



GIOVANNI VIOL ASSIS

**DESENVOLVIMENTO DE TERRENOS DE
UM JOGO DE RPG COM TEMA MEDIEVAL
PARA A PLATAFORMA ANDROID
UTILIZANDO A ENGINE UNITY 3D**



LAVRAS - MG

2014

GIOVANNI VIOL ASSIS

**DESENVOLVIMENTO DE TERRENOS DE UM JOGO DE RPG
COM TEMA MEDIEVAL PARA A PLATAFORMA ANDROID
UTILIZANDO A ENGINE UNITY 3D**

Trabalho de Conclusão de Curso de Graduação
apresentado ao Colegiado do Curso de
Bacharelado em Ciência da Computação, para
obtenção do título de Bacharel.

Orientador

Prof. Dr. Joaquim Quinteiro Uchôa

LAVRAS - MG

2014

GIOVANNI VIOL ASSIS

**DESENVOLVIMENTO DE TERRENOS DE UM
JOGO DE RPG COM TEMA MEDIEVAL PARA A
PLATAFORMA ANDROID UTILIZANDO A
ENGINE UNITY 3D**

Trabalho de Conclusão de Curso de
Graduação apresentado ao Colegiado do
Curso de Bacharelado em Ciência da
Computação, para obtenção do título de
Bacharel.

APROVADA em 18 de novembro de 2014.

Dr. André Pimenta Freire

Dr. Neumar Costa Malheiros



Dr. Joaquim Quinteiro Uchôa (Orientador)

**LAVRAS-MG
Novembro/2014**

RESUMO

A adoção de *engines* facilita o desenvolvimento de jogos por oferecer um conjunto de ferramentas para a sua implementação. Neste trabalho, é proposto o desenvolvimento de terrenos de um jogo de RPG com o tema medieval para a plataforma Android, utilizando a *engine* Unity 3D. O trabalho discute acerca das etapas para a criação dos terrenos, a saber: ilha, deserto, floresta e montanhas. A realização do projeto de criação dos terrenos foi baseada em leituras na área, videoaulas, fóruns e um minicurso sobre essa *engine*. Além dos terrenos, a aplicação possui várias telas que permitem a navegação do usuário por meio das cenas de créditos e dos quatros terrenos desenvolvidos, sendo a exploração desses cenários realizada em primeira pessoa. Após o término do trabalho, foram feitos testes com usuários para verificar a compatibilidade da aplicação entre vários dispositivos Android com versões e arquiteturas de processadores distintas. Após os resultados dos testes, concluiu-se que é possível desenvolver jogos utilizando-se a Unity 3D para a criação de gráficos realistas para dispositivos Android e que há uma grande compatibilidade da aplicação desenvolvida com várias versões dessa plataforma.

Palavras-chave: desenvolvimento de jogo, Android, Unity 3D, criação de terrenos.

LISTA DE TABELAS

TABELA 1: RESPOSTA DA QUESTÃO 1 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	42
TABELA 2: RESPOSTA DA QUESTÃO 2 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	43
TABELA 3: RESPOSTA DA QUESTÃO 3 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	45
TABELA 4: ENUNCIADO DA QUESTÃO 7 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	76
TABELA 5: ENUNCIADO DA QUESTÃO 8 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	76
TABELA 6: ENUNCIADO DA QUESTÃO 9 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	76

LISTA DE FIGURAS

FIGURA 1: CRESCIMENTO DO SISTEMA OPERACIONAL ANDROID EM RELAÇÃO AOS SEUS CONCORRENTES.	12
FIGURA 2: TENDÊNCIA DA PLATAFORMA ANDROID SUPERAR O MERCADO DE COMPUTADORES PESSOAIS.	13
FIGURA 3: ARQUITETURA DO SISTEMA OPERACIONAL ANDROID	14
FIGURA 4: CATEGORIAS MAIS UTILIZADAS DO SISTEMA OPERACIONAL ANDROID E APP STORE	16
FIGURA 5: EXEMPLO DE DESENVOLVIMENTO DE TERRENOS UTILIZANDO A <i>ENGINE</i> UNITY	22
FIGURA 6: INTERFACE DA <i>ENGINE</i> UNITY 3D	23
FIGURA 7: GRAFO DE NAVEGAÇÃO DE CENAS DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	27
FIGURA 8: <i>PRINT SCREEN</i> DA CENA MENU PRINCIPAL DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	34
FIGURA 9: <i>PRINT SCREEN</i> DA CENA CRÉDITOS DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	37
FIGURA 10: <i>PRINT SCREEN</i> DA SELEÇÃO DE CENAS 1 DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	37
FIGURA 11: <i>PRINT SCREEN</i> DA SELEÇÃO DE CENAS 2 DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	38
FIGURA 12: <i>PRINT SCREEN</i> DA CENA CARREGANDO DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	38
FIGURA 13: <i>PRINT SCREEN</i> DA CENA ILHA DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	39
FIGURA 14: <i>PRINT SCREEN</i> DA CENA DESERTO DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	39
FIGURA 15: <i>PRINT SCREEN</i> DA CENA FLORESTA DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	40
FIGURA 16: <i>PRINT SCREEN</i> DA CENA MONTANHAS DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	40
FIGURA 17: <i>PRINT SCREEN</i> DA INTERFACE AO USUÁRIO DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	41
FIGURA 18: <i>PRINT SCREEN</i> DA INTERFACE PAUSE DA APLICAÇÃO DESENVOLVIDA NESTE PROJETO	41
FIGURA 19: GRÁFICO DO RESULTADO DA QUESTÃO 1 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	42
FIGURA 20: GRÁFICO DO RESULTADO DA QUESTÃO 2 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	44
FIGURA 21: GRÁFICO DO RESULTADO DA QUESTÃO 3 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	46
FIGURA 22: <i>PRINT SCREEN</i> DO RESULTADO DA QUESTÃO 4 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	47
FIGURA 23: <i>PRINT SCREEN</i> DO RESULTADO DA QUESTÃO 5 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	48
FIGURA 24: <i>PRINT SCREEN</i> DO RESULTADO DA QUESTÃO 6 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	49
FIGURA 25: <i>PRINT SCREEN</i> DO RESULTADO DA QUESTÃO 7 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	50
FIGURA 26: <i>PRINT SCREEN</i> DO RESULTADO DA QUESTÃO 8 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	51
FIGURA 27: <i>PRINT SCREEN</i> DO RESULTADO DA QUESTÃO 9 DO FORMULÁRIO DOS TESTES COM USUÁRIOS	52

FIGURA 28: *PRINT SCREEN* DO RESULTADO DA QUESTÃO 10 DO FORMULÁRIO DOS TESTES COM USUÁRIOS
.....54

SUMÁRIO

1	INTRODUÇÃO	9
1.1	CONTEXTUALIZAÇÃO.....	9
1.2	OBJETIVOS.....	10
2	REFERENCIAL TEÓRICO	11
2.1	ANDROID.....	11
2.2	JOGOS ELETRÔNICOS.....	15
2.3	DESENVOLVIMENTO DE JOGOS	17
2.4	UNITY 3D.....	19
3	METODOLOGIA	24
3.1	MATERIAIS E MÉTODOS	24
3.2	DESENVOLVIMENTO DO PROJETO.....	24
3.2.1	AQUISIÇÃO DOS CONHECIMENTOS NECESSÁRIOS.....	24
3.2.2	IMPORTAÇÃO DOS PACOTES E CRIAÇÃO DAS CENAS	26
3.2.3	CRIAÇÃO DA CENA MENU PRINCIPAL	27
3.2.4	CRIAÇÃO DA CENA DE CRÉDITOS.....	28
3.2.5	CRIAÇÃO DA SELEÇÃO DE CENAS.....	28
3.2.6	CRIAÇÃO DO TERRENO ILHA.....	30
3.2.7	CRIAÇÃO DO TERRENO DESERTO.....	32
3.2.8	CRIAÇÃO DO TERRENO FLORESTA	33
3.2.9	CRIAÇÃO DO TERRENO MONTANHAS.....	33
3.2.10	CRIAÇÃO DA INTERFACE DOS TERRENOS.....	34
3.2.11	A COMPATIBILIDADE DESTA APLICAÇÃO	34
3.2.12	OS TESTES COM OS USUÁRIOS	35
4	RESULTADOS E DISCUSSÃO	36

4.1	RESULTADOS DAS CRIAÇÕES DE CENAS.....	36
4.2	RESULTADOS DAS CRIAÇÕES DOS TERRENOS.....	38
4.3	RESULTADOS DA CRIAÇÃO DA INTERFACE.....	40
4.4	RESULTADOS DOS TESTES COM OS USUÁRIOS.....	42
5	CONCLUSÃO.....	55
6	REFERÊNCIAS BIBLIOGRÁFICAS.....	57
7	APÊNDICE A.....	58
7.1	MENU PRINCIPAL.....	58
7.2	CRÉDITOS.....	59
7.3	SELEÇÃO DE CENAS.....	60
7.4	PAUSE.....	64
7.5	JOYSTICK.....	67
8	APÊNDICE B.....	75

1 INTRODUÇÃO

1.1 Contextualização

Este trabalho faz parte de um projeto maior que visa o desenvolvimento de um jogo digital para a plataforma Android. O gênero do jogo que será desenvolvido é RPG (*Role-Playing Game*), que em português significa jogo de interpretação de personagens. Nesse gênero, são criadas regras predefinidas, nas quais os jogadores podem improvisar livremente através do personagem escolhido, criando assim uma narrativa, cujas escolhas de jogo influenciam diretamente na história. A temática do jogo desenvolvido é inspirada na era medieval, um tema bem explorado em filmes, séries e jogos na atualidade. É uma era com poucas construções humanas e com bastante vegetação.

O mercado de jogos para plataformas móveis vem crescendo consideravelmente nos últimos anos com a popularização dos dispositivos móveis. Dentre os sistemas operacionais utilizados para rodar jogos, o Android tem se destacado. Em consequência disso, essa foi a plataforma utilizada para o desenvolvimento deste trabalho.

Segundo o site da Canaltech¹, o Android é o sistema operacional presente na maioria dos dispositivos móveis do planeta. Jogar no celular passou a ser comum, a partir da era dos *smartphones* e da difusão do acesso à internet. Com isso, empresas de desenvolvimento de jogos estão aproveitando este momento para lançar seus produtos no mercado. É perceptível o crescimento constante na área de jogos e, por esse motivo, pesquisas sobre o assunto devem ser realizadas para que haja um maior desenvolvimento neste campo.

¹ SITE CANALTECH: informações a respeito do Android atualmente. Disponível em: <<http://canaltech.com.br/noticia/mercado/Aparelhos-com-Android-dominam-70-das-vendas-de-dispositivos-moveis-diz-pesquis/>> Publicada em 26/02/2014. Acesso em 31/05/2014.

Este trabalho teve como meta o desenvolvimento de terrenos virtuais para um jogo por meio de dispositivos móveis.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver terrenos para um jogo de RPG com o tema medieval utilizando a engine Unity 3D para a plataforma Android. Além do objetivo geral, é possível destacar três específicos:

1. Criação do projeto, que inclui a criação de terrenos, de objetos dentro de cada terreno, de cenas, uma para cada terreno e de animações para os objetos presentes em seus respectivos terrenos;
2. Implementação do código de terrenos para a realização das animações e adaptações;
3. Realização de testes e ajustes das animações, colisões, implementações de cada objeto, terreno e cena.

Após a conclusão deste trabalho, espera-se criar diversos terrenos e cenas para o jogo de RPG, de modo que sejam bem diversificados e detalhados. Ao final do projeto, testes com usuários foram realizados de forma que cada voluntário testará a aplicação em seus dispositivos Android com a finalidade de relatar a compatibilidade nos seus aparelhos e opinar sobre a interface e desempenho da aplicação.

2 REFERENCIAL TEÓRICO

2.1 Android

O Android² é um sistema operacional de código aberto baseado no *Kernel* do Linux 2.6. Isso possibilita uma maior facilidade para que empresas desenvolvam seus *smartphones* e *tablets*, utilizando esse sistema operacional. Trata-se de um sistema que requer menos memória e processamento comparado a um sistema operacional para desktops da mesma geração, pois foi desenvolvido especialmente para dispositivos móveis.

Seu mantenedor é o Google Inc³. Cada versão lançada pela empresa possui um nome de sobremesa. A versão mais atual do Android para consumidores atualmente é o Kit Kat⁴, mas em junho de 2014 foi anunciado, no evento Google I/O⁵, uma nova versão denominada Lollipop⁶ que já está disponível para desenvolvedores e em novembro deste ano já estará disponível a atualização para os consumidores.

² SITE ANDROID™ <<http://www.android.com/>> O Android é um sistema operacional, de código livre, para dispositivos móveis.

³ <https://www.google.com/intl/pt-BR/about/company/>

⁴ <http://www.android.com/versions/kit-kat-4-4/>

⁵ <https://www.google.com/events/io>

⁶ <http://www.android.com/versions/lollipop-5-0/>

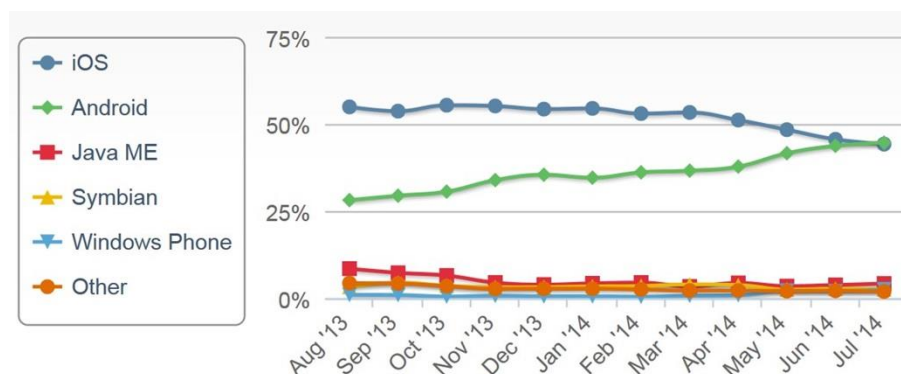


Figura 1: Crescimento do sistema operacional Android em relação aos seus concorrentes.

Fonte: Tecmundo⁷.

O blog Tecmundo⁸, que é um dos maiores blogs de tecnologias no Brasil atualmente, divulgou uma matéria afirmando que o Android é a plataforma mais utilizada no mundo a partir de agosto de 2014. A Figura 1 apresenta o gráfico comparativo com a concorrência.

Já o site da Canaltech⁹, em um artigo de fevereiro deste ano, aparelhos com Android dominam 70% do mercado. Várias empresas lançam seus dispositivos móveis com esse sistema. Uma delas é a Samsung, uma das líderes de venda deste mercado atualmente, com mais de 30% do total de vendas em 12 regiões do mundo. Outras grandes empresas são LG, Motorola e Sony. Os sistemas operacionais principais que concorrem com o Android atualmente são o iOS da empresa Apple¹⁰, com 22% do mercado, e o Windows Phone da Microsoft, com apenas 4%, mas em grande expansão após a compra

⁷ <http://www.tecmundo.com.br/android/60002-android-passa-ios-sistema-operacional-movel-usado-mundo.htm>

⁸ <http://www.tecmundo.com.br/>

⁹ SITE CANALTECH: informações a respeito do Android atualmente. Disponível em: <<http://canaltech.com.br/noticia/mercado/Aparelhos-com-Android-dominam-70-das-vendas-de-dispositivos-moveis-diz-pesquis/>> Publicada em 26/02/2014. Acesso em 31/05/2014.

¹⁰ <https://www.apple.com/br/>

da Nokia. Ainda existem outros concorrentes menos conhecidos, como o da empresa BlackBerry, que está em declínio.

