AMA Computer College Naga City

Panganiban Dr, Camarines Sur, Naga

Documentary Requirement Presented

to the Senior High School Department for Work Immersion

In Partial Fulfillment of the Requirements

For

Information Communication and Technology (ICT) Strand

PROGRAMMING

By:

JOHCEL GENE BITARA

Submitted To:

JENEFER BERMUSA

Work Immersion Adviser

March 2021

ACKNOWLEDGEMENT

I would like to express my thanks of gratitude to our teacher online as well as our principal in our branch for their guidance and for allowing me to showcase the fruit of my skills throughout this whole school year which helped me learn new things and grow as a better student, for that, I am thankful for it.

I would also like to thank my parents for their time and effort in taking care of me while I work hours and days on this project and to my supported friends that helped me create ideas, advice, test, and give proper feedback on my project, their time and effort are greatly appreciated as it helped me balanced my project that made it better.

TABLE OF CONTENTS

I. COMPANY NAME AND BRIEF DESCRIPTION

This project is targeted to small companies that are Game Studios. Preferably Indie Game Studios such as Hyperbeard.

Hyperbeard is a Game Development Company mostly known for their simple and minimalistic android games such as *Adorable Home, KleptoCats, Bunny Buns* and many more all of which doesn't really use any online interaction features yet.

Through the success of their games, they have also come to be known by the top platform for publishing really high quality and a highly rated casual game all of which can be found on Google Plays and the Apple Appstore.

II. SYSTEM BRIEF DESCRIPTION

This project is a chat system designed for any use of online interaction either by a system that can be used as a public chat/world chat or a private chat by using the library Photon Network; a Unity package library for multiplayer games. This project uses Unity Engine which is used mostly for Game Development and uses a programming language called C-Sharp; An Object-Oriented Language similar to Java which uses classes and objects for handling data.

LOGIN

INPUT: User first gets to login and input their username to identify and display in the chat room.

PROCESS: gets stored in a key under the function PlayerPrefs(key); a function in Unity C#. And checks if the username is valid, if so redirect to Lobby room.

LOBBY

INPUT: Find available rooms in the Photon Server. Using the Find Button.

PROCESS: If the user did not find or failed to join a room then the user automatically creates a new room and wait for other users to come. Where in the "Lobby.cs" script it checks if the number of players inside the room is greater than 2 then they can get redirected in the "Main Scene (Chatroom)".

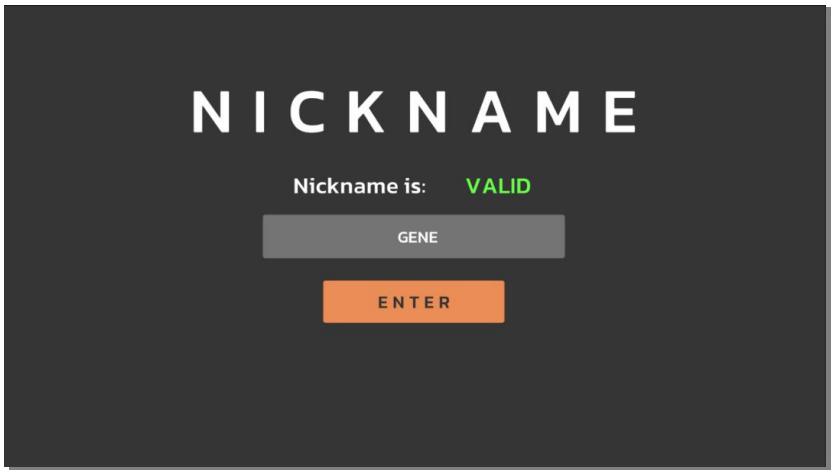
CHATROOM

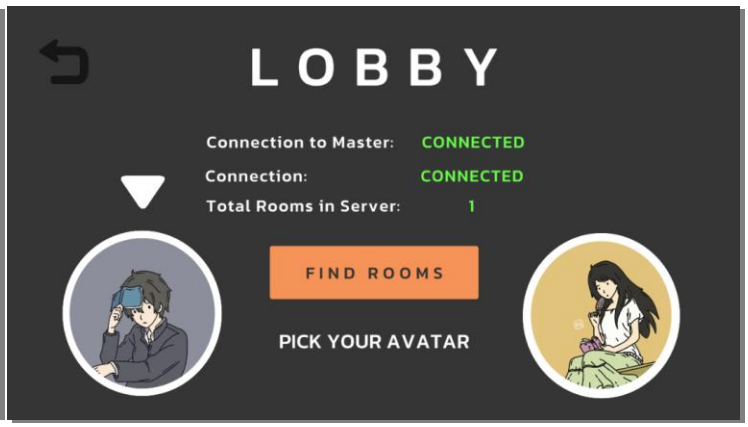
INPUT: Type messages in the message field that the user wants to say.

