

---

# Online Gym: um ginásio virtual 3D integrando a Kinect – análise comparativa de bibliotecas de suporte

**Fernando Cassola**

INESC TEC / UTAD Universidade  
de Trás-os-Montes e Alto Douro  
Porto - Portugal  
fernandocassola@gmail.com

**Benjamim Fonseca**

INESC TEC / UTAD Universidade  
de Trás-os-Montes e Alto Douro  
Porto - Portugal  
benjaf@utad.pt

**Leonel Morgado**

INESC TEC / UTAD Universidade  
de Trás-os-Montes e Alto Douro  
Porto - Portugal  
leonelm@utad.pt

**Paulo Martins**

INESC TEC / UTAD Universidade  
de Trás-os-Montes e Alto Douro  
Porto - Portugal  
pmartins@utad.pt

**Hugo Paredes**

INESC TEC / UTAD Universidade  
de Trás-os-Montes e Alto Douro  
Porto - Portugal  
hparedes@utad.pt

**Fausto de Carvalho**

PT Inovação  
Aveiro - Portugal  
cfausto@ptinovacao.pt

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

**Resumo**

A sincronização online de ginástica potencia novas possibilidades para melhorar o bem-estar físico e social das pessoas com restrições de deslocação. A nossa proposta passa pela criação de uma plataforma 3D - Online Gym - que permita que os utilizadores interajam e participem em sessões online de ginástica em grupo através do Microsoft Kinect.

No presente artigo, com vista à concretização desta proposta, analisamos três alternativas tecnológicas para a implementação de serviços de deteção de movimentos que possam ser integrados em plataformas gráficas multiutilizador. Nos resultados, expõem-se as características de cada uma e o impacte respetivo para sua utilização na concretização desta proposta ou outras iniciativas similares.

**Palavras-chave**

ginástica online, Second Life, OpenSimulator, Mundos Virtuais, Kinect, Captura de movimentos, Interação Humano computador, interfaces naturais

**Palavras-chave ACM**

H.5.1 Multimedia Information Systems; H.5.2 User Interfaces

## **Introdução**

A ampla disponibilidade de acesso à Internet por banda larga, inclusive para populações como os idosos e os portadores de deficiência, possibilita o surgimento de novos serviços online de cooperação para a interação social e jogos. Em paralelo, tem vindo a ser reconhecido que as plataformas tecnológicas para desenvolver a atividade física podem promover o bem-estar físico e mental das pessoas idosas [1]. Estes dois aspetos já confluíram em vários produtos, tendo-se constatado que a possibilidade de realizar estas atividades online em grupo, com inclusão de especialistas como monitores, formadores e/ou treinadores, pode contribuir para envolver as populações nestas práticas, com um nível adequado de aconselhamento profissional [2].

Expõe-se neste trabalho uma proposta de expansão da área da ginástica online em grupo, tirando partido da presente disponibilidade a baixo custo de dispositivos de captura de movimentos, tais como o Microsoft Kinect, o PrimeSense Sensor, o ASUS Xtion, entre outros [3]. Consiste tal proposta na criação de um "ginásio online", que corresponda a um espaço virtual tridimensional onde vários utilizadores - fisicamente distantes - possam realizar em conjunto, orientados por um monitor, formador ou treinador, atividades de ginástica em grupo, onde a deteção de movimentos para esse espaço virtual seja efetuada por um dispositivo de baixo custo.

Como primeiro passo para concretização desta proposta, estabeleceu-se um protótipo, que assume para cada utilizador acesso à Internet, um computador pessoal e um Microsoft Kinect enquanto dispositivo de baixo custo [4]. Para esse protótipo teve-se em

consideração as plataformas de mundos virtuais com suporte à renderização 3D e comunicação entre utilizadores, para que o trabalho de estudo e desenvolvimento do protótipo se centre nas características específicas da proposta de ginásio online – não nas particularidades genéricas da renderização e gestão de utilizadores.

Neste trabalho, como primeiro passo para concretização do protótipo, apresentamos uma análise comparativa das ferramentas de software atuais que permitem interligar a captura de movimentos e a sua transmissão para um ambiente externo multiutilizador. Esta análise poderá ser igualmente útil para outras iniciativas que pretendam desenvolver sistemas multiutilizador apoiados na captura de movimentos e em mundos virtuais ou outras plataformas 3D.

