Overview - Code Development

Rogue Fighter

Term Project for Computer Graphics (IS-F311)

Group Members

1. Nidheesh Jain - 2019A7PS0024P (Team Leader)
2. Ishaan Tiwari - 2017A3PS0866P
3. Dhruv Rawat - 2018A7PS0021P

Overview:

Initially a background scene was set up and an image background was set up in it. Player1, Player2 models were imported and set on the screen with a custom shader to cast shadows on the ground. Next phase was to set up the animations for each character. After setting up the animations we implemented the following movement, fight and transition codes. The last menu was developed at enter and exit the application.

Code:

HealthBars.cs

Implementing the Health System of the game. Characters have full health when the game starts and the match ends when the health of one player drops to 0. Light kick and light punch does a minor damage and Heavy kick and Heavy Punch does a major damage. When a fighter blocks, no damage is taken.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
```

```
public class HealthBars : MonoBehaviour
{
        public Image Player1Green;

        public Image Player2Green;

        public Image Player1Red;

        public Image Player2Red;

        // Start is called before the first frame update
        void Start()
        {
        }

        // Update is called once per frame
        void Update()
        {
        Player1Green.fillAmount = SaveScript.Player1Health;
        Player2Green.fillAmount = SaveScript.Player2Health;

        if (SaveScript.Player2Timer > 0)
        {
        SaveScript.Player2Timer -= 2.0f * Time.deltaTime;
        }
        if (SaveScript.Player1Timer > 0)
        {
        SaveScript.Player1Timer -= 2.0f * Time.deltaTime;
        }

        if (SaveScript.Player2Timer <= 0)
        {
        if (Player2Red.fillAmount > SaveScript.Player2Health)
        {
        Player2Red.fillAmount -= 0.003f;
        }
        }
        if (SaveScript.Player1Timer <= 0)
        {
        if (Player1Red.fillAmount > SaveScript.Player1Health)
        {
        Player1Red.fillAmount -= 0.003f;
        }
        }
        }
}
```

Player1Actions.cs

Implementing the physical actions of the fighter such as punches(heavy,light), kicks (heavy,light), blocking. Sound effects for these actions were also added.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player1Actions : MonoBehaviour
{
    public float JumpSpeed = 1.5f;

    public GameObject Player1;

    private Animator Anim;

    private AnimatorStateInfo Player1Layer0;

    private AudioSource MyPlayer;

    public AudioClip PunchWoosh;

    public AudioClip KickWoosh;

    public static bool Hits = false;

    // Start is called before the first frame update
    void Start()
    {
    Anim = GetComponent<Animator>();
    MyPlayer = GetComponent<AudioSource>();
    }

    // Update is called once per frame
    void Update()
    {
    // Get Animator
    Player1Layer0 = Anim.GetCurrentAnimatorStateInfo(0);

    if (Player1Layer0.IsTag("Motion"))
    {
    if (Input.GetButtonDown("Fire1"))
    {
    Anim.SetTrigger("LightPunch");
    Hits = false;
    }
    if (Input.GetButtonDown("Fire2"))
    {
    Anim.SetTrigger("HeavyPunch");
    Hits = false;
    }
    if (Input.GetButtonDown("Fire3"))
    {
    Anim.SetTrigger("LightKick");
    Hits = false;
    }
    if (Input.GetButtonDown("Fire4"))
    {
    Anim.SetTrigger("HeavyKick");
    Hits = false;
    }
    if (Input.GetButtonDown("Block"))
    {
    Anim.SetTrigger("BlockOn");
    }
    }
    if (Player1Layer0.IsTag("Block"))
    {
    if (Input.GetButtonUp("Block"))
    {
    Anim.SetTrigger("BlockOff");
    }
    }
    }

    public void PunchWooshSound()
    {
    MyPlayer.clip = PunchWoosh;
    MyPlayer.Play();
    }

    public void KickWooshSound()
    {
    MyPlayer.clip = KickWoosh;
    MyPlayer.Play();
    }

    public void JumpUp()
    {
    //Player1.transform.Translate(0,JumpSpeed,0);
    //StartCoroutine(JumpDelay());
    }
```

```
	// IEnumerator JumpDelay()
	// {
	//		//yield return new WaitForSeconds(0.2f);
	//		//Player1.transform.Translate(0,-JumpSpeed,0);
	// }
}
```

