

Trinity College Dublin

Final Project Real-Time Animation

Music Video Implementation

Student

Darshan Umapathi
20310597

Professor


Dr. Rachel McDonnell
Department of Computer Science

Date:25/04/2021

Introduction

A part of the final assignment in Real-Time Animation included the implementation of a music video (interactive). I implemented a video of 2 min music video with 2 scenes and characters from Maximo and using Cinemachine and Timeline features from Unity to develop a sequence of animation. This introduced me to the basics of animation with Unity.

Story and Concept:

 Storyline: The music video was inspired by the growth in technology and that would reduce human interaction with society and nature. The protagonist in the video lives in a city with robots and will be sad initially due to the schedule of work and no interaction. On the way to work, he finds a rock that transports him to a new place (forest view) where he will see the forest and animals. Initially, he will be scared and try to run back to the city. But later he turns around and sees a small fox and the same magical stone in the final scene and he must choose between going back to the city or staying in the new home.

Implementation:

We had been given a set of requirements to be fully filled as part of the assignment and here is the list of implemented features.

Engine Used: Unity

Plugins/Frameworks: Cinemachine. Timeline, Navigation Mesh (For AI navigation).

Characters: Mixamo

Then a character selected from mixamo and animations that were relevant to the project was downloaded with the properties.

Character1: Sad Joe

This is the first character that will be introduced in Voxel City. He will be sad and will be traveling to work. I made changes to the character by giving the character white hair as material to make him look old.



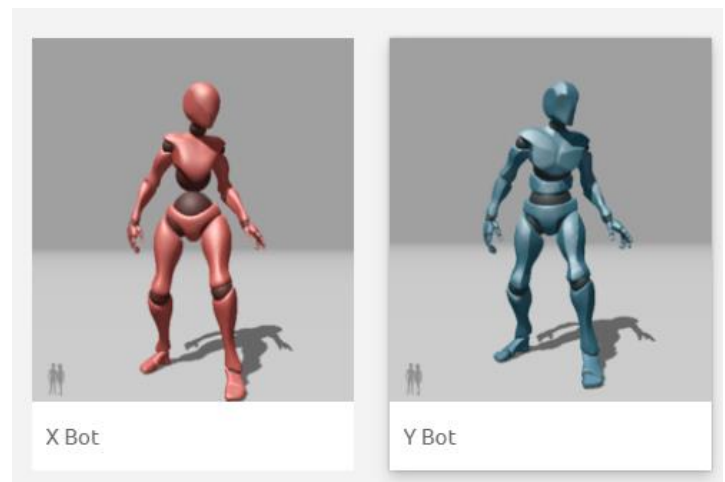
Character2: Remy

This character is a younger version of the joe when he is transported to the second terrain (forest).



Character 3: Bots

There is part of the voxel city and there are used for crowd simulation.



Requirements

✚ **Must have 3-dimensional objects and views**

I had 2 terrains in my music video

Terrain 1

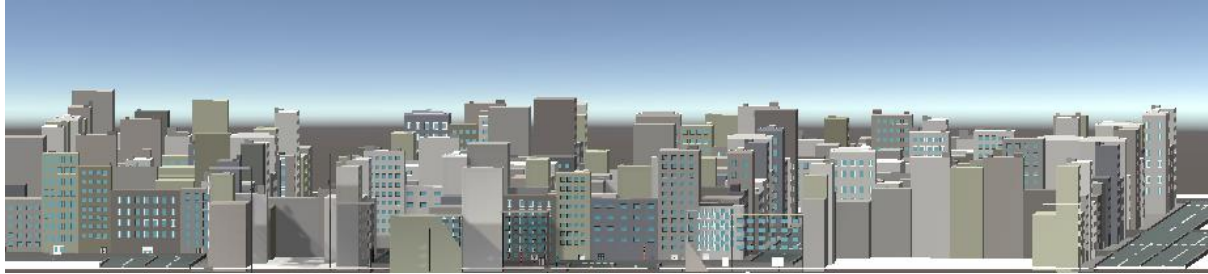


Fig 1: Voxel City

I added the NavMesh to this city to divide the area as walkable and not. Here we had the character 1 and bots (as Crowd)