## **Dispositivos de captura de movimentos**

Existem vários dispositivos com elevadas capacidades de captura de movimentos, em particular muitos se tornaram conhecidos do grande público devido à sua utilização na indústria do cinema. No entanto, os custos de aquisição são elevados, a configuração física é complexa e necessitam de trabalhos elaborados de configuração, sendo inadequados para os contextos domésticos em causa neste trabalho [5]. Os dispositivos tecnológicos de baixo custo, já referidos, viabilizam o recurso a captura de movimentos no contexto doméstico. A relevância destas interfaces deve-se essencialmente a dois fatores: o primeiro de cariz económico, pelo custo relativamente baixo de aquisição do equipamento; o segundo pela componente da usabilidade, porque a facilidade de instalação, configuração e utilização é elevada e bastante intuitiva [6]. Estas novas interfaces têm vindo a ser analisadas

(em conjunto com outras como as táteis) por um leque alargado de investigadores, no âmbito da área conhecida por NUI ("Natural User Interfaces"). Como características determinantes das interfaces integrantes desta área, elencam-se a sua natureza intuitiva (possibilidade de utilização com pouco treino); a sua flexibilidade (possibilidade de adaptação às necessidades); e a sua fluidez (possibilidade de interação com a interface quase sem perceber que a estamos a usar)[6].

Existem vários dispositivos com estas características que utilizam diferentes tipos de captura de movimentos, entre estes o Microsoft Kinect [4], o PrimeSense Sensor [7] ou o ASUS Xtion [7]. Estes sistemas possuem câmaras que permitem capturar os movimentos bidimensionais de todo o corpo de forma direta e sensores de profundidade ou outros métodos de deteção da mesma, proporcionando dados diretamente em 3D. Outros dispositivos também proporcionam dados para reconhecimento de gestos, inferindo-os através de sensores inerciais e/ou câmaras, mas sem deteção de profundidade, como sejam o Wii Remote [8] o PlayStation Move [9] ou o EyeToy [10].

Entre estes, o Kinect tem sido alvo de particular atenção na área do 3D, graças às suas características de rápido reconhecimento do corpo/pose sem necessidade de configuração prévia, ao baixo custo do equipamento, à fiabilidade e à velocidade de medição dos dados em 3D [11].

Tecnologicamente além de outros componentes que não são relevantes para o presente trabalho, o Kinect "contém um sensor de profundidade, uma câmara a

cores e uma matriz de quatro microfones que proporcionam a captura do movimento de todo o corpo em 3D, o reconhecimento facial e ainda capacidades de reconhecimento de voz" [12]. Isto permite que as ferramentas de software usem e trabalhem os dados enviados pelo Kinect para rastrear os movimentos de diferentes pontos do esqueleto do corpo humano, incluindo pernas, braços, mãos e cabeça [13].

### **Bibliotecas de captura de movimentos**

Para processar os dados fornecidos pelo Kinect (ou outros dispositivos similares), as aplicações podem recorrer a várias bibliotecas de middleware na área das NUI (doravante, "bibliotecas NUI"). Estas bibliotecas fazem a gestão do processamento do fluxo de dados de baixo nível e disponibilizam-nos às aplicações. Duas dessas bibliotecas NUI são o Microsoft Kinect SDK [4] e a OpenNI [14], esta última desenvolvida por um consórcio sem fins lucrativos com o mesmo nome, que tem por objetivo padronizar a compatibilidade e a interoperabilidade das interfaces naturais.

### **Microsoft Kinect SDK**

Este SDK (Software Development Kit) fornece as ferramentas e a API (Application Program Interface) necessárias para desenvolver aplicações nativas à plataforma Windows usando o Kinect. Em termos de arquitetura este componente pode ser dividido em cinco áreas. Uma primeira área, que representa o hardware, é composta por todos os dispositivos físicos do Kinect, incluindo os sensores e o *hub* USB através do qual se efetua a ligação ao computador. Outra área é a parte dos controladores, que representam um conjunto de software que é instalado no sistema operativo durante o processo de instalação da Microsoft Kinect SDK. Existem também componentes de áudio e

vídeo, ao nível do seguimento do esqueleto, da cor, da profundidade e do som. Fica integrada nas API padronizadas do sistema operativo Windows, ao nível do som e do reconhecimento da fala.

