

```

using System.Collections.Generic;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using UnityEngine;
using System.Linq;

public class Score//:IEquatable<Score>
{
    public float _SaveBasicControl { get; set; }
    public float _SaveBasicSpeed { get; set; }
    public float _SaveFinalSpeed { get; set; }
    public float _SaveFinalTime { get; set; }

}

public class WindowTOPplayerScene : MonoBehaviour
{
    public Text TextScoreSpeed;
    public Text TextScoreTime;
    public static List<Score> _score = new List<Score>(10);

    public Button _buttonExit;
    public Button _buttonRestart;

    void Start()
    {
        _buttonExit.onClick.AddListener(() => ExitButton());
        _buttonRestart.onClick.AddListener(() => RestartButton());

        if (_score.Count < 10)
        {
            _score.Add(new Score() { _SaveBasicControl =
ButtonCarsScenes.SaveBasicControl, _SaveBasicSpeed =
ButtonCarsScenes.SaveBasicSpeed, _SaveFinalSpeed =
MoveCar.SaveFinalSpeed, _SaveFinalTime = WindowGameScene.SaveFinalTime
});
        }
        else
        {
            if(_score.Count >= 10)
            {
                Score deleteLast = _score.LastOrDefault(us =>
us._SaveFinalSpeed <= us._SaveFinalSpeed);
                if (deleteLast != null)
                {
                    _score.Remove(deleteLast);
                }
                _score.Insert(9,new Score() { _SaveBasicControl =
ButtonCarsScenes.SaveBasicControl, _SaveBasicSpeed =
ButtonCarsScenes.SaveBasicSpeed, _SaveFinalSpeed =
MoveCar.SaveFinalSpeed, _SaveFinalTime = WindowGameScene.SaveFinalTime
});
            }
            SortBySpeed();
            int i = 0;
            foreach (Score p in _score)
            {
                i++;
                TextScoreSpeed.text += $"{i}.Basic Control=
{p._SaveBasicControl:0.00}\tBasic Speed= {p._SaveBasicSpeed:0.00}\n Max
Speed= {p._SaveFinalSpeed:0.00}\t Time play= {p._SaveFinalTime:0.00}\n";
            }
        }
    }
}

```

```

    }
    SortByTime();
    int j = 0;
    foreach (Score p in _score)
    {
        j++;
        TextScoreTime.text += $"{j}.Basic Control=
{p._SaveBasicControl:0.00}\tBasic Speed= {p._SaveBasicSpeed:0.00}\n Max
Speed= {p._SaveFinalSpeed:0.00}\t Time play= {p._SaveFinalTime:0.00}\n";
    }
}

private void RemoveListener()
{
    _buttonExit.onClick.RemoveAllListeners();
    _buttonRestart.onClick.RemoveAllListeners();
}

void SortBySpeed()
{
    _score.Sort(delegate (Score x, Score y)
    {
        if (x._SaveFinalSpeed < 1 && y._SaveFinalSpeed < 1) return 0;
        else if (x._SaveFinalSpeed < 1) return -1;
        else if (y._SaveFinalSpeed < 1) return 1;
        else return y._SaveFinalSpeed.CompareTo(x._SaveFinalSpeed);
    });
}

void SortByTime()
{
    _score.Sort(delegate (Score x, Score y)
    {
        if (x._SaveFinalTime < 1 && y._SaveFinalTime < 1) return 0;
        else if (x._SaveFinalTime < 1) return -1;
        else if (y._SaveFinalTime < 1) return 1;
        else return y._SaveFinalTime.CompareTo(x._SaveFinalTime);
    });
}

public void ExitButton()
{
    SceneManager.LoadScene(1);
}

public void RestartButton()
{
    MoveCar.SaveFinalSpeed = 0f;
    WindowGameScene.SaveFinalTime = 0f;
    ButtonCarsScenes.SaveBasicControl = 0f;
    ButtonCarsScenes.SaveBasicSpeed = 0f;
    TriggersGameScene.namber = 0;
    HealthPlayerGameScene.Health = 100f;
    SceneManager.LoadScene(2);
}

void Update()
{
    if (Input.GetMouseButtonUp(0))
    {
        RemoveListener();
    }
    if (Input.GetKeyDown(KeyCode.Escape))
    {
        SceneManager.LoadScene(1);
    }
}

```

}
}
}