

## **EXPLORAÇÃO DOS BENEFÍCIOS MÚTUOS ENTRE BIM E JOGOS DE SIMULAÇÃO**

**Vítor Mineiro <sup>(1)</sup>, Rosaldo Rossetti <sup>(1)</sup>, Miguel Azenha <sup>(2)</sup>**

(1) Faculdade de Engenharia da Universidade do Porto, Porto

(2) Universidade do Minho, Guimarães

### **Resumo**

Nos últimos anos tem-se verificado um crescimento de jogos sérios e de simulação. Este tipo de jogos tem como principais objetivos apoiar o treino de pessoas ou equipas, estudar os comportamentos dos jogadores perante determinadas situações ou simulação de vários cenários de forma a prever resultados finais de soluções adotadas.

O envolvimento do jogador no jogo torna-se um fator de grande relevância para a eficácia do uso de jogos sérios e jogos de simulação na análise comportamental de pessoas em situações de emergência ou risco, como por exemplo evacuação de um edifício em chamas.

No desenvolvimento de jogos, a modelação do cenário de jogo é realizada normalmente de raiz pelos programadores. Esta tarefa é tanto mais custosa quanto maior for o detalhe do cenário. Contudo, ao replicar um edifício real de raiz, os programadores estão a repetir uma tarefa que já foi executada por arquitetos e engenheiros aquando da sua conceção. Esta duplicação de esforços, para além de desnecessária, por vezes fica aquém do expectável. A dificuldade dos programadores em obterem a informação necessária para a criação do cenário leva a que os cenários se afastem da realidade.

Com a crescente aplicação do conceito BIM nos projetos de edifícios, a representação geométrica do edifício, o mobiliário, os materiais aplicados (propriedades físicas e visuais) passam a estar agrupados digitalmente no projeto do edifício. Desta forma, os projetos BIM tornam-se elementos ricos para o desenvolvimento de jogos sérios e de simulação.

Este trabalho resulta de dissertação de mestrado [1] na qual foi explorada a integração do conceito BIM e jogos de simulação, no sentido de aproveitar as potencialidades do BIM para a criação de cenários de jogos de simulação, e ao mesmo tempo, melhorar a navegação e interação dos modelos BIM através dos motores de jogos. Assim o resultado deste trabalho é uma

solução, baseada numa arquitetura Cliente-Servidor, que permite a interoperabilidade de pacotes de software. A solução engloba o protocolo de mensagens para viabilizar a troca de informação entre os pacotes de software e inclui o processo de criação de cenários de jogo a partir de modelos BIM.

## 1. Introdução

Em 1970, Clark C. Abt [2] introduz o conceito jogos Sérios com uma simples definição que ainda é utilizada nos dias de hoje e que resume os jogos sérios como sendo os jogos cujo principal objetivo não é o entretenimento (em oposição aos tradicionais videojogos). Jogos sérios tem como principal objetivo o treino e formação. No entanto, apesar da diferença do objetivo fundamental do jogo, os videojogos normalmente contêm também uma componente de aprendizagem, pelo menos numa fase inicial do jogo. Assim o jogador executa um determinado conjunto de ações, as quais devem concluir com sucesso, à semelhança do método de ensino tradicional [3].

João Ribeiro usa os exercícios de simulação de emergência realizados em edifícios públicos como objeto de estudo [4], onde realça a dificuldade das pessoas muitas vezes não levarem estes exercícios a sério. Como alternativa, apresenta o uso de sistemas de simulação virtual como fonte de informação útil para os exercícios de evacuação de emergência. Neste sentido explora o conceito de jogos sérios [5]. Para tal, apresenta um simulador de evacuação de emergência, desenvolvido no âmbito do seu estudo, como ferramenta valiosa para o apoio e melhoria dos exercícios reais, dado que os jogos podem ser bastante atrativos e educativos [4]. Uwe Rüppel [6] apresenta um projeto de investigação para desenvolver uma solução de resposta e recuperação para ajudar as equipas de salvamento em encontrar o caminho mais curto em edifícios públicos e disponibilizar-lhes informação importante. A solução consiste em recorrer à informação BIM para mostrar as plantas do edifício em dispositivos móveis e gerar os percursos a seguir pela equipa de salvamento. Sara Freitas [7] realça um conjunto de características que estão presentes nos videojogos e que os tornam um recurso útil no ensino e treino de pessoas e equipas. São realçadas características como uma maior motivação das pessoas que estão a aprender, taxas de conclusão superiores, possibilidade de aceitar novos alunos ou formandos, criação de atividade colaborativas e a aprendizagem e o ganho de experiência direta.

