

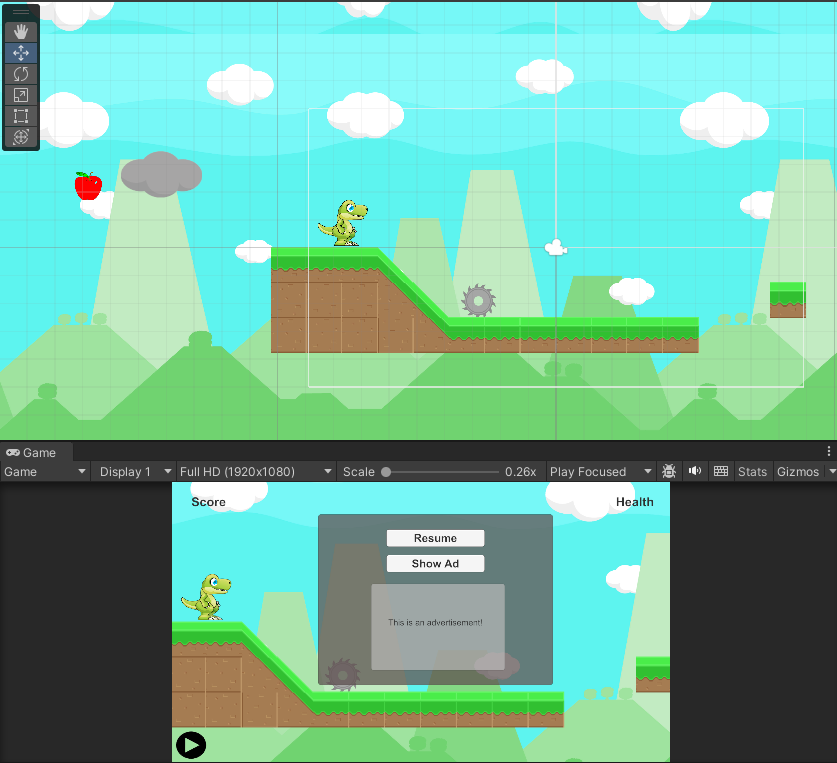
**COMSATS University Islamabad (Lahore Campus)**

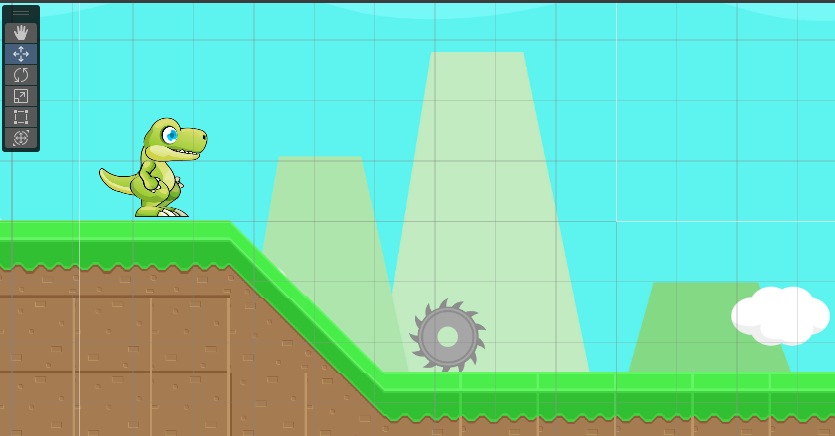
**<Lab Assignment 3> – SPRING 2024**

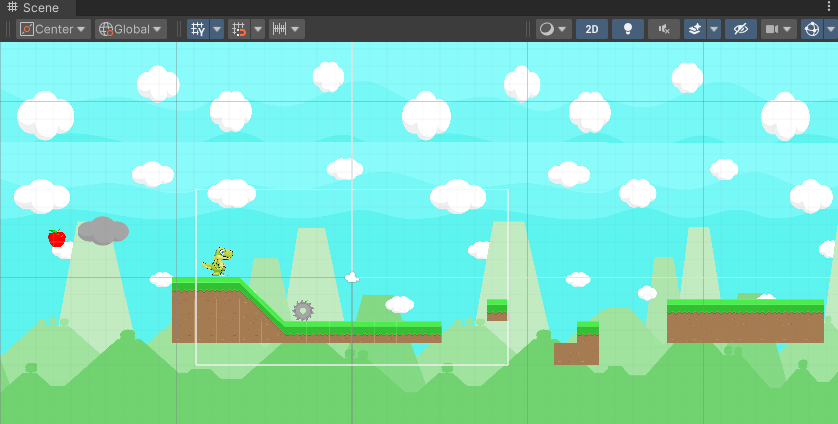
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Course Title: | Game Development | Course Code: | CSC495 | Credit Hours: | 3 |
| Course Instructor/s: | Saira Aslam | Program | BSE | | |
| **Submission Deadline** | **11:55 pm, 01-5-2024** | **Maximum Marks:** | **50** | | |
| **Important Instructions / Guidelines:**   * Submit a 10 seconds video, 4 screenshots, and all your scripts in Lab Assignment 3 submission at google classroom * Paste your screenshots and scripts inside Lab Assignment-1 submission file provided | | | | | |

