

Exploração do High Fidelity para desenvolvimento de jogos em mundos virtuais imersivos multiutilizador

José Martins

INESC TEC e Universidade
Aberta
Lisboa, Portugal
martins.josemanuel@gmail.com

Leonel Morgado

INESC TEC e Universidade
Aberta
Coimbra, Portugal
leonel.morgado@uab.pt

Vítor Cardoso

CIAC e Universidade Aberta
Lisboa, Portugal
vitor.cardoso@uab.pt

RESUMO

Apresentamos o resultado de uma exploração prática da tecnologia de mundos virtuais imersivos multiutilizador High Fidelity, baseada em tecnologia Web. Esta tecnologia permite criar mundos virtuais cujas formas de interligação, controlo e interação tiram partido de uma Interface de Programação de Aplicações em JavaScript. Através do caso prático de desenvolvimento de um protótipo de jogo educativo simples, descrevemos a tecnologia High Fidelity, incluindo o tipo de *scripts*, a arquitetura inerente e as suas características de desenvolvimento e utilização. Destacamos as dificuldades inerentes ao estado atual da plataforma, em constante reformulação e algumas peculiaridades.

Palavras-chaves do autor

Mundos virtuais; simulações; jogos; High Fidelity

INTRODUÇÃO

A utilização de simulações e/ou jogos nas empresas e outras organizações nos sistemas de gestão de formação, baseada em tecnologia Web, vem sendo desenvolvida há vários anos, em áreas tão diversas como a manutenção mecânica aeronáutica, condução automóvel, controlo de submarinos, fisioterapia remota ou ciclismo [1]. A tecnologia High Fidelity é uma proposta recente nesta área, que recebeu alguma atenção mediática por provir de uma empresa fundada pelo fundador do Second Life, Philip Rosedale. O High Fidelity posiciona-se como uma tecnologia aberta, que instalável em computadores pessoais e que disponibiliza uma interface para programação de aplicações (API, *Application Programming Interface*) em JavaScript, visando permitir a qualquer pessoa com conhecimentos técnicos criar e partilhar um ambiente de realidade virtual.

É uma tecnologia recente, encontrando-se disponível em <https://highfidelity.io/> [acedido a 10 de outubro de 2016] à data do presente artigo ainda em versão de testes. Aliás, durante a exploração aqui descrita, o ritmo de lançamento de versões era praticamente diário. Assim, salienta-se que aquando da leitura do presente artigo a tecnologia poderá

ter sofrido alterações.

Contudo, o desenvolvimento para High Fidelity defronta-se com faltas de documentação várias. Aspetos aparentemente cruciais, como a localização de ficheiros de *backups* ou de configuração, não são descritos na documentação disponível, o que dificulta sobremaneira a utilização desta tecnologia. Desta forma, o resumo aqui apresentado, resultado da nossa experimentação ao longo de alguns meses, poderá contribuir para que mais pessoas possam tirar partido dela e explorá-la.

HIGH FIDELITY- BREVE DESCRIÇÃO

Conforme já se mencionou, o High Fidelity é uma tecnologia de mundos virtuais imersivos multiutilizador. A empresa que a produz designa por “metaverso” o conjunto desses mundos virtuais, opção comum na terminologia da área [11]. Cada mundo criado com o High Fidelity, porção finita do metaverso, é designado por “domínio”. A cada domínio pode-se associar um *place name* que o identifica no metaverso do High Fidelity [6]. Pode ainda ter-se domínios isolados do metaverso, designados por domínios locais. Conforme a configuração o acesso a um domínio pode estar aberto a qualquer utilizador, inclusivamente para editar conteúdos e adicionar entidades (configuração predefinida), ou ser restrito em maior ou menor grau [8].

Cada utilizador interage com um domínio (ou todo o metaverso) do High Fidelity através de um programa cliente, que efetua a ligação aos servidores e produz a representação visual do espaço. Os utilizadores, como é habitual neste tipo de plataformas, são representados como parte integrante do mundo virtual, sob a forma de um avatar. No HighFidelity, este programa cliente designa-se “Interface”, disponibilizando formas de interação habituais, como rato, teclado e ecrã ou mais imersivas como óculos de realidade virtual (*head-mounted displays*) ou controladores espaciais para as mãos.