### OpenNI

A biblioteca OpenNI é uma *framework* que trabalha entre a camada física e as bibliotecas de middleware do sistema operativo, permitindo o desenvolvimento de aplicações que utilizem dispositivos de interação natural de forma independente do hardware [11]. A interface de aplicação fornece acesso aos sensores de profundidade compatíveis com OpenNI. Concretamente permite inicializar os sensores e receber os dados acerca da profundidade, das cores (RGB) e do *streaming* de vídeo dos infravermelhos.

Em termos de arquitetura, esta pode ser dividida em quatro grandes áreas. Uma primeira área, que representa o hardware, é composta por todos os dispositivos físicos, incluindo a ligação ao computador. Outra área inclui os controladores de ligação aos dispositivos, uma camada de abstração do hardware, o core e a API. Uma terceira área corresponde a uma biblioteca de middleware, que é exemplificada num conjunto de aplicações padronizadas que exploram as potencialidades do SDK, nomeadamente rastreamento do corpo, reconstrução 3D, reconhecimento de objetos, ferramentas de análise, entre outras. A última camada desta arquitetura é a das aplicações propriamente ditas, que corresponde ao software que irá interagir com o SDK em causa.

### OpenNI vs. Microsoft Kinect SDK

Na **Tabela 1** apresentamos uma comparação entre estas bibliotecas. Resumidamente, uma distinção

fundamental é que o dispositivo de origem dos dados, no caso da biblioteca OpenNI, pode ser um entre vários, enquanto o Microsoft Kinect SDK se destina exclusivamente ao Kinect. Além deste aspeto, a OpenNI é disponibilizada em código aberto e aplicável a vários sistemas, contrariamente ao Kinect SDK da Microsoft que é fornecido em código fechado exclusivamente para sistemas Windows. Por outro lado, a OpenNI necessita de calibração para iniciar a captura, requerendo que o utilizador se encontre parado em frente ao dispositivo para iniciar o processo, enquanto o Kinect SDK permite iniciar automaticamente a captura.

	OpenNI	Microsoft SDK
Dispositivos suportados	Microsoft Kinect, PrimeSense Sensor, ASUS Xtion, outros	Microsoft Kinect
Código-fonte	Aberto	Fechado
N.º de pontos do esqueleto	Captura até 20 pontos distintos	Captura até 20 pontos distintos
Inicialização da captura	Requer calibração	Imediata
Sistema operativo	Linux, Windows, iOS	Windows

**Tabela 1** - Quadro comparativo entre OpenNI e Kinect SDK

## Ferramentas de desenvolvimento de software

Os dados disponibilizados por ambas as bibliotecas, que trabalham ao nível do middleware, são de baixo nível. Existem camadas de software que processam estes dados para facilitar a sua utilização pelas aplicações, com maior grau de abstração. Apresentamos nesta secção três desses pacotes de software.

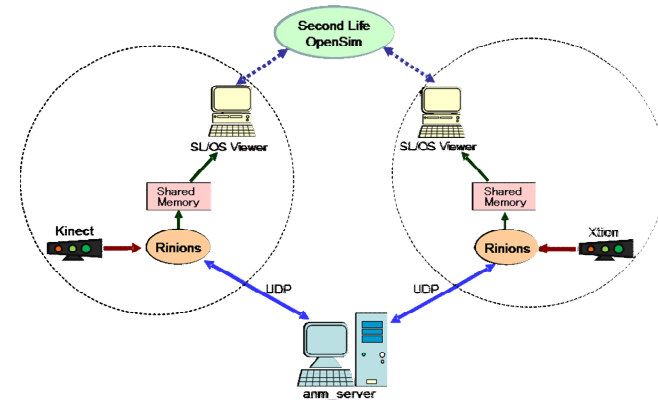
### RINIONS

Este pacote de software, desenvolvido no laboratório de Redes e Sistemas da Universidade de Tóquio, tinha na sua versão original o nome SLKinect2 [15]. Tem por foco transferir os movimentos de animação realizados pelo utilizador usando o Kinect para um software-cliente ("viewer") de mundos virtuais do tipo Second Life ou OpenSimulator e propagá-los para *viewers* remotos. Recorre para o efeito a uma alteração ao código-fonte dos *viewers*. Está disponível em código-fonte aberto, contudo sem documentação de referência [16].

Baseia-se na utilização de dois programas, além dos *viewers* dos utilizadores e do servidor Second Life/OpenSimulator: um cliente que captura os movimentos através do Kinect e um servidor que os propaga entre *viewers* (ver **Figura 1**).