Na loja de aplicações do Google, denominada Play Store¹¹, encontram-se quatro diferentes seções pelas quais um usuário pode navegar e baixar itens para seu dispositivo móvel: aplicativos, jogos, filmes e livros. Dentre essas seções, a de jogos é uma das mais acessadas e se constitui como o foco principal de investigação deste trabalho.

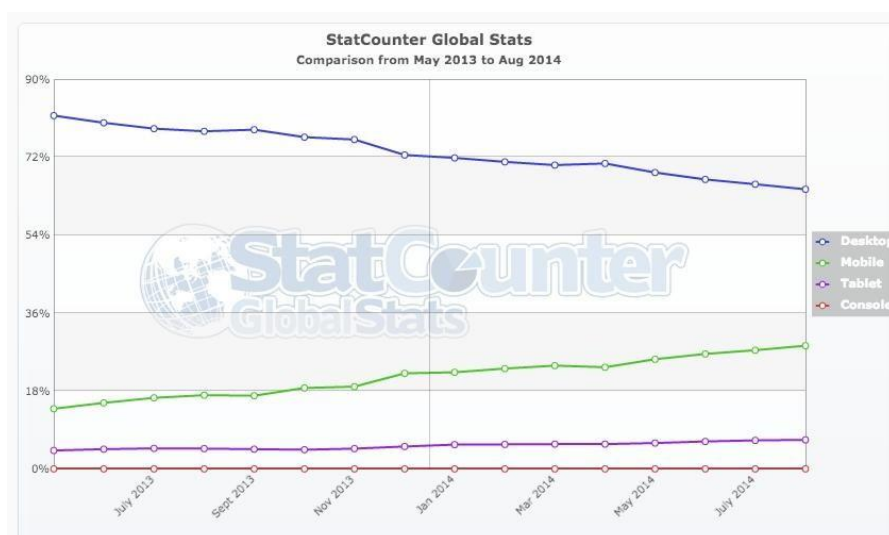


Figura 2: Tendência da plataforma Android superar o mercado de computadores pessoais.

Fonte: Acontecendoaqui¹².

O blog Acontecendoaqui¹³, divulgou uma matéria comparando o mercado de computadores pessoais, *smartphones*, *tablets* e consoles. A partir do gráfico da Figura 2, é possível concluir que há uma tendência das vendas dos celulares inteligentes superarem as vendas de computadores pessoais com o passar dos anos.

¹¹ <https://play.google.com/store>

¹² <http://www.acontecendoaqui.com.br/hora-e-vez-dos-dispositivos-moveis/>

¹³ <http://www.acontecendoaqui.com.br/>

A arquitetura do Android segundo Passos (2009, p.5), é dividida em cinco camadas: *Linux Kernel*, *Libraries*, *Android Runtime*, *Application Framework* e *Applications*. A Figura 3 representa essas camadas. É fundamental conhecer bem a arquitetura do Android e fazer uso desses conhecimentos ao trabalhar com programação.

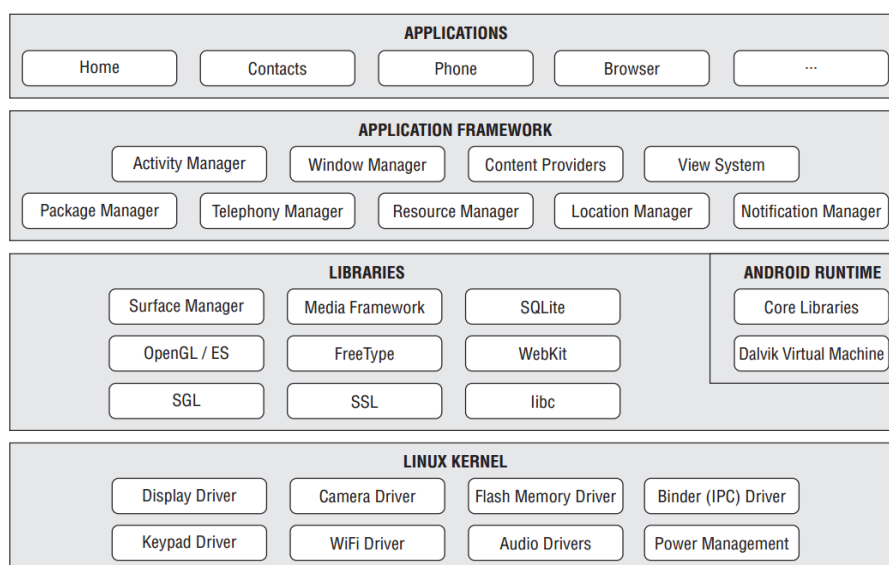


Figura 3: Arquitetura do sistema operacional Android.
 Fonte: (LEE, 2011).

2.2 Jogos Eletrônicos

O termo jogo, de acordo com Gregory (2009, p.8), engloba atividades diversas, como jogos de tabuleiro, de cartas, de estratégias de guerra, entre outros. No contexto eletrônico, a palavra denota um conjunto de imagens bi ou tridimensionais inseridas em mundos virtuais que agregam figuras de humanos, animais ou veículos como o personagem principal que comanda as ações ocorridas em fases ou em uma estória. Com o avanço da internet e dos dispositivos móveis, muitas pessoas jogam diariamente em qualquer hora e local e compartilham seus resultados na *Web* a fim de os comparar com os de outros jogadores.

A respeito do suporte a jogos na plataforma Android, muitos recursos estão sendo melhorados e aperfeiçoados. É possível conectar *joysticks* nos dispositivos e transmitir a imagem em uma televisão, o que aumenta bastante a diversão no jogo e o interesse geral pelos jogos eletrônicos. Projetos como o Oculus Rift¹⁴, que são óculos de realidade virtual, estão em andamento e quando chegarem no mercado provavelmente se iniciará uma nova era de jogos.

¹⁴ <http://www.oculusvr.com>

Percent of Apps on Platform by Normalized Category

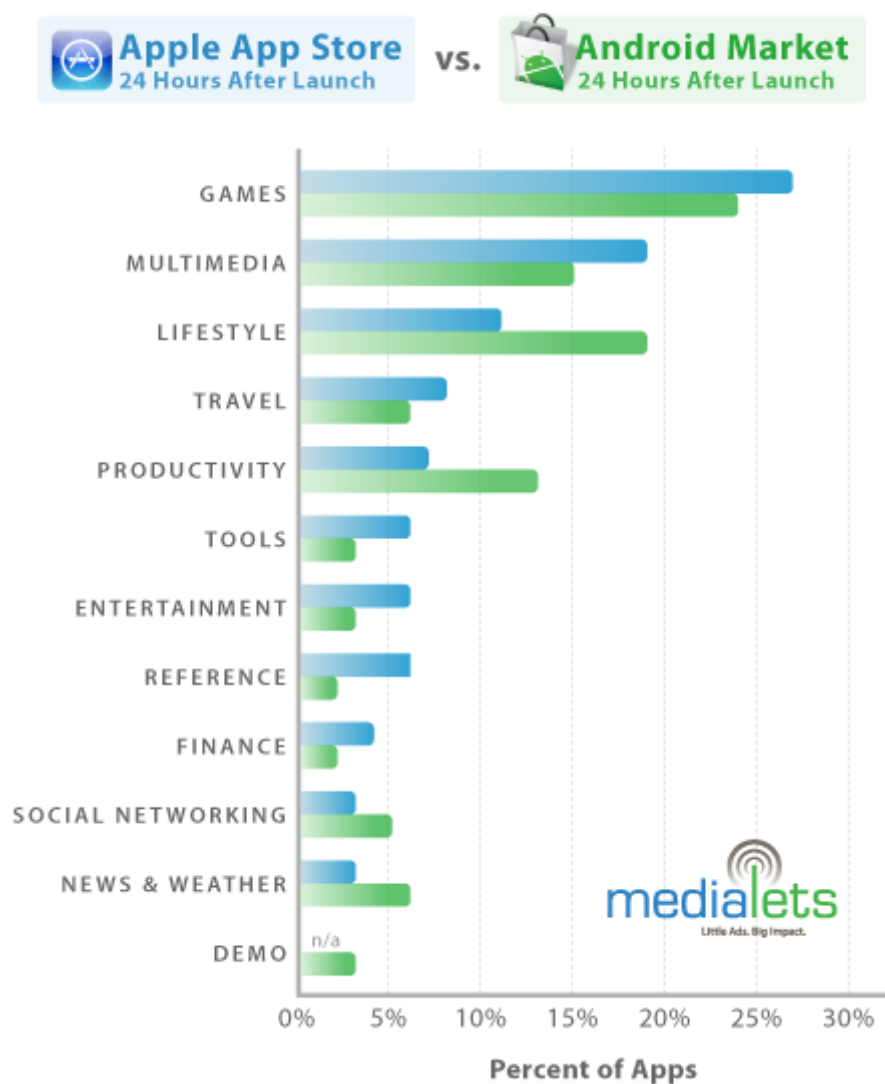


Figura 4: Categorias mais utilizadas do sistema operacional Android e App Store¹⁵.
Fonte: Techcrunch¹⁶.

O blog Techcrunch¹⁷, divulgou uma matéria na qual compara as categorias que se destacam mais nas lojas virtuais Play Store e App Store da

¹⁵ <http://store.apple.com/br>

¹⁶ <http://techcrunch.com/2008/10/23/android-market-takes-on-apples-app-store-games-still-rule-but-should-they/>

¹⁷ <http://techcrunch.com/>

empresa Apple, a maior concorrente do Android. A categoria jogos é a que possui a maior influência. A Figura 4 apresenta o gráfico comparativo com a concorrência.

A indústria de jogos, como aponta Mastrocola (2012, p.44), é o setor que mais cresce na área do entretenimento. Contudo, o autor argumenta que o mercado de desenvolvimento de jogos eletrônicos ainda é pequeno no Brasil se comparado com a de países de primeiro mundo como os Estados Unidos, cujo faturamento atual é por volta de U\$70 bilhões. Para mudar esse quadro, há associações no Brasil que apoiam a criação de jogos e seus desenvolvedores, como a Associação Brasileira de Desenvolvedores de Jogos Eletrônicos (ABRAGAMES).

Criada em 2004 por um grupo de empresas de desenvolvimento de jogos, os principais objetivos da ABRAGAMES¹⁸, são organizar, coordenar e promover a indústria brasileira de jogos digitais através de eventos e parcerias entre interessados na área. Na seção a seguir, é explorada com mais profundidade essa área de atuação, enfatizando o desenvolvimento de jogos pelas *engines*.

2.3 Desenvolvimento de jogos

O desenvolvimento de jogos no século XXI tem sido um trabalho menos árduo que há décadas atrás, visto que antes não haviam bibliotecas de física e interfaces gráficas, por exemplo, à disposição dos programadores. Uma *engine* é um programa motor de um jogo que possui um conjunto de bibliotecas que simplifica e abstrai o trabalho do programador no desenvolvimento de um jogo. Atualmente existe várias *engines* no mercado como a CryEngine¹⁹ da empresa Crytek²⁰, Unity 3D produzida pela Unity

¹⁸ <http://www.abragames.org/>

¹⁹ <http://cryengine.com>

²⁰ <http://www.crytek.com>

Technologies²¹, Fox Engine, desenvolvida pela Kojima Productions²² da empresa Konami e Unreal²³ da empresa Epic Games²⁴. As *engines* facilitam a criação de jogos nas áreas:

- Gráfica;
- Sonora;
- Física;
- Programação;
- Animação;
- Inteligência Artificial;
- *Networking*;
- Gerenciamento de memória;
- Gerenciamento de arquivos;
- Gerenciamento de linha de execução;
- Grafos de cenas e entidades.

Com o uso de uma *engine*, o programador é capaz de adaptar objetos 3D (personagens, cenários), imagens, sons, *scripts* e outros aspectos que são essenciais para a construção de um jogo, de uma maneira simplificada e rápida.

Ao começar a desenvolver um jogo, é necessário predefinir qual o seu tipo: realista ou animação. Define-se também seu gênero: RPG (*Role-Playing Game*), corrida, esportes, quebra-cabeças, aventura, FPS (*First-Person Shooter*) ou estratégia. Especificar se o jogo será 3D ou 2D também é um ponto crucial, pois influencia bastante na jogabilidade e por sua vez na diversão do jogo.

O valor de uma *engine* varia muito e algumas delas nem são vendidas ou disponibilizadas, como é o caso da Fox Engine. Uma versão anterior à atual da CryEngine e da Unreal foram disponibilizadas para o público, mas sem permissão de lançamento comercial. Já a Unity tem uma versão gratuita que possibilita a usuários desenvolver seus jogos e vendê-los para algumas

²¹ <https://unity3d.com/pt/company>

²² http://www.konami.jp/kojima_pro/english/

²³ <https://www.unrealengine.com>

²⁴ <http://epicgames.com/>

plataformas gratuitamente. Com isso, a *engine* adotada para o desenvolvimento deste trabalho é a Unity, apresentada na próxima seção.

2.4 Unity 3D

A *engine* selecionada para o desenvolvimento deste trabalho é Unity, uma *engine* de desenvolvimento de jogos, também conhecida como Unity 3D²⁵. Possui conteúdos interativos em 2D ou 3D e inúmeras ferramentas, sendo algumas delas para transladar, rotacionar e redimensionar objetos e fazer animações com tecnologia *Mecanim*, que garante maior eficiência, estabilidade e renderização DirectX 11 e traz uma sensação de realidade nos jogos.

O papel do Unity neste trabalho é adequar o tamanho, rotação, transladação, comportamento e animação do objeto no contexto do jogo. É possível utilizar alguns objetos que vem juntamente com o próprio Unity, como cubo, esfera, cilindro, plano, dentre outros. Um componente presente no Unity bastante utilizado em jogos são as partículas, sendo possível configurá-las de forma que fiquem velozes, devagar, controlar o tempo de duração de cada uma, o tamanho, comportamento e se é do tipo 2D ou 3D. Segundo Rollings e Morris (2003, p.640), tendências futuras apontam que o motor do jogo realizará boa parte de seu desenvolvimento, pois o processo de montagem de um jogo será automatizado, permitindo ao profissional se preocupar mais com a criação que o desenvolvimento em si.

As primeiras versões do Unity tinham suporte apenas para Mac OS X. Já as atuais possuem suporte para Playstation 3 e 4, PS Vita, Xbox 360 e One, Windows, Windows Phone 8, Windows Store Apps, Linux Standalone, Android, iOS, BlackBerry e Web Player. O Unity 3D possui a versão paga (Unity Pro) que custa US\$1500,00 e pode ser adquirida no site oficial na seção

²⁵ SITE UNITY 3D: informações sobre a *engine* Unity 3D. Disponível em: <<http://unity3d.com/pt/unity>> Acesso em 31/05/2014.

de compras²⁶. Dessa forma, o usuário tem à sua disposição todos os recursos da versão adquirida oferecidos por essa *engine* por tempo ilimitado. É possível também optar pelo pagamento de US\$75,00 para ter direito à versão Pro durante 30 dias. Por fim, há a versão totalmente gratuita que pode ser obtida no site oficial da Unity²⁷ e a possibilidade de testar a versão completa gratuitamente durante 30 dias.

Há diferenças entre as versões paga e gratuita da *engine*, em relação à exportação do projeto desenvolvido e à quantidade de funcionalidades de cada uma, por exemplo. Há uma lista comparativa extensa dos recursos disponíveis em ambas as versões no site da *engine*. Apesar dessas diferenças, a versão gratuita atende à necessidade de muitos desenvolvedores *indies* (independentes) que estão começando a desenvolver aplicações e até mesmo quem já se profissionalizou nesta área.

