nnn-nn-Unity-Python-CAPP-1-v1-YY-2022-3-3

CTI One Corporation

Version: 1.0
Date: March 3, 2022
Project Lead: Harry Li, Ph.D.
Members: Yusuke Yakuwa,

Left-Handed Coordinate System

The coordinate system in Unity is Left-Handed Coordinate system.

x: Right side direction
y: Height
z: Forward direction

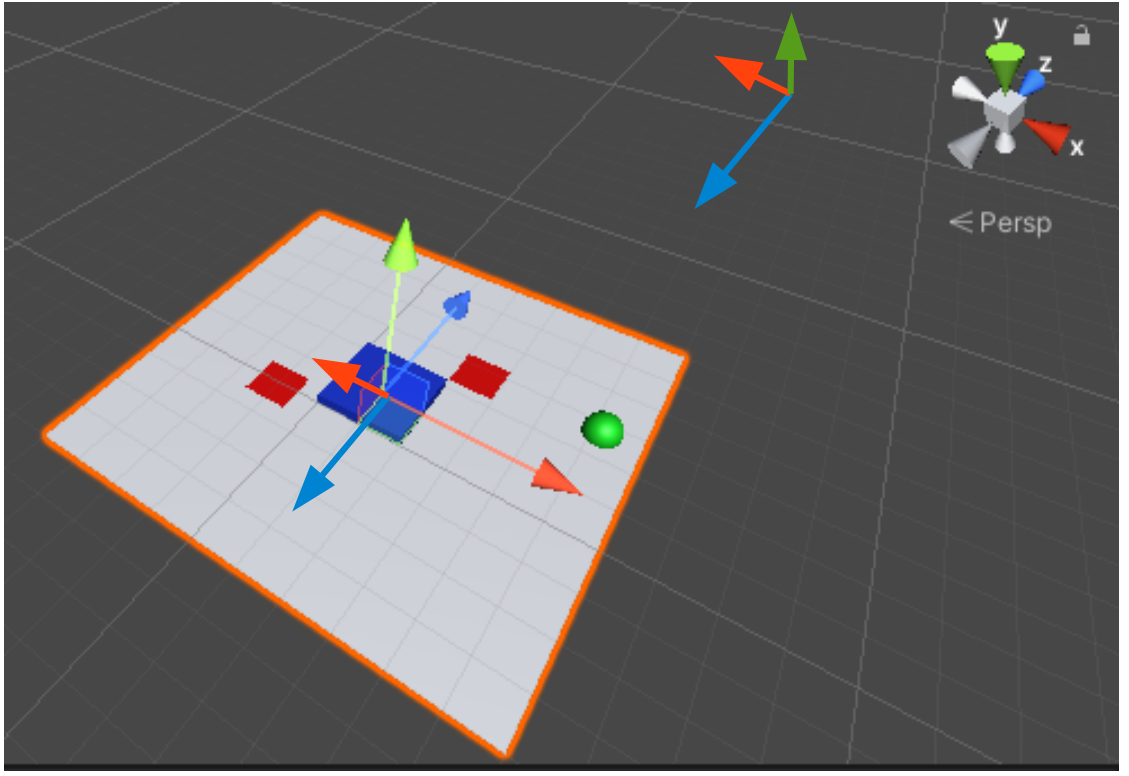


Fig 1. Unity Coordinate System

Right-Handed VS Left-Handed Coordinate System

CTIOne AGV2000 and AGV4000

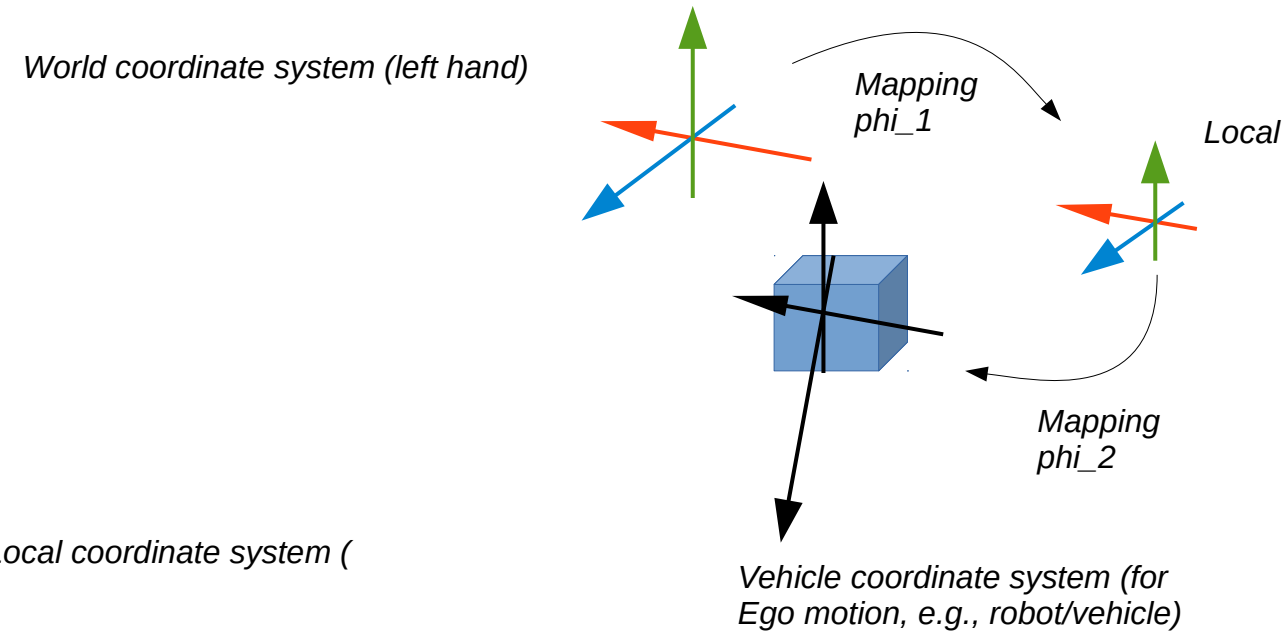
CTIOne BBD Robot

https://www.techarthub.com/a-guide-to-unitys-coordinate-system-with-practical-examples/#:~:text=Unity%20uses%20a%20left%2Dhanded,unexpected%20results%20for%20the%20uninitiated.

World vs Local Coordinate System In Unity

World Coordinate System: **Coordinate based on Unity world**. Point (0, 0, 0) is the center of the Unity world.

