

MANUAL

A REALIDADE AUMENTADA ATRAVÉS DE BLENDER | UNITY | VUFORIA

ÁTILA PEREZ | THALES PANKE

A mescla de conteúdos virtuais com o mundo real ganha, cada vez mais, popularidade, em vista de suas possibilidades e benefícios quando considerada a questão da interpretação visual humana. A Realidade Aumentada é muito classificada como uma ferramenta de visualização que tende a auxiliar no âmbito de interpretação e visualização de determinadas ideias dentro de um devido contexto. Atualmente, este recurso é muito utilizado auxiliando nos mais variados meios, desde em materiais didáticos, atividades educativas para crianças, jogos, ou até mesmo no seu dia-a-dia, em dispositivos de interação, como em aplicativos de celular.

Em vista disso, neste trabalho, será demonstrado por meio de um manual, um método efetivo e prático para a criação de materiais que utilizem a Realidade Aumentada em sua interface. Desta forma, serão possíveis criações de múltiplas aplicações desta ferramenta, com intuito de promover uma contribuição no âmbito da aprendizagem que seja prática e eficaz, no sentido do entendimento de assuntos complexos dentro de sala de aula. Desta forma, este processo se dará a partir de modelagens no *software* gratuito *Blender 3D*, no entanto, as modelagens desejadas podem ser produzidas por meio de outros programas de modelagem, tais como *Autodesk Revit*, *SketchUp*, *Autodesk 3DS Max*, etc., que deverão ser exportados em formatos específicos.

Para a realização deste tutorial, serão necessários os seguintes requisitos mínimos listados a seguir:

- 1) Um computador que possua no mínimo:
 - Windows 7 SP1 ou Windows 8 / 8.1 de 64 bits
 - Processador Intel Core i5-2500K @ 3.3GHz ou AMD FX-8350 @ 4.0GHz

- Placa Gráfica: NVIDIA GeForce GTX 680 ou AMD Radeon HD 7970 (2 GB VRAM)

- Memória de 6GB RAM
- Conexão com a Internet

2) Os seguintes *softwares* e kits baixados e preparados para instalação:

- *Blender 3D*
- *Unity 3D*
- *Android SDK*
- *Java JDK*

O funcionamento deste projeto se dá a partir da importação dos arquivos de modelagem realizados no *Blender 3D*, ao motor gráfico de jogos *Unity 3D* que era emular o aplicativo em Realidade Aumentada para o dispositivo desejado. O *Unity 3D* atualmente possui suporte para *Vuforia SDK* que será responsável pela importação de conteúdos necessários para o funcionamento do aplicativo final. Além disso, é necessário ter instalado no computador, o *software Android Studio* que será responsável por fornecer a compatibilidade do aplicativo de acordo com o sistema operacional *Android* desejado.

- **PASSO 1 – ANDROID STUDIO E JAVA SDK**

Para que o aplicativo final possa ser compilado em um único arquivo executável para *Android*, é necessário que sejam baixadas no computador as versões SDK das versões correspondentes dos celulares a serem utilizados. Para isso, basta que seja instalado o ambiente de desenvolvimento para a plataforma *Android*, *Android Studio*, fornecido gratuitamente através do endereço www.developer.android.com/studio. A partir de “*Configure*>*SDK Manager*” disponibilizado pelo *software*, será possível que este programa instale automaticamente no computador as versões SDKs das versões dos *Androids* escolhidos pelo usuário, passo que se fará muito importante no fim deste manual.

Assim que instalados as versões SDKs correspondentes aos celulares que executarão o aplicativo, será necessário que seja baixado o JDK (*Java Development Kit*) que também poderá ser adquirido gratuitamente, desta vez, através do endereço www.oracle.com/technetwork/pt/java/javase/downloads/jdk8-downloads-2133151.html. Este será necessário para que o *Unity 3D* reconheça posteriormente as configurações necessárias para a compilação do *software* que funcione dentro de um dispositivo móvel. Basta instalar o executável e seguir para o próximo passo.

- **PASSO 2 – BLENDER: INSTALAÇÃO E EXPORTAÇÃO**

Neste passo, faz-se necessário o *download* do programa de modelagem e animação gratuito, *Blender3D*, podendo ser facilmente encontrado através do site oficial da empresa, www.blender.org. A instalação é simples e intuitiva, portanto, foi decidido que não era necessária a inserção deste passo neste manual. Com a instalação realizada, basta iniciar o programa, cuja interface abrirá automaticamente um novo projeto. Neste caso, o objeto escolhido foi uma caixa de madeira, um objeto simples e intuitivo, que pode ser facilmente realizado a partir do programa Blender, por meio de qualquer versão disponível.

Desta forma, este tutorial inicia-se pela modelagem de um objeto genérico que seja meramente ilustrativo para o funcionamento do dispositivo de Realidade Aumentada. Neste caso, será utilizada a caixa padrão fornecida pelo programa quando aberto que poderá ter como material, uma madeira semelhante às de jogos de videogame como *Crash Bandicoot*. A texturização no *Blender 3D* é realizada por *unwrap* (desdobramento do inglês), neste caso, sendo uma função necessária para que a textura se encaixe perfeitamente no objeto quando exportado para outros programas.

Entretanto, a caixa escolhida possui várias versões já desdobradas disponíveis *on-line*, desta forma, será utilizada uma textura escolhida objetivamente que sirva de base para que a exportação para o *Unity 3D* seja sucedida. Desta forma, a partir do *software Blender 3D*, utilizou-se a caixa padrão disponível na inicialização do aplicativo, sendo aplicada a uma textura correspondente à da FIGURA 1 abaixo. Desta forma, o produto foi salvo em formato *blend*.

FIGURA 1 – Textura utilizada para a criação da caixa



Fonte: O autor

É importante que seja salva a imagem da FIGURA 1 separadamente da textura, para que seja posteriormente importada para o *software Unity 3D*, onde as texturas deverão ser reconfiguradas de acordo.

- **PASSO 3 – INSTALAÇÃO DO UNITY 3D E APRESENTAÇÃO**

Com a caixa e o material já aplicados, basta salvar o arquivo em formato *blend* em uma pasta qualquer desejada, juntamente com a textura desdobrada dentro do *Blender 3D*.