Player1Move.cs

Implementing the movement of the fighters. Movement includes crouch, jump and normal x-axis movement. The players can not move out of the stage as well

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player1Move : MonoBehaviour
{
	private Animator Anim;

	public float WalkSpeed = 0.05f;

	private bool isJumping = false;

	private bool isCrouch = false;

	private AnimatorStateInfo Player1Layer0;

	private bool CanWalkLeft = true;

	private bool CanWalkRight = true;

	public GameObject Player1;

	public GameObject Opponent;

	private Vector3 OppPosition;

	private bool FacingLeft = false;

	private bool FacingRight = true;

	public AudioClip LightPunch;
public AudioClip HeavyPunch;

public AudioClip LightKick;

public AudioClip HeavyKick;

private AudioSource MyPlayer;

// public GameObject Restrict; // #60
public Rigidbody RB;

// public Collider BoxCollider;

public Collider CapsuleCollider;

// Start is called before the first frame update
void Start()
{
Anim = GetComponentInChildren<Animator>();
StartCoroutine(FaceRight());
MyPlayer = GetComponentInChildren<AudioSource>();
}

// Update is called once per frame
void Update()
{
// Check if knocked output
if (SaveScript.Player1Health <= 0)
{
Anim.SetTrigger("KnockedOut");
Player1.GetComponent<Player1Actions>().enabled = false;

// this.GetComponent<Player1Move>.enabled = false;
StartCoroutine(KnockedOut());
}

if (SaveScript.Player2Health <= 0)
{
Anim.SetTrigger("Victory");
Player1.GetComponent<Player1Actions>().enabled = false;
this.GetComponent<Player1Move>().enabled = false;
}

// Get Animator
Player1Layer0 = Anim.GetCurrentAnimatorStateInfo(0);
```

```
// Screen Bounds
Vector3 ScreenBounds =
Camera.main.WorldToScreenPoint(this.transform.position);

if (ScreenBounds.x > Screen.width - 200)
{
CanWalkRight = false;
}
if (ScreenBounds.x < 0 + 200)
{
CanWalkLeft = false;
}
else if (ScreenBounds.x > 200 && ScreenBounds.x < Screen.width - 200)
{
CanWalkRight = true;
CanWalkLeft = true;
}

// Flipping to Face Opponent
OppPosition = Opponent.transform.position;

//Facing Left or Right of the Opponent
if (OppPosition.x > Player1.transform.position.x)
{
StartCoroutine(FaceLeft());
}
if (OppPosition.x < Player1.transform.position.x)
{
StartCoroutine(FaceRight());
}

// Horizontal Axis
if (Player1Layer0.IsTag("Motion"))
{
if (Input.GetAxis("Horizontal") > 0)
{
if (CanWalkRight == true)
{
Anim.SetBool("Forward", true);
transform.Translate(WalkSpeed, 0, 0);
}
}
if (Input.GetAxis("Horizontal") < 0)
{
if (CanWalkLeft == true)
{
Anim.SetBool("Backward", true);
transform.Translate(-WalkSpeed, 0, 0);
}
}
}
if (Input.GetAxis("Horizontal") == 0)
{
Anim.SetBool("Forward", false);
Anim.SetBool("Backward", false);
}

