegeix un callback a la llista de subscriptors.

Parameters:

Name	Type	Description	Default
`callback`	`callable`	Funció a cridar quan es rep un missatge.	*required*

4.3.1 `MessageProtocol`

Protocol per a la creació i validació de missatges JSON

`create_message(msg_type, source, target, payload, status='SUCCESS', context=None)`

`staticmethod`

Crea un missatge estructurat seguint el protocol definit.

Parameters:

Name	Type	Description	Default
`msg_type`	`str`	Tipus de missatge.	*required*
`source`	`str`	Origen del missatge.	*required*
`target`	`str`	Destí del missatge.	*required*
`payload`	`dict`	Dades del missatge.	*required*
`status`	`(str, opcional)`	Estat del missatge. Per defecte "SUCCESS".	`'SUCCESS'`
`context`	`(dict, opcional)`	Context addicional. Per defecte None.	`None`

Returns:

Name	Type	Description
`dict`	`dict`	Diccionari amb l'estructura del missatge.

`validate_message(msg)` `staticmethod`

Valida que un missatge contingui tots els camps requerits.

Parameters:

Name	Type	Description	Default
`msg`	`dict`	Missatge a validar.	*required*

Returns:

Name	Type	Description
`bool`	`bool`	True si el missatge és vàlid, False sino.

4.4 Discovery

Mòdul de descobriment reflexiu per a registre automàtic d'agents i estratègies. Utilitza les capacitats de reflexió de Python per descobrir i carregar mòduls dinàmicament.

4.4.1 `discover_agents()`

Descobreix totes les classes d'agent.

Returns:

Name	Type	Description
`dict`		Mapa d'agents

4.4.1 `discover_build_plans()`

Descobreix tots els plans

Returns:

Name	Type	Description
`dict`		Mapa de plans

4.4.1 `discover_classes(package_name, base_class)`

Descobreix i registra automàticament classes que hereten de base_class.

Parameters:

Name	Type	Description	Default
`package_name`		Nom del paquet a escannejar	*required*
`base_class`		Classe base que han d'heretar les classes descobertes	*required*

Returns:

Name	Type	Description
`dict`		Mapa de noms de classes a objectes de classe

4.4.1 `discover_strategies()`

Descobreix totes les estratègies

Returns:

Name	Type	Description
`dict`		Mapa d'estratègies

4.5 Functional Utils

Utilitats de programació funcional per a l'anàlisi de logs.

4.5.1 `count_logs_by_level(logs)`

Compta quants logs hi ha de cada nivell usant reduce.

4.5.1 `filter_logs(logs, **criteria)`

Filtra els logs basant-se en criteris clau-valor. Exemple: filter_logs(logs, level="ERROR", logger="MinerBot")

4.5.1 `get_agent_activity(logs)`

Analitza l'activitat per logger amb reduce.

4.5.1 `load_logs(file_path)`

Generador que llegeix logs d'un fitxer línia per línia (lazy). Això és eficient ja que podem tenir molts logs.

4.5.1 `parse_log_line(line)`

Parseja una línia de log en format JSON. Gestionem errors amb try-except.

4.6 Logging Config

4.6.1 `StructuredFormatter`

Bases: `Formatter`

Formatter que produeix logs estructurats en JSON.

4.6.1 `setup_logging()`

Configura logging estructurat per a tots els agents i sistema.

4.7 Validators

4.7.1 `es_fila_valida(row)`

Valida si una fila de CSV conté les dades necessàries per definir un bloc. Requereix: dx, dy, dz (enters) i material (string no "").

4.8 Visuals

4.8.1 `mark_bot(mc, x, y, z, wool_color=11, label=None)`

Col·locar un bloc de llana de color a la ubicació donada. Si s'especifica el label, també es publica un missatge al xat.