Com o objeto finalizado, é hora de realizar a importação do arquivo de modelagem para o motor gráfico do *Unity3D*. A instalação do *Unity3D* é um processo simples, mas que exige certa atenção a alguns pontos. O instalador pode ser

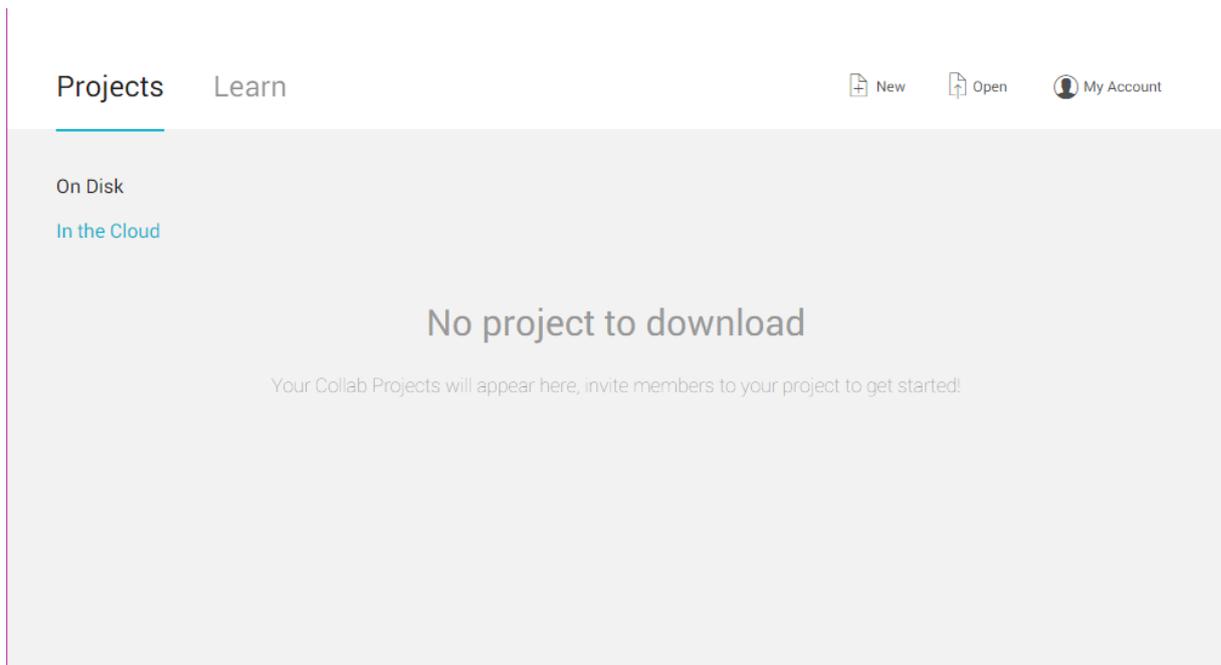
facilmente obtido através do site oficial da empresa, www.unity3d.com. Após o *download* do instalador do programa, deve-se abrir o executável como administrador e seguir os passos a seguir:

Assim que o executável iniciar, será aberta uma janela do assistente de instalação, notificando o usuário que este instalador necessita da utilização da *internet* para seu funcionamento. Clicando em *Next*, o assistente lidera o usuário a uma janela contendo o termo de compromisso necessário para a utilização do *software*. Concordando com o termo, basta clicar em *Next*. Feito este passo, é aberta uma nova janela lhe perguntando quais componentes deverão ser instalados na máquina. Neste caso, pede-se que sejam marcadas as caixas diálogo contendo os seguintes itens:

- ✓ *Unity* (a versão correspondente)
- ✓ *Documentation*
- ✓ *Android Build Support*
- ✓ *iOS Build Support*
- ✓ *Vuforia Augmented Reality Support*
- ✓ *Windows Build Support.*

Desta forma, o instalador providenciará a possibilidade de construir compilações para os devidos dispositivos listados (celulares com *Android*, *iOS*, *Windows Phone*, etc.). Clicando em *Next*, uma nova janela é aberta para a especificação do diretório a ser instalado o programa. Escolhido o diretório, basta aguardar a instalação. Após baixado e instalado, deve-se executar o programa e aguardar a tela de abertura do *Unity3D* surgir, algo parecido com a FIGURA 1.

FIGURA 2 – Janela de inicialização do *Unity3D*



Fonte: O autor.

Assim que aberta esta janela, basta clicar no ícone simbolizando um novo projeto (*New*) que irá direcionar o usuário a uma nova janela de especificação de criação de projeto. Nesta janela, são definidos o nome do projeto, qual *template* a ser utilizado dentro do motor, e a localização da pasta de arquivos que será criada. Após definidos a devida localização do arquivo e o nome do projeto, escolhe-se o *template* padrão, o 3D. Não será necessária a marcação da caixa de diálogo do *Unity Analytics* e bastará o simples clique no ícone *Create Project*.

Outra janela representando o *Unity Package Manager* se abrirá notificando que alguns *assets* do programa estão sendo criados, importados e carregados junto de *scripts* necessários para o funcionamento do programa. Assim que o *Unity3D* abrir seu projeto pela primeira vez, deve-se obter uma janela semelhante à FIGURA 2 abaixo.

FIGURA 3 – Área de trabalho do *Unity3D*



Fonte: O autor.

A janela mostrada na FIGURA 2 exibe a área de trabalho tradicional do motor gráfico da *Unity3D*. No canto superior esquerdo, encontram-se as opções básicas do *software*, onde poderão ser importados, abertos ou salvos projetos e realizar modificações neles. Abaixo, disto, em *Sample Scene* encontram-se as hierarquias de entidades presentes dentro de determinada cena de um projeto, neste caso, estão presentes a *Main Camera*, padrão do *Unity3D* e *Directional Light* que tem como função a iluminação de objetos não luminosos.

Em centro, encontra-se a cena principal de seu projeto. Nela, será possível a visualização de todos os objetos inseridos na cena, incluindo itens importantes, tais como a câmera, objetos de iluminação, modelagens, etc. Saindo da guia “*Scene*”, nota-se a guia *Game*, onde será mostrado a execução do projeto como prévia do resultado final do seu projeto. Ao lado, encontra-se a guia *Asset Store*, onde poderão ser baixados e importados *Assets* produzidos pela comunidade em sua cena.

No canto inferior esquerdo, estão as propriedades do projeto, onde poderão ser gerenciados os *Assets* a serem importados e edição de itens como textura, material ou até mesmo propriedades físicas de determinados objetos importados em cena. A janela disponível à direita exibe a pasta selecionada na guia *Project*, facilitando a operação dentro do *software*. Vale ressaltar que a guia “*Console*” permite

a utilização de códigos e *scripts* que facilitam a utilização de usuários mais experientes, no entanto, não será utilizada neste manual.

