

# ACROBOARD: APLICAÇÃO DE REALIDADE VIRTUAL E GAMIFICAÇÃO PARA AUXILIAR PACIENTES EM TRATAMENTO DE ACROFOBIA

Acadêmico:

**Alan Felipe Jantz**

[afjantz@furb.br](mailto:afjantz@furb.br)



Orientador:

**Dalton Solano dos Reis**

[dalton@furb.br](mailto:dalton@furb.br)

# Roteiro

- Introdução
- Objetivos
- Fundamentação Teórica
- Trabalhos Correlatos
- Descrição do aplicativo
- Resultados
- Conclusões

# Introdução

- Fobia é um medo persistente, excessivo e irreal, classificado como um transtorno de ansiedade, pode ser tratado com remédios ou psicoterapias, com acompanhamento médico;
- Uma das psicoterapias é o tratamento de exposição à realidade virtual, que melhorou com a evolução dos ambientes virtual, ajudando na imersão e no senso de presença dos usuários;
- O senso de presença dá aos usuários a percepção de estar no ambiente virtual, é importante para estimular nos usuários a sensação de desconforto e ansiedade;
- Com estes sentimentos que causam desconforto, é possível utilizar técnicas de gamificação para manter a motivação e engajamento do usuário.

# Objetivos

Disponibilizar um aplicativo para auxiliar pacientes em tratamento de acrofobia utilizando terapia de exposição a realidade virtual junto a mecanismos de gamificação.

**Os objetivos específicos são:**

- gerar ambiente virtual que favoreça o sentimento de ansiedade relacionado ao medo de altura;
- explorar mecanismos de desafios que o usuário deve realizar dentro do ambiente virtual e dar pontuações ao completar tais desafios;
- coletar dados da posição do usuário durante a utilização do aplicativo para permitir o acompanhamento da evolução do paciente;
- validar facilidade de uso e aceitação do aplicativo com profissionais da área.

# Realidade Virtual e Senso de Presença

- Realidade alternativa criada artificialmente, um ambiente digital que pode ser experimentado como um ambiente real
  - Sentidos como visão e audição são frequentemente explorados nestes ambientes
- Senso de presença é percepção de estar no ambiente virtual, o mundo real é “desligado” e apenas o mundo virtual é visto e ouvido pelo usuário
  - Varia de usuário para usuário
- A imersão mede o quão preciso um ambiente virtual é capaz de fazer com o que o usuário se sinta em uma realidade diferente da qual ele se encontra
  - Qualidade da imagem
  - Campo de visão
  - Precisão do tempo de resposta
  - Interação com elementos do ambiente virtual

# Gamificação

- Utilização de mecanismos de jogos como uma metodologia
- Tem por objetivo tornar tarefas tediosas ou repetitivas mais agradáveis para melhorar o engajamento e a motivação de determinado público-alvo
- Entre os elementos que podem ser utilizados, estão:
  - Metas
  - Respostas às ações do usuário
  - Desafios
  - Pontuação de desempenho
  - Recompensas

