**PhaseChange.cs**

using UnityEngine;

using System.Collections;

//This method PhaseChange gets data from KinectManager.cs and set up different phase values, which is used by other scripts.

public class PhaseChange : MonoBehaviour {

//PhaseInput1 and PhaseInput2 are used to determine if the Kinect is getting data that switch phases

public int PhaseInput1;

public int PhaseInput2;

//Phase is the actual value that determine the phase

public int Phase;

//Not used now

public float test;

//timer1 and timer2 are used to make sure that phase changes under the condition that the data getting from Kinect are consistent for over a certain amount of time

private float timer1;

private float timer2;

// Use this for initialization

void Start () {

}

// Update is called once per frame

void FixedUpdate() {

//Start the timer when getting data from Kinect

if (PhaseInput1 == 1)

{

timer1 = timer1 + Time.deltaTime;

}

if (PhaseInput2 == 1)

{

timer2 = timer2 + Time.deltaTime;

}

//Dump the timer when no phase changing is happening

if (PhaseInput1 == 0)

{

timer1 = 0;

}

if (PhaseInput2 == 0)

{

timer2 = 0;

}

//Trigger change on value of Phase if the Kinect streams consistent data over 1 second

if (timer1 > 1)

{

Phase = 1;

}

if (timer2 > 1)

{

Phase = 2;

}

}

}

**Transformation.cs**

using UnityEngine;

using System.Collections;

//The method Transformation extract data from PhaseChange.cs and gets data from KinectManager.cs, determine whether it's going to transform from male to female or female to male or stay the same, then stream result into DoubleSideShaderMorph.cs and DoubleSideShaderMorph1.cs

public class Transformation : MonoBehaviour

{

//MaleGesture and FemaleGesture are used to gather data from KinectManager.cs to determine whether the Kinect is recognizing male gesture or female gesture

public bool MaleGesture;

public bool FemaleGesture;

//Access the two scripts DoubleSideShaderMorph.cs and DoubleSideShaderMorph1.cs

//I do not remember why I defined two different variables for the same scripts, I am not sure whether if it's necessary or if it's actually doing something

public DoubleSideShaderMorph DoubleSideShaderMorph\_lower;

public DoubleSideShaderMorph DoubleSideShaderMorph\_upper;

public DoubleSideShaderMorph1 DoubleSideShaderMorph1\_lower;

public DoubleSideShaderMorph1 DoubleSideShaderMorph1\_upper;

//Access PhhaseChange.cs

public PhaseChange PhaseChange;

//KeyCheck is used in DoubleSideShaderMorph.cs and DoubleSideShaderMorph1.cs to control model transformation between male version and female version

private float KeyCheck;

//A set of timers are used to determine whether a pose is hold long enough to trigger the event. Thus important events won't be triggered by random poses

public float timer1;

public float dumptimer1;

public float timer2;

public float dumptimer2;

//Phase is the local variable that matches PhaseChange.cs's Phase value

private int Phase;

// Use this for initialization

void Start()

{

}

// Update is called once per frame

void FixedUpdate()

{

//Extract Phase value from PhaseChange.cs and apply it to local variable Phase

Phase = PhaseChange.Phase;

//Phase 1 events

if (Phase == 1)

{

//If Kinect regonizes a pose as malegesture, timer1 starts and reset dumptimer1

if (MaleGesture)

{

timer1 = timer1 + Time.deltaTime;

dumptimer1 = 0;

}

//If Kinect records a non-malegesture pose the dumptimer starts, this was create to prevent Kinect give random false value which result restarting timer1

if (MaleGesture != true)

{

dumptimer1 = dumptimer1 + Time.deltaTime;

}

//If the the false value is streamed from Kinect for more than half seconds also it does not get dumped cause there is no male gesture being recorded then it's considered a "real" false value, when that happens reset timer1

if (dumptimer1 > 0.5)

{

timer1 = 0;

}

//If the male gesture is held for more than 1 second, then proceed to next step

if (timer1 > 1)

{

//Checking if the model is already in "male state", if yes, do nothing. If no, access DoubleSideShaderMorph.cs and DoubleSideShaderMorph1.cs to change the KeyCheck value to 1 to initiate transformation from female state to male state

if ((DoubleSideShaderMorph\_lower.blendshape < 100) & (DoubleSideShaderMorph\_lower.blendshape > 0) & (DoubleSideShaderMorph\_upper.blendshape < 100) & (DoubleSideShaderMorph\_upper.blendshape > 0))

{

return;

}

else

{

DoubleSideShaderMorph\_lower.KeyCheck = 1;

DoubleSideShaderMorph\_upper.KeyCheck = 1;

DoubleSideShaderMorph1\_lower.KeyCheck = 1;

DoubleSideShaderMorph1\_upper.KeyCheck = 1;

}

}

//If Kinect regonizes a pose as malegesture, timer2 starts and reset dumptimer2

if (FemaleGesture)

{

timer2 = timer2 + Time.deltaTime;

dumptimer2 = 0;

}

//If Kinect records a non-malegesture pose the dumptimer starts, this was create to prevent Kinect give random false value which result restarting timer2

if (FemaleGesture != true)

{

dumptimer2 = dumptimer2 + Time.deltaTime;

}

//If the the false value is streamed from Kinect for more than half seconds also it does not get dumped cause there is no male gesture being recorded then it's considered a "real" false value, when that happens reset timer2

if (dumptimer2 > 0.5)

{

timer2 = 0;

}

//If the female gesture is held for more than 1 second, then proceed to next step

if (timer2 > 1)

{

//Checking if the model is already in "female state", if yes, do nothing. If no, access DoubleSideShaderMorph.cs and DoubleSideShaderMorph1.cs to change the KeyCheck value to 1 to initiate transformation from male state to female state

if ((DoubleSideShaderMorph\_lower.blendshape < 100) & (DoubleSideShaderMorph\_lower.blendshape > 0) & (DoubleSideShaderMorph\_upper.blendshape < 100) & (DoubleSideShaderMorph\_upper.blendshape > 0))

{

return;

}

{

DoubleSideShaderMorph\_lower.KeyCheck = 2;

DoubleSideShaderMorph\_upper.KeyCheck = 2;

DoubleSideShaderMorph1\_lower.KeyCheck = 2;

DoubleSideShaderMorph1\_upper.KeyCheck = 2;

}

}

}

//Phase 2 events, these are for testing purpose only now, no pratical functions yet.

if (Phase == 2)

{

DoubleSideShaderMorph1\_lower.KeyCheck = 1;

DoubleSideShaderMorph1\_upper.KeyCheck = 1;

}

}

}