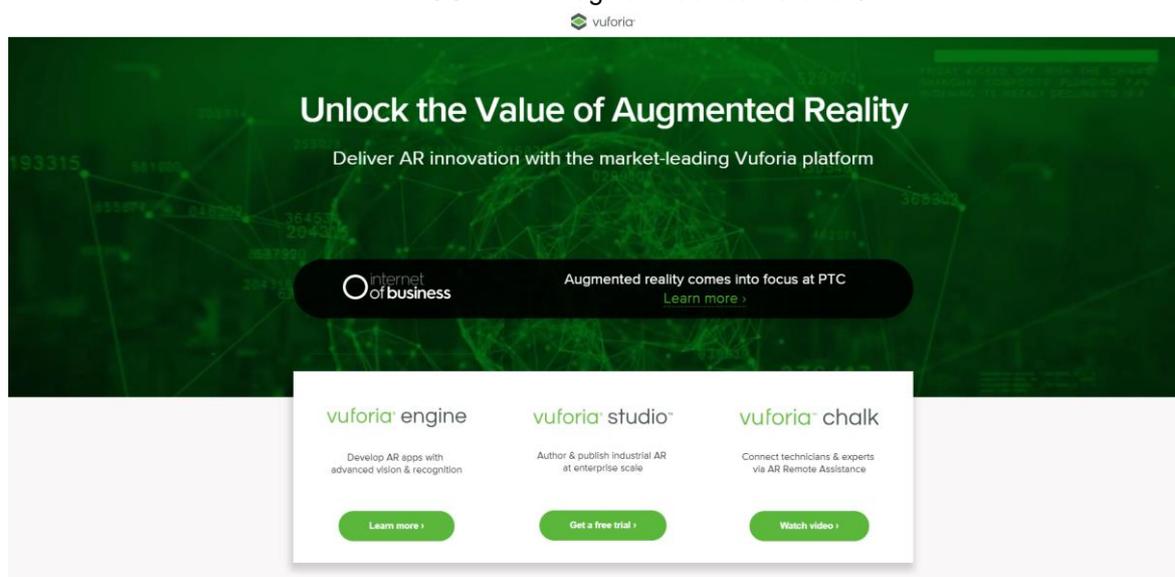
À direita, situa-se a janela *Inspector*, onde poderão ser ajustadas configurações importantes de objetos, como coordenadas de localização, posição em relação a algum ponto, propriedades de material, kits de *software*, tais como o SDK do Vuforia, ou até mesmo do *Android*, juntamente de outras funções. Com o *Unity 3D* já instalado, somente mais um passo relacionado à registro de *softwares* é necessário: registrar uma nova conta no site oficial de kits de Realidade Aumentada *Vuforia*.

- **PASSO 4 – REGISTRO DE MARCADORES NA VUFORIA SDK**

Abrindo o site oficial da empresa, www.vuforia.com, é encontrado um *layout* semelhante ao da FIGURA 3.

O próximo passo a ser seguido é o registro de uma nova conta dentro da interface do site oficial do *Vuforia*. É oferecido pela empresa, o registro gratuito de cerca de 1000 marcadores por usuário, dando maior liberdade de criação e contribuição geral a criadores de materiais em Realidade Aumentada que buscam *softwares* e *frameworks* gratuitos para a facilitação do trabalho de programação.

FIGURA 4 – Página inicial do *Vuforia SDK*

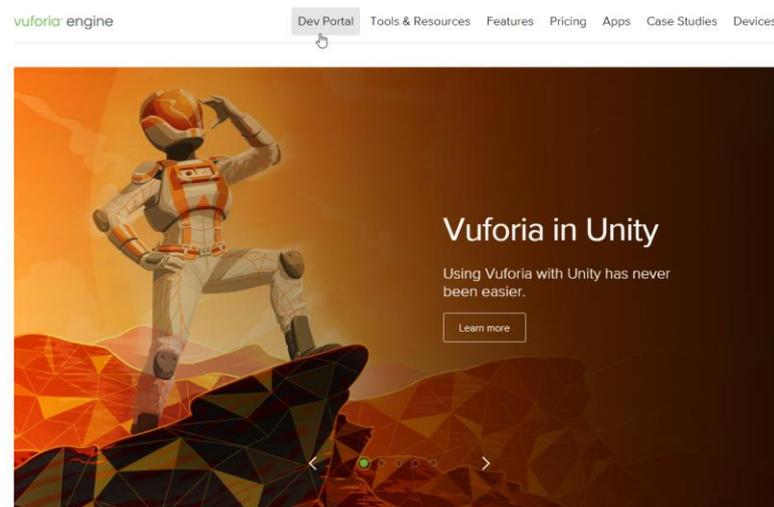


Fonte: O autor.

Outros pacotes podem ser obtidos, contendo conteúdos extras e vantagens para usuários mais especializados na área ou empreendedores, mas para este projeto, se utilizará a versão simples do kit. Para a criação de conta dentro da *Vuforia*, basta clicar primeiramente na aba que simboliza a área da *Vuforia Engine*, onde têm-se escrito “*Learn More*”.

Uma nova guia se abrirá, contendo mais propagandas e possibilidades que podem ser obtidas através dos kits oferecidos pela companhia. Para que uma conta seja criada neste ambiente, deve-se clicar na aba “*Dev Portal*”, diminutivo para *Developer Portal* onde poderão ser ajustadas todas as configurações dos marcadores e suas devidas inscrições de dados dentro do sistema, como mostrado na FIGURA 4.

FIGURA 5 – Página demonstrativa da *Vuforia Engine*

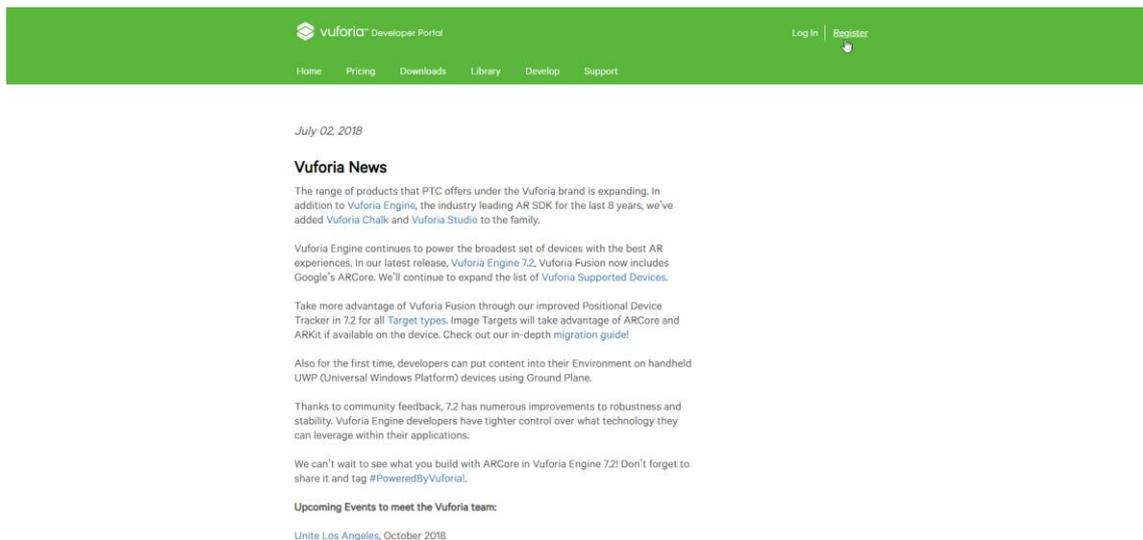


Vuforia is the leading AR platform.