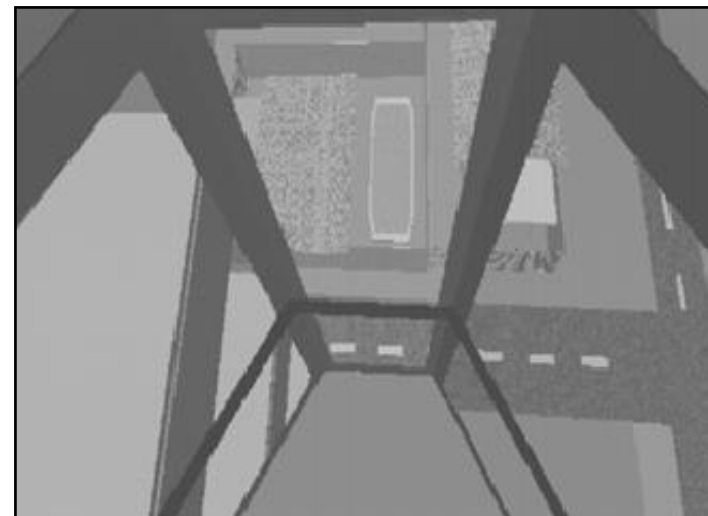
# Acrofobia

- Fobia específica relacionada à altura
  - Comumente associado a locais altos, como montes, janelas ou aviões
  - Também ocorre com situações específicas, como abismos, edifícios altos ou precipícios
- Não está necessariamente relacionado a experiências traumáticas
- Pode ocorrer ao estar em um local que estimule a fobia, quando imagina uma situação ou visualiza uma imagem
- Atinge qualquer faixa etária, mas possui ocorrência maior com pessoas entre 5 e 7 anos e por volta dos 14 anos de idade

# Emprego da Realidade Virtual no tratamento de fobia de altura

Estácio, Jacob e Artero (2016)

- Tem como objetivo utilizar realidade virtual para verificar, experimentalmente, os resultados que podem ser obtidos através desta metodologia no tratamento de fobia de altura
- Possui um mundo virtual onde o paciente pode explorar uma vizinhança e utilizar um elevador panorâmico





# Emprego da Realidade Virtual no tratamento de fobia de altura

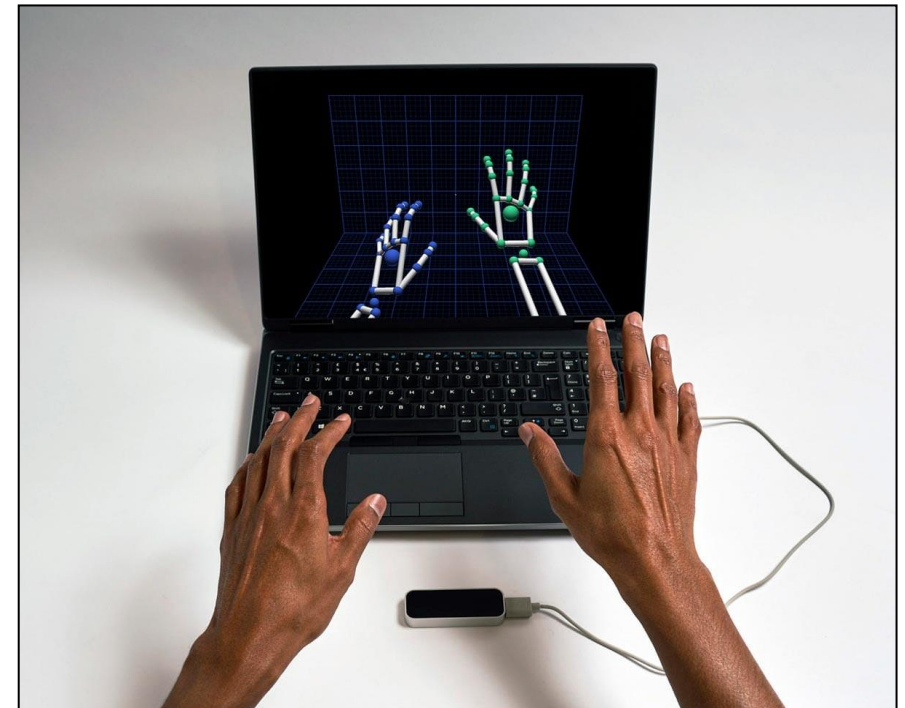
Estácio, Jacob e Artero (2016)

- Concluiu que o ambiente virtual pode gerar comportamentos similares ao de um ambiente real, destacando que com a ajuda de equipamentos específicos de RV, é possível melhorar a imersão do indivíduo

# Gamificação de procedimentos médicos

Adilkhan e Alimanova (2020)

- Tem como objetivo tornar o processo de reabilitação da movimentação das mãos mais lúdico
- Possui um ambiente virtual interativo
- Utilização do sensor Leap Motion para captar a movimentação das mãos
- Exploração da gamificação através de fases com diferentes cenários



Sensor Leap Motion

# Gamificação de procedimentos médicos

Adilkhan e Alimanova (2020)

- Concluiu que este tipo de abordagem pode ajudar na motivação e comprometimento dos pacientes
- Recompensas dadas durante o uso da aplicação gamificada e tratamentos diferentes dos convencionais ajuda na motivação dos pacientes