O programa cliente Interface disponibiliza também ferramentas para manipular o domínio, o seu conteúdo e para executar *scripts* em JavaScript. O conteúdo consiste em modelos tridimensionais, que podem ser importados nos formatos *.fbx* ou *.obj*; e em *scripts*, com particularidades de alojamento e execução que se descrevem na secção seguinte.

Supported by research effort FourEyes of project TEC4Growth. “TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020” is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

SCRIPTS NO HIGH FIDELITY

Os *scripts* suportados pelo High Fidelity são escritos em JavaScript. Há uma particularidade: além de poderem estar alojados no servidor do domínio HighFidelity, podem estar alojados em qualquer servidor com tecnologia Web, inclusivamente em serviços de *cloud* públicos. E podem ser executados de forma distribuída: no programa cliente, no servidor de domínio ou noutros servidores, especificamente para execução de *scripts* [10]. Dividem-se em três categorias: *scripts* de interface, *scripts* de entidades e *scripts* de clientes de atribuição [9].

Os *scripts* de interface permitem personalizar a experiência de interação do utilizador num domínio, alterando o funcionamento do programa cliente Interface. Por exemplo, através da criação de novos menus e de acessórios. Ou seja, permitem personalizar a interface de utilizador com aspetos pessoais, inacessíveis e não visíveis aos outros visitantes do domínio. Podem, contudo, afetar o domínio. Se um *script* de interface afetar o domínio, o resultado será visto por todos os programas clientes, que são informados da alteração pelo servidor do domínio HighFidelity. Os *scripts*

de interface são executados pelo programa cliente Interface, através de um *script* próprio, designado Runner Script. A sua execução termina quando se fecha o programa cliente Interface.

Os *scripts* de entidades são associados às entidades do domínio, entendidas como objetos e restante conteúdo visual do espaço tridimensional. São usados para permitir interações com as entidades ou executar alguma ação automática. Este tipo de *scripts* é pré-carregado pelo programa cliente Interface quando o utilizador se aproxima da entidade e descarregado quando se afasta dela, pois são executados concorrentemente pelos programas clientes dos utilizadores. Qualquer alteração que ocorra na entidade em consequência da execução de um *script* de entidade é reportada pelo programa cliente ao domínio, o que origina a sua visualização nos programas clientes dos demais utilizadores presentes nesse domínio. Estes *scripts* de entidades podem ter o código alojado no servidor do domínio (por inserção direta do código nas propriedades da entidade) ou ter o código alojado em servidores na Web (indicando URL do código do *script* nessas propriedades).