O processamento computacional de hoje em dia torna possível a análise de vários cenários suportados por simulações. Tratando-se de um edifício, as simulações em fase de utilização do edifício podem resultar em elevados custos de correção de falhas/erros de projeto do edifício. Assim, Uwe Rüppel [8-9] propõe o desenvolvimento de uma plataforma que combine o BIM e diferentes simuladores. Através da integração de simuladores de fogo e fumo, simuladores de evacuações e o BIM, é esperado que a plataforma obtenha um ambiente imersivo com resultados mais realistas. Desta forma, Rüppel procura realizar avaliações mais assertivas do modelo BIM e ter uma visualização única de varias simulações.

O presente artigo estará focado numa proposta de fluxo de trabalho para garantia de adequada interoperabilidade entre plataformas BIM e motores de jogo, garantindo a possibilidade do uso de representações fidedignas dos edifícios no processo de conceção de jogos sérios.

## 2. Integração de BIM e Jogos de Simulação

A figura 1 ilustra a visão geral da solução idealizada neste trabalho. Como se pode verificar existem três partes fundamentais. Uma parte é definida por um conjunto de serviços de armazenamento de dados BIM, *Data Providers*, e que poderão ser utilizados pelos pacotes de software BIM, *BIM Providers* e pelos designados *Consumers*. Os *BIM Providers* representam o conjunto de pacotes de software BIM que permitem a manipulação de informação nos objetos, materiais e modelos dos edifícios. Os *Consumers* representam o conjunto pacotes de software que utilizam a informação BIM na sua aplicação, mas que não tem capacidade de manipular essa informação. Neste conjunto fazem parte pacotes de software como motores de jogo, simuladores, ou plataformas para simulação social utilizando inteligência artificial distribuída a partir de entidades autónomas, conhecidas como agentes, que permitem desenvolver aplicações com o objetivo de enriquecer os modelos BIM através da sua análise.

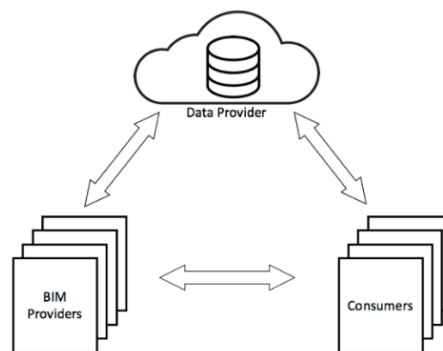


Figura 1: Vista geral da solução.

### 2.1 Arquitetura e implementação

A arquitetura de alto nível da solução idealizada neste trabalho está ilustrada na figura 2. A arquitetura baseia-se numa solução Cliente-Servidor assente em quatro camadas principais e duas camadas de abstração.

A camada de Cliente é caracterizada pelo bloco *Consumers*. Este bloco representa as aplicações clientes que pretendem utilizar os serviços disponibilizados pelos servidores.

A camada Servidor, por sua vez apresenta um conjunto de subcamadas mais complexo que integra o BIM. Nesta camada, o processo de implementação foi dividido em duas fases.