# Uma proposição de gamificação em sistemas m-Health para o engajamento dos usuários

Cechetti (2018)

- Tem como objetivo apresentar uma proposição de gamificação para favorecer o engajamento dos usuários durante o tratamento com a utilização de aplicativos de saúde m-Health
- Alguns dos elementos de gamificação são:
  - Barras de progresso com pontos coletados pelos usuários
  - Recomendação de desafios para coleta de novos pontos
  - Sistema de classificação entre pacientes
  - Feedback de desempenho ao completar desafios



# Uma proposição de gamificação em sistemas m-Health para o engajamento dos usuários

Cechetti (2018)

- Foi concluído que a gamificação favoreceu o engajamento dos participantes com elementos de jogos, inclusive motivando aqueles que não possuíam adesão ao tratamento antes do teste
- Outro fator que ajuda na aderência de pacientes na utilização de aplicativos como este é o supervisionamento por profissionais da saúde.

Ranking

Por Paciente

Por Médico

Médico:

Filtrar por:

Pressao

Ranking por Médico - Pressao

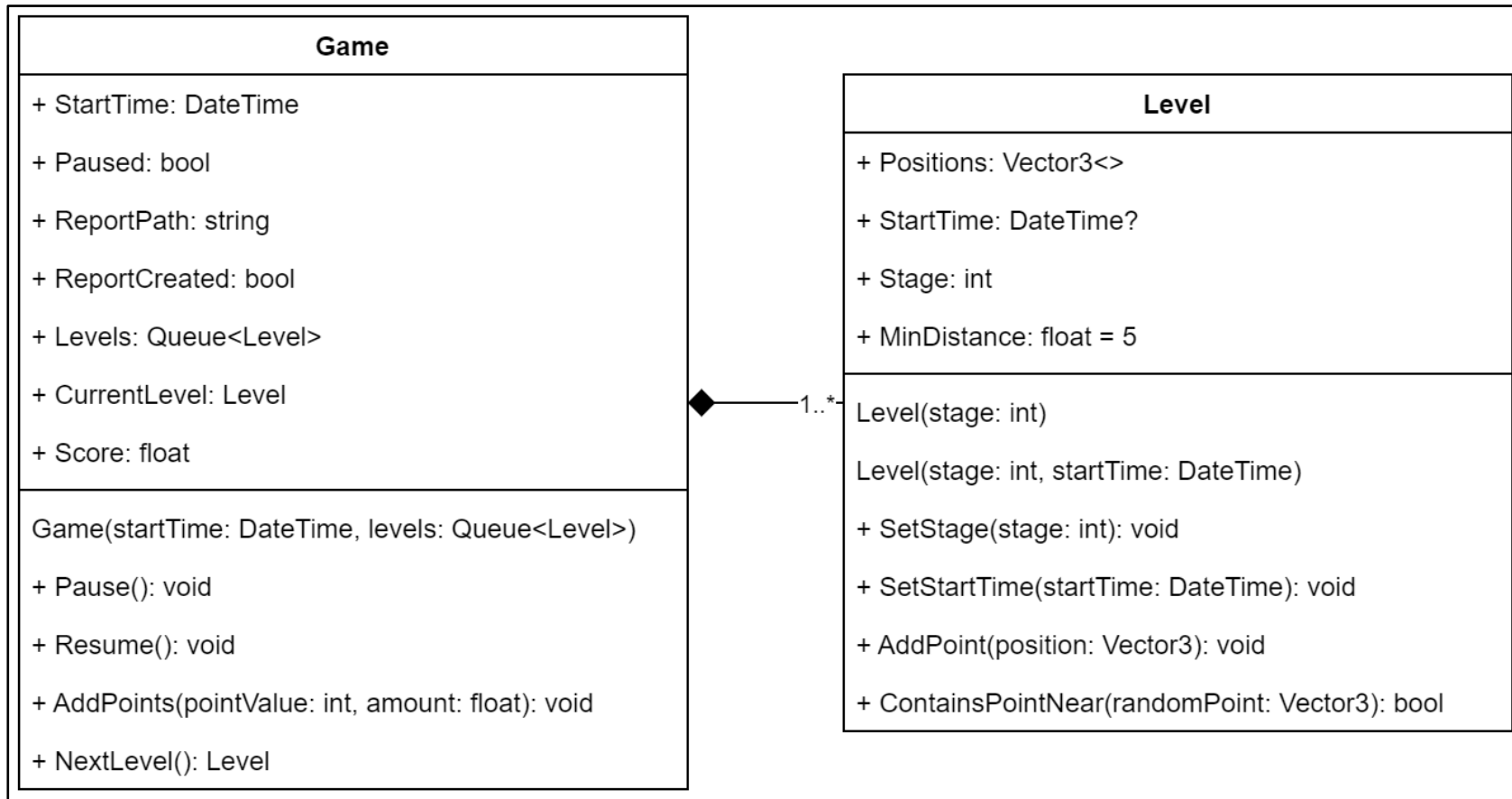
Colocação no Ranking Geral	Código do Paciente	Pontuação Pressao
2°	P56	150 pontos
3°	P64	150 pontos
7°	P79	135 pontos
8°	P77	120 pontos

