

```

4 using UnityEngine.Playables;
5
6 public class TesterDirector : MonoBehaviour {
7
8     public PlayableDirector pd;
9
10
11     // Use this for initialization
12 void Start () {
13
14     }
15
16     // Update is called once per frame
17 void Update () {
18     if (Input.GetKeyDown(KeyCode.H)) {
19         pd.time = 0;
20         pd.Stop();
21         pd.Evaluate();
22         pd.Play ();
23     }
24 }

```

计算当下每个物件所在的位置

控制Timeline不断从头开始播放

```

[Serializable]
public class MySuperPlayableBehaviour : PlayableBehaviour
{
    public ActorManager am;
    public float myFloat;

    PlayableDirector pd;

    public override void OnPlayableCreate (Playable playable)
    {

    }

    public override void OnGraphStart(Playable playable){
        pd = (PlayableDirector)playable.GetGraph ().GetResolver ();
    }

    public override void OnGraphStop(Playable playable){
        if (pd != null) {
            pd.playableAsset = null;
        }
    }
}

```

playable拿对应的导演对象

动态绑定轨道对象

```

47 }
48
49 public void PlayFrontStab(string timelineName, ActorManager attacker, ActorManager victim){
50     if (timelineName == "frontStab") {
51         pd.playableAsset = Instantiate(FrontStab);
52
53         TimelineAsset timeline = (TimelineAsset)pd.playableAsset;
54         //Debug.Log (timeline.name);
55
56         foreach (var track in timeline.GetOutputTracks()) {
57             if (track.name == "Attacker Script") {
58                 pd.SetGenericBinding (track, attacker);
59             }
60         }
61
62         // foreach (var trackBinding in pd.playableAsset.outputs) {
63         //     if (trackBinding.streamName == "Attacker Script") {
64         //         pd.SetGenericBinding (trackBinding.sourceObject, attacker);
65         //     }
66         //     else if (trackBinding.streamName == "Victim Script") {
67         //         pd.SetGenericBinding (trackBinding.sourceObject, victim);
68         //     }
69         //     else if (trackBinding.streamName == "Attacker Animation") {
70         //         pd.SetGenericBinding (trackBinding.sourceObject, attacker.ac.anim);
71         //     }
72         //     else if (trackBinding.streamName == "Victim Animation") {
73         //         pd.SetGenericBinding (trackBinding.sourceObject, victim.ac.anim);
74         //     }
75         // }
76         pd.Play ();

```

TimeLineAsset方式获取track后绑定

获取trackbinding进行绑定

```

//         pd.SetGenericBinding (trackBinding.sourceObject, attacker.ac.anim)
//     }
//     else if (trackBinding.streamName == "Victim Animation") {
//         pd.SetGenericBinding (trackBinding.sourceObject, victim.ac.anim);
//     }
// }

pd.Evaluate (); ← pd.Evaluate ==> 手动计算一次数值

pd.Play ();
}

```

修改自定义playablebehaviour上的属性值

```

}

public void PlayFrontStab(string timelineName, ActorManager attacker, ActorManager victim){
    if (timelineName == "frontStab") {
        pd.playableAsset = Instantiate(frontStab);

        TimelineAsset timeline = (TimelineAsset)pd.playableAsset;
        //Debug.Log (timeline.name);

        foreach (var track in timeline.GetOutputTracks()) {
            if (track.name == "Attacker Script") {
                pd.SetGenericBinding (track, attacker);
                foreach (var clip in track.GetClips()) {
                    MySuperPlayableClip myclip = (MySuperPlayableClip)clip.asset;
                    MySuperPlayableBehaviour mybehav = myclip.template;
                    //Debug.Log (mybehav.myFloat);
                    mybehav.myFloat = 777;
                }
            }
            else if (track.name == "Victim Script") {
                pd.SetGenericBinding (track, victim);
            }
            else if (track.name == "Attacker Animation") {
                pd.SetGenericBinding (track, attacker.ac.anim);
            }
            else if (track.name == "Victim Animation") {
                pd.SetGenericBinding (track, victim.ac.anim);
            }
        }
    }

    foreach (var trackBinding in pd.playableAsset.outputs) {
        if (trackBinding.streamName == "Attacker Script") {