Terrain 2

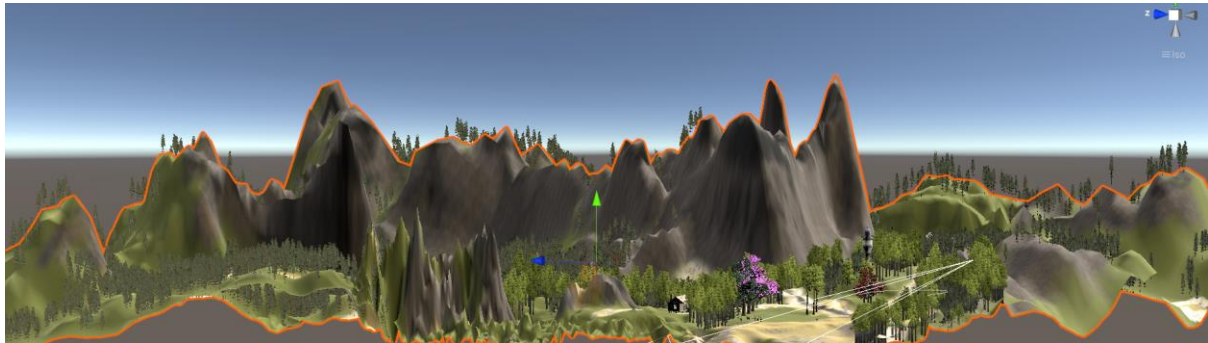


Fig 1.2 The Forest

I developed this terrain from scratch added various elements such as trees, water bodies, and grass.

I have also various animals and birds in the scene: 1. Deer 2. Fox 3. Birds

✚ **Must be 30 seconds or longer**

The video was about 2min long

✚ **Must have a non-linear animated camera path.**

I installed the Cinemachine package from the external package store.

I used Cinemachine virtual cameras. I have created 17 virtual cameras. For example clear shot camera, Dolly and track Camera, to follow view and lookAt view for various characters in the video. Cameras were positioned in such a way to switch between the best shot in the scene this was done by a clear shot camera set up {Cinemachine Brain}.

The below Screenshots shows the camera path for Dolly and track and Cut scene.

Blending two cameras also give a non-linear path as shown in figure 3.2

Fig 3.1 shows the camera path of the starting from 0 to 4 in a curved and looking at the character as a focus creating an arc movement.

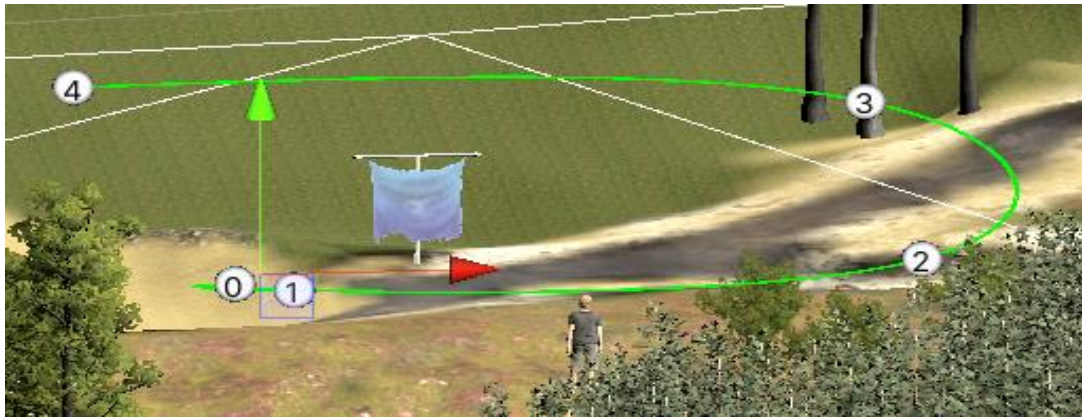


Fig 3.1

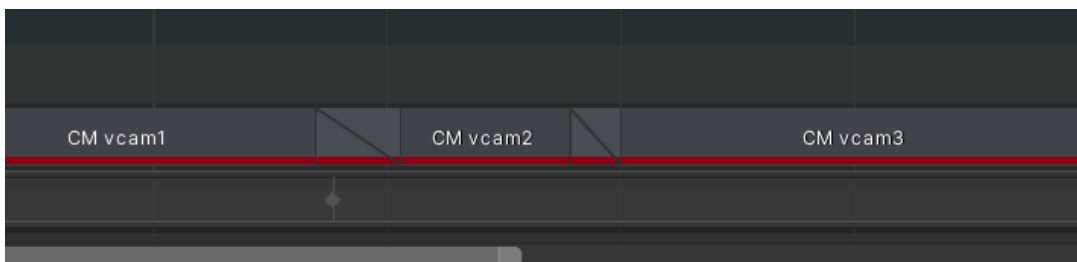


Fig 3.2

✚ **Must have one reasonably realistically moving articulated animated character, as part of the storyline and visible during the camera path:**