# Descrição do aplicativo

# Requisitos

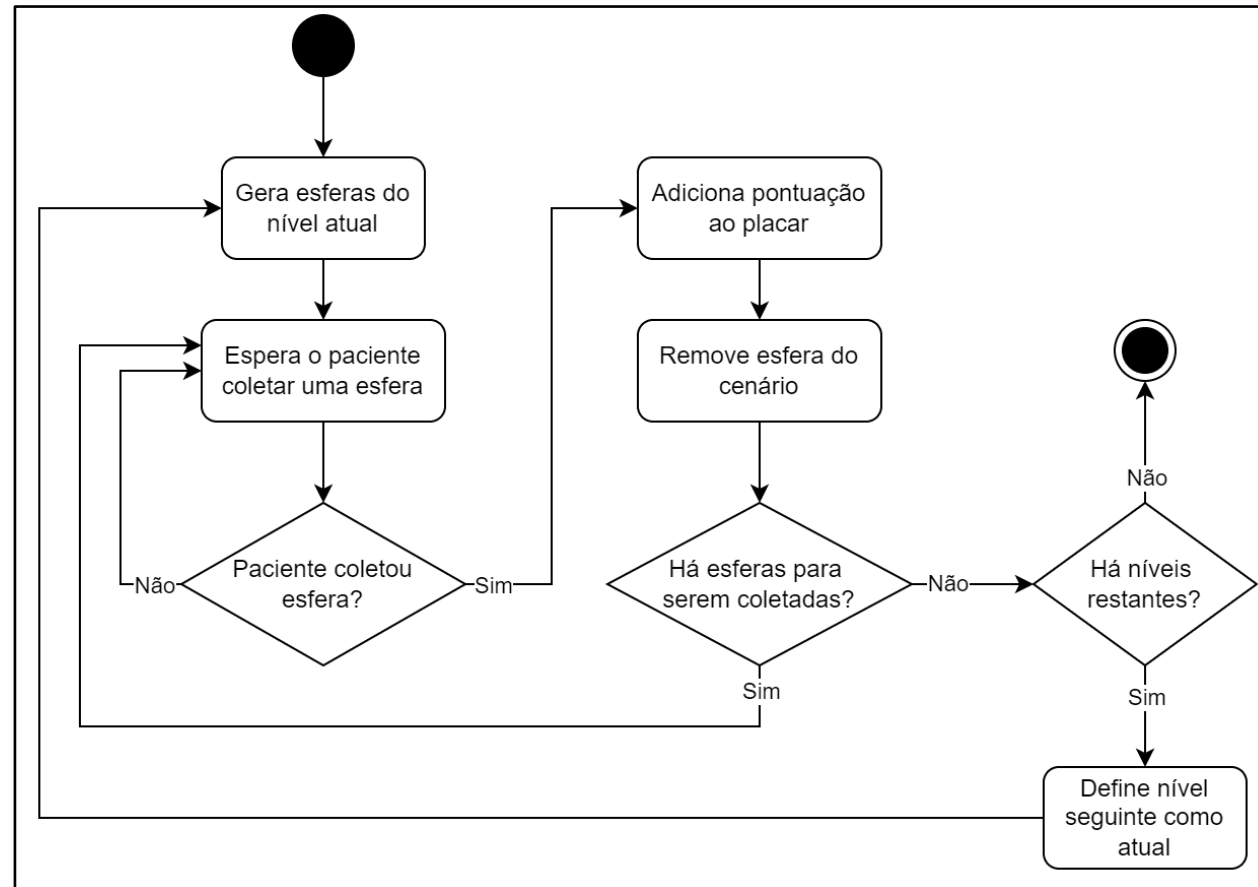
- Renderizar um ambiente virtual (RF);
- Permitir que o usuário se locomova dentro do ambiente virtual (RF);
- Proporcionar ao usuário sentimento de desconforto/ansiedade (RF);
- Implementar mecanismos de gamificação, tais quais níveis e pontuação ao atingir objetivos (RF);
- Coletar dados da posição do usuário durante a utilização do aplicativo para acompanhamento do psicólogo (RF);
- Emitir um relatório de desempenho do usuário durante a utilização do aplicativo para acompanhamento do psicólogo (RF);
- Utilizar o motor de jogos Unity (RNF);
- Ser compatível com o sistema operacional Android (RNF);
- Ser implementado utilizando a linguagem C# (RNF).