```

通过track拿到上面的clip, 然后通过clip拿到对应的playableBehaviour

修改自定义playablebehaviour上的ExposedReference属性值，

```

{
    public MySuperPlayableBehaviour template = new MySuperPlayableBehaviour ();
    public ExposedReference<GameObject> myCamera;

    public ClipCaps clipCaps
    {
        get { return ClipCaps.Blending; }
    }

    public override Playable CreatePlayable (PlayableGraph graph, GameObject owner)
    {
        Debug.Log ("create playable");
        var playable = ScriptPlayable<MySuperPlayableBehaviour>.Create (graph, template);
        MySuperPlayableBehaviour clone = playable.GetBehaviour ();
        myCamera.exposedName = GetInstanceID().ToString();
        clone.myCamera = myCamera.Resolve (graph.GetResolver ());
        return playable;
    }
}

```

expose变量必须初始化

```

using UnityEngine.Playables;
using UnityEngine.Timeline;

[Serializable]
public class MySuperPlayableClip : PlayableAsset, ITimelineClipAsset

{
    public MySuperPlayableBehaviour template = new MySuperPlayableBehaviour ();
    public ExposedReference<GameObject> myCamera;

    public ClipCaps clipCaps
    {
        get { return ClipCaps.Blending; }
    }

    public override Playable CreatePlayable (PlayableGraph graph, GameObject owner)
    {
        Debug.Log ("create playable");
        var playable = ScriptPlayable<MySuperPlayableBehaviour>.Create (graph, template);
        MySuperPlayableBehaviour clone = playable.GetBehaviour ();
        myCamera.exposedName = GetInstanceID().ToString();
        clone.myCamera = myCamera.Resolve (graph.GetResolver ());
        return playable;
    }
}

```

ExposeReference 动态获取方式

```

47
48
49 public void PlayFrontStab(string timelineName, ActorManager attacker, ActorManager victim){
50     if (timelineName == "frontStab") {
51         pd.playableAsset = Instantiate(frontStab);
52
53         TimelineAsset timeline = (TimelineAsset)pd.playableAsset;
54         //Debug.Log (timeline.name);
55
56         pd.Evaluate ();
57
58         foreach (var track in timeline.GetOutputTracks()) {
59             if (track.name == "Attacker Script") {
60                 pd.SetGenericBinding (track, attacker);
61                 foreach (var clip in track.GetClips()) {
62                     MySuperPlayableClip myclip = (MySuperPlayableClip)clip.asset;
63                     MySuperPlayableBehaviour mybehav = myclip.template;
64                     mybehav.myFloat = 777;
65                     pd.SetReferenceValue (myclip.myCamera.exposedName, GameObject.Find("A"));
66                     Debug.Log (myclip.myCamera.exposedName);
67                     //mybehav.myCamera = GameObject.Find ("A");
68                 }
69             }
70             else if (track.name == "Victim Script") {
71                 pd.SetGenericBinding (track, victim);
72                 foreach (var clip in track.GetClips()) {
73                     MySuperPlayableClip myclip = (MySuperPlayableClip)clip.asset;
74                     MySuperPlayableBehaviour mybehav = myclip.template;
75                     mybehav.myFloat = 666666;
76                     pd.SetReferenceValue (myclip.myCamera.exposedName, GameObject.Find("B"));
77                     Debug.Log (myclip.myCamera.exposedName);
78                 }
79             }
80             else if (track.name == "Attacker Animation") {
81                 pd.SetGenericBinding (track, attacker.ac.anim);
82             }
83             else if (track.name == "Victim Animation") {
84                 pd.SetGenericBinding (track, victim.ac.anim);
85             }
86         }
87     }
88 }

```

给exposeRenfence变量动态赋值，pd是playableDirector