

CS7GV5 Report

Name:	Long Pan
Student ID:	21332147
Youtube link:	https://youtu.be/ZY5mjh-AUNs
Declaration:	I declare that this work is my own and that all the resources that I have used have been referenced.

Required feature: Articulated Animated Character

Screenshot(s) of feature:



Describe your implementation of a - c:

- **Keyframed animation**

About the Keyframed animation part, I have used this to control the transformation of my characters, such as moving crocodiles and deers. This is the method I use the most, during the making of this project.

- **Motion State Machines**

About this part, I mainly used it to convert different animation states of my characters. For example, when the crocodile is going to eat the deer, there will be some animation changes, such as moving forward slowly, then waiting for the bite, and finally biting forward. Deer will also have corresponding animations, first drinking water, and then jumping. The advantage of using a state machine is that I can control the animation to transition at a specified moment.

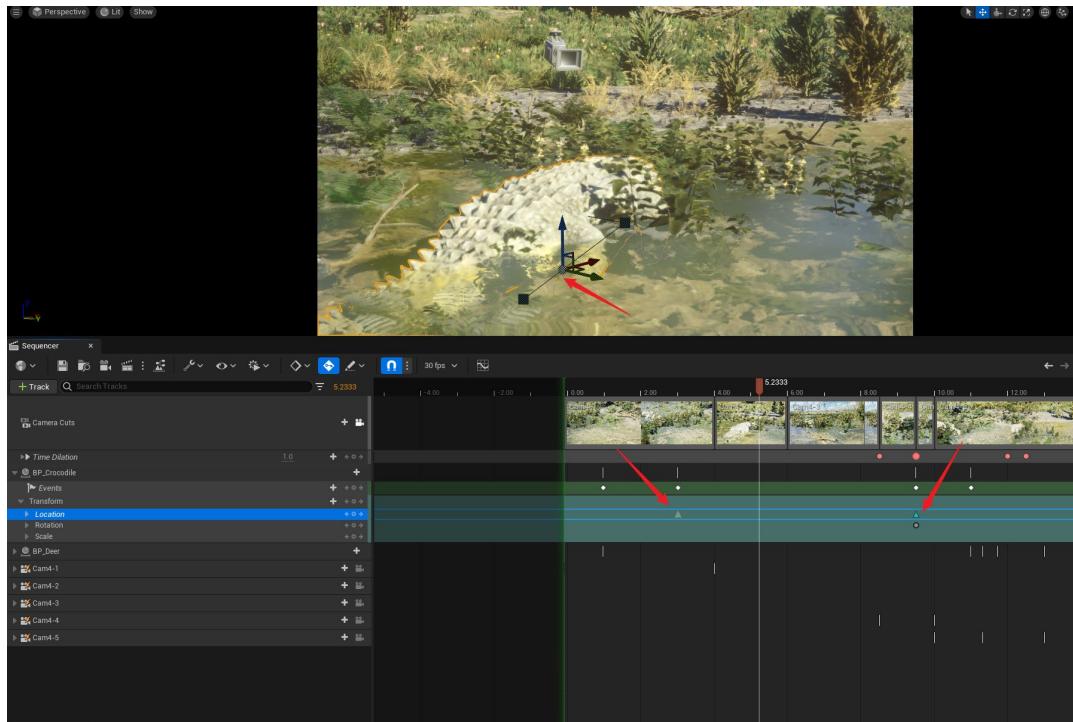
- **Motion Editing - Blending**

For cases where the desired animation is not available, using Blending is the best option. It can be blended based on existing animations, or it can blend multiple

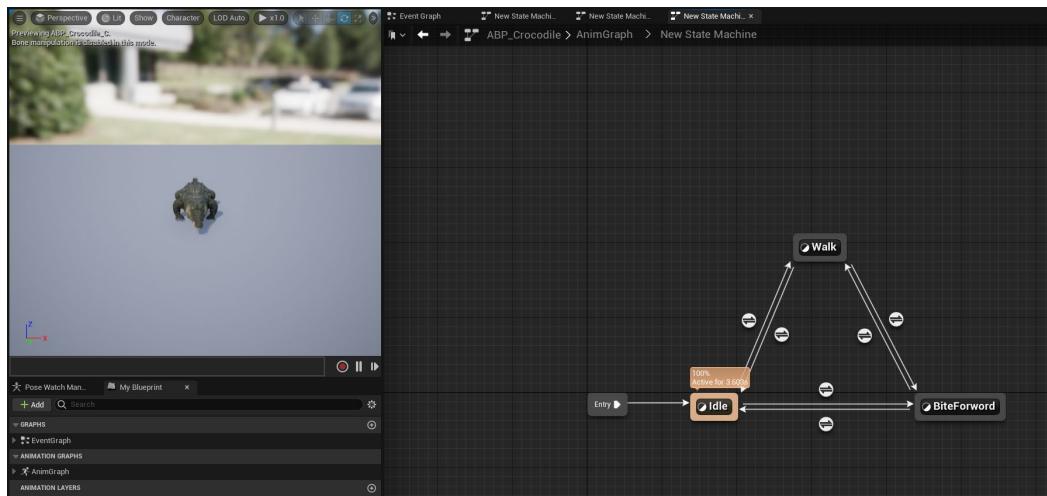
animations and then take the blended state. In this project, I mainly use two kinds of blending, one is the blending when using keyframe for direct animation switching. The purpose of this is to make animation transitions less rigid.

Code/Blueprint Screenshot:

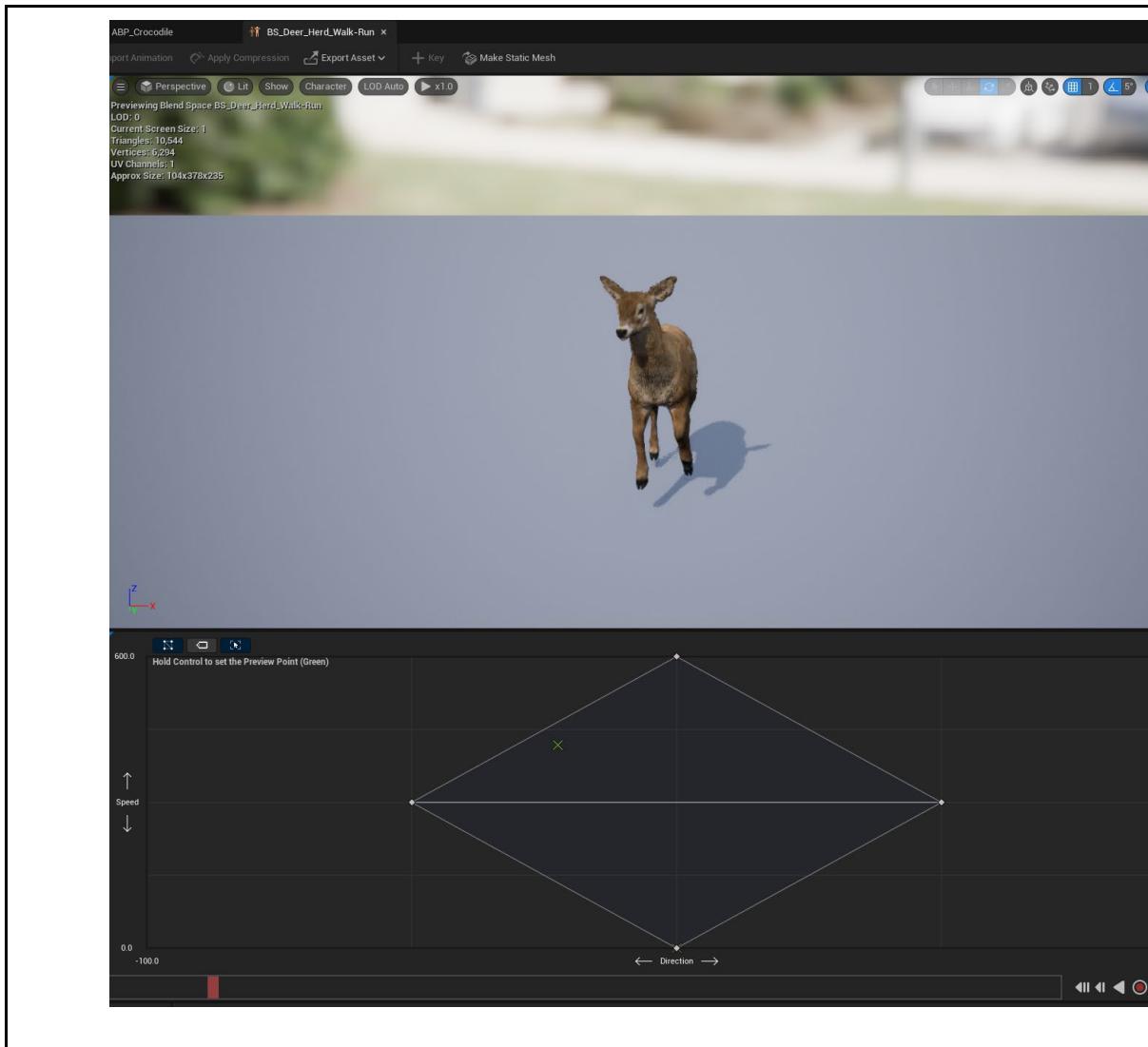
- **Keyframed animation**



- **Motion State Machines**



- **Motion Editing - Blending**



Required feature: Interactive Element

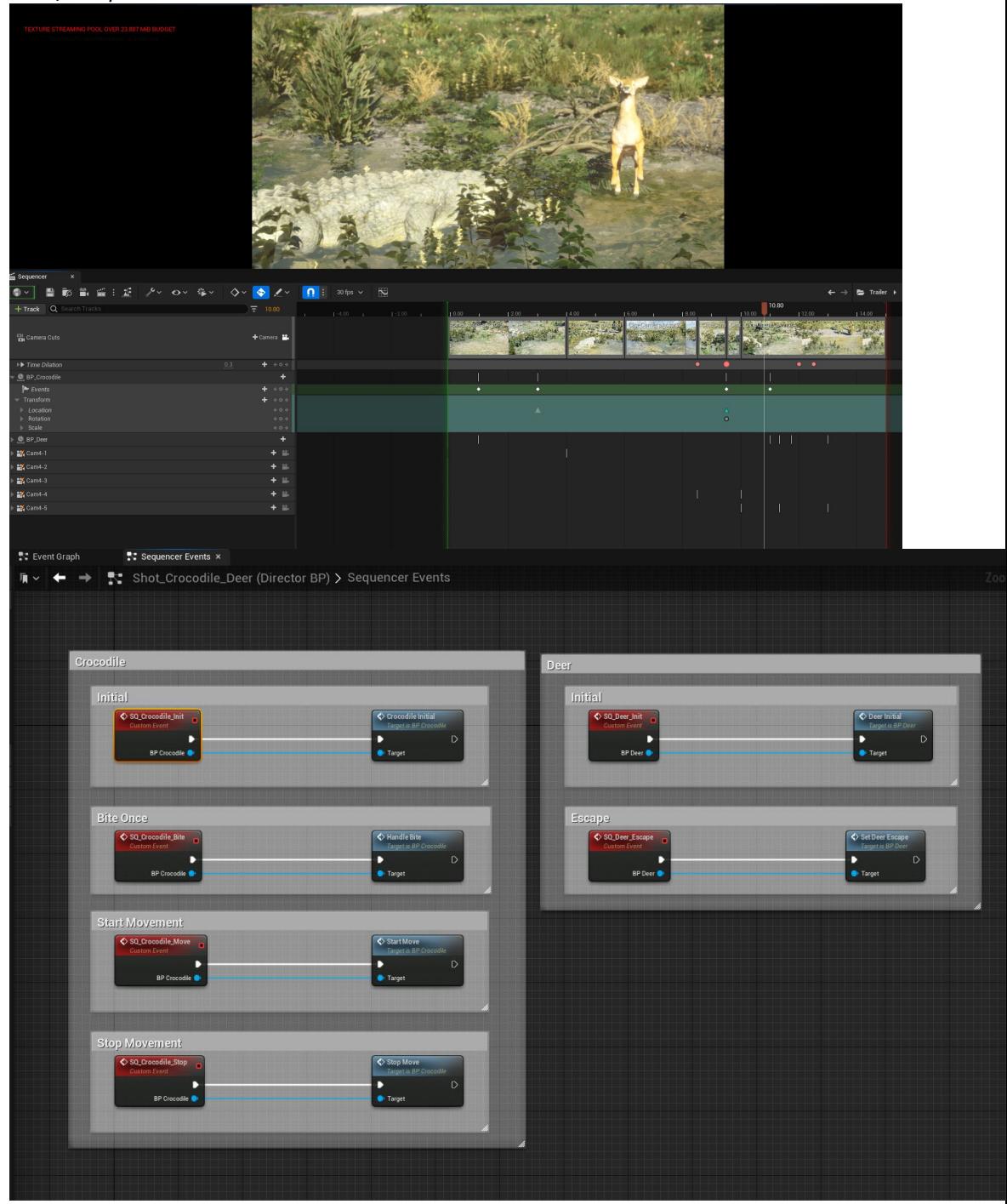
Screenshot(s) of feature:

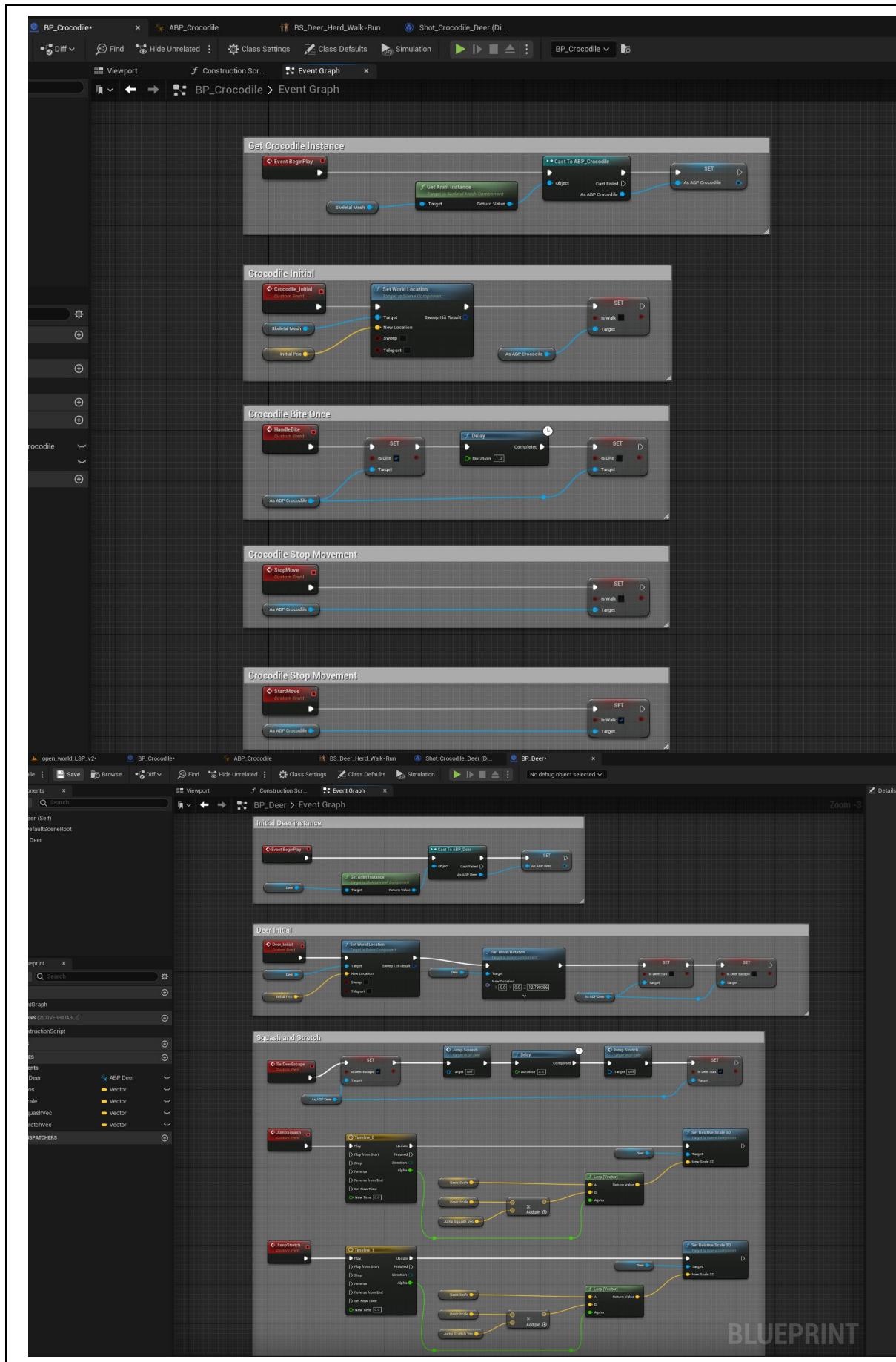


Describe your implementation:

I created a scene where a crocodile eats a drinking deer. When the crocodile was about to eat the deer, the deer reacted ahead of time and jumped up to escape. This is the part where I do the interaction of the two different characters. The principle of implementation is to control the animation of different characters through the state machine, and then set the conversion rule as a variable. Write a custom function in the corresponding character blueprint class to manipulate the corresponding variables. Then add a trigger event at the corresponding position in the sequence. In the blueprint of the sequence, the trigger event calls the corresponding custom function, thereby modifying the variable that controls the animation state.

Code/Blueprint Screenshot:





Required feature: Principles of Animation

Screenshot(s) of feature:



Describe your implementation of a - e:

- **Staging**

In the scene where the crocodile eats the deer, I used multiple cameras to shoot the scene from different perspectives, and switched the perspectives of different characters to better express the development of the whole story.

- **Squash and Stretch**

When the crocodile was about to eat the deer, the deer reacted from the action of drinking water, and then in order to show that it was frightened, I used the Squash and stretch animation to make its jump more effective.

- **Ease in, Ease out**

This animation principle is used in many places throughout the project. In the keyframe of the sequence, the switch of each keyframe uses the Ease in, Ease out principle, which can make the switching between different states more natural .

- **Anticipation**

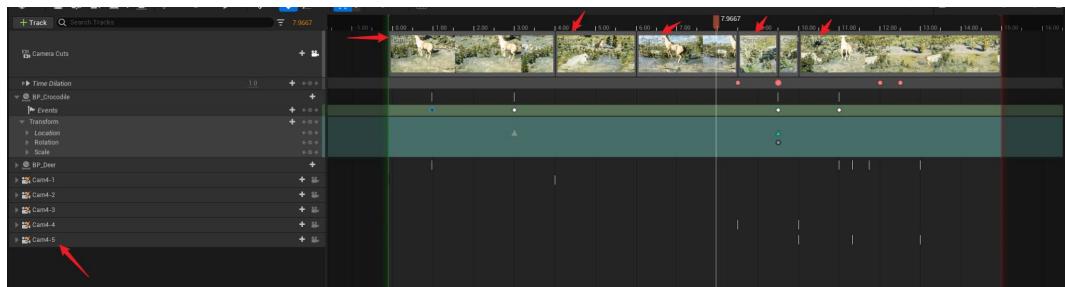
Before the crocodile was about to eat the deer, I deliberately made a pause, in order to allow the crocodile to have a process of accumulating energy from moving forward slowly to biting forward.

