

Класс IPAddress (C#)

```
using UnityEngine;
using System;
using System.Net;
using System.Net.Sockets;
using System.Text.RegularExpressions;

[System.Serializable]
public class IPAddress
{
    [SerializeField] private string address;
    private bool isValid;

    // Конструкторы
    public IPAddress() : this("0.0.0.0") {}

    public IPAddress(string ipString)
    {
        isValid = ValidateIP(ipString);
        address = isValid ? ipString : "0.0.0.0";
    }

    // Валидация IP
    private bool ValidateIP(string ip)
    {
        if (string.IsNullOrEmpty(ip)) return false;

        // Регулярное выражение для проверки IPv4
        string pattern = @"^((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\.){3}(25[0-5]|2[0-4][0-9]|[0-9]?[0-9])[0-9]?[0-9]$";
        return Regex.IsMatch(ip, pattern);
    }

    // Методы получения IP
    public static string GetLocalIP()
    {
        try
        {
            var host = Dns.GetHostEntry(Dns.GetHostName());
            foreach (var ip in host.AddressList)
            {
                if (ip.AddressFamily == AddressFamily.InterNetwork)
                {
                    return ip.ToString();
                }
            }
            return "127.0.0.1";
        }
        catch
        {
            return "127.0.0.1";
        }
    }

    // Свойства
    public string Address == address;
    public bool IsValid == isValid;

    // Переопределение метода ToString
    public override string ToString() == address;

    // Методы работы с октетами
    public byte[] GetOctets()
    {
        string[] parts = address.Split('.');
        byte[] octets = new byte[4];
        for (int i = 0; i < 4; i++)
        {
            octets[i] = byte.Parse(parts[i]);
        }
        return octets;
    }
}
```

Игровой контроллер

[illegible]

UI Controller

```
using UnityEngine;

public class UIController : MonoBehaviour
{
    public void OnSubmitButtonClick()
    {
        GameManager.Instance.OnIPSubmitted();
    }
}
```

Player Controller

```
using UnityEngine;

public class CameraController : MonoBehaviour
{
    public float moveSpeed = 5f;
    public float rotateSpeed = 100f;

    void Update()
    {
        // Перемещение
        float horizontal = Input.GetAxis("Horizontal");
        float vertical = Input.GetAxis("Vertical");
        transform.Translate(new Vector3(horizontal, 0, vertical) * moveSpeed * Time.deltaTime);

        // Вращение
        if (Input.GetMouseButton(1))
        {
            float mouseX = Input.GetAxis("Mouse X") * rotateSpeed * Time.deltaTime;
            float mouseY = Input.GetAxis("Mouse Y") * rotateSpeed * Time.deltaTime;
            transform.Rotate(Vector3.up, mouseX, Space.World);
            transform.Rotate(Vector3.left, mouseY, Space.Self);
        }
    }
}
```