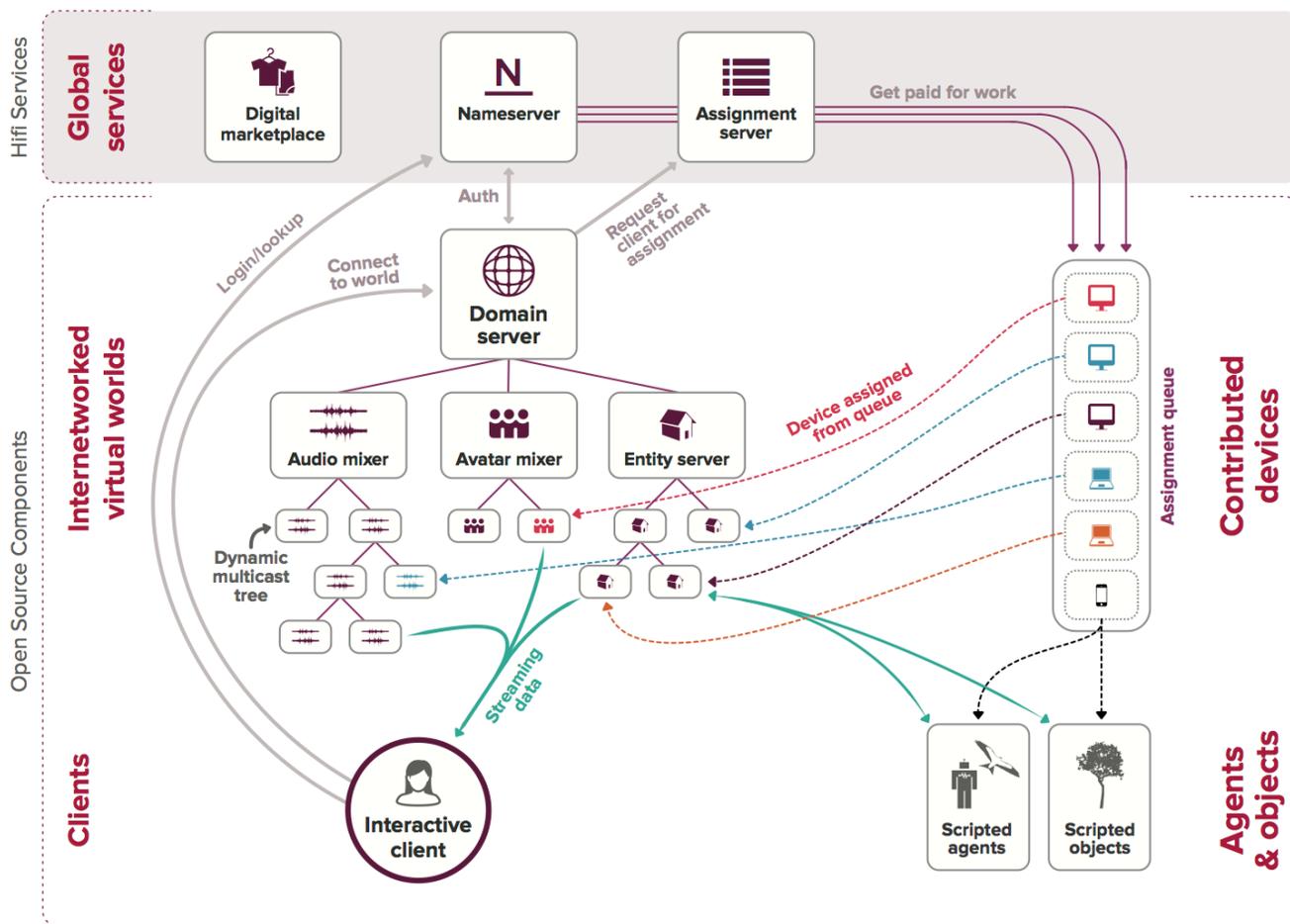


Figura 2. Visão na terceira pessoa com âmini-espelho.

Fonte: https://files.readme.io/fE5d8oNxSaqPIUSa7Qs3_bfkWsEouTfGTsG3PEHVi.png [10 de outubro de 2016].

Esta última opção não estava disponível no momento em que escrevemos este artigo [2], apesar de ter estado disponível em versões alfa durante o nosso período de exploração.

Os *scripts de clientes de atribuição* são *scripts* cuja execução está a cabo de servidores externos ao domínio. Permitem coordenar o estado do mundo virtual, jogo ou aplicação entre várias entidades, ou entre estas e avatares, além de permitir distribuir a carga de processamento. Esses servidores de execução de *scripts* são designados “clientes de atribuição” ou “agentes”. Estão ligados ao servidor de domínio e os resultados da execução são comunicados por este a todos os programas cliente (Interface). Assim, os *scripts* de clientes de atribuição são executados desde que o servidor de domínio esteja em funcionamento. Os URL dos servidores de atribuição são registados nas configurações do domínio.

ARQUITETURA DO HIGH FIDELITY

Um domínio é gerido por uma pilha de quatro servidores virtuais que tem no topo o Servidor de Domínio, como ilustra a Figura 2. O Servidor de Domínio é o principal, sendo iniciado para criar o mundo virtual. É responsável por fornecer o seu número IP aos clientes Interface, para comunicação por parte destes; e por distribuir tarefas de simulação aos outros servidores: serviços globais (nomes e atribuições); Misturador de Áudio; Misturador de Avatares; e Servidor de Entidades. Estes servidores estão normalmente na máquina que aloja o domínio, mas podem estar noutra [6].

