**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;

}

}

}

**DoubleSideShaderMorph.cs**

using UnityEngine;

using System.Collections;

using System.Collections.Generic;

//The method of DoubleSideShaderMorph is used along with DoubleSideShaderMorph1 to achieve the function of a female model transforming into a male model

public class DoubleSideShaderMorph : MonoBehaviour {

//KeyCheck is used in DoubleSideShaderMorph.cs and DoubleSideShaderMorph1.cs to control model transformation between male version and female version, 0 is default/initiate value, 1 is female to male process, 2 is male to female process

public int KeyCheck;

//speedcoefficient determines how fast the metamorphosis happens, speedcoefficient>=0

public float speedcoefficient;

//blendshape is the blendshape value accessed from the model, 0-(0-1)<=blendshape<=100+(0~1)

public float blendshape;

//Access PhhaseChange.cs

public PhaseChange PhaseChange;

//texturemorph is the transparency of the shader, 0<=texturemorph<=1

private float texturemorph;

//Access the shader of the object

private Renderer rend;

//blendshapespeed is the final value of how fast the metamorphosis happens after speedcoefficient is multiplied by 100 times so that the change speed is fast

private float blendshapespeed;

//Access the model's blendshape value

private SkinnedMeshRenderer skinnedMeshRenderer;

//textureprogress is used to calculate the value of texturemorph

private float textureprogress;

//Phase is the value that determine the what event to happen under different values, its value is extracted from PhaseChange.cs

private int Phase;

//Set default values to variables and assign variables to accessed scripts

void Start () {

//Access PhaseChange.cs to extract the value of its Phase to local

Phase = PhaseChange.Phase;

//Assign default value to Keycheck

KeyCheck = 0;

//Access the value of user input changespeed and multiply it by 100 times to make it work to the script

blendshapespeed = 100 \* speedcoefficient;

//Assign rend to access the shader of the object

rend = GetComponent<Renderer>();

//This was used to make sure that the transparent value of the shader was assigned, but is no longer needed

//rend.material.shader = Shader.Find("Ciconia Studio/Double Sided/Transparent/Diffuse Bump");

//Assign skinnedMeshRenderer to access the model's blendshape value

skinnedMeshRenderer = GetComponent<SkinnedMeshRenderer>();

}

// Update is called once per frame

// Check the value of KeyCheck every frame to confirm the transformation (blendshapes and texture)

void FixedUpdate() {

//For debug purpose, allows keyboard input to change the value of KeyCheck

if (Input.GetKeyDown("1"))

{

KeyCheck = 1;

}

if (Input.GetKeyDown("2"))

{

KeyCheck = 2;

}

//The defalut Phase, right now there's only onw phase

if (Phase == 1)

{

//KeyCheck is 1 which means the process is to transform to male model and texture from female ones

if (KeyCheck == 1)

{

//If the blendshape value of the model is over 100, which means the model is already in male state, then reset the KeyCheck and textureprogress

//It should be blendshape >= 0

if (blendshape > 100)

{

KeyCheck = 0;

textureprogress = 0;

}

//If it's not in male state, then this happens

else

{

//Change the blendshape value based on deltaTime from 0 to 100

blendshape = blendshape + blendshapespeed \* Time.deltaTime;

//Assign the value to the model

skinnedMeshRenderer.SetBlendShapeWeight(0, blendshape);

//Change the textureprogress value based on deltaTime

textureprogress = textureprogress + speedcoefficient \* Time.deltaTime;

//Transparency value goes from 1 to 0, transparency value of Female\_Torso\_Texture\_1's shader should be on 1 when the model is in female states, it should be on 0 when the model is in male states

texturemorph = Mathf.Lerp(1f, 0f, textureprogress);

//Assign the value of shader transparency value to the shader

rend.material.SetFloat("\_Transparency", texturemorph);

}

}

//KeyCheck is 2 which means the process is to transform to female model and texture from male ones

if (KeyCheck == 2)

{

//If the blendshape value of the model is below 100, which means the model is already in female state, then reset the KeyCheck and textureprogress

//It should be blendshape <= 0

if (blendshape < 0)

{

KeyCheck = 0;

textureprogress = 0;

}

//If it's not in male state, then this happens

else

{

//Change the blendshape value based on deltaTime from 100 to 0

blendshape = blendshape - blendshapespeed \* Time.deltaTime;

//Assign the value to the model

skinnedMeshRenderer.SetBlendShapeWeight(0, blendshape);

//Change the textureprogress value based on deltaTime

textureprogress = textureprogress + speedcoefficient \* Time.deltaTime;

//Transparency value goes from 0 to 1, transparency value of Female\_Torso\_Texture\_1's shader should be on 1 when the model is in female states, it should be on 0 when the model is in male states

texturemorph = Mathf.Lerp(0f, 1f, textureprogress);

//Assign the value of shader transparency value to the shader

rend.material.SetFloat("\_Transparency", texturemorph);

}

}

}

// not filled in yet, waiting for determined activity to be put into production

if (Phase == 2)

{

}

}

}

**DoubleSideShaderMorph1.cs**

using UnityEngine;

using System.Collections;

using System.Collections.Generic;

