



Universidade Federal de Pernambuco

Centro de Informática

Graduação em Ciência da Computação

Análise de ferramentas e desenvolvimento de jogo para treinamento de paratletas

Deyvson Lazaro da Silva

Trabalho de Graduação

Recife, Julho de 2018



Universidade Federal de Pernambuco

Centro de Informática

Graduação em Ciência da Computação

Análise de ferramentas e desenvolvimento de jogo para treinamento de paratletas

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Sérgio Castelo Branco Soares

Co-orientador: Samuel Carlos Romeiro

Recife, Julho de 2018

Agradecimentos

Agradeço ao meu orientador, Sérgio Soares não só pelas orientações referentes a este trabalho, mas também por todas oportunidades oferecidas e orientações que transcendem o escopo deste trabalho.

À minha família, em especial a minha madrinha e a minha mãe, que sempre por todo apoio e esforço dedicado.

Agradeço especialmente a Samuel Romeiro e Adriano Gomes, que em diversos momentos serviram como verdadeiros guias durante a execução deste trabalho.

Por fim, agradeço a todos amigos que tornaram esta jornada muito mais prazerosa.

Resumo

A escolha de uma game engine adequada para o desenvolvimento de um jogo é uma etapa muito importante do processo de desenvolvimento de um jogo. Pois, apesar da escolha de uma engine cujas características não sejam as melhores para o desenvolvimento do jogo, não inviabilizar o desenvolvimento deste. Esta escolha irá impactar em todo processo de desenvolvimento e em aspectos do jogo desenvolvido. Sendo a escolha da engine, uma atividade importante do desenvolvedor de jogos. Através do levantamento e análise das game engines em destaque no mercado. Foi realizado um estudo comparativo entre as principais game engines. Analisa qual melhor game engine para desenvolvimento de um jogo para treinamento de paratletas, onde vemos que não existe uma game engine ótima para todas situações e a Unity se apresentou como melhor opção. E apresenta o desenvolvimento de um jogo para treinamento de paratletas, desenvolvido com a Unity e o Kinect, como proposta para resolver alguns dos grandes problemas no treinamento de atletas paralímpicos, que são a falta de outras pessoas com deficiência adeptas no mesmo esporte, de um local adequado para realizar os treinamentos e a falta de um instrutor. Este trabalho possui como principais contribuições, um melhor entendimento sobre o que é uma game engine, qual o cenário atual do mercado de game engines e como ocorre o processo de desenvolvimento de um jogo.

Palavras-chave: Game Engines, Serious Game, Treinamento de Paratletas, Estudo Comparativo.

Abstract

One of the most important steps for software development is the choice of an appropriate set of tools to be used. At this point, the development of games does not differ from other branches of software development. In this way, since different game engines have architectures, support tools, support and will produce different results, impacting both the development process and the results achieved. It is part of the role of the game developer to choose the right game engine according to the characteristics of the development team and the project to be developed. It is necessary a survey and analysis of the game engines in evidence. This work performs a comparative study between the main game engines. It analyzes the best game engine for the development of a game for training Paralympians, where we see that there is not an optimal game engine for all situations and Unity presented itself as the best option. And presents the development of a game for training Paralympians, developed with Unity and Kinect, as a proposal to solve a major problem in the training of Paralympic athletes, it is lack of other disabled people adept in the same sport. This work contributes to a better understanding of what a game engine is, the current state of the game engines market and the process of developing a game.

Keywords: Game Engines, Serious Game, Paralympians Training, Comparative Study.

Lista de tabelas

Tabela 1. Resultado do estudo comparativo entre game engines	24
--	----

Lista de figuras

Figura 1 Arquitetura de game engine	12
Figura 2 Arquitetura de game engine	13
Figura 3 Treino de flexibilidade em pausa	33
Figura 4 Treino de reação	34
Figura 5 Treino de resistência em pausa	35
Figura 6 Treino de agilidade em pausa	36
Figura 7 Treino de força	37
Figura 8 Cadastro	40
Figura 9 Trocar usuário	40
Figura 10 Menu principal	41
Figura 11 Ranking	41
Figura 12 Perfil do atleta	42
Figura 13 Opções de tela	42
Figura 14 Opções do Kinect	43
Figura 15 Opções de áudio	43
Figura 16 Selecionar treino	44
Figura 17 Preparação Flexibilidade	44
Figura 18 Treino de Flexibilidade Finalizado	45

Índice

1. Introdução	9
1.1 Contexto e motivação	9
1.2 Objetivo	10
1.3 Estrutura do trabalho	10
2. Referencial teórico	11
2.1 Desenvolvimento de jogos	11
2.2 Game Engines	12
2.2.1 Unity	14
2.2.2 Unreal	15
2.2.3 Cryengine	16
2.2.4 Lumberyard	17
2.3 Escolha dos critérios para comparação	18
3. Comparação entre Engines	20
3.1 Parâmetros comparativos	20
3.2 Resultado	22
4. Desenvolvimento do jogo	27
4.1 Contexto e motivação	27
4.2 Proposta de desenvolvimento	28
4.2.1 Restrições	29
4.2.2 Requisitos	29
4.3 Desenvolvimento	31
5. Considerações finais	38
5.1 Trabalhos futuros	38
Apêndices A - Jogo desenvolvido	40
Referências	46

1. Introdução

Este capítulo tem como objetivo apresentar o contexto e motivação deste trabalho, assim como também esclarecer seus objetivos e a organização da estrutura do documento. Este capítulo está dividido em três seções. A primeira seção, apresenta o contexto e motivação para o estudo realizado, assim como para a aplicação desenvolvida. Os objetivos deste trabalho são apresentados na segunda seção. E por fim, a estrutura deste trabalho é descrita na terceira seção.

1.1 Contexto e motivação

Uma das etapas mais importantes para o desenvolvimento de um software é a escolha de um conjunto adequado de ferramentas a serem utilizadas. Neste ponto, o desenvolvimento de jogos não difere dos demais ramos de desenvolvimento de software.

Jogos 3D são softwares inerentemente complexos, precisando executar em tempo real e conectar uma série de módulos bastante variados. Muitas vezes, estes módulos representam o estado da arte na área das pesquisas em ciência da computação [1]. Desta forma, a escolha de uma game engine, ferramentas que conectam, abstraem e simplificam a implementação e utilização destes módulos, é muito importante para o desenvolvimento de um jogo. Uma vez que diferentes game engines possuem arquiteturas, ferramentas de apoio, suporte e produzirão resultados diferentes, impactando tanto no processo de desenvolvimento quanto nos resultados atingidos. Faz parte do papel do desenvolvedor de jogos, escolher a game engine adequada de acordo com as características da equipe de desenvolvimento e do projeto a ser desenvolvido. Sendo necessário um levantamento e análise das game engines em evidência.

Beneficiando-se das características normalmente associadas a jogos digitais, como execução em tempo real, imersão e diversão, existe uma série de aplicações que buscam simular situações reais. Estas aplicações têm a finalidade de melhorar o desempenho dos jogadores na execução de suas tarefas diárias, e são normalmente denominadas serious games.

Com a finalidade de prover uma melhor qualidade ao treinamento individual de paratletas com dificuldades de acesso a um grande centro de treinamento, uma empresa de Minas Gerais, buscou o SENAI, através do Edital de Inovação para Indústria, onde o projeto foi aprovado e desenvolvido em conjunto entre o Instituto SENAI de Inovação para Tecnologias da Informação e Comunicação (ISI-TICs) e o Senai CECOTEG. Como a propriedade intelectual da aplicação

desenvolvida pertence a empresa que propôs o projeto, este trabalho não demonstrará códigos relacionados ao jogo desenvolvido e o nome do jogo nas imagens será ocultado.

1.2 Objetivo

Este trabalho destina-se a análise de *game engines* para o desenvolvimento de um jogo de treinamento de paratletas da categoria basquete em cadeira de rodas. Como resultado será entregue um framework para escolha de *game engines*, assim como demonstração e análise de um jogo desenvolvido utilizando uma das *game engines* analisadas que se demonstrou adequada em relação aos requisitos do jogo proposto.

Com isto, serão estudadas características técnicas de diversas *game engines*, envolvendo desde aspectos como desempenho, linguagem de programação envolvida, integrações, tamanho da comunidade e modelo de negócio da *engine*, dentre outros aspectos. Além disto, será realizado um estudo de caso através da escolha de uma engine e desenvolvimento de um jogo para treinamento de paratletas, que foi proposto por uma empresa de Minas Gerais em uma parceria entre o ISI-TICs e o Senai CECOTEG.

1.3 Estrutura do trabalho

Este trabalho é dividido seguindo a seguinte descrição. O Capítulo 2 apresenta o referencial teórico acerca dos conceitos abordados ao longo deste trabalho, apresentando suas características e o estado atual da arte. Um estudo comparativo entre as *game engines*, apresentando os trabalhos relacionados, explicando as diferenças entre eles, o que absorvemos para este trabalho e o que faremos de diferente é apresentado no Capítulo 3, que também realiza uma análise das *game engines* apresentadas e seleção de uma das *game engines* para desenvolvimento da aplicação. O Capítulo 4 apresenta a solução desenvolvida assim como justifica as ferramentas selecionadas para auxiliar no desenvolvimento da solução apresentando os resultados alcançados. Por fim, o Capítulo 5 conclui este trabalho e apresenta os possíveis trabalhos futuros e melhorias.

2. Referencial teórico

Este capítulo tem como objetivo abordar o conteúdo necessário para o entendimento deste trabalho como um todo. Este capítulo está dividido em três seções: a primeira aborda conceitos relacionados a área de desenvolvimentos de jogos, a segunda apresenta os conceitos relacionados a game engines e descreve as características das principais game engines disponíveis no mercado, e a terceira apresenta os trabalhos relacionados, suas características e como eles devem servir de base para os objetivos deste trabalho.

2.1 Desenvolvimento de jogos

Dentro da academia existem várias definições sobre o que é um jogo. Alguns acadêmicos definem como um sistema fechado que representa um subconjunto da realidade [2]. Há também outros acadêmicos como Bernard Suits [3] que defende que jogar é um esforço voluntário para superar obstáculos desnecessários. Porém estas definições falham para englobarem jogos atuais ou para delimitarem a partir de que ponto algo pode ser considerado um jogo. Para todos fins utilizaremos a definição dada por Scott Rogers, que define um jogo como uma atividade que requer ao menos um jogador, possui regras e uma condição de vitória.

Apesar das definições sobre o que é um jogo, ferramentas e aplicações são definidas como um jogo de forma mais tácita, onde frequentemente surgem aplicações que quebram várias das diferentes definições de jogo. O próprio Scott Rogers [4] demonstra que um jogo não é necessariamente um conjunto da realidade ou um esforço desnecessário, sendo necessário algo mais para estas ferramentas e ações serem consideradas como parte de um jogo. Porém, podemos facilmente encontrar exceções a definição de Scott Rogers, como por exemplo, o jogo Flappy bird, pertencente à categoria de jogos conhecida como infinity runner, que são jogos que consistem em movimentar-se infinitamente para uma certa direção, evitando alguns obstáculos que possam aparecer, tornando-se difícil dizer que existe uma condição de vitória em jogos desta categoria. Portanto, seguiremos com a definição de um jogo como sendo uma atividade com aspectos lúdicos, que envolve ao menos um jogador e um conjunto de regras.

