

Instructions

Download required programs/packages:

1. Download python from <https://www.python.org/downloads/>
 - My version used was 3.9.1 (recommended)
2. Download pip for python using the following command on a terminal:
 - `python -m pip install pip`
3. Use pip to install required packages from requirements.txt:
 - `pip install -r requirements.txt`
 - or
 - `python -m pip install -r requirements.txt`
4. For any issues that may arise with versions of the packages to be installed you can use the following command to install them one by one and with the versions you want (replace "numpy" with the intended package):
 - `pip install numpy`
 - or
 - `python -m pip install numpy`

Note: you can choose what version to install by writing `package==version`. For example, `numpy==1.22.4`

5. The needed packages are:
 - numpy
 - onnx
 - onnx-tf
 - opencv-python
 - Pillow

Files needed:

1. Server (Unity3D app)
 - a. server.py
2. Client (Used by students)
 - a. client.py
 - b. gnet15D.onnx

Files placement and Unity changes:

- Place the server file (server.py) in the Unity project under the following directory. (Where the text files were, and where the new ones will be created)

\3D AVATARS\Assets\Resources

- Make the following changes inside *Student.cs* file in Unity

```
// Start is called before the first frame update
void Start()
{
    myAnimations = new List<int>();
    // Changed student IDs to match how many clients I had running
    readTextFile("actions_" + studentId);
    //readTextFile("actions");
    animationsIndex = 0;
    playAnimation(animationsIndex);
}

private void Update()
{
    if (animLabel.text == "")
    {
        Debug.Log("Animation ended, reading text file again");
        myAnimations = new List<int>();
        //readTextFile("actions");
        readTextFile("actions_" + studentId);
        animationsIndex = 0;
        playAnimation(animationsIndex);
    }
}
```

- I changed the text files names so not to change anything else about the previous version of the application.
- I added the *Update()* function for continuous reading of the text files, as soon as the current animation ends.
- NOTE: The number (ID) of each avatar in Unity has to match an existing file. If the clients running at a given moment are less than the avatars present, then you should assign the IDs accordingly. (Two avatars to have the same ID for example, or remove an avatar)

Run programs:

1. To run python files:
 - *python server.py*
 - *python client.py*
2. Run first the server side and wait for it to start listening and then the client side
3. The client:
 - It will first ask you to choose if you want to run it as a standalone program or along with the server, by typing "a" or "s" respectively
 - If you choose to run it with the server, it will then ask you for an ID to be used for the specific avatar.
 - NOTE: No two clients must have the same ID