O cliente Rinions permite efetuar a captura de movimentos através das duas bibliotecas apresentadas (Kinect SDK e OpenNI). As versões alteradas de *viewers* de Second Life/OpenSimulator para integração com o Rinions estão disponíveis online para vários desses *viewers* (*viewer* oficial do Second Life, Imprudence, FireStorm, etc.): consistem numa abordagem de memória partilhada para comunicação

entre a deteção de movimentos efetuada pelo cliente Rinions e a reprodução dos mesmos no *viewer* do mundo virtual. Os *viewers* alterados leem ciclicamente na memória partilhada um ficheiro de poses de esqueleto (formato BVH), cujo conteúdo é constantemente reescrito pelo cliente Rinions, refletindo a captura da posição do utilizador pela Kinect.



**Figura 1** - Modelo arquitetónico do software Rinions (adap.) [16]

### FAAST (Flexible Action and Articulation Skeleton Toolkit)

Este pacote de software, um projeto do Institute for Creative Technologies da University of Southern California, interpreta os dados fornecidos pela OpenNI, efetuando o reconhecimento de aspetos como a inclinação ou torção do corpo e fazendo-os corresponder a animações previamente definidas (o código-fonte ainda não está disponível, mas foi anunciada a intenção dessa disponibilização) [17]. Não inclui funcionalidades para propagação dos movimentos

a outros sistemas. Como não produz resultados genéricos, só se revela adequado a cenários em que os movimentos a apresentar possam constar de um conjunto pré-determinado [18].

O FFAST inclui um programa de demonstração que reconhece movimentos pré-definidos e os faz corresponder a teclas de um teclado. Este exemplo visa expor que é possível ao utilizador controlar jogos utilizando o corpo, mesmo que esses jogos não tenham outra opção de entrada senão por teclado. Neste exemplo, é igualmente demonstrado que a identificação de ações é configurável sem necessidade de parar a aplicação. Nomeadamente, para ajustar a sensibilidade do sensor às características e preferências do utilizador [17].

### **NUILIB**

Este pacote de software, de código-fonte aberto, processa os dados fornecidos pelos dispositivos NUI e associa-os a uma estrutura de dados mais abstrata, visando apoiar a integração destes dispositivos em aplicações.

Como ferramenta de mapeamento, as aplicações podem trabalhar através dessa estrutura de dados com posições e gestos concretos das articulações, evitando a complexidade de processamento dos dados de baixo nível. Está disponível em código-fonte aberto e com documentação de referência [19].

Associado a esta biblioteca existe um exemplo de utilização para as plataformas Second Life/OpenSimulator: um *viewer* específico, que se chama Armadillo. Este *viewer* recorre à NUILIB para apresentar ao utilizador uma funcionalidade similar à do FFAST: reconhece gestos concretos e converte-os

em comandos de teclado ou rato (note-se que o exemplo do FFAST é genérico e este exemplo do NUILIB é específico para estes mundos virtuais).

### **Comparação das ferramentas**

Na Tabela 2 identificamos as diferenças principais entre as ferramentas apresentadas. Ao nível da renderização de movimentos livres (isto é, os que forem feitos pelo utilizador na prática de ginástica, sem restrição prévia) em plataformas de mundos virtuais, o único que proporciona suporte direto para tal é o Rinions. No caso do FFAST, o foco é na identificação de gestos pré-definidos, não em movimentos livres, embora, dado ser de código-fonte aberto, se possa recorrer às estruturas de dados internas para tentar obter dados desses movimentos. No caso da NUILIB, os dados dos movimentos livres estão disponíveis na estrutura de dados da biblioteca. Em qualquer destes dois casos, não existe suporte à partida para reprodução desses dados no mundo virtual.

Por outro lado, para identificação automática de movimentos pré-definidos, tais como andar, voar ou olhar, mas também como poses ou movimentos de sessões de ginástica pré-determinados, o Rinions é o único que não suporta essa identificação.

Como forma de integrar as ferramentas apresentadas com as plataformas de mundos virtuais (Second Life / OpenSimulator) o Rinions requer uma adaptação aos *viewers* tradicionais, já disponível para vários deles e de pequena extensão; a FFAST e a NUILIB requerem a implementação nos *viewers* de toda a reprodução em gestos dos dados por elas identificados. No caso de identificação de gestos específicos, a reprodução destes

em mundos virtuais tem igualmente de ser implementada, neste caso para todas as bibliotecas.