# Especificação





# Especificação



# AcroboardConfiguration

<u>AcroboardConfiguration</u>
+ <u>FilePath: string</u>
+ <u>PlayerVelocity: float</u>
+ <u>PlatformVelocity: float</u>
+ <u>PlayerViewMaxDistance: float</u>
+ <u>LevelsAmount: int</u>
+ <u>SpectatorMode: bool</u>
+ <u>Tutorial: bool</u>
+ <u>Reset(): void</u>
- <u>GetStringValue(key: string, defaultValue: string): string</u>
- <u>GetFloatValue(key: string, defaultValue: float): float</u>
- <u>GetIntValue(key: string, defaultValue: int): int</u>

```
private static string GetStringValue(string key, string defaultValue)
{
    var result = PlayerPrefs.GetString(key);

    if (string.IsNullOrEmpty(result))
    {
        PlayerPrefs.SetString(key, defaultValue);
        result = defaultValue;
    }

    return result;
}

private static float GetFloatValue(string key, float defaultValue)
{
    var result = PlayerPrefs.GetFloat(key);

    if (result == 0)
    {
        PlayerPrefs.SetFloat(key, defaultValue);
        result = defaultValue;
    }

    return result;
}

private static int GetIntValue(string key, int defaultValue)
{
    var result = PlayerPrefs.GetInt(key);

    if (result == 0)
    {
        PlayerPrefs.SetInt(key, defaultValue);
        result = defaultValue;
    }

    return result;
}
```

# LevelGenerator

```
public static Queue<Level> Genetare(int amount)
{
    var result = new Queue<Level>();

    for (int stage = 0; stage < amount; stage++)
    {
        var level = new Level(stage + 1);

        List<LevelPointRange> pointsRange = defaultPointRanges;

        if (pointRangePerLevel.ContainsKey(stage))
            pointsRange = pointRangePerLevel[stage];

        foreach (var pointRange in pointsRange)
        {
            var randomPoint = pointRange.GetRandomRange();

            while (level.ContainsPointNear(randomPoint))
                randomPoint = pointRange.GetRandomRange();

            level.AddPoint(randomPoint);
        }

        result.Enqueue(level);
    }

    return result;
}
```

```
private static readonly Dictionary<int, List<LevelPointRange>> pointRangePerLevel =
    new Dictionary<int, List<LevelPointRange>>()
{
    [0] = PointRangeGenerator.Generate(3, 01, 10),
    [1] = PointRangeGenerator.Generate(4, 05, 25),
    [2] = PointRangeGenerator.Generate(5, 15, 35),
    [3] = PointRangeGenerator.Generate(5, 25, 45),
    [4] = PointRangeGenerator.Generate(6, 35, 55),
    [5] = PointRangeGenerator.Generate(6, 45, 65),
    [6] = PointRangeGenerator.Generate(7, 55, 75),
    [7] = PointRangeGenerator.Generate(8, 65, 85),
};

private static readonly List<LevelPointRange> defaultPointRanges =
    PointRangeGenerator.Generate(5, 1, Platform.CurrentMaxHeight);
```