// Vertical Axis
if (Input.GetAxis("Vertical") > 0)
{
if (isJumping == false)
{
isJumping = true;
Anim.SetTrigger("Jump");
StartCoroutine(JumpPause());
}
}
if (Input.GetAxis("Vertical") < 0)
{
if (isCrouch == false)
{
isCrouch = true;
Anim.SetTrigger("Crouch");
StartCoroutine(CrouchPause());
}
}

if (Player1Layer0.IsTag("Block"))
{
RB.isKinematic = true;
// BoxCollider.enabled = false;
CapsuleCollider.enabled = false;
}
else
{
// BoxCollider.enabled = false;
CapsuleCollider.enabled = false;
RB.isKinematic = true;
}
}

private void OnTriggerEnter(Collider other)
{
if (other.gameObject.CompareTag("FistLight"))
```

```
{
Anim.SetTrigger("HeadReact");
MyPlayer.clip = LightPunch;
MyPlayer.Play();
}
if (other.gameObject.CompareTag("FistHeavy"))
{
Anim.SetTrigger("BigReact");
MyPlayer.clip = HeavyPunch;
MyPlayer.Play();
}
if (other.gameObject.CompareTag("KickLight"))
{
Anim.SetTrigger("HeadReact");
MyPlayer.clip = LightPunch;
MyPlayer.Play();
}
if (other.gameObject.CompareTag("KickHeavy"))
{
Anim.SetTrigger("BigReact");
MyPlayer.clip = HeavyPunch;
MyPlayer.Play();
}
}

IEnumerator JumpPause()
{
yield return new WaitForSeconds(1.0f);
isJumping = false;
}

IEnumerator CrouchPause()
{
yield return new WaitForSeconds(1.0f);
isCrouch = false;
}

IEnumerator FaceLeft()
{
if (FacingLeft == true)
{
FacingLeft = false;
FacingRight = true;
yield return new WaitForSeconds(0.15f);
Player1.transform.Rotate(0, -180, 0);
Anim.SetLayerWeight(1, 0);
}
```

```
    }

    IEnumerator FaceRight()
    {
    if (FacingRight == true)
    {
    FacingLeft = true;
    FacingRight = false;
    yield return new WaitForSeconds(0.15f);
    Player1.transform.Rotate(0, 180, 0);
    Anim.SetLayerWeight(1, 1);
    }
    }

    IEnumerator KnockedOut()
    {
    yield return new WaitForSeconds(0.5f);
    this.GetComponent<Player1Move>().enabled = false;
    }
}
```

Player1Trigger.cs

This file enables the fighter to further damage the opponent. Without this file, the opponent doesn’t register the hits after the first one.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player1Trigger : MonoBehaviour
{
    public Collider Col;

    public float DamageAmt = 0.1f;

    // Update is called once per frame
    void Update()
    {
    if (Player1Actions.Hits == false)
    {
    Col.enabled = true;
    }
    else
    {
    Col.enabled = false;
```

```
	}
	}

	private void OnTriggerEnter(Collider other)
	{
	if (other.gameObject.CompareTag("Player2"))
	{
	// Col.enabled = false;
	Player1Actions.Hits = true;
	SaveScript.Player2Health -= DamageAmt;
	if (SaveScript.Player2Timer < 2.0f)
	{
	SaveScript.Player2Timer += 2.0f;
	}
	}
	}
}
```

Player2Actions.cs

Same as Above for the second player.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player2Actions : MonoBehaviour
{
	public float JumpSpeed = 1.5f;

	public GameObject Player1;

	private Animator Anim;

	private AnimatorStateInfo Player1Layer0;

	private AudioSource MyPlayer;

	public AudioClip PunchWoosh;

	public AudioClip KickWoosh;

	public static bool HitsP2 = false;

	// Start is called before the first frame update
	void Start()
	{
```

```
Anim = GetComponent<Animator>();
MyPlayer = GetComponent<AudioSource>();
}

// Update is called once per frame
void Update()
{
// Get Animator
Player1Layer0 = Anim.GetCurrentAnimatorStateInfo(0);

