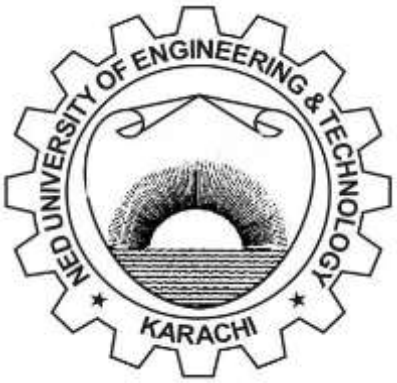


Object Oriented Concepts & Programming

Spring 2020

Project Report

Project name : HIT THE NET

Platform : UNITY (FOR INTERFACE)

Coding : VISUAL STUDIO (C#)

Members Collaborated :

1. Muhammad Hassan (SE-058)
2. Muhammad Sameed Abbasi (SE-059)
3. Fariya Khan (SE-066)
4. Fasih ur Rehman Ansari (SE-090)

Submitted to : ENGR. ASMA KHAN

FINAL PROJECT

GAME: HIT THE NET

THEME OF THE GAME

A fun and challenging first person's perspective game, build to attract audience with vibrant visuals and challenging levels, supported by life-like in-game mechanics and easy controls. The mission of the game is to throw the ball in the designated target (basket), which earns points. Scoring enough points in the predefined time restraint, allows ascension to the next level. Completion of the first (commemoratory practice) level introduces a defender in level 2. Defender stays in-between player and his target, stopping any direct or easy attempts. The final level stacks up the difficulty with Moving Targets. Playing the game makes one believer of the popular line:

"Easier said than Done."

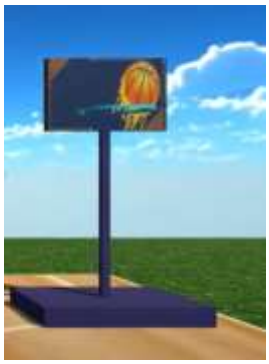
Can you do it? Just play and find out!

GAME OBJECTS:

- **Player:**
  A first person perspective player coded in such a way which allows simple player motions like moving in all directions, viewing 360^{o} and jumping.

- **Defender:**
  An opponent placed between the player and the basket in such a way that it follows the player movements and serves as a hindrance for the player to basket the ball easily.

- **Basket:**
  There are three baskets placed with different heights and the player can score in any of these. The baskets reduce to two and start moving in level 3 to make the game more challenging.



- **Ball:**
  A basketball which needs to be thrown through the hoop in order to score.

SCRIPTS:

Windows Form:

```
using System;
using System.Diagnostics;
using System.Threading;
using System.Windows.Forms;
using Hit_The_Ball_game;

namespace Hit_The_Ball_game
{
    public partial class Form1: Form
    {
        public Form1()
        {
            InitializeComponent();

        }
        System.Media.SoundPlayer bgmusic = new
System.Media.SoundPlayer(@"C:\Users\lenovo\Downloads\Game_bg_music.wav");
        System.Media.SoundPlayer buttonclick = new
System.Media.SoundPlayer(@"C:\Users\lenovo\Downloads\selection.wav");

        private void Form1_Load(object sender, EventArgs e)
        {
            bgmusic.Play();
        }

        private void button1_Click(object o, EventArgs eA)
        {
            buttonclick.Play();
            bgmusic.Stop();
            Process.Start(@"D:\unity\OOP Project basketball\sample game.exe");
        }

        private void button3_Click(object sender, EventArgs e)
        {
            buttonclick.Play();
            Thread.Sleep(300);              // delay
            this.Close();               //current form (this)
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form2 f2 = new Form2();
            f2.ShowDialog ();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Form3 f3 = new Form3();
            f3.ShowDialog();
        }
    }
}
```

First Person Player movement:

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

public class playermovement : MonoBehaviour
{
    public CharacterController controller;
    public float speed = 12f;
    public float gravity = -9.81f;
    public float jumpHeight = 3f;

    public Transform groundCheck;
    public float groundDistance = 0.4f;
    public LayerMask groundMask;