Embora das definições de jogos geralmente não incluam o fator diversão, como um parâmetro para classificar algo como um jogo, os jogos normalmente são apresentados como uma ferramenta de diversão e entretenimento. Porém, existem linhas de pesquisa e de desenvolvimento

voltadas para uma categoria de jogos que não possuem a diversão e o entretenimento como objetivos primários, uma destas categorias de jogo é denominada serious games [5]. Definida por David Michael e Sande Chen [6], serious games são jogos pensados explícita e cuidadosamente com um propósito educacional. Não possuindo diversão e entretenimento como objetivos primários, o que não significa que os mesmos devam abdicar dessas características.

A indústria de jogos tem se desenvolvido fortemente, tanto nas aplicações voltadas para diversão quanto em diversas áreas relacionadas a educação, setores industriais e etc., se tornando uma indústria altamente competitiva. No entanto, devido a uma série de características comuns a jogos digitais, como renderização 3D, sonorização, modelos físicos genéricos, sincronização de tarefas, tratamento de colisões e etc., o reuso de código foi se tornando cada vez mais comum e necessário, fazendo com que essas características começassem a ser agrupadas em módulos e o agrupamento desses módulos resultou no surgimento das game engines [7].

2.2 Game Engines

Games Engines são ambientes de desenvolvimento (IDE) responsáveis por gerenciar o game loop, possuindo uma coleção de módulos e pode ser usada para criação de jogos diferentes sem que seja necessário realizar grandes modificações.

A Figura 1 apresenta o formato de um game loop em seu estado mais básico, um game loop nada mais é do que o processo básico do jogo, que representa seu ciclo de vida. Apesar de normalmente os game loops implementados pelas game engines serem relativamente complexos, todos seguem o mesmo algoritmo básico, resumindo-se a processar entradas do jogo, atualizar o estado do jogo, renderizar o jogo em seu estado atual e repetir estes processos novamente [8].

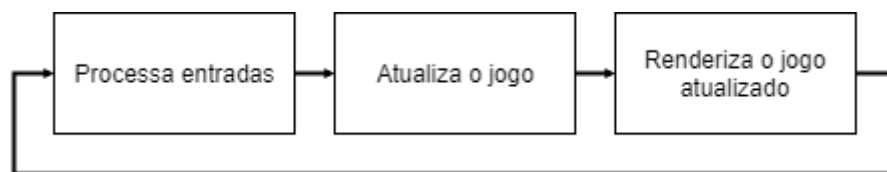


Figura 1 Arquitetura de game engine

Fonte: Adaptação de Valente [8]

A definição de uma game engine não é regida quanto a sua arquitetura, ainda assim é possível observar um padrão arquitetural comum as principais game engines. A Figura 2 demonstra a estrutura de uma game engine.

Os blocos brancos representam recursos externos a engine, mas que são utilizados pelo jogo durante a sua execução, contendo hardware, drivers, sistema operacional e bibliotecas de terceiros. A utilização da SDK DirectX para gráficos 3D, pela game engine, como uma camada de acesso ao driver de vídeo de um computador com Windows, é um exemplo de como as engines utilizam essas camadas.

Os blocos em azul representam camadas internas a game engine. A core systems representa um pacote de utilitários como gerenciamento de memória e verificação de erros. A camada de recursos fornece uma interface para acesso aos recursos do jogo, como imagens e sons. Os módulos se referem a módulos já oferecidos pela game engine ou criados pelos desenvolvedores do jogo, como por exemplo gerenciamento de áudio ou sistema de armas. Por fim a cena representa o mundo onde o jogo ou nível do jogo se passa [9].

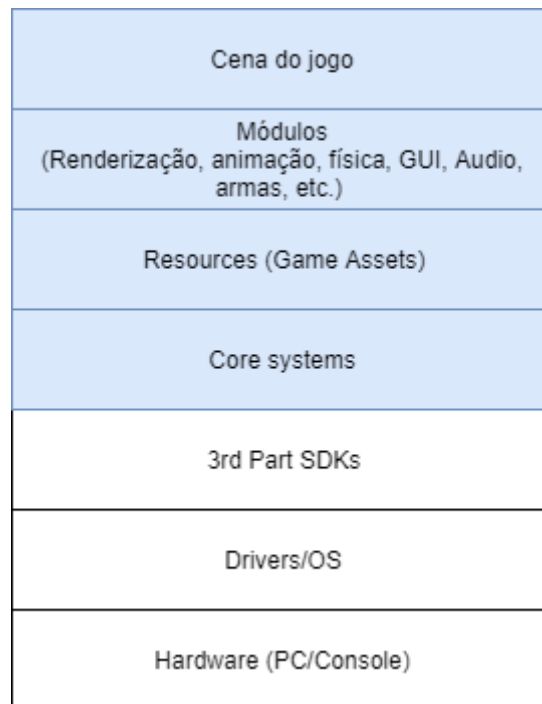


Figura 2 Arquitetura de game engine

Fonte: Adaptação de Gregory [9]

Existe hoje uma série de game engines com características bem distintas, onde a escolha correta de uma game engine se torna um fator bastante importante no desenvolvimento de um projeto. Apesar do grande número de game engines disponíveis no mercado, existem algumas que

se destacam, dentre elas estão CryEngine, Unreal Engine, Unity3D e a Lumberyard, que serão apresentadas nas seções a seguir.

2.2.1 Unity

A Unity possui uma série de produtos e ferramentas para auxiliar desenvolvedores de jogos nas diversas fases de desenvolvimento. Seu principal produto, a Unity engine, atualmente em sua versão 2018.1, é uma das game engines mais populares do mundo [10]. Contando com uma vasta gama de jogos desenvolvidos na engine e alguns títulos de destaque [11].

Dentre os títulos desenvolvidos na Unity podemos destacar, o jogo Monument Valley 2, detentor do título de melhor jogo mobile de 2017 e um dos melhores jogos chineses do mesmo ano, segundo o The Game Awards, um dos eventos mais importantes para a comunidade de jogos digitais ao redor do mundo. Outro grande exemplo de jogos desenvolvidos na Unity é o Cuphead, que durante o The Game Awards 2017, foi vencedor das categorias de melhor direção de arte, melhor jogo indie e melhor jogo indie debutante, figurando também entre os finalistas das categorias melhor trilha sonora e melhor jogo de ação [12].

A Unity Engine é bastante flexível permitindo importar artes e assets em diversos formatos, suportando modelos 3D nos formatos .fbx, .dae (Collada), .3ds, .dxf, .obj, .skp, 3D Studio Max, Maya, Blender, Cinema4D, Modo, LightWave e Cheetah3D. Trabalhando diretamente com áudios nos formatos .mp3, .ogg, .wav, .aiff, .aif, .mod, .it, .s3m e .xm. Além de Imagens nos formatos BMP, TIF, TGA, JPG, e PSD [13].

Além da grande flexibilidade nos formatos de arquivos suportados, a Unity é uma game engine cross-platform, dando suporte ao desenvolvimento de jogos para IOS, Android, Windows, Windows Store, Mac OS X, Linux, WebGL, Playstation 4, PsVita, Xbox One, 3DS, Oculus Rift, Google Cardboard, Steam, Playstation VR, Samsung Gear VR, Windows Mixed Reality, Daydream, Android TV, Samsung SMART TV, TVOS, Nintendo Switch, Fire OS 5, Facebook Gameroom, ARKit, ARCore e Vuforia [14].

A unity engine possui um editor com uma série de ferramentas integradas e possibilidades de extensões, que permitem que artistas, designers e programadores trabalhem dentro do mesmo ambiente. Contando com ferramentas de arte e design como o Probuilder que une ferramentas de modelagem e design de níveis [14]. Além de contar com uma loja de assets que permite comprar pacotes de arte, modelos, scripts, ferramentas de produtividade, fragmentos de jogos e demais

conteúdos que possam ser importados na engine, possuindo um total de mais de 50 mil assets disponíveis [15].

Em complemento a engine, a unity também dispõe de vários produtos que complementam o trabalho da engine e abarcam outras áreas, como serviços de hospedagem, analytics, relatórios de performance, monetização, uma rede social profissional e etc.

Além de todas ferramentas oferecidas, a Unity também busca incentivar a iniciação e desenvolvimento de estudantes e entusiastas da área de jogos. Contando com guias, tutoriais e documentação que englobam as linguagens oficialmente suportadas (C#, Unityscript, Boo). Disponibiliza projetos completos de exemplo. Organiza sessões ao vivo, workshops, cursos e certificações. Oferece recursos gratuitos para alunos, professores e instituições acadêmicas [16]. E mantém uma base de conhecimento pública com dúvidas respondidas acerca de seus serviços, contas, licenças, asset store, comunidade e editor [17].

A Unity engine está disponível gratuitamente para usuários e empresas com receita bruta anual ou fundos abaixo de 100 mil dólares, oferecendo suas principais funcionalidades, Ads, compras dentro da aplicação, todas plataformas e multiplayer para até 20 jogadores ao mesmo tempo. Se o usuário ou empresa possuir receita ou fundos acima de 100 mil dólares, ou quiser ter acesso ao código fonte da engine e demais funcionalidades, é necessário pagar uma licença mensal [18].

2.2.2 Unreal

A Possuindo como um de seus principais produtos, a Unreal engine, atualmente na versão Unreal Engine 4. A Unreal Engine é uma das game engines mais populares do mundo, lado a lado com a Unity.

Além da sua popularidade, também possui uma série de premiações e jogos de sucesso desenvolvidos utilizando a engine. Recebendo o título de melhor game engine do ano na Develop Industry Excellence Awards em oito das últimas dez edições, não sendo vencedora desta categoria apenas nas edições de 2014 e 2015 [19], situando-se novamente como uma das finalistas para edição de 2018 [20]. Possuindo também uma vasta quantidade de jogos em seu portfólio. Sendo representada por 16 jogos durante a The Game Awards 2017, onde podemos destacar os premiados: Injustice 2, Hellblade: Senua's Sacrifice, PlayerUnknown's Battlegrounds e What Remains of Edith Finch [21].

A Unreal permite desenvolvimento para 21 plataformas, sendo estas: Playstation 4, Nintendo Switch, Xbox One, Mac OS X, Steam, Oculus Rift, Playstation VR, Samsung Gear VR, ARCore, Daydream, Magic Leap, SteamVR, HTC Vive, Linux, HTML5, Windows, Android, IOS, ARKit, Facebook Gameroom e PsVita [22, 23].

A Unreal disponibiliza também gratuitamente uma série de assets do jogo Paragon, estes assets possuem custo estimado de mais de 12 milhões de dólares [24]. Além permitir compra de plugins, projetos de exemplo, artes e demais materiais desenvolvidos pela comunidade e compatíveis com a engine diretamente na sua loja de assets[25], contabilizando mais de 4800 assets. Além do material disponível em sua loja de assets, a Unreal permite importação dos principais formatos de arquivos do mercado. Suportando modelos 3D no formato .FBX, sons no formato .wav e texturas nos formatos .bmp, .float, .pcx, .png, .psd, .tga, .jpg, .exr, .dds e .hdr [23].

A engine e todos seus recursos estão disponíveis gratuitamente, porém requer que seja pago royalties no valor de 5% do valor bruto das vendas para cada produto desenvolvido na engine. Não contabilizando os primeiros três mil dólares de cada trimestre para cada produto e os primeiros cinco milhões de dólares para cada produto na Oculus Store [26].

Além de todos os recursos da engine, a Unreal também oferece valores de cinco mil a 50 mil dólares, sem restrições, através do programa Unreal Dev Grants, com um fundo total de cinco milhões de dólares [27]. E disponibiliza gratuitamente a Unreal Studio em sua versão beta [28].