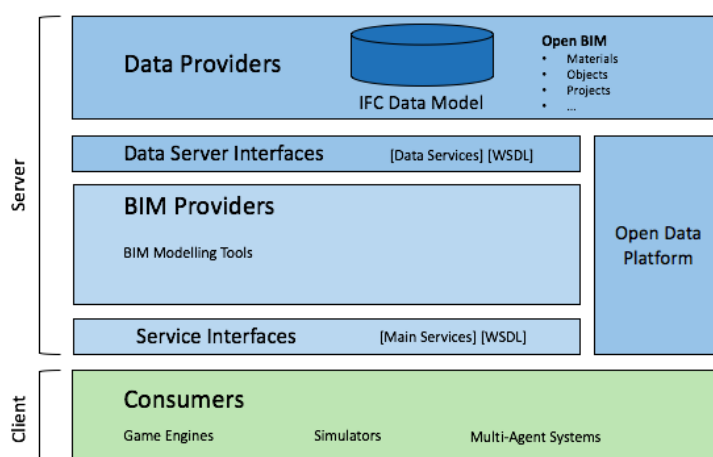


Figura 2: Arquitetura de alto nível.

A primeira fase, que é objeto deste trabalho, consiste no desenvolvimento e implementação da camada *Consumers* das subcamadas *BIM Providers* e *Service Interfaces* (figura 3). Da subcamada *BIM Providers* fazem parte o conjunto de pacotes de software de modelação BIM, como Autodesk Revit ou ArchiCAD, onde são implementados serviços que os *Consumers* utilizarão. Estes serviços são definidos através da linguagem de definição de web-services (WSDL) e tem como principal objetivo abstrair o cliente do servidor e o servidor do cliente.

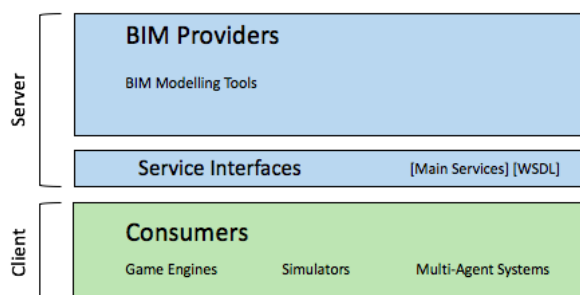


Figura 3: Arquitetura de alto nível (Fase 1).

A fase posterior consiste no desenvolvimento e implementação de uma plataforma *Open Data*, representada pelos blocos *Data Providers*, *Data Server Interfaces* e *Open Data Platform*.

## 2.2 Problemas inerentes à integração de tecnologias e soluções propostas

Ao longo do desenvolvimento deste trabalho surgiram várias dificuldades. Este ponto apresenta um resumo das principais dificuldades apresentadas em [1]. Assim são apresentadas dificuldades na utilização do modelo de dados, bem como no desenvolvimento de *add-in* do Autodesk Revit (plataforma BIM) e no Unity 3D (motor de jogo).

### Modelo IFC (2x3)

O formato mais utilizado na AEC para partilha de informação BIM obedece ao modelo de dados Industry Foundation Classes (IFC) que tem sido desenvolvido e mantido pela buildingSMART.

Contudo, a perda de informação é frequente no processo de exportação e importação do modelo. Existe um conjunto de informação do modelo BIM que não é convertida em entidades IFC quando este é exportado para IFC.

A informação relativa à textura usada num material é um exemplo de informação que se perde quando um modelo é exportado para o formato IFC. Um cenário de jogo onde os objetos apresentam formas e aparências próximas da realidade terá um maior impacto na imersão do jogador, pelo que as informações relacionadas com a aparência dos materiais e elementos são muito importantes para a criação de um cenário de jogo.

O formato adotado foi o formato proprietário FBX (.fbx), desenvolvido pela Kaydara e detido pela Autodesk. O utilizador precisa apenas de exportar o modelo para este formato. Ao exportar o modelo a partir do Revit, é possível selecionar a opção de exportar o modelo considerando o nível de detalhe geométrico. O modelo pode assim ser importado no motor de jogo com facilidade e ser adicionado à cena de jogo como um objeto de jogo. Por sua vez, o objeto de jogo contém os vários elementos do modelo. Cada elemento do modelo contém no seu nome o número de ID usado como referência na plataforma BIM. Assim a informação geométrica do modelo é transferida para o motor de jogo recorrendo ao formato FBX e a informação adicional é passada através de uma arquitetura cliente-servidor, onde o motor de jogo (cliente) efetua pedidos ao software BIM (servidor) para recolher a informação pretendida.