# PointRangeGenerator

```
public static List<LevelPointRange> Generate(int pointsAmount, float heightMin, float heightMax)
{
    var result = new List<LevelPointRange>();
    var usedRanges = new List<int>();

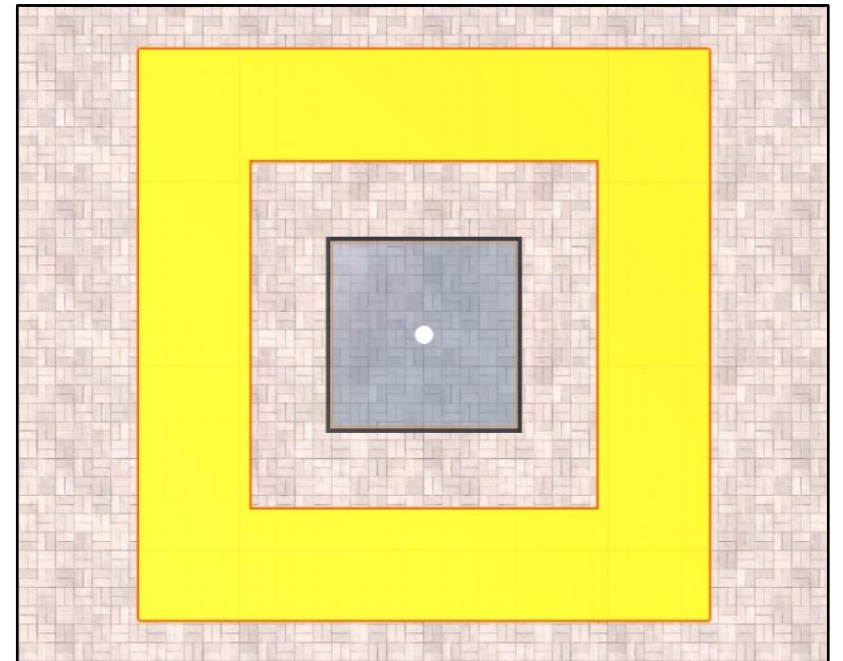
    for (int i = 0; i < pointsAmount; i++)
    {
        var index = Random.Range(0, PointsRange.Count);

        if (usedRanges.Count < 4)
            while (usedRanges.Contains(index))
                index = Random.Range(0, PointsRange.Count);
        else
            usedRanges.Clear();
        usedRanges.Add(index);

        var range = PointsRange[index];

        result.Add(new LevelPointRange(new Vector3(range.Min.x, heightMin, range.Min.y),
                                                new Vector3(range.Max.x, heightMax, range.Max.y)));
    }

    return result;
}
```



Área predefinida ao redor da plataforma  
(PointsRange)

# LogHandler e ReportManager

```
public class LogHandler : MonoBehaviour
{
    void Start()
    {
        InvokeRepeating("SaveCurrentStatus", 1f, 1f);
    }

    void SaveCurrentStatus()
    {
        ReportManager.LogStatus(
            GameHandler.GameStartTime,
            Platform.CurrentPlatformHeight,
            PlayerStatusReport.GetPlayerLookingDirection(
                Camera.main.transform.rotation.eulerAngles.x
            );
    }

    void OnApplicationQuit()
    {
        GameHandler.GetInstance().SaveGameReport();
    }
}
```

Verifica se o relatório já não foi criado, caso não tenha sido, chama o método de criação do relatório de desempenho da classe FileManager

```
public static class ReportManager
{
    private static GameReport _currentGameReport;

    public static GameReport GetCurrentGameReport() => _currentGameReport;

    public static void CreateGameReport(int expectedStages)
    {
        _currentGameReport = new GameReport(expectedStages);
    }

    public static void AddLevel(Level level)
    {
        _currentGameReport?.AddLevel(level.Stage, level.StartTime.GetValueOrDefault());
    }

    public static void EndLevel(DateTime endTime)
    {
        if (_currentGameReport?.HasCurrentLevel ?? false)
            _currentGameReport.EndLevel(endTime);
    }

    public static void AddPoint(double playerHeight, double pointHeight)
    {
        _currentGameReport?.AddPoint(DateTime.Now, playerHeight, pointHeight);
    }

    public static void LogStatus(DateTime gameStartDateTime, double height,
        PlayerLookingDirection playerLookingDirection)
    {
        FileManager.LogPlayerStatus(gameStartDateTime, height, playerLookingDirection);
    }
}
```