    Vector3 velocity;
    bool isGrounded;

    // Update is called once per frame
    void Update()
    {
        isGrounded = Physics.CheckSphere(groundCheck.position, groundDistance,
groundMask);

        if (isGrounded && velocity.y < 0)
        {
            velocity.y = -2f;
        }

        float x = Input.GetAxis("Horizontal");
        float z = Input.GetAxis("Vertical");

        Vector3 move = transform.right * x + transform.forward * z;

        controller.Move(move * speed * Time.deltaTime);

        if (Input.GetButtonDown("Jump") && isGrounded)
        {
            velocity.y = Mathf.Sqrt(jumpHeight * -2f * gravity); //PHY EQ 2gh
        }
    }
}
```

Throwing the ball, its physics and interaction with player:

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

public class player : MonoBehaviour
{
    public ball bol;
    public GameObject playercamera;
    public float balldistance = 2f;
    public float throwingballforce = 5f;
    static public int trie = 5;
```

```
    public int tries;
    public bool holdingball = true;

    // Start is called before the first frame update
    void Start()
    {
        bol.GetComponent<Rigidbody>().useGravity = false;
    }

    // Update is called once per frame
    void Update()
    {
        if (holdingball)
        {
            bol.transform.position = playercamera.transform.position +
playercamera.transform.forward * 2f;

            if (Input.GetMouseButtonDown(0))
            {
                trie-= 1;
                tries = trie;
                Debug.Log(tries); //it print in console

                holdingball = false;
                bol.activatetrail();
                bol.GetComponent<Rigidbody>().useGravity = true;
                bol.GetComponent<Rigidbody>().AddForce(playercamera.transform.forward *
throwingballforce);
            }
        }
    }
    public void resettries()
    {
        trie = 5;

    }
}
```

Managing the score and detecting movement of ball through the hoops:

```
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using System.Threading;
using Unity.Collections.LowLevel.Unsafe;
using System.IO;

public class scorearea : MonoBehaviour
{
    public GameObject effectobj;
    public static int score;
    public Text scoretext;
    public Text scoremsg;
    public AudioSource appsound;
    public float timer = 3f;

    void Start()
    {
        effectobj.SetActive(false);
```

```
        scoremsg.text = "";
        appsound = GetComponent<AudioSource>();
    }
    void Update()
    {
           scoretext.text = "SCORE : " + score.ToString() ;
        if (score >= 3)
        {
            scoremsg.text = "Congratulations you win!!";
            timer -= Time.deltaTime;
            if (timer <= 0)
            {
                score = 0;
                FindObjectOfType<player>().resettries();
                SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
            }
        }
        else if (FindObjectOfType<player>().tries == 0 && score < 3)
        {
            scoremsg.text = "You Lose \n Try Again ";
            timer -= Time.deltaTime;
            if (timer <= 0)
            {
                score = 0;
                FindObjectOfType<player>().resettries();
                SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex );
            }
        }
    }
    void OnTriggerEnter(Collider othercollider)
    {
        if (othercollider.GetComponent<ball>() != null)
        {
            effectobj.SetActive(true);
            appsound.Play();
            score += 1;
            scoretext.text = "SCORE : " + score.ToString();
        }
    }
}
```

Controlling level ascension and managing tries:

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

public class gamecontroller : MonoBehaviour
{
    public player playr;    //TO CONNECT
    public float resettimer = 3f;

    void Update()
    {
        if (playr.holdingball == false)
        {
            resettimer -= Time.deltaTime;
```

