Scale Model Games (SMG): An Introduction to a New Type of Game Play

Alain Simons, Karsten Pedersen, Hasan Abdulaziz, Davide Melacca
Bournemouth University
Poole House, Talbot Campus
UK BH12 5BB, Poole,
United Kingdom
asimons@bournemouth.ac.uk| kpedersen@bournemouth.ac.uk

ABSTRACT

In this paper, we introduce a new and innovative game play approach that utilizes collaborative multiplayer video game technology to control and manipulate physical miniature models remotely to offer an immersive gaming experience for playing out battlefield scenarios. Classical gameplay with miniature soldier and cannon models for specific war scenes is still in use today and many miniature toys are available for children and adults from toy manufacturers. During the nineties of the last century computer games became more popular and the development of old school toys such as these miniatures is in decline. Remote Controlled scale models (toys) do not give the same immersion as interactive realistic motion control and image display as found in video games. We propose Scale Model Games (SMG), a new type of game play that will provide a solution to this problem and make the gameplay a lot more immersive by integrating the game play of physical models with online multiplayer. This paper defines exactly what SMG stands for and presents a methodology that could be exploited for different gameplays such as miniature toys like tanks, cars, airplanes and boats for hobby or professional use. As a working example, the methodology for a tank battle game is described in this paper.

Keywords
Immersive scale model game play, Remote control, Real-Time Strategy Game, Imaginative Play, Real world tank battle.

1. INTRODUCTION - GAMEPLAY

Playing with scale models [1] is a legacy from the past experiencing strong competition from digital games in recent years. However, technology development in the area of data streaming and computer miniaturization issues new possibilities to make the gameplay more competitive with that of digital games. Playing with scale models gets a new dimension that demands a new definition of a gameplay. There is a wide variety of scale models, cars, tanks, airplanes, ships and dioramas are the most common. For our test case a tank battle is used. Tank battles are quite immersive games and the pure digital solutions lack a lot of the feeling of reality. If the proposed methodology will work for a tank battle it can be easily transferred to use in conjunction with other scale models such as cars or airplanes. To test the methodology, cooperation with a tank museum is undertaken to get as much possible input from different types of game players.
In the end our methodology can be a useful addition to their portfolio.

The lack of immersion in tank battle games is not only noticed by our research. Manufacturers of scale models, like Italeri try to make a link between popular computer game (in this case World of Tanks) and their own product, a model kit [2]. Vice versa software developers try to make the imaginative or also called pretend play more immersive by developing a VR games version of that pretend play [3]. Biggest issues with both solutions are that there is no direct contact between the virtual and real world. Both are still separated worlds, there is a gap that has to be covered by an imaginative play. For all of this, we propose the Scale Model Game (SMG) to bridge the gap. The goal of SMG is to offer a realistic immersive gameplay by utilizing collaborative multiplayer video game to control and play physical miniature models remotely. This new type of game play will make use of real world physics in video games by controlling and operating the scale model through video game control. Players of the game can collaborate via the video game play, whilst video streams of real world scale model action are transmitted into the video game. Let us take the example of a tank battle. Scale modelling

