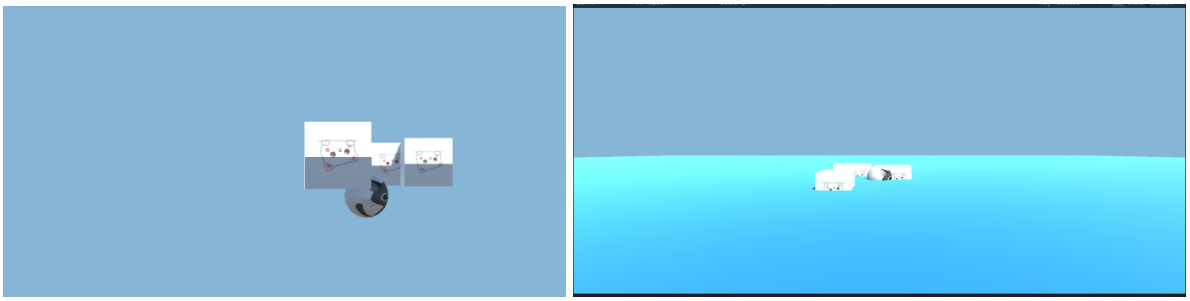

First 3D Scene

A plane was inserted into the scene with its x, y and z coordinates scaled to be 20 each in order to cover a large area of the ground. Additionally, a light blue colour was chosen for the plane to help give the scene a calming atmosphere. The objects in the scene use cat images as materials to further enhance the soothing nature of it. .

The player, which is represented by a cat textured sphere, was implemented using translation with a vector that either goes up, down, right or left depending on the user input [1].

The white textured cat cubes were also implemented using translation [2] with a direction vector that is either moved to the left or the right depending on its transform position [3]. A small random range is multiplied with the direction vector to make its movement change upon replaying the scene.

In order to populate the scene with more than one cube, an *Instantiate* function was placed in a for loop that uses a 3D vector to spawn the cubes at random points across the x and z coordinates [4]. The y coordinate of the vector was left to be 0 to avoid the cubes from obscuring the player game object.

```
// the function was created, so that the spawn object code could be reused for other objects
1 reference
private void SpawnObject(GameObject gameObject)
{
    /**
     * Code modified from (comments have been added accordingly)
     * https://answers.unity.com/questions/1623695/how-to-make-multiple-objects-spawn-at-random-order.html
     **/
    // the random range of the vector is kept small so that the user can see the cubes moving back and forth
    Instantiate(gameObject, new Vector3(Random.Range(-5f, 5f), 0, Random.Range(-5f, 5f)), transform.rotation);
}

Unity Message | 0 references
```

For the purposes of the extension, a *CameraController* script was created that follows the player character; its offset is created by subtracting the camera's transform position with that of the player's [5], which enables the camera to move depending on user input [1].

```
[SerializeField] GameObject player;
private Vector3 offset;
private float verticalOffset; // sets a vertical offset allows the user to see what is happening from a specific distance

Unity Message | 0 references
void Start()
{
    offset = (transform.position - player.transform.position);
    verticalOffset = Random.Range(1.5f, 2.0f); // make the vertical offset random, so that the camera is neither too close or far
}

Unity Message | 0 references
void LateUpdate()
{
    transform.position = (player.transform.position + offset) * verticalOffset; // multiply the transform position with the vertical offset
}
```

A vertical offset was added to the transform position in the *LateUpdate* method, so that the user has a better idea of how the camera is following the player object.

References

[1] https://www.c-sharpcorner.com/article/transforming-objects-movement-using-c-sharp-scripts-in-unity/

[2] https://gamedevbeginner.com/how-to-move-objects-in-unity/#transform_translate

[3] https://answers.unity.com/questions/1558555/moving-an-object-left-and-right.html

[4] https://answers.unity.com/questions/1623695/how-to-make-multiple-objects-spawn-at-random-order.html

[5] https://learn.unity.com/tutorial/moving-the-camera?uv=2019.4&projectId=5f158f1bedbc2a0020e51f0d#