```
            if (resettimer <= 0)
            {
                SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
            }
        }
    }
}
```

Defender movement:

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

public class defender : MonoBehaviour
{
    public Transform player;
    public Transform basket;
    public Transform opponent;
    public Vector3 defenderposition;
    public float defender_to_player_distance = 0.02f;

  void FixedUpdate()
    {
        defenderposition.x = ((player.position.x - defenderposition.x) *
defender_to_player_distance) + ((player.position.x + basket.position.x) / 2);
        defenderposition.z = ((player.position.z - defenderposition.z) *
defender_to_player_distance) + ((player.position.z + basket.position.z) / 2);
        defenderposition.y = opponent.position.y;
        transform.position = defenderposition;
        transform.LookAt(player);
    }
}
```

Ball trail feature:

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

public class ball : MonoBehaviour
{
    public GameObject trailobj;

    // Start is called before the first frame update
    void Start()
    {
        trailobj.SetActive(false);
    }
    public void activatetrail()
    {
        trailobj.SetActive(true);
    }
}
```

Basket movement:

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

public class move2 : MonoBehaviour
```

```
{
	public Transform basket;
	public Vector3 val;
	public int speed = -1;

	// Update is called once per frame
	void Update()
	{
		val.x += speed * Time.deltaTime;
		val.y = -1.7f;
		val.z = 9.5f;
		transform.position = (val);
		if (transform.position.x >= 10f || transform.position.x <= -9f)
		{
			speed *= -1;
		}
	}
}
```

Loading same scene after each try:

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

public class b1 : MonoBehaviour

{
    public void play1(int scnindex)
    {
        SceneManager.LoadScene(scnindex);
    }
}
```

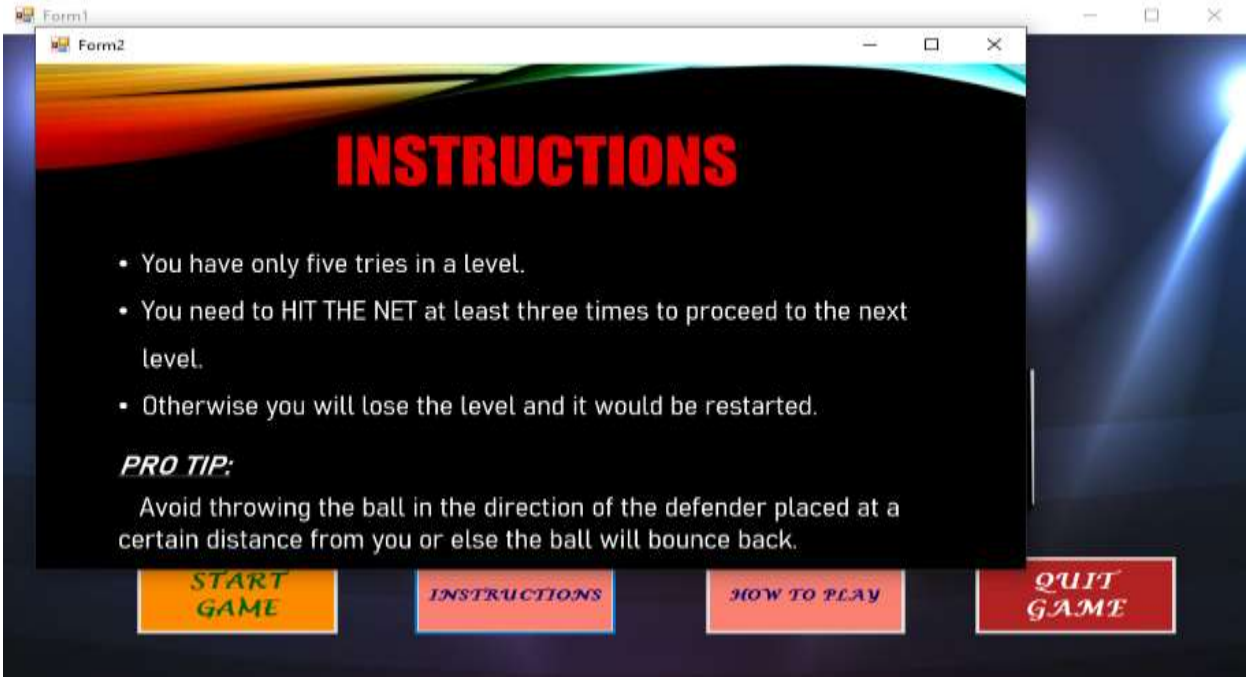
Audio management and sound effects:

```
using UnityEngine.Audio;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class audiomanager : MonoBehaviour
{
    public static AudioClip scoresound;
    static AudioSource auds;

 void Start()
    {
        scoresound = Resources.Load<AudioClip>("applause8");
        auds = GetComponent<AudioSource>();
    }

    // Update is called once per frame
    void Update()
    { }
    public static void playsound()
    {
        auds.PlayOneShot(scoresound);
    }
}
```

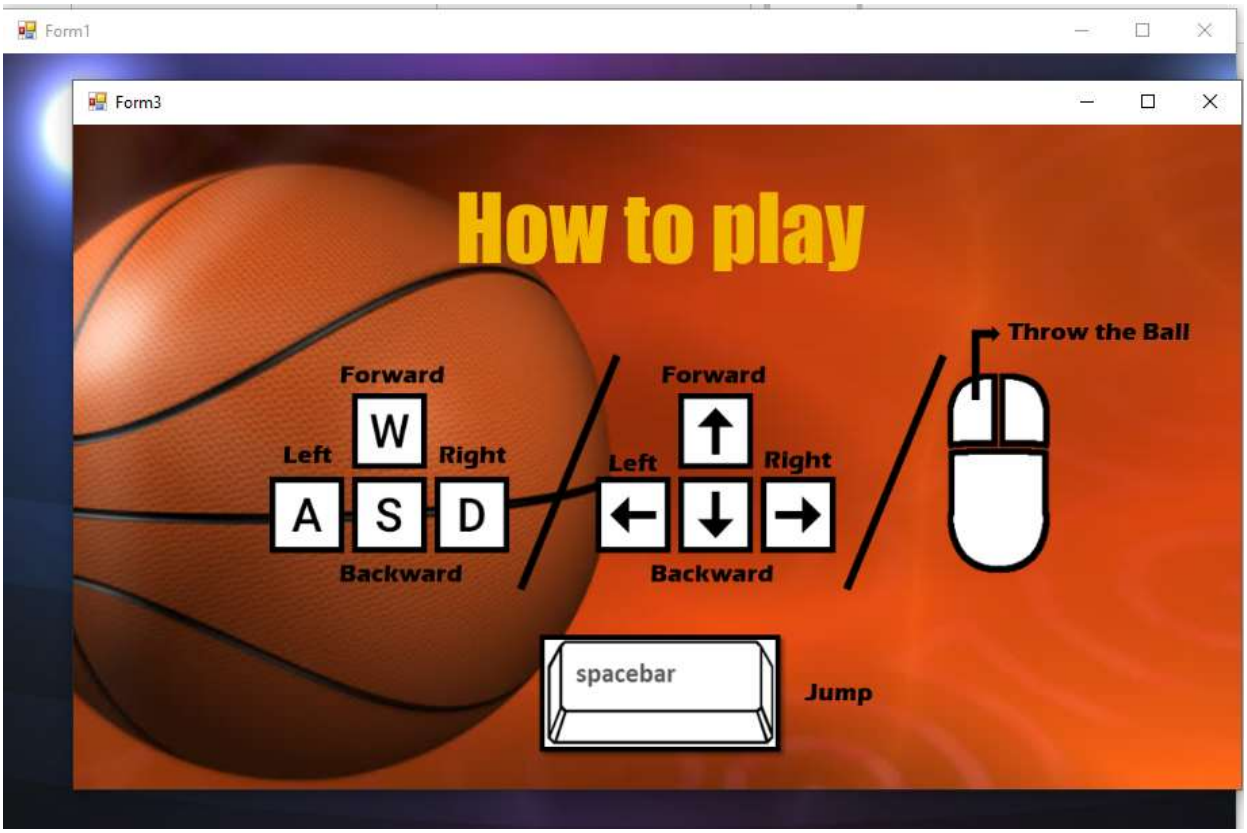
