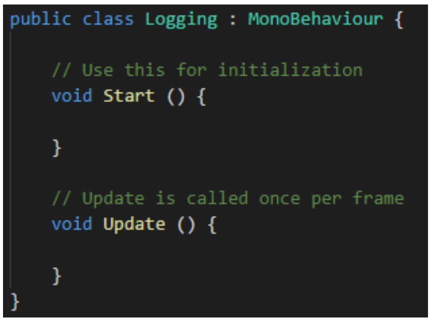
Exercises - Scripting

Exercise 1 - Logging

Attach a script to a GameObject that prints to the log whenever Start or Update is called on the component by the Unity Engine.



```
public class Logging : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

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

    }
}
```

Consider:

- *How can you attach a script to a GameObject?*
- *What is the relationship between the script and the GameObject?*
- *When is Start() and Update() called?*
- *What class do you use to log to the console? What methods can you use on that class?*
- *Is the console filled with update logs? Try pressing the “Collapse” button.*

Exercise 2 - Fireworks

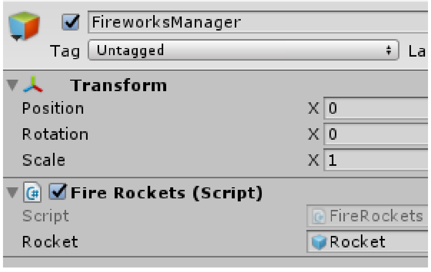
Write a script that moves GameObjects upwards by manipulating their transform component. After some time has elapsed, the script should instantiate an explosion prefab and destroy the GameObject it is attached to.

Consider:

- *How do you manipulate the position of a GameObject? Where should the code be put? Why?*
- *What is Time.deltaTime, and why do you need it for this exercise?*
- *How could you manipulate the speed of the rockets through the editor while in play mode?*
- *What does Vector3.up do compared to new Vector3(0,1,0)?*
- *Where in the code are you destroying the GameObject?*
- *Are you instantiating the explosion in the OnDestroy() event function? How do you use the Instantiate() method to instantiate the explosion at a certain position?*
- *You don’t have to create a fancy explosion - a sphere that destroys itself after some time is okay. If you’re feeling adventurous, however, you can use the ParticleSystem component, or import an explosion from the Asset Store.*

Exercise 3 - Fireworks Manager

Create an empty GameObject that can manage the firework rockets from exercise 2. It should be able to instantiate a rocket prefab at a random position when you push a button. It should also be possible to push another button to start or stop a fireworks show with multiple rockets.



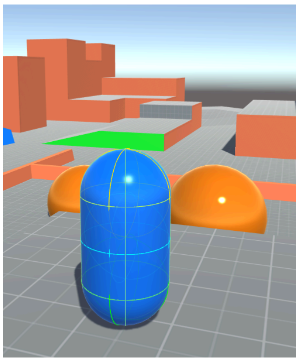
☑ FireworksManager
Tag Untagged La

Transform
Position X 0
Rotation X 0
Scale X 1

☑ Fire Rockets (Script)
Script FireRockets
Rocket Rocket

Consider:

- *How and where do you listen for input? How do you make sure that you don't instantiate more than one GameObject when you push a button?*
- *Do you use Random.Range as vector arguments to spawn rockets at random positions?*
- *Are you using InvokeRepeating() and CancelInvoke() for the fireworks show?*



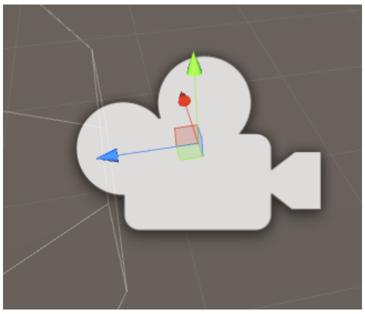
Exercise 4 - Character Controller

Create a basic character controller for a capsule. The cylinder should be able to move in four directions and it should be possible to change the speed of motion in the inspector.

Consider:

- *What is the difference between setting a variable to public and using the attribute [SerializeField] to expose variables in the editor?*
- *Is it possible to move in diagonal directions? What do you need to change in your code to enable or disable this?*
- *Why do you think floats are more used than doubles in game development?*

Exercise 5 - Dynamic Camera

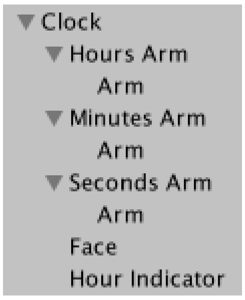


Write a script that makes the main camera follow the transform of the cylinder from exercise 4.

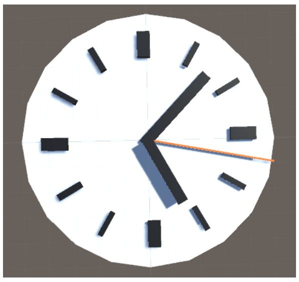
Consider:

- *Why can't we just make the camera a child of the cylinder?*
- *Have you tried selecting the camera in the scene hierarchy and pressing ctrl+shift+f to map the transform of the scene view camera to the transform of the game camera?*
- *How do you calculate and store the offset between the main camera and the cylinder?*

Exercise 6 - It’s Time To Create a Clock!



- ▼ Clock
  - ▼ Hours Arm
    - Arm
  - ▼ Minutes Arm
    - Arm
  - ▼ Seconds Arm
    - Arm
  - Face
  - Hour Indicator



Built a clock out of 3D objects. Write one script that rotates all the clock’s arms to show the current time.

Consider:

- *Parenting GameObjects is essential to completing this exercise since children are affected by the transform of its parent. Start out with an empty GameObject and use more of these whenever you want to create pivot points for rotations.*
- *How are you accessing the transform of the three clock arms inside one script?*
- *Are you using DateTime.Now from the System namespace to access the current time?*
- *How do you convert from e.g. hours to degrees?*
- *Are you using transform.localRotation = Quaternion.Euler() to rotate the arms?*

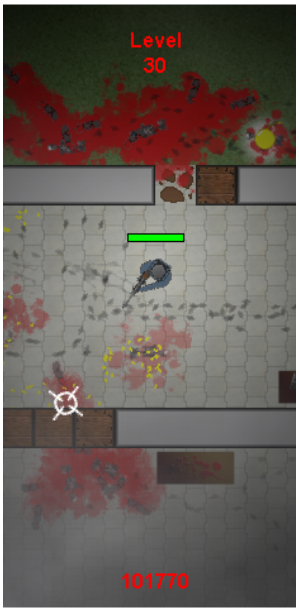
Exercise 7 - GTA Character Controller

Create a character controller similar to the 2D GTA games.

- “A” and “D” should rotate the character
- “W” should move the character forward
- “D” should move the character backwards
- The camera should follow the position of the player from the sky with a 90 degree fixed rotation on the x axis, and 0 degrees on the y and z axis.

Consider:

- *How do you rotate the GameObject?*
- *How do you translate GameObjects in local space compared to world space? What is the difference between the two?*



Exercise 8 - Top Down Shooter

Use everything you have learned to to create a top down shooter using the character controller from exercise 7. Here are some suggestions to what you can implement:

- A Shooting mechanic
- Random spawning enemies or NPCs
- Multiple weapon types with a weapon swap mechanic
- Ammo system and reloading
- Sprinting
- Pick-ups (Health, ammo, power-ups, etc.)
- An advanced camera following script
- Score counter, timer and enemy wave feature
- Vehicles

Consider:

- *Some features require detection of when GameObjects collide or overlap (e.g. shooting enemies or picking up health).*