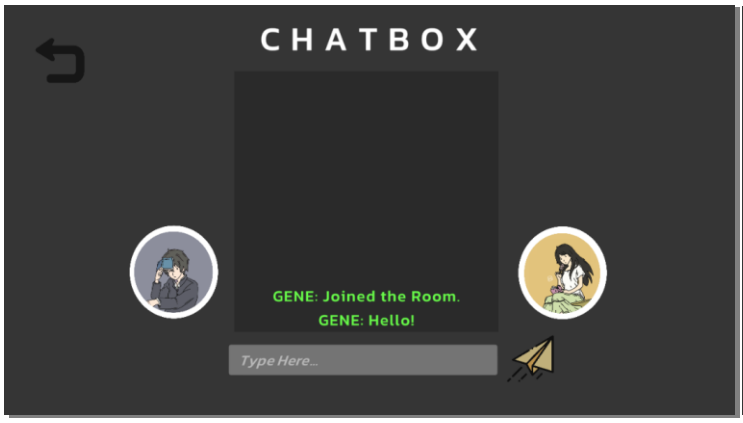
PROCESS: The value in input field gets stored in a variable that will be used to sending messages across the network and into the recipient screen.

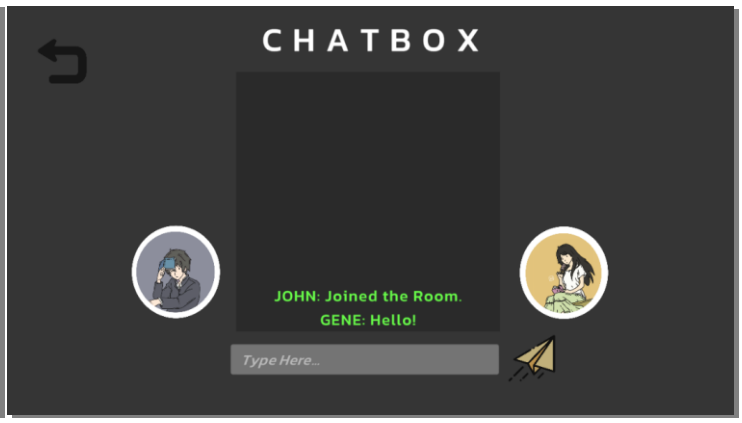
OUTPUT: Using the PublishMessage(message); used in sending message over the network using the function on library Photon, both user gets to receive the same message logs in their screen with their name on display when chatting and the sender name attached used by getting the name key in PlayerPrefs(key); thus making sending and receiving messages across clients is made possible.

III. SCREENSHOT OF SYSTEM









CLIENT A SCREEN **CLIENT B SCREEN**

IV. PROGRAM CODE

LOGIN SCRIPT

```
using UnityEngine.SceneManagement;
using System.Collections;
using UnityEngine;
using Photon.Pun;
using TMPro;

namespace Scripts.PlayerName
{
    public class SavePlayerName : MonoBehaviour
    {
        [SerializeField] private TextMeshProUGUI statusText = null;
        [Header ("Color Indicator")]
        [SerializeField] private Color green;
        [SerializeField] private Color red;

        [Space]
        [SerializeField] private TMP_InputField textField = null;

        public static string playerName;
        public static bool validName = false;


        private const string playerPrefsNameKey = "PlayerName";

        private void Start() => textField.text = PlayerPrefs.GetString(playerPrefsNameKey);

        public void SaveName()
        {
            playerName = textField.text;

            if (!string.IsNullOrEmpty(playerName))
            {
                validName = true;
                statusText.color = green; // GREEN
                statusText.text = "VALID";
                PlayerPrefs.SetString(playerPrefsNameKey, playerName);

                PhotonNetwork.NickName = playerName;
                StartCoroutine(LoadLobby());
            }
            else
            {
                statusText.color = red; // RED
                statusText.text = "INVALID";
            }
        }
        private IEnumerator LoadLobby()
        {
            int waitTime = 2;
            yield return new WaitForSeconds(waitTime);
            // Load to Lobby Scene
            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex+1);
        }
    }
}
```