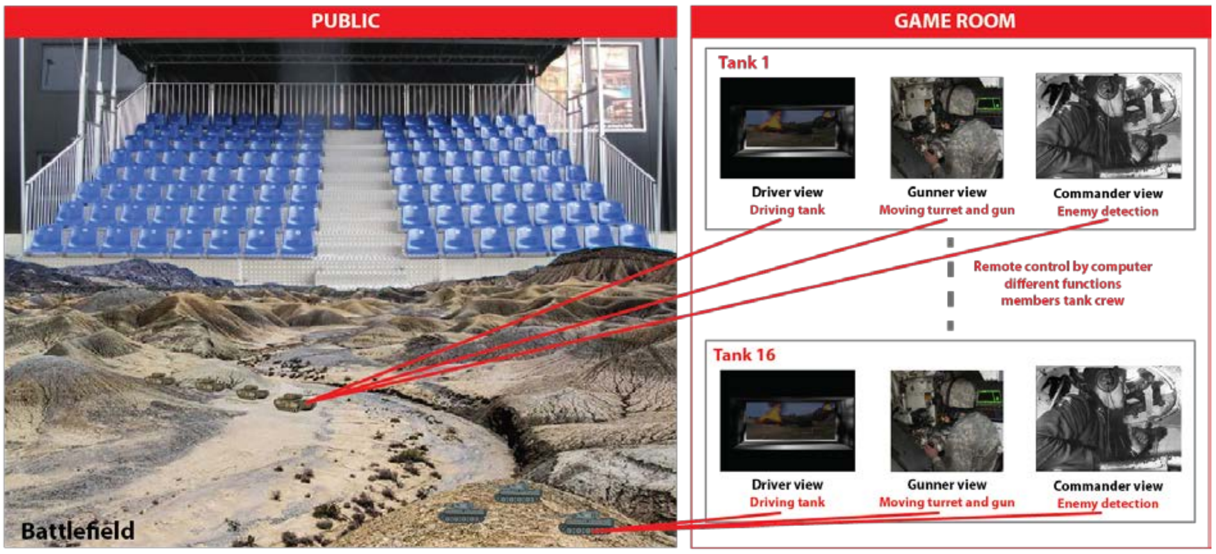

Figure 1: *Setup of SMG as a RTS game for a tank battle.*

companies brought different types of remote controlled tanks on the market. Especially some scale 1/16 scale models are very advanced and have the same functionality as a real world tank including shooting [4]. However, they are controlled with an external device whereby the player has no idea which physics the tank, and most important, the tank crew is undergoing. Another weakness comes forwards when a replay of ancient tank battles needs to be done. The scale of 1/16 is so huge that a full-scaled battlefield will take too much space to play. The goal of SMG is that those tank battles can be played in front of public so at least a descent overview of the battlefield for the public is needed. A frontline of 4 miles (6.4 km) scaled back to 1/16 is equal to a 400m. For that reason SMGs final aim is to use scale models 1/25 (256m frontline in this example) or 1/32 (200m frontline on scale). However, the hardware to make a tank move at those smaller scales is available it will be harder to get a functional shooting system in the smaller scaled tanks. For that reason the first prototype of SMG uses an existing 1/16 scale model. At the same time, the research team will develop a full functioning scale model at 1/25 and/or 1/32.

SMG can be defined as a one or multiplayer game. As a one-player game children can play it (with limited functionality due to safety reasons) at home to support an imaginative play. A later study into AI control of enemy tanks will be undertaken to provide opposition in this scenario. In case of a multiplayer it will become a Real-Time Strategy game (RTS). In the latter case a complete tank crew will take part in an organised tank battle which can be played remotely from a distance or at location. Organisations as tank museums could organise events were the public could not only watch but also participate in a tank battle. **Figure 1** illustrates the setup of a SMG tank for tank battle fields.

A tank crew will consist of three members. Those are a commander, a gunner and a driver. Crew members have to collaborate with each other and communicate