The movie contains 6 animated characters including the bots used for crowd simulation. Scene1 had the character Joe who is sad [*melancholy walk animation* - > *followed by a sudden transition to run after seeing the time*] and the animation of walk and he watches the time and realizes it late and begins to run Fig 4.1.



Fig 4.1

Similarly, in the second scene, we have a happy animation of the character walking in joy and also the deer eating and running.

Here I had to change the keyframes to the animation of pointing at the clock manually to make sure the hand points in the correct direction.

✚ Must have an interactive element

I have included one interactive element in the entire video where we can transport from one scene to another using the key button input

Spacebar will take us to scene2 (The Forest) and when we press Q we will take back to the scene1(The Voxel City).

Fade Effect was added for transition.

Code Snippet:

```
Unity Message | 0 references
void Update()
{
    if (Input.GetKey(KeyCode.Space))
    {
        Fadeinto();
        OnComplete(_destination);
    }
    if (Input.GetKey(KeyCode.Q))
    {
        Fadeaway();
        OnCompleted(_destination);
    }
}

1 reference
public void Fadeinto()
{
    anim.SetTrigger("fade2");
    // anim.SetTrigger("fade1");
}

1 reference
public void Fadeaway()
{
    //anim.SetTrigger("fade1");
    anim.SetTrigger("fade2");
}
```

✚ Must clearly demonstrate the following principles of animation in the context of the movie.

Squash and Stretch: This effect introduced when the character from scene 1 stops (Slow motion effect) suddenly realized something shinning

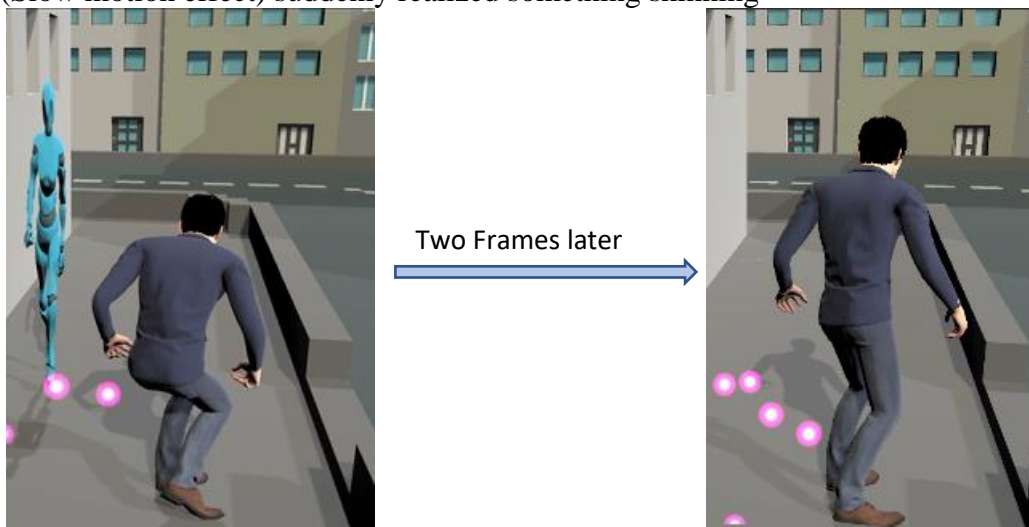


Fig 5.1

Arcs: The arcs were demonstrated in-camera motion, the running motion of the deer. The camera transitioned in between them gave a feel of the artificial curve and zooming in when it moved from front vision to back vision. Fig 3.2

Drag: This is an animation principle that is similar to squash and stretch where the following particles follow the character even after the action is stopped. I have used this effect in the slow-motion effect I have described in fig5.2. This effect was also seen in the cloth simulation of the flag when there was more wind the cloth was flying high but never went behind the pole

Slow in and Slow out: I have used this effect in the camera motion and the character motion from walk to run to again stop. It started slow and then increased speed and then it was later slowed at the end. I had a chance the curve from linear to spline.