- **Exaggeration**

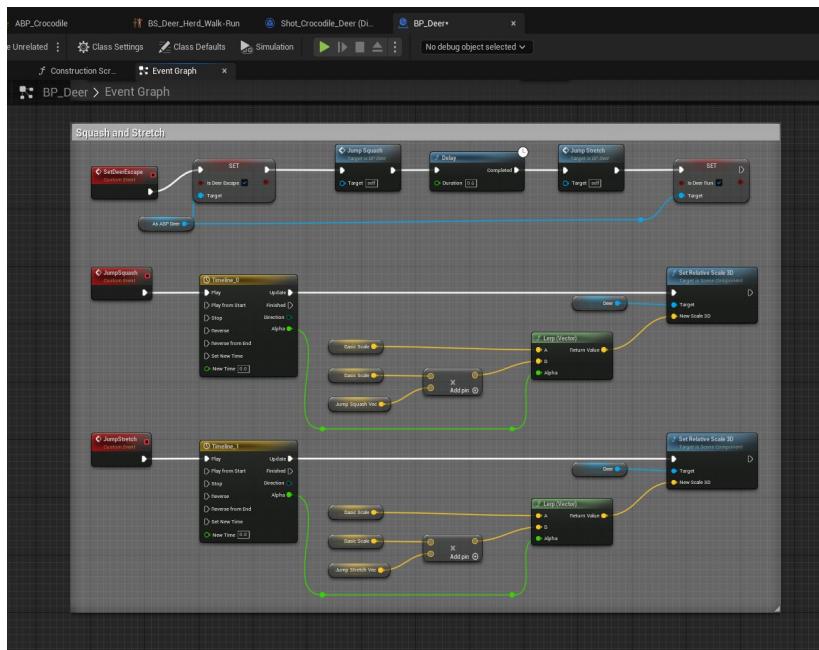
I added this animation principle to the place where the deer jumped after being scared by the crocodile, so that the original deer jumped higher. This can also show that the deer is really scared by the crocodile.

Code/Blueprint Screenshot:

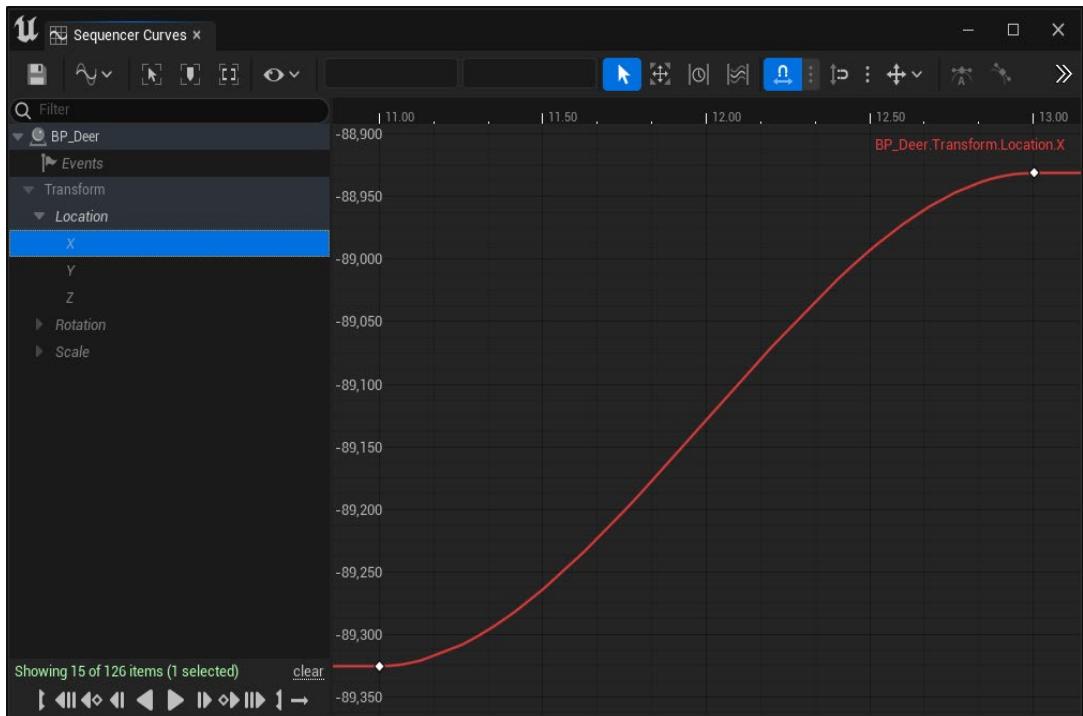
- **Staging**



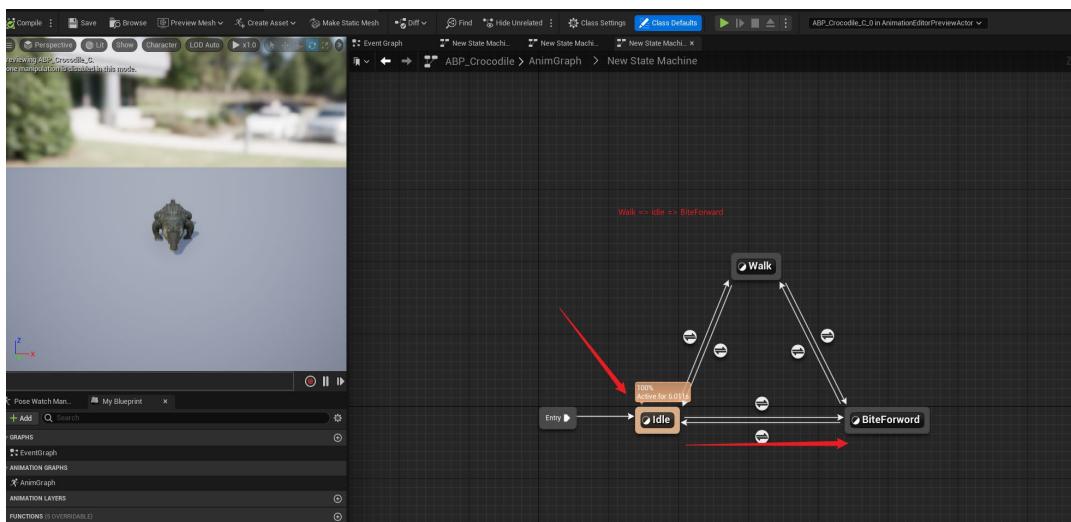
- **Squash and Stretch**



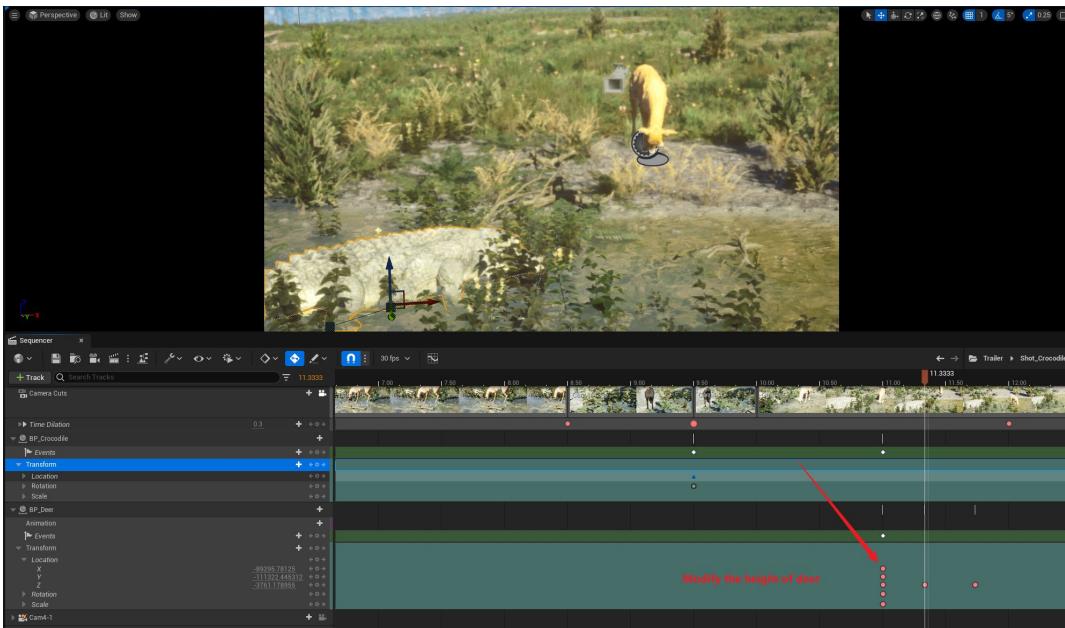
- Ease in, Ease out



- Anticipation



- **Exaggeration**



Additional feature: Crowd simulation

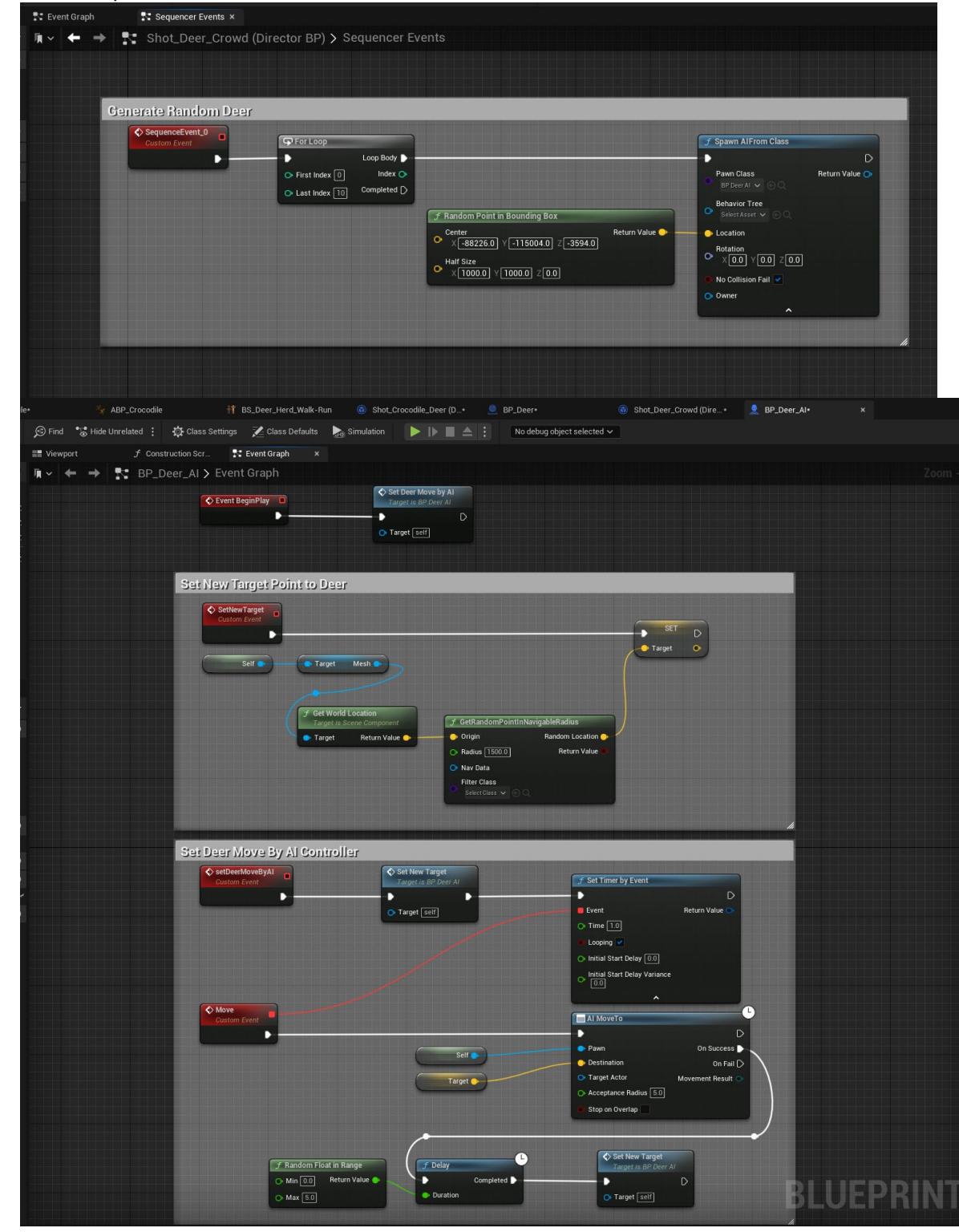
Screenshot(s) of feature:



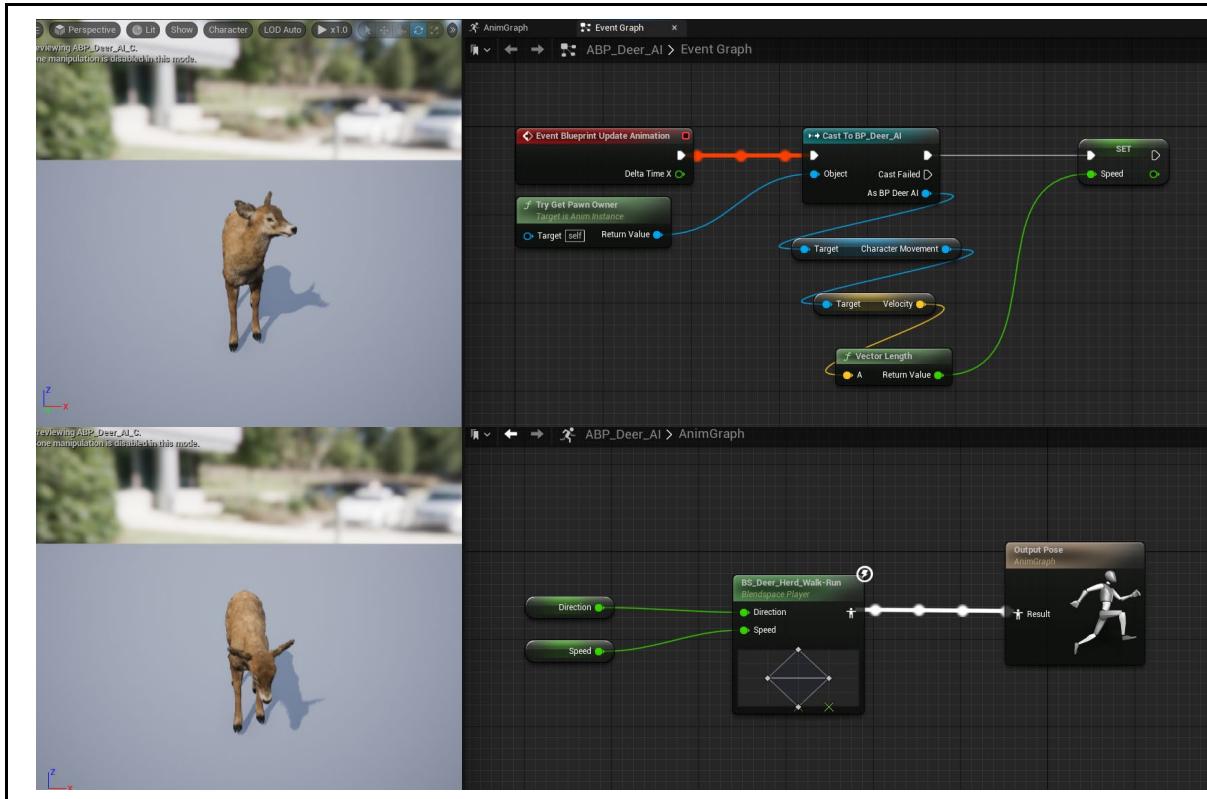
Describe your implementation:

I have realized a scene about deer drinking water by the lake. In this scene, there will be some drinking deer, some walking, and some running. These deer are generated using scripts, and I can control the number of spawns at will, so I think this belongs to the part of crowd simulation. In the implementation process, I used a blueprint function node:**Spawn AIForm Class**. With it I can map a character blueprint class at will. Then use the "for loop" node in the blueprint to adjust the number of times you want to render.

Code/Blueprint Screenshot:



BLUEPRINT



Additional feature: Advanced Particle systems

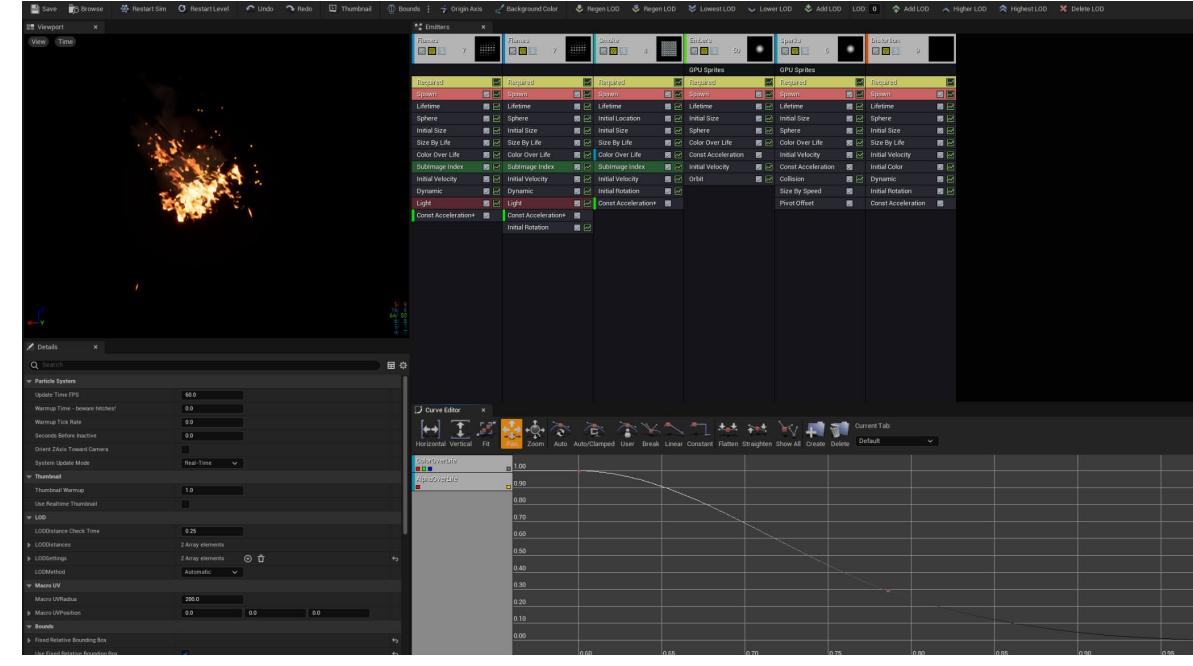
Screenshot(s) of feature:



Describe your implementation:

I have a scene with a fire source element added, the fire source is made using an advanced particle system.

