The Virtual Reality Mental Health Therapy Platform

Michael Sumaylo, Nicolas Steger, Riki MacMillan
Project Sponsors: Dr. Mehdi Mirakhorli, Dr. Katy Tarrit Mirakhorli

ICS 496
Spring 2024

Problem

The prevalence of mental illness has increased in recent years. In the United States, approximately 22.8% of all adults aged 18 or older struggle with mental illness[1].
Key barriers to accessing mental health services include[2]:

- Not enough time to meet in person
- Concerned about being committed
- Did not know where to go for services

References:
1. NIMH, “Mental Illness,” 2023. nimh.nih.gov/statistics.
2. Conroy et al., “Why People Aren’t Getting Care,” APA, 2021. apa.org/monitor/2020/07/datapoint-care.

Solution

Implemented a VR standalone application in Unity that allows patients to have therapy sessions with health professionals in a more interactive manner compared to a video or phone call.

Methodology

Management Approach: Agile/Incremental

- Weekly Scrums and Biweekly Product Demonstrations: Showcasing current achievements, updates, and gathering feedback.

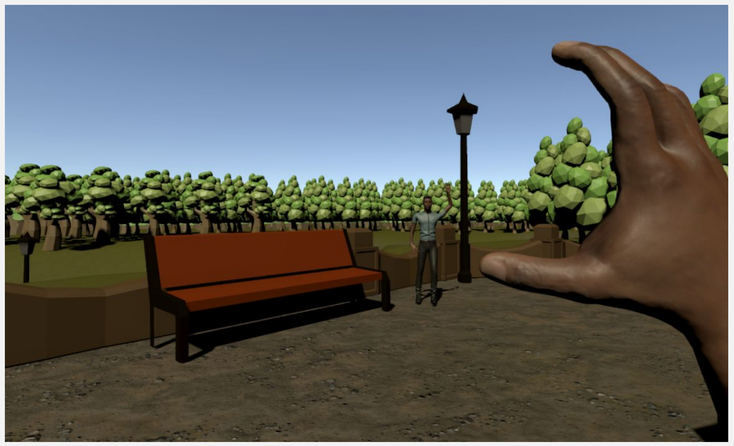
Github Workflow Management

- Task Management: Individual tasks are assigned and tracked via GitHub.
- Version Control: Pull and merge operations ensure fluid development across different tasks.
- Resource Management: Efficient handling of packages and assets.

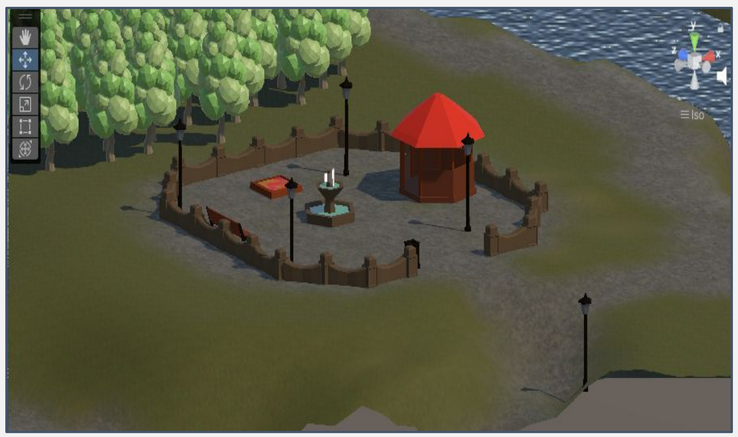
Tech Stack

VIVOX
COMMUNICATION IS CRITICAL
Unity

Example Scene



Scene Development



C# Code for Backend

```
0 references
public async void CreateLobby()
{
    try
    {
        Allocation allocation = await RelayService.Instance.CreateAllocationAsync(maxPlayers);
        string joinCode = await RelayService.Instance.GetJoinCodeAsync(allocation.AllocationId);

        string lobbyName = lobbyNameField.text;
        string password = lobbyPasswordField.text;

        CreateLobbyOptions options = new CreateLobbyOptions
        {
            Password = password,
            Data = new Dictionary<string, DataObject> {
            { "JOIN_CODE", new DataObject(DataObject.VisibilityOptions.Public, joinCode) }
        }
        };

        // Create the lobby with the Relay join code included
        Lobby lobby = await LobbyService.Instance.CreateLobbyAsync(lobbyName, maxPlayers, options);

        currentLobby = lobby;
        HeartbeatManager.Instance.StartHeartbeat(lobby.Id);
        Debug.Log($"Created lobby! Name: {lobby.Name}, MaxPlayers: {lobby.MaxPlayers}, ID: {lobby.Id}, Password: {password}, JoinCode: {joinCode}");

        UnityTransport transport = NetworkManager.Singleton.GetComponent<UnityTransport>();
        transport.SetHostRelayData(allocation.RelayServer.IpV4, (ushort)allocation.RelayServer.Port, allocation.AllocationIdBytes, allocation.Key, allocation.ConnectionData);

        await LoadSceneAsync(sceneToLoad);

        NetworkManager.Singleton.StartHost();
        Debug.Log("is heartbeat running?" + isHeartbeatRunning);

    }
    catch (LobbyServiceException e)
    {
        Debug.LogError($"Failed to create lobby: {e.Message}");
    }
}
```

Accomplishments

- Allow users to host sessions that require entering the correct lobby and password
- Use full-body models
- Implement voice chat using Vivox
- Synchronizing users in a multiplayer scenario using Unity's Netcode for GameObjects
- Provide various environments to suit user needs and preferences

Challenges

- Understanding how to implement VR play over a network
- Understanding and implementing voice chat
- Character model bone rigging
- Scene loading and management to allow for network-related objects to work properly
- Package compatibility across different headset models (Meta vs non-Meta headsets)

Takeaway

Using Unity alongside C# offers a practical approach to understanding development and programming, enabling users to create interactive 3D applications.

- First Unity Project for ⅔ of the team
- Learning and improving C# and Unity skills
- Using Unity's netcode for the first time
- Applying Agile methodology
- Working with a sponsor and incorporating their input to guide project development

Next Steps

- More user model options with customization
- Improved scenes and environments
- Moderation settings for hosts
- Test with large group of users in a single session