

Balázs Dénes Kovács

Unity Editor Tools Development

Avatar Content Editor

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

01 March 2016

|  |  |  |
| --- | --- | --- |
| Author(s)  Title  Number of Pages  Date | Balázs Dénes Kovács  Unity Editor Tools Development  xx pages + x appendices  5 May 2016 | |
| Degree | Bachelor of Engineering | |
| Degree Programme | Information Technology | |
| Specialisation option | Software Engineering | |
| Instructor(s) | Juho Iso-Markku, Title (for example: Project Manager)  Antti Laiho, Title (for example: Principal Lecturer) | |
|  | | |
| Keywords | |  |

**Contents**

[1 Introduction 1](#_Toc454980548)

[2 Theoretical Background 1](#_Toc454980549)

[2.1 Unity Engine 1](#_Toc454980550)

[2.2 Unity Editor 2](#_Toc454980551)

[2.2.1 The Editor in General 2](#_Toc454980552)

[2.2.2 Scene View 2](#_Toc454980553)

[2.2.3 Game View 2](#_Toc454980554)

[2.2.4 Hierarchy 2](#_Toc454980555)

[2.2.5 Project Panel 3](#_Toc454980556)

[2.2.6 Inspector 3](#_Toc454980557)

[2.2.7 Toolbar 3](#_Toc454980558)

[2.3 Unity Editor Scripting 3](#_Toc454980559)

[3 The Avatar Content Editor 5](#_Toc454980560)

[3.1 Functionality 5](#_Toc454980561)

[3.2 Items and Products 5](#_Toc454980562)

[3.3 Components 5](#_Toc454980563)

[3.4 Settings 5](#_Toc454980564)

[3.5 Tabs 5](#_Toc454980565)

[3.6 Browsers and Pickers 5](#_Toc454980566)

[3.7 Atrributes 5](#_Toc454980567)

[3.8 Validation 5](#_Toc454980568)

[4 Results 5](#_Toc454980569)

[5 Conclusion 5](#_Toc454980570)

[References 5](#_Toc454980571)

Appendices

Appendix 1. Title of the Appendix

Appendix 2. Title of the Appendix

# Introduction

Today more and more developers use Unity as the game engine for their games. There are over 4.5 million registered developers of Unity as of now [1]. As the number of Unity developers increase so does the amount of complex game projects. These projects require more than only a team of skilled programmers and artists to work together using only the platform Unity provides. However, in order to develop their games efficiently they need to rely on tools made for Unity for special purposes. These tools can be generic to an extent and thus coded by an external personnel or they can also be highly specialized to fit specific needs of the game project.

In this paper I am going to describe the development of The Avatar Content Creator (ACE) that was proposed and developed in Sulake Corporation OY for Project X. The ACE is a highly specialized tool to create content for the avatars of the game. The purpose of this project was to re-implement an already existing tool called the Cloth Editor to facilitate more and better functionalities. Furthermore, the new tool had to support a faster and more advanced art and balancing process. In addition to this, the communication with the data servers of the game had to be overhauled to enable direct editing of the database of items. Finally, the project aimed to increase productivity and improve the stability of the avatar content creation process.

# Theoretical Background

## Unity Engine

Unity is a versatile game engine which allows developers to create games across numerous platforms. Unity was developed by Unity Technologies and is mainly used for creating games, however several developers use it for various other purposes. The game engine currently supports scripts in three programming languages such as C#, JavaScript and Boo. [1, 5.]

The ACE was written entirely in C# using Unity libraries. Although the server side code for ACE is written in Java and was written by another team member.

## Unity Editor

### The Editor in General

The GUI of the Unity Editor provides a simple and intuitive way for creating and modifying elements in games. Developers have many helpful features built-in for example an extensive asset browser, animation visualization, detailed inspector, etc.

### Scene View

Furthermore, the Editor window features a scene and game window. In the scene the users can manipulate game objects and see certain debugging features drawn in this view. With the combination of the game elements and debugging entities this window is one of the most important and useful tool in Unity. Inside the game view the developers can see their games in action as well as gain the ability to test their game with all the visual effects, sounds, player controls, etc. that the scene view omits.