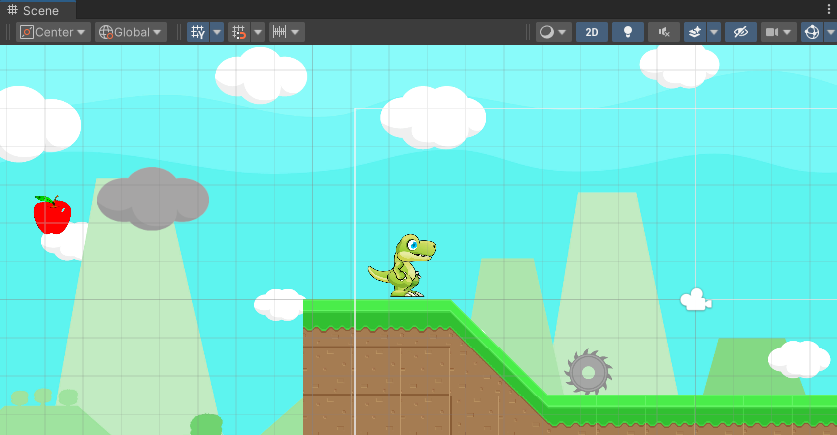
***Question No 1.***

***CLO: <3>; Bloom Taxonomy Level: <****Apply****>* [80]**



****

****

****

**dinoMove.cs**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;

public class dinoMove : MonoBehaviour

{

    Animator anim;

    public CharacterController2D controller;

    float lefthorizontalmove = -10f;

    float righthorizontalmove = 10f;

    bool jump = false;

    bool croach = false;

    public Text ScoreText;

    public GameObject gameOverPanel;

    private float score = 0;

    private float health = 100;

    public Button pauseButton;

    public Slider healthBar;

    float startTouchPosition,endTouchPosition;

    public Text WinningText;

    // Start is called before the first frame update

    void Start()

    {

        anim = GetComponent<Animator>();

        // HealthText.text = "Health: "+health.ToString();

        healthBar.value = health;

        ScoreText.text = "Score: "+score.ToString();

        gameOverPanel.SetActive(false);

    }

    void Update()

    {

        // if(Input.GetKey(KeyCode.Space))

        // {

        //     transform.Translate(Vector2.up \* 20 \*Time.fixedDeltaTime);

        // }

        if(transform.position.x >= 42f)

        {

            Vector2 pos = transform.position;

            pos.x = -6.8f;

            transform.position = pos;

        }

        if(health<=0 || transform.position.y <= -6.5) //pause the game and show game over scene

        {

            pauseButton.interactable = false;

            //Time.timeScale = 0;

            gameOverPanel.SetActive(true);

        }

        for(int i=0;i<Input.touchCount;i++)

        {

            var touch = Input.GetTouch(i);

            if(touch.phase == TouchPhase.Began)

            {

                startTouchPosition = touch.position.y;

            }

            else if(touch.phase == TouchPhase.Ended)

            {

                endTouchPosition = touch.position.y;

            }

            if(endTouchPosition > startTouchPosition)

            {

                transform.Translate(Vector2.up \* 20 \* Time.fixedDeltaTime);

            }

        }

        if(score >= 100)

        {

            Time.timeScale = 0;

            WinningText.text = "You Won!!";

            gameOverPanel.SetActive(true);

        }

    }

    // Update is called once per frame

    void FixedUpdate()

    {

        // if(Input.GetKey(KeyCode.LeftArrow))

        // {

        //     controller.Move(lefthorizontalmove \*Time.fixedDeltaTime,croach,jump);

        //     anim.SetTrigger("walk");

        // }

        // if(Input.GetKey(KeyCode.RightArrow))

        // {

        //     controller.Move(righthorizontalmove \*Time.fixedDeltaTime,croach,jump);