Na Asset Store²⁸, a loja virtual da *engine*, há vários pacotes que facilitam o desenvolvimento de projetos no Unity, que são divididos nas seguintes categorias: modelos 3D, animação, aplicações, áudio, projetos completos, editor de extensão, sistema de partículas, *scripting*, serviços, sombras, texturas e materiais, sendo que há os pacotes gratuitos e os pagos. Além dos pacotes disponibilizados na loja virtual, há também alguns que são nativos da *engine*. São eles: *Character Controller*, *Glass Refraction*, *Image Effects*, *Light Cookies*, *Light Flares*, *Particles*, *Physic Materials*, *Projectors*, *Scripts*, *Skyboxes*, *Standard Assets (Mobile)*, *Terrain Assets*, *Tessellatin Shaders (DX11)*, *Toon Shading*, *Tree Creator*, *Water (Basic e Pro)*. Os pacotes *Glass Refraction* e *Image Effects* são disponíveis somente na versão Pro.

Existe também no Unity 3D um objeto denominado *terreno* que pode ser configurável. É possível definir o seu tamanho máximo em pixels ou em

²⁶ <https://store.unity3d.com/>

²⁷ SITE UNITY 3D: informações sobre download da engine Unity 3D. Disponível em: <<http://unity3d.com/pt/unity/download>> Acesso em 31/05/2014.

²⁸ <https://www.assetstore.unity3d.com/en/>

proporção. Dentro da manipulação do terreno há várias ferramentas para amassar, esculpir, colorir, realçar, etc. Segundo informações encontradas no site sobre a qualidade de terrenos em Unity²⁹, o mecanismo de terrenos dessa *engine* inclui uma ferramenta de criação de árvores, ideal para criar florestas ou bosques. Com essa ferramenta, pode-se usar uma mistura de geração de procedimentos e técnicas manuais em diferentes partes da árvore. O mecanismo de terrenos do Unity vem integrado com o *Beast lightmapper*, que permite obter a combinação perfeita entre terrenos e outros elementos gráficos.

O terreno é composto por vários elementos visuais e sonoros pelo qual os personagens transitam durante todo o jogo. No Unity, existem ferramentas específicas para criação de terrenos, que possibilitam esculpir, subir, descer, entalhar, pintar, utilizar efeitos de luz, vento, extremamente naturais, folhagens exuberantes, camadas, inserção de árvores, entre outros. O desenvolvimento de cenários é um aspecto bastante relevante na criação de jogos, pois é nele que se passa todo o jogo.

Os tipos de efeitos visuais e sonoros frequentemente utilizados são: vento interagindo com o cenário, movimento de água, animações de árvores, som ambiente com cigarras, com pássaros, dentre outros animais. A respeito da criação de cenas³⁰, há uma discussão sobre o botão de adicionar componentes e economizar tempo com o uso de objetos pré-fabricados, sendo este um modo para agilizar o uso repetitivo de objetos complexos. É ressaltado ainda o uso de *snaps* para a criação de esferas, caixas, cápsulas e malhas no ambiente e a facilidade de se posicionar, ajustar e girar esses objetos como for desejado.

É possível “criar” objetos no Unity utilizando outros já existentes como cubos, esferas, cilindros, planos entre outros. No entanto, segundo Passos et al. (2009, p.4), o desenvolvimento de jogos 3D é uma atividade

²⁹ SITE UNITY 3D: informações sobre a terrenos na Unity 3D. Disponível em:<<http://unity3d.com/pt/unity/quality/terrains>> Acesso em 31/05/2014.

³⁰ <http://unity3d.com/pt/unity/workflow/scene-building>

desafiadora, pois diversas habilidades de diferentes áreas do conhecimento são requeridas, além de muita criatividade para atrair o público.

Na Figura 5, são apresentados alguns cenários disponíveis no site do Unity.



Figura 5: Exemplo de desenvolvimento de terrenos utilizando a *engine* Unity.
Fonte: Unity Terrains³¹.

A Figura 6 mostra a interface desta *engine*.

³¹ <http://unity3d.com/pt/unity/quality/terrains>

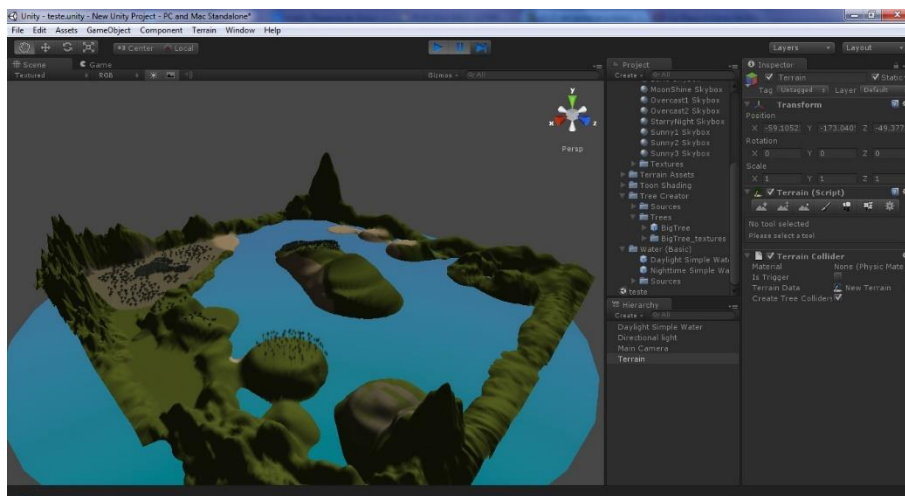


Figura 6: Interface da engine Unity 3D.

3 METODOLOGIA

Como foi citado na introdução, este trabalho faz parte de um projeto maior com o objetivo de desenvolver um jogo de RPG com o tema medieval para a plataforma Android. Esse trabalho consiste somente no desenvolvimento de terrenos, portanto não haverá uma mecânica do jogo, personagens, regras, e nem história, somente algumas cenas com seus respectivos terrenos.

3.1 Materiais e Métodos

Para a realização deste projeto, foi utilizado a ferramenta Unity 3D versão 4.5.4 gratuita disponível no site oficial.

Os equipamentos utilizados neste projeto foram computadores pessoais: um *desktop* com processador da Intel quad 2 core, 4GB de memória RAM DDR2, placa de vídeo Radeon HD 6950 2GB, *mouse* de alta sensibilidade Razer Taipan, monitor *full HD* 23" Samsung. O notebook Asus K45VM, com processador da Intel Core i7, 8GB de memória RAM DDR3, placa de vídeo Geforce GT 630M 2GB, *mouse* de alta precisão Razer Orochi, ambos os computadores com o mesmo sistema operacional Windows 8. Os *smartphones* utilizados para testes são: Sony Ericsson Xperia Arc com o Android na versão 4.0.4 (Ice Cream Sandwich) e o Samsung Galaxy S5 com o Android na versão 4.4.2 (Kit Kat).

3.2 Desenvolvimento do projeto

3.2.1 Aquisição dos conhecimentos necessários

Para alcançar os objetivos propostos, primeiramente foi analisado como é o desenvolvimento, como é realizada a criação, a modificação e a implementação de terrenos na *engine* Unity 3D. Na página da própria

ferramenta³², existem vários tutoriais no formato de videoaula que demonstram passo a passo todo o processo referente a terrenos. Além disso, os fóruns da Unity 3D são muito utilizados³³ para esclarecer algumas dúvidas que podem surgir durante o desenvolvimento. No momento da implementação, foi consultada a página de *script reference*³⁴, para saber quais bibliotecas foram utilizadas no desenvolvimento e também quais parâmetros em cada método em sua implementação. No caso deste projeto foi utilizada a linguagem *Java Script*, por ser de nosso domínio. Há também o manual da *engine* disponível³⁵ on-line, que aponta cada componente de cada função da ferramenta.

Para a aquisição dos conhecimentos, foram vistos os vídeos: *Terrain Sculpting*³⁶, *Environment Details*³⁷, *Mobile Development*³⁸, *Unity 3d Tutorials Episode: 1 (Creating Terrains)*³⁹, *Unity 3d Tutorials Episode: 4 (More Details and Image Effects)*⁴⁰, *Unity Touch Input Buttons Tutorial (Part 1)*⁴¹, *[Unity3D] Creating underwater effects in Unity3D*⁴² e Configurando o Unity3D para Android⁴³. Os três primeiros vídeos citados são oficiais do Unity 3D e são encontrados no site oficial na categoria aprender, tutoriais no tópico de arquivo aprender ao vivo⁴⁴ na lição para iniciantes. Os demais vídeos foram encontrados no Youtube. O conteúdo é de criação e detalhamento de terrenos e também implementação do *touch screen* para Android.

³² <http://unity3d.com/pt/learn/tutorials/modules>

³³ <http://forum.unity3d.com>

³⁴ <http://docs.unity3d.com/ScriptReference/>

³⁵ <http://docs.unity3d.com/Manual/>

³⁶ <http://unity3d.com/learn/tutorials/modules/beginner/live-training-archive/terrain-sculpting>

³⁷ <http://unity3d.com/learn/tutorials/modules/beginner/live-training-archive/environment-details>

³⁸ <http://unity3d.com/pt/learn/tutorials/modules/beginner/live-training-archive/mobile-development>

³⁹ <https://www.youtube.com/watch?v=EYDBrOLke8w>

⁴⁰ <https://www.youtube.com/watch?v=CNjsnKx7Pfl>

⁴¹ <https://www.youtube.com/watch?v=uUIXFL2ic7k>

⁴² <https://www.youtube.com/watch?v=FoZwgRE5LYI>

⁴³ <https://www.youtube.com/watch?v=tG6XacGn8JQ>

⁴⁴ <http://unity3d.com/pt/learn/tutorials/modules/live-training-archive>

3.2.2 Importação dos pacotes e criação das cenas

O projeto no Unity foi criado e foram escolhidos os pacotes que são utilizados para o desenvolvimento dos terrenos do trabalho. São eles: *Free Rocks*⁴⁵, *Light Flares*, *Particles*, *Skyboxes*, *Standard Assets (Mobile)*, *Terrain Assets*, *Water (Basic)* e *World Builder*⁴⁶. É relevante ressaltar que os pacotes, *Free Rocks* e *World Builder* são gratuitos e disponíveis na Asset Store⁴⁷ do Unity. Existem dois pacotes *Terrain Assets*⁴⁸, um nativo da *engine* e outro disponível em sua loja virtual. Ambos foram utilizados para a execução do trabalho. Os demais citados são nativos da *engine*.

Para a navegação entre os terrenos foram criadas várias cenas para facilitar a exploração de cada informação pelo usuário, considerando que a maior parte dos *smartphones* possui telas pequenas. Portanto não é viável disponibilizar muitos dados na mesma tela. A Figura 7 apresenta a navegação de cenas desta aplicação.

⁴⁵ <https://www.assetstore.unity3d.com/en#!/content/19288>

⁴⁶ <https://www.assetstore.unity3d.com/en#!/content/11333>

⁴⁷ <https://www.assetstore.unity3d.com/en/>

⁴⁸ <https://www.assetstore.unity3d.com/en#!/content/6>

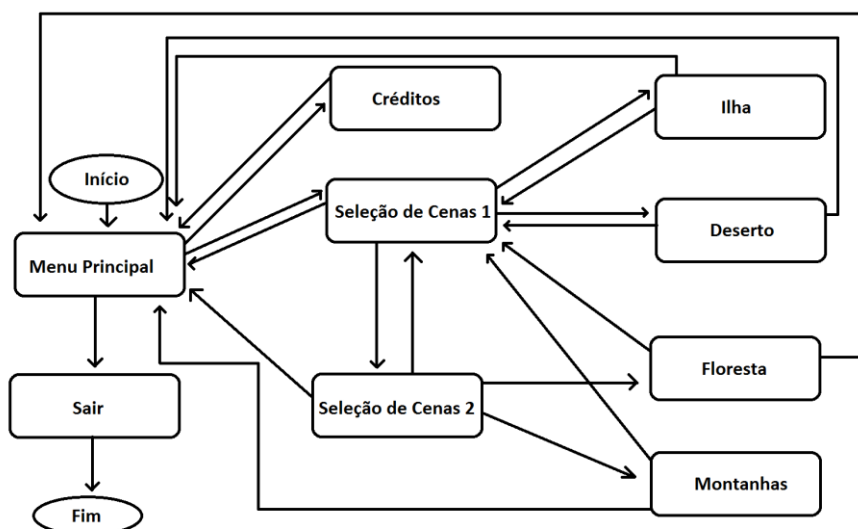


Figura 7: Grafo de navegação de cenas da aplicação desenvolvida neste projeto.

Além das telas de terrenos foram criadas as do menu principal (inicial), onde é possível acessar a tela de seleção de cenas, créditos e também sair da aplicação. Na tela seleção de cenas é possível acessar os dois primeiros terrenos da aplicação e também uma segunda página onde há os demais terrenos, além de ter a opção de voltar para o menu principal. Nas cenas de terrenos é possível ir para o menu principal e também para a seleção de cenas.

3.2.3 Criação da cena Menu Principal

A primeira tela criada foi a do Menu Principal. Foram utilizadas as ferramentas GUI, do inglês *Graphical User Interface*, nativa da Unity, *Button* (botão), para a interação com o usuário e a *Label* (rótulo), para mostrar um texto na tela. Foram criados dois *labels*, um com texto de “Projeto de Monografia” e outro “Giovanni Viol Assis” que foram centralizados e alterados para a cor branca. Para facilitar a leitura, foi criada uma imagem transparente com o contraste alfa de 60%. Esta foi colocada como fundo de cada *label* do início da largura até o final da largura da tela. Foram criados três *buttons*: o primeiro com o texto de “Seleção de cenas”, o segundo “Créditos”

e o terceiro “Sair”. Os *buttons* estão centralizados em relação à largura da tela. A sua largura é a mínima necessária para mostrar todo o texto do primeiro botão que possui o maior texto. Os demais botões possuem o mesmo tamanho. Além da criação dos *labels* e *buttons* há a necessidade de adaptá-los ao tamanho da tela de cada dispositivo. Para isto, o script altera o tamanho da fonte de ambos em 6% do tamanho da largura da tela. O tamanho de cada botão também varia de acordo com a quantidade de pixels do dispositivo. Foi criado também um *Quad* (*game object* do Unity) para uma imagem de fundo, utilizando-se uma imagem do próprio projeto. O material criado a partir dessa imagem teve o *Shader* (sombra) alterado para *Unlit/Texture* para não criar sombras. Removi o componente *Mesh Collider* deste *Quad*, pois ele não será usado, e posicionei-o de forma que ocupasse toda a tela. A Figura 4 (Seção Resultados e Discussão) apresenta a interface da cena.

3.2.4 Criação da cena de Créditos

Para a realização da tela de créditos, foram utilizados os mesmos recursos que o menu principal. Foram criados cinco *labels*: o primeiro “Projeto de Monografia”, o segundo “UFLA – Ciência da Computação”, o terceiro “Prof. Orientador: Joaquim Uchôa” (que teve a palavra Professor abreviada para poder manter a mesma fonte), o quarto *label* “Desenvolvedor: Giovanni Viol Assis” e o último “Contato: giubravo@yahoo.com.br”. Foi criado também um *button*, para poder voltar para o menu principal, localizado na parte inferior da tela. Como na cena do menu principal, foi criado um *quad* para se acrescentar uma imagem de fundo à cena. A Figura 5 (Seção Resultados e Discussão) apresenta esta tela.