Local Coordinate System: **Coordinate based on the parent object**. Point (0, 0, 0) is the center of the parent object. If the objects do not have its parent object, the objects belong to the world, in this case, world coordinator and local coordinator are same.



Object Position

“LocalArea” object (“Floor” whose center is):
1. No parent object
2. In world coordinate system: (X=10,Y= 0, Z=10)
3. In local coordinate system: (X=10,Y= 0, Z=10)

“Point_A” object (Red Cube in Fig 1.):
1. Parent object is “LocalArea” object
2. In world coordinate system: (X=10,Y= 0, Z=10)
3. In local coordinate system: (X=0,Y= 0, Z=0)

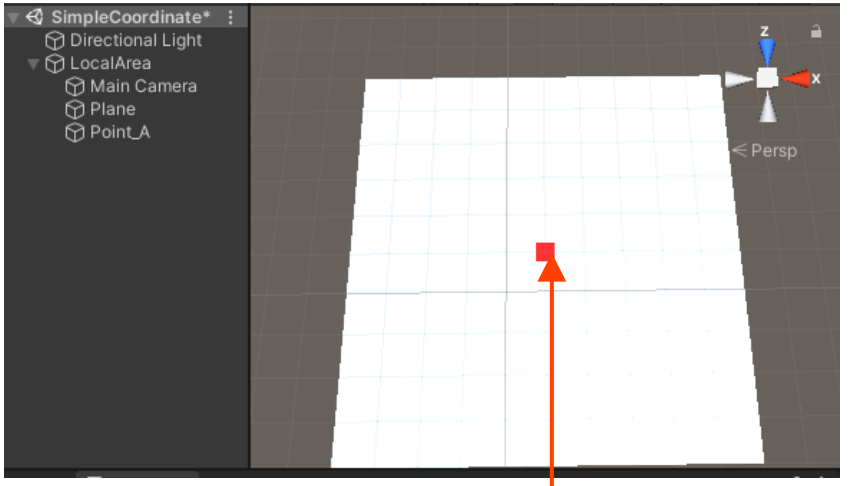


Fig 2. Object Position

In world coordinate system: (X=10,Y= 0, Z=10)
In local coordinate system: (X=0,Y= 0, Z=0)

C# Code For Displaying The Coordinator

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SimpleCoordinate : MonoBehaviour
{
    GameObject localArea;
    GameObject point_a;

    void Start()
    {
        localArea = GameObject.Find("LocalArea");
        point_a = GameObject.Find("Point_A");