# Google Cardboard XR Plugin for Unity

```
public class CameraPointer : MonoBehaviour
{
    private GameObject _gazedObject = null;

    public void Update()
    {
        if (Physics.Raycast(transform.position, transform.forward, out RaycastHit hit,
            AcroboardConfiguration.PlayerViewMaxDistance))
        {
            if (_gazedObject != hit.transform.gameObject)
            {
                _gazedObject = hit.transform.gameObject;

                if (_gazedObject.CompareTag(Constants.PointSphereTag))
                    _gazedObject.SendMessage("OnLook");
            }
        }
        else
        {
            _gazedObject = null;
        }
    }
}
```



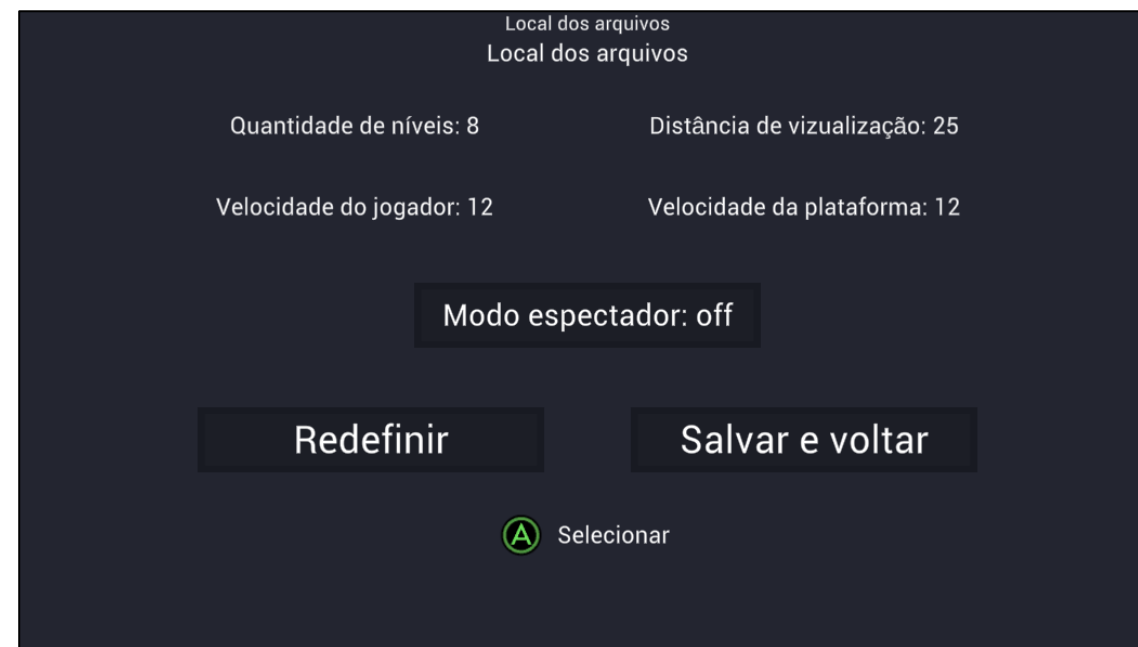
Aplicativo com visão estereoscópica

# Operacionalidade da aplicação

## Telas da aplicação



Menu inicial



Menu de configurações



Operacionalidade da aplicação

# Elementos do cenário



Esfera de pontuação

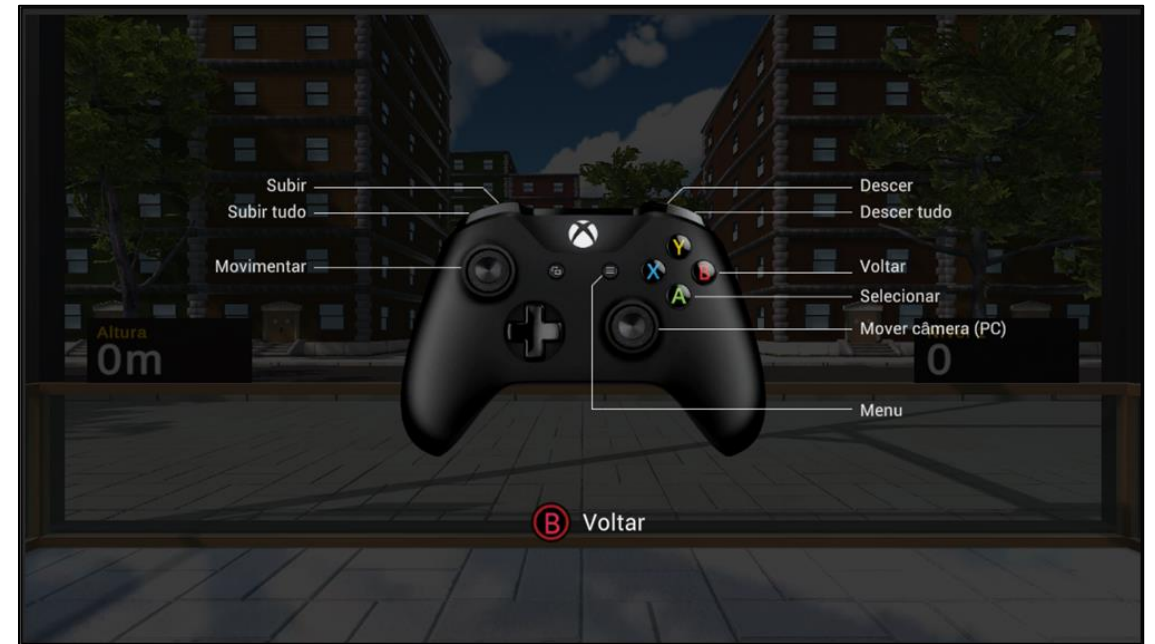


## Operacionalidade da aplicação

# Menus dentro do cenário



Menu de pausa



Mapeamento de teclas utilizadas pelo controle

# Testes de funcionalidade do aplicativo

Realizados durante o desenvolvimento no Editor Unity e no dispositivo Samsung Galaxy S20+ com o Cardboard VR Shinecon.

- Garantir a geração dos níveis e esferas;
- Comportamento de movimento do jogador com o Controle sem fio Xbox;
