



2006-01-01

# Using the Source Engine for Serious Games

Andrew Ritchie

*Dublin Institute of Technology*

Patrick Lindstrom

*Dublin Institute of Technology*

Bryan Duggan

*Dublin Institute of Technology*, [bryan.duggan@comp.dit.ie](mailto:bryan.duggan@comp.dit.ie)

Follow this and additional works at: <http://arrow.dit.ie/scschcomcon>

## Recommended Citation

Ritchie, A., Lindstrom, P. & Duggan, B. (2006) Using the source engine for serious games. *9th. International Conference on Computer Games: AI, Animation, Mobile, Educational and Serious Games*, Dublin Institute of Technology, Dublin, 22-24 November, 2006

This Conference Paper is brought to you for free and open access by the School of Computing at ARROW@DIT. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@DIT.

For more information, please contact [yvonne.desmond@dit.ie](mailto:yvonne.desmond@dit.ie), [arrow.admin@dit.ie](mailto:arrow.admin@dit.ie), [brian.widdis@dit.ie](mailto:brian.widdis@dit.ie).



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/)



# Using the Source Engine for Serious Games

Andrew Ritchie, Patrick Lindstrom, Bryan Duggan

School of Computing,  
Dublin Institute of Technology,  
Kevin St., Dublin 8.

Email: [andrew.richie@student.dit.ie](mailto:andrew.richie@student.dit.ie)  
[patrick.lindstrom@gmail.com](mailto:patrick.lindstrom@gmail.com), [bryan.duggan@comp.dit.ie](mailto:bryan.duggan@comp.dit.ie)  
<http://seriousgordon.blogspot.com/>

## Keywords

Serious Games, Source SDK, Half Life 2, Valve, Food Safety, Food Hygiene, Restaurant Kitchen.

## Abstract

In the paper we describe our work in using the Source engine from the game Half Life 2 to develop a *serious game* - Serious Gordon. Serious Gordon is used to teach principals of food safety and food hygiene in a restaurant kitchen environment. The target users of Serious Gordon are students of catering in the DIT Faculty of Tourism and Food. The paper describes the formation of the Serious Gordon team and the development of a story for the game. It continues with a description of the features from Half Life 2 that we retained and those that we removed for the game. We then describe the process we undertook to implement the game including a description of the tools we used. We conclude with a brief evaluation of the project and present future work.

## 1. Introduction

The Source Engine is an award winning 3D game engine developed by Valve Software [PC Gamer, 2004; PC Zone, 2004; IGN.com, 2006]. It is regarded as one of the most advanced games engines on the market supporting such features as realistically simulated physics using the Havok engine, DirectX 9.0 support including High Dynamic Range lighting, skeletal animation, sound systems and many other features [Valve, 2006]. It is the engine that the successful game Half Life 2 is built on. The source engine comes with the Source SDK which can be used to build game mods. The SDK is freely available to owners of the Half Life 2 game and provides access to the tools used by Valve themselves to create the Half Life 2 series of games, Day of Defeat Source, Counter Strike Source and upcoming games including Portal and Team Fortress 2. It is also used by many independent developers including Ritual the creators

of SIN: Episodes and Arkane the developers of the game Dark Messiah [Ritual, 2006; Sin, 2006; Arkane, 2006].

This paper describes our work developing the “Serious Gordon” project. The aim of Serious Gordon was to test the feasibility of using the Source engine to develop a *serious game*. Serious games are games that have non-entertainment purposes such as education or training [Johnson *et al.*, 2004]. Serious Gordon is a game that can be utilized as an aid in teaching food safety in a restaurant kitchen in a unique, safe and fun environment. The target users of Serious Gordon are students of catering in the DIT Faculty of Tourism and Food. Using a game engine, our aim was to simulate situations that could not be recreated safely and easily in the real world. In Serious Gordon the player takes the role of a new junior employee in a restaurant kitchen and is given a number of tasks such as changing into appropriate clothing, observing personal hygiene, first aid and storing deliveries of food in the correct locations.

## 2. Specification & requirements gathering

To develop the project, a team of domain experts was first assembled to address the technical and pedagogical aspects of the project. The team consisted of experts in three areas, from three faculties in DIT:

- Software development
- Learning technology
- Food safety

To implement the project, a number of candidate developers were interviewed and two undergraduate students with significant games development experience were recruited and retained for the summer of 2006. The students employed an iterative prototyping methodology to develop the system. In other words, the team proceeded on several fronts - requirements gathering, research and development of prototypes in parallel [Fullerton *et al.*, 2004].

The development of a “story” for the game posed an interesting challenge that was unfamiliar to the student software team. The Serious Gordon project required that a player interacting with the game would achieve a set of learning outcomes, whereas the students were used to thinking of gaming outcomes in terms of scores, end of level bosses and so on.

This game story was considered to be core to the development of a game that delivered the learning outcomes and so several weeks of meetings with food safety experts took place to develop realistic scenarios for the game. These meetings had input from experts in not only Food safety but also in learning technology. The *script* for the game went through many revisions, as the team developed an awareness of the capabilities and limitations of the Source engine. Iterative prototypes were developed to test games concepts.

The story of Serious Gordon places the player in the role of a new employee in a restaurant kitchen. The player starts the game similarly to a real restaurant employee by observing certain personal hygiene pre-requisites and dressing appropriately. The player then has to respond to an accident which requires them to administer first aid to a fellow employee. They then are required to stack food deliveries into appropriate locations in the kitchen - for example by placing frozen food in the freezer and chilled food in the chill room. Throughout the game the player receives feedback on their activities from the “head chef” (Figure 6). Once the story was finalised, the assets (maps, models, animations, sounds and so on) in Half Life 2 were compared to those required by the story and the required gameplay elements had been tested the decision made that the project was feasible and could proceed.

### 3. Removing Half Life 2 Gameplay Components

Half Life 2 is a first person shooter (FPS) [Sanchez & Dalmau, 2004]. The Serious Gordon project only shares only the first person perspective with Half Life 2. The first step in transitioning the Half Life 2 single player experience to something that would provide a foundation for Serious Gordon was to remove gameplay elements such as Non Player Characters (NPC) that would attack the player. Weaponry carried by both the player and NPCs were also removed and most of the player health based statistics such as the players HEV (Hazardous Environment Suit), damage taking and displays were removed. Figure 1 is a screenshot from Half Life 2 giving examples of these elements.

Scripting in the Source engine is achieved by implementing modules in C++. Valve provides the full source code for the games Half Life 2 and Counter Strike and consequently modules from these games were adapted for use in Serious Gordon. In some cases changes were simple and involved removing code that instantiated existing features, such as the HEV (Hazardous Environment Suit) suit statistics. Once these features were removed the game was ready for the Serious Gordon assets to be integrated and the code to script the gameplay could be added.



**Figure 1: The HUD (Heads Up Display), weapons and non player characters in Half Life 2**

### 4. Programming Serious Gordon Gameplay

Serious Gordon was developed using several software tools, including Microsoft Visual Studio .NET 2003 for coding, Valve’s Hammer Editor for mapping, modelling using Milkshape3D, choreography using Valves’ Face Poser, materials creation using Photoshop CS2, sound recording using Audacity and text editors for many other features.

Developing Serious Gordon was decomposed into the core tasks identified below:

- Developing context specific physics and character interaction menus
- Developing new clothing and inventory entities
- Developing new clothing and inventory Heads Up Display (HUD) panels
- Developing information panels
- Programming player tasks
- Programming events and triggers to advance the script

The source code provides access to the Valve Graphical User Interface (VGUI) which is used by Valve to create graphical panels both inside the Source Engine and outside it in Steam (Valve’s content delivery system). Figure 2 is an example from the Serious Gordon source code that draws the HUD inventory items using Source SDK API calls.

In Half Life 2, pressing the “use key” triggers an interaction between the player and the object in front of the player in the game. Using this system, the interaction menu seen in Figure 3 was created. This menu appears when the user interacts with a wash hand basin and it allows a player to simulate washing their hands in the game.

This was implemented by adapting the class `CTextWindow` from the Source SDK. This class has a member function `OnCommand` that receives a message indicating the user interface control that generated the event. We have implemented a flexible framework that facilitates the creation of interface elements such as buttons and text messages from text files.

```

void CHudInventoryStatus::Paint()
{
    C_BaseHLPlayer *pPlayer =
    (C_BaseHLPlayer*)C_BasePlayer::GetLocalP
    layer();
    if (!pPlayer)
        return;

    // draw the suit power bar
    surface()->DrawSetColor(m_SquadIconColor );
    surface()->DrawSetTextFont( m_hIconFont
    );
    surface()-
    >DrawSetTextPos(m_flIconInsetX,
    m_flIconInsetY);
    surface()->DrawUnicodeChar('D');

    surface()-
    >DrawSetTextFont(m_hTextFont);
    surface()-
    >DrawSetColor(m_SquadTextColor);

    surface()->DrawSetTextPos(text_xpos,
    text_ypos);
    surface()->DrawPrintText(m_wcItemName,
    wcslen(m_wcItemName));
}

```

**Figure 2: An example from the Serious Gordon Source code that uses Source SDK API's to draw the inventory**



**Figure 3: Interaction menu example - A player interacting with a wash hand basin.**

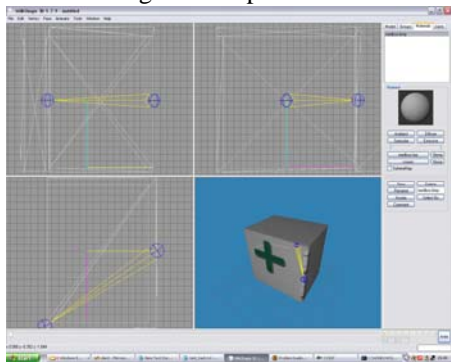
Flexibility is again demonstrated in the approach taken in displaying the Heads Up Display (HUD). The HUD itself is derived from elements of the VGUI with Serious Gordon specific HUD elements derived from the HUD classes. Figure 4 presents the HUD from Half Life 2 and compares it with the HUD we developed for Serious Gordon. This approach provides a common look and functionality amongst HUD elements. Although the Serious Gordon elements provide different information, they do not come as a change to a player familiar with the Half Life 2 game.



**Figure 4: The Half Life 2 HUD (on the left) and the Serious Gordon HUD (on the right)**

An *entity* is an object that has some kind of interaction with and exists inside the game world. The Serious Gordon entities are again all based of the existing Half Life 2 entities. HUD elements on the other hand have no interaction with the world itself and are merely indicators of specific attributes of an entity.

Model assets that did not exist in the Half Life 2 collection, such as the chef character, and kitchen implements were modeled using Milkshape3D as illustrated in Figure 5.



**Figure 5: Editing Serious Gordon models in Milk-Shape3D**

Meshes were created using vertex placement tools and connected to vertices with faces using the built in Face tool. Once the meshes were completed the sections were laid out onto a UV map using the Milkshape3D's built in mapping tool. A UV map maps texels on a bitmap to vertices on a 3D model. From Milkshape3D the models were exported to the Source proprietary file format, which Milkshape3D natively supports.

The existing textures and models for Half Life 2 fit the Half Life 2 environment - a run down Eastern European city. These were brightened and cleaned to better fit into the Serious Gordon maps, by using Adobe Photoshop with a Valve Texture File plug-in. Custom models for Serious Gordon had skins created in Photoshop and then exported to the proprietary format. Some of the characters from Half Life 2 (for example the Father Gregory character from Ravenholm) were re-skinned and reused in Serious Gordon.



**Figure 6: Father Gregory from Half Life 2 (left) and reskinned to make Chef from Serious Gordon (right)**

Using the prewritten storyboard, dialogue was scripted for the character voices. These were recorded using the open source audio editing tool Audacity. We engaged the services of a professional voice actor from the Gaeity School of Acting to record the dialogue [Gaeity, 2006]. The actor could reproduce several speaking accents on demand and so provided the voices for several Serious Gordon game characters. The Source engine uses standard WAV files for vocals.

Source's proprietary tool, "Face Poser", was used to generate character animation and motion from voice audio files.

Valve's proprietary level editor, the Valve Hammer Editor (VHE) was used to create the kitchen and restaurant environments used in Serious Gordon. In order to accurately model the environment, the team carried out some on-site research in the college restaurant kitchen. The developers also received significant input into the environment design from the catering experts from the Faculty of Tourism and Food on the team. Our aim was to accurately simulate a real kitchen restaurant environment - a requirement unique to this type of serious game. This is obviously in contrast to a fantasy FPS like Half Life 2 which adapts elements of the real world to the gameplay.

Concessions were made however to, for example the proportions of game entities so as to make the game playable.

## 5. Conclusions and Future Work

The game was completed in September 2006 and was made available for download by students on the college VLE (WebCT). We carried out limited internal testing of Serious Gordon and we feel that our original project aims have been exceeded. Based on our internal testing, everyone who played the game indicated that their knowledge of food safety issues when working in a restaurant kitchen was better after playing the game than before. We therefore conclude that the Source Engine can be successfully used to create a serious game. During the development of the project, we tracked releases in the Source SDK. We did however encounter problems with tools from different releases which meant that facial animation was not incorporated into the final release of the game. The software team made all of the models for Serious Gordon and so the game retains much of the look and feel of Half Life 2.

Future work will focus on the creation of art assets to increase the realism of the game. We also hope to expand the number of learning scenarios developed and do a more structured validation of Serious Gordon by doing a usability test with first year students from the faculty of Tourism and Food.

## 6. About the Authors

Andrew Richie is an undergraduate student of the DIT, studying Computer Science with Games Programming. He has been interested in computer gaming from a young age and has a deep knowledge of the Source SDK from his experience as a lead developer on a number of Half Life 2 modding projects including Flanders Fields and Resitance and Liberation.

Patrick Lindstrom graduated with a first class honours in computer science from the Dublin Institute of Technology in 2006 and is currently pursuing a career in software architecture at Datalex Ltd., Ireland

Bryan Duggan is a lecturer in the school of computing at the DIT in Kevin St. He hold a first class honours degree in computer science and software engineering from the University of Dublin (studied at the DIT) and a Masters Degree from the DIT. He is presently working on a PhD with a working title of "Modeling Creativity in Traditional Irish Flute Playing". He lectures on games programming and music technology in the DIT School of Computing.

## References

- [Arkane, 2006] Arkane Studios Home Page <http://www.arkane-studios.com/en/index.php> Accessed September 2006.
- [Fullerton *et al.*, 2004] Fullerton, T., Swain, C., Hoffman, S.: Game Design Workshop: Designing, Prototyping, and Playtesting Games, CMP Books, 2004
- [Gaeity, 2006] Gaeity School of Acting Home Page, <http://www.gaietyschool.com/>, Accessed September 2006
- [IGN.com,2006] PC Game of the Year 2004.
- [PC Zone, 2004] PC Zone UK (Editor and Readers Choice), December 2004
- [PCGamer, 2004] PC Gamer (US), December 2004
- [Ritual, 2006] Ritual Home Page <http://www.ritual.com/>, Accessed September 2006
- [Sanchez & Dalmau, 2004] Sanchez, D., Dalmau, C.: Core Techniques and Algorithms in Game Programming, New Riders, 2004
- [Sin, 2006] Sin Episodes Home Page <http://www.sinepisodes.com/> Accessed September 2006.
- [Valve, 2006] Source Information Sheet [http://valvesoftware.com/SOURCE\\_InfoSheet.pdf](http://valvesoftware.com/SOURCE_InfoSheet.pdf), Accessed September 2006.