3.2.5 Criação da Seleção de Cenas

Já na tela de seleção de cenas, primeiramente foram criados dois *labels*, como nas cenas anteriores. O primeiro “Projeto de Monografia” e o segundo

“Giovanni Viol Assis”. Logo após, foram criados dois *buttons* somente com imagens referentes a sua respectiva cena, o primeiro à ilha e o segundo ao deserto. Na parte inferior de cada *button* foi criado um *label* com um estilo diferente dos demais, não centralizado e na cor preta. O primeiro *label* foi denominado “Ilha” e o segundo “Deserto”. Logo abaixo foi criado um botão centralizado em relação à largura da tela com o texto “Menu Principal” e logo a sua direita outro *label*, para indicar que esta é a primeira página de duas (“1/2”) da seleção de cenas, sendo que este seguiu o último estilo criado. A sua direita foi criado outro *button* (“>”), para ter acesso a segunda página da seleção de cenas. Além dos *labels* e *buttons* foi criado também um *quad* para se acrescentar uma imagem de fundo e este foi posicionado para ocupar toda a tela. A imagem escolhida para esta cena foi uma foto tirada da janela de um avião em voo. A Figura 6 (Seção Resultados e Discussão) mostra a primeira página da seleção de cena.

Na segunda página da seleção de cenas, manteve-se todos os componentes da primeira. Porém os *buttons* tiveram suas imagens alteradas: o primeiro com uma da floresta e o segundo da montanha. Os *labels* logo abaixo desses *buttons* foram renomeados para “Floresta” e “Montanhas”, respectivamente. O último *label* também foi renomeado para “2/2” representando assim que o usuário está na segunda página. O último *button* teve seu texto alterado para “<”. Nele é possível retornar para a primeira página. Esta tela foi criada na mesma cena que a da primeira página da seleção de cenas no projeto do Unity, com a finalidade de se criar menos objetos na hierarquia e ocupar menos espaço. Os *buttons* e *labels* são alterados via script quando se muda de página, dando a impressão que se mudou de cena. Na Figura 7 (Seção Resultados e Discussão) é apresentada a segunda página da seleção de cenas.

Ainda na tela de seleção de cenas, quando algum *button* com imagem é acessado ele desaparece, assim como seus respectivos *labels* e o referente que indica a página e seu *button*. Cria-se então um *texture* com uma imagem transparente com contraste alfa de 60% em toda a tela e também um *text*

(elemento do GUI do Unity de texto) “Carregando...” para dar o *feedback* ao usuário de que a cena está sendo carregada em *buff* no dispositivo. Como são muitos componentes nessas cenas é comum demorar alguns segundos para que a tela seja carregada totalmente e assim é necessário esse aviso. Cria-se também um *button* para voltar à seleção de cenas. A Figura 8 (Seção Resultados e Discussão) ilustra essa situação.

3.2.6 Criação do terreno Ilha

Esta parte refere-se a elaboração dos terrenos. O primeiro terreno construído foi a ilha. Após criar a nova cena e logo em seguida um terreno, alterou-se o relevo utilizando a ferramenta contida no pacote *World Builder*. Após ele ser importado é possível utilizá-lo em *window*, *world buider*. Esse primeiro terreno foi gerado com os parâmetros padrões desta ferramenta. Logo após a dimensão do terreno foi alterada para 200 unidades de largura e comprimento e 60 de altura. Então alterou-se a posição X e Z para -100 unidades para que a origem do terreno se tornasse a origem da cena para facilitar futuros posicionamentos.

Começou-se então a fase de detalhamento do terreno, primeiramente pelo céu. Adicionou-se o *prefab skybox* contido no pacote *Standard Assets (Mobile)* à hierarquia do projeto que é nativo da *engine*, localizado na pasta *Prefabs*. Para posicioná-lo de acordo com o terreno, alterou-se a escala no X e Z para 20000 e Y para 10000. Para tratar a colisão foi adicionado um componente, o *Mesh Collider*. Logo após foi criado um *directional light* renomeado para sol e adicionado o *sun*, no seu parâmetro *flare*, contido no pacote importado *Light Flares*. É necessário então posicionar e rotacionar de acordo com o sol da imagem do *skybox* criado na etapa anterior para que ela fique coerente com o *skybox*.

Logo após a água foi criada, um *prefab*, do pacote *Water (Basic)* nativo do Unity. Aumentou-se a escala de forma que ela ficasse maior que o *skybox* e posicionou-a para enquadrar somente as bordas. Para dar o efeito de

debaixo d'água, é necessário duplicar esse *prefab*, rotacioná-lo no eixo X em 180° e movê-lo -0,1 no eixo Y para não sobrepor o anterior.

A próxima etapa foi a texturização do terreno. Quando se adiciona a primeira textura, a *engine* preenche todo o terreno com a que foi escolhida, portanto recomenda-se escolher a de maior área. No caso a primeira textura escolhida foi a de grama presente no pacote *Terrain Assets*. Adicionou-se a textura areia para fazer a borda do terreno com a textura “areia molhada”. Para fazer áreas rochosas utilizou-se a textura “pedra”. Para as partes intermediárias adicionou-se a textura “pedra com grama”. E para fazer o caminho foi utilizada a textura “grama com terra”.

A fim de ficar mais próximo da realidade, foi utilizado o pacote *Free Rocks* disponível na *Asset Store* para adicionar pedras ao terreno. Este pacote vem com quatro *prefabs*. Utilizou-se então cada um, posicionando-os em uma parte conveniente, para manter a coerência com o terreno e também foram rotacionados e aumentados de tamanho.

Além de um terreno com relevo, céu, água e pedras é necessário que haja também vegetações, pois quanto maior a realidade maior a imersão virtual e por consequência maior a diversão, que é o foco de um jogo. A vegetação nesse projeto foi classificada como árvores grandes e pequenas, com e sem animação. Quanto mais se insere objetos animados maior o processamento durante a aplicação. Primeiramente foram adicionados seis tipos de árvores classificadas como grandes e sem animação: *Alder*, *Palm*, *Tree*, *Scots Pine Type A*, *Willow* e *Bamboo*. Elas foram espalhadas pelo terreno de forma que ficassem harmoniosas no ambiente. Posteriormente foram criadas mais três árvores de pequeno porte: *Bush 4*, *Bush 7* e *Fern*. Já essas, classificadas como pequenas e sem animação, foram posicionadas próximas das maiores, porém em maior quantidade. Posteriormente, adicionou-se a vegetação com animação: *Grass*, *Grass 2*, *Ferns*, *Bush 6* e *Weed*.

Para que se tenha uma cena jogável, é necessário que haja personagens. Como o *prefab* do personagem em terceira pessoa, nativo do

Unity, não se encaixa com os terrenos criados neste projeto, foi utilizado o do modo em primeira pessoa presente no pacote *Standard Assets (Mobile)*. Foi necessário fazer algumas alterações neste personagem para se encaixar melhor neste projeto, como acrescentar os componentes (da própria *engine*) *Rigidbody*, marcar a opção *is Kinematic* (que é utilizada para a física do jogo) e o *Capsule Collider*, alterar a altura deste para 2 unidades, para o sistema de colisões. Ambos foram incluídos no *prefab Player*. Além disso, o *GUI Texture* (Textura) que vem presente nele não se adapta automaticamente a posição nem o tamanho em relação ao tamanho da tela do dispositivo. Foi alterado então o *script* nele presente, *Joystick*, para que as adaptações fossem feitas. Por último, foi adicionado um som ambiente: as ondas do mar.

Foi criado o primeiro terreno do projeto, a Ilha. A Figura 9 (Seção Resultados e Discussão) ilustra esta cena.

3.2.7 Criação do terreno Deserto

A construção dos demais terrenos foi bem parecida com a da ilha, porém com alguns aprimoramentos. No caso da cena do deserto, ao invés da água nos limites, criou-se montanhas altas para o jogador não conseguir ultrapassá-las. Essas montanhas foram criadas usando a ferramenta *Paint height* do próprio terreno, posteriormente suavizada com a ferramenta *Smooth Height*, também do próprio terreno. O som ambiente neste caso é de vento. A textura predominante é a areia, com pouca rocha e grama. Já nas montanhas, é “barro vermelho”. A predominância da vegetação é de árvores pequenas sem movimentação. Esse terreno foi criado com menos detalhes em relação ao anterior, propositalmente com a finalidade de testes em dispositivos com menos processamento e memória. Além disso, o relevo é mais montanhoso em relação ao anterior. A Figura 10 (Seção Resultados e Discussão) ilustra esta cena.

3.2.8 Criação do terreno Floresta

O próximo terreno construído foi a floresta. A vegetação é bem densa nesta cena. Predomínio de árvores altas sem movimento, havendo também vegetações baixas com e sem movimentação. A textura predominante foi a grama. Nas regiões próximas à água utilizou-se a textura “rocha com grama” e pelo caminho foi usada a “grama desgastada”. O destaque deste terreno é a cachoeira, que, para ser criada, utilizou-se o pacote *Particles*. Tal pacote é dividido em dois *prefabs*: *WaterFall* e *Water Surface Splash*. Para aproveitar toda a altura da partícula *WaterFall*, foi necessário alterar o relevo. Para criar o lago da parte superior foi utilizada a ferramenta *Paint height* para elevar o relevo até a altura de 30 unidades. Logo após, com um *brush* menor e altura de 25 unidades, criou-se o relevo do lago. Para suavizar a diferença de relevo criada por esta modificação, foi utilizada a ferramenta *Raise / Lower Terrain* contida no próprio terreno e posteriormente a *Smooth Height*. O lago da parte inferior é bem parecido, porém com menores unidades. Ajustou-se então a posição e a rotação das partículas de forma que ficassem próximo da realidade. Foi pertinente adicionar pedras e som ambiente da cachoeira e em todo o terreno de cigarras e grilos para dar uma maior imersão virtual ao usuário. A Figura 11 (Seção Resultados e Discussão) ilustra esta cena em execução.

3.2.9 Criação do terreno Montanhas

O último terreno desenvolvido foi o das montanhas. O relevo ficou parecido com o deserto, inclusive os seus limites. A textura predominante foi “rochas”. O caminho foi feito com uma outra textura de rocha um pouco mais escura, com bastante presença de neve em todo o terreno, principalmente nas extremidades das montanhas. A vegetação tem predomínio de árvores altas e sem movimento, típica dessa região, mas há a presença também de baixas com e sem movimentação. O diferencial deste terreno foi a chuva de neve em tempo real. Vários *prefabs* *Light Snow*, contidos no pacote *Particles*, foram

distribuídos em todo o terreno, de forma que a área de cada um ficasse visível para o usuário e não muito próximos uns dos outros. A Figura 12 (Seção Resultados e Discussão) ilustra esta cena em execução.

3.2.10 Criação da interface dos terrenos

A jogabilidade desta aplicação nos terrenos é realizada através de dois *joysticks* virtuais representados no canto inferior da tela: o da esquerda para movimentar o personagem para frente, trás e lados e o da direita para rotacionar a câmera em todas as direções, além de, se o usuário interagir de forma que ele toque duas vezes neste *joystick* rapidamente, o personagem pula. Foram criados dois *labels* para instruir o usuário como interagir. É possível desabilitar estes avisos ativando o *button* “Desativar Dicas” no canto superior esquerdo e reativá-las se o usuário achar necessário. Existe também o *button* “II” para pausar a aplicação. Quando ele é acessado, aparece um menu com quatro *buttons*: o primeiro “Continuar”, o segundo “Recomeçar”, o terceiro “Seleção de Cenas” e o último “Menu Principal”. Surge também um *texture* com uma imagem transparente com contraste de alfa em 60% para escurecer o fundo. A Figura 13 (Seção Resultados e Discussão) reproduz a visão do usuário nas cenas de terrenos e a 14 (Seção Resultados e Discussão) mostra o que acontece quando o *button* de *pause* é pressionado. As implementações referentes a cada cena e funcionalidades Jogabilidade, Dicas e Pause se encontram no Apêndice A.

3.2.11 A compatibilidade desta aplicação

A aplicação é compatível com a plataforma Android 2.3.1 (*Gingerbread*) ou superior, com arquitetura de processador ARM (*Advanced RISC Machine*) 7 ou superior e também compatível com a x86, ambas com 32 bits. O APK (*Android Package*), que é o arquivo de instalação, possui aproximadamente 55 MB (*mega bytes*) sendo, portanto, necessário haver esse espaço disponível

no armazenamento interno do dispositivo para que seja possível a instalação. Posteriormente há a opção de mover a aplicação para o cartão SD para liberar espaço na memória interna. O arquivo de instalação está disponível em três servidores: Dropbox⁴⁹, Google Drive⁵⁰ e Onedrive⁵¹.

3.2.12 Os testes com os usuários

Para testar a aplicação foram realizados testes com usuários para registrar quantos e quais dispositivos móveis a aplicação é realmente compatível, além de testar também seu desempenho em cada aparelho. O questionário foi desenvolvido no site Survey Monkey⁵², que é próprio para criação de formulários. Ele foi dividido em 3 páginas, cada uma a respeito de uma etapa do teste contendo 10 questões no total. O questionário é apresentado Apêndice B.

⁴⁹

<https://www.dropbox.com/s/2j41erv0otrs7zk/Projeto%20Monografia%20Giovanni.apk?dl=0>

⁵⁰

https://drive.google.com/file/d/0B7H_1sPeErulSDU4dWN5OU12V0k/view?usp=sharing

⁵¹ <https://onedrive.live.com/redir?resid=11793333CF3102EE%21315>

⁵² <https://pt.surveymonkey.com/>

4 RESULTADOS E DISCUSSÃO

4.1 Resultado das criações de cenas

A criação da cena do Menu Principal resultou na Figura 8.



Figura 8: *Print Screen* da cena Menu Principal da aplicação desenvolvida neste projeto.

A cena de créditos teve o resultado da Figura 9.

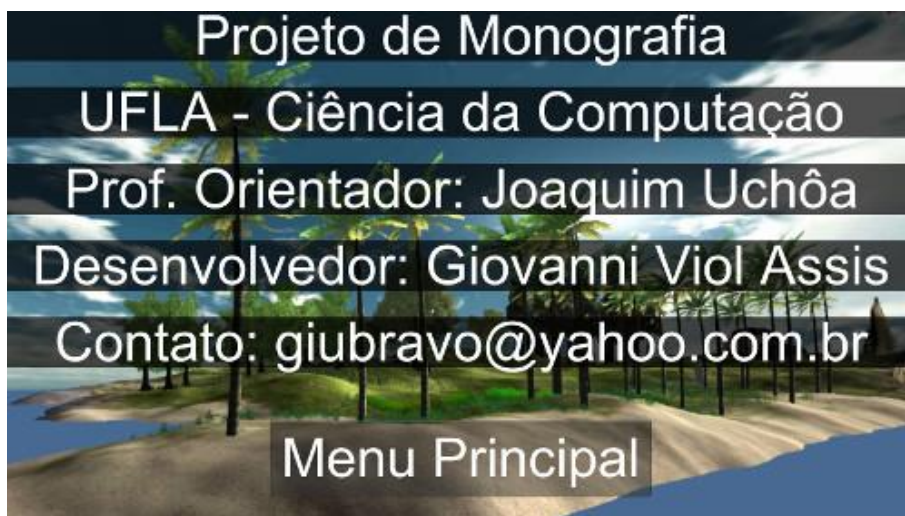


Figura 9: *Print Screen* da cena Créditos da aplicação desenvolvida neste projeto.

A 1ª página da Seleção de Cenas teve o resultado da Figura 10.



Figura 10: *Print Screen* da Seleção de Cenas 1 da aplicação desenvolvida neste projeto.

Já a 2ª página da Seleção de Cenas teve o resultado da Figura 11.



Figura 11: *Print Screen* da Seleção de Cenas 2 da aplicação desenvolvida neste projeto.