Fonte: O autor.

Ao clicar em *Dev Portal*, uma nova janela se abrirá, contendo notícias sobre as novas atualizações, termos, produtos e mais vantagens de pacotes pagos da própria *Vuforia*. Uma sequência de abas se localiza na seção superior da página, contendo itens tais como *Home*, *Pricing*, *Downloads*, *Library*, *Develop* (onde serão encontradas as configurações e imagens dos marcadores registrados) e *Support*. Mais acima, no canto superior direito, encontram-se as opções de log-in e registro de conta dentro da plataforma da *Vuforia*, como mostrado na FIGURA 5.

FIGURA 6 – Developer Portal - Vuforia



Fonte: O autor

Clicando em *Register*, uma nova janela se abre, contendo caixas de preenchimento dos dados da conta. Após o preenchimento, basta realizar a confirmação do *e-mail* requisitada pela empresa. Assim, faz-se necessária a realização do Log-in na plataforma da *Vuforia*, para que finalmente sejam determinadas as configurações dos marcadores. Abrindo novamente a página de *Developer Portal*, clica-se na aba “*Develop*”.

Uma nova página se abrirá, contendo novas opções e tarefas a serem realizadas. Nota-se que o usuário é automaticamente encaminhado para uma guia menor chamada “*License Manager*” que é responsável pelas configurações das chaves de utilização de marcadores a serem registrados. Ao lado, existe outra guia chamada “*Target Manager*”, onde será disponibilizado o *DataBase* do marcador que deverá ser importado posteriormente para o *Unity 3D*.

Em suma, o primeiro passo neste momento é obter a chave de desenvolvedor (*Development Key*). Basta simplesmente clicar na grande aba contendo este item escrito e uma nova janela se abrirá, como mostrado na FIGURA 6. Desta vez, será requisitado o nome do aplicativo. Neste trabalho, utilizamos como nome “*Teste*” para o aplicativo criado, mas ressalta-se que qualquer nome poderá ser utilizado. Nota-se que é notificado em “*License Key*” os itens adquiridos com este pacote gratuito. Basta clicar em “*Confirm*”.

FIGURA 7 – Página de registro de licença – Vuforia

The screenshot shows the Vuforia Developer Portal interface. At the top, there is a green navigation bar with the Vuforia logo and the text 'vuforia™ Developer Portal'. Below this, there are several menu items: 'Home', 'Pricing', 'Downloads', 'Library', 'Develop' (which is highlighted), and 'Support'. Underneath the 'Develop' menu, there are two sub-tabs: 'License Manager' and 'Target Manager'. Below the navigation bar, there is a link that says 'Back To License Manager'. The main heading is 'Add a free Development License Key'. Below this, there is a section for 'App Name' with a text input field containing the word 'Teste' and a note that says 'You can change this later'. Below the 'App Name' section, there is a section for 'License Key' with the following details: 'Develop', 'Price: No Charge', 'Reco Usage: 1,000 per month', 'Cloud Targets: 1,000', 'VuMark Templates: 1 active', and 'VuMarks: 100'. Below the license key details, there is a checkbox that is checked, with the text: 'By checking this box, I acknowledge that this license key is subject to the terms and conditions of the Vuforia Developer Agreement.' At the bottom of the form, there are two buttons: 'Cancel' and 'Confirm'. A mouse cursor is pointing at the 'Confirm' button.

Fonte: O autor

Agora, a página atualizará e a guia “*License Manager*” agora conterà o novo projeto recém-criado “Teste” como hiperlink. Clicando neste projeto, será aberta uma nova janela que conterà o código da *License Key* e mais alguns status do aplicativo a ser criado. Neste passo, faz-se necessário que este código seja copiado para um bloco de notas seguro (FIGURA 7) para que seja então copiado para dentro do aplicativo *Unity 3D* em passos posteriores.

Em sequência, é hora de registrar o marcador a ser utilizado. Clicando na aba ao lado de *License Key*, *Target Manager*, se abrirá uma nova janela onde serão criados os *DataBases* a serem utilizados dentro do *Unity 3D* como marcadores. Note que estes *DataBases* se tratarão de imagens criadas pelo usuário que serão responsáveis pelo posterior reconhecimento do marcador no aplicativo, sendo assim, um passo imprescindível para o funcionamento do projeto.

O *DataBase* pode ser criado em *Add DataBase* ou simplesmente clicando em *Create your first Database* logo abaixo da pequena aba. Uma nova janela se abrirá lhe perguntando o nome do *DataBase* (neste manual, será “TesteDB”). Em sequência, basta selecionar a caixa de diálogo para definir o *DataBase* para *Device* e por final, *Create*, como mostrado na FIGURA 8.

a imagem como um marcador. Para adicionar um *Target*, clica-se simplesmente em “Add Target”, onde uma nova janela aparecerá como na FIGURA 9.

FIGURA 10 – Configuração dos *Targets* - *Vuforia*

Add Target

Type:

			
Single Image	Cuboid	Cylinder	3D Object

File:

Choose File

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

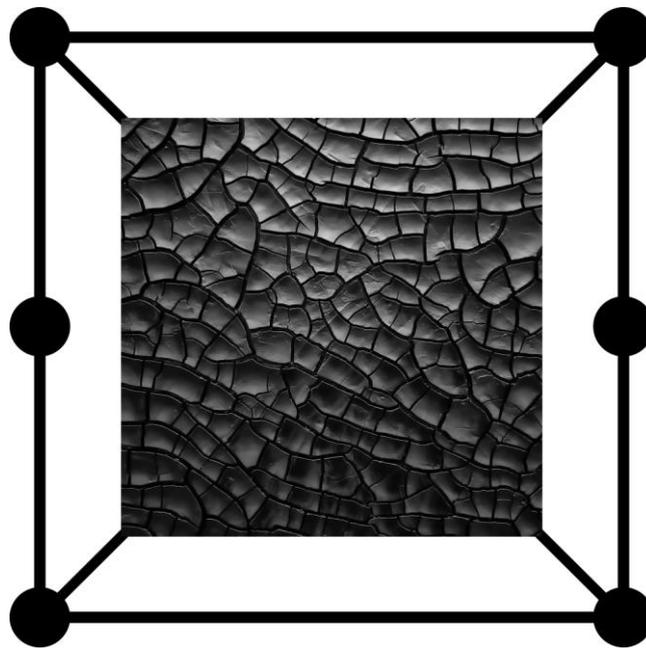
Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Fonte: O autor