Por fim, todos os recursos da Unreal Engine também estão disponíveis para educadores, estudantes e instituições de ensino, contando com uma licença, recursos e instalação própria para o meio acadêmico que contém alguns recursos extras além de todos recursos disponíveis para os demais usuários da game engine[29]. Todos usuários da Unreal engine ainda contam com documentação, tutoriais em vídeo e fóruns disponibilizados pela Unreal.

2.2.3 Cryengine

A Cryengine, engine desenvolvida pela Crytek Company, é conhecida como uma das game engines de melhor desempenho dentre as disponíveis no mercado. Sendo reconhecida também pelos títulos de sucesso desenvolvidos utilizando a game engine, onde podemos citar o jogo The Climb [30], um dos 5 melhores jogos de ação do ano de 2017 durante a The game awards [12].

Apesar de possuir um nome de peso no mercado de jogos, a Cryengine não possui uma comunidade tão grande e ativa quanto a Unity e Unreal, assim como também deixa a desejar em

relação a qualidade de sua documentação, fazendo com que o acesso a informações sobre a engine seja mais escasso.

A Cryengine, atualmente em sua versão Cryengine V, é open source e está disponível gratuitamente sendo necessário pagar royalties de 5% sobre o valor de revenda bruta dos títulos desenvolvidos na engine após estes títulos atingirem uma receita bruta de 5 mil dólares. Porém é necessário entrar em contato com a Crytek para desenvolver jogos da categoria serious games e aplicações cujo principal foco não é o entretenimento [31].

A Cryengine é uma game engine cross-platform permitindo desenvolvimento para 8 plataformas, sendo elas: Windows, Linux, PlayStation 4, Xbox One, Oculus Rift, OSVR, PSVR e HTC Vive. Além das plataformas para as quais a CryEngine permite desenvolvimento, ela também se demonstra bastante flexível em termos dos formatos de arquivos que permite importar em um projeto. Dando suporte para modelos 3D nos formatos .FBX, .DXF, .DAE, .OBJ e 3DS Max. Além de suportar áudios nos formatos .wav e Ogg, e cerca de 35 formatos de imagens [32].

Além do código fonte, também estão disponíveis no site da Cryengine, tutoriais e uma documentação completa em todas linguagens disponíveis na engine, C++, C# e Lua[32]. Contando também com programas de certificação para artistas técnicos, designers e programadores [33]. E por fim, uma loja de assets, denominada marketplace, que apesar de ainda estar em fase beta, conta com mais de 1200 assets disponíveis [33].

2.2.4 Lumberyard

A Amazon Lumberyard, game engine gratuita e open source desenvolvida pela Amazon é a mais nova dentre as game engines que abordamos [34]. Atualmente em versão beta, a Lumberyard foi lançada em fevereiro de 2016 [35]. Baseada na CryEngine, a Lumberyard começou a ser desenvolvida pela Amazon após licenciamento da Crytek em 2015 [36].

Apesar do pouco tempo de existência, a Lumberyard já figura entre uma das principais game engines do mercado, sendo uma das finalistas ao título de game engine do ano [37]. Porém, ainda assim problemas relativos ao seu pouco tempo de existência são bastante aparentes, como a falta de uma loja de assets, artigos relacionados a engine e conteúdo criado pela comunidade. O que limita bastante o acesso a engine, se tornando uma das barreiras relacionados ao aprendizado e adoção da engine.

Um dos pontos de destaque da Lumberyard é a integração com serviços da Amazon como Amazon Gamelift e serviços da Twitch [38]. Além disto, Amazon tem focado em melhorar o acesso a engine provendo faq, documentação e tutoriais, assim como desenvolvendo novas features que facilitam o uso da engine, como script canvas que permite desenvolvimento através de visual script como uma alternativa as linguagens C++ e Lua já presente na engine [39, 40].

A Lumberyard suporta os principais formatos imagens, sons e modelos 3D do mercado. Permitindo utilização de imagens nos formatos .bmp, .jpg, .pgm, .png, .raw, .r16, .tga e .tif. Modelos 3D nos formatos .cgf, .chr, .skin, .fbx, .abc, .cax e .trb, sendo compatível também com Maya e 3DS Studio. Para trabalhar com áudios a Lumberyard utiliza uma camada de tradução de áudio como interface entre a Lumberyard e o Wave Works Interactive Sound Engine (Wwise) que suporta o formato .WAV [41].

Por fim, a Amazon Lumberyard é uma game engine cross-platform e permite desenvolvimento para PC, Xbox One, PlayStation 4, iOS (iPhone 5S+ e iOS 7.0+) e Android (Nexus 5 e versões equivalentes compatíveis com o OpenGL 3.0+) [39]

2.3 Escolha dos critérios para comparação

Para escolha dos critérios a serem usados na comparação entre as engines, realizamos um levantamento de trabalhos de pesquisa que tiveram como foco essa ação. Com base nesse levantamento, identificamos quais critérios foram usados. Assim, a maioria dos critérios foram retirados de trabalho anteriores de comparação entre engines.

Daniel Valente, Yvens Rebouças e Maria Andréia [42] realizaram uma comparação entre a Unity 3D, CryEngine e Unreal Engine 4. Onde foram utilizados escalabilidade, linguagem de programação, portabilidade, curva de aprendizado e custo/mês e licença como parâmetros comparativos. Porém, tais parâmetros são insuficientes para tomada de decisão durante um projeto.

Anis Zarrad [43] apresentou um estudo onde comparava 20 game engines, onde foram utilizadas modularidade, usabilidade, eficiência, recursos e qualidade das imagens e efeitos de renderização.

Petridis et al. [44] realizou um estudo comparativo entre game engines para serious games de alta fidelidade, durante o estudo foram comparadas CryEngine, Source Engine, Unreal e Unity, este trabalho se destaca pela quantidade e completude das características comparadas. Durante o

estudo os parâmetros de comparação foram agrupados em cinco diferentes tópicos. Fidelidade audiovisual, fidelidade funcional, compatibilidade, acessibilidade, network e heterogeneidade. Possuindo um total de 26 variáveis comparativas.

Apesar da abrangência dos estudos anteriores, muitos dos seus resultados precisam ser revistos devido a atualizações das game engines analisadas e surgimentos de novas. Porém as metodologias e grande parte dos parâmetros utilizados para comparações ainda continuam válidos.

Desta forma, neste trabalho pretende-se apresentar um framework que reflita melhor o estado atual das ferramentas analisadas, uma vez que elas se desenvolveram muito nos últimos anos. Além de analisar parâmetros desconsiderados nos trabalhos anteriores.

3. Comparação entre Engines

Este capítulo apresenta o estudo comparativo entre as games engines apresentadas no capítulo anterior. Este capítulo está dividido em duas seções. A primeira seção apresenta e descreve os parâmetros de comparação. Os resultados das comparações e a escolha de uma game engine para o desenvolvimento de um jogo são apresentados na segunda seção.

3.1 Parâmetros comparativos

Com base nos estudos anteriores, utilizaremos como parâmetros: Sistema de script, curva de aprendizado, licenciamento e custo, documentação e suporte, heterogeneidade, compatibilidade e qualidade visual. Além dos parâmetros utilizados, devemos levar em consideração que os ecossistemas das game engines também se desenvolveram fortemente neste período, desta forma também adicionaremos loja de assets e o ecossistema ao redor da engine. Os parâmetros selecionados serão explicados nos tópicos a seguir:

Sistema de script

Muitas engines oferecem um sistema de script, pelo qual é possível definir as regras do jogo, seja através de uma linguagem de programação tradicional ou alguma ferramenta para construção de algoritmos de forma visual.

Curva de aprendizado

A curva de aprendizado diz respeito a quão difícil é o aprendizado de uma game engine para desenvolvedores que possuem pouca ou nenhuma experiência prévia com game engines. Para se analisar uma curva de aprendizado de uma game engine, podemos usar alguns parâmetros como quantidade e preço dos assets disponíveis, tutoriais e guias para iniciantes criados pelos desenvolvedores ou pela comunidade da engine, completude e qualidade da documentação, nível de abstração do sistema de script e complexidade da arquitetura da engine.

Para classificar a curva de aprendizado das engines utilizaremos três níveis: baixa, alta e média. A classificação reflete a dificuldade de desenvolvimento para iniciantes em relação a outras engines.

Licenciamento e Custo

O licenciamento especifica o modelo de pagamento da utilização da game engine, como mensalidade ou royalties por exemplo. Assim como as limitações sobre o uso da engine, como tipos e finalidade dos projetos desenvolvidos utilizando-se a engine, modelos de licença que podem ser utilizados, formatos de distribuição e restrições de acesso e modificações ao código-fonte.

Documentação e suporte

A variável comparativa relativa a documentação e suporte, descreve a forma como a documentação da game engine está disponível, descrevendo se existe uma documentação oficial e se completa, tutoriais relacionados a documentação, guias para iniciantes, projetos de exemplo, FAQs e suporte técnico.

Heterogeneidade

A heterogeneidade descreve as plataformas de destino para as quais é possível se desenvolver utilizando a engine.

Compatibilidade

No parâmetro compatibilidade, descrevemos os formatos de arquivos suportados como assets do jogo, como formatos de sons, imagens, vídeos e modelos.

Qualidade visual

O parâmetro qualidade visual, reflete o poder de renderização da engine. Representando a capacidade de renderização fotorrealística, fidelidade das sombras, reflexão das luzes, capacidade de criação de shaders, sistema e partículas, dentre outros atributos.

Para classificar a qualidade visual das engines utilizaremos 3 níveis, baixa, alta e média. Onde a classificação, reflete a capacidade da engine de simular ambientes da forma mais realista o possível, em relação as outras engines comparadas, onde possuir um nível baixo, não significa que possui um baixo poder de renderização. Apenas que as outras engines estão mais evoluídas neste quesito.

Loja de assets

A loja de assets é um ambiente integrado com a engine que permite comprar, baixar e importar assets ou projetos inteiros para serem utilizados durante o desenvolvimento de um jogo.

Ecossistema

Por fim, o ecossistema descreve as ferramentas e serviços externos a engine que complementam suas features, como por exemplo, serviço de hosting, analytics e publicidade.

3.2 Resultado

Uma vez que não faz parte do escopo deste trabalho testar manualmente cada uma das engines, para os atributos curva de aprendizado e qualidade de visual também serão consideradas materiais que indiquem a opinião da comunidade de desenvolvimento de jogos, como discussões em fóruns, matérias em blogs, artigos e trabalhos de graduação [45, 46, 47].

A Unity é a game engine que possui a maior comunidade, contando com uma quantidade massiva de tutoriais desenvolvidos pela comunidade. Possui uma documentação de alto nível, com exemplos de código e tutoriais acerca dos principais conceitos. Conta com mais de 50 mil assets em sua asset store. A Unity também conta com C#, uma linguagem mais abstrata que C++ e Lua e uma interface mais simples que as demais. Por outro lado, é a única game engine que não possui visual script, se tornando um pouco menos amigável para não programadores. Mesmo contando com algumas limitações, dentre as game engines é a que permite desenvolver para mais plataformas e possui mais opções de importação de arquivos. Segundo informações obtidas nas threads de discussões de fóruns de desenvolvimento de jogos, artigos e blogs, a Unity é atualmente a game engine mais amigável entre as comparadas, sendo geralmente recomendada para iniciantes ou desenvolvedores independentes. Em termos de qualidade visual, ela é frequentemente citada em blogs e artigos, como inferior às demais quando se fala em termos de fotorrealismo e processamento de quantidade massivas de partículas.