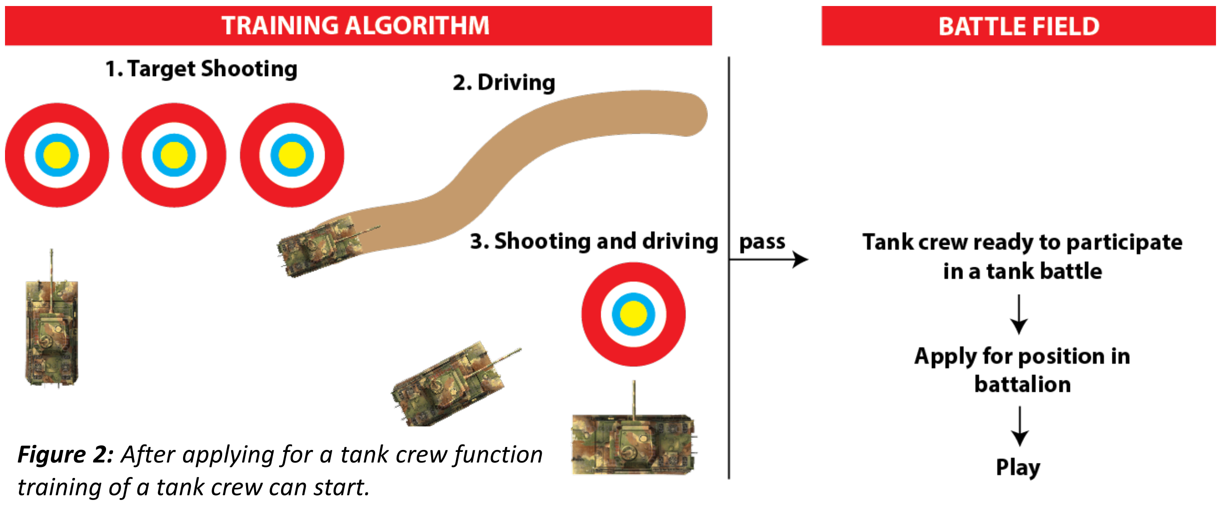

Figure 2: *After applying for a tank crew function training of a tank crew can start.*

with other tank crews to be successful in a tank battle. If a player wants to become member of a tank crew the player has to apply for a specific function. An automatic system will compose the tank crews before the player can join the game. There are two stages provided, training and battle. Before a tank crew can participate in a battle they need to receive the necessary training to handle the tank in a correct way. They also have to learn to play/work as a team. Training will cover, driving, shooting and a combination of both. Once a tank crew passes the training they can become part of a tank battalion and join a battle. **Figure 2** shows the workflow. The two armies on the battlefield have the same goal namely conquer enemy territory and assets with as ultimate objective the destruction or surrender of the enemy. Other gameplays such as scale model car racing games can also be controlled by the SMG methodology.

Summarize gameplay SMG methodology:

- Apply for crewmember (online process).
- Become crewmember and start training.
- Succeeded training crew can apply for place in tank battalion.
- Play tank battle.

This methodology is quite similar to those used in current tank battles. The novelty however is that is now applied in a real environment and not in a virtual one.

2. RECENT SOLUTIONS

Beyond the gameplay a technical setup is needed to control game objects. In this example the methodology could also be applied, with minor modifications, to other types of games such as car racing, airplanes and sea battles. The key is how the game objects are controlled. Despite 1/16 scale models are able to handle the same functions as real tanks using traditional remote control methods, there is only one player involved who is outside the tank. The player can see the physics the tank is undergoing but is not experiencing the physics themselves. There is no view from the inside as in a real situation. In addition a lot of those remote controlled tanks use a limited number of radio frequency which makes it hard to use as a multiplayer game. A next issue is the shooting. Some shooting systems use real bullets but IR is often used as a solution that limits the angle a tank can be hit. The air soft is a much more realistic solution but not safe for children because it is using fast moving projectiles. However, most importantly a tank cannot be destroyed or damaged. Tamiya for example uses a system of digital damage percentages [5] to adjust the functionality of the tank. No parts can break off the tank that is an unrealistic behaviour.

Figure 3: *A classical remote control.*

Summarized there are following remote controlled issues:

- Player is not undergoing any physics.
- Shooting is not destroying/damaging the tank.
- No realistic viewing from position of crew members inside the tank.
- No radio contact between different crew members.

On the other hand Virtual Reality games also have issues. To take a well-known example, Worlds of Tanks [6], most of the gameplay happens outside the tank. Using Virtual Reality devices like the Oculus Rift [7] will make the game more immersive. As the player is not handling with real objects, there is no connection with the reality, VR is still a simulation of the real world. Although manufacturers like Italeri brought out model kits referencing to the tanks used in a VR game, World of Tanks, to allow players to use their imagination to link the model tanks to the video game. Real damage or destroy an object is not possible and often digital damage percentages are used. As a final there is a heavy load on the player's computer system and game servers when playing remotely [8] due the calculation of physics.