Neste passo, é necessário que a imagem que será utilizada para um marcador já esteja pronta e finalizada, para que então seja carregada no aplicativo. Para o seu funcionamento dentro do kit do *Vuforia*, um marcador deve conter contrastes altos e uma complexidade suficientemente elevada para o reconhecimento do dispositivo. Vale ressaltar também, que o leitor de marcadores concedido pelo *Vuforia* funciona de forma acromática, ou seja, não se reconhece cores. Desta forma, é importante que seja criada uma imagem que contenha contrastes compostos por outros fatores que não sejam as suas devidas cores.

O marcador criado a ser utilizado neste manual se trata de uma textura aleatória que fosse complexa para o reconhecimento do marcador (FIGURA 10). Suas dimensões são de 1000 pixels por 1000 pixels, e a imagem deve se encontrar em formato PNG. Ainda que esteja colorida, ressalta-se a importância da obtenção de contraste através de objetos e não de cores, já que o leitor realiza o reconhecimento de maneira acromática.

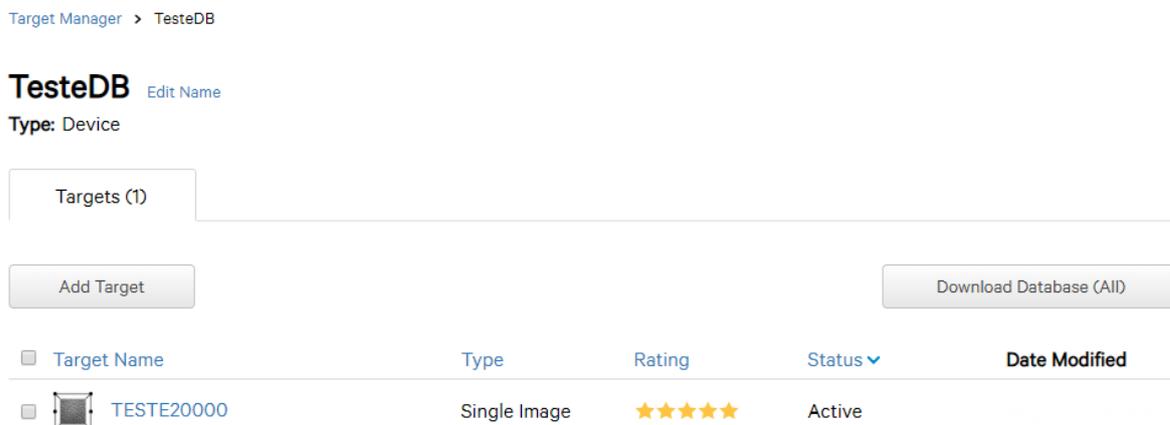
FIGURA 11 – Marcador criado para este manual



Fonte: O autor

Assim, basta que a imagem seja carregada na opção “*Choose File*” e então especificada suas devidas dimensões a serem carregadas dentro do *software*. Neste contexto, como a imagem possui 1000x1000 pixels de largura e altura, põe-se o simples valor de largura dentro de sua respectiva caixa de resposta (*Width*), sendo assim, “1000”. Isto definirá a proporção e escala de seu marcador dentro do *Unity 3D*. Com relação à escolha do nome deste *Target*, deve-se escolher um nome que nunca tenha sido utilizado antes. Desta maneira, foi escolhido neste manual, um mero exemplo, sendo este, “*TESTE2000*”, como mostrado na FIGURA 11.

FIGURA 12 – Demonstração de eficiência do marcador (*Target*)



Fonte: O autor

Nota-se na figura acima que, para confirmar a eficiência do marcador dentro do kit *Vuforia*, deve-se atentar ao *rating* recebido pelo site. No caso do marcador escolhido, já se tem 5 estrelas necessárias para o reconhecimento posterior do *software*. Se o marcador criado pelo usuário possuir 3 ou menos estrelas, é recomendado que seja revisto o design do marcador para que este tenha mais funcionalidade no produto final.

Assim, em suma, está concluída a criação do marcador. Vale ressaltar que caso seja utilizado mais de um marcador em um aplicativo, deve-se seguir este mesmo passo de adição de *Targets*, cada um se referindo ao respectivo marcador, assim, cada um possuindo imagens suficientemente divergentes uma da outra, para que não haja erros de reconhecimento no aplicativo. À vista disso, basta somente realizar o *download* do *DataBase* do marcador (es) criado (s).

Quando clicado na opção “*Download DataBase*”, será aberta uma nova janela contendo as opções de *download* disponíveis (FIGURA 12). Neste caso, opta-se pelo formato *Unity Editor*, já que será carregado no *software Unity 3D*. Os dados serão compilados e o arquivo será baixado. Recomenda-se baixar na mesma pasta de projeto criada previamente no *Unity 3D*. Assim que este arquivo for baixado, deve-se executá-lo, com o *Unity 3D* já aberto. Deste modo, os dados do marcador serão importados automaticamente para dentro do programa e poderão ser utilizados sem quaisquer erros subsequentes. Assim que importados os arquivos da *DataBase* do

marcador, os próximos passos serão executados exclusivamente dentro da plataforma do *Unity3D*.