O Misturador de Avatares é responsável por tudo o que esteja relacionado com presença dos avatares no domínio: posição, modelo, movimentação, expressões faciais, etc. O Misturador de Áudio mistura os sons do ambiente com os dos avatares, apresentando uma mistura a cada avatar de acordo com a sua posição em relação às fontes sonoras. O Servidor de Entidades controla todas as entidades do domínio, que têm *scripts* associados. Qualquer alteração numa entidade do domínio é-lhe comunicada, que a retransmitirá para todos os clientes Interface.

Dos serviços globais, o Servidor de Nomes permite encontrar domínios e locais específicos dentro de um domínio, através de nomes únicos. Permite ainda autenticar utilizadores: quando um domínio é criado, o acesso ao mesmo pode ser restringido a um conjunto de utilizadores, cabendo ao servidor de nomes verificar as identidades [7]. O Servidor de Atribuições permite registar no High Fidelity um dispositivo computacional na Internet para executar tarefas: servidor de domínio ou serviços de execução de *scripts*. O Servidor de Atribuição delega tarefas aos dispositivos disponíveis aos utilizadores (*ibid.*).

INSTALAÇÃO E CONFIGURAÇÃO

A tecnologia High Fidelity encontra-se disponível para Windows, Mac Os e Linux. O modo mais simples de

configurar um domínio é através da aplicação *Sandbox*, disponível para Windows e Mac OS. Também é possível compilar o servidor domínio e os clientes de atribuição a partir do código fonte disponível.

A *Sandbox* cria um domínio *Home* na máquina local (<http://localhost:40100>). Quando se encontra em execução inicia o servidor de domínio e disponibiliza um ícone na barra de tarefas. Neste, acede-se a um menu onde a opção *Go Home* executa o cliente Interface neste domínio *Home*, com o avatar na posição configurada para o domínio, por predefinição as coordenadas (0,0,0). Apesar do nome, este domínio pode ser acedido por outros utilizadores na rede, se as configurações de rede o permitirem.

Quando a *Sandbox* é executada pela primeira vez, é atribuído ao domínio *Home* um *place name* temporário no servidor de nomes. O novo domínio, para ser reconhecido no metaverso no High Fidelity, deve estar associado a uma conta no site do High Fidelity, configurada na opção *Metaverse/Networking* do menu.

Pode-se restringir o acesso ao domínio a um utilizador específico, bem como limitar a capacidade máxima de utilizadores, as permissões destes e a largura de banda de envio de dados para cada utilizador.

A natureza distribuída do High Fidelity permite ter os recursos num servidor público na Internet e/ou no servidor de recursos do domínio, que se ativa com a opção *Asset-server* da configuração. Nesta situação, detetámos que os recursos locais em servidores Windows são alojados por em `<PastaDoUtilizador>\AppData\Roaming\High Fidelity\ assignment-client\assets\files` (não testámos a localização noutros sistemas operativos).

Em relação aos *backups*, os mesmos podem ser configurados selecionando a opção *Entity Server Settings*. Neste item, entre outras configurações, é possível definir vários tipos de *backups* em termos de intervalos de tempo e número máximo de versões; o tempo máximo de vida para a entidades temporárias; e o caminho para o ficheiro com os *backups*. Os *backups* são guardados em `<PastaDoUtilizador>\AppData\Roaming\High Fidelity\ assignment-client\entities`.

A importação dos *backups* é efetuada através do cliente Interface, na opção *Import Entities* do menu *Edit*. Esta só se encontra disponível se a opção *Advanced Menus* do menu *Settings* tiver sido ativada. Para restaurar um domínio através do ficheiro do *backup*, é necessário que este esteja descompactado. Além disso, só são aceites para importação ficheiros com dados em formato JSON.

