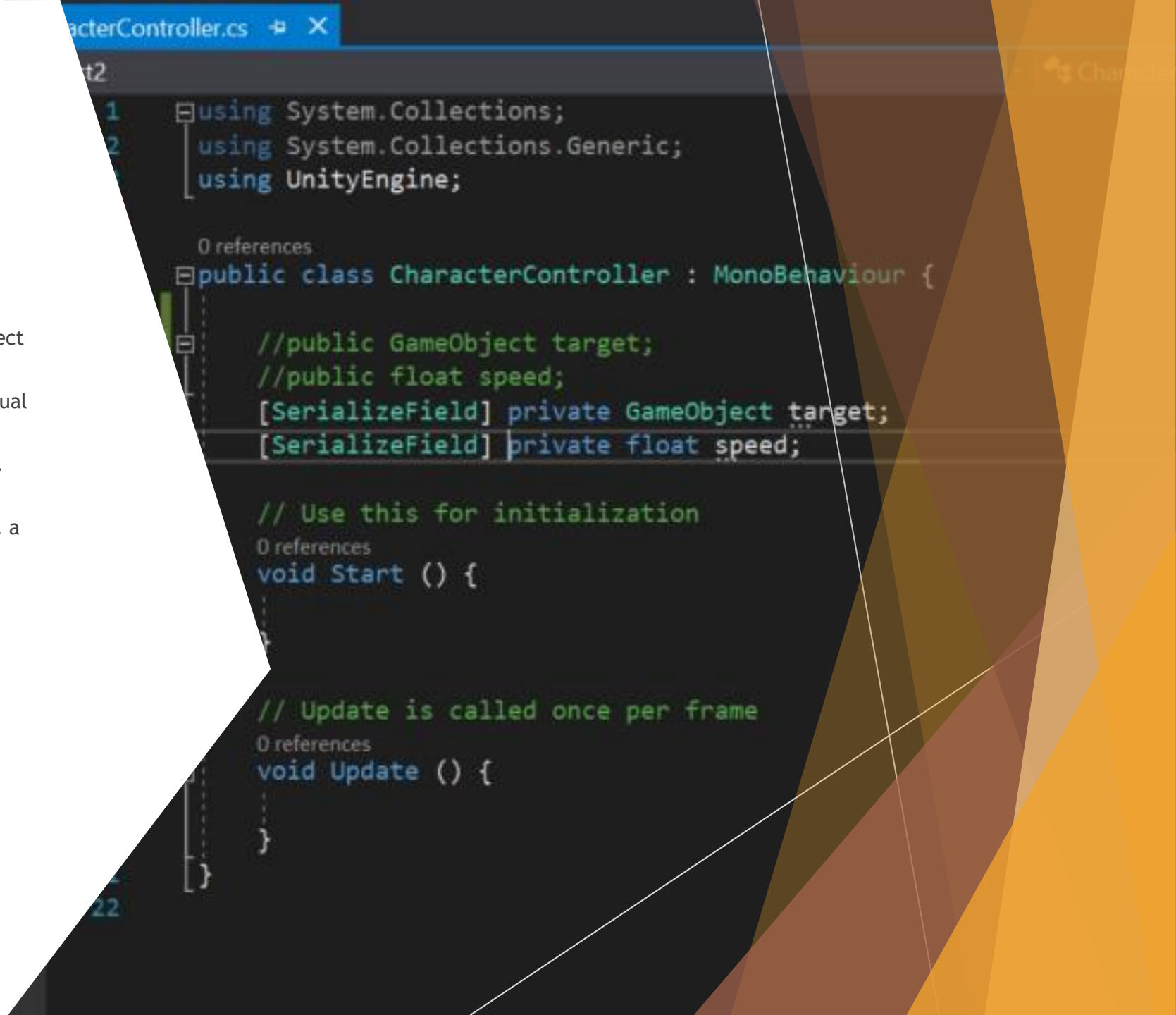
Unity 3D

Scripting

Script Basics

- Right click and create a new C# script in the project panel. Open it.
- You can setup Unity to edit scripts either with Visual Studio “or MonoDevelop (included with Unity)”.
- Check the two functions already declared for you. What do they do?
- Attach your script to an object in your scene (e.g. a cube).
  - 1. You can drag it on the object or
  - 2. You can add it from the inspector (Add Component) after selecting the object.
- Add a public variable to your script of type GameObject. Name it Target.
- Add another public float variable and name it speed.
- Note that public variables appear in the object Inspector and can be edited from there.



Transforms

- Add the following code to your update and run it.
  - Vector3 direction = new Vector3(0.01f,0,0);
  - this.transform.Translate(direction);
- Rotate your object a little on the Y axis and rerun.
- What is the difference between the code from 1. and this:
  - this.transform.Translate(direction, Space.World);
- Instead of 0.001f, use the speed variable. Try changing it while the program is running.
  - What happens?
  - Why is this useful?
- Set the same speed with your classmates near you.
  - Do your objects move at the same speed when you run ? (check with someone using a laptop!).
  - Can you explain this?

- Try this:
  - this.transform.Translate(direction *Time.deltaTime, Space.World);
  - What is deltaTime?
  - Precisely how fast is your object moving now?
- Make the object rotate around its Z axis while its moving.
- Setup a target
  - Add a second object in the scene somewhere away from your main object.
  - Select your main object (the one with the script attached) and set it's target variable to be the new object (you can also drag the object on there).
  - Change the movement direction to be towards the target (you can access the target's position using target.transform.position).
  - Try moving the target around during runtime.

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

0 references
public class CharacterController : MonoBehaviour {

    //public GameObject target;
    //public float speed;
    [SerializeField] private GameObject target;
    [SerializeField] private float speed;

    // Use this for initialization
    0 references
    void Start () {

    }

    // Update is called once per frame
    0 references
    void Update () {

        if (Input.GetButtonDown("Fire1"))
        {
            Ray ray = Camera.main.ScreenPointToRay(
            RaycastHit hit;
            if (Physics.Raycast(ray,out hit))
            {
                Vector3 direction = hit.point;
                transform.Translate(direction * T
            }
        }

    }
}
```

Input

- Make a floor for your scene. You can use a cube and scale it in the x/z axis.
- Add this to your Update:

```
if (Input.GetButtonDown("Fire1"))
{
        Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);

        RaycastHit hit;

        if (Physics.Raycast(ray, out hit)) {

        }

}
```

- Finish the code so that it sets the target to the hit point and run it.
- Search the documentation for Input. Set a key press to rotate the object around its Y axis when the button is pressed. This should happen while the object is moving (e.g. using the mouse).