        //     anim.SetTrigger("walk");

        // }

        // if(Input.GetKey(KeyCode.Space))

        // {

        //     anim.SetTrigger("jump");

        // }

        // else{

        //     anim.SetTrigger("idle");

        // }

        if(Input.touchCount > 0)

        {

            var touch = Input.GetTouch(0);

            if(touch.position.x < Screen.width/2) //left move

            {

                controller.Move(lefthorizontalmove \*Time.fixedDeltaTime,croach,jump);

                anim.SetTrigger("walk");

            }

            if(touch.position.x > Screen.width/2)

            {

                controller.Move(righthorizontalmove \*Time.fixedDeltaTime,croach,jump);

                anim.SetTrigger("walk");

            }

            else{

                anim.SetTrigger("idle");

            }

        }

    }

    private void OnCollisionEnter2D(Collision2D col)

    {

        if(col.gameObject.name.StartsWith("Saw"))

        {

            health -= 10;

            // HealthText.text = "Health: "+health.ToString();

            healthBar.value = health;

            if(health == 0)

            {

                anim.SetTrigger("dead");

            }

        }

        if(col.gameObject.CompareTag("cloud"))

        {

            transform.gameObject.transform.parent = col.gameObject.transform;

        }

        if (!col.gameObject.name.StartsWith("cloud"))

        {

            transform.gameObject.transform.parent = null;

        }

        if(col.gameObject.name.StartsWith("Apple") || col.gameObject.name.StartsWith("coin"))

        {

            score += 10;

            ScoreText.text = "Score: "+score.ToString();

            Destroy(col.gameObject);

        }

    }

    public void PlayAgain() //handler for play again button

    {

        //Time.timeScale = 1;

        gameOverPanel.SetActive(false);

        SceneManager.LoadScene(0);

    }

}

**sawMovement.cs**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class sawMovemement : MonoBehaviour

{

    public Vector2 pos1;

    Vector2 pos2;

    public Vector2 positionDiff = new Vector2(7.2f,0f);

    public float speed = 0.5f;

    // Start is called before the first frame update

    void Start()

    {

        pos1 = transform.position;

        pos2 = pos1 + positionDiff;

    }

    // Update is called once per frame

    void Update()

    {

        transform.position = Vector2.Lerp(pos1, pos2,Mathf.PingPong(Time.time\*speed,1.0f));

        transform.Rotate(0,0,3f);

    }

}

**cloudMovement.cs**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class cloudMovement : MonoBehaviour

{

    public Vector2 pos1;

    Vector2 pos2;

    public Vector2 positionDiff = new Vector2(70f,0f);

    public float speed = 0.5f;

    // Start is called before the first frame update

    void Start()

    {

        pos1 = transform.position;

        pos2 = pos1 + positionDiff;

    }

    // Update is called once per frame

    void Update()

    {

        transform.position = Vector2.Lerp(pos1, pos2,Mathf.PingPong(Time.time\*speed,1.0f));

    }

}

**spawnManager.cs**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class appleController : MonoBehaviour

{

    public GameObject apple,coin;

    // Start is called before the first frame update

    void Start()

    {

        for(int i = 0;i<=10;i++){

            Vector2 applePos = new Vector2(Random.Range(0,50),Random.Range(-4,4));

            Instantiate(apple,applePos,Quaternion.identity);

        }

        for(int i = 0;i<=10;i++){

            Vector2 coinPos = new Vector2(Random.Range(0,50),Random.Range(-4,4));

            Instantiate(coin,coinPos,Quaternion.identity);

        }

    }

}

**UIManager.cs**

using UnityEngine;

using UnityEngine.UI;

public class UIManager : MonoBehaviour

{

    public GameObject MenuPanel;

    public GameObject AdPanel;

    public GameObject button;

    public Sprite playSprite;

    public Sprite pauseSprite;

    Image \_spriteRef;

    void Start()

    {

        \_spriteRef = button.GetComponent<Image>();

        MenuPanel.SetActive(false);

        AdPanel.SetActive(false);

    }

    public void PauseGame()

    {

        Time.timeScale = 0;

        MenuPanel.SetActive(true);

        \_spriteRef.sprite = pauseSprite;

    }

    public void ResumeGame()

    {

        Time.timeScale = 1;

        MenuPanel.SetActive(false);

        AdPanel.SetActive(false);

        \_spriteRef.sprite = playSprite;

    }

    public void ShowAdPanel()

    {

        AdPanel.SetActive(true);

    }

}