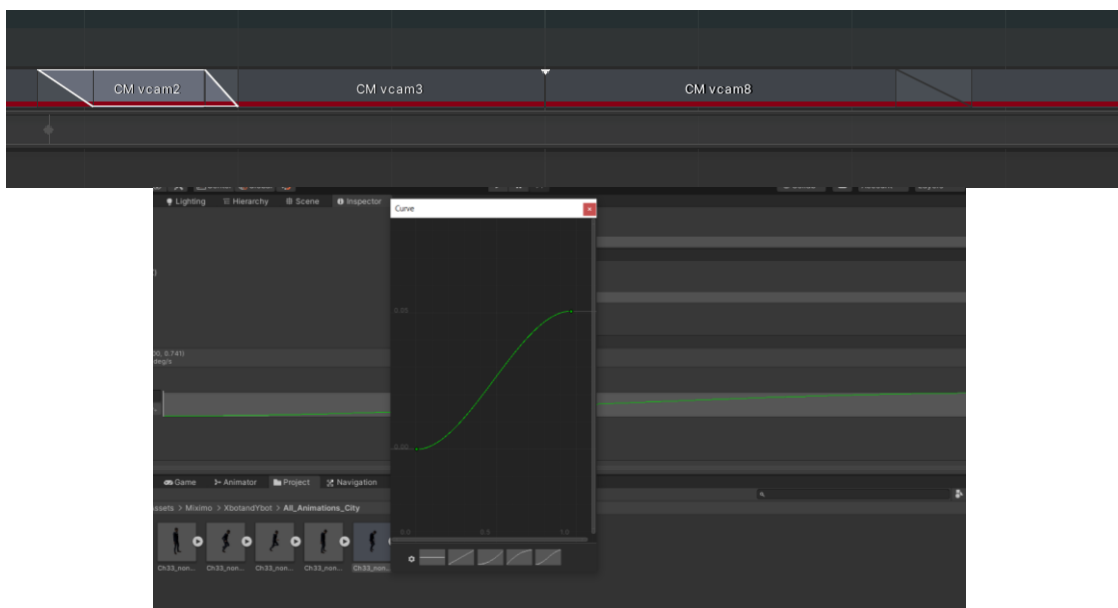


Fig 5.2

Anticipation: This is a principle where the character shows and gives a feeling of what is about to happen next. In the video, I had this principle implemented when the character turns to see the shining stones and wants to pick them up.



Fig 5.2

Exaggeration: As shown in Figure 4.1, The main character gets shocked when he sees the time (late) suddenly enters into running motion from a walk. Also, he reacts to viewing the fox and turns and run with fear.



Fig 5.3

Staging: It is the process of making some ideas that are perfectly clear. This was introduced in scene 2 when the camera rotates around the character and he will be surprised and looking at his hand and leg to see what is changed. He would have become younger. Secondly, when the camera was just focused on the deer and not anything else till the deer leaves the scene.



Fig 5.4

Secondary Action: Principle to secondary action was achieved by making two animations in one movement. In Scene1 (The Voxel City) we see that the animation was related to walking and being sad. The walk was the first (primary) animation and shoulders shrugged and sad was the secondary animation.

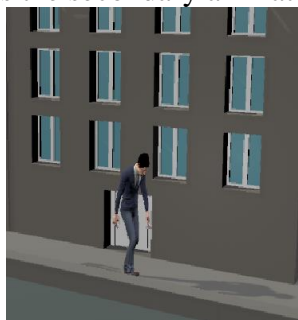


Fig 5.5

Appeal: All the characters introduced in the video had a clear visual appeal about themselves. They were unique and gives the right effect when they are on the screen.

Setting a music track

The music track was selected to match the sad and happy feeling. Selected the royalty-free music and concatenated it on external software to mix the song. After this, it was added to the virtual camera and the cinemachine camera brain was able to carry the music with the active camera as the shelf setting that can be used for the developer.