A Unreal é a segunda game engine mais popular dentre as analisadas. Possui uma boa documentação, uma comunidade bastante ativa e disponibiliza seu código fonte no github para usuários da engine. Apesar de possuir C++ como linguagem de programação, grande parte do desenvolvimento pode ser realizado utilizando Blueprints facilitando parte do desenvolvimento. Possui uma loja de assets com uma boa quantidade de assets, porém em quantidade relativamente

abaixo da Unity. Permite desenvolvimento para as principais plataformas e possui uma boa quantidade de formatos de importação. Em relação a usabilidade, a Unreal é frequentemente relacionada como mais difícil que a Unity para desenvolvedores iniciantes, porém mais fácil de se trabalhar com alguns componentes como manipulação de terrenos, partículas ou cinematics. Por outro lado, é apontada como uma das game engines mais poderosas em termos de renderização. A Unreal é recomendada para desenvolvimento de jogos com altos requisitos gráficos ou fotorrealísticos, possuindo uma curva de aprendizado um pouco maior que a da Unity para desenvolvimento do gameplay, mas ainda viável para iniciantes.

A CryEngine apesar de ser uma das engines mais populares do mercado, possui uma das menores comunidades entre as engines analisadas, o que reflete na quantidade de tutoriais encontrados e de assets em sua asset store. Devido a grande competitividade oferecida pela Unreal, Unity e mais recente pela Lumberyard, a CryEngine tem remodelado sua licença, adotando um modelo de licença semelhante a Unreal e se tornando open source. Apesar de contar com uma documentação completa e tutoriais, a opinião da comunidade em relação a usabilidade da CryEngine é que a sua documentação não é suficientemente amigável, sendo recomendada apenas para profissionais experientes na área. O ponto de destaque da CryEngine é seu poder gráfico, sendo considerada uma opção a Unreal para desenvolvimento de jogos com altos requisitos gráficos ou fotorrealistas, geralmente lembrada em cenários de ambiente abertos.

Por fim, a Lumberyard é a game engine que possui a menor comunidade dentre as analisadas, sendo a única completamente gratuita, porém, também é única que não possui uma loja de assets. Apesar de ter nascido como um fork da CryEngine, em seus dois anos de existência tem se distanciado da CryEngine. A Amazon tem focado no desenvolvimento de ferramentas mais amigáveis, integrações e na melhoria da documentação da engine. Possui qualidade de visual próxima a CryEngine e já começa a ocupar espaço dentro da comunidade de jogos. Porém, mesmo com os esforços da Amazon, o pouco tempo de existência da engine, faz com que seja difícil encontrar artigos e materiais relativos a engine. A Lumberyard é recomendada para desenvolvedores experientes e em projetos de jogos que possuem requisitos gráficos elevados ou se beneficiam das integrações com serviços da AWS ou Twitch.

A Tabela 1 resume as características analisadas para cada game engine

Tabela 1. Resultado do estudo comparativo entre game engines

Característica	Unity	Unreal	CryEngine	Lumberyard
Sistema de script	C#, UnityScript(javascript) e Boo.	C++ e Blueprint.	C++, C#, Lua e Schematyc Visual Scripting.	C++, Lua e script canvas.
Curva de aprendizado	Baixa	Média	Alta	Alta
Licenciamento e Custo	Gratuito com algumas limitações e versão completa com mensalidade de 125 dólares.	Versão completa gratuita, com código disponível e pagamento de royalties de 5% após receita bruta trimestral de 3 mil dólares por jogo desenvolvido utilizando a engine. Licença especial para estudantes e instituições e ensino.	Versão completa gratuita com pagamento de royalties de 5% após receita bruta anual de 5 mil dólares por jogo desenvolvido utilizando a engine.	Gratuita
Documentação e suporte	Documentação completa, guias e tutoriais. Workshops, cursos e certificações.	Documentação completa, tutoriais e fórum.	Código fonte, tutoriais e documentação completa disponíveis.	Código fonte, tutoriais e documentação disponíveis, porém ainda em

			Certificação para designers, programadores e artistas técnicos.	desenvolvimento .
Heterogeneidade	27 plataformas	21 plataformas	8 plataformas	5 plataformas
Compatibilidade	Modelos 3D: 12 formatos. Áudios: 9 formatos. Imagens: 5 formatos.	Modelos 3D: .FBX. Áudio: .WAV. Texturas: 10 formatos.	Modelos 3D: 5 formatos. Áudio: 2 formatos. Imagens: 35 formatos.	Modelos 3D: 9 formatos. Imagens: 8 formatos. Sons: .wav
Qualidade visual	Baixa	Média	Alta	Alta
Loja de assets	Mais de 50 mil assets	Mais de 4800 assets	Mais de 1200 assets	Indisponível
Ecosistema	Unity Connect, Live Ops Analytics, Performance Reporting, Monetization e Advertising.	Unreal Studio e Unreal Dev Grants.	Sem ferramentas e ações externas.	Integração com Twitch e serviços da Amazon AWS.

Após análise das game engines, a Unity foi escolhida para desenvolvimento de um jogo para treinamento de paratletas. Devida a curta duração do projeto, os parâmetros de maior peso para escolha, foram os parâmetros relacionados a curva de aprendizado e documentação. Onde a Unity se destaca devido a qualidade da sua documentação, facilidade de encontrar materiais na internet e tamanho da sua comunidade, tornando-se mais acessível e uma escolha mais segura para o desenvolvimento.

Além disto a Unity também possui integração fornecida pela própria Microsoft com o Kinect, oferecendo uma opção mais segura de desenvolvimento de aplicações que utilizam a Unity e o Kinect, em relação a outras engines que possuem integrações não oficiais [48].

Vale ressaltar que as demais engines são frequentemente lembradas em relação ao seu poder de processamento, se destacando em cenários onde fotorrealismo e ambientes abertos fazem parte dos requisitos. Além disto, a Unreal também é lembrada pelas suas ferramentas de suporte a designers e artistas, como seu editor de partículas e seu sistema de criação de cinematics, mesmo após a Unity adicionar a ferramenta Timeline.

Porém, uma vez que fotorrealismo não faz parte dos requisitos da aplicação que se pretende desenvolver, o desempenho não é o grande diferencial e os requisitos artísticos da aplicação não são tão elevados ao ponto de exigir um sistema de partículas complexo ou uma cena de filme, a Unity se mostra a melhor opção para desenvolvimento dado que as suas demais características são iguais ou melhores que as das demais engines. Desta, forma prosseguiu-se com a utilização da Unity durante o desenvolvimento de um jogo para treinamento de paratletas. O jogo desenvolvido será descrito com mais detalhes no capítulo a seguir.

4. Desenvolvimento do jogo

Este capítulo aborda o desenvolvimento de um jogo para treinamento de paratletas utilizando a Unity engine. Este capítulo está dividido em três seções. A primeira seção aborda o contexto nacional do paradesporto, demonstrando como a falta e a necessidade de desenvolvimento de aplicações nesta área. A segunda seção apresenta uma proposta de desenvolvimento, indicando como se pretende corrigir alguns dos problemas levantados na primeira seção, assim como as restrições e requisitos do sistema desenvolvido. E por fim, a terceira seção, apresenta o jogo desenvolvido e o que foi utilizado durante o desenvolvimento.

4.1 Contexto e motivação

O Brasil é considerado uma potência mundial no cenário paraolímpico, devido aos títulos conquistados e estrutura oferecida aos paratletas de nível profissional. Porém, a iniciação no esporte adaptado possui algumas barreiras, como a falta de suporte financeiro e estrutural [48]. Mesmo com investimentos crescentes no paratletismo e dos atletas de alto rendimento possuírem acesso a centros de treinamento de qualidade, os atletas em fases iniciais não possuem acesso a uma infraestrutura e ferramentas que lhe permitam realizar treinamentos rotineiramente, principalmente aqueles que se encontram longe dos grandes centros de treinamento, presentes em poucos estados [49].

Dessa forma, é lógico pensar que os atletas devem buscar meios alternativos para realizar seus treinamentos e jogos digitais surgem como uma possibilidade. Existem muitos estudos que apresentam competências e habilidades desenvolvidas através da utilização de jogos digitais onde podemos citar menor tempo de reação, melhor desempenho relacionado às habilidades visuais básicas e a atenção, número de objetos percebidos simultaneamente, atenção seletiva e atenção dividida e maior habilidade de fazer mais de uma tarefa ao mesmo tempo [50]. Um exemplo que reforça a informação anterior é o jogo Brain Workshop, onde é possível notar um aumento de habilidades visuoespaciais e de atenção nos jogadores [51].

Complementando os estudos que indicam benefícios e habilidades relacionados ao uso regular de jogos digitais, ainda é possível encontrar alguns estudos que demonstram que indicam que ao utilizar jogos aliados a interfaces de interação natural é possível prover experiências semelhantes à de alguns esportes. Arroxelas et al. analisou a semelhança entre um jogo de boliche

no Kinect e o esporte Bocha adaptada. Nesta análise os autores concluíram que, apesar de não serem idênticos, os gestos apresentam grandes semelhanças, podendo ser utilizados para prática e desenvolvimento de algumas variáveis específicas, tornando jogos de realidade virtual um fator a ser considerado na rotina de treinamentos [52].

Porém, mesmo diante de diversos estudos que demonstram uma série de competências adquiridas através do uso regular de jogos, quando falamos sobre jogos relacionados a treinamento de paratletas, nos deparamos com uma área praticamente intocada, carecendo de estudos e aplicações desenvolvidas para este fim.

4.2 Proposta de desenvolvimento

O sistema consiste em uma solução tecnológica capaz de promover o treinamento de atletas paralímpicos conforme as suas capacidades. Uma solução composta por software e hardware que permite a interação do atleta com imagens projetadas em uma parede.

O sistema deverá funcionar como um simulador que capta os movimentos dos atletas paralímpicos e o mesmo poderá interagir com uma projeção ou tela de forma a conseguir fazer diversas atividades de treinamento esportivo. Desta forma, pode-se otimizar e diferenciar os treinamentos de força, agilidade, resistência, flexibilidade e reação. Além disso, objetiva-se auxiliar no treinamento cardiovascular dos atletas.

Também poderá ser utilizado pelos treinadores para treinar pequenos grupos de atletas, com vistas de aumentar a interação e o entrosamento do grupo ou realizar programas de treinamento individualizados, respeitando as características dos atletas e buscando desenvolver habilidades. O uso da interação através de imagens projetadas em uma parede será largamente utilizado na fase de treinamento de coordenação fina e ainda permite melhor coordenação olho-mão dos atletas cadeirantes; melhor desenvolvimento neurológico; com baixíssimo risco de lesões; sendo muito bem adaptável a qualquer idade e grau de habilidade na modalidade esportiva.

Este sistema irá amenizar e de certa forma resolver um dos grandes problemas dos treinamentos dos atletas paralímpicos, que é a falta de outras pessoas com deficiência adeptas do mesmo esporte. Normalmente, um atleta paralímpico treina em recintos e ginásios onde ele não encontra outros atletas de sua mesma categoria. Com esta solução, ele irá conseguir simular este treinamento com facilidade.

Descrição do Sistema

O sistema é uma solução tecnológica que permite o treinamento de atletas paralímpicos. Esta solução (formada por software e hardware) permitirá uma interação do atleta com imagens projetadas em uma parede. Esta tecnologia irá unir um projetor de imagens (de alta definição e com potência de pelo menos 5 mil lumens), sensores de movimento de alto desempenho (deverá não ser muito grande - no máximo 30 cm – Além de : Uma câmera RGB, para ter uma base de reconhecimento facial; Microfone embutido, para detectar mais de uma pessoa por voz; sensor de profundidade de infravermelho, que permite conhecer o ambiente em volta em três dimensões; processador e sistema próprios de controle; e detectar pelo menos 48 pontos de articulação do corpo) e um sistema de interfaces de grande fidelidade. Ademais, é importante ressaltar, que essa solução busca o desenvolvimento de tecnologia brasileira.