Testámos as aplicações de demonstração destas bibliotecas e verificámos que, sem surpresa, o Rinions responde com exatidão e pouco atraso, visto que se limita a reproduzir diretamente as posições e movimentos das articulações no espaço, enquanto a FFAST e a NUILIB apresentam maior atraso e alguma imprecisão, em virtude de estarem a tentar interpretar os dados diretos e fazê-los corresponder a gestos pré-definidos.

	<b>RINIONS</b>	<b>FFAST</b>	<b>NUILIB</b>
Identificação de movimentos pré-definidos	Não	Sim	Sim
Integração com Second Life / OpenSimulator	<i>Viewer</i> adaptado (pequena extensão)	É necessário implementar nos <i>viewers</i> toda a reprodução dos gestos	É necessário implementar nos <i>viewers</i> toda a reprodução dos gestos
Código-fonte	Aberto	Anunciado mas não disponível	Aberto
Software base	Microsoft Kinect SDK / OpenNI	OpenNI	Microsoft Kinect SDK / OpenNI

Tabela 2 - Quadro comparativo das ferramentas apresentadas

### **Conclusões e trabalho futuro**

Tendo em conta as ferramentas apresentadas e as respetivas características, podemos concluir que, para reprodução em plataformas de mundos virtuais de gestos livres, o pacote Rinions é o que mais diretamente apresenta uma solução, estando inclusivamente disponível um exemplo da sua utilização para as plataformas Second Life e OpenSimulator.

Já para identificação de gestos concretos, pré-definidos, este pacote não cumpre tal funcionalidade, que é prestada pelas bibliotecas FFAST e NUILIB. A comparação precisa entre estas é presentemente limitada, em virtude de não estar disponível o código-fonte nem a documentação da FFAST. Pelas funcionalidades visíveis, dispomos na FFAST apenas da identificação de gestos, enquanto na NUILIB se acrescenta igualmente a possibilidade de consultar os dados de suporte a essa identificação (contudo, tal apreciação pode vir a equilibrar-se quando da disponibilização do código-fonte da FFAST, se ocorrer).

Em ambos estes projetos (NUILIB e FFAST), percebe-se que o objetivo principal não é a reprodução de gestos livres em plataformas 3D, mas a sua identificação. Destinam-se a apoiar o desenvolvimento de aplicações que tenham necessidade de identificar gestos, independentemente do destino dessa identificação. Por exemplo, para controlo de aplicações ou jogos, onde se podem associar movimentos pré-definidos a comandos específicos. Se a aplicação ou jogo pretender fazer corresponder um gesto de avatar a um gesto pré-identificado, compete à aplicação efetuar essa reprodução, não tendo apoio para tal nestas bibliotecas.

No âmbito da proposta de criação de um ginásio online, recorrendo a plataformas virtuais 3D e dispositivos de captura de movimentos, o desenvolvimento baseado em Rinions permite mais rapidamente concretizá-la e estudar os seus requisitos e desafios. Contudo, não estará fora de causa a necessidade de recorrer à FFAST e/ou à NUILIB, caso nesse processo de estudo se constate a necessidade de identificar gestos pré-definidos.

Antevê-se a necessidade de estudar as soluções de transmissão remota dos gestos realizados em frente ao Kinect, aspeto suportado pelo Rinions, mas onde se perspectivam desafios a nível da compensação de problemas de latência e largura de banda na rede. Por outro lado, é também necessário perceber até que ponto a execução direta de gestos é adequada a utilizadores com restrições motoras, ou seja, até que ponto a identificação de gestos pré-definidos não será um aspeto a ter em conta.

Por último, embora a opção por plataformas de mundos virtuais permita prototipar rapidamente um ginásio online, importa refletir e estudar os aspetos em causa antecipando uma eventual implementação noutras plataformas. Há tecnologias completamente distintas para renderização e multiutilização 3D, que terão de ser tidas em conta numa eventual evolução de protótipo para produto. Será desejável que a implementação das funcionalidades do ginásio online seja, tanto quanto possível, independente da tecnologia de base do Second Life e OpenSimulator, face à existência de outras plataformas (Jibe [20], CloudParty [21], Unity [22], etc.) com formas de desenvolvimento são bastante diferentes.

## Agradecimentos

Este trabalho é financiado pela Portugal Telecom Inovação, no âmbito do Plano de Inovação do Grupo PT, e por Fundos FEDER através do Programa Operacional Fatores de Competitividade – COMPETE e por Fundos Nacionais através da FCT – Fundação para a Ciência e a Tecnologia no âmbito do projeto «FCOMP - 01-0124-FEDER-022701».

## Referências

1. Atkinson, N.L. and R.S. Gold, *The promise and challenge of eHealth interventions*. American Journal of Health Behavior, 2002. **26**: p. 494–503.
2. Patrick, H. and A. Canevello, *Methodological overview of a self-determination theory-based computerized intervention to promote leisure-time physical activity*. Psychology of Sport and Exercise, 2011. **12**(1): p. 13-19.
3. Gerling, K.M., et al. *Full-Body Motion-Based Game Interaction for Older Adults*. in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems CHI 12*. 2012. Austin, Texas, USA: ACM Press.
4. Microsoft. *Kinect for Windows*. [cited 2013 03/06/2013]; Kinect for Windows gives computers eyes, ears, and the capacity to use them. With Kinect for Windows, thousands of businesses and developers are creating applications that put people first—allowing their customers to interact naturally with computers by simply gesturing and speaking.]. Available from: <http://www.microsoft.com/en-us/kinectforwindows/>.
5. Oshita, M. *Motion-Capture-Based Avatar Control Framework in Third-Person View Virtual Environments*. in *Proceedings of the 2006 ACM*



- SIGCHI international conference on Advances in computer entertainment technology ACE 06*. 2006. ACM Press.
6. Steinberg, G., *Natural User Interfaces*, in *CHI'12*, ACM, Editor 2012, ACM: Austin, Texas, USA.
  7. OpenNI. *3D sensors*. [cited 2013 12-06-2013]; OpenNI-compliant 3D depth sensors]. Available from: [http://www.openni.org/3d-sensors/#.UbhO\\_PnV9HQ](http://www.openni.org/3d-sensors/#.UbhO_PnV9HQ).
  8. Nintendo. *Wii Accessories*. [cited 2013 12-06-2013]; Available from: <http://www.nintendo.co.uk/Wii/Accessories/Accessories-Wii-Nintendo-UK-626430.html>.
  9. Entertainment, S.C. *PlayStation Move*. [cited 2013 12-06-2013]; Available from: <http://playstationmove.com/>.
  10. Sony, C.E., *Câmara USB EyeToy*.
  11. Kean, S., J. Hall, and P. Perry, *Meet the Kinect - an introduction to programming natural user interfaces2011*: Apress. 2011.
  12. Zeng, W. and Z. Zhang, *Multimedia at Work Microsoft Kinect Sensor and Its Effect*. Ieee Multimedia, 2012. **19**(2): p. 4-10.
  13. Ekelmann, J. and B. Butka. *Kinect Controlled Electro-Mechanical Skeleton*. in *2012 Proceedings of IEEE Southeastcon*. 2012. Orlando, FL.
  14. OpenNI. *About OpenNI*. 2013 [cited 2013 29-05-2013]; Available from: <http://www.openni.org/about/#.UacvxkDV-zo>.
  15. Iseki, F. *SLKinect2 - SecondLife Viewer Project*. 2012 [cited 2013 2013-0506]; Available from: <http://www.nsl.tuis.ac.jp/xoops/modules/xpwiki/?SLKinect2>.
  16. Iseki, F. *RINIONS is Real Time Input from NI/NUI and Output to the Network and Shared Memory System*. 2013 [cited 2013 2013-05-20]; Available from: <http://www.nsl.tuis.ac.jp/xoops/modules/xpwiki/?Rinions>.
  17. Suma, E., et al. *Flexible Action and Articulated Skeleton Toolkit (FAAST)*. [cited 2013 22/05/20013]; Available from: <http://projects.ict.usc.edu/mxr/faast/>.
  18. Suma, E., et al., *FAAST: The Flexible Action and Articulated Skeleton Toolkit*, in *Proceedings of IEEE Virtual Reality2011*. p. 247-248.
  19. McCaffery, J., A. Miller, and C. Allison, *Gonna Build Me a TARDIS: Virtual Worlds for Immersive Interactive Experience*. Proceedings of the 13th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, 2012.
  20. Inc., R. *Jibe*. 2012 [cited 2013 12-06-2013]; Available from: <http://reactiongrid.com/>.
  21. Cloud Party, I. *CloudParty - A Free Virtual World In Your Browser*. 2012 [cited 2013 10-06-2013]; Available from: <http://www.cloudparty.com/>.
  22. Technologies, U. *Unity - Game Engine*. [cited 2013 10-06-2013]; Available from: <http://unity3d.com/>.