LOBBY SCRIPT

```
using UnityEngine;
using Photon.Pun;
using Photon.Realtime;
using TMPro;

public class Lobby : MonoBehaviourPunCallbacks
{
    [Header ("Connection to Master")]
    [SerializeField] private TextMeshProUGUI connMasterStatText = null;
    [SerializeField] private Color greenM;
    [SerializeField] private Color redM;

    [Header ("Status text")]
    [SerializeField] private TextMeshProUGUI statusText = null;
    [SerializeField] private Color greenS;
    [SerializeField] private Color redS;

    [Header ("Room Counter")]
    [SerializeField] private TextMeshProUGUI roomCountText = null;
    [SerializeField] private Color greenR;
    [SerializeField] private Color redR;

    public static int currentPlayerInRoom = 0;

    private RoomOptions roomOptions;
    private const string GAME_VERSION = "0.1";
    public static int MAX_PLAYERS = 2;

    private void Awake() => PhotonNetwork.AutomaticallySyncScene = true;

    private void Start()
    {
        PhotonNetwork.ConnectUsingSettings();
        PhotonNetwork.GameVersion = GAME_VERSION;
    }
    private void Update()
    {
        if (PhotonNetwork.CountOfRooms > 0)
        {
            roomCountText.text = PhotonNetwork.CountOfRooms.ToString();
            roomCountText.color = greenR;
        }
        else
        {
            roomCountText.text = PhotonNetwork.CountOfRooms.ToString();
            roomCountText.color = redR;
        }
    }
    //----------BUTTONS-----------
    public void FindRooms()
    {
        statusText.text = "FINDING";
        statusText.color = redS;

        PhotonNetwork.JoinRandomRoom();
    }

    public void LeaveRoom()
    {
        PhotonNetwork.LeaveRoom();
        PhotonNetwork.Disconnect();
    }

//----------CONNECTION LOBBY-----------
    public override void OnConnectedToMaster()
    {
        connMasterStatText.text = "CONNECTED";
        connMasterStatText.color = greenM;

        statusText.text = "CONNECTED";
        statusText.color = greenS;
    }
```

```
    public override void OnCreatedRoom()
    {
        statusText.text = "CREATED ROOM";
        statusText.color = greenS;

        PlayerAvatarAssign.host = true;
    }

    public override void OnJoinedRoom()
    {
        currentPlayerInRoom = PhotonNetwork.CurrentRoom.PlayerCount;
        statusText.text = "JOINED ROOM";
        statusText.color = greenS;

        if (currentPlayerInRoom == MAX_PLAYERS)
        {
            statusText.text = "STARTING";
            statusText.color = greenS;

            StartCoroutine(LoadMainScene());
        }
        else
        {
            statusText.text = "WAITING";
            statusText.color = greenS;
        }
    }

    public override void OnPlayerEnteredRoom(Player newPlayer)
    {
        currentPlayerInRoom = PhotonNetwork.CurrentRoom.PlayerCount;

        if (currentPlayerInRoom == MAX_PLAYERS)
        {
            statusText.text = "STARTING";
            statusText.color = greenS;

            StartCoroutine(LoadMainScene());
        }
        else
        {
            statusText.text = "WAITING";
            statusText.color = greenS;
        }
    }

    public override void OnLeftRoom()
    {
        statusText.text = "YOU LEFT";
        statusText.color = redS;

        currentPlayerInRoom = PhotonNetwork.CountOfRooms;
    }

    //-------DISCONNECT-------
    public override void OnCreateRoomFailed(short returnCode, string cause)
    {
        statusText.text = "ROOM CREATE FAILED";
        statusText.color = redS;
    }

    public override void OnJoinRandomFailed(short returnCode, string cause)
    {
        statusText.text = "NO ROOMS";
        statusText.color = redS;
        PhotonNetwork.CreateRoom("MyRoom", roomOptions);
    }

    public override void OnDisconnected(DisconnectCause cause)
    {
        statusText.text = "DISCONNECTED";
        statusText.color = redS;

        connMasterStatText.text = "DISCONNECTED";
        statusText.color = redM;

        SceneManager.LoadScene("Login");
    }

    private IEnumerator LoadMainScene()
    {
        float waitTime = 2f;
        yield return new WaitForSeconds(waitTime);
        // Go to Main Scene (Both Clients)
        PhotonNetwork.LoadLevel(SceneManager.GetActiveScene().buildIndex+1);
    }
}
```