DOMÍNIO EXEMPLIFICATIVO: PROTÓTIPO DE JOGO

Descrição do Domínio

A exploração que permitiu efetuar a descrição aqui efetuada foi efetuada no decurso da criação de um domínio exemplificativo no High Fidelity, que simula um jogo sério

para aprendizagem de matemática, nomeadamente sólidos geométricos. O espaço de aprendizagem restringe-se ao interior de uma casa, numa região de montanha. O exterior da casa é caracterizado por um ambiente de montanha com algumas árvores, iluminado ao longo do dia pelo percurso do sol (Figura 5). No interior desta pode-se encontrar uma secretária com livros, uma folha com a planificação do cubo, um candeeiro a óleo, uma cadeira, um fogão com uma caixa de fósforos, um fósforo, sólidos geométricos espalhados pela casa, duas caixas, uma mochila, uma página *web* com conteúdos sobre sólidos geométricos e outros elementos decorativos.

O objetivo é que o utilizador, após estudar sólidos geométricos numa página *web*, arrume os sólidos geométricos dispersos pela casa nas caixas correspondentes, identificadas como “Poliedros” e “Não poliedros”.



Figure 5. Exterior do domínio ilustração.

Interações

Os elementos do domínio que permitem interação (agarrar e mover) por possuírem características físicas, são: mochila, sólidos geométricos, livros, cadeira, folha com a planificação, candeeiros, caixa de fósforos e o fósforo.

A página *web* contém os conteúdos necessários a adquirir para organizar os sólidos na caixa correta, Figura 6. Quando um sólido é colocado na caixa correspondente o nome do sólido é apresentado no quadro da caixa. Quando no interior da caixa estiverem todos os sólidos correspondentes que estavam espalhados pela casa (Figura 7) e apenas os correspondentes, é emitido um som informativo de resposta correta.

Ao clicar com a roda do rato na folha com a planificação do cubo que se encontra sobre a secretária é exibida uma nota informativa sobre o que fazer com essa folha, que poderia ser usada como pista para novas tarefas.

Dependendo do momento do dia em que a casa é visitada, poderá ser necessário usar o candeeiro a petróleo. Este possui um regulador de luz com seis níveis de intensidade. Para aumentar tem que se clicar no regulador com o lado esquerdo do rato e para diminuir tem que se clicar com a roda do rato (Figura 8).



Figure 6. Página web embebida no domínio.

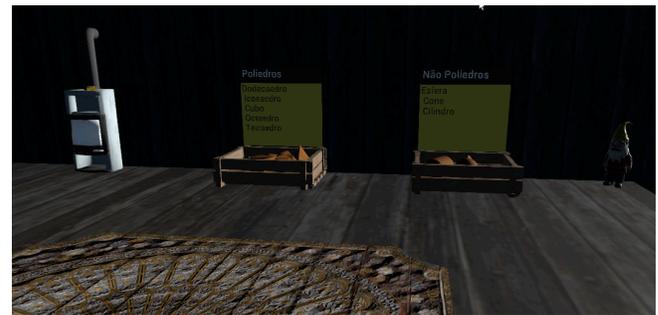


Figura 7. Sólidos nas caixas correspondentes.

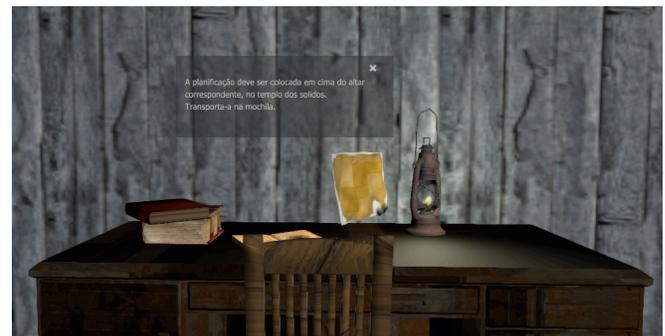


Figura 8. Interações com o candeeiro e com a folha com a planificação.

EXPERIÊNCIA DE UTILIZAÇÃO DO HIGH FIDELITY

Basta dispor de dispositivos tradicionais de interação pessoa-computador (teclado, rato e monitor), mas o High Fidelity suporta controladores somáticos (gestos das mãos) e óculos de realidade virtual.