if (Player1Layer0.IsTag("Motion"))
{
if (Input.GetButtonDown("Fire1P2"))
{
Anim.SetTrigger("LightPunch");
HitsP2 = false;
}
if (Input.GetButtonDown("Fire2P2"))
{
Anim.SetTrigger("HeavyPunch");
HitsP2 = false;
}
if (Input.GetButtonDown("Fire3P2"))
{
Anim.SetTrigger("LightKick");
HitsP2 = false;
}
if (Input.GetButtonDown("Fire4P2"))
{
Anim.SetTrigger("HeavyKick");
HitsP2 = false;
}
if (Input.GetButtonDown("BlockP2"))
{
Anim.SetTrigger("BlockOn");
}
}
if (Player1Layer0.IsTag("Block"))
{
if (Input.GetButtonUp("BlockP2"))
{
Anim.SetTrigger("BlockOff");
}
}
}

public void PunchWooshSound()
```

```
    {
    MyPlayer.clip = PunchWoosh;
    MyPlayer.Play();
    }

    public void KickWooshSound()
    {
    MyPlayer.clip = KickWoosh;
    MyPlayer.Play();
    }

    public void JumpUp()
    {
    //Player1.transform.Translate(0,JumpSpeed,0);
    //StartCoroutine(JumpDelay());
    }

    // IEnumerator JumpDelay()
    // {
    //      //yield return new WaitForSeconds(0.2f);
    //      //Player1.transform.Translate(0,-JumpSpeed,0);
    // }
}
```

Player2Move.cs

Same as Above for the second player.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player2Move : MonoBehaviour
{
    private Animator Anim;

    public float WalkSpeed = 0.05f;

    private bool isJumping = false;

    private bool isCrouch = false;

    private AnimatorStateInfo Player1Layer0;

    private bool CanWalkLeft = true;

    private bool CanWalkRight = true;
```

```
public GameObject Player1;

public GameObject Opponent;

private Vector3 OppPosition;

private bool FacingLeft = false;

private bool FacingRight = true;

public AudioClip LightPunch;

public AudioClip HeavyPunch;

public AudioClip LightKick;

public AudioClip HeavyKick;

private AudioSource MyPlayer;

// public GameObject Restrict; // #60
public Rigidbody RB;

// public Collider BoxCollider;

public Collider CapsuleCollider;

// Start is called before the first frame update
void Start()
{
Anim = GetComponentInChildren<Animator>();
StartCoroutine(FaceRight());
MyPlayer = GetComponentInChildren<AudioSource>();
}

// Update is called once per frame
void Update()
{
// Check if knocked output
if (SaveScript.Player2Health <= 0)
{
Anim.SetTrigger("KnockedOut");
Player1.GetComponent<Player2Actions>().enabled = false;

// this.GetComponent<Player2Move>.enabled = false;
StartCoroutine(KnockedOut());
```

```
}
if (SaveScript.Player1Health <= 0)
{
Anim.SetTrigger("Victory");
Player1.GetComponent<Player2Actions>().enabled = false;
this.GetComponent<Player2Move>().enabled = false;
}

// Get Animator
Player1Layer0 = Anim.GetCurrentAnimatorStateInfo(0);

// Screen Bounds
Vector3 ScreenBounds =
Camera.main.WorldToScreenPoint(this.transform.position);

if (ScreenBounds.x > Screen.width - 200)
{
CanWalkRight = false;
}
if (ScreenBounds.x < 0 + 200)
{
CanWalkLeft = false;
}
else if (ScreenBounds.x > 200 && ScreenBounds.x < Screen.width - 200)
{
CanWalkRight = true;
CanWalkLeft = true;
}

// Flipping to Face Opponent
OppPosition = Opponent.transform.position;

//Facing Left or Right of the Opponent
if (OppPosition.x > Player1.transform.position.x)
{
StartCoroutine(FaceLeft());
}
if (OppPosition.x < Player1.transform.position.x)
{
StartCoroutine(FaceRight());
}

// Horizontal Axis
if (Player1Layer0.IsTag("Motion"))
{
if (Input.GetAxis("HorizontalP2") > 0)
{
```

```
if (CanWalkRight == true)