Research and Additional Features:

1. Crowd Simulation and AI unity

The terrain was added with a navmesh component then there was an area that is divided into walkable and non-walkable. I have added over 20 bots in this video for crowd simulation. Here the bots are nav agents and they have a rigid body component that avoids them from colliding with each other. They are then set to a goal using the below script. When they reach the goal they are then moved to another goal location at random. Hence, these bots will always be in motion during play mode. To make them independent of each other every bot was given its own speed and offset for movement. Therefore, all the bots are walking at an independent speed making all the bots unique.

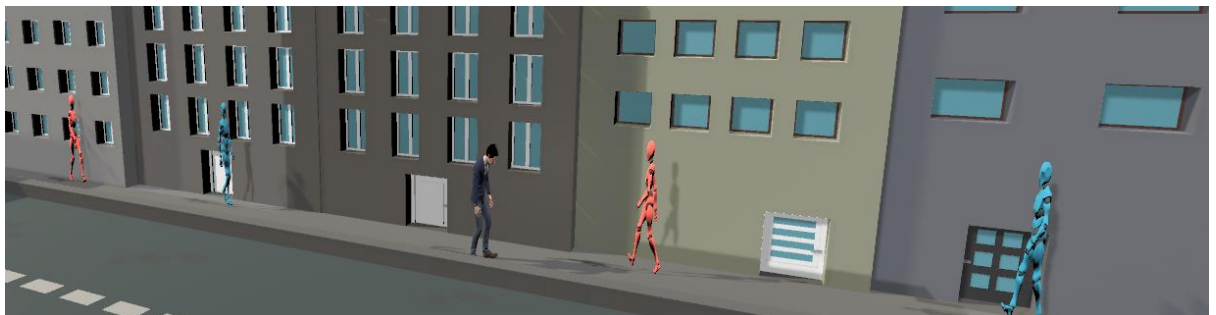
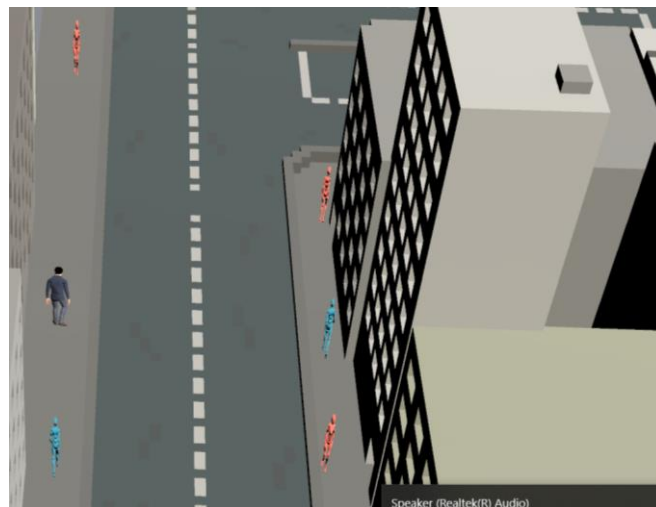


Fig 6.1 Crowd walking to destination

Code Snippet for changing speed and goal for the simulated crowd

```
// (Scripting)
GameObject[] goalLocations;
NavMeshAgent agent;
Animator anim;
@ Unity Message | 0 references
void Start()
{
    goalLocations = GameObject.FindGameObjectsWithTag("Goal");
    agent = this.GetComponent<NavMeshAgent>();
    agent.SetDestination(goalLocations[Random.Range(0,goalLocations.Length)].transform.position);
    anim = this.GetComponent<Animator>();
    anim.SetFloat("Woffset", Random.Range(0, 1));
    anim.SetTrigger("walking");
    float sm = Random.Range(0.75f, 2.0f);
    anim.SetFloat("Soffset", sm);
    agent.speed *= sm;
}

// Update is called once per frame
@ Unity Message | 0 references
void Update()
{
    if(agent.remainingDistance < 1)
    {
        agent.SetDestination(goalLocations[Random.Range(0, goalLocations.Length)].transform.position);
    }
}
```

Fig 6.1.1

2. Complex camera motion

As mentioned above in the non-linear camera path. I have leveraged the cinemachine feature of Unity to add over 15 virtual cameras in the entire video and the cinemachine brain attached to the main camera controls the transition between these cameras based on the priority suggested in the console window.

The blend between the cameras can also be controlled by the TimeLine feature in Unity cut scene we will use different cameras with no blending. If we want a non-linear path we will blend the path of two cameras and the cinemachine brain creates the best path for it.