4.2.1 Restrições

Esta seção apresenta as restrições e premissas que foram consideradas na implementação do sistema. São elas:

1. Não haverá integração direta com os sistemas externos. As funcionalidades previstas nos sistemas desse projeto serão executadas de forma independente;
2. Os equipamentos, microcomputador, sensor e projetor devem estar ligados e conectados entre si para que o sistema opere corretamente;
3. O sistema será desenvolvido utilizando o sensor Microsoft Kinect v2 da linha Xbox One as tecnologias C# e Unity 3D;
4. O sistema de gerenciamento de banco de dados utilizado será o MySQL;
5. Algumas funcionalidades exigirão a conexão com a internet para operar corretamente;
6. O sistema deverá ser executado em um sistema operacional Windows.

4.2.2 Requisitos

Nesta seção são agrupados os requisitos que foram levantados para o desenvolvimento do sistema, descrevendo as funções que o sistema deverá realizar ou atender:

1. Permitir ao usuário cadastrar-se na aplicação. O atleta deve entrar com seus dados cadastrais como nome, modalidade e classificação funcional (0.5 | 1 | 1.5 | 2 | 2.5 | 3 | 3.5 | 4 | 4.5) [54];

2. Permitir ao usuário visualizar e alterar os seus dados cadastrais como nome, modalidade, classificação funcional;
3. Permitir a escolha de um usuário que já esteja cadastrado, caso ainda não possua usuários cadastrados, fornece a opção de cadastro de usuário;
4. Detectar e rastrear uma bola durante os treinos de agilidade, força e resistência;
5. Permitir ao atleta executar um treino do tipo flexibilidade. Este treino será composto por um círculo que identifica o limite da amplitude articular do atleta. Luzes aparecem nos extremos deste círculo e o atleta deve apagar estas luzes utilizando as mãos;
6. Permitir ao atleta executar um treino do tipo reação. Este treino será composto por um painel com uma matriz de luzes, onde o jogador deverá levar a mão até a luz que estiver acesa. As luzes acendem em pontos diferentes da matriz à medida que o atleta acerta a posição da luz que estiver acesa no momento. Durante o treino devem ser exibidos na tela um relógio executando uma contagem regressiva e a quantidade de bolas acertadas pelo atleta;
7. Permitir ao atleta executar um treino do tipo resistência. Durante este treino o atleta deverá seguir as orientações do sistema e driblar a bola de basquete alternando os lados da cadeira. O software deve detectar a posição da bola em relação ao atleta e a quantidade de dribles realizados pelo mesmo, além de exibir em tela a quantidade de dribles e um relógio executando uma contagem regressiva;
8. Permitir ao atleta executar um treino do tipo agilidade. Este treino é composto por um canhão arremessando pequenas bolas de tamanho equivalente a uma bola de baseball em direção ao atleta enquanto o mesmo realiza dribles com uma bola. Durante este treino deverão ser exibidos em tela a quantidade de dribles realizados pelo atleta, a quantidade de bolas defendidas pelo atleta e um relógio executando uma contagem regressiva;
9. Permitir ao atleta executar um treino do tipo força. Durante este treino o atleta deve posicionar a bola em círculos que irão aparecer na linha da cintura ou sobre a cabeça do atleta e permanecer um certo tempo com a bola dentro deste círculo;
10. Permitir que o usuário execute um treino no modo circuito, onde o mesmo realizará sequencialmente todos os tipos de treinos existentes para a sua modalidade;
11. Permitir que o atleta pause o treino a qualquer momento, podendo reiniciar o treino ou para a tela principal do menu;

12. Permitir ao atleta visualizar o seu desempenho, seus ganhos de força, reação, resistência, agilidade e flexibilidade, bem como sua pontuação geral;
13. Permitir ao atleta visualizar um ranking com a quantidade de pontos que ele atingiu até o momento em comparação com outros atletas que também utilizaram o sistema;
14. Permitir ao usuário cadastrar-se na aplicação utilizando uma conta do facebook;
15. Funcionar mesmo que o usuário esteja desconectado da internet e sincronizar os dados ao detectar uma conexão.

4.3 Desenvolvimento

Durante o desenvolvimento foi utilizada a Unity Engine com a linguagem de programação C#. Para rastrear o corpo do jogador foi utilizado o Kinect v2, que foi utilizado junto com a Unity através do pacote Unity Pro disponibilizada pela Microsoft [48].

Além de rastrear o corpo do usuário, também era necessário identificar objetos que o jogador poderia utilizar durante alguns treinos específicos, no nosso caso, foi necessário identificar uma bola para algumas modalidades de treino. Para identificar e rastrear o posicionamento da bola, foi utilizado o OpenCVSharp, um Wrapper de OpenCv para C# [54].

Complementando o jogo desenvolvido na Unity, também foi implementado uma API WEB em PHP para provar uma camada de abstração de alto nível para operações de armazenamento e consulta no banco MySQL. A API também ficou responsável por manusear o cadastro de atletas através do Facebook, através da *Facebook SDK v5 for PHP*, SDK oficial disponibilizada pelo Facebook [55].

A primeira funcionalidade implementada foi a de cadastro de usuários, permitindo que os usuários pudessem se cadastrar para ter acesso ao jogo, durante o cadastro o usuário deve informar o nome, e-mail, classificação funcional e poderá tirar uma foto utilizando o sistema ou utilizar uma imagem padrão. Durante o cadastro o nome e e-mail podem ser inseridos através do Facebook.

Após a finalização do cadastro de usuários foram implementadas as funcionalidades de trocar usuário, onde é exibida uma lista contendo todos usuários do sistema, permitindo a seleção de um perfil.

Em seguida, foi implementado o menu principal, que permite acessar as demais funcionalidades do sistema, nesta tela o usuário pode escolher dentre as seguintes opções:

Visualizar o ranking do sistema, acessar perfil do usuário, selecionar treinos, trocar usuário, configurações do sistema.

Após a implementação do menu principal, foi implementado o ranking do sistema, que permite ao usuário visualizar o ranking geral do sistema ou o ranking referente a sua classificação funcional.

Além de visualizar o ranking, também foi implementada a função de visualização do perfil do usuário, que permite ao usuário verificar suas informações e desempenho com um maior grau de detalhamento. Assim além da sua pontuação total, é possível também visualizar a pontuação atual e máxima dos treinos de força, resistência, reação, agilidade e flexibilidade. Nesta tela também é possível acessar a tela de edição do perfil do usuário, onde é dada a opção de edição de informações, tais quais atualizar o nome, e-mail, classificação funcional e imagem do usuário.

Além de alterar suas informações cadastrais, o usuário também pode alterar as configurações do sistema. Na tela de opções é possível selecionar a resolução e qualidade gráfica da tela do sistema, as mãos utilizadas durante os treinos e o volume geral do sistema.

Por fim foram implementados os fluxos dos treinos, começando pela funcionalidade de selecionar treino, que permite que o usuário selecione qual treino deseja executar, sendo estes os treinos de flexibilidade, reação, resistência, agilidade ou força. Além dos cinco possíveis modos de treino, também é possível escolher a opção circuito, que executa todos os demais treinos em sequência.

Desta forma prosseguimos com a implementação dos treinos, começando pelo treino de flexibilidade. Ao iniciar um treino é exibida uma imagem com instruções do treino e a opção de iniciar o treino. Assim como informações de como pausar o treino. Ao pausar um treino, a pontuação deste treino é descartada e são exibidas as opções de reiniciar o treino ou voltar para o menu principal.

Durante o treino de flexibilidade, demonstrado na figura 3 em modo de pausa. São exibidas luzes verdes na borda de um círculo, ícones de mãos que representam o mapeamento das mãos do usuário no ambiente virtual e uma contagem regressiva para o usuário tocar nas luzes que aparecem. Ao acertar um determinado número de luzes, o círculo se expande, o usuário recebe uma certa pontuação e a contagem regressiva é reiniciada. Caso a contagem regressiva chegue ao fim, o treino é finalizado.

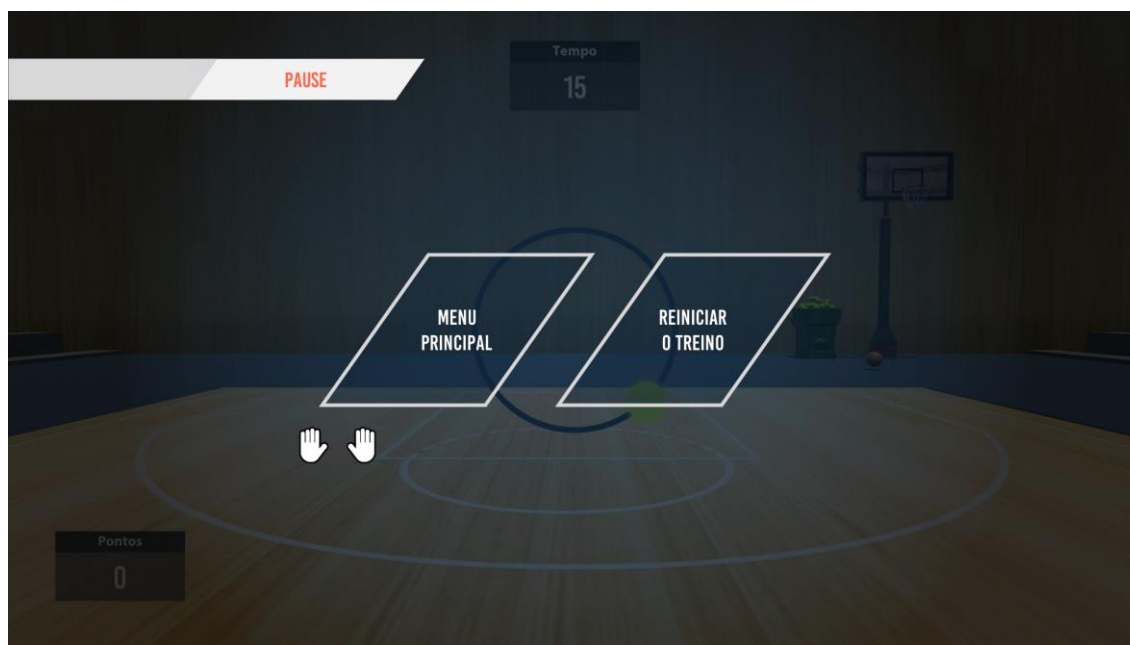


Figura 3 Treino de flexibilidade em pausa

Ao finalizar um treino é exibida uma tela com o feedback de progressão do usuário, informando o ranking atualizado, destacando o usuário atual, quantos pontos ele obteve durante o treino e as opções de treinar novamente ou voltar para o menu principal.

Após o treino de flexibilidade, foi implementado o treino de reação, demonstrado na figura 4. Durante este treino, são exibidas luzes em posições aleatórias em um painel. Ao tocar nas luzes acesas no painel o usuário recebe uma certa quantidade de pontos. O treino é finalizado ao final da contagem regressiva.

A figura 4 demonstra um momento da execução do treino de flexibilidade, onde é possível observar na parte central superior da tela um relógio que indica que falta um segundo para finalização do treino. No painel é possível observar uma luz verde indicando um local a ser tocado pelo atleta, assim como uma pontuação, +10, que indica que o atleta acabou de tocar em uma luz recebendo dez pontos. No canto inferior esquerdo da tela, é possível observar a pontuação atual do atleta durante este treino.