O cliente Interface vem por predefinição com um conjunto de *scripts* configurado para execução automática, que determinam aspetos da experiência do utilizador e disponibilizam ferramentas para criação de um domínio. Pode dar-se o caso de não arrancarem e é necessário verificar a sua execução: *progress.js* (evolução do carregamento das entidades inseridas no domínio por URL); *grab.js* (permite agarrar com o rato entidades móveis); *users.js* (lista de utilizadores online e controlo da

visibilidade do utilizador no metaverso do High Fidelity); `handControllerGrab.js` (permite agarrar as entidades móveis com controladores somáticos); `notifications.js` (gera notificações no cliente Interface); `dialTone.js` (emite um som quando se liga ou se desliga do servidor); `squeezeHand.js` (orienta a animação das mãos com base no controlador somático); `handControllerPointer.js` (permite controlar o rato com o controlador somático); `examples.js` (acesso à biblioteca de exemplos); `selectAudioDevice.js` (menu *Audio* do cliente Interface); `defaultScripts.js` (iniciação de todos estes *scripts* enumerados) [4].

Para navegar no metaverso do High Fidelity, utilizando o cliente Interface tem de se começar por definir o mundo virtual (domínio) a explorar, através do menu *Navigate*. Este permite também aceder a um local específico dentro de um domínio. O destino é dado através do respetivo *place name* (registado no servidor de nomes), endereço IP ou caminho para um determinado local.

O avatar é o representante do utilizador no mundo virtual. No cliente Interface, o menu *Avatar* permite configurar o tamanho do avatar e anexar elementos/modelos através do seu endereço URL a diferentes partes do corpo (Figura 4). Outros aspetos, como atribuir um nome ao avatar, definir o som de colisão do avatar com outras entidades e escolher a aparência, estão no submenu *Avatar*, do menu *Settings*.



Figura 4. Avatar com elementos anexados.

De referir que algumas das funcionalidades dos menus da interface, indicadas na documentação, apenas estão disponíveis aquando a ativação da opção *Advanced Menu* do menu *Settings*.

A exploração inicial do High Fidelity foi muito instável. Evidenciou dificuldades nos carregamentos dos domínios, visualização e interação com as entidades, assim como visualização do avatar. Contudo, as mesmas foram atenuadas com atualizações posteriores. Também há que destacar a instabilidade no momento de guardar o trabalho realizado. O High Fidelity guarda as alterações no domínio automaticamente, podendo-se definir diferentes tipos de

backups nas configurações do domínio. Não é referido na documentação oficial o local dos *backups* nem como os restaurar. Assim, a informação dada sobre *backups* que prestámos na seção “Instalação e Configuração” foi obtida através da consulta de fóruns e exploração da máquina de instalação. Foi experienciado que em algumas máquinas os *backups* podem não ser realizados de todos, perdendo-se o domínio assim que se desligue o servidor. A abordagem a que recorremos para contornar esta situação foi criar um *script* que carregasse e configurasse todas as entidades do domínio, o que assumimos é pouco prático e de morosa manutenção. Por exemplo, é preciso posicionar e configurar o modelo no domínio e posteriormente colocar esses dados no código do *script*. Também não está documentada a forma de repor *backups*, tendo sido constatado que é possível fazê-lo através da importação de entidades, do menu *Edit*.

A deteção e verificação de erros e anomalias nos *scripts* de entidades também não é uma tarefa fácil no High Fidelity. Sempre que se pretende efetuar uma alteração num *script* de entidade, há que reiniciar o domínio para que o mesmo seja atualizado. Os erros são reportados no *log* do programa cliente Interface, misturados com os demais registos do Interface. Apesar deste programa cliente disponibilizar um editor de *scripts*, este não permite associá-los a uma entidade.

Foram detetados alguns problemas na inserção de texturas em modelos, bem como na inclusão de invólucros convexos para as colisões entre entidades. Os modelos inseridos com texturas separadas do modelo nem sempre apareciam. Uma forma encontrada para ultrapassar este problema foi incorporar as texturas no ficheiro do modelo aquando da exportação do modelo para o formato *.fbx*. Em relação aos invólucros convexos obtidos recorrendo ao software V-HACD, que por vezes surgiam no High Fidelity desproporcionais em relação ao modelo e apresentavam frechas, deixando passar o avatar ou outra entidade. A solução passou por testar configurações do V-HACD e/ou adicionado elementos geométricos ao invólucro do modelo.