CHAT MANAGER SCRIPT

```
using UnityEngine;
using Photon.Chat;
using Photon.Pun;
using ExitGames.Client.Photon;
using TMPro;
using System;

public class PhotonChatManager : MonoBehaviour, IChatClientListener
{
    [Header ("Chat")]
    [SerializeField] private TMP_InputField inputMessageField = null;
    [SerializeField] private TextMeshProUGUI messageLogs = null;

    private ChatClient chatClient;
    private string username;
    private string chatRoom;

    private void Start()
    {
        username = PlayerPrefs.GetString("PlayerName");

        chatClient = new ChatClient(this);
        chatClient.Connect(PhotonNetwork.PhotonServerSettings.AppSettings.AppIdChat,
        PhotonNetwork.AppVersion, new AuthenticationValues(username));

        chatRoom = "chatRoom";
    }

    private void Update()
    {
        chatClient.Service();
    }
//---------------SEND MESSAGE--------------->
    public void SendMessage()
    {
        if (!string.IsNullOrEmpty(inputMessageField.text))
        {
            chatClient.PublishMessage(chatRoom, inputMessageField.text);
            inputMessageField.text = "";
        }
    }
//---------------IMPLEMENTED INTERFACE--------------->
    public void OnConnected()
    {
        Debug.Log("Connected");
        chatClient.Subscribe(new string[] { chatRoom });
        messageLogs.text = $"{username}: Joined the room.";
    }

    public void OnDisconnected()
    {
        Debug.Log("Connected");
    }


    public void OnGetMessages(string channelName, string[] senders, object[] messages)
    {
        for (int i = 0; i < messages.Length; i++)
        {
            messageLogs.text = messageLogs.text + Environment.NewLine + $"{senders[i]}: {messages[i]}";
        }
    }

    public void OnSubscribed(string[] channels, bool[] results)
    {
        Debug.Log("Subscribed");

        for (int i = 0; i < channels.Length; i++)
        {
            Debug.Log(channels[i]);
        }
    }

    public void OnUnsubscribed(string[] channels)
    {
        Debug.Log("Unsubscribed");
    }

```

V. RESUME

JOHCEL GENE BITARA

Developer

Wellville homes Subdivision, San Felipe, Camarines Sur Naga City, 4400

0947-145-9566

bitaragene@gmail.com

> ABOUT ME

I find passion in tech and will always have, I build projects ranging from OOP, GUI Backend and Frontend to Unity Game Engine and create projects such as Android Game Applications to PC Applications.

> PERSONAL DETAILS

Sex	Male
Age	18
Date of Birth	July 4, 2002
Nationality	Filipino

> SKILLS

- Knowledge in using GitHub/Git.
- Experience with desktop & android/pc game development.
- Experience in designing and building pc applications and GUI apps.
- Good Understanding of Object-Oriented Programming.
- Troubleshoot minimal errors Hardware or Software.
- Proficient in Windows OS.
- Communication skills.

> LANGUAGE

ENGLISH

BICOL

TAGALOG

> EDUCATION

- **Senior High School 2019-2021** AMACC Naga Camarines Sur, Naga
- **Junior High School 2017-2019** Naga College Foundation, Inc. Camarines Sur, Naga
- **Elementary** Naga Seventh-Day Adventist Elementary School, Inc. Camarines Sur, Naga

> ACHIEVEMENT

- Junior High School 2016-2019 NCF

  Honors/Conduct Award/Perfect Attendance Award
- Junior High School 2017-2018 NCF

  Campus Journalism Award

> PROGRAMMING LANGUAGE

JAVA

JAVASCRIPT

C#

NETWORKING

> LINKS

https://github.com/genebit

https://www.facebook.com/gene.bitara