



Community Experience Distilled

Getting Started with UDK

Build a complete tower defense game from scratch using the Unreal Development Kit

John P. Doran

[PACKT]
PUBLISHING

Table of Contents

[Getting Started with UDK](#)

[Credits](#)

[About the Author](#)

[About the Reviewer](#)

[www.PacktPub.com](#)

[Support files, eBooks, discount offers and more](#)

[Why Subscribe?](#)

[Free Access for Packt account holders](#)

[Preface](#)

[What this book covers](#)

[What you need for this book](#)

[Who this book is for](#)

[Conventions](#)

[Reader feedback](#)

[Customer support](#)

[Downloading the example code](#)

[Downloading the color images of this book](#)

[Errata](#)

[Piracy](#)

[Questions](#)

[1. Augmenting the UDK](#)

[What we will achieve](#)

[Before we begin](#)

[Block out simple-level geometry](#)

[Prepare for lift-off](#)

[Engaging thrusters](#)

[Objective complete](#)

[Supplemental information](#)

[Defining Kismet](#)

[Using a third-person perspective](#)

[Engage thrusters](#)

[Objective complete](#)

[Classified information](#)

[Kismet primer](#)

[Parts of a sequence object](#)

[Benefits and drawbacks of using Kismet](#)

[Benefits of using Kismet](#)

[Drawbacks of using Kismet](#)

[Summary](#)

[2. Tower Defense](#)

[Spawning enemies](#)

[Enemies damaging the base](#)

[Creating/Spawning a single tower](#)

[Multiple towers made easy – prefabs](#)

[Engage thrusters](#)

[Summary](#)

[3. Detailing Environments](#)

[Applying materials](#)

[Placing staircases](#)

[Adding in-level boundaries](#)

[Spawning weapons](#)

[Objective complete](#)

[Summary](#)

[4. Finishing Touches](#)

[Obtaining Flash](#)

[Setting up Flash](#)

[Creating our main menu](#)

[Creating our HUD](#)

[Creating the main menu into UDK](#)

[Cooking and packaging our game](#)

[Taking the project to the next level](#)

[Summary](#)

[Index](#)

Getting Started with UDK

Getting Started with UDK

Copyright © 2013 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: July 2013

Production Reference: 1040713

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham B3 2PB, UK.

ISBN 978-1-84969-981-5

www.packtpub.com

Cover Image by Abhishek Pandey (<abhishek.pandey1210@gmail.com>)

Credits

Author

John P. Doran

Reviewer

Dan Weiss

Acquisition Editor

Erol Staveley

Commissioning Editor

Yogesh Dalvi

Technical Editors

Mausam Kothari

Vaibhav Pawar

Copy Editors

Insiya Morbiwala

Laxmi Subramanian

Project Coordinator

Suraj Bist

Proofreader

Elinor Perry-Smith

Indexer

Priya Subramani

Production Coordinator

Shantanu Zagade

Cover Work

About the Author

John P. Doran is a technical game designer who has been creating games for over 10 years. He has worked on an assortment of games in teams from just himself to over 70 in student, mod, indie, and professional projects.

He previously worked at LucasArts on *Star Wars 1313* as a game design intern. He later graduated from DigiPen Institute of Technology in Redmond, WA, with a Bachelor of Science in Game Design.

John is currently a software engineer at DigiPen's Singapore campus and is tutoring and assisting students with difficulties in computer science concepts, programming, linear algebra, game design, and advanced usage of UDK, Flash, and Unity in a development environment.

This is his third book after *UDK iOS Game Development Beginner's Guide* and *Mastering UDK Game Development*, both of which are also available from *Packt Publishing*.

He can be found online at <http://johnpdoran.com> and can be contacted at [<john@johnpdoran.com>](mailto:john@johnpdoran.com).

I want to thank my brother Chris Doran and my fiancée Hannah Mai, for being supportive and patient with me as I spent my free time and weekends away from them as I had to spend time writing the book.

On that same note, I also want to thank Samir Abou Samra and Elie Hosry for their support and encouragement while working on this book, as well as the rest of the DigiPen Singapore staff.

I want to thank Erol Staveley who approached me for writing again as well as everyone else at Packt who were so helpful, as always!

Last but not the least, I'd love to thank my family as well as my parents, Joseph and Sandra Doran, who took me seriously when I told them I wanted to make games for a living.

About the Reviewer

Dan Weiss is currently a programmer working at Psyonix Studios in San Diego, CA. He is a 2010 graduate of DigiPen Institute of Technology, having worked on titles such as *Attack of the 50ft Robot!* during his time there. He has been working in the Unreal engine since 2004, independently producing the mod *Unreal Demolition* for Unreal Tournament 2004 and Unreal Tournament 3. At Psyonix, he has been involved with Unreal engine work on mobile devices, having released ARC Squadron for iOS devices.

Support files, eBooks, discount offers and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [<service@packtpub.com>](mailto:service@packtpub.com) for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Preface

The UDK, which is a free version of the popular and award-winning Unreal 3 engine, is an amazing and powerful tool to use for projects of any kind. You can use it to create high-quality games and make your dream games a reality. UDK can be a little intimidating based on the level of games it has contributed to the ever growing and exciting world of gaming. Overcome all your apprehensions with this step-by-step guide and build a complete project within the Unreal Development Kit with unique gameplay, custom menus, and a triple A-rated finish.