        Debug.Log("LocalArea World Cordinate: " + localArea.transform.position.ToString());
        Debug.Log("LocalArea Local Cordinate: " + localArea.transform.localPosition.ToString());

        Debug.Log("Point A World Cordinate: " + point_a.transform.position.ToString());
        Debug.Log("Point A Local Cordinate: " + point_a.transform.localPosition.ToString());

    }
}
```

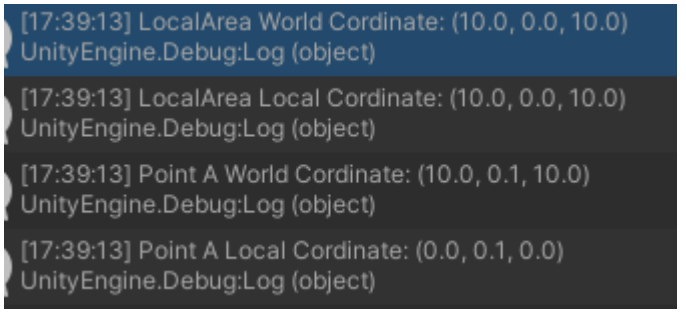


Fig 3. Result of displaing the coodinator

C# Code For Moving Object In Local Coodinate System

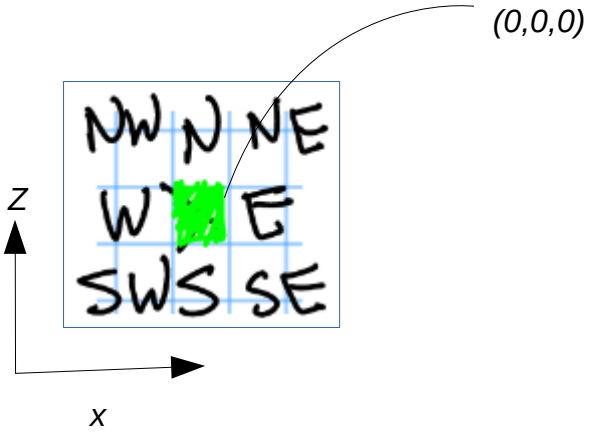
```
machine = GameObject.Find("Machine");

// To North
machine.transform.localPosition += new Vector3(0, 0, 1);

// To South
machine.transform.localPosition += new Vector3(0, 0, -1);

// To East
machine.transform.localPosition += new Vector3(1, 0, 0);