Code/Blueprint Screenshot:



Additional feature: Physically-based animation

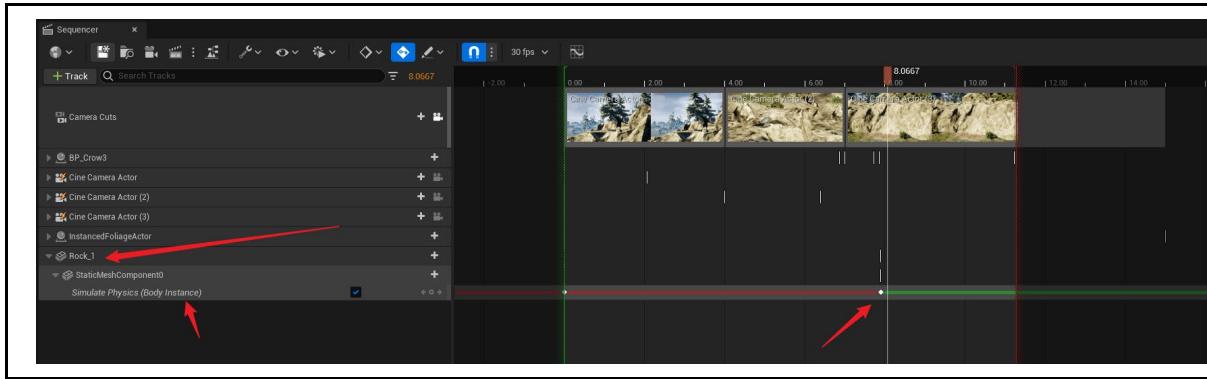
Screenshot(s) of feature:



Describe your implementation:

I have a scene about a crow pushing a small stone down a hill. In this story scene, the pebbles are animated based on physics when they fall. It is also very simple to implement. I just need to check the physical simulation in the object of the small stone, and then selectively turn on the physical simulation in the sequence.

Code/Blueprint Screenshot:



Additional feature: Gesture or personality modelling for the character

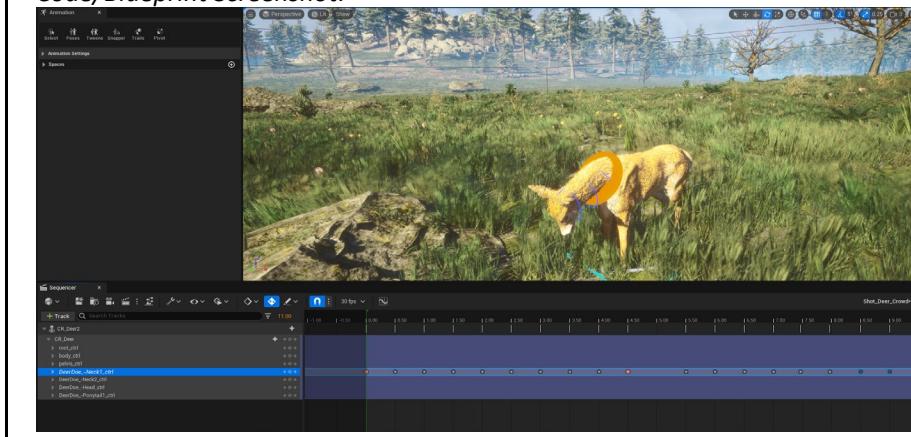
Screenshot(s) of feature:

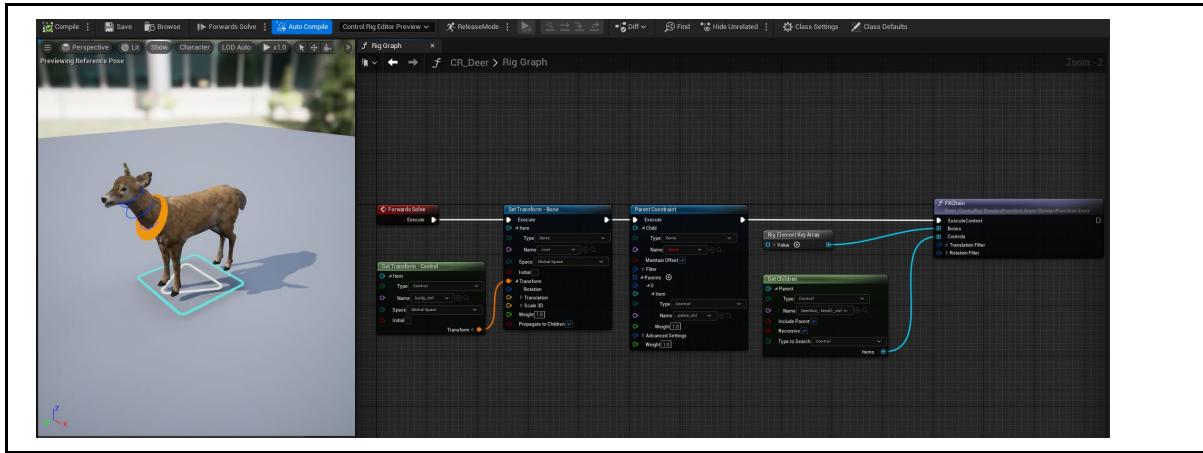


Describe your implementation:

I animated the deer in the scene, using a rig to set up a custom animation of the deer. Then in the sequence, by calling keyframes to control the different bones of the deer to perform different displacement and rotation, I can achieve the personality modeling animation I want for the deer. Here I implemented nodding and shaking head animations.