SCREENS:

1. **Main Menu:**

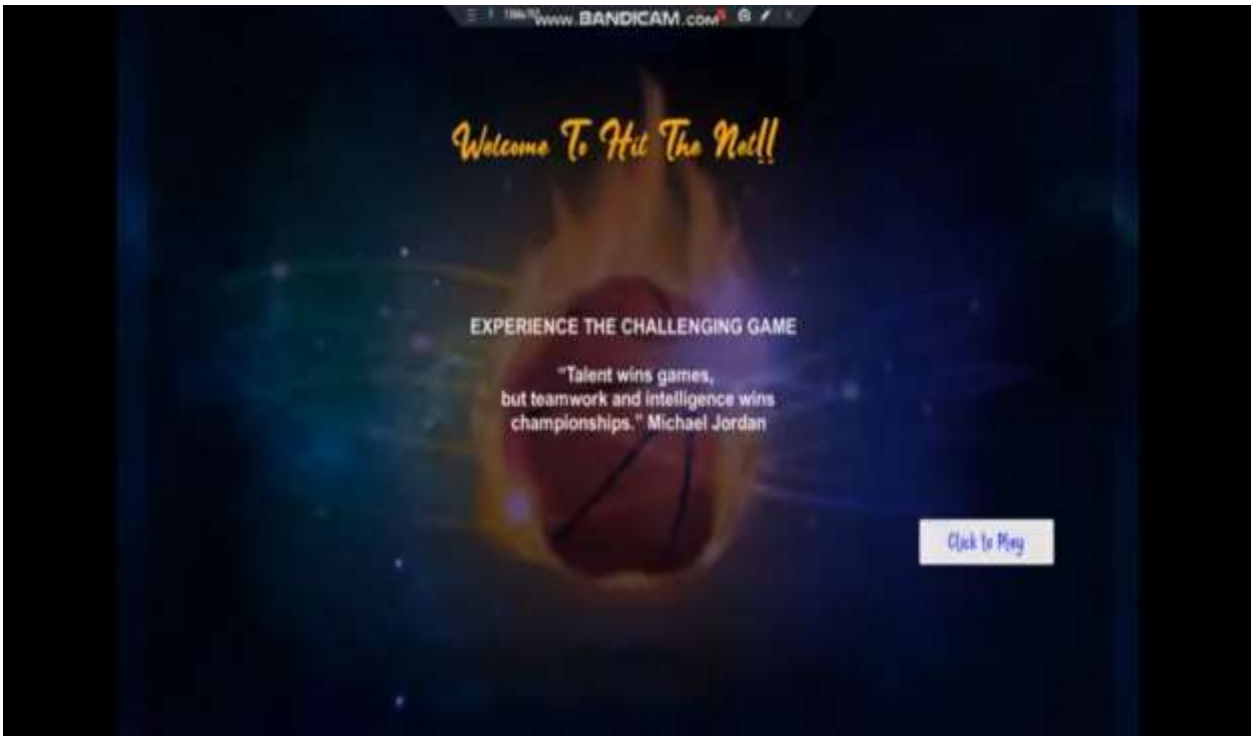
2. **Instructions:**



3. How to Play:



4. Start Game Screen:



5. **Losing a level:**

6. **Hitting the Net:**

7. **Winning a level:**

8. **Dodging the Defender:**

9. Credits:


Credits:
A KarByte® production
PROJECT MANGER:
Muhammad Hassan
INTERFACE DESIGN:
Fariya khan
Muhammad Sameed Abbasi
PROGRAMMERS:
Fariya Khan
Muhammad Sameed Abbasi
Fasih-ur-rehman Ansari
Muhammad Hassan
GAME DESIGNER:
Fasih-ur-Rehman Ansari