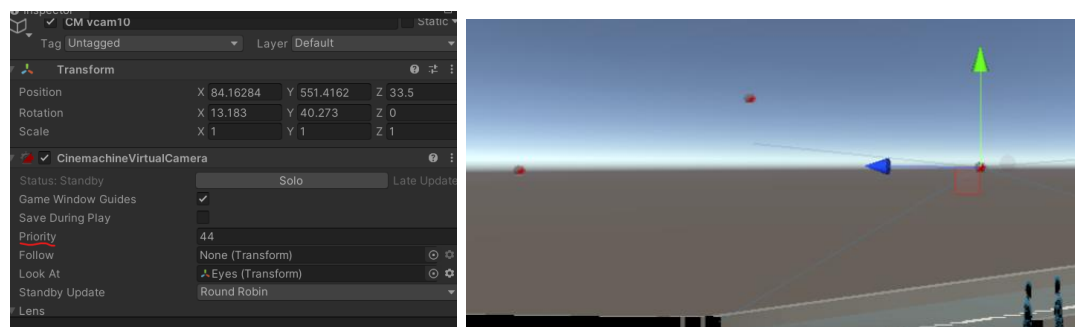


Fig 6.2

3. Cloth Simulation

Simulation of Cloth in Games and Animation plays a significant role. For this video, I have implemented a basic signal flag in the scene2. I used an asset as a reference to replicate the same. I added the effects such as adding the vertical movement of the cloth and the varied speed of the wind and the effect of it on the cloth.

I adjusted the speed of the wind and generated artificial wind in the play mode

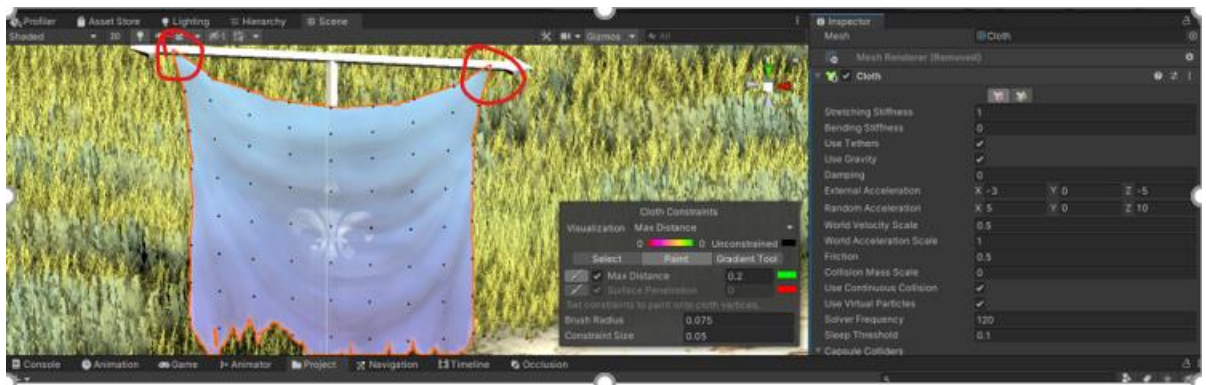


Fig 6.3: Adding the endpoints to the flag to keep it attached

4. Advanced Particle system

Unity has an inbuilt particle system generator that can be used for special effects using particles. Created a particle system generator with properties as follows

Shape-Cone created a custom color (pink) added gravity so it falls back to the ground rather than flying up in the sky. No of particles emitted-300. The first particle system has a code that triggers to increase in the number of particles to 1200 when the stone is being picked from its place. Refer Fig6.4. There was another created at the end of the video which depicts that if the character picks the stone he will send it back to the scene1(Controlled by key Press)



Fig 6.4

5. Motion State Machine and Motion Capture

I have downloaded the animations for characters from Mixamo.com and used them with Animation Animator and Timeline

Fox animation: I downloaded a Unity Package “The Fox” Which had few animations readily available. I added the right turn and left turn animations using the keyframe using rotation towards the right and left.