- Movimentação da plataforma com o Controle sem fio Xbox;
- Geração dos relatórios;
- Visibilidade dos elementos do cenário e ações da interface com visão estereoscópica.



Cardboard modelo VR Shinecon

# Testes de utilização por um profissional da área

Duas conversas foram feitas com o prof. Carlos Nunes, do curso de psicologia na FURB.

## **Antes do desenvolvimento:**

- Identificar pontos a serem abordados;
- Dificuldades enfrentadas pelos aplicativos existentes.

## **Após o desenvolvimento:**

- Validação do cenário e seus elementos;
- Possibilidade de utilização;
- Sugestões de melhorias.



Prof. Carlos Nunes utilizando o aplicativo

# Conclusões

- O aplicativo possui potencial para ser utilizado como ferramenta por psicólogos durante psicoterapias
- **Unity** se mostrou um ótimo motor de jogos para a criação do cenário e comportamentos dos elementos do cenário
- O *plug-in* **Google Cardboard XR Plugin for Unity** facilitou a aplicação da visão estereoscópica e controle de movimentação da cabeça

# Sugestões

- Adicionar mais componentes ao ambiente da cidade;
- Implementar uma versão web/desktop que se conecta ao aplicativo móvel para acompanhamento do psicólogo;
- Permitir a alteração dos valores na tela de configurações para personalizar a experiência do usuário;
- Coletar dados fisiológicos, como batimentos cardíacos ou frequência respiratória, durante a utilização do aplicativo.



Muito obrigado!