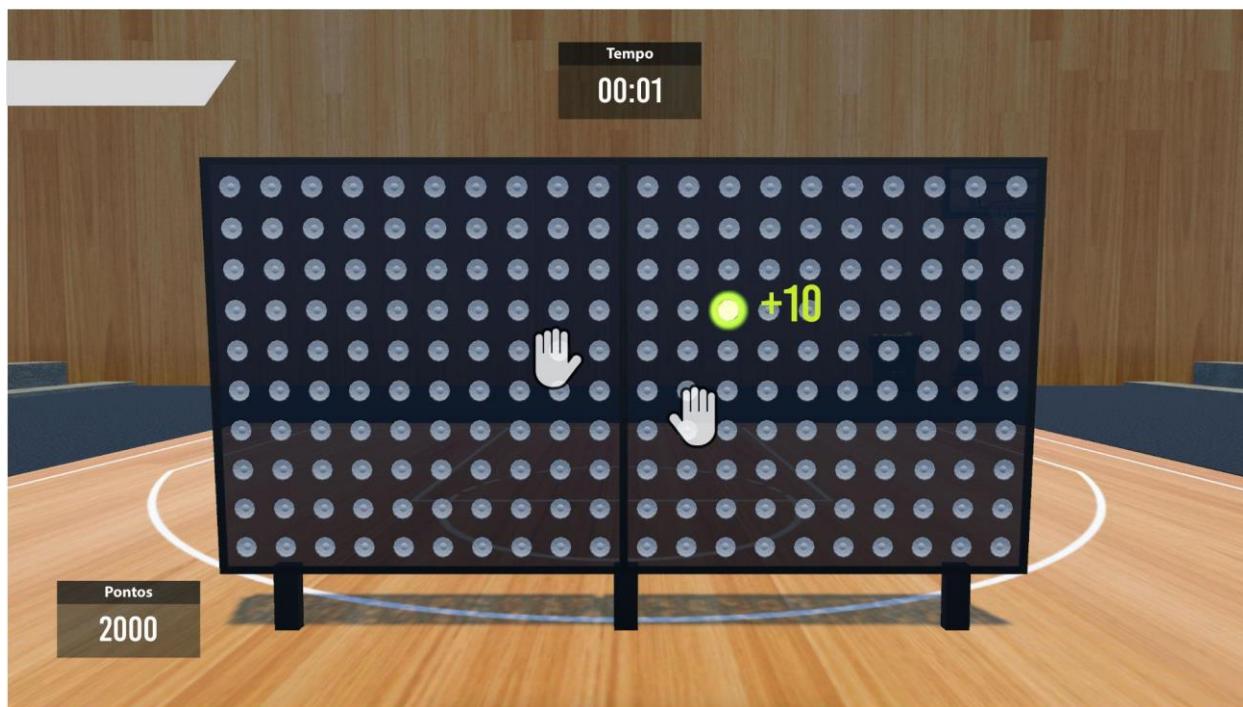


Figura 4 Treino de reação

Antes da implementação dos demais treinos, foi necessário implementar a funcionalidade de detectar bola, que permite que o usuário utilize uma bola como acessório para alguns treinos e exibe a posição da bola no ambiente virtual.

Com a detecção da bola, foi possível implementar o treino de resistência. Durante este treino, o usuário deve driblar a bola do lado indicado pelo sistema. O sistema indicará aleatoriamente se o usuário deve driblar a bola a sua direita ou a sua esquerda. Ao driblar a bola do lado correto, o usuário recebe uma certa pontuação e o sistema escolhe novamente de qual lado o usuário deve driblar a bola. O treino é finalizado ao fim da contagem regressiva.

A Figura 5 demonstra uma captura de tela do treino resistência em modo de pausa. Na figura é possível perceber que existe um modelo 3D de uma bola de basquete, indicando que uma bola foi identificada ao lado esquerdo do atleta. Uma cadeira de rodas ao centro, que facilita ao atleta a visualização da posição da bola, tornando mais fácil identificar se a bola está posicionada a esquerda ou à direita. Um +5 sobre a bola, indicando que o atleta acabou de driblar a bola correta. E por fim, é possível observar um círculo verde posicionado no chão da quadra, indicando o próximo local onde a bola deve ser driblada.

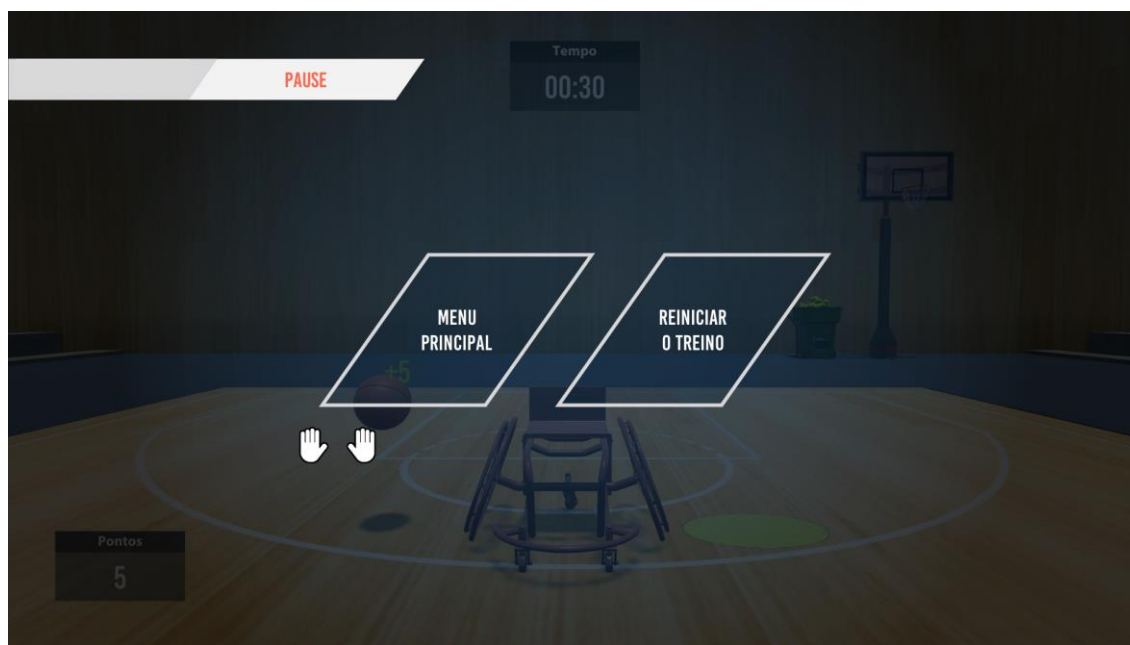


Figura 5 Treino de resistência em pausa

Logo em seguida foi implementado do treino de agilidade. Durante este treino, o usuário deve driblar a bola com uma mão e rebater bolas que o sistema lançar em sua direção. O usuário recebe uma certa quantidade de pontos cada vez que rebate uma bola ou realiza um drible. O treino é finalizado ao final da contagem regressiva.

A Figura 6 demonstra uma captura de tela do treino de agilidade durante uma pausa. Onde se observa, uma bola de basquete na parte central inferior da figura, indicando a posição em que a bola física de basquete está sendo mapeada no ambiente. Na parte central da figura existe um lançador de bolas, que irá lançar bolas na direção do atleta. Na parte central superior da figura existe um ícone de uma mão, com um círculo preto ao seu redor, que será utilizado para defender as bolas lançadas. Uma vez que o jogo está em modo de pausa, é possível observar dois ícones menores de mãos, estes são utilizados para interagir com os botões menu principal e reiniciar o treino, localizados no centro da figura.

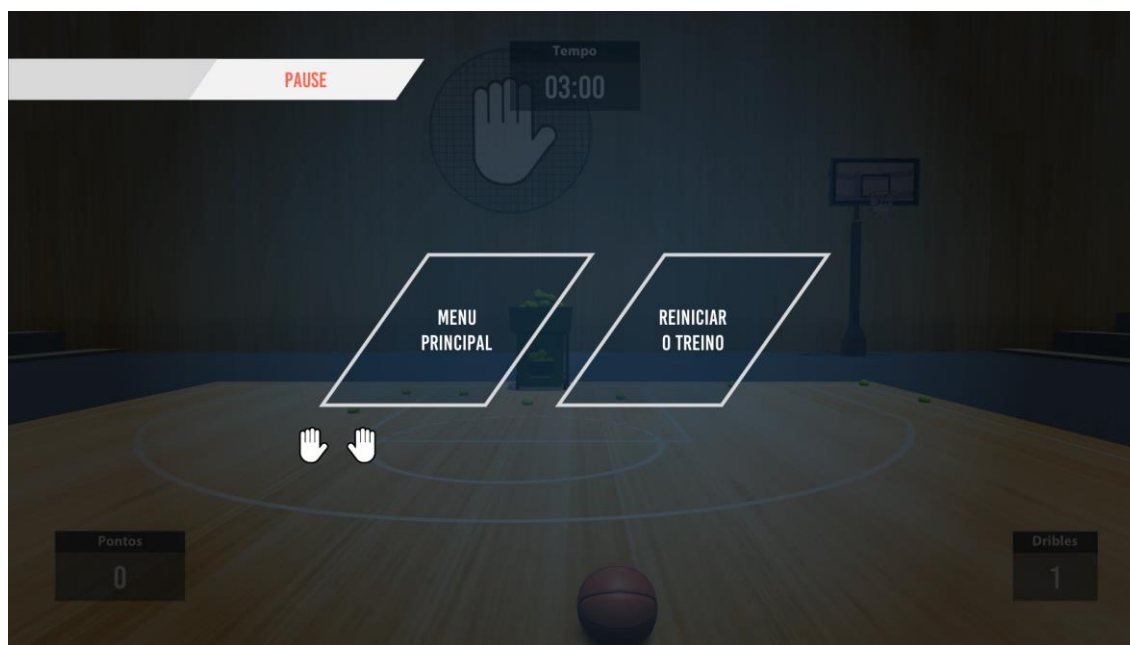


Figura 6 Treino de agilidade em pausa

O último treino implementado foi o treino de força. Durante este treino, são exibidos círculos acima da cabeça do usuário ou sobre a cintura do usuário. Após permanecer com a bola dentro do círculo por um certo tempo, o usuário recebe uma certa quantidade de pontos e o círculo aparece em outra posição. Caso o usuário retire a bola de dentro do círculo antes do tempo, o usuário perde pontos. O treino é finalizado ao final da contagem regressiva.

A Figura 7 demonstra uma captura de tela do treino de força no momento em que o atleta suspende a bola e consegue realizar uma pontuação. Onde se observa a bola de basquete posicionada ao centro superior da figura, indicando que o atleta está com uma bola de basquete em uma posição elevada. A bola está posicionada dentro de um círculo verde, que indica a posição onde o atleta dever manter a bola suspensa para pontuar. Por fim, ao lado do círculo, existe uma pontuação indicando que o atleta acabou de realizar um movimento de forma correta e obteve dez pontos por este movimento.