FIGURA 13 – Plataforma a ser escolhida - *Vuforia*

Download Database

1 of 1 active targets will be downloaded

Name:

TesteDB

Select a development platform:

Android Studio, Xcode or Visual Studio

Unity Editor

Cancel

Download

Fonte: O autor

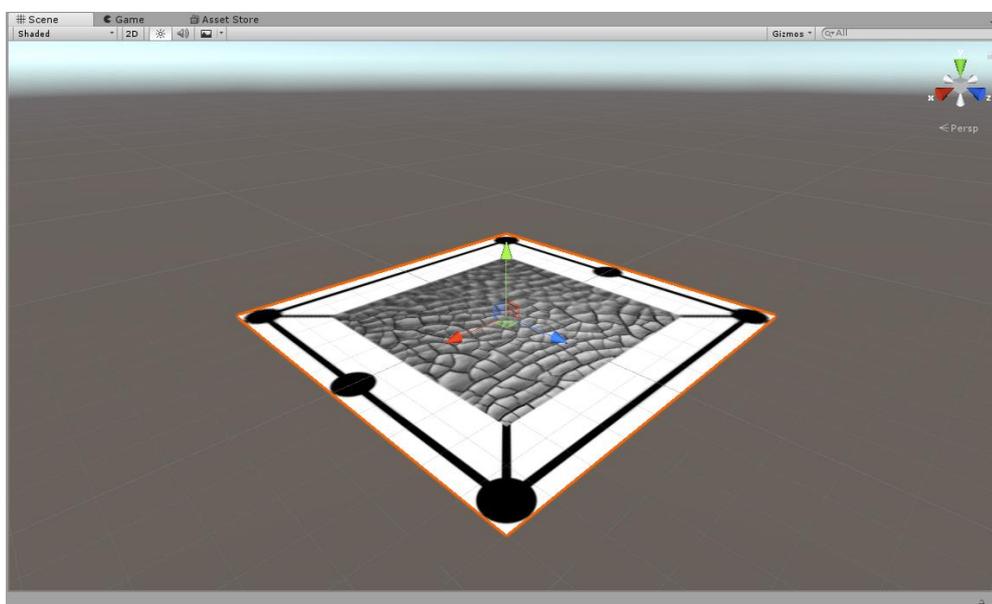
- **PASSO 5 – DESENVOLVIMENTO DO SOFTWARE NO UNITY 3D**

Feito todos os passos anteriores, serão realizados os passos mais rápidos deste manual, ainda que sejam muitos comandos. Foi observado que após uma grande repetição do processo, a fabricação de aplicativos em Realidade Aumentada, quando os marcadores já estão configurados, se torna um procedimento fácil e rápido de ser realizado. Neste sentido, a primeira etapa a ser seguida dentro desse *software* é ativar a opção da *Vuforia Augmented Reality*. Para isso, clica-se em *Edit > Project Settings > Player*. Feito este passo, na janela *Inspector*, à direita da interface, se abrirão as opções de *Player*. Na seção inferior, encontram-se as *XR Settings* onde poderá ser ativado a *Vuforia Augmented Reality* no *Unity3D*, tanto para PCs e *Android*.

Quando marcada a caixa de diálogo, serão importados os arquivos correspondentes ao kit do *Vuforia*¹. Assim que os arquivos forem importados, faz-se necessária a adição da câmera especializada em Realidade Aumentada da própria *Vuforia*. Para que este passo seja realizado, bata que seja seguido estes seguintes caminhos: *GameObject* > *Vuforia* > *AR Camera*. Nota-se que uma *AR Camera* será adicionada à cena na área lateral esquerda da interface, na janela *SampleScene*.

Considerando que agora temos uma *AR Camera* responsável pelo projeto, nota-se que não se faz mais necessária a presença da câmera padrão determinada pelo *Unity 3D*, a *Main Camera*. Seleciona-se a *Main Camera* na janela “*Sample Scene*” à esquerda e clica-se em “Delete”. Assim, percebe-se que a cena restará com somente a *Directional Light* e a *AR Camera* carregados. O próximo passo, é o carregamento do marcador. Clica-se em *GameObject* > *Vuforia* > *Image*. Na janela de cena, será possível visualizar a imagem do marcador sendo carregada no cenário, de acordo com a resolução estabelecida dentro do site da *Vuforia*, como mostrado na FIGURA 13.

FIGURA 14 – Carregamento do marcador configurado – *Unity 3D*

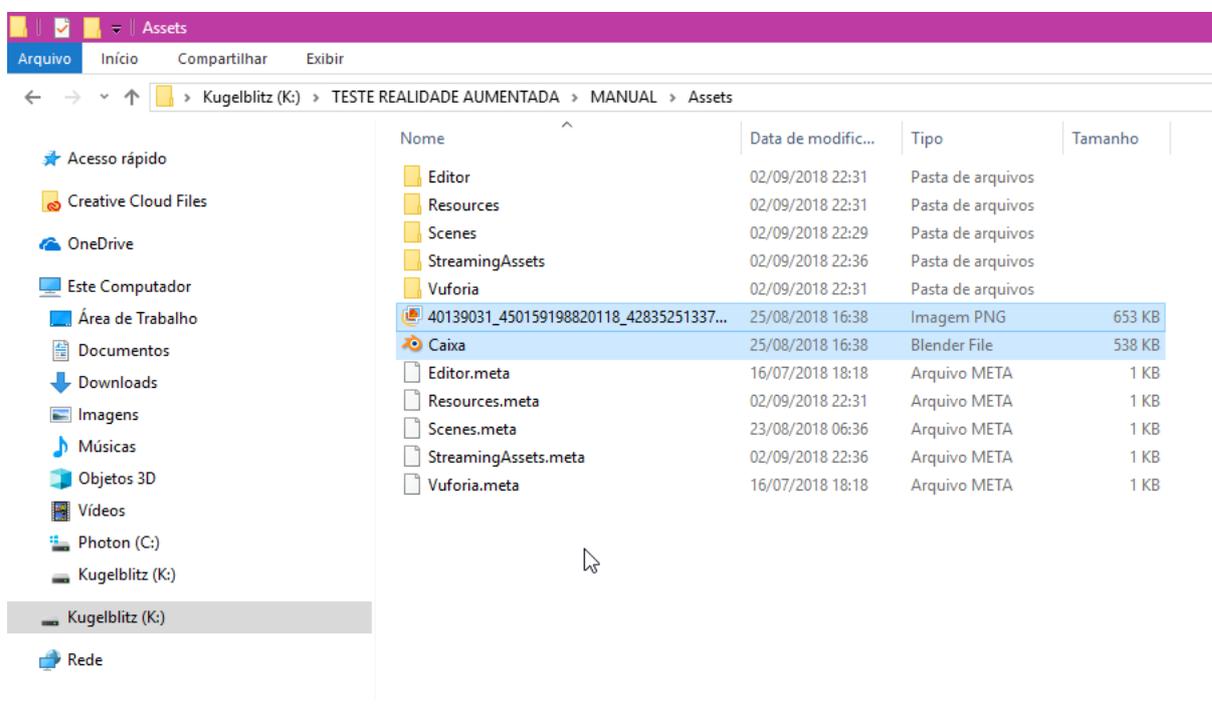


Fonte: O autor

¹ Talvez seja necessária a reinicialização do programa para que estes arquivos sejam importados de acordo.