// To West
machine.transform.localPosition += new Vector3(-1, 0, 0);
```



Local VS World Coordinate System For Moving Object

"LocalArea" : Parent object, Rotaion(X:0, Y:45, Z:0)

"Machine" : Child object of "Local Area"

Below Code A and Code B are same meaning

Code A (Local):
machine.transform.localPosition += new Vector3(0, 0, 1);

Code B (World):
machine.transform.position += new Vector3(0.7, 0, 0.7);

This is a translation, but please add rotation code sample. ???

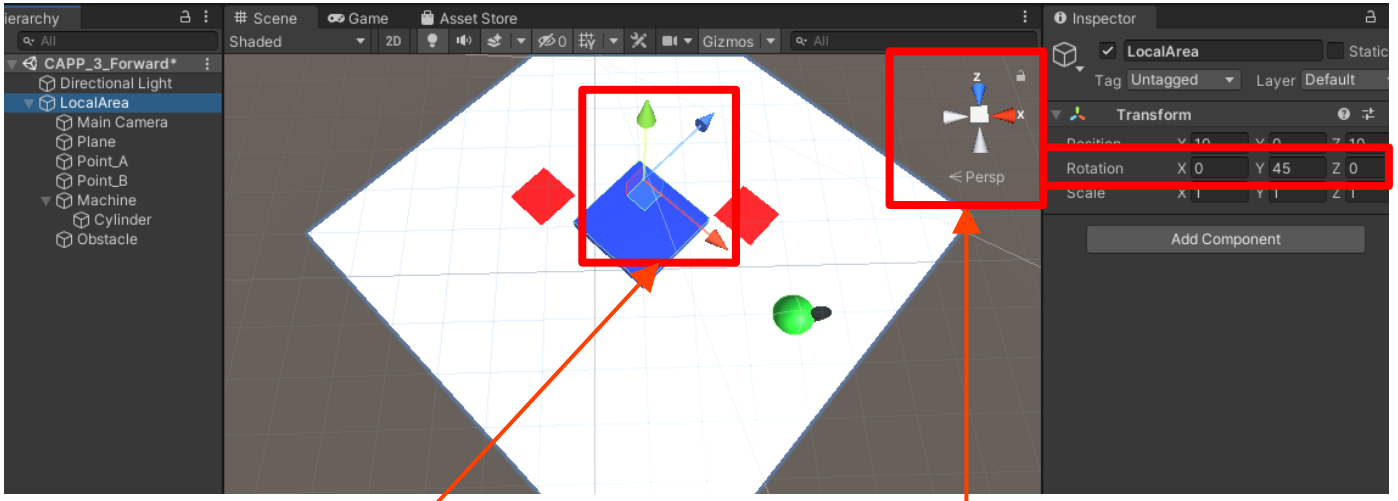
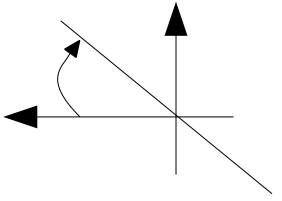


Fig 4. Axes of coordinate system

Axes of World Coordinate System

Axes of Local Coordinate System

1. w.r.t. positive x; and 2. clock wise direction



Normalized Vector (1)

https://docs.unity3d.com/ScriptReference/Transform.html

GameObject.Transform has properties normalized vectors: forward, right, up

forward: Z axis
right: X axis
up: Y axis

The normalized vectors are in world coordinate system.

If the object orientation is the same as axes of world coordinate system;
forward returns (x: 0, y: 0, z: 1)
right returns (x: 1, y: 0, z: 0)
up returns (x: 0, y: 1, z: 0)

If the object orientation rotates 45 degree from axes of world coordinate system;
forward returns (x: 0.7, y: 0, z: 0.7)
right returns (x: 0.7, y: 0, z: -0.7)
up returns (x: 0, y: 1, z: 0)

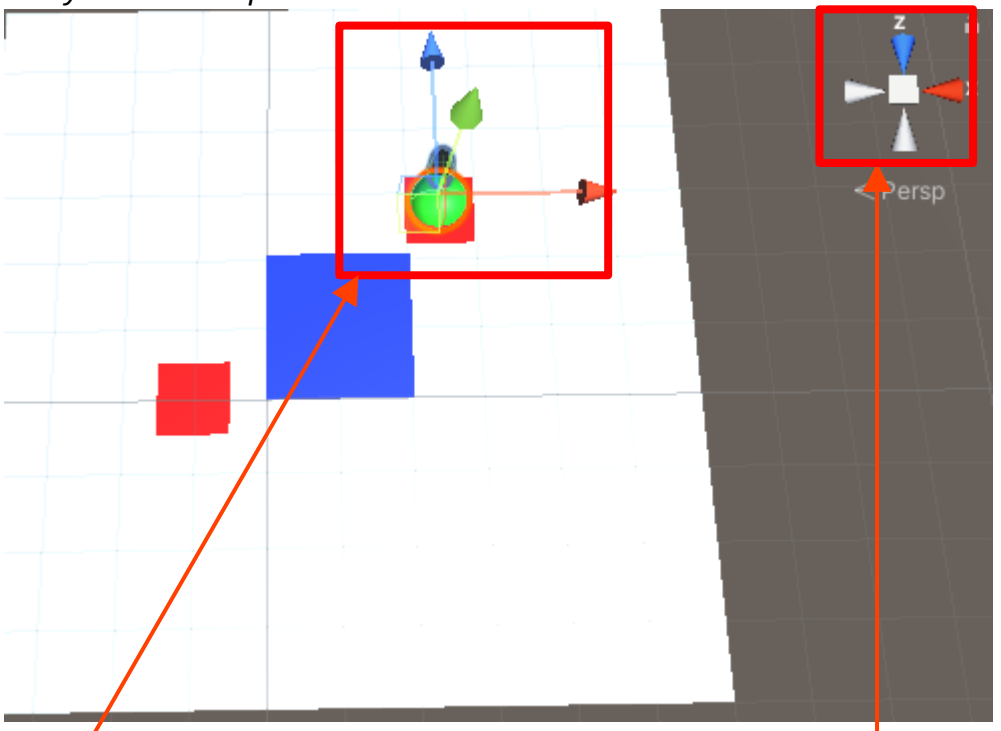


Fig 5. Object's Orientaion

Object's Axes

Axes of World Coordinate System

C# Code For Displaying Normalized Vector

```
Debug.Log("Machine Forward Coordinator: " + machine.transform.forward.ToString());
Debug.Log("Machine Right Coordinator: " + machine.transform.right.ToString());
Debug.Log("Machine Up Coordinator: " + machine.transform.up.ToString());
```

END