#### Autodesk Revit®

A Autodesk disponibiliza uma API (*Application Programming Interface*) para que os seus utilizadores possam desenvolver *add-ins* para implementar e automatizar algumas das suas rotinas de modelação. Apesar da API suportar a framework .NET 4.5 existem muitas funcionalidades que não estão exploradas. O uso de tarefas assíncronas é um desses casos.

A complexidade da estrutura de dados do proprietário utilizada no Revit revelou ser uma das maiores dificuldades deste trabalho. A quantidade de entidades e o facto de não haver uma documentação clara sobre a estrutura e relacionamento dessas entidades, tornou-se num obstáculo difícil de contornar.

As atualizações anuais do Revit e das respetivas API's mostraram incompatibilidades entre versões. Estas incompatibilidades levaram a que rotinas implementadas em versões anteriores não pudessem ser implementadas de igual forma na versão atual. Por exemplo, na versão do Revit 2012, as informações sobre a textura de um determinado material estavam acessíveis através de métodos disponibilizados na API. A partir da versão do Revit 2014, a API sofreu alterações de tal forma que estas informações deixaram de estar acessível pela API.

#### Unity 3D

No sentido de ultrapassar estas dificuldades na conexão dos dois sistemas via web-services a solução foi implementada de acordo com o protocolo TCP (*Transmission Control Protocol*). Assim, esta solução permite desenvolver aplicações de servidor e cliente de forma independente do protocolo de comunicação. Desta forma evita-se a necessidade de alterar os serviços já implementados caso novos protocolos de comunicação sejam implementados.

Para além das dificuldades relacionadas com os materiais já mencionadas, a aplicação dos materiais no modelo importado apresentou também algumas falhas. O modelo pode ser exportado utilizando o nível de detalhe gráfico dos elementos. Um elemento composto por mais do que um componente, por exemplo uma porta, ao ser exportado sem ser considerado o seu nível de detalhe gráfico, transforma as *mesh*'s de cada componente numa única *mesh*. Desta forma, quando o elemento é importado para o cenário de jogo terá uma única *mesh*. Se o mesmo elemento for exportado considerando o nível de detalhe gráfico, as *mesh*'s de cada elemento ficam agrupadas no mesmo elemento, mas de forma independente.

Para contornar este problema foi implementado o serviço "GetMaterials" para adquirir os materiais de um dado elemento. Assim o Unity efetua um pedido ao Revit para saber quais são os materiais aplicados num dado elemento. Posteriormente o Unity cria os materiais no seu editor de forma a replicar a biblioteca de materiais do Revit no Unity.

Outro aspeto relacionado com os problemas na importação dos materiais é a ausência de padronização na estrutura e propriedades dos materiais nos diferentes pacotes de software. O uso de diferentes propriedades para caracterizar as características dos materiais origina ambiguidade na interpretação das propriedades correspondentes entre as duas plataformas. As propriedades de *shininess* e *smoothness* dos materiais são um dos exemplos onde o Revit e Unity apresentam designações diferentes para a caracterização destas propriedades. Por sua vez, estas propriedades têm uma enorme importância na caracterização e renderização do cenário. A incorreta definição destas propriedades pode causar um resultado completamente diferente do esperado.

### 2.3 Processo de Integração

A solução desenvolvida segue uma lógica Cliente-Servidor, o que originou a necessidade de processo de integração separado em etapas e atividades quer no lado do Cliente quer no lado do Servidor.

#### Servidor

A primeira etapa do processo de integração é definida pela exportação do modelo do edifício para o formato FBX, tendo o cuidado de exportar o modelo com o nível de detalhe gráfico selecionado. Uma vez exportado o modelo é necessário iniciar o servidor para que o cliente possa aceder à informação dos elementos do modelo.