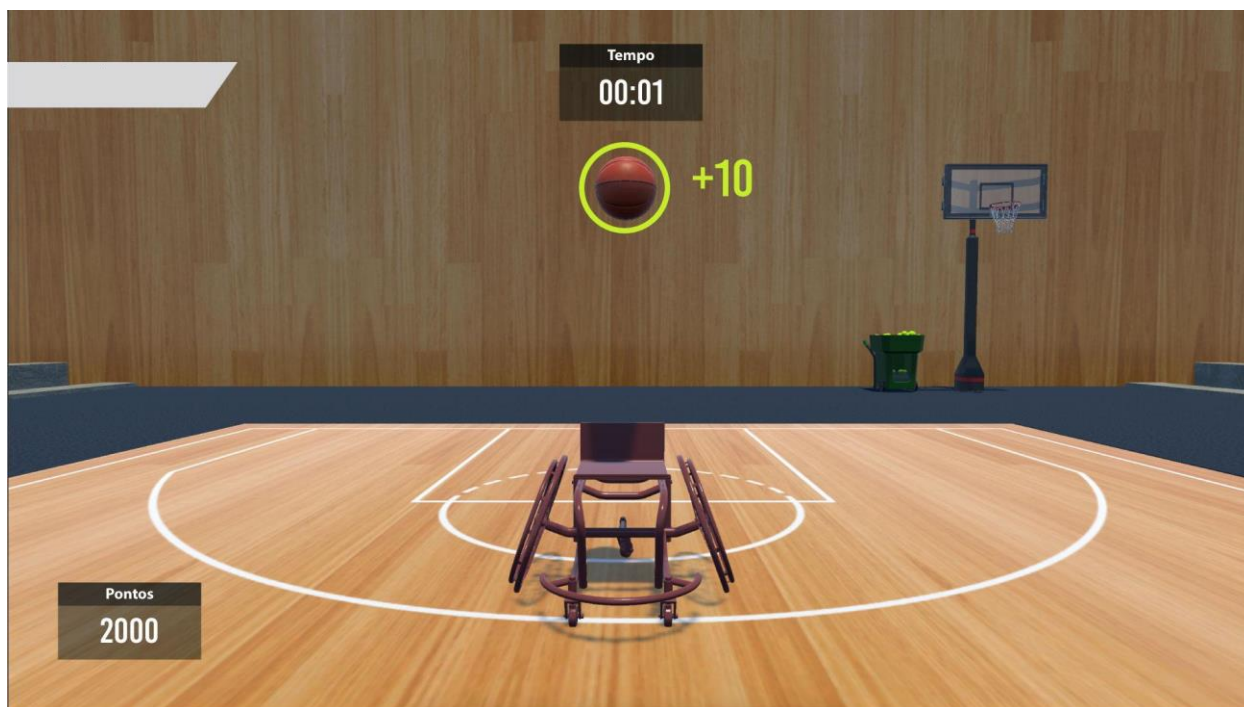


Figura 7 Treino de força

Por fim, foi implementada a funcionalidade de sincronização offline, possibilitando ao atleta realizar todas ações do sistema, com exceção de criar ou editar usuário, uma vez que já tenha efetuado login ao menos uma vez mesmo que não seja possível acessar o servidor que armazena os dados do sistema naquele momento.

5. Considerações finais

Este trabalho tinha como objetivo analisar diferentes game engines para o desenvolvimento de uma aplicação para treinamento de paratletas. Para isto, foram selecionadas e analisadas as principais game engines do mercado. Durante a análise das engines, foram observadas as principais características das engines, assim como as opiniões da comunidade em relação a alguns destes atributos. Após as análises, foi possível chegar à conclusão de que não existe hoje uma engine que seja melhor que os demais em termos genéricos. Porém a Unity apresentou-se como melhor opção para o desenvolvimento deste projeto, uma vez que se tratava de um projeto de curta duração e sem necessidade de renderização fotorrealista. Assim, após a seleção da game engine, foi desenvolvido um jogo para treinamento de paratletas, onde são apresentadas as potenciais vantagens da utilização de jogos para treinamentos, potenciais vantagens do jogo desenvolvido e suas principais características.

Este trabalho contribuiu para um melhor entendimento do que é uma game engine e do estado atual do mercado de game engines. Podendo servir como guia para outros trabalhos de desenvolvimento de jogos. Além disto, este trabalho contribui também para o entendimento do processo de desenvolvimento de um jogo.

5.1 Trabalhos futuros

Dentre os trabalhos futuros podemos citar:

- Aprofundar a pesquisa em relação a comparação das games engines, através de métodos mais sistemáticos e concretos para comparação de alguns atributos como qualidade visual e curva de aprendizado.
- Adicionar mais game engines para comparação, a fim de aumentar a cobertura do estudo.
- Realizar estudos de usabilidade sobre a ferramenta desenvolvida para avaliar se os resultados pretendidos através do desenvolvimento da aplicação foram atingidos.
- Realizar o desenvolvimento do mesmo jogo em diferentes engines para se analisar mais precisamente a diferença de dificuldade de desenvolvimento da mesma aplicação em outras engines, assim como a diferença de desempenho e qualidade visual nas demais engines.

- Realizar testes da aplicação com paratletas a fim de colher feedbacks e identificar pontos de melhoria do jogo, validar se o jogo de fato sana ou ao menos minimiza os problemas apresentados e se os benefícios previstos através da utilização do jogo de fato se refletem na prática.
- Testar outras estratégias para rastreamento da bola, uma vez que o Kinect possui frame rate máximo de trinta frames por segundo, driblar a bola muito rapidamente pode ocasionar em falhas na contabilização dos dribles, além de poder captar ruídos devido a informações do ambiente em alguns casos de uso, como por exemplo, possuir várias bolas no ambiente.
- Para cadastro através do Facebook é necessário sair da aplicação e interagir com o navegador para disponibilizar acesso da API do jogo ao Facebook, o que pode causar quebra de fluxo, talvez seja valido verificar formas de realizar essa interação sem que seja necessário o uso de um navegador como intermediador entre o Facebook e o jogo.

APÊNDICE A - Jogo desenvolvido

As figuras abaixo são capturas de tela que demonstram partes do jogo relacionadas as funcionalidades complementares as sessões treinamento.