O resultado da cena Carregando é representado pela Figura 12.



Figura 12: *Print Screen* da cena Carregando da aplicação desenvolvida neste projeto.

4.2 Resultado das criações dos terrenos

Parte do resultado do desenvolvimento do 1º terreno: Ilha foi representado na Figura 13.



Figura 13: *Print Screen* da cena Ilha da aplicação desenvolvida neste projeto.

Parte do resultado do desenvolvimento do 2º terreno: Deserto foi representado na Figura 14.



Figura 14: *Print Screen* da cena Deserto da aplicação desenvolvida neste projeto.

Parte do resultado do desenvolvimento do 3º terreno: Floresta foi representado na Figura 15.

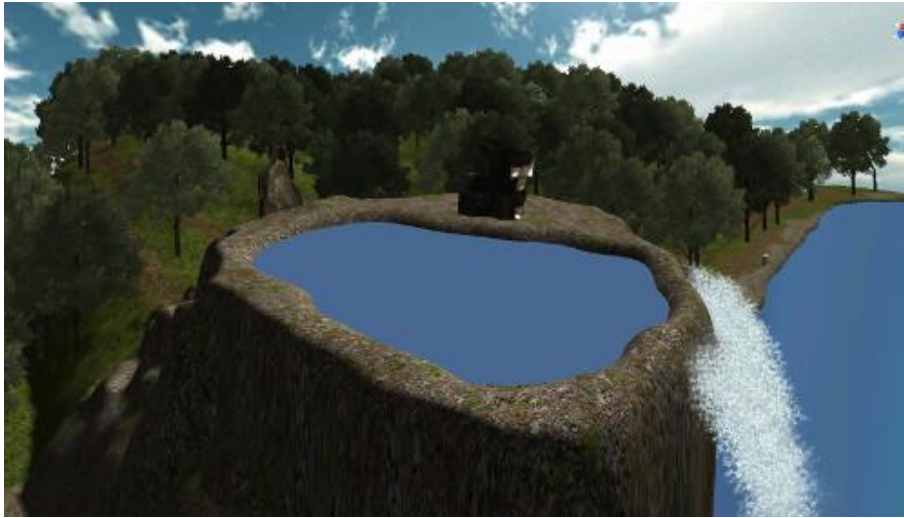


Figura 15: *Print Screen* da cena Floresta da aplicação desenvolvida neste projeto.

Parte do resultado do desenvolvimento do 4º terreno: Montanhas foi representado na Figura 16.

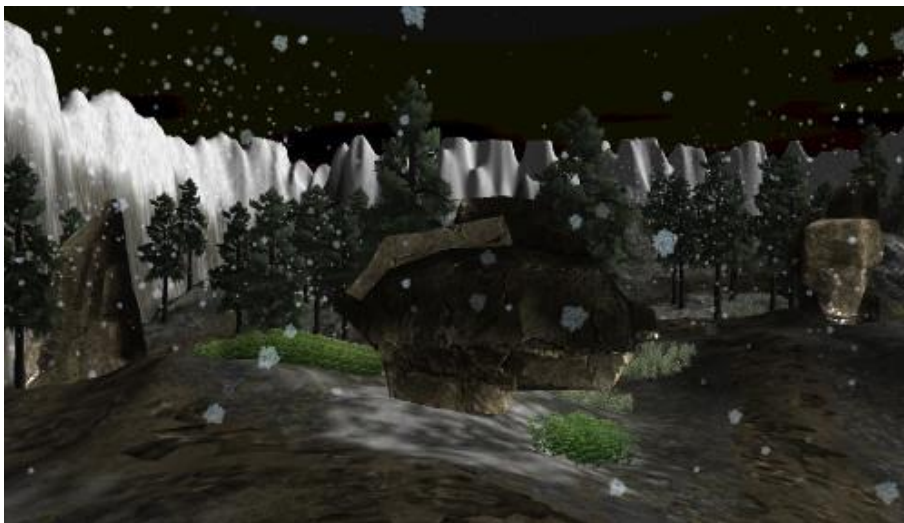


Figura 16: *Print Screen* da cena Montanhas da aplicação desenvolvida neste projeto.

4.3 Resultado da criação da interface

A interface de jogabilidade nos terrenos teve o resultado representado na Figura 17.



Figura 17: *Print Screen* da interface ao usuário da aplicação desenvolvida neste projeto.

A interface do Pause é apresentada na Figura 18.

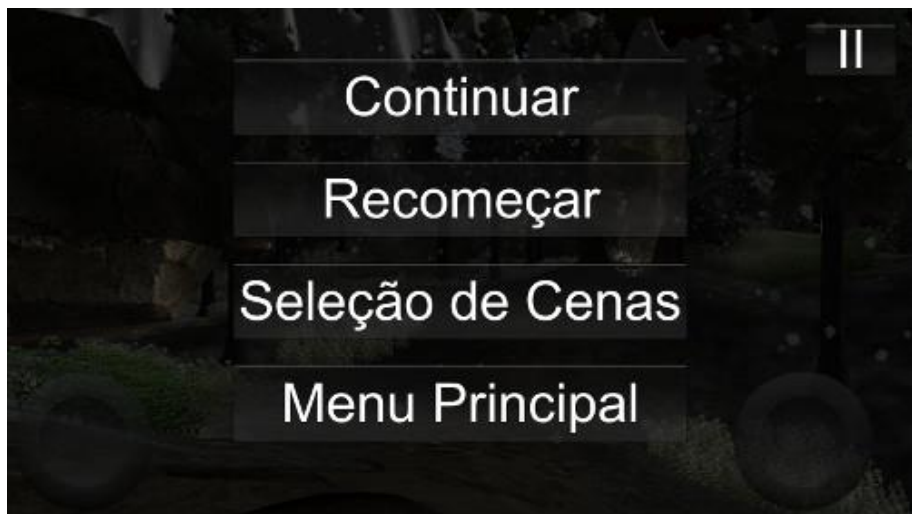


Figura 18: *Print Screen* da interface Pause da aplicação desenvolvida neste projeto.

4.4 Resultados dos testes com os usuários

A maioria dos usuários convidados a testar a aplicação são ou foram alunos da UFLA (Universidade Federal de Lavras). O número total de questionários respondidos foi 35.

O resultado da 1ª questão foi:

Tabela 1: Resposta da questão 1 do formulário dos testes com usuários.

Marcas:	Número de dispositivos utilizados nestas marcas:
Samsung	19
Motorola	9
LG	4
Genesis	1
Asus	1
Computador Pessoal	1
Total:	35

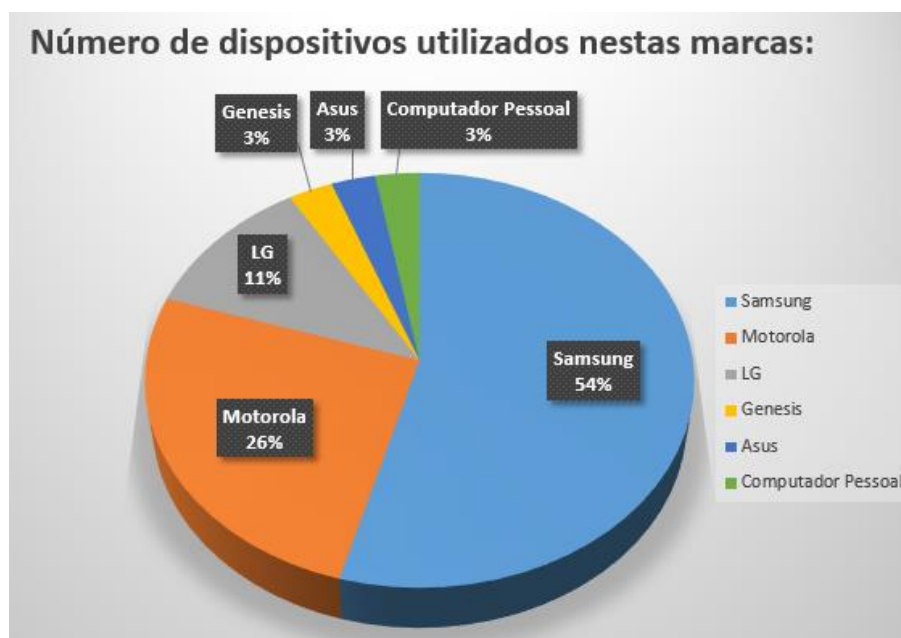


Figura 19: Gráfico do resultado da questão 1 do formulário dos testes com usuários.

Ao analisar os resultados do questionário percebe-se na primeira questão, referente à marca do dispositivo, que a maioria dos usuários testaram a aplicação em algum dispositivo da marca Samsung. Em segundo lugar veio a marca Motorola, seguida da LG, Asus e Genesis. A Figura 19 ilustra estes resultados em um gráfico de pizza.

O resultado da 2ª questão foi:

Tabela 2: Resposta da questão 2 do formulário dos testes com usuários.

Modelos:	Número de dispositivos utilizados nestes modelos:
Galaxy S5	1
Galaxy S3	5
Galaxy S3 neo	1
Galaxy S3 mini	1
Galaxy S2	2
Galaxy Pocket	1
Galaxy Tab	1
Galaxy Note 3	1
Galaxy Ace	2
Galaxy Y	1
Galaxy Win Duos	3
Moto X 2ª geração	1
Moto X	1
Moto G	4
L70	1
Razr i	1
Razr D1	1
Razr D3	1
GT-7240R	1
Zenfone 5	1

LG-P350f	1
Optimus L7	1
LG-C660h	1
Computador pessoal	1
Total:	35

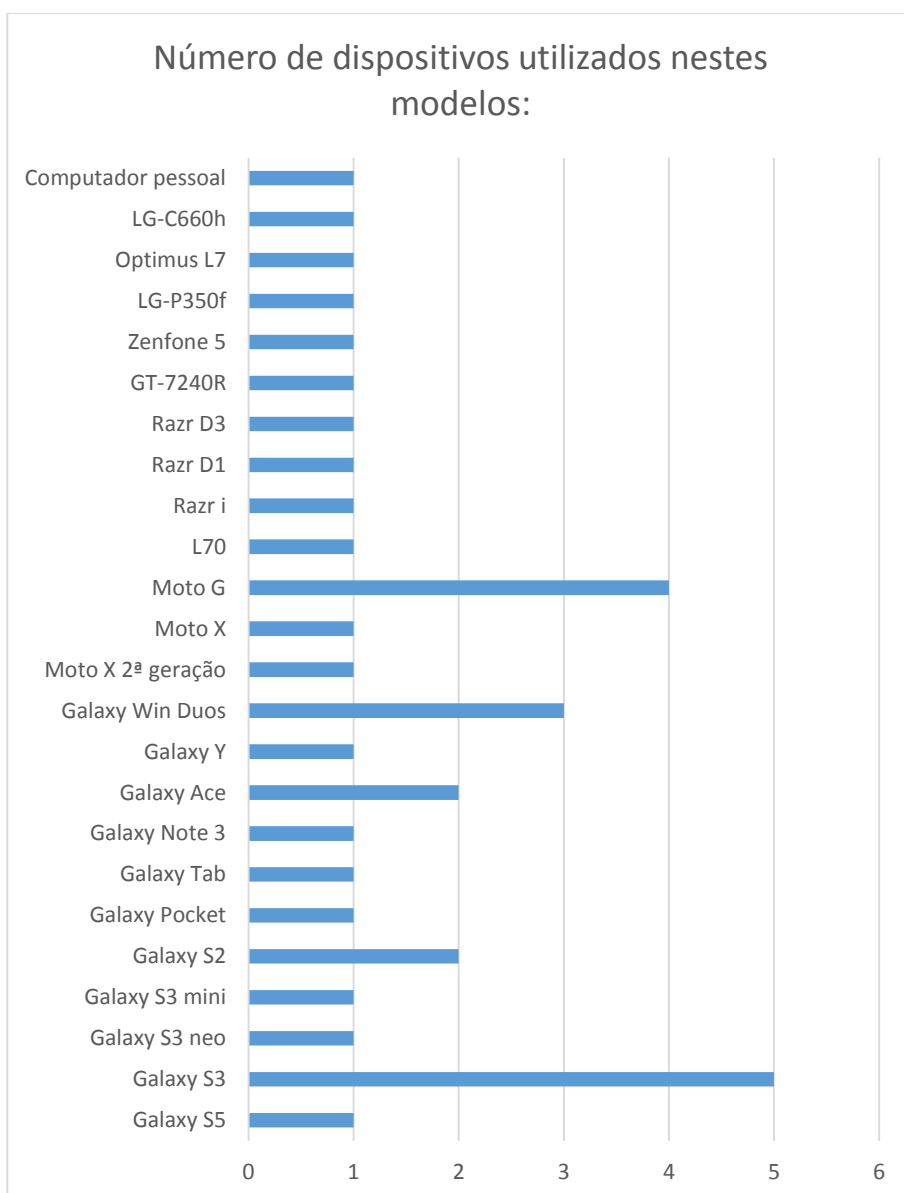


Figura 20: Gráfico do resultado da questão 2 do formulário dos testes com usuários.

A segunda questão se referia ao modelo do dispositivo utilizado nos testes com os usuários. Vários modelos foram citados, desde os mais antigos e simples do ano de 2010 até os mais novos e sofisticados de 2014. A Figura 20 possui o gráfico com estes resultados.

O resultado da 3ª questão foi:

Tabela 3: Resposta da questão 3 do formulário dos testes com usuários.

Versões do Android:	Número de dispositivos utilizados nestas versões:
4.4.4 (KitKat)	6
4.4.2 (KitKat)	3
4.3 (Jelly Bean)	4
4.1.2 (Jelly Bean)	5
4.1.1 (Jelly Bean)	2
4.0.3 (Ice Cream Sandwich)	2
4 (Ice Cream Sandwich)	5
2.3.6 (Gingerbread)	4
2.3.4 (Gingerbread)	1
2.3.3 (Gingerbread)	1
2.3.1 (Gingerbread)	1
2.2.2 (Froyo)	1
Total:	35

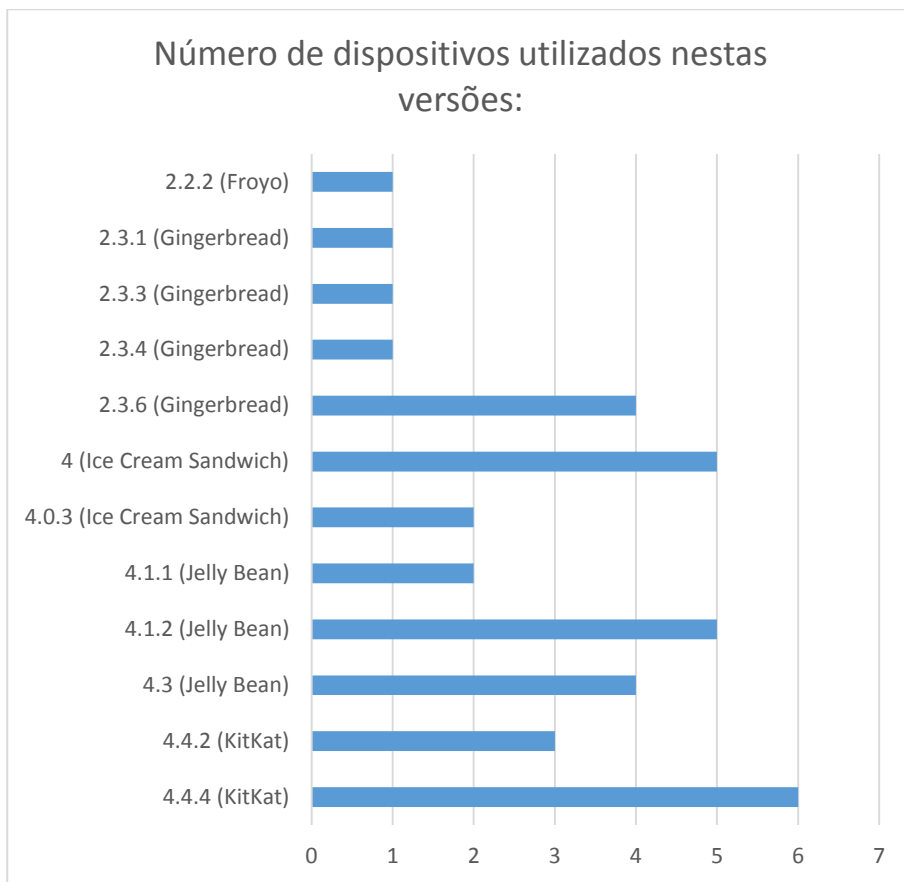


Figura 21: Gráfico do resultado da questão 3 do formulário dos testes com usuários.