#### Cliente

A figura 4 apresenta a sequência de atividades a seguir para criação do cenário de jogo no lado do cliente. As atividades foram agrupadas em quatro fases distintas. Assim temos as três primeiras fases para preparação do cenário e a última fase para o desenvolvimento do jogo.

A primeira fase, designada "Model Geometry", inicia-se com a importação do modelo do edifício em formato FBX para o projeto, seguido da sua adição à cena de jogo. Com estes passos é adicionado um objeto de jogo que de agora em diante será designado de cenário. Este objeto de jogo contém todos os elementos do edifício que por sua vez são também objetos de jogo. Contudo, é necessário estender a informação destes elementos de forma a integrar a informação BIM do edifício. Para tal é executado o passo, "Parse Elements". Neste passo, os objetos de

jogo contidos no cenário são estendidos com um componente "ModelObject" que permitirá guardar a informação BIM do objeto. São adicionados o número de identificação e o nome do objeto, de forma a poder ser referenciado no servidor.

A segunda fase, "BIM Model Info", consiste na atualização da informação BIM dos objetos do cenário. Nesta fase é esperado que o cliente complete a informação dos objetos de jogo incluídos do cenário. Assim, através da atividade "Gather Elements Info", o cliente solicita a informação de cada objeto incluído no cenário e atualiza os dados do objeto com a informação BIM recebida. Após os objetos de jogo estarem atualizados, são atualizados os materiais aplicados nos objetos. Para tal são executadas as atividades "Gather Materials" e "Update Materials Library and Game Object Materials". A primeira consiste no pedido dos materiais (e suas propriedades) de um determinado elemento. A segunda consiste no processamento e tratamento da resposta do servidor. Assim se algum dos materiais recebidos ainda não estiver replicado na biblioteca de materiais do cliente, este é criado. Se o material existir, cabe ao utilizador decidir se atualiza ou não a informação do material. Desta forma é possível manter no motor de jogo uma biblioteca de materiais que é o reflexo da biblioteca do software BIM. Uma vez atualizados os elementos e os materiais do modelo, a conexão ao servidor pode ser terminada, uma vez que não estão implementados outros serviços.

A terceira fase, "Model Improvement", consiste no refinamento do modelo de forma a corrigir erros oriundos do formato FBX. A primeira atividade desta fase consiste em corrigir os materiais aplicados em elementos simples. Posteriormente é preciso aplicar manualmente os materiais a cada *submesh* de elementos compostos. Assim, o utilizador deve corrigir os materiais dos elementos compostos para cada tipo de elemento composto. Para agilizar o processo, foi implementada uma rotina que permite replicar a aplicação dos materiais de um objeto aos restantes objetos desse mesmo tipo.

Por fim, vem a fase de desenvolvimento do jogo em si, "Game Development". Nesta fase o utilizador deve implementar os scripts com as interações pretendidas de acordo com o objetivo do jogo. Na figura 4 foram destacados os *scripts* associados à abertura e fecho de portas, ao acender e apagar luzes, ao jogador e à lógica de jogo, uma vez que são as interações que permitem esta solução destacar-se das soluções existentes no mercado.