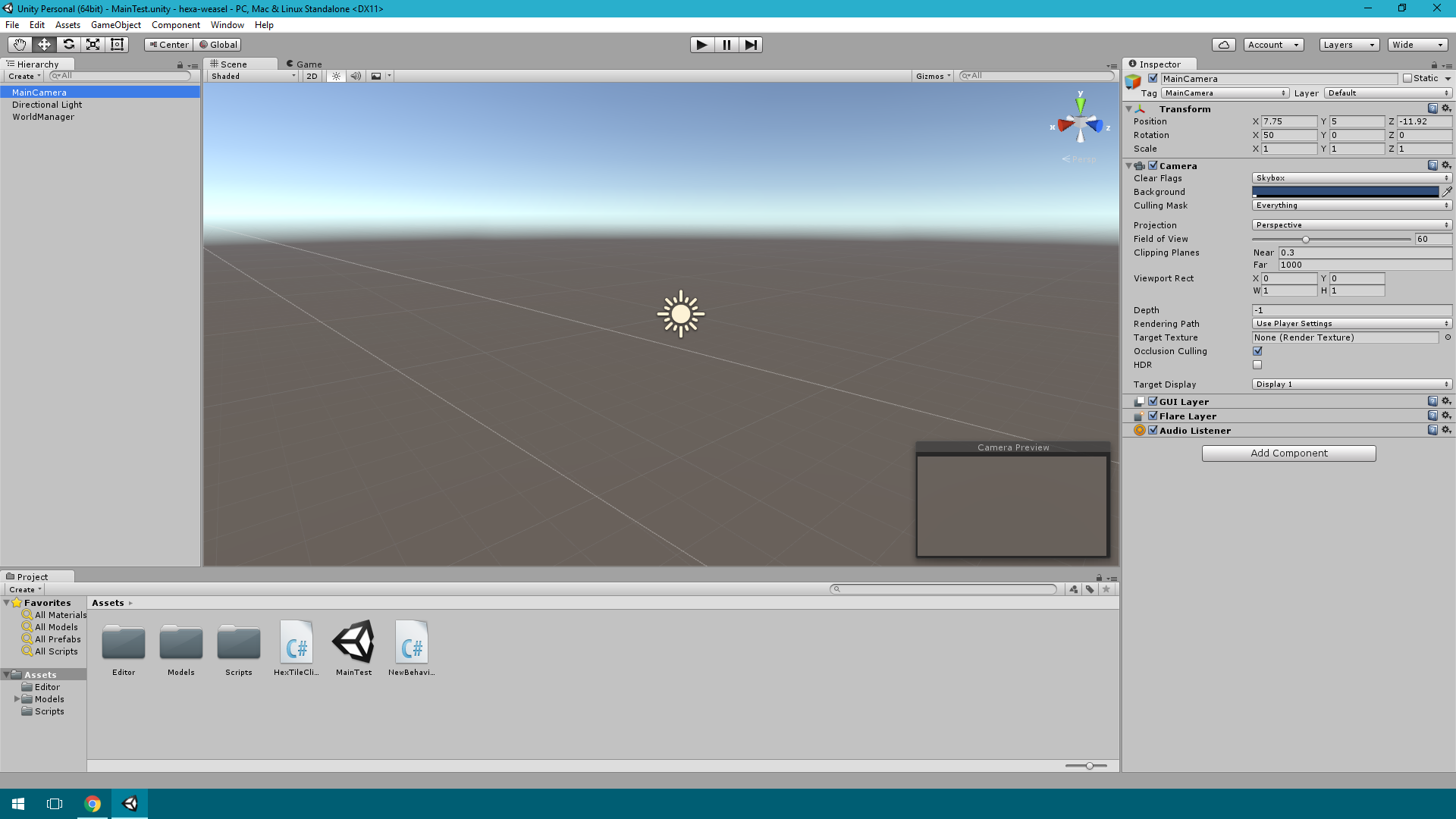


Figure 1. Unity Editor Window.

### Game View

To see the game running in the game panel, the developer has to put the editor into play mode. This can be done by pressing the play button on the toolbar as illustrated on figure 1. [2, 26-27.]

### Hierarchy

The Hierarchy window contains all the elements present in the current scene of the project. Models, lights, cameras and prefabs can be displayed here. By clicking any of the items the user can select and highlight items in the scene as well. [2, 28.]

### Project Panel

The project panel is essentially a file browser for the project. Using this window, the files of the game can be arranged into directories without having to switch to the external file browser of the operating system. [2, 29.]

### Inspector

The inspector displays the properties of an object which is currently highlighted. The content of the inspector window depends on what type of object the developer is inspecting. Inspecting a camera object for example show camera specific lighting and viewport options as shown on figure 1. [2, 30.]

### Toolbar

The toolbar is an essential part of the editor. From this section we can access all the object manipulations tools which are used in the scene view. These tools enable translation, rotation and scaling of game objects. However, there are certain objects which lack size and thus rotation and scaling has no effect on them. These empty objects or points in space are usually used for grouping certain items. In addition to these buttons the toolbar has three buttons for controlling the game view. The play, pause and step-through buttons. With the step-through one can pause the game and advance it frame-by-frame to inspect the game further. Finally, the toolbar has two dropdowns, one for organising the layout of the editor by certain pre-sets and a layers dropdown for selecting and organizing groups of game objects. [3, 11.]

## Unity Editor Scripting

Editor scripting in Unity is a useful practice in large projects. It allows the developers to build tools and automation for the game creation process. These custom tools made by editor scripting can range from custom inspectors to any form of internal utilities which generally speed up the production. [4, 2.]

Using editor scriptsone can create custom inspectors. Unmodified inspector views show all the basic properties of the object selected in the hierarchy. Once the game object gets too many properties and becomes a burden to handle, making a custom inspector can be a good idea. With the help of the editor scripts developers can organize the fields of the objects into logical groups and even implement methods to handle the data from the inspector in a specific way. [4, 49.] For example, displaying the level of a character object, which only has information about the experience points it gathered.

Additionally, the Unity editor features classes such as GUIStyle and GUISkin to allow further modification of the default look of the editor window elements. With the help of the GUIStyle class the font, the color, even the size of the elements can be changed among many other options. [4, 147-150.] Using the GUISkin class the developers may define new skin packages for the default editor window components [4, 156]. Applying a GUISkin rather than using GUIStyles with each element is generally a better approach.

Furthermore, saving data into scriptable objects persistently is also possible with editor scripts. Scriptable objects always exist in the project without being attached to a game object. This is why they are widely used for saving changes during play mode. Moreover, saving information into scriptable objects have some benefits over using XML or JSON files for the same purpose. Finally, allowing persistent changes to certain elements of the game gives liberty to game designers when they have to adjust certain values to make the game a better experience for players.

Unity is an extensive set of tool for rapid game development across multiple platforms. The Editor of Unity allows great creative and engineering freedom to developers and designers alike. The many tools of the Unity Editor allow developers to visualize their work in the process and manipulate scenes with ease. With the use of editor scripts developers can modify the environment of Unity to benefit the developer team. Moreover, the custom editors are customizable not only with functionality, but with looks as well. Finally scriptable objects provide a persistent way of storing important values of the game changed during testing and developing.

# The Avatar Content Editor

## Functionality

## Items and Products

## Components

## Settings

## Tabs

## Browsers and Pickers

## Atrributes

## Validation

# Results

# Conclusion

# References

x

|  |  |
| --- | --- |
| 1 | Navarro A, Pradilla JV, Rio O. Open Source 3D Game Engines for Serious Games Modeling. Alexandru C, editor. Rijeka, Croatia: InTech; 2012. |
| 2 | Henson C, Henson R. Unity 4.x Game Development by Example Beginner's Guide. Birmingham, UK: Packt Publishing Ltd; 2013. |
| 3 | Smith M, Queiroz C. Unity 4.x Cookbook. Birmingham, UK: Packt Publishing Ltd.; 2013. |
| 4 | Tadres A. Extending Unity with Editor Scripting. Birmingham, UK: Packt Publishing Ltd.; 2015. |

x

**Title of the Appendix**

Content of the appendix is placed here.

**Title of the Appendix**

Content of the appendix is placed here.