A terceira questão foi sobre a versão do Android utilizada. As versões também variaram bastante, desde a 2.2.2 (Froyo) até a 4.4.4 (KitKat). Como a aplicação é compatível somente com as versões 2.3.1 (Gingerbread) ou superiores, os usuários do Froyo não conseguiram instalar a aplicação. Alguns que possuem a Gingerbread não conseguiram instalá-la provavelmente por falta de compatibilidade com a arquitetura do processador, pois a aplicação é compatível somente com a ARM7 ou superior e x86. Sabe-se que alguns dispositivos mais antigos com a versão Gingerbread possuem a ARM6. A Figura 21 possui um gráfico de barras ilustrando estes resultados.

A segunda página do questionário, que abrange as questões de 4 a 6, referia-se ao processo de instalação da aplicação. O resultado da 4ª questão foi:

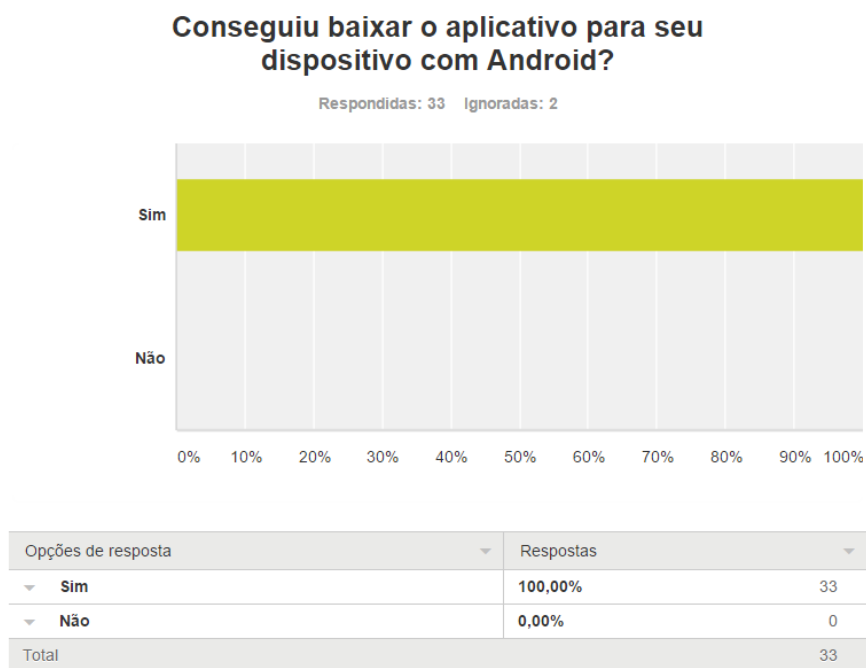


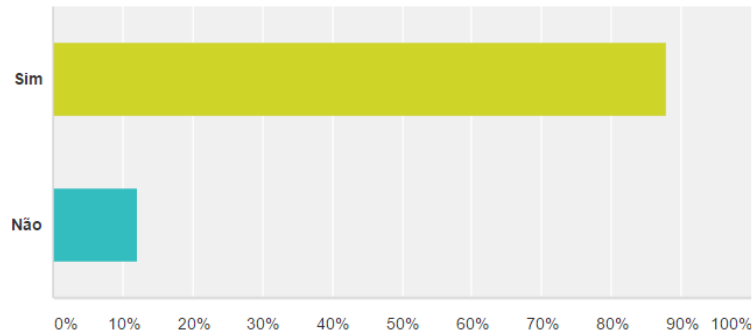
Figura 22: *Print screen* do resultado da questão 4 do formulário dos testes com usuários.

A quarta pergunta questionou se o usuário conseguiu baixar o apk, que é o instalador. A resposta a essa questão foi 100% positiva. A Figura 22 ilustra estes resultados.

O resultado da 5ª questão foi:

Conseguiu instalar a aplicação?

Respondidas: 33 Ignoradas: 2



Opções de resposta	Respostas
Sim	87,88% 29
Não	12,12% 4
Total	33

Figura 23: *Print screen* do resultado da questão 5 do formulário dos testes com usuários.

Esta questão investigou se os usuários conseguiram instalar a aplicação. 87,88% apontaram que foi possível concluir a instalação com êxito. A Figura 23 possui um gráfico com estes resultados.

O resultado da 6ª questão foi:

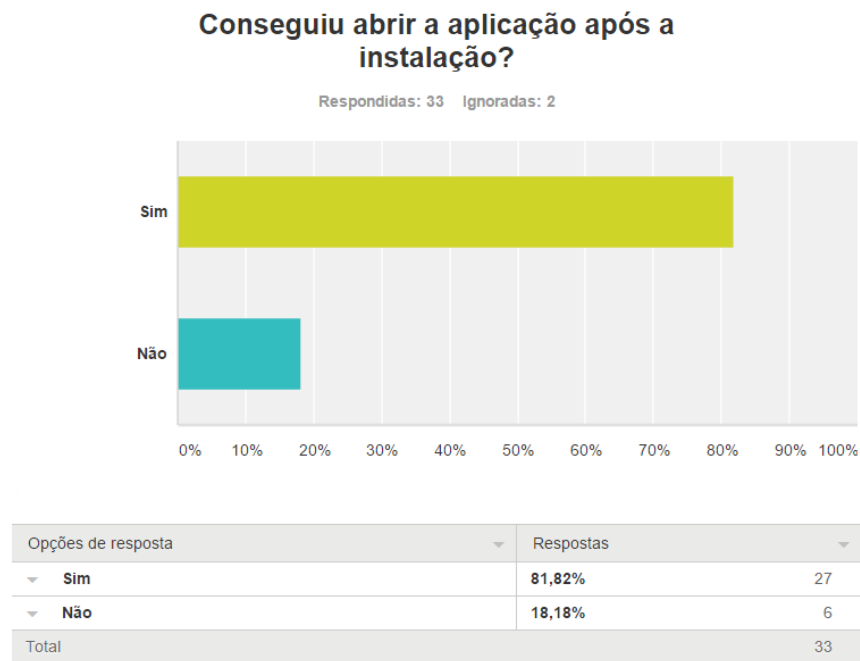


Figura 24: *Print screen* do resultado da questão 6 do formulário dos testes com usuários.

A última pergunta dessa página, de número seis, questionou se além de baixar e instalar a aplicação foi possível abri-la. O resultado obtido foi de 81,82% de sucesso. A Figura 24 possui um gráfico com estes resultados.

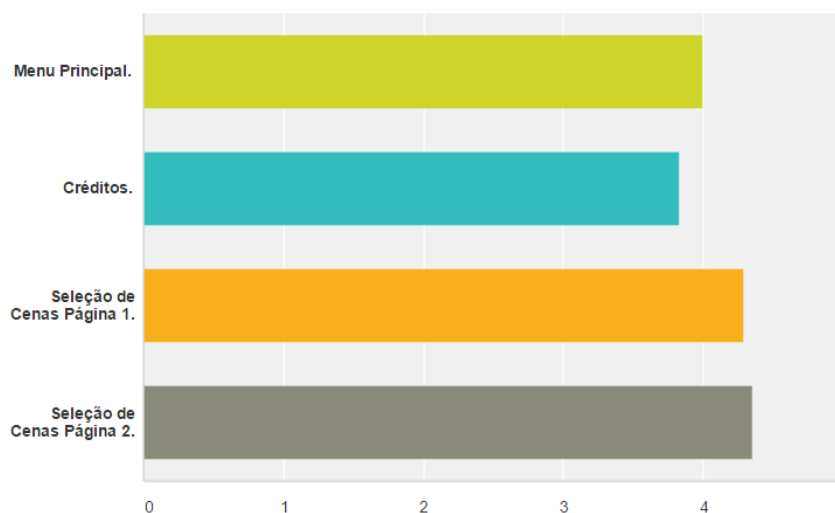
É importante ressaltar que caso algum usuário marcasse não em alguma questão dessa página não haveria o porquê de responder a próxima. E se algum usuário utilizou mais de um dispositivo para o teste foi recomendado que ele respondesse um questionário referente a cada dispositivo, havendo assim a necessidade de responder somente até a 2ª página para os demais dispositivos.

A 3ª página foi referente à opinião dos usuários sobre a interface de cada cena e dos terrenos e sobre a jogabilidade e desempenho da aplicação em seus dispositivos.

O resultado da 7ª questão foi:

Avalie a interface das seguintes telas:

Respondidas: 26 Ignoradas: 9



	Muito ruim	Ruim	Médio	Bom	Muito Bom	Total	Avaliação média
Menu Principal.	0,00% 0	7,69% 2	19,23% 5	38,46% 10	34,62% 9	26	4,00
Créditos.	0,00% 0	13,04% 3	21,74% 5	34,78% 8	30,43% 7	23	3,83
Seleção de Cenas Página 1.	0,00% 0	0,00% 0	25,00% 6	20,83% 5	54,17% 13	24	4,29
Seleção de Cenas Página 2.	0,00% 0	0,00% 0	26,09% 6	13,04% 3	60,87% 14	23	4,35

Figura 25: *Print screen* do resultado da questão 7 do formulário dos testes com usuários.

Já esta questão, foi sobre a avaliação da interface das telas Menu Principal, Créditos, Seleção de cena página 1 e Seleção de cena página 2.

Dentre os usuários que responderam essa questão, 0% acharam a interface Menu Principal muito ruim, 7,69% ruim, 19,23% médio, 38,46% bom e 34,62% muito bom. No total 26 pessoas responderam e o restante delas ignorou as perguntas, provavelmente por não conseguirem abrir a aplicação. Em relação à página Créditos, 0% acharam a interface muito ruim, 13,04% ruim, 21,74% médio, 34,78% bom e 30,43% muito bom, totalizando 23

usuários. Sobre a tela Seleção de cena página 1, 0% acharam a interface muito ruim, 0% ruim, 25% médio, 20,83% bom e 54,17% muito bom, com 24 respostas. E na tela Seleção de cena página 2, 0% acharam a interface muito ruim, 0% ruim, 26,09% médio, 13,04% bom e 60,87% muito bom, totalizando 23 respostas. A Figura 25 possui um gráfico com estes resultados.

O resultado da 8ª questão foi:



Figura 26: *Print screen* do resultado da questão 8 do formulário dos testes com usuários.

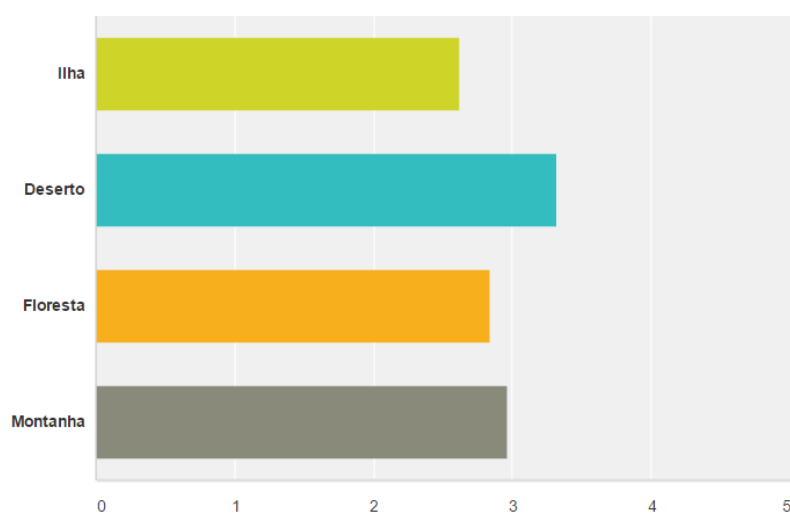
Esta questão foi como o usuário avaliou a interface dos seguintes terrenos: ilha, deserto, floresta e montanhas. Dos que responderam essa questão, 0% acharam a interface da ilha muito ruim, 0% ruim, 7,69% médio,

46,15% bom e 46,15% muito bom, totalizando 26 respondentes. Em relação ao terreno deserto, 0% acharam a interface muito ruim, 0% ruim, 3,85% médio, 46,15% bom e 50% muito bom, de 26 respostas. Sobre o terreno da floresta, 0% acharam a interface muito ruim, 3,85% ruim, 7,69% médio, 30,77% bom e 57,69% muito bom, também de 26 respostas. E no terreno de montanhas, 0% acharam a interface muito ruim, 0% ruim, 8% médio, 36% bom e 56% muito bom, com 25 respondentes. A Figura 26 possui um gráfico com estes resultados.

O resultado da 9ª questão foi:

Qual o desempenho dos seguintes terreno:

Respondidas: 26 Ignoradas: 9



	Muito pesado	Pesado	Médio	Leve	Muito leve	Total	Avaliação média
Ilha	26,92% 7	19,23% 5	19,23% 5	34,62% 9	0,00% 0	26	2,62
Deserto	8,00% 2	8,00% 2	32,00% 8	48,00% 12	4,00% 1	25	3,32
Floresta	12,00% 3	20,00% 5	40,00% 10	28,00% 7	0,00% 0	25	2,84
Montanha	8,00% 2	28,00% 7	24,00% 6	40,00% 10	0,00% 0	25	2,96

Figura 27: Print screen do resultado da questão 9 do formulário dos testes com usuários.

Esta questão⁵³ foi referente à opinião dos usuários em relação ao desempenho da aplicação em seus dispositivos nos terrenos já citados. Dos usuários que responderam essa questão, 26,92% acharam que o terreno ilha requer todo o recurso computacional disponível exigindo o máximo do dispositivo, 19,23% muito recurso computacional exigindo muito do dispositivo, 19,23% médio recurso computacional exigindo moderadamente do dispositivo, 34,62% pouco recurso computacional exigindo pouco do dispositivo e 0% praticamente nenhum recurso computacional exigindo muito pouco do dispositivo, com um total de 26 respostas.

Em relação ao terreno deserto, 8% acharam o desempenho requer todo recurso computacional exigindo o máximo do dispositivo, 8% muito recurso computacional exigindo muito do dispositivo, 32% médio recurso computacional exigindo moderadamente do dispositivo, 48% pouco recurso computacional exigindo pouco do dispositivo e 4% praticamente nenhum recurso computacional exigindo muito pouco do dispositivo, totalizando 25 respostas.

Sobre o terreno da floresta, 12% acharam que o desempenho requer todo recurso computacional exigindo o máximo do dispositivo, 20% muito recurso computacional exigindo muito do dispositivo, 40% médio recurso computacional exigindo moderadamente do dispositivo, 28% pouco recurso computacional exigindo pouco do dispositivo e 0% praticamente nenhum recurso computacional exigindo muito pouco do dispositivo, de 25 respondentes.

Já no terreno de montanhas 8% acharam o desempenho requer todo recurso computacional exigindo o máximo do dispositivo, 28% muito recurso computacional exigindo muito do dispositivo, 24% médio recurso computacional exigindo moderadamente do dispositivo, 40% pouco recurso

⁵³ As perguntas referentes ao uso do recurso computacional utilizado em cada terreno foram alteradas com o intuito de facilitar a comunicação de forma objetiva com os usuários. Originalmente, as questões trazem os termos: muito pesado, pesado, médio, leve e muito leve.

computacional exigindo pouco do dispositivo e 0% praticamente nenhum recurso computacional exigindo muito pouco do dispositivo, de 25 respostas. A Figura 27 possui um gráfico com estes resultados.

O resultado da 10ª questão foi:

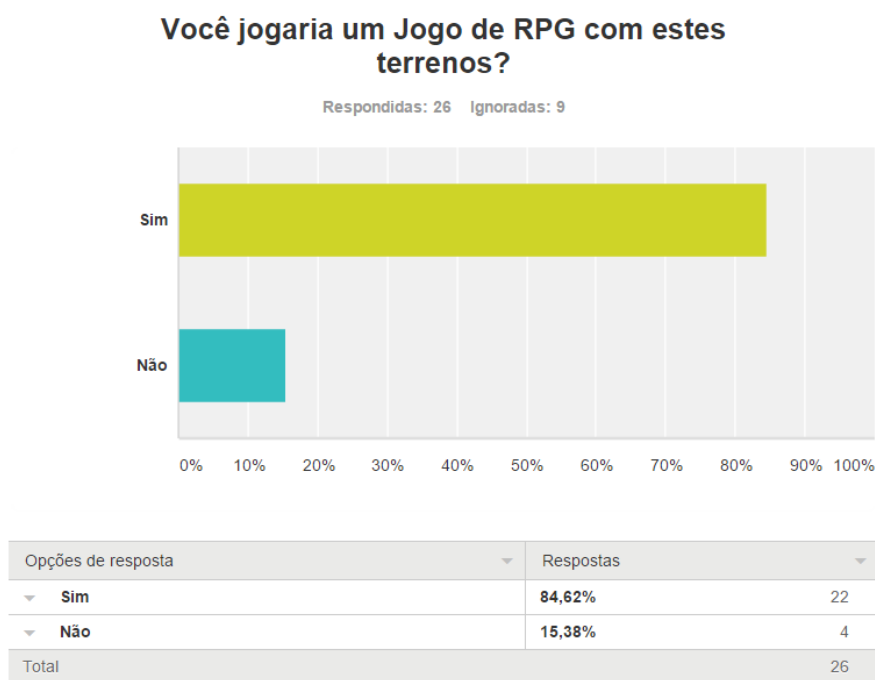


Figura 28: *Print screen* do resultado da questão 10 do formulário dos testes com usuários.

A última questão do questionário abordou se o usuário que testou a aplicação jogaria um jogo de RPG utilizando esses terrenos. 84,62% responderam sim de 26 usuários. A Figura 28 ilustra com um gráfico destes resultados.

No geral, ao analisar esses resultados, percebe-se que a maioria dos usuários conseguiram baixar, instalar e testar a aplicação em seus dispositivos. A interface das telas e dos terrenos teve a avaliação classificada como muito boa, o desempenho dos terrenos como médio e a maioria dos usuários afirmou que jogaria um jogo de RPG com os terrenos desenvolvidos neste projeto.

5 CONCLUSÃO

Neste projeto foram criados terrenos para um jogo de RPG com o tema medieval para a plataforma Android utilizando a *engine* Unity 3D. Foram criados 4 tipos de terrenos: ilha, deserto, floresta e montanhas. É possível explorar os terrenos através de *joysticks* virtuais que movimentam a câmera. Além de terrenos, a aplicação criada possui várias cenas que permitem acessar os créditos e também todos os terrenos citados.

Com este trabalho foi possível perceber o quanto as *engines* facilitam o desenvolvimento de um jogo e a portabilidade para múltiplas plataformas, incluindo a compatibilidade entre várias versões e arquiteturas de processadores. Para verificar a compatibilidade entre várias marcas e modelos distintos, testes com os usuários foram realizados.

Ao analisar os resultados e comentários dos usuários, pode-se concluir que esta aplicação obteve uma compatibilidade aceitável e também uma boa avaliação das interfaces. O objetivo geral proposto foi alcançado com êxito, pois os terrenos foram desenvolvidos e a aplicação foi compatível com vários dispositivos Android. Quanto aos objetivos específicos, a criação, o detalhamento, a animação e os sons dos terrenos foram concluídos, porém não foi possível fazer algumas correções das colisões propostas.

A contribuição deste trabalho é mostrar que desenvolvedores de jogos podem implementar terrenos por meio de uma *engine* gratuita e com grande compatibilidade dentre as versões de dispositivos móveis Android.

Recomenda-se que pesquisas futuras desenvolvam aplicações em que os terrenos sejam criados de forma aleatória a cada vez que se iniciar uma cena e que as vegetações, texturas e rochas estejam de acordo com o relevo gerado. Sugere-se também que seja gerado um mini mapa do terreno na interface para o usuário a fim de facilitar sua localização no mapa.

É possível otimizar esse projeto para que a aplicação tenha um menor processamento e utilização de memória que a versão atual. Para isso, pode-se reduzir o tamanho das texturas para o mínimo possível (sem perdas de

qualidade), além de mapeá-las em um único arquivo de imagem denominado Atlas. A geração de terrenos de forma procedural é outro método que influencia bastante no tamanho da aplicação, reduzindo-a de forma considerável. Desabilitar os objetos que não estão ao alcance do usuário no momento é outra forma de otimização. A versão PRO da *engine* realiza algumas dessas otimizações automaticamente de forma que o desenvolvedor não precise se preocupar com tais detalhes. Essas otimizações não foram aplicadas a esse projeto, pois o foco foi o desenvolvimento dos terrenos.

Participei do evento Unite Brasil 2014 em São Paulo, onde houve uma promoção da empresa Intel, na qual quem conseguisse fazer funcionar um jogo desenvolvido utilizando a *engine* Unity 3D nas arquiteturas ARM e x86 em um *tablet* com processador da Intel (utilizam arquitetura x86) e chegar na conclusão que o apk gerado para esta arquitetura é mais eficiente que a outra.

Com os conhecimentos obtidos na elaboração do projeto deste trabalho foi possível participar de uma promoção da Intel no evento Unite Brasil 2014 em São Paulo. A promoção exigia o funcionamento de um jogo construído por meio da *engine* Unity 3D nas arquiteturas ARM e x86 em um *tablet* com processador Intel de arquitetura x86 para que se chegasse a uma conclusão que o apk gerado para a arquitetura x86 é mais eficiente nesse caso. A aplicação descrita neste trabalho foi testada e consegui ganhar um tablet como prêmio.

6 REFERÊNCIAS BIBLIOGRÁFICAS

GREGORY, J. **Game Engine Architecture**. Wellesley: A K Peters, Ltd., 2009.

LEE, W. M. **Beginning Android™ Application Development**. Indianapolis: Wiley Publishing, Inc, 2011. 450 p.

MASTROCOLA, V.M. **Ludificador**: um guia de referências para o game designer brasileiro. São Paulo: Independente, 2012.

PASSOS E. B.; SILVA- JR. J. R.; RIBEIRO, F. E. C.; MOURÃO, P. T. **Desenvolvimento de Jogos com Unity 3D**. VIII Brazilian Symposium on Games and Digital Entertainment Rio de Janeiro, 2009.

PASSOS, T. S. **Android, Arquitetura e Desenvolvimento**. Trabalho apresentado à disciplina de Trabalho de Diplomação da Pontifícia Universidade Católica de Minas Gerais, Poços de Caldas, 2009.

ROLLINGS, A.; MORRIS, D. **Game Architecture and Design**: a new edition. New York: New Riders, 2003.

7 APÊNDICE A

Neste apêndice é apresentado alguns códigos em *Javascript* desenvolvidos neste projeto.

7.1 Menu Principal

Neste *script* os *buttons* e os *labels* são criados, posicionados, adaptados ao tamanho da tela na cena Menu principal.

Código:

```
#pragma strict

var style : GUIStyle;

function OnGUI () {

    var larguraBotao = Screen.width * 0.49;
    var larguraCentralizada = Screen.width / 2 - larguraBotao / 2;
    var alturaBotao = Screen.height * 0.15;

    var buttonStyle= new GUIStyle("Button");
    var Multiplicador: float = 0.06;

    var MultiplicadorLabel: float = 0.06;

    style.fontSize = Screen.width * MultiplicadorLabel;

    buttonStyle.fontSize = Screen.width * Multiplicador;
    buttonStyle.normal.textColor = Color.white;
    buttonStyle.alignment = TextAnchor.MiddleCenter;

    GUI.Label(Rect(0, Screen.height/15, Screen.width,
Screen.height/10),"Projeto de Monografia", style);

    GUI.Label(Rect(0, Screen.height * 0.2, Screen.width,
Screen.height/10),"Giovanni Viol Assis", style);
```

```

        if (GUI.Button(Rect(larguraCentralizada, Screen.height * 0.4,
larguraBotao, alturaBotao), "Seleção de cenas", buttonStyle)){
            Application.LoadLevel("Seleção de cenas");
        }

        if (GUI.Button(Rect(larguraCentralizada, Screen.height * 0.6,
larguraBotao, alturaBotao), "Créditos", buttonStyle)){

            Application.LoadLevel("Créditos");
        }

        if (GUI.Button(Rect(larguraCentralizada, Screen.height * 0.8,
larguraBotao, alturaBotao), "Sair", buttonStyle)){
            Application.Quit();
        }
    }
}

```

7.2 Créditos

Este é responsável pela criação, posicionamento, adaptação ao tamanho da tela e exibição dos *labels* e *buttons* referentes à cena dos créditos.

Código:

```

#pragma strict

var style : GUIStyle;

function OnGUI(){

    var larguraBotao = Screen.width * 0.42;
    var larguraCentralizada = Screen.width / 2 - larguraBotao / 2;
    var alturaBotao = Screen.height * 0.15;

    var buttonStyle= new GUIStyle("Button");
    var Multiplicador: float = 0.06;
    var MultiplicadorLabel: float = 0.06;

    style.fontSize = Screen.width * MultiplicadorLabel;

    buttonStyle.fontSize = Screen.width * Multiplicador;

```

```

        buttonStyle.normal.textColor = Color.white;
        buttonStyle.alignment = TextAnchor.MiddleCenter;

        GUI.Label(Rect(0, 0, Screen.width, Screen.height/10), "Projeto
de Monografia", style);

        GUI.Label(Rect(0, Screen.height * 0.15, Screen.width,
Screen.height/10), "UFLA - Ciência da Computação", style);

        GUI.Label(Rect(0, Screen.height * 0.3, Screen.width,
Screen.height/10), "Prof. Orientador: Joaquim Uchôa", style);

        GUI.Label(Rect(0, Screen.height * 0.45, Screen.width,
Screen.height/10), "Desenvolvedor: Giovanni Viol Assis", style);

        GUI.Label(Rect(0, Screen.height * 0.6, Screen.width,
Screen.height/10), "Contato: giubravo@yahoo.com.br", style);

        if(GUI.Button(Rect(larguraCentralizada, Screen.height * 0.8,
larguraBotao, alturaBotao),"Menu Principal", buttonStyle)){
            Application.LoadLevel("Menu Principal");
        }
    }
}

```

7.3 Seleção de Cenas

Este script cria, posiciona e adapta os *labels* e *buttons* referentes à seleção de cenas.

Código:

```

#pragma strict

var imagemIlha : Texture2D;

var imagemDeserto : Texture2D;

var imagemFloresta : Texture2D;

var imagemMontanhas : Texture2D;

```

```

var style1 : GUIStyle;

var style2 : GUIStyle;

var carregando : boolean = false;

var paginal : boolean = true;

function OnGUI() {

    var larguraBotao1 = Screen.width * 0.4;
    var larguraCentralizada1 = Screen.width / 2 -
larguraBotao1 / 2;
    var alturaBotao1 = Screen.height * 0.11;

    var larguraBotao2 = Screen.width * 0.5;
    var larguraCentralizada2 = Screen.width / 2 -
larguraBotao2 / 2;
    var alturaBotao2 = Screen.height * 0.24;

    var buttonStyle= new GUIStyle("Button");
var Multiplicador: float = 0.06;
    var MultiplicadorLabel: float = 0.06;

    style1.fontSize = Screen.width * MultiplicadorLabel;
    style2.fontSize = Screen.width * MultiplicadorLabel;

    buttonStyle.fontSize = Screen.width * Multiplicador;
    buttonStyle.normal.textColor = Color.white;
    buttonStyle.alignment = TextAnchor.MiddleCenter;

    gameObject.Find("Carregando").guiText.fontSize =
Screen.width * Multiplicador;

    GUI.Label(Rect(0, 0, Screen.width, Screen.height/10),
"Projeto de Monografia", style1);

    GUI.Label(Rect(0, Screen.height * 0.11, Screen.width,
Screen.height/10), "Giovanni Viol Assis", style1);

    if(carregando == false){

```

```

        if(paginal == true){
            if (GUI.Button(Rect(Screen.width * 0.03,
Screen.height * 0.3,Screen.height * 0.8, Screen.height * 0.42),
imagemIlha)){

                carregando = true;
                gameObject.Find("Fundo
escuro").guiTexture.enabled = true;

                gameObject.Find("Carregando").guiText.enabled = true;

                Application.LoadLevel("Ilha");
            }

            if (GUI.Button(Rect(Screen.width * 0.52,
Screen.height * 0.3,Screen.height * 0.8, Screen.height * 0.42),
imagemDeserto)){

                carregando = true;
                gameObject.Find("Fundo
escuro").guiTexture.enabled = true;

                gameObject.Find("Carregando").guiText.enabled = true;

                Application.LoadLevel("Deserto");
            }

            GUI.Label(Rect(Screen.width * 0.2,
Screen.height * 0.71, Screen.width * 0.4, Screen.height/10), "Ilha",
style2);

            GUI.Label(Rect(Screen.width * 0.65,
Screen.height * 0.71, Screen.width * 0.4, Screen.height/10),
"Deserto", style2);

            if(GUI.Button(Rect(larguraCentralizada1,
Screen.height * 0.83, larguraBotao1, alturaBotao1),"Menu Principal",
buttonStyle)){

                Application.LoadLevel("Menu Principal");
            }

            GUI.Label(Rect(Screen.width * 0.8, Screen.height
* 0.83, Screen.width * 0.4, Screen.height/10), "1/2", style2);

```

```

        if(GUI.Button(Rect(Screen.width * 0.89,
Screen.height * 0.83, Screen.width * 0.08, alturaBotao1), ">",
buttonStyle)){
            paginal = false;
        }

    } else if(paginal == false){

        if (GUI.Button(Rect(Screen.width * 0.03,
Screen.height * 0.3,Screen.height * 0.8, Screen.height * 0.42),
imagemFloresta)){

            carregando = true;
            gameObject.Find("Fundo
escuro").guiTexture.enabled = true;

            gameObject.Find("Carregando").guiText.enabled = true;

            Application.LoadLevel("Floresta");
        }

        if (GUI.Button(Rect(Screen.width * 0.52,
Screen.height * 0.3,Screen.height * 0.8, Screen.height * 0.42),
imagemMontanhas)){

            carregando = true;
            gameObject.Find("Fundo
escuro").guiTexture.enabled = true;

            gameObject.Find("Carregando").guiText.enabled = true;

            Application.LoadLevel("Montanhas");
        }

        GUI.Label(Rect(Screen.width * 0.15,
Screen.height * 0.71, Screen.width * 0.4, Screen.height/10),
"Floresta", style2);

        GUI.Label(Rect(Screen.width * 0.6,
Screen.height * 0.71, Screen.width * 0.4, Screen.height/10),
"Montanhas", style2);

```



```

        if(GUI.Button(Rect(larguraCentralizada1,
Screen.height * 0.83, larguraBotao1, alturaBotao1),"Menu Principal",
buttonStyle)){
            Application.LoadLevel("Menu Principal");
        }

        GUI.Label(Rect(Screen.width * 0.8, Screen.height
* 0.83, Screen.width * 0.4, Screen.height/10), "2/2", style2);

        if(GUI.Button(Rect(Screen.width * 0.89,
Screen.height * 0.83, Screen.width * 0.08, alturaBotao1),"<",
buttonStyle)){
            paginal = true;
        }
    }

} else if(carregando == true){

        if(GUI.Button(Rect(larguraCentralizada2, Screen.height *
0.73, larguraBotao2, alturaBotao2),"Voltar para\nSeleção de Cenas",
buttonStyle)){ //\n para pular linha.
            Application.LoadLevel("Seleção de
Cenas");
        }
    }
}
}

```

7.4 Pause

Neste *script* um *button* de *pause* é criado, posicionado, adaptado ao tamanho da tela e mostrado no canto superior direito nas cenas dos terrenos quando ele é acessado. A aplicação pausa e surge uma imagem transparente e escura no fundo e os *buttons* “Continuar”, “Recomeçar”, “Seleção de cenas” e “Menu principal”.

Código:

```
#pragma strict
```

```

private var pauseGame : boolean = false;
private var showGUI : boolean = false;
private var showGUILabel : boolean = true;

var style : GUIStyle;

function OnGUI(){

    var larguraBotao = Screen.width * 0.5;
    var larguraCentralizada = Screen.width / 2 - larguraBotao / 2;
    var alturaBotao = Screen.height * 0.15;

    var buttonStyle= new GUIStyle("Button");
    var Multiplicador: float = 0.06;

    buttonStyle.fontSize = Screen.width * Multiplicador;
    buttonStyle.normal.textColor = Color.white;
    buttonStyle.alignment = TextAnchor.MiddleCenter;

    style.fontSize = Screen.width * 0.04;

    if(showGUILabel == true) {

        GUI.Label (Rect(0, Screen.height * 0.64, Screen.height *
0.3, Screen.height/10), "Movimente por aqui", style);

        GUI.Label (Rect(Screen.width * 0.57, Screen.height *
0.58, Screen.height * 0.3, Screen.height/10), "Movimente a câmera ou",
style);

        GUI.Label (Rect(Screen.width * 0.57, Screen.height *
0.64, Screen.height * 0.3, Screen.height/10), "Dê dois taps para
pular", style);

    }

    if(pauseGame == false && showGUILabel == true){
        if(GUI.Button(Rect(0, Screen.height/30, Screen.width *
0.45, Screen.height * 0.11), "Desativar Dicas", buttonStyle)) {
            showGUILabel = false;
        }
    } else if(pauseGame == false && showGUILabel == false){

```

```

        if(GUI.Button(Rect(0, Screen.height/30, Screen.width *
0.35, Screen.height * 0.11), "Ativar Dicas", buttonStyle)) {
            showGUILabel = true;
        }
    }

    if(GUI.Button(Rect(Screen.width * 0.88, Screen.height/30,
Screen.width * 0.1, Screen.height/10), "II", buttonStyle))
    {
        pauseGame = !pauseGame;

        if(pauseGame == true)
        {
            showGUILabel = false;
            Time.timeScale = 0;
            showGUI = true;
        }
    }

    if(pauseGame == false)
    {
        Time.timeScale = 1;
        showGUI = false;
    }

    if(showGUI == true)
    {
        gameObject.Find("Fundo Pause").guiTexture.enabled = true;

        if(GUI.Button(Rect(larguraCentralizada, Screen.height * 0.1,
larguraBotao, alturaBotao), "Continuar", buttonStyle)){
            Time.timeScale = 1;
            pauseGame = false;
            showGUI = false;
        }

        if(GUI.Button(Rect(larguraCentralizada, Screen.height * 0.3,
larguraBotao, alturaBotao), "Recomeçar", buttonStyle)){
            Application.LoadLevel(Application.loadedLevel);
        }

        if(GUI.Button(Rect(larguraCentralizada, Screen.height * 0.5,
larguraBotao, alturaBotao), "Seleção de Cenas", buttonStyle)){

```

```

        Application.LoadLevel("Seleção de Cenas");
    }

    if(GUI.Button(Rect(larguraCentralizada, Screen.height * 0.7,
larguraBotao, alturaBotao), "Menu Principal", buttonStyle)){
        Application.LoadLevel("Menu Principal");
    }
}

else
{
    gameObject.Find("Fundo Pause").guiTexture.enabled = false;
}
}

```

7.5 Joystick

Este *script* executa o movimento do jogador: a movimentação no canto inferior esquerdo, a movimentação da câmera e a habilidade “pular” no canto inferior direito. Esse script já está contido no pacote *Standard Assets (Mobile)*, porém foi necessário alterá-lo para que as texturas se adaptassem ao tamanho da tela.

Código:

```

////////////////////////////////////
// Joystick.js
// Penelope iPhone Tutorial
//
// Joystick creates a movable joystick (via GUITexture) that
// handles touch input, taps, and phases. Dead zones can control
// where the joystick input gets picked up and can be normalized.
//
// Optionally, you can enable the touchPad property from the editor
// to treat this Joystick as a TouchPad. A TouchPad allows the finger
// to touch down at any point and it tracks the movement relatively
// without moving the graphic
////////////////////////////////////

#pragma strict

```

```

@script RequireComponent( GUITexture )

// A simple class for bounding how far the GUITexture will move
class Boundary
{
    var min : Vector2 = Vector2.zero;
    var max : Vector2 = Vector2.zero;
}

static private var joysticks : Joystick[];
    // A static collection of all joysticks
static private var enumeratedJoysticks : boolean = false;
static private var tapTimeDelta : float = 0.3;
    // Time allowed between taps

var touchPad : boolean;
    // Is this a TouchPad?
var touchZone : Rect;
var deadZone : Vector2 = Vector2.zero;
    // Control when position is output
var normalize : boolean = false;
    // Normalize output after the dead-zone?
var position : Vector2;
    // [-1, 1] in x,y
var tapCount : int;
    // Current tap count

private var lastFingerId = -1;
    // Finger last used for this joystick
private var tapTimeWindow : float;
    // How much time there is left for a tap to occur
private var fingerDownPos : Vector2;
private var fingerDownTime : float;
private var firstDeltaTime : float = 0.5;

private var gui : GUITexture;
    // Joystick graphic
private var defaultRect : Rect;
    // Default position / extents of the joystick
graphic
private var guiBoundary : Boundary = Boundary(); //
Boundary for joystick graphic

```

```

private var guiTouchOffset : Vector2;
        // Offset to apply to touch input
private var guiCenter : Vector2;
        // Center of joystick

function Start()
{

        // Cache this component at startup instead of looking up every
frame
        gui = GetComponent( GUITexture );

        //Modificado Por Giovanni Viol Assis

        gui.pixelInset.width = Screen.height * 0.3;
        gui.pixelInset.height = Screen.height * 0.3;

        gameObject.Find("RightTouchPad").guiTexture.pixelInset.position.x
= Camera.main.ScreenToWorldPoint(Vector2(Screen.width, 0)).x -
Screen.height * 0.3;

        // Store the default rect for the gui, so we can snap back to
it
        defaultRect = gui.pixelInset;

        defaultRect.x += transform.position.x * Screen.width;// +
gui.pixelInset.x; // - Screen.width * 0.5;
        defaultRect.y += transform.position.y * Screen.height;// -
Screen.height * 0.5;

        transform.position.x = 0.0;
        transform.position.y = 0.0;

        if ( touchPad )
        {
                // If a texture has been assigned, then use the rect
ferom the gui as our touchZone
                if ( gui.texture )
                        touchZone = defaultRect;
        }
        else

```

```

    {
        // This is an offset for touch input to match with the
top left
        // corner of the GUI
        guiTouchOffset.x = defaultRect.width * 0.5;
        guiTouchOffset.y = defaultRect.height * 0.5;

        // Cache the center of the GUI, since it doesn't change
        guiCenter.x = defaultRect.x + guiTouchOffset.x;
        guiCenter.y = defaultRect.y + guiTouchOffset.y;

        // Let's build the GUI boundary, so we can clamp
joystick movement
        guiBoundary.min.x = defaultRect.x - guiTouchOffset.x;
        guiBoundary.max.x = defaultRect.x + guiTouchOffset.x;
        guiBoundary.min.y = defaultRect.y - guiTouchOffset.y;
        guiBoundary.max.y = defaultRect.y + guiTouchOffset.y;
    }
}

function Disable()
{
    gameObject.SetActive(false);
    enumeratedJoysticks = false;
}

function ResetJoystick()
{
    // Release the finger control and set the joystick back to the
default position
    gui.pixelInset = defaultRect;
    lastFingerId = -1;
    position = Vector2.zero;
    fingerDownPos = Vector2.zero;

    if ( touchPad )
        gui.color.a = 0.025;
}

function IsFingerDown() : boolean
{
    return (lastFingerId != -1);
}

```

```

function LatchedFinger( fingerId : int )
{
    // If another joystick has latched this finger, then we must
    release it
    if ( lastFingerId == fingerId )
        ResetJoystick();
}

function Update()
{
    if ( !enumeratedJoysticks )
    {
        // Collect all joysticks in the game, so we can relay
        finger latching messages
        joysticks = FindObjectsOfType( Joystick ) as Joystick[];
        enumeratedJoysticks = true;
    }

    var count = Input.touchCount;

    // Adjust the tap time window while it still available
    if ( tapTimeWindow > 0 )
        tapTimeWindow -= Time.deltaTime;
    else
        tapCount = 0;

    if ( count == 0 )
        ResetJoystick();
    else
    {
        for( var i : int = 0; i < count; i++)
        {
            var touch : Touch = Input.GetTouch(i);

            var guiTouchPos : Vector2 = touch.position -
guiTouchOffset;

            var shouldLatchFinger = false;
            if ( touchPad )
            {
                if ( touchZone.Contains( touch.position )
)

```



```

        shouldLatchFinger = true;
    }
    else if ( gui.HitTest( touch.position ) )
    {
        shouldLatchFinger = true;
    }

    // Latch the finger if this is a new touch
    if ( shouldLatchFinger && ( lastFingerId == -1
|| lastFingerId != touch.fingerId ) )
    {

        if ( touchPad )
        {
            gui.color.a = 0.15;

            lastFingerId = touch.fingerId;
            fingerDownPos = touch.position;
            fingerDownTime = Time.time;
        }

        lastFingerId = touch.fingerId;

        // Accumulate taps if it is within the
time window

        if ( tapTimeWindow > 0 )
            tapCount++;
        else
        {
            tapCount = 1;
            tapTimeWindow = tapTimeDelta;
        }

        // Tell other joysticks we've latched
this finger

        for ( var j : Joystick in joysticks )
        {
            if ( j != this )
                j.LatchedFinger(
touch.fingerId );
        }
    }

```

```

    }

    if ( lastFingerId == touch.fingerId )
    {
        // Override the tap count with what the
iPhone SDK reports if it is greater
        // This is a workaround, since the iPhone
SDK does not currently track taps
        // for multiple touches
        if ( touch.tapCount > tapCount )
            tapCount = touch.tapCount;

        if ( touchPad )
        {
            // For a touchpad, let's just set
the position directly based on distance from initial touchdown
            position.x = Mathf.Clamp( (
touch.position.x - fingerDownPos.x ) / ( touchZone.width / 2 ), -1, 1
);

            position.y = Mathf.Clamp( (
touch.position.y - fingerDownPos.y ) / ( touchZone.height / 2 ), -1, 1
);

        }
        else
        {
            // Change the location of the
joystick graphic to match where the touch is
            gui.pixelInset.x = Mathf.Clamp(
guiTouchPos.x, guiBoundary.min.x, guiBoundary.max.x );
            gui.pixelInset.y = Mathf.Clamp(
guiTouchPos.y, guiBoundary.min.y, guiBoundary.max.y );
        }

        if ( touch.phase == TouchPhase.Ended ||
touch.phase == TouchPhase.Canceled )
            ResetJoystick();
    }
}

if ( !touchPad )
{

```

```

        // Get a value between -1 and 1 based on the joystick
        graphic location
        position.x = ( gui.pixelInset.x + guiTouchOffset.x -
guiCenter.x ) / guiTouchOffset.x;
        position.y = ( gui.pixelInset.y + guiTouchOffset.y -
guiCenter.y ) / guiTouchOffset.y;
    }

    // Adjust for dead zone
    var absoluteX = Mathf.Abs( position.x );
    var absoluteY = Mathf.Abs( position.y );

    if ( absoluteX < deadZone.x )
    {
        // Report the joystick as being at the center if it is
        within the dead zone
        position.x = 0;
    }
    else if ( normalize )
    {
        // Rescale the output after taking the dead zone into
        account
        position.x = Mathf.Sign( position.x ) * ( absoluteX -
deadZone.x ) / ( 1 - deadZone.x );
    }

    if ( absoluteY < deadZone.y )
    {
        // Report the joystick as being at the center if it is
        within the dead zone
        position.y = 0;
    }
    else if ( normalize )
    {
        // Rescale the output after taking the dead zone into
        account
        position.y = Mathf.Sign( position.y ) * ( absoluteY -
deadZone.y ) / ( 1 - deadZone.y );
    }
}

```

8 APÊNDICE B

Este apêndice é referente ao questionário utilizado para realizar os testes com usuários. Foi desenvolvido no site Survey Monkey.

Título: Testes com Usuários da aplicação Projeto de Monografia do Giovanni

Página 1: Dispositivo e sistema utilizados para este teste.

Descrição: Esta página é referente a qual dispositivo e sistema foram utilizados para a realização deste teste.

Questões:

1) Qual a marca do dispositivo que foi utilizado neste teste?

2) Qual o modelo?

3) Qual a versão do Android?

Página 2: Instalação do aplicativo no dispositivo Android.

Descrição: Esta página é para o feedback sobre o download da aplicação e da instalação do .apk no dispositivo Android.

Questões:

4) Conseguiu baixar o aplicativo para seu dispositivo com Android?

Sim

Não

5) Conseguiu baixar o aplicativo para seu dispositivo com Android?

Sim

Não

6) Conseguiu abrir a aplicação após a instalação?

Sim

Não

Página 3: Interface e Jogabilidade.

Descrição: Questões referente a interface e jogabilidade da aplicação.

Questões:

7) Avalie a interface das seguintes telas:

Tabela 4: Enunciado da questão 7 do formulário dos testes com usuários.

	Muito ruim	Ruim	Médio	Bom	Muito bom
Menu Principal	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Créditos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Seleção de Cenas 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Seleção de Cenas 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

8) Avalie a interface dos seguintes terrenos:

Tabela 5: Enunciado da questão 8 do formulário dos testes com usuários.

	Muito ruim	Ruim	Médio	Bom	Muito bom
Ilha	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deserto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Floresta	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Montanha	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

9) Qual o desempenho dos seguintes terreno:

Tabela 6: Enunciado da questão 9 do formulário dos testes com usuários.

	Muito pesado	Pesado	Médio	Leve	Muito leve
Ilha	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deserto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Floresta	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Montanha	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

10) Você jogaria um Jogo de RPG com estes terrenos?

Sim

Não