Depois do marcador ser carregado com sucesso para dentro do *Unity 3D*, o próximo passo se resume em inserir a modelagem desejada para dentro do *software*. Neste caso, a modelagem do objeto desejado foi realizada dentro do *software* gratuito *Blender 3D* e salva em formato “*blend*”. O método mais simples de importação de modelagens dentro do *Unity 3D* se inseri-las nas *Assets* de um determinado projeto. Desta forma, basta que a textura do objeto já desdobrada e a modelagem sejam transportadas para dentro da pasta *Assets* de seu projeto, como mostrado na FIGURA 14.

FIGURA 15 – Transferência dos arquivos de modelagem para dentro da pasta *Assets*



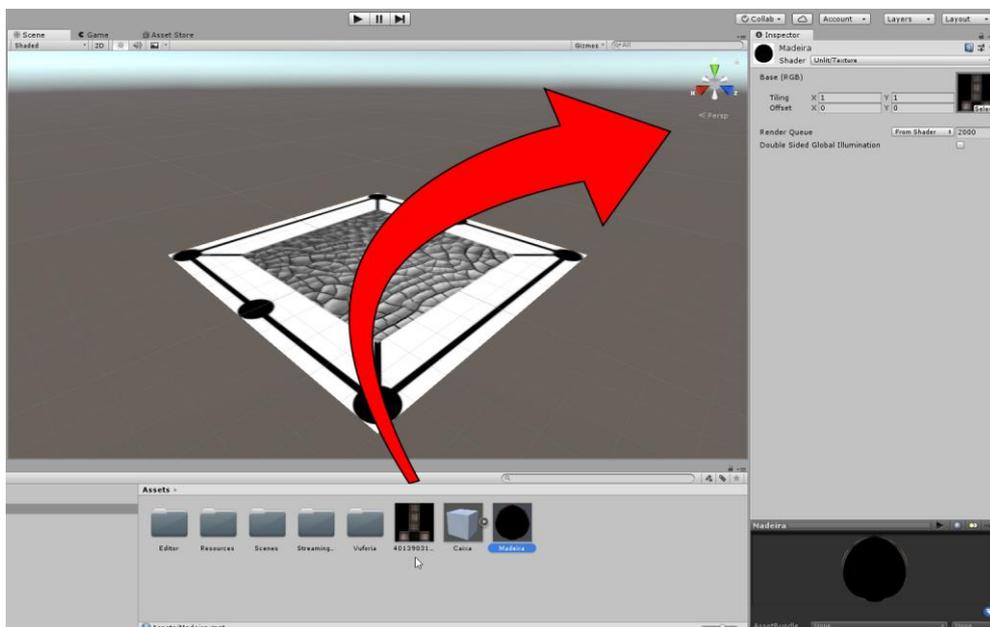
Fonte: O autor

Simple e rápido. Quando se abre a janela do *Unity 3D* novamente, observa-se que a modelagem realizada já se encontra dentro do programa, na janela inferior “*Projects*”, onde pode-se navegar pelas pastas do projeto. Entretanto, percebe-se que as texturas já aplicadas dentro do programa *Blender 3D* foram completamente removidas do objeto, o deixando com um material padrão escolhido pela própria *Unity 3D*. Para que isto seja alterado, basta que um novo material seja criado dentro da *Unity 3D*, desta vez, utilizando as texturas já preparadas para esta tarefa (neste caso, a imagem desdobrada em PNG).

Na própria janela “*Projects*”, onde o objeto se encontra, clica-se com o botão direito e segue-se o seguinte caminho: *Create > Material*. Observa-se que um novo material aparecerá ao lado de seu objeto, exigindo uma nomeação. Neste manual, nomeou-se de “Madeira”. Clica-se somente uma vez em cima deste material, com o botão esquerdo do mouse. Ao lado direito, na janela de *Inspector*, aparecerão configurações possíveis de seu material, desde cores, mapas, detalhes que poderão ser facilmente alterados pelas janelas.

Neste manual, se usará a textura já criada pelo Blender 3D, desta forma, basta que seja um material *Unlit*, ou seja, que não necessite de iluminação para que seja visível no programa. Assim, com o material “Madeira” selecionado, clica-se em *Shader* no painel *Inspector*, onde se abrirá uma lista de *shaders* disponíveis para utilização em seu projeto, muitos deles, promovendo efeitos foto realísticos. Entretanto, neste caso, se clicará em *Unlit > Texture*. Note que o painel *Inspector* mudou novamente, desta vez, possuindo as configurações da textura de seu material. No slot disponível para uma textura, basta que seja arrastada a imagem dentro da pasta *Assets*, na janela *Project*, como mostrado na FIGURA 15.

FIGURA 16 – Configuração da textura do objeto – Unity 3D



Fonte: O autor

Feito este passo, agora chega a hora de se adicionar o objeto propriamente dito em cena. Na mesma janela inferior, *Project*, seleciona-se o objeto, neste caso a

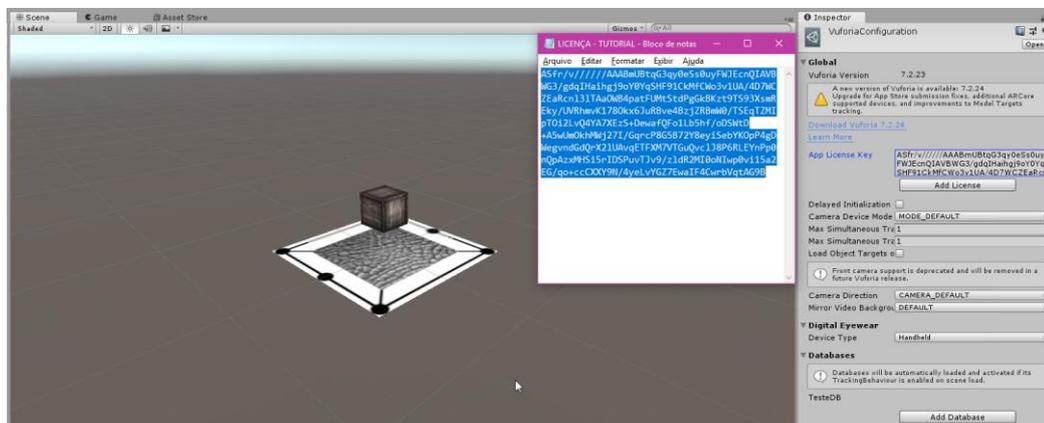
caixa, e o arrasta em direção à janela de cena, *SampleScene*, mais especificamente para dentro do *ImageTarget*. Assim, a janela à esquerda, *SampleScene* conterá agora os seguintes itens em sua composição: *Directional Light*, *AR Camera*, *ImageTarget* e dentro da *ImageTarget*, a caixa.