Code/Blueprint Screenshot:





Additional feature: Complex/scripted camera motion

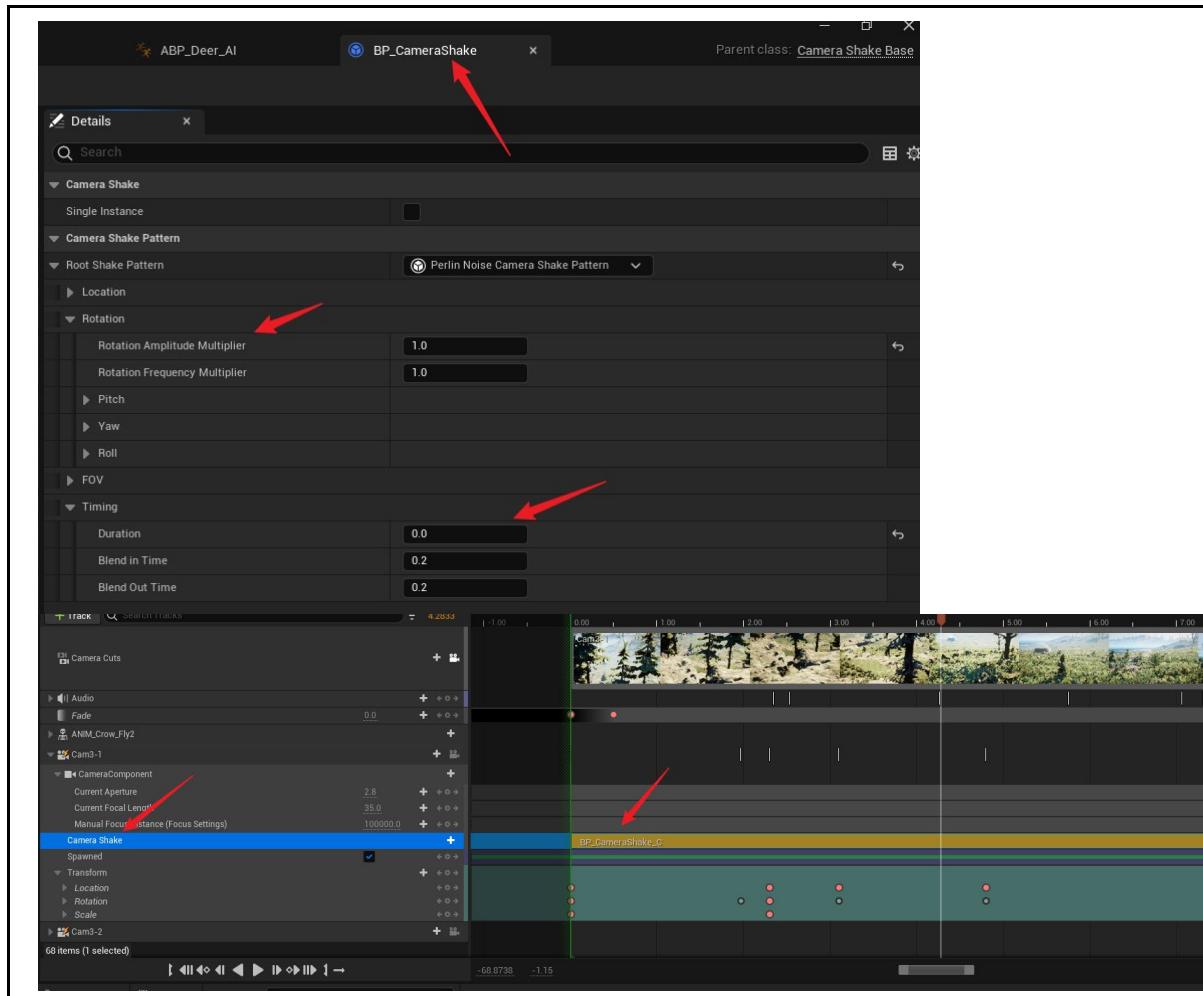
Screenshot(s) of feature:



Describe your implementation:

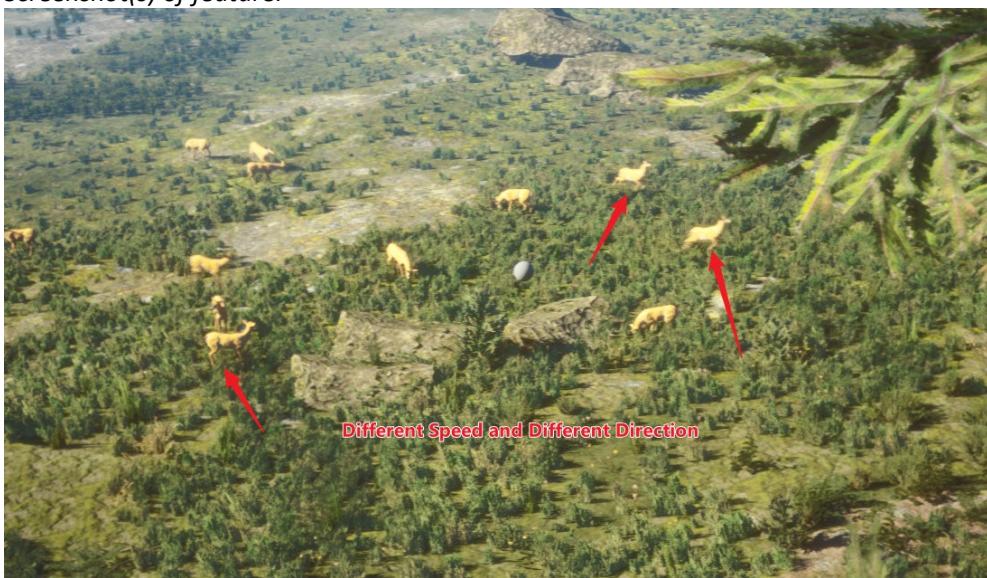
There is a scene in the project about a crow flying down from the mountain. When it is flying, the camera follows it. In this case, I added a camera shake animation to simulate the swing during the flight. Here I use the blueprint class of CameraShakeBase to control the shaking of the camera, and then set the shaking parameters for the animation that needs to be shaken in the sequence to achieve the corresponding effect.

Code/Blueprint Screenshot:



Additional feature: Interesting character behaviours / AI

Screenshot(s) of feature:



Describe your implementation:

When generating the herd of deer, in order to make the deer more natural, I blended some basic animations of the deer, and then used AI to randomly position, direction and speed the deer when generating the corresponding deer blueprint object. And set the target point for each deer, when the deer is created, it will run to the target point at the corresponding speed. Here I use the AIMoveTo function to represent the process of AI controlling the deer's automatic pathfinding.

Code/Blueprint Screenshot:

