

UNIVERSIDADE ESTADUAL PAULISTA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO

João de Carvalho Milone

Mayan Shooter
(Aplicação de RA)

Bauru - SP
2017

RESUMO

Por meio da *engine Unity* associada à plataforma de RA (Realidade Aumentada) *Vuforia*, foi desenvolvido um projeto de tiro ao alvo com o uso de dois marcadores (um para a arma e outro para os alvos). Para os alvos foi construído um cenário 3D com temática Maia onde os alvos se encontram em destaque para o jogador. Para a arma foi usado um modelo de uma escopeta junto com um laser para auxiliar na mira, acompanhados de um botão virtual que efetuará os disparos. O objetivo é alcançar a maior pontuação em 30 segundos, tentando acertar o maior número de barris possíveis. Conseguindo-se acertar os barris na lava, tem-se um bônus de pontos.

SUMÁRIO

1. <i>Assets Usados</i>	3
1.1. Cenário.....	3
1.2. Alvos.....	4
1.3. Arma.....	4
1.4. UI.....	5
2. <i>Scripts</i>	6
2.1. <i>BalaScript</i>	6
2.2. Contador.....	6
2.3. <i>GunScript</i>	6
2.4. <i>HighScoreScript</i>	6
2.5. <i>LataScript</i>	6
2.6. <i>ScoreMenu</i>	7
2.7. <i>ScreenLoader</i>	7
2.8. <i>VirtualButtonScript</i>	7
3. <i>Aplicação</i>	8
3.1. Tela Inicial.....	8
3.2. Tela Jogo.....	8
3.3. Tela <i>GameOver</i>	8
4. <i>Bibliografia</i>	10

1. ASSETS USADOS

1.1. Cenário:

Em cima do *ImageTarget* dos alvos foi montado um cenário 3D usando modelos de um pack Maia. *BoxColliders* foram incluídos nessas estruturas para que colisões com outros objetos do jogo fossem possíveis. E por baixo foi incluso um rio de lava com a *tag* de “lava” para que em meio a colisões um objeto pudesse ser destruído ao encostar nela e atribuir mais pontos para o jogador.

Pedras foram colocadas em volta do cenário para cobrir a borda da lava dando maior credibilidade para um cenário vulcânico com temática Maia.

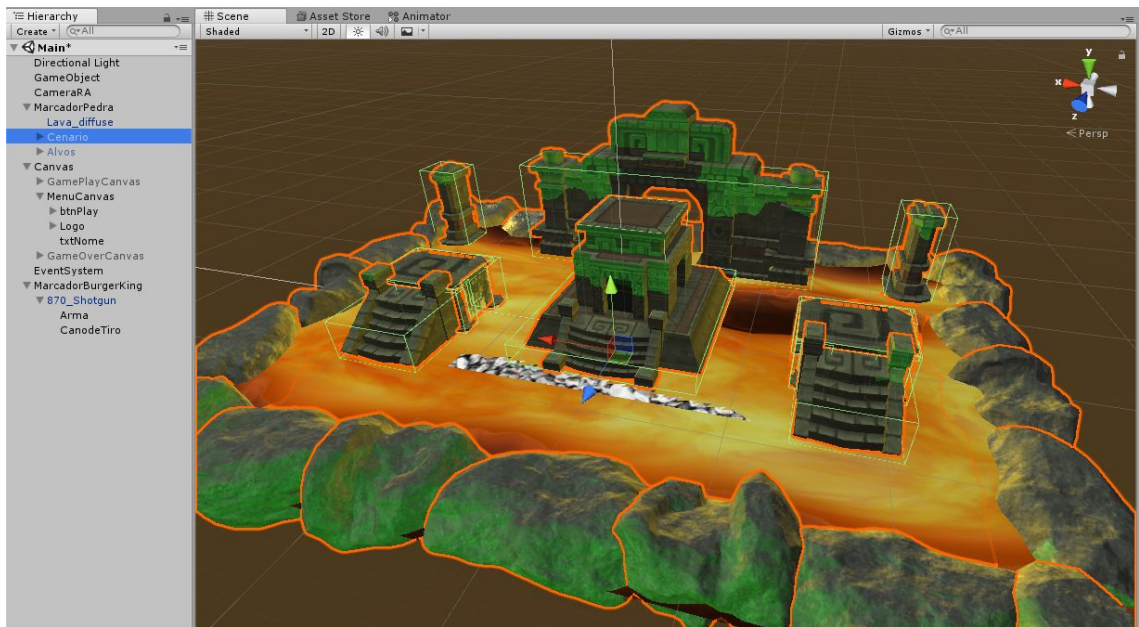


Imagem do cenário no *Unity*.

1.2. Alvos:

Barris foram usados como alvos, sendo atribuídos a eles, uma textura mais chamativa. Foi adotado um *script* padrão para os alvos. Por meio deste *script*, verifica-se a colisão foi por um projétil ou outro barril, para que assim o jogador possa ser recompensado com um bônus de pontos. Além disso, faz-se uma verificação de que o barril entrou em contato com a lava, para que assim ele possa ser destruído com o instanciamento de uma explosão e fumaça no local. O *script* também atribui um aumento e diminuição do tamanho dos barris em tempo real, para dar um maior destaque para eles no cenário.

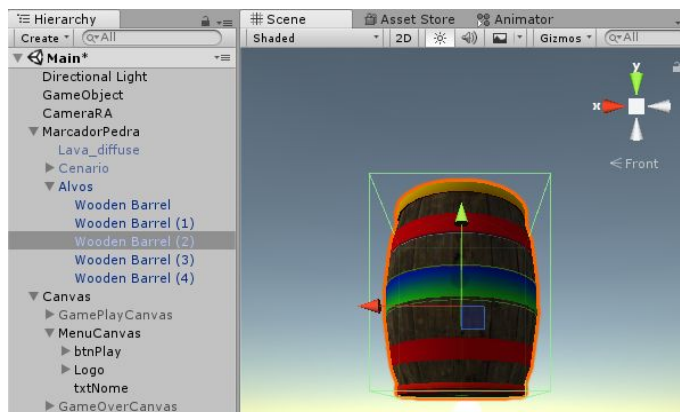


Imagem do barril usado como alvo no *Unity*.

1.3. Arma:

Foi utilizado um marcador menor e a arma foi rotacionada, para que fosse possível atirar com o marcador na vertical para a câmera, aprimorando o rastreamento da mesma. Para a arma foi usada uma textura vermelha para que ficasse chamativa na tela do jogo, como também foi atribuído um objeto que renderiza um laser que auxilia na mira e a ele uma fonte de luz verde foi inclusa para dar uma maior interação entre o laser e o cenário. Na ponta da arma, foi atribuído o instanciador de balas que lança projéteis, por meio do script *GunScript* que permite que o jogador atire e um efeito de explosão na ponta da arma seja acionado. É permitido dar um tiro a cada dois segundos. Durante esse intervalo, um som de carregamento é acionado junto com uma animação característica de uma escopeta.

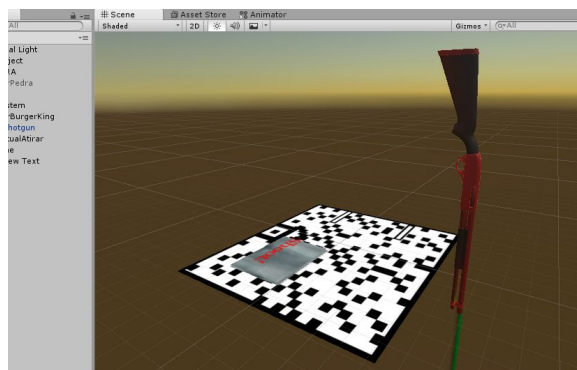


Imagem da arma atribuída ao seu marcador no *Unity*.

1.4. UI:

Foram criados 3 telas criadas para o jogador:

1.4.1. Tela Inicial:

Inclui o nome do jogo, recorde atual do jogador, e um botão com a opção de começar uma partida. Por meio do *click* deste botão, desativa-se o menu principal e ativar a tela de jogo.

1.4.2. Tela de Jogo:

Dá a opção de atirar com a arma com um botão virtual no marcador da arma, voltar para o menu e recomeçar uma partida. Tem informações cruciais como um cronômetro para saber quantos segundos ainda restam, como também disponibiliza os pontos atuais. Logo após os 30 segundos do jogo acabarem, a tela irá automaticamente mudar para a tela de *GameOver*.

1.4.3. Tela de *GameOver*:

Mostra o quanto de pontos foram obtidos pelo jogador, e faz uma comparação automática com o recorde anterior para ver se ele foi quebrado. Desta forma, uma nova pontuação é registrada. A tela de *GameOver* também possui opções de voltar para o menu ou começar uma nova partida.



2. SCRIPTS

No total foram usados 8 *scripts* nos objetos:

2.1. *BalaScript*:

Atribuído aos projéteis. O *script* verifica se houve colisão com os barris ou lava, para que ele possa ser destruído e uma explosão com fumaça possa ocorrer se tiver entrado em contato com a lava. Apenas o *GameObject* de explosão é público nesse *script*. Para que assim ele seja instanciado quando em contato com a lava.

2.2. *Contador*:

Atribuído ao objeto tipo *Text* na tela de Jogo. É utilizado para manter o registro de pontos do jogador, como também realizar a contagem regressiva, e ao final dela poder levar o jogador para a tela de *GameOver*. Nesse *script* são puxados como objetos públicos o *txtPontos* para que nele os pontos possam ser atualizados para o jogador, os barris para o início de uma nova partida. O *Canvas* atual e as telas de *GameOver* e a tela atual, para que ele possa desativar e ativar as telas quando necessário.

2.3. *GunScript*:

Atribuído a um objeto vazio na ponta da arma. Utilizado para efetuar os disparos pela função pública *Shoot()*, que é ativada com o click de um botão virtual habilitado na tela de Jogo. Nessa função ocorre o instanciamento dos projéteis com uma dada velocidade atribuída e um método de recarga. O *script* inicia a animação de recarga, e toca o som de tiro e carregamento da arma. Como objetos públicos são requisitados, o *GameObject* projétil, a explosão, arquivo de áudio do tiro, emissor de áudio para o tiro e o gerenciador de animação da arma.

2.4. *HighScoreScript*:

Usado para verificar se os pontos registrados no *Contador* ao final de cada partida são maiores que o *HighScore* anterior. O recorde fica registrado como valor inteiro em um *PlayerPrefs*.

2.5. *LatasScript*:

Atribuído a todos os alvos. Este *script* verifica se houve colisão com algum projétil, lava ou outro barril, para recompensar o jogador com pontos, como também atribui uma animação que interfere na *scale* do objeto por meio de uma função seno em tempo real. O *script* também instancia um *hitmarker* que irá dar um *feedback* visual para o jogador de que houve um acerto, junto com um efeito sonoro e vibração do aparelho. Como objetos públicos temos uma explosão para ser instanciada ao contato com a lava, a imagem do *hitmarker* e o efeito sonoro do mesmo.

2.6. *ScoreMenu*:

Atribuído ao objeto do tipo *Text* no menu inicial. É usado para carregar o recorde atual do jogador como informação.

2.7. *ScreenLoader*:

Atribuído a um objeto vazio na tela de *loading*. Usado para dar um *feedback* visual do carregamento do jogo por meio de uma barra de *loading*. Utiliza dois objetos públicos para atribuir a eles o *feedback* visual do progresso de carregar, sendo eles um *Slider* que funciona como barra de carregamento, e um *Text* que irá mostrar a porcentagem do carregamento em números.

2.8. *VirtualButtonScript*:

Utilizado no *ImageTarget* da arma, ele é usado para simular botões virtuais no marcador e registrar ações de *ButtonPressed* e *ButtonReleased*. Como objeto publico somente é necessário o *VirtualButton* usado no *ImageTarget* da arma.

4. APLICAÇÃO

Precisa-se, então, desenvolver uma aplicação para smartphone gerada em um *apk*, para a instalação em um aparelho com *Android* (no mínimo versão 4.3 '*Jelly Bean*'). Com as devidas telas de execução.

4.1. Tela Inicial

- Logo do jogo
- Recorde anterior
- Botão *Play*
- Informação do desenvolvedor

4.2. Tela Jogo

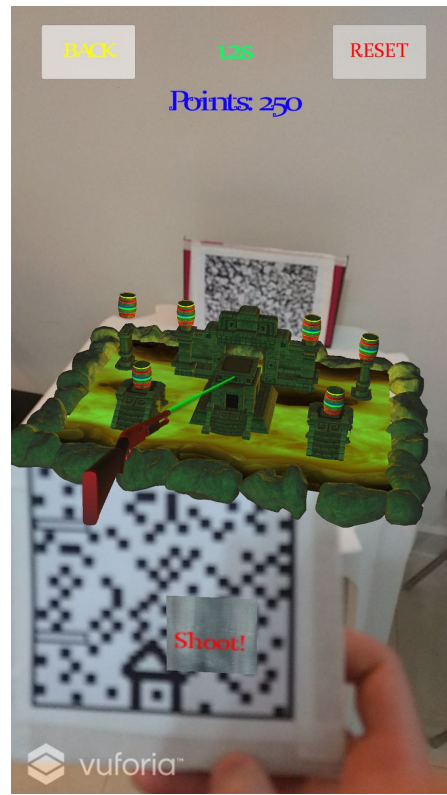
- Botão voltar para o menu
- Botão recomeçar jogo
- Botão *Shoot* virtual no marcador da arma
- Texto do cronômetro
- Texto pontuação

4.3. Tela *GameOver*

- Botão voltar para o menu
- Botão recomeçar jogo
- Texto do atual recorde
- Texto pontuação do jogo



Tela Inicial.



Tela Jogo.



Tela GameOver.

4. BIBLIOGRAFIA

4.1. HOLLMANN, Matias. *Modern Weapons*, DevAssets.

<http://devassets.com/assets/modern-weapons/>

4.2. BITGEMS. *Mayan Temple*, DevAssets.

<http://devassets.com/assets/mayan-temple/>

4.3. ALLEN, Doug. *Western Props Pack*, DevAssets.

<http://devassets.com/assets/western-props-pack/>

4.4. UNITY. *Unity Particle Pack*, Unity Asset Store.

<https://www.assetstore.unity3d.com/en/#!/content/73777>

4.4. CARNIVORE, Moonflower. *Lava Flowing Shader*, Unity Asset Store.

<https://www.assetstore.unity3d.com/en/#!/content/33635>

4.4. R&P studios. *Rocks HD Pack*, Unity Asset Store.

<https://www.assetstore.unity3d.com/en/#!/content/81901>