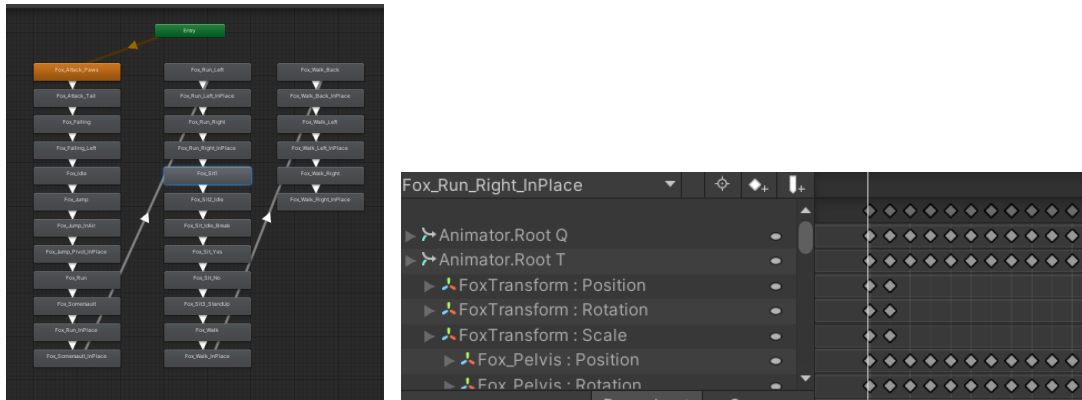


Fig 6.5 Fox animation

Deer Animation: Similar to Fox animation I downloaded the main animations for the deer and made changes to walk and walk speed. Eat and Eat speed to make it look natural.

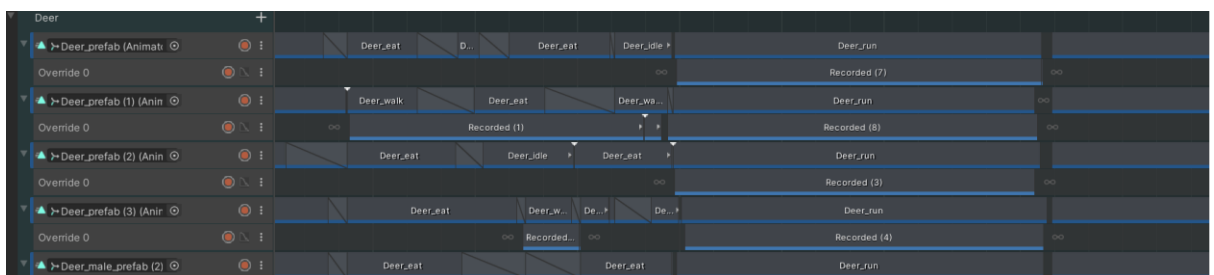
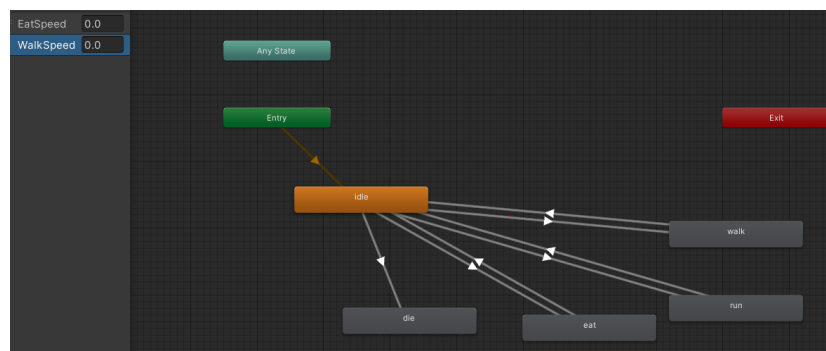


Fig 6.5.1 Deer Animation

From 6.5.1 We can see how we can blend animations using TimeLine and also override it for our requirement and can control all the characters individually. The Overrides in the above pictures tell what animation to override and how to animate it based on the requirement.

6. Motion Editing – blending

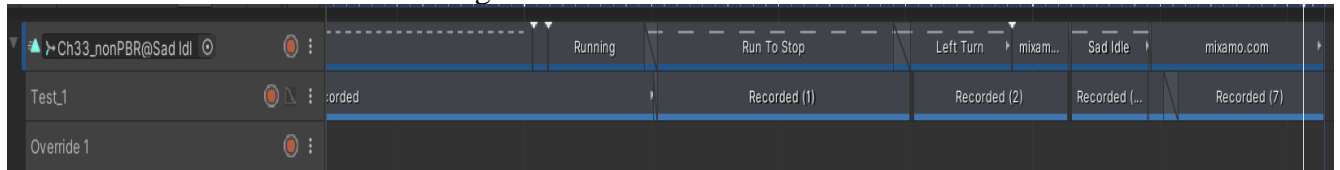
The Unity Timeline is a feature that allows you to visually edit music, play animations, show/hide objects, control particles, etc. along the time axis.