Com o zoom aberto na cena, ao centro da tela, nota-se que o objeto pode estar fora de escala. Assim, para ajustar o posicionamento e dimensões dos objetos, basta que este objeto, no caso a caixa, esteja selecionada, e que estas configurações sejam alteradas no painel *Inspector*. Feito este passo, note que o objeto ainda está sem material. Como a criação do material já foi realizada, basta que seja arrastada o material criado, neste manual, a madeira, em direção à caixa, já dentro do projeto, assim, finalizando a importação de arquivos para dentro do *Unity 3D*.

- **PASSO 6 – COMPILANDO O APLICATIVO A PARTIR DO *UNITY 3D***

É chegado o momento de realizar as configurações do aplicativo a ser criado. Primeiro, é necessário que ajustemos as configurações de licença da *Vuforia* dentro do *Unity 3D*. Para isso, basta clicar em *AR Camera*, no painel *SampleScene* e então, no painel *Inspector*, clicar em “*Open Vuforia Configuration*”. O *Inspector* abrirá uma nova janela contendo as configurações da *Vuforia* dentro do *Unity 3D*. Note que existe uma janela para que seja preenchida a licença do aplicativo “*App License Key*”. Neste passo, basta que seja copiada do bloco de notas a licença adquirida previamente no site oficial da *Vuforia SDK*, como mostrado na FIGURA 16.

FIGURA 17 – Adicionando a licença da *Vuforia* no *Unity 3D*



Fonte: O autor

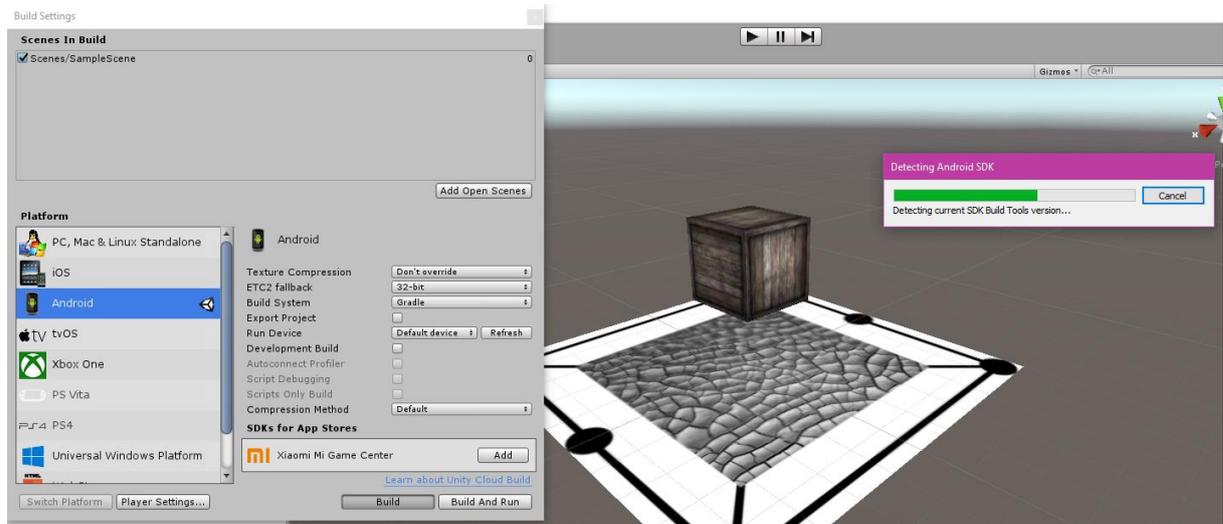
Feito este passo importante, é hora de configurar a *Build*, mas para isso, é preciso que sejam localizados os SDKs e JDK necessários para a compilação, estes, baixados no início deste manual. Clica-se em *Edit > Preferences* para que sejam abertas as preferências do *Unity 3D*. Em seguida, clica-se em *External Tools*. No final da janela aberta, se encontram as janelas dos diretórios dos SDKs e JDK, apenas esperando para serem especificados. Desta forma, basta na seção *Android*, clica-se em *Browse* na janela *SDK* e na janela *JDK*. O *Unity 3D* geralmente localiza os arquivos automaticamente, mas caso isto não aconteça, basta que lidere o aplicativo em direção à pasta das quais foram instalados o *Android Studio* e o *Java SDK*.

Outro passo importante a ser ajustado, é a configuração das versões de *Android* em que o aplicativo será compilado. Neste caso, segue-se o seguinte caminho: *Edit > Project Settings > Player > Other Settings*. Em *Company Name*, utilizou-se UFPR. Abaixo, em *Identification*, deve-se nomear a identificação deste aplicativo de acordo com a *Company* nomeada e o nome do produto, por exemplo, “*com.company.nomedoproduto*” (letras em minúsculo). Neste manual, a identificação se resume em: *com.ufpr.manual*. Logo abaixo, escolhe-se as versões correspondentes do *Android*, em *Minimum API Level* (*Android* mínimo aceito para execução) e *Target API Level* (*Android*-alvo a ser usado).

Assim que estes passos forem realizados, abre-se novamente a janela de *Build*, através do endereço “*File > Build Settings*”. A partir disto, se abre uma janela contendo vários elementos que podem ser configurados. Em centro e acima, encontra-se a janela *Scenes In Build*, na qual pode-se adicionar o remover cenas a serem compiladas no aplicativo. Nesta janela, já se encontra a cena criada, “*SampleScene*”. Na janela *Platform*, seleciona-se a plataforma escolhida para que o aplicativo seja executado, neste caso, seleciona-se, “*Android*”.

Quando selecionado, pode-se mudar algumas alterações referentes ao aplicativo, tais como SDKs extras que adicionam novas funções ao aplicativo, configurações de compressão de texturas, formas de exportação, qual dispositivo específico a ser compilado (neste caso, conecta-se algum celular no computador para que ele reconheça os dispositivos disponíveis). Isto posto, serão mantidas as configurações padronizadas do próprio *software* e realizar a compilação, clicando em “*Build*”. Uma nova janela aparecerá, confirmando o diretório a ser salvo, e um processo relativamente longo se iniciará, conforme a FIGURA 17.

FIGURA 18 – Compilando o projeto dentro do *Unity 3D*



Fonte: O autor

Quando a compilação finalizar, se abrirá uma janela notificando a finalização da *Build*, e basta que o arquivo APK seja transferido para o dispositivo a ser utilizado. Assim, o último passo deste tutorial se resume à instalação do *software* dentro do dispositivo, que é muito intuitiva, já que basta somente abrir o executável APK gerado pelo *Unity 3D*. Com o marcador impresso separadamente, basta que a câmera do aplicativo seja aberta e apontada em direção ao marcador, revelando a imagem 3D do objeto criado.