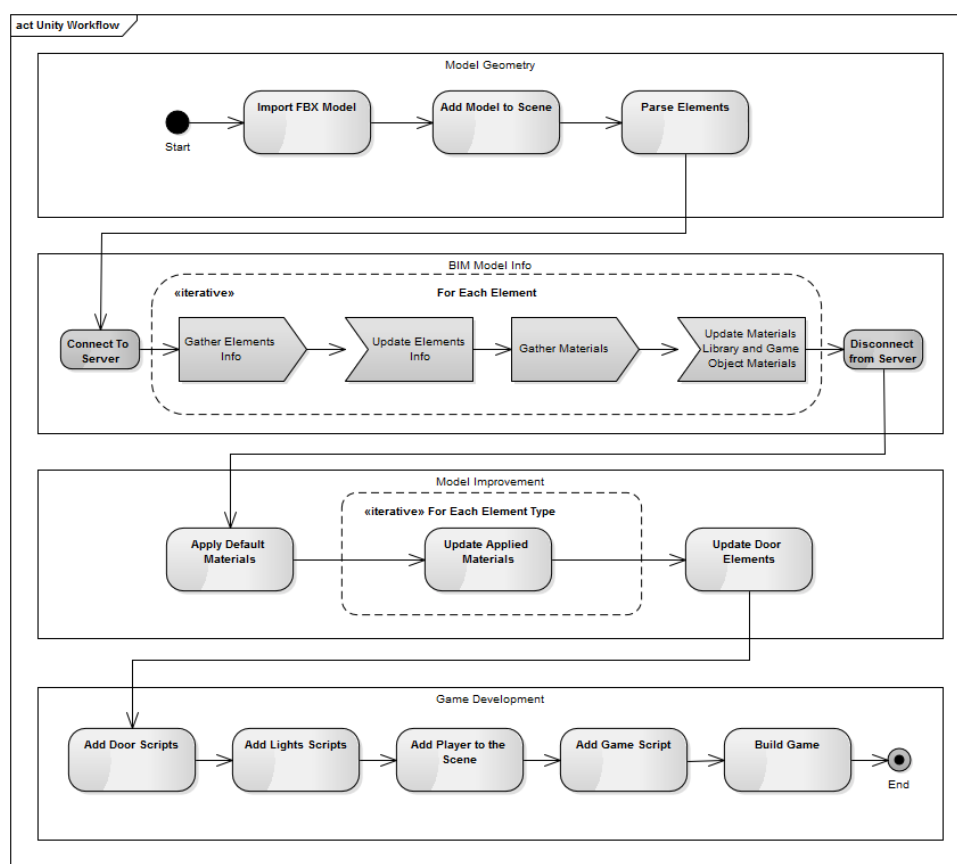


Figura 4: Diagrama de atividades da solução.

### 3. Casos de Estudo

No sentido de avaliar a solução desenvolvida neste trabalho foram realizados quatro casos de estudo. Os casos de estudo consistem em executar um conjunto de ações para alcançar um fim comum, que neste caso será um ambiente virtual de um edifício onde o utilizador possa navegar livremente pelo edifício segundo uma vista de primeira pessoa. Cada caso de estudo usará um conjunto de ferramentas diferentes e será avaliado num conjunto de parâmetros. Posteriormente é feita uma comparação entre os resultados obtidos em cada caso de estudo no sentido de obter uma avaliação relativa entre os vários conjuntos.

Cada caso de estudo foi avaliado de acordo com um conjunto de métricas de forma a avaliar a solução individualmente e posteriormente comparar com as restantes soluções. As métricas definidas para avaliação das soluções são: (i) nível de esforço necessário; (ii) necessidade de tarefas manuais; (iii) quantidade de ferramentas necessárias; (iv) dependência de software; (v) experiência do utilizador; (vi) nível de realismo do cenário (renderização); (vii) interações do utilizador com o modelo; (viii) expansão e integração com outros sistemas.

Nos casos de estudo deste trabalho utilizaram-se ferramentas utilizadas atualmente na AEC para a exploração do modelo e por fim a solução desenvolvida. Assim, o primeiro consiste em utilizar a ferramenta de visualização avançada disponível com o Autodesk Revit. No segundo



caso é utilizado o software BIMx, da empresa Graphisoft. O caso três consiste em utilizar o motor de jogo Stingray, da Autodesk. O Stingray foi adquirido pela Autodesk para promover a exploração em tempo real, animação e interação do modelo do edifício. Assim este caso é o caso de estudo que mais se aproxima da solução desenvolvida neste trabalho. Por fim, o último caso consiste em avaliar a solução desenvolvida usando os pacotes de software Autodesk Revit® e Unity3D.

Nos casos de estudo um, três e quatro foi utilizado o modelo avançado Revit disponibilizado pela Autodesk no seu sítio Web. A escolha deste modelo resultou da necessidade de utilizar um modelo com complexidade geométrica elevada de forma abranger o maior número de situações possíveis. Ao mesmo tempo era necessário que os elementos tivessem um nível de desenvolvimento tal que permitisse extrair informações sobre a composição dos elementos e materiais aplicados. No caso de estudo 2, dada a dependência do formato da ferramenta, foi utilizado outro edifício.