It is similar to video editing software but for animations. It may be used to create a spatial piece of work where the movement of multiple objects and the timing of music are combined to create a storyline that unfolds.

We can create any number of timelines but the need to be associated with a GameObject I have used Timeline for Blending the animations like walk to run or run to stop.

Every character animation is individually controlled and played at the right time using Timeline, We can also override these inbuilt animations in the timeline to create custom animations.

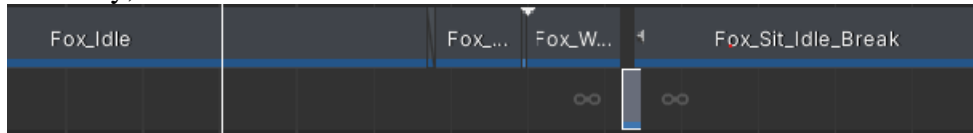
Character 1 Animation with Blending and override



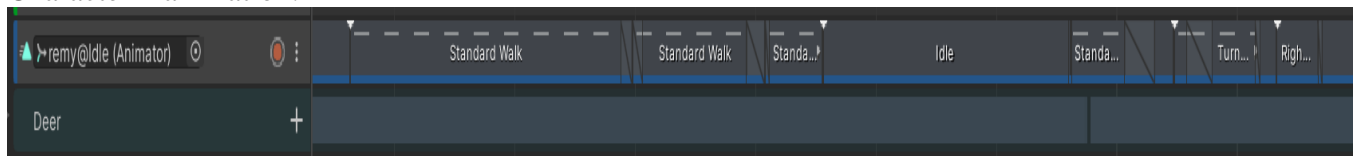
Recorded(1), Recorded(2) etc are overridden animations.

We saw the Deer animations in Fig 6.5.1

Similarly, we have Fox animation:



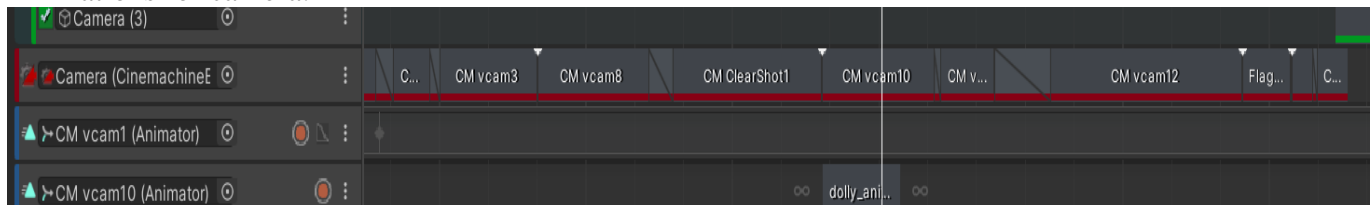
Character 2 animation:



Timeline gives a feature to add a camera and its transition into the timeline to control the flow and change of camera. I have used 17 virtual camera which includes clear shot virtual camera and dolly and tracks virtual camera.

I can blend them independently and create blends and transitions without making them active and writing code for the movement. This will be taken care of by the cinemachine brain

Animations for camera:



References:

- [1] Characters: [Mixamo](#)
- [2] [NavMesh with AI for Crowd Simulation](#)
- [3] [Crowd Simulation](#)
- [4] [Cloth Simulation](#)
- [5] [Timeline and Cinemachine Tutorial from Unity](#)

[6]Assets:

- 1. Birds: <https://assetstore.unity.com/packages/3d/characters/animals/birds/living-birds-15649>
- 2. Cute Fox: <https://assetstore.unity.com/packages/3d/characters/animals/toon-fox-183005>
- 3. Grass and Flowers for Terrain: <https://assetstore.unity.com/packages/2d/textures-materials/nature/grass-and-flowers-pack-1-17100>
- 4. Standard Assets: <https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2018-4-32351>

Youtube link: <https://youtu.be/bishH-467mo>