Em algumas situações deparámo-nos com alterações aos nomes das propriedades das entidades em JavaScript (<http://jsref.docs.highfidelity.com/v1.0/docs/entity-properties> [acedido a 10 de outubro de 2016]). A solução encontrada foi criar a entidade no domínio na posição e configuração pretendida e posteriormente exportá-la em formato JSON.

CONCLUSÃO

A tecnologia High Fidelity é bastante promissora, mas ainda em fase de testes, com grande instabilidade de execução e falta de documentação. Tal acrescentou dificuldades e contrariedades na concretização do exemplo aqui apresentado, mas provou ser fonte rica de descoberta: constantes atualizações do software, sobretudo na versão alfa; *bugs*; problemas com os *backups*; problemas com

texturas dos modelos e com os invólucros convexos para colisões; atualização dos *scripts* de entidades no domínio; detecção de erros dos *scripts* no *log* do cliente Interface.

Há, por isso, ainda muito para explorar no High Fidelity. Como pontos interessantes a explorar no futuro, sugerem-se a análise do funcionamento dos *scripts* de atribuição e sua interação com os *scripts* de entidades; as interações específicas para o uso de controladores de mãos e de óculos de realidade virtual; a análise do que sucede quando vários avatares pretendem interagir com a mesma entidade, provocando situações de contenção de recursos; exploração da comunicação do High Fidelity com um servidor externo para, por exemplo, definir e controlar ações ou obter e atualizar informações numa base de dados.

REFERÊNCIAS

- [1] A. Gaspar, “Simulação @ INESC TEC,” 15 fevereiro 2015. [Online]. Disponível: http://www.afceaportugal.pt/2015/eventos/A_MS260_Seminar_INESCP.pdf [acedido a 10 de outubro de 2016].
- [2] “Attaching a Script to an Entity,” High Fidelity, 2015. [Online]. Disponível: <https://readme.highfidelity.com/docs/attaching-a-script-to-an-entity> [acedido a 10 de outubro de 2016].
- [3] “Building Content,” High Fidelity, 2015. [Online]. Disponível: <https://readme.highfidelity.com/docs/getting-started-1> [acedido a 10 de outubro de 2016].
- [4] “The Default Scripts”, High Fidelity, 2015. [Online]. Disponível: <https://readme.highfidelity.com/docs/the-default-scripts> [acedido a 10 de outubro de 2016].
- [5] High Fidelity Inc, “EntityItemProperties,” 2015. [Online]. Disponível: <http://jsref.docs.highfidelity.com/docs/entity-properties> [acedido a 10 de outubro de 2016].
- [6] “High Fidelity,” High Fidelity, [Online]. Disponível: <https://readme.highfidelity.com/docs/architecture-overview> [acedido a 10 de outubro de 2016].
- [7] P. Rosedale, “High Fidelity System Architecture,” 24 abril 2014. [Online]. Disponível: <http://blog.highfidelity.com/blog/2014/04/high-fidelity-system-architecture> [acedido a 10 de outubro de 2016].
- [8] “Server Settings - reference guide,” High Fidelity, [Online]. Disponível: <https://readme.highfidelity.com/v1.0/docs/server-settings-reference> [acedido a 10 de outubro de 2016].
- [9] “Scripts Overview,” High Fidelity, 2015. [Online]. Disponível: <https://readme.highfidelity.com/docs/scripts-digging-deeper> [acedido a 10 de outubro de 2016].
- [10] “Welcome – High Fidelity Documentation,” High Fidelity, 2015. [Online]. Disponível: <https://readme.highfidelity.com/docs/contributing-to-docs> [acedido a 10 de outubro de 2016].
- [11] L. Morgado, 2009. Interconnecting virtual worlds, Journal of Virtual Worlds Research 1 (3): 4-7.