```
#include "CrashAngelArts.h"
#include "VJTimeline.h"
AVJTimeline::AVJTimeline(){ PrimaryActorTick.bCanEverTick = true; }
void AVJTimeline::Tick(float xDeltaTime){ Super::Tick(DeltaTime); }
void AVJTimeline::TimeLine_Update(float xValue){ ReturnFloat = xValue; }
void AVJTimeline::BeginPlay()
{
Super::BeginPlay();
bool StartConfig = false;
bool xTimeLineUpdate = true;
bool xTimeLineChange = false;
TimeLine = *new FTimeline;
TimerHandle = *new FTimerHandle;
xEditCurve = *new FRichCurve;
xCurveFloat = ConstructObject<UCurveFloat>(UCurveFloat::StaticClass());
TimeLine.PlayFromStart();
GetWorldTimerManager().SetTimer(TimerHandle, this, &AVJTimeline::TickTimeline,
DeltaTime, true, 0.0f);
}
void AVJTimeline::VJTimeLine_Execute(){
if (StartConfig == false){ AVJTimeline::Timeline_Execute_Change(); StartConfig = true; }
if (xTimeLineUpdate == true && xTimeLineChange == true){
xTimeLineUpdate = true;
TimeLine.SetLooping(false);
FScriptDelegate FinishedEvent;
FinishedEvent.BindUFunction(this, TEXT("Timeline_Execute_Change"));
TimeLine.SetTimelineFinishedFunc(FOnTimelineEvent(FinishedEvent));
}
}
void AVJTimeline::Timeline_Execute_Change(){
float xTimeLength = CalculateBPMSecondsToBeats(BPM, Beats);
FOnTimelineFloat ProgressEvent{};

xEditCurve.Reset();
```

```
TimeLine.SetTimelineLengthMode(ETimelineLengthMode::TL_LastKeyFrame);
TimeLine.SetTimelineLength(CalculateDegree(Degree));
TimeLine.SetLooping(IsLoop);
KeyHandle = xEditCurve.AddKey(0.0f, 0.0f);
KeyHandle = xEditCurve.AddKey((xTimeLength / 2), CalculateDegree(Degree), false,
KeyHandle);
KeyHandle = xEditCurve.AddKey(xTimeLength, 0.0f, false, KeyHandle);

xEditCurve.SetKeyInterpMode(KeyHandle, Curve);
xCurveFloat->FloatCurve = xEditCurve;
ProgressEvent.BindUFunction(this, "TimeLine_Update");
TimeLine.AddInterpFloat(xCurveFloat, ProgressEvent, FName{ TEXT("TimelineProgress") });
TimeLine.Play();

xTimeLineUpdate = false;
xTimeLineChange = false;
}
void AVJTimeline::VJTimeLine_Update(float xValue){
xTimeLineUpdate = true;
if (xTimeLineChange == true){ xTimeLineChange = false; }
else{ xTimeLineChange = true; }
}

void AVJTimeline::TickTimeline(){
if (TimeLine.IsPlaying()){
if (xTimeLineUpdate == true){
AVJTimeline::VJTimeLine_Execute();
xTimeLineUpdate = false;
}
TimeLine.TickTimeline(DeltaTime);
}
else{
GetWorldTimerManager().ClearTimer(this, &AVJTimeline::TickTimeline);
SetLifeSpan(0);
}
}
float AVJTimeline::CalculateBPMSecondsToBeats(float xBPM, ETimeline_Beats xBeats) {
float xBeatsPerSecond = (xBPM / 60);
float x1Beat = (1 * xBeats);
```

```
float xDivide = x1Beat / xBeatsPerSecond;
return xDivide;
}
float AVJTimeline::CalculateDegree(ETimeline_Degree xDegree){
float xReturn = 0.f;
if (xDegree == ETimeline_Degree::Timeline_Degree_025){ xReturn = 25.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_045){ xReturn = 45.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_075){ xReturn = 75.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_090){ xReturn = 90.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_115){ xReturn = 115.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_135){ xReturn = 135.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_155){ xReturn = 155.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_180){ xReturn = 180.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_205){ xReturn = 205.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_225){ xReturn = 225.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_245){ xReturn = 245.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_270){ xReturn = 270.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_295){ xReturn = 295.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_315){ xReturn = 315.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_335){ xReturn = 355.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_360){ xReturn = 360.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_720){ xReturn = 720.f; }
else if (xDegree == ETimeline_Degree::Timeline_Degree_1080){ xReturn = 1080.f; }
return xReturn;
}
```