Introduction to C# programming and Unity

Week 3 - Exercise 17

Teddy bear explosions

The **Animator** is an interface to control the animation mechanism. To be more specific of the exercise, **Animator.GetCurrentAnimatorStateInfo()** fetches data from the current state of the Animator. The **normalizedTime** property of the method returns the number of time a state has been looped.

1. *Get the explosion working*

The Animation is accessed through the Animator class and the current state of the animator is accessed via the GetCurrentAnimatorStateInfo method. Therefore, **Animator.GetCurrentAnimatorStateInfo(0).normalizedtime = 0.5** means half the completion of the associated animation. A value of 1 means the completion of the animation. Once the specified condition is reached the gameObject i.e., Explosion is destroyed.

2. *Get teddy bears spawning in scene*

The teddybear script (which applies impulse force to move the teddy bear with relative speed) is attached to the teddybear prefab. Along with which the essential Rigidbody 2d component and box collider is attached. Additionally, the teddybear spawner script (which spawns the teddy bears for a given amount of time) is attached to the main camera along with specifying the teddybear prefab onto the Prefab Teddy Bear field in the script in the Inspector. This spawns the teddy bears periodically in the game space.

3. *Kill teddy bears in multiple ways*

The teddy bear script contains the code to destroy gameObject once the death timer is finished. The teddybear tag is created and added to the teddybear prefab. The code is added to the teddybear script to destroy the teddybears once they collide with each other. Further, the explosion prefab is added to the prefab explosion field of the script. Along with this, the explosion prefab is instantiated within the script (**Instantiate<GameObject>(prefabExplosion, transform.position, Quarternion.Identity)**) which creates an explosion animation whenever two teddybears collide with each other.