#### **4. Análise de Resultado**

No ponto anterior cada caso de estudo foi analisado individualmente. Os resultados obtidos para cada critério analisado, em cada uma das soluções, encontram-se resumidos na Tabela 1.

A solução do Revit torna-se uma solução válida quando se pretende realizar simples visualizações e explorações do modelo. Nestas situações o utilizador procura realizar ações rápidas de visualização sem precisar de realizar interações com o modelo. Como se pretende que estas ações sejam rápidas, o facto de o utilizador ter de utilizar mais do que um *software* para obter um ambiente virtual torna-se uma desvantagem. Contudo, a navegação do utilizador disponível nesta solução é pouco prática para o utilizador, pelo que neste aspeto as restantes soluções são uma melhor opção. Se o utilizador pretender navegar pelo modelo com um nível de renderização aceitável as restantes soluções conseguem melhores resultados, uma vez que para obter nível de renderização elevados a navegação pelo modelo fica impraticável.

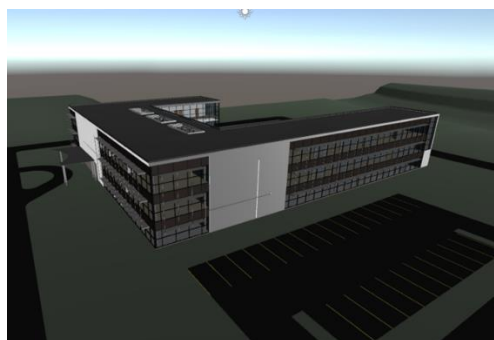
Já o BIMx consegue um nível de renderização aceitável ao mesmo tempo que permite uma navegação na perspetiva de primeira pessoa bastante fluida. No entanto tem o inconveniente de obrigar a que o modelo seja exportado a partir do ArchiCAD, limitando a escolha dos pacotes de software de modelação da equipa projetista do modelo. As interações do utilizador com o modelo são também limitadas. O utilizador pode atravessar portas sem que haja colisões e impedimentos entre a porta e o utilizador o que torne o ambiente menos realista do desejável.

A solução do Stingray por sua vez apresenta um bom resultado na renderização do ambiente e permite a implementação de diversas interações entre o modelo e o utilizador. No entanto a necessidade de o modelo passar por quatro ferramentas (ver [1]) para gerar o ambiente virtual navegável por qualquer utilizador torna a solução dissuasora dos utilizadores menos experientes.

Tabela 1: Análise comparativa das soluções.

	Autodesk Revit®	BIMx	Stingray	Unity 3D
Esforço	Baixo	Baixo	Elevado	Médio
Correções no modelo	Não	Não	Sim	Sim
N.º de ferramentas	1	1	4	2
Dependências de software	Sim	Sim	Sim	Não
Conhecimento de utilização	Sim	Não	Não	Não
Renderização	Baixo	Aceitável	Bom	Aceitável
	Luzes	Não	Não	Sim
Interações	Portas	Não	Não	Sim
	Info	Sim	Limitada	Sim
Integração de outros sistemas (simuladores, sistemas multi-agentes, ...)	Não	Não	Sim	Sim

Por sua vez, a solução proposta neste trabalho e que foi implementada no caso de estudo do Unity 3D, permite que a equipa projetista do modelo escolha as suas próprias ferramentas de modelação BIM, desde que estas implementem a solução proposta. O software de edição do ambiente virtual fica também ao critério dos utilizadores. Desta forma a solução promove a interoperabilidade entre pacotes de software. O resultado da renderização do ambiente não é tão bom como a renderização da solução do Stingray mas é aceitável. Contudo, o esforço necessário para gerar o ambiente virtual é bastante inferior ao esforço requerido pela solução do Stingray, pelo que torna a solução mais apelativa para os utilizadores. Já a nível de expansibilidade a solução proporciona a possibilidade de integrar com sistemas simuladores e multiagentes à semelhança do Stingray. As figuras 5.a) e 5.b) apresentam o ambiente virtual resultante do processo definido na solução. Conforme é possível verificar o resultado obtido é um ambiente virtual bastante aceitável mesmo sem depender grandes esforços no Unity a configurar as propriedades de iluminação, fator que tem grande influência no realismo do cenário.



a) Vista exterior



b) Vista interior

Figura 5: Ambiente virtual final em Unity 3D.

