Minecraft Agent Framework

The goal of this task is to implement a Python framework enabling the development and execution of Python coded agents in a Shared Minecraft server. The agent should be able to move, interact with the environment, build and destroy blocks, and send or receive messages from the chat.

Create examples like InsultBot (insulting in the chat and bothering users), TNTBot (an agent deploying TNT and exploding it) or an OracleBot (answering questions in the chat using a list of answers or a LLM like ChatGPT).

Demonstrate the use of funcional programming in Python with different examples.

Demonstrate the use of reflective programming in Python with different examples.

Create a github repository including documentation, examples and unit tests.

Include a github badge (github actions) demonstrating the code coverage of the code in your repository.

Deliver all code and documentation in a self-contained zip file in moodle. Share with professor in the interview your github repository to demonstrate codecov badge.

Optional:

1. Use the Pyro library to enable a remote client to interact with the server. https://pyro4.readthedocs.io/en/stable/
2. Using your framework, create a ChatBot agent that enables interaction with ChatGPT using remote APIs. You will need to include a VISA card. Never include login and pasword information in your github repositories.
3. Create a Minecraft world with a part of Tarragona (city) and propose educational exercises or games on top of your framework related to Tarragona. This task is designed for Tarragona schools (primary and secundary) so it must be relatively simple.

Minecraft server configuration

The server that will be used for the practice is available directly from the Adventures in Minecraft GitHub.

https://github.com/AdventuresInMinecraft/AdventuresInMinecraft-Linux

> Please note that there are different versions depending on the operating system we want to use (Windows, Mac, and Linux).

Regardless of the OS being used, the project will always have the following structure:

As can be seen, there is a file called *StartServer*, which will have the extension *.bat*, *.sh*, or *.command* for Windows, Linux, and Mac, respectively. If this file is executed, the server will start running with the default memory requirements (1024 MB).

If this memory is not enough to perform the necessary tasks, it can be modified directly in the *start.{bat|sh|command}* file, located in the */Server* directory.

By default, the following command is present:

```
1  java -Xmx1024M -jar -DIReallyKnowWhatIAmDoingISwear craftbukkit.jar
```

The previous command can be modified to increase the memory available for the server. In the example below, the memory is increased to 1 GB.

```
1  java -Xms1G -Xmx1G -jar craftbukkit.jar
```

If it becomes necessary to modify some of the server's properties, this can be done directly from */Server/server.properties*.

Once the command to start the server is executed, we need to wait a few seconds, and once it is fully loaded, we should have something similar to what is shown in the following image.

```
*** Error, this build is outdated ***
*** Please download a new build as per instructions from https://www.spigotmc.org/ ***
*** Server will start in 15 seconds ***
Loading libraries, please wait...
[18:37:35 INFO]: Starting minecraft server version 1.12
[18:37:35 INFO]: Loading properties
[18:37:35 INFO]: Default game type: CREATIVE
[18:37:35 INFO]: Generating keypair
[18:37:35 INFO]: Starting Minecraft server on *:25565
[18:37:35 INFO]: Using default channel type
[18:37:35 INFO]: This server is running CraftBukkit version git-Bukkit-ed8c725 (MC: 1.12) (Implementing API version 1.12
-R0.1-SNAPSHOT)
[18:37:35 INFO]: [RaspberryJuice] Loading RaspberryJuice v1.10
[18:37:35 WARN]: **** SERVER IS RUNNING IN OFFLINE/INSECURE MODE!
[18:37:35 WARN]: The server will make no attempt to authenticate usernames. Beware.
[18:37:35 WARN]: While this makes the game possible to play without internet access, it also opens up the ability for ha
ckers to connect with any username they choose.
[18:37:35 WARN]: To change this, set "online-mode" to "true" in the server.properties file.
[18:37:36 INFO]: Preparing level "world"
[18:37:36 INFO]: Preparing start region for level 0 (Seed: 837832613987290238)
[18:37:36 INFO]: Preparing start region for level 1 (Seed: 837832613987290238)
[18:37:37 INFO]: Preparing start region for level 2 (Seed: 837832613987290238)
[18:37:37 INFO]: [RaspberryJuice] Enabling RaspberryJuice v1.10
[18:37:37 INFO]: [RaspberryJuice] Using port 4711
[18:37:37 INFO]: [RaspberryJuice] Using ABSOLUTE locations
[18:37:37 INFO]: [RaspberryJuice] Using BOTH clicks for hits
[18:37:37 INFO]: [RaspberryJuice] ThreadListener Started
[18:37:37 INFO]: Server permissions file permissions.yml is empty, ignoring it
[18:37:37 INFO]: Done (1,615s)! For help, type "help" or "?"
>
```

The server can also be started directly from a terminal. While positioned in the */Server* directory, we can execute the second command to start the server.

CONNECT TO THE SERVER

Since the server has been opened on our machine, we can access it directly by entering "localhost" in the “server address” field when creating it in the game's "MultiPlayer" tab.

MODIFY THE WORLD FROM PYTHON

To interact with the Minecraft world, the AdventuresInMinecraft library contains various API libraries with methods that allow us to perform different actions. The only one that needs to be included in the project is *mcpi*, which can be found in */AdventuresInMinecraft/MyAdventures/mcpi*. The other folders are additional API libraries that allow interaction with the game via Arduino, RaspberryPi, MicroBit, etc.

Below is an example of a small Python program that performs several actions in the game.

```
2  # Import necessary modules
3  import mcpi.minecraft as Minecraft
4  import mcpi.block as block
5
6  # Connect to the Minecraft game
7  mc = minecraft.Minecraft.create()
8
9  # Interact with the Minecraft world
10 mc.postToChat("Hello Minecraft World")
11 pos = mc.player.getTilePos()
12 mc.setBlock(pos.x+3, pos.y, pos.z, block.STONE.id)
```

References

Adventures in Minecraft

https://github.com/AdventuresInMinecraft/AdventuresInMinecraft-Linux

Functional Python programming

https://realpython.com/python-functional-programming/

https://docs.python.org/3/howto/functional.html

Reflective Programming in Python

https://dev.to/bshadmehr/series/23935

https://gamedevacademy.org/python-reflection-tutorial-complete-guide/

Codecov

https://docs.codecov.com/docs/flags https://docs.codecov.com/docs/status-badges

https://about.codecov.io/ https://docs.codecov.com/docs/quick-start

TLauncher (Free Minecratf)

https://tlauncher.org/en/