//The method of DoubleSideShaderMorph1 is modified from DoubleSideShaderMorph to achieve the function of a male model transforming into a female model

public class DoubleSideShaderMorph1 : MonoBehaviour {

//KeyCheck is used in DoubleSideShaderMorph.cs and DoubleSideShaderMorph1.cs to control model transformation between male version and female version, 0 is default/initiate value, 1 is female to male process, 2 is male to female process

public int KeyCheck;

//speedcoefficient determines how fast the metamorphosis happens, speedcoefficient>=0

public float speedcoefficient;

//blendshape is the blendshape value accessed from the model, 0-(0-1)<=blendshape<=100+(0~1)

public float blendshape;

//Access PhhaseChange.cs

public PhaseChange PhaseChange;

//texturemorph is the transparency of the shader, 0<=texturemorph<=1

private float texturemorph;

//Access the shader of the object

private Renderer rend;

//blendshapespeed is the final value of how fast the metamorphosis happens after speedcoefficient is multiplied by 100 times so that the change speed is fast

private float blendshapespeed;

//Access the model's blendshape value

private SkinnedMeshRenderer skinnedMeshRenderer;

//textureprogress is used to calculate the value of texturemorph

private float textureprogress;

//Phase is the value that determine the what event to happen under different values, its value is extracted from PhaseChange.cs

private int Phase;

//Set default values to variables and assign variables to accessed scripts

void Start () {

//Access PhaseChange.cs to extract the value of its Phase to local

Phase = PhaseChange.Phase;

//Assign default value to Keycheck

KeyCheck = 0;

//Access the value of user input changespeed and multiply it by 100 times to make it work to the script

blendshapespeed = 100 \* speedcoefficient;

//Assign rend to access the shader of the object

rend = GetComponent<Renderer>();

//This was used to make sure that the transparent value of the shader was assigned, but is no longer needed

//rend.material.shader = Shader.Find("Ciconia Studio/Double Sided/Transparent/Diffuse Bump");

//Assign skinnedMeshRenderer to access the model's blendshape value

skinnedMeshRenderer = GetComponent<SkinnedMeshRenderer>();

}

// Update is called once per frame

// Check the value of KeyCheck every frame to confirm the transformation (blendshapes and texture)

void FixedUpdate() {

//For debug purpose, allows keyboard input to change the value of KeyCheck

if (Input.GetKeyDown("1"))

{

KeyCheck = 1;

}

if (Input.GetKeyDown("2"))

{

KeyCheck = 2;

}

//The defalut Phase, right now there's only onw phase

if (Phase == 1)

{

//KeyCheck is 1 which means the process is to transform to male model and texture from female ones

if (KeyCheck == 1)

{

//If the blendshape value of the model is over 100, which means the model is already in male state, then reset the KeyCheck and textureprogress

//It should be blendshape >= 0

if (blendshape > 100)

{

KeyCheck = 0;

textureprogress = 0;

}

//If it's not in male state, then this happens

else

{

//Change the blendshape value based on deltaTime from 0 to 100

blendshape = blendshape + blendshapespeed \* Time.deltaTime;

//Assign the value to the model

skinnedMeshRenderer.SetBlendShapeWeight(0, blendshape);

//Change the textureprogress value based on deltaTime

textureprogress = textureprogress + speedcoefficient \* Time.deltaTime;

//Transparency value goes from 0 to 1, transparency value of Female\_Torso\_Texture\_2's shader should be on 0 when the model is in female states, it should be on 1 when the model is in male states

//This is the part that is different from DoubleSideShaderMorph.cs

texturemorph = Mathf.Lerp(0f, 1f, textureprogress);

//Assign the value of shader transparency value to the shader

rend.material.SetFloat("\_Transparency", texturemorph);

}

}

//KeyCheck is 2 which means the process is to transform to female model and texture from male ones

if (KeyCheck == 2)

{

//If the blendshape value of the model is below 100, which means the model is already in female state, then reset the KeyCheck and textureprogress

//It should be blendshape <= 0

if (blendshape < 0)

{

KeyCheck = 0;

textureprogress = 0;

}

//If it's not in male state, then this happens

else

{

//Change the blendshape value based on deltaTime from 100 to 0

blendshape = blendshape - blendshapespeed \* Time.deltaTime;

//Assign the value to the model

skinnedMeshRenderer.SetBlendShapeWeight(0, blendshape);

//Change the textureprogress value based on deltaTime

textureprogress = textureprogress + speedcoefficient \* Time.deltaTime;

//Transparency value goes from 1 to 0, transparency value of Female\_Torso\_Texture\_2's shader should be on 0 when the model is in female states, it should be on 1 when the model is in male states

//This is the part that is different from DoubleSideShaderMorph.cs

texturemorph = Mathf.Lerp(1f, 0f, textureprogress);

//Assign the value of shader transparency value to the shader

rend.material.SetFloat("\_Transparency", texturemorph);

}

}

}

// not filled in yet, waiting for determined activity to be put into production

if (Phase == 2)

{

}

}

}