Unity Game Scripting – Week 1 Lab Assignment

Before the lab:

- Install a Unity 2021.3.X version, with Visual Studio included.
- Follow the lecture, and/or look at the lecture notes/slides.
- Download and open the Week 1 Unity project from Blackboard.
- Open the Assignment1 scene.

During the lab:

- Work through the points below one by one.
- Keep the lecture notes/slides and relevant Unity documentation at hand!
- Don't stay stuck for too long on one task; ask your lab teacher for help!
- Even if you solved the tasks, ask for feedback from your lab teacher – you do not need to finish all tasks before doing this!
- Don't worry if you can only finish a small part of these tasks during the lab! (These tasks are created also with self-study time in mind.)

Lab assignment:

1. Create a building next to the "play arena". Use Unity primitives (mainly cubes) and create new materials (without textures). You can simply make a house with sloped roof, or you can be more creative, but challenge yourself by using rotation angles not divisible by 90, and using non uniform scaling for elements of the building. *(Tip:* use the scene editing tricks and shortcuts shown in the lecture! Ask your lab teacher for further feedback and tips.)
2. Quickly create a number of copies and variants of this building next to the arena. For some of them, rotate the entire building. *(Tip:* the easiest way to do this is temporarily add an empty parent object, rotate the parent object, and then unparent the building.)
3. Make the player turn left and right when using the mouse (use *GetAxis.* Think First Person shooter controls.)
4. Make the player move using keyboard input, relative to its current rotation. So pressing UP or W always moves the player forward in its current direction. Make sure you can set the speed in the inspector. *(Tip:* experiment with transform.Translate, adding a vector to transform.position, transform.forward and Vector3.forward. Experiment with GetAxis, GetKey and GetKeyDown, and choose the best solution.)
5. Also include "strafing": the player moves sideways when pressing LEFT / RIGHT or A / D.
6. Make the player constrained to the arena: add bounds in the inspector (minimum and maximum x and z values), and simply move the player back to where he was at the start of the frame if the player moves outside of the bounds.
7. Make the camera follow the player, at a fixed position relative to the player, such that the camera is always looking in the same direction as the player (=standard third person controls, or first person if you move the camera inside the player capsule).

8. Create an *EnemyChase* script, and add it as component to the *Enemy* game object. Make sure that the enemy is always looking at the player, and moves towards the player (=forward) at a fixed, slow speed. You can link the objects by dragging the player into the inspector for one of the components of the enemy. *(Tip:* compute the difference vector between the player and the enemy, and use this with Quaternion.LookRotation. Next, use transform.forward to move the enemy forward. You can *scale* this vector to change the enemy speed.)
9. When the enemy is close to the player: print “GAME OVER” to the console, and set the player speed to zero. *(Tip:* compute the difference vector between both objects, and compute its length using *magnitude.)*
10. Replace the enemy capsule by the *alien* model in the assets. Make sure the scale is similar to the player scale, and the enemy looks forward. *(Tip:* the easiest way is to make a new alien game object child of the enemy game object, and disable the capsule renderer.)