Summarized the issues for Virtual Reality:

- Simulated Physics.
- No realistic score system.
- Massive load on player's computer system and game servers.

3. PROPOSED SOLUTION

The SMG solution is to eliminate the issues raised from VR game play and the remote controlled games. In this paper we present our development on the most

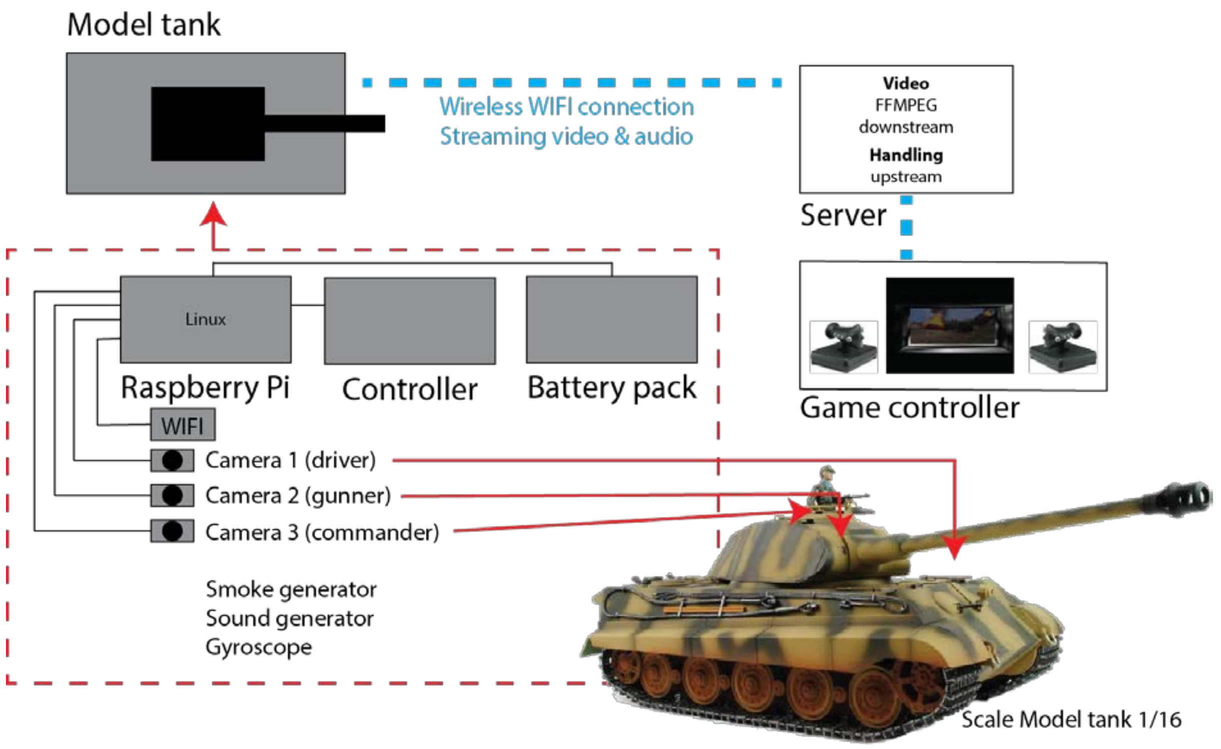

Figure 4: *Streaming data from cameras on game object to sever and game controller.*

important issue namely streaming video from the different positions of the tank members to a game server/controller. There are two major research challenges in the development

- The gameplay and developing video streaming methodology;
- Developing the needed hardware for shooting/ damage, tracking and controlling the tanks especially on scale 1/25.

We will test our solution for use on scaled tank battle games as a demonstration example of our proposed approach. For other gameplays some parts of the methodology need to be adjusted again.