This book will help you create a custom Tower Defense game within UDK and a game you can show your friends, even if you have absolutely no prior knowledge of UDK game development.

In next to no time, you will learn how to create any kind of environment within the UDK. With your basic environment created, you will make use of simple visual scripting to create a complete Tower Defense game with enemies attacking in waves. We then finish off the game with custom menus and a Heads Up Display. The final step is to release your game into the world and give others the excitement of playing it.

What this book covers

[Chapter 1](#), *Augmenting the UDK*, introduces us to the UDK and helps us create our gameplay environment out of nothing but making use of CSG and briefly touching on Kismet to create third-person gameplay.

[Chapter 2](#), *Tower Defense*, teaches us how to implant the basic gameplay for our project making use of Kismet to spawn enemies, and how to create spawnable towers in the game world making use of prefabs.

[Chapter 3](#), *Detailing Environments*, discusses the role of an environment artist doing a texture pass on the environment. After that, we will place meshes to make our level pop with added details. Finally, we will add a few more things to make the experience as nice looking as possible.

[Chapter 4](#), *Finishing Touches*, helps us create the basis of a Heads Up Display making use of Scaleform importing a project from Flash and touch on how to communicate between UDK and Flash. The HUD will adjust based on variables we've created in Kismet. We will also create a quick main menu level, which we can use to publish our final game! Then we will actually publish our game making use of the Unreal Frontend and share it with the world!

What you need for this book

Before we start, let's make sure that we have the latest version of the UDK (February 2013 as of this writing), which can be downloaded at <http://www.unrealengine.com/udk/downloads/>. When installing the program, make sure that the **UT Sample Game** option is checked.

Apart from that, all of the assets used in this project should already be included within the base UDK install.

This project and all projects assume that the user has used the UDK to some extent in the past, and is familiar with the concepts of navigating around the game environment.

For those wanting to know more about basic movement, please see Epic's UDN page that lists Hotkeys that may be useful at <http://udn.epicgames.com/Three/EditorButtons.html>.

That being said, I do my best to be as descriptive as possible in the steps needed to create the game and explain why I'm doing each step.

Who this book is for

If you have ever had the urge to know more about how all those amazing games you played for countless hours are created, then this book is definitely for you! This step-by-step tutorial will teach you how to create a complete game within the UDK.

Even if you have no prior experience of the UDK, you can still start building the games you want today!

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "In the **Properties** window, type `behindview 1` as the value for **[0]** in **Commands**"