## 5. Conclusões e trabalho futuro

A solução resultante deste trabalho comprova que a integração do BIM nos jogos de simulação contribui para um desenvolvimento de ambientes virtuais realistas. Os criadores vêem reduzido o esforço de criação de ambientes virtuais realistas, que por sua vez incluem significativas quantidades de informação sobre materiais e outras propriedades. Como resultado é possível definir comportamentos também mais realistas. Por sua vez, a integração de jogos de simulação no BIM contribui para uma melhor navegação e exploração do edifício, ao mesmo tempo que se tira proveito do potencial dos jogos de simulação para apoiar as tomadas de decisão da equipa projetista do edifício e dos restantes intervenientes, através de simulações e análises efetuadas nos jogos de simulação. Confirmou-se, portanto, que existe benefício mútuo na integração do BIM nos jogos de simulação e os jogos de simulação no BIM.

Com este trabalho foi possível identificar diversas dificuldades que, embora tenham sido ultrapassadas, merecem ainda um aprofundamento no sentido de melhorar os resultados obtidos. Assim, reúne-se os trabalhos futuros em três grupos: Interoperabilidade, Plataforma Open Data e Jogos BIM. A implementação da arquitetura Cliente-Servidor SOAP através de Web Services da WCF reflete-se numa maior adoção da solução por parte da indústria. O desenvolvimento e implementação de uma plataforma Open Data deverá ser capaz de dissociar o modelo do edifício dos pacotes de *software* BIM, integrando numa base de dados relacional o modelo de dados IFC. Neste sentido a plataforma funcionaria como repositório de elementos BIM ao mesmo tempo que poderia ser utilizada como uma plataforma colaborativa durante a conceção do projeto do edifício. Por sua vez, a plataforma serviria também de biblioteca de materiais com as suas propriedades físicas e aparência. Outra área que pode servir de base para futuros trabalho consiste no desenvolvimento de jogos de simulação que se adequem às diversas fases de conceção do projeto, incluindo as fases operacionais do edifício.

## Referências

- [1] V. Mineiro, Exploração dos benefícios mútuos entre BIM e jogos de simulação. Dissertação de Mestrado. Porto, Faculdade de Engenharia da Universidade do Porto, Portugal, 2016.
- [2] C. A. Clark, *Serious Game*, 1970.
- [3] D. R. Michael e S. L. Chen, “Serious games: Games that educate, train, and inform,” 2005.
- [4] J. Ribeiro, J. E. Almeida, R. J. F. Rossetti, A. Coelho e A. L. Coelho, “Using serious games to train evacuation behaviour,” em *7th Iberian Conference on Information Systems and Technologies (CISTI)*, 2012.
- [5] E. Cordeiro, A. L. Coelho, R. J. F. Rossetti e J. E. Almeida, “Human behavior under fire situations—portuguese population,” em *Fire and Evacuation Modeling Technical Conference*, 2011.
- [6] U. Rüppel e K. M. Stuebbe, “BIM-based indoor-emergency-navigation-system for complex buildings,” *Tsinghua Science & Technology*, vol. 13, p. 362–367, October 2008.
- [7] S. Freitas, “Using games and simulations for supporting learning,” em *Learning, media*

- and technology, 2006.
- [8] U. Rüppel e P. Abolghasemzadeh, “Bim-based immersive evacuation simulations,” em 18th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering, 2009.
  - [9] U. Rüppel e K. Schatz, “Designing a bim-based serious game for fire safety evacuation simulations,” *Advanced Engineering Informatics*, vol. 25, p. 600–611, 2011.