Figura 8 Cadastro

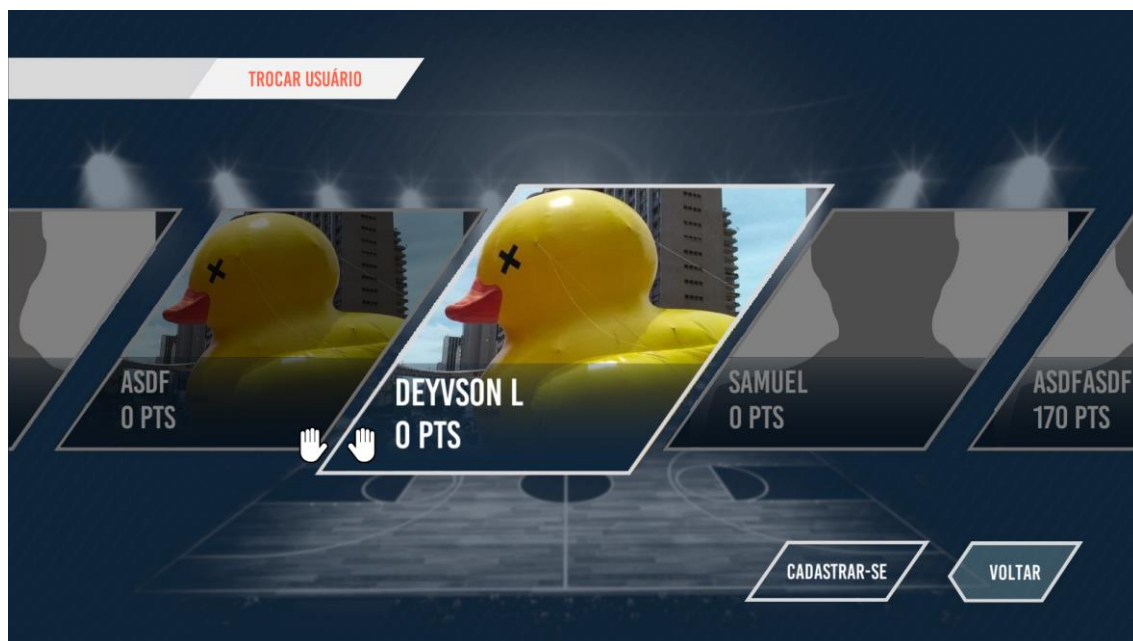


Figura 9 Trocar usuário



Figura 10 Menu principal



Figura 11 Ranking

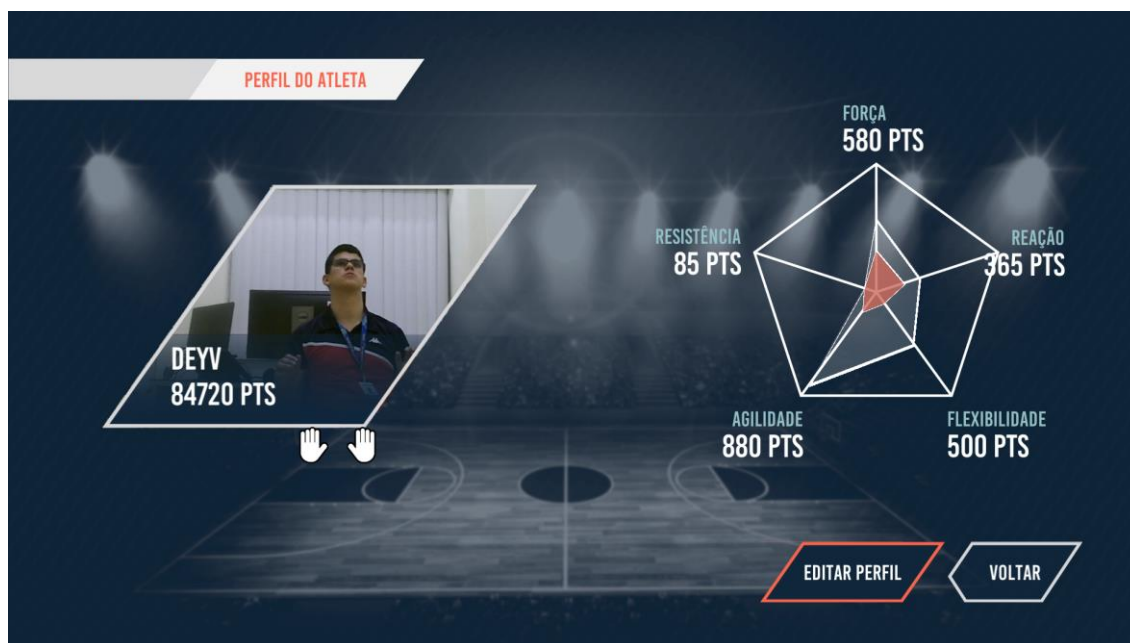


Figura 12 Perfil do atleta



Figura 13 Opções de tela

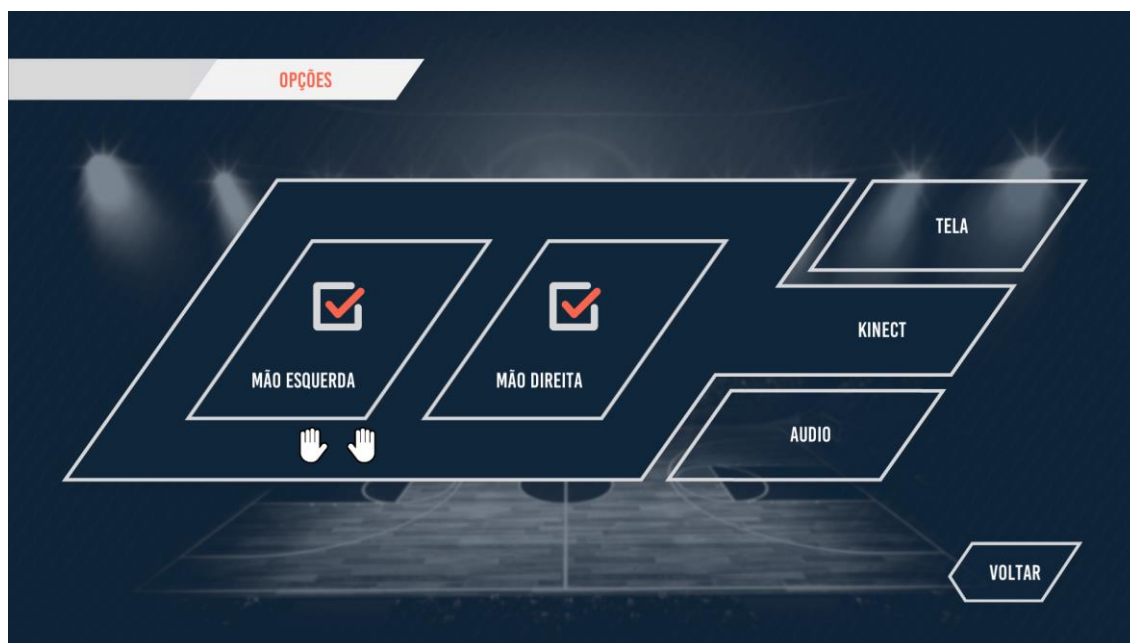


Figura 14 Opções do Kinect



Figura 15 Opções de áudio



Figura 16 Selecionar treino



Figura 17 Preparação Flexibilidade

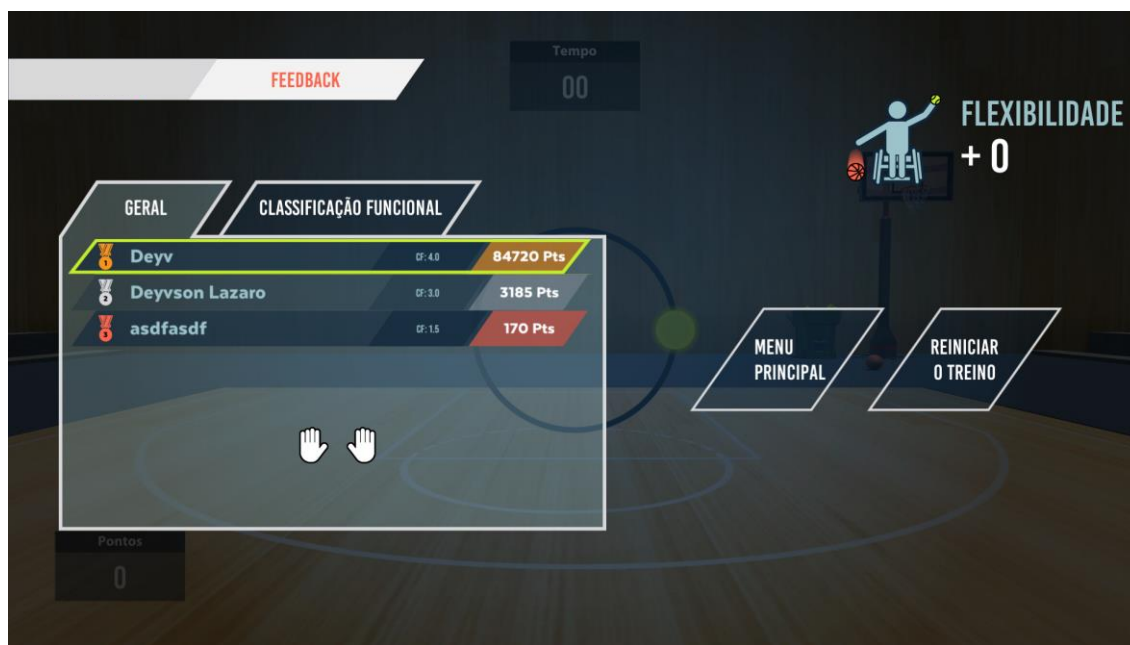


Figura 18 Treino de Flexibilidade Finalizado

Referências

- [1] CLUA, Esteban Walter Gonzalez; BITTENCOURT, João Ricardo. Desenvolvimento de jogos 3D: concepção, design e programação. In: **XXIV Jornadas de Atualização em Informática (JAI) Part of XXIV Congresso da Sociedade Brasileira de Computação**. 2005. p. 22-29.
- [2] CRAWFORD, Chris. The art of computer game design. 1984.
- [3] BERNARD, Suits. The Grasshopper: Games, Life, and Utopia. 1978.
- [4] SCOTT, Rogers. Level Up! The Guide to Great Videogame Design. 2010.
- [5] MACHADO, Liliane S.; MORAES, Ronei M.; NUNES, Fátima. Serious games para saúde e treinamento imersivo. **Abordagens práticas de realidade virtual e aumentada**, v. 1, p. 31-60, 2009.
- [6] MICHAEL, David R.; CHEN, Sandra L. **Serious games: Games that educate, train, and inform**. Muska & Lipman/Premier-Trade, 2005.
- [7] BATISTA, Thiago; MACHADO, Liliane; COSTA, Thaíse. Comparativo entre Ferramentas para Construção de um Ambiente Virtual 3D.
- [8] VALENTE, Luis; CONCI, Aura; FEIJÓ, Bruno. Real time game loop models for single-player computer games. In: Proceedings of the IV Brazilian Symposium on Computer Games and Digital Entertainment. 2005. p. 99.
- [9] GREGORY, Jason. **Game engine architecture**. AK Peters/CRC Press, 2014.
- [10] BOŠNIAK, Mateo; OREHOVAČKI, Tihomir. Measuring Quality of an Indie Game Developed Using Unity Framework. In: **41st International Convention on Information and Communication Technology, Electronics and Microelectronics**. 2018.
- [11] Game made with unity, <https://unity.com/madewith>
- [12] Awards - The Game Awards, <http://thegameawards.com/awards/>
- [13] Unity - Manual, <https://docs.unity3d.com/Manual/>
- [14] Products - Unity, <https://unity3d.com/pt/unity>
- [15] Unity Asset Store, <https://assetstore.unity.com/>

- [16] Unity - Learn, <https://unity3d.com/pt/learn>
- [17] Unity, <https://support.unity3d.com/hc/en-us>
- [18] Unity store, <https://store.unity.com/>
- [19] Awards, <https://www.unrealengine.com/en-US/awards>
- [20] Shortlist 2018: Develop Awards 2018,
<https://www.nbmevents.uk/developawards2018/shortlist2018>
- [21] Unreal Engine developers Take Center Stage at The Game Awards 2017,
<https://www.unrealengine.com/en-US/blog/unreal-engine-developers-take-center-stage-at-the-game-awards-2017>
- [22] What is Unreal Engine 4 - <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>
- [23] Unreal Engine 4 Documentation, <https://docs.unrealengine.com/en-us/>
- [24] FERREIRA, Pedro. Epic disponibiliza gratuitamente \$12 milhões de dólares em mais-valias de paragon, <http://pt.ign.com/paragon/57220/news/epic-disponibiliza-gratuitamente-12-milhoes-de-dolares-em-ma>
- [25] Marketplace - UE4 Marketplace, <https://www.unrealengine.com/marketplace/store>
- [26] Unreal Engine | EULA, <https://www.unrealengine.com/en-US/eula>
- [27] Introducing Unreal Dev Grants, <https://www.unrealengine.com/en-US/unrealdevgrants>
- [28] Unreal Studio | 1005 royalty-free license, <https://www.unrealengine.com/en-US/studio>
- [29] Unreal Engine | Education, <https://www.unrealengine.com/en-US/education>
- [30] The Climb: The Reveal!, <https://www.cryengine.com/news/the-climb-the-reveal>
- [31] CRYENGINE/LICENSE.md at release · CRYTEK/CRYENGINE,
<https://github.com/CRYTEK/CRYENGINE/blob/release/LICENSE.md>,
- [32] CRYENGINE V Manual - Documentation,
<http://docs.cryengine.com/display/CEMANUAL/CRYENGINE+V+Manual>
- [33] CRYENGINE, <https://www.cryengine.com/>
- [34] Rune-Status/Lumberyard, <https://github.com/Rune-Status/Lumberyard>
- [35] Gamasutra - Amazon launches new, free, high-quality game engine: Lumberyard,

https://gamasutra.com/view/news/265425/Amazon_launches_new_free_highquality_game_engine_Lumberyard.php

- [36] Amazon and Crytek Agree to Licensing, <https://www.gamespot.com/articles/amazon-and-crytek-agree-to-licensing-deal-worth-50/1100-6426438/>
- [37] Shortlist 2018: Develop Awards 2018, <https://www.nbmevents.uk/developawards2018/shortlist2018>
- [38] Amazon Lumberyard: recursos, <https://aws.amazon.com/pt/lumberyard/details/>
- [39] Amazon Lumberyard: Crie jogos mais ambiciosos, <https://aws.amazon.com/pt/lumberyard/>
- [40] A New Year, A New Engine | Amazon Game Tech Blog, <https://aws.amazon.com/blogs/gametech/1-12/>
- [41] LAVIERI, Dr. Edward. **Learning AWS Lumberyard Game Development**. Birmingham: Packt Publishing Ltd, 2016. 268 p.
- [42] DE MACEDO, Daniel Valente; RODRIGUES, Maria Andréia Formico; SERPA, Yvens Rebouças. Desenvolvimento de Aplicações Gráficas Interativas com a Unreal Engine 4. **Revista de Informática Teórica e Aplicada**, v. 22, n. 2, p. 181-202, 2015.
- [43] ZARRAD, Anis. Game Engine Solutions. In: **Simulation and Gaming**. InTech, 2018.
- [44] PETRIDIS, Panagiotis et al. Game engines selection framework for high-fidelity serious applications. **International Journal of Interactive Worlds**, v. 2012, p. 1, 2012.
- [45] ŠMÍD, Antonín. **Srovnání Unity a Unreal Engineu**. 2017. Trabalho de Conclusão de Curso. České vysoké učení technické v Praze. Vypočetní a informační centrum.
- [46] CryEngine vs Unreal vs Unity: Select the Best Game Engine, <https://medium.com/@thinkwik/cryengine-vs-unreal-vs-unity-select-the-best-game-engine-eaca64c60e3e>
- [47] WRIGHT, Chris. Unreal VS Unity VS Cry - MACH 2 Simulations Developer Talks - Quora, <https://mach2.quora.com/Unreal-VS-Unity-VS-Cry>
- [48] Kinect - Desenvolvimento de aplicativos do Windows, <https://developer.microsoft.com/pt-br/windows/kinect>
- [49] CARDOSO, Vinícius Denardin. O desenvolvimento da carreira esportiva de atletas

paraolímpicos no Brasil. 2016.

- [50] KARINE RAMOS, Daniela. Jogos cognitivos eletrônicos: contribuições à aprendizagem no contexto escolar. **Ciências & Cognição**, v. 18, n. 1, p. 19-32, 2013.
- [51] ARAÚJO, Danilo de Freitas. **Avaliação da relação entre qualidade de sono e uma intervenção com jogos para o desempenho de crianças e adolescentes**. 2012. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte.
- [52] DE ARROXELLAS, Raquel Daffre et al. Bocha adaptada: análise cinemática do arremesso e sua relação com a realidade virtual. **Revista Brasileira de Ciências do Esporte**, v. 39, n. 2, p. 160-167, 2017.
- [53] shimat/opencvsharp: .NET Framework wrapper for OpenCV,
<https://github.com/shimat/opencvsharp>
- [54] Basquete em Cadeira de Rodas - Modalidades Visualização - CPB,
http://www.cpb.org.br/modalidades-visualizacao/-/asset_publisher/4O6JOgZOhDhG/content/id/22670
- [55] PHP SDK - Web SDKs, <https://developers.facebook.com/docs/reference/php/>