{
Anim.SetBool("Forward", true);
transform.Translate(WalkSpeed, 0, 0);
}
}
if (Input.GetAxis("HorizontalP2") < 0)
{
if (CanWalkLeft == true)
{
Anim.SetBool("Backward", true);
transform.Translate(-WalkSpeed, 0, 0);
}
}
}
if (Input.GetAxis("HorizontalP2") == 0)
{
Anim.SetBool("Forward", false);
Anim.SetBool("Backward", false);
}

// Vertical Axis
if (Input.GetAxis("VerticalP2") > 0)
{
if (isJumping == false)
{
isJumping = true;
Anim.SetTrigger("Jump");
StartCoroutine(JumpPause());
}
}
if (Input.GetAxis("VerticalP2") < 0)
{
if (isCrouch == false)
{
isCrouch = true;
Anim.SetTrigger("Crouch");
StartCoroutine(CrouchPause());
}
}

if (Player1Layer0.IsTag("Block"))
{
RB.isKinematic = true;
// BoxCollider.enabled = false;
CapsuleCollider.enabled = false;
}
```

```
else
{
// BoxCollider.enabled = false;
CapsuleCollider.enabled = false;
RB.isKinematic = true;
}
}

private void OnTriggerEnter(Collider other)
{
if (other.gameObject.CompareTag("FistLight"))
{
Anim.SetTrigger("HeadReact");
MyPlayer.clip = LightPunch;
MyPlayer.Play();
}
if (other.gameObject.CompareTag("FistHeavy"))
{
Anim.SetTrigger("BigReact");
MyPlayer.clip = HeavyPunch;
MyPlayer.Play();
}
if (other.gameObject.CompareTag("KickLight"))
{
Anim.SetTrigger("HeadReact");
MyPlayer.clip = LightPunch;
MyPlayer.Play();
}
if (other.gameObject.CompareTag("KickHeavy"))
{
Anim.SetTrigger("BigReact");
MyPlayer.clip = HeavyPunch;
MyPlayer.Play();
}
}

IEnumerator JumpPause()
{
yield return new WaitForSeconds(1.0f);
isJumping = false;
}

IEnumerator CrouchPause()
{
yield return new WaitForSeconds(1.0f);
isCrouch = false;
}
```

```
    IEnumerator FaceLeft()
    {
    if (FacingLeft == true)
    {
    FacingLeft = false;
    FacingRight = true;
    yield return new WaitForSeconds(0.15f);
    Player1.transform.Rotate(0, -180, 0);
    Anim.SetLayerWeight(1, 0);
    }
    }

    IEnumerator FaceRight()
    {
    if (FacingRight == true)
    {
    FacingLeft = true;
    FacingRight = false;
    yield return new WaitForSeconds(0.15f);
    Player1.transform.Rotate(0, 180, 0);
    Anim.SetLayerWeight(1, 1);
    }
    }

    IEnumerator KnockedOut()
    {
    yield return new WaitForSeconds(0.5f);
    this.GetComponent<Player2Move>().enabled = false;
    }
}
```

Player2Trigger.cs

Same as Above for the second player.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player2Trigger : MonoBehaviour
{
    public Collider Col;

    public float DamageAmt = 0.1f;

    // Update is called once per frame
    void Update()
    {
    if (Player2Actions.HitsP2 == false)
    {
    Col.enabled = true;
    }
    else
    {
    Col.enabled = false;
    }
    }

    private void OnTriggerEnter(Collider other)
    {
    if (other.gameObject.CompareTag("Player1"))
    {
    // Col.enabled = false;
    Player2Actions.HitsP2 = true;
    SaveScript.Player1Health -= DamageAmt;
    if (SaveScript.Player1Timer < 2.0f)
    {
    SaveScript.Player1Timer += 2.0f;
    }
    }
    }
}
```

SaveScript.cs

Describes the decrement time for the health bar animation.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SaveScript : MonoBehaviour
{
    public static float Player1Health = 1.0f;

    public static float Player2Health = 1.0f;

    public static float Player1Timer = 2.0f;

    public static float Player2Timer = 2.0f;

    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
    }
}
```