One of the key features of SMG is the connection between the scale model and the game server. In this first prototype system, SMG will use a downstream for the video and a separate upstream for the handling as shown in **Figure 4**. As a first test case a 1/16 scaled tank is equipped with 3 cameras connected to a Raspberry Pi 3 (later tests will be done with a Raspberry Pi A+). Our goal is to get a fluent video streaming of HD resolution (1080 x 720px). The captured images will be displayed on a computer screen using an internet browser window. A technical complexity involved with streaming video footage from the model tank stems from the relatively low specifications of the hardware involved, including the limited bandwidth of the wireless connection between model tank and server. This will require careful testing and potentially an efficient load balancing system may be needed.

There are a variety of streaming formats available, ranging from proprietary formats from Apple and Adobe to more open formats and those suitable for the web (which is currently our primary deployment target). Below is a brief summary of some of the protocols evaluated.

VNC / RDP – Primarily remote desktop solutions but due to extensions to the protocol [10], the streaming of fast moving graphics (such as computer games) can be achieved. These traditional systems do not stream directly to the web and instead a shim layer needs to be used [11] [12] in order to translate between the Winsock / Berkeley Sockets TCP streams design and Web socket packets which require an additional payload.

HLS – Still very young in terms of development and support. Pioneered by Apple that also makes the main

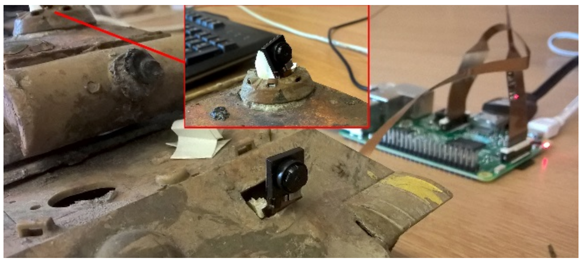

Figure 5: *Proximally positions of the cameras on the tank for a first test.*

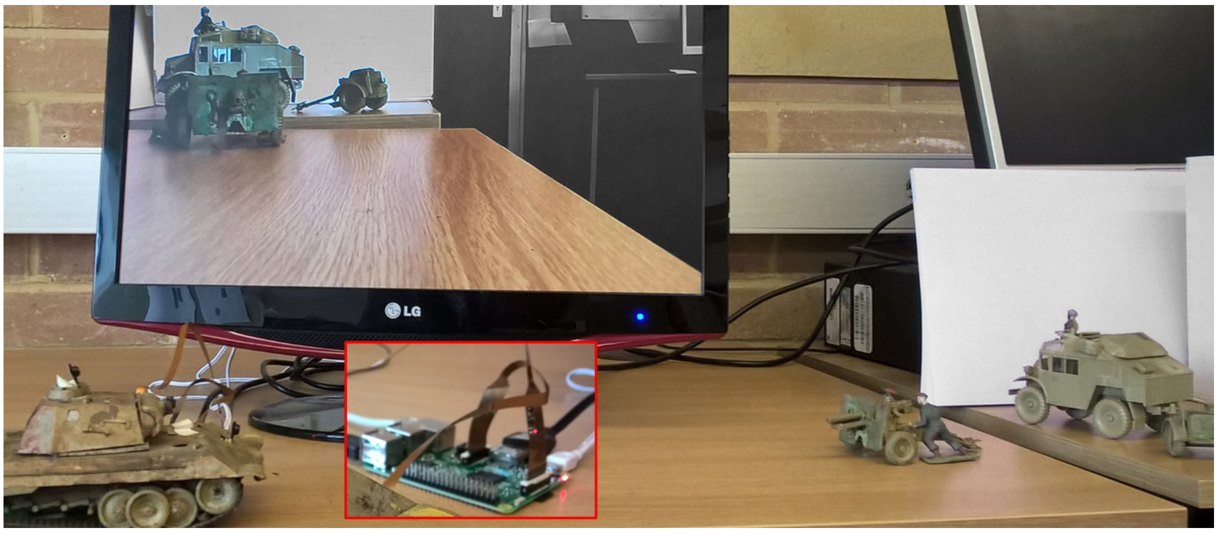

Figure 6: Video captured from the tank for stream to the game server.

platform focus Apple's Safari web browser. This potentially makes it unsuitable for this project.

FFMPEG / HTML5 Canvas - Using the open source FFMPEG video server, a generic stream can be sent in the open MPEG open format which can subsequently be captured and decoded using JavaScript on the client's web browser. This decoded data can then be displayed on the browsers HTML5 canvas. This solution will require relatively powerful clients but should mean that the hardware in the model tanks and the distribution server should be entirely adequate.

Getting the right technology to stream video from the tank to the player is one issue. Most important is also the exact position of the camera on the tank and the framing. The camera should deliver a view which is similar to that one the crew experience in a real tank to make the game most immersive.

4. RESULTS

Getting a Raspberry Pi and cameras connected was quite easy to do. Figure 5 and 6 show how a first practical setup was done. The quality of the used cameras was excellent. Auto focus is working very well and the refresh on movement is very fluent without any blur. As can be seen the images on the screen are very crisp and very useful for the SMG setup **(Figure 6**). Only the images are very wide but this can be adjusted once the data is streamed to the game server. Another issue is the number of connections for cameras on the Raspberry Pi. Although more than one connector is available only one can be used for streaming. The other cameras will have to be connected to an USB port. This needs further investigation on how it will affect the streaming. Alternatively, when there is enough space inside the tank, two extra Raspberry pi could be used to connect the other cameras directly.

The FFMPEG/HTML5 canvas was harder to setup. The proposed system was designed in such a way that the main grunt of processing is delegated to the main server and connected clients. This reduces the work done on the hardware inside the model tanks. The streaming process is detailed as follows.

The cameras connected to the Raspberry Pi use the standard V4L interface (allowing the use of standard USB camera hardware) which has its output converted to an MPEG format (via FFMPEG) and streamed to a central server via the HTTP protocol with a specific GET header donating which camera the stream is for (i.e. gunner, commander, driver).
The central server running tankd (Tank Daemon, written in C) accepts multiple streams and assigns based on this GET header an in-memory tank structure. This is where traditionally a Game Engine would then render out this tank and do the processing but this is not needed here.
A client web browser via Web sockets connects to tanks and using a very simple protocol requests which stream they want to view. This stream, upon

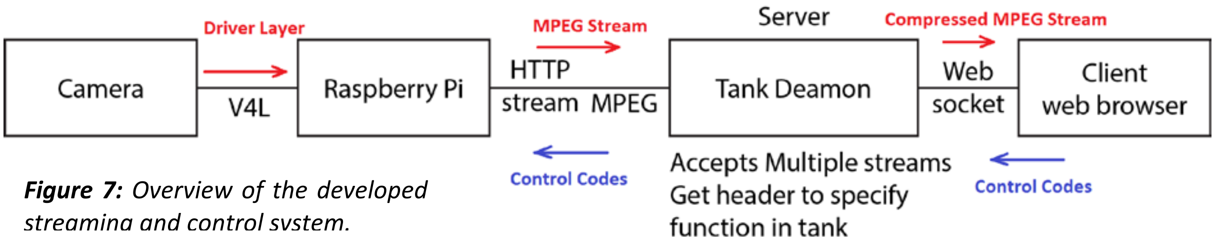

Figure 7: Overview of the developed streaming and control system.

receiving is assembled into pixels that is then displayed onto a HTML5 canvas. This currently is the most portable solution (working for browsers as old as Internet Explorer 9). This is a relatively expensive operation but even modest smartphones are very much capable of this task. Platform specific features could be taken advantage of to provide greater performance here such as ASM.js or even native software or app developed in C or C++ for maximum performance, avoiding the overhead of web protocols and JavaScript entirely **(Figure 7)**. This is for clients to get a visual output but also works in the reverse so that clients can pass input back to the tank. The control code (such as KEYDOWN_FORWARDS is passed through Web sockets back to tankd which sends it on to the correct hardware within the tank model through a raw socket interface. During post battle breaks, all the tanks can then be disabled centrally through this server.

As it stands, the bottlenecks in our solution is very much in the transmission of the data via Wi-Fi. Depending on the number of tanks, a large number of streams may require greater infrastructure to balance the load. However due to the modular nature of this system, this is entirely possible. Perhaps multiple tankd servers could be introduced, one for each tank battalion. Compression performed on the tankd server could also reduce the bandwidth required for multicasting to the multiple connected clients. However, compression performed on the tank hardware is unlikely to yield adequate performance.

5. CONCLUSIONS

The results show that a fluent streaming from the miniature cameras positioned on the right place on a game object (a tank in this test case) is possible with the proposed methodology. Also the framing of the captured video corresponds to what is experienced in the real world. The proposed methodology is definitively a way to get real world physics to the player without using a game engine for that specific purpose. This doesn’t mean a game engine can be ruled out for other functions but that will be further investigated in continuous research.

There is also a lot of fine-tuning to do. The miniature tank movements are more direct than those of real tank are. Therefore, the movements of the tanks have to be adjusted to get a better match with the real world movements. This is something that is easy to adjust in a game engine but could take more work to do on the scale model itself. With the new proposed methodology and the new gameplay workflow, game assets creators are making assets for a game so these can be used in a Virtual Environment or send to a 3D printer to be used in a SMG based methodology.

6. REFERENCES

[1] Wells, H.G. (1913) *little wars*, UK: Frank Palmer.

[2] Italeri (2016) *World of Tanks ROLL OUT*, Available at:*http://www.italeri.com/categoria.asp?idCategoria=52* (Accessed: 25th May 2016).

[3] Shaoxuan, W. and Sujit, D. (2010) 'Addressing Response Time and Video Quality in Remote Server Based Internet Mobile Gaming', *2010 IEEE Wireless Communication and Networking Conference,* (), pp. 1-6.

[4] Taigen Tanks (2016) *Taigen and Torro Tanks,* Available at: *http://www.taigentanks.com/* (Accessed: 25th May 2016).

[5] Tamiya (2016) *Battle System,* Available at:*http://www.tamiya.com/english/products/53447battle_system/battle_system.htm* (Accessed: 25th May 2016).

[6] World of Tanks (2016) *Version 9.15: Release on Wednesday 25 May,* Available at: http://worldoftanks.eu/ (Accessed: 25th May 2016).

[7] Russell Brandom (2014) 'The Norwegian Army is using the Oculus Rift to drive tanks',*The Verge*, 5th May.

[8] Ohashi, O., Ochiai, E. and Kato Y. (2014) 'A Remote Control Method for Mobile Robots using Game Engines', *28th International Conference on Advanced Information Networking and Applications Workshops,* (), pp. 79 - 84.

[9] Chou, K., Hsiu, M. and Wang C. (2015) 'Fighting Gulliver : An Experiment with Cross-Platform Players Fighting a Body-Controlled Giant', *CHI 2015 Seoul,* (), pp. 65 - 68. D. Aydin, “Transitions on Computational Collective Intelligence”, Springer Lecture Notes in Computer Science. 6220, pp. 39-55, May 2010.

[10] Osaka D. (2014) *Remote Desktop Services Blog,* Available at:*https://blogs.msdn.microsoft.com/rds/2014/06/06/understanding-and-evaluating-remotefx-vgpu-on-windows-server-2012-r2/* (Accessed: 25th May 2016).

[11] Martin J., Mannehed S., Astrand P. and Ross S. (2015) *noVNC: HTML5 VNC Client,*Available at: *https://github.com/kanaka/noVNC/blob/master/README.md* (Accessed: 25th May 2016).

[12] Martin J. (2016) *Websockify is a WebSocket to TCP proxy/bridge,* Available at:*https://github.com/kanakawebsockify* (Accessed: 25th May 2016.