A block of code is set as follows:

```
//Import events so that we can have something happen every frame
import flash.events.*;
//Add an event to happen every frame
stage.addEventListener(Event.ENTER_FRAME, Update);
function Update(evt:Event):void
{
    // Every frame we want to set the variables to
    // what we set them in Kismet
    cash.text = "$" + String(playerCash);
    // The wave number that we are at
    hudWaveNumber.text = String(waveNumber);
    // The times an enemy can hit our tower before we loose
    hudLives.text = String(lives);
    // If we have info to tell the player (Game Over) we can give
    // it here
    hudInfoText.text = infoText;
    // Let the player know the progress that he is making
    waveProgress.text = killedEnemies + "/" + totalEnemies;
    // The bar will fill as the player kills enemies but we don't
    // want to divide by zero so we just use a small number for //the
scale
    if(totalEnemies> 0)
        waveBar.scaleX = killedEnemies/totalEnemies;
    else
        waveBar.scaleX = 0.01;
}
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "If your viewport is zoomed in like the previous screenshot, click on the **restore viewports** button on the top right of each of the viewport".

Note

Warnings or important notes appear in a box like this.

Tip

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to <feedback@packtpub.com>, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

Downloading the color images of this book

We also provide you a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from

http://www.packtpub.com/sites/default/files/downloads/Getting_Started_with_UDK.pdf

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [<copyright@packtpub.com>](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at <questions@packtpub.com> if you are having a problem with any aspect of the book, and we will do our best to address it.

Chapter 1. Augmenting the UDK

The **Unreal Development Kit (UDK)**, the free version of Epic Games' Unreal Engine 3, is truly a sight to behold.

There are plenty of tutorials available on creating specific things in games, but in my experience there have been very little in terms of explaining how games are created in the actual game industry. In this book, I plan to expose those processes while creating a game from scratch using the Unreal Development Kit, including things that most tutorials leave out, such as creating menus, custom GUI, and publishing your game.

The game that we will be creating will be a basic third-person shooter / Tower Defense hybrid game using the default UDK assets. Tower Defense games have been quite popular on game sites, and we will be creating gameplay similar to that found in the popular titles Monday Night Combat and Dungeon Defenders, both of which were created using Unreal.

In this chapter, we will be creating the first playable version of our game. It will be split into five tasks. It will be a simple step-by-step process from beginning to end. Here is the outline of our tasks:

- Block out simple-level geometry
- Enable third-player mode

We will first approach the project using nothing but the UDK Editor and Kismet.

What we will achieve

Once we finish this chapter we will have the base layout of our gameplay environment done. We will also obtain a foundational knowledge in how to build areas out with CSG Brushes and exposure to Kismet before going more in depth in future chapters.

Before we begin

Before we start, let's make sure that we have the latest version of the UDK (February 2013 as of this writing), which can be downloaded at <http://www.unrealengine.com/udk/downloads/>. When installing the program, make sure that the **UT Sample Game** option is checked.

Aside from that, all of the assets used in this project should already be included within the base UDK install.

For those wanting to know more about basic movement, please see Epic's UDN page at <http://udn.epicgames.com/Three/EditorButtons.html> that lists hotkeys that may be useful.

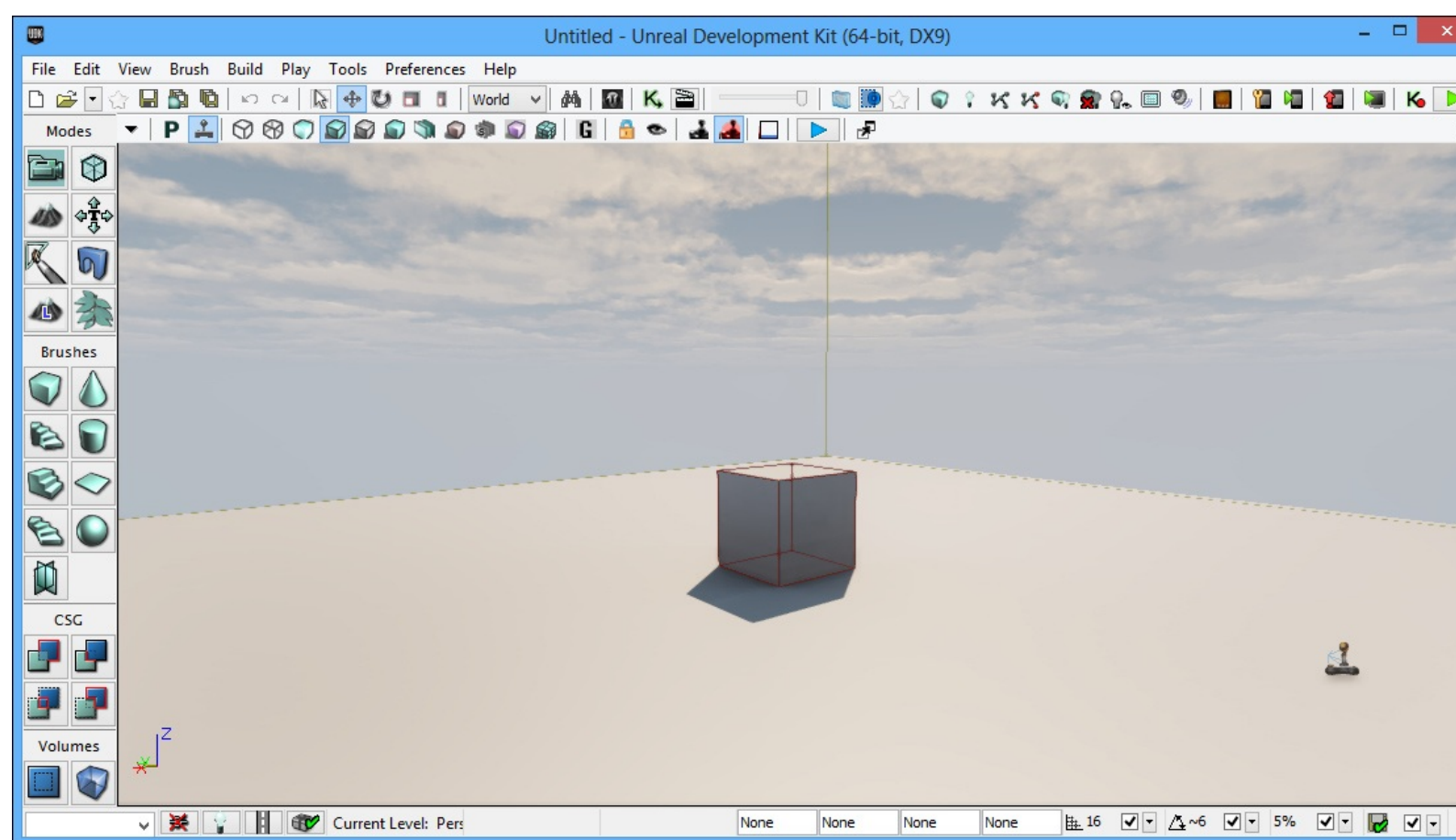
That being said, I'll do my best to be as descriptive as possible about how to make a playable version of the game.

Block out simple-level geometry

A fitting start to our project would be to create a new level and create the area in which we want to base our game.

Prepare for lift-off

Before we start working on the project, we must first create a new map. To do this, we must first navigate to **File | New Level...**, and from the pop up that comes up, select one of the four options (I selected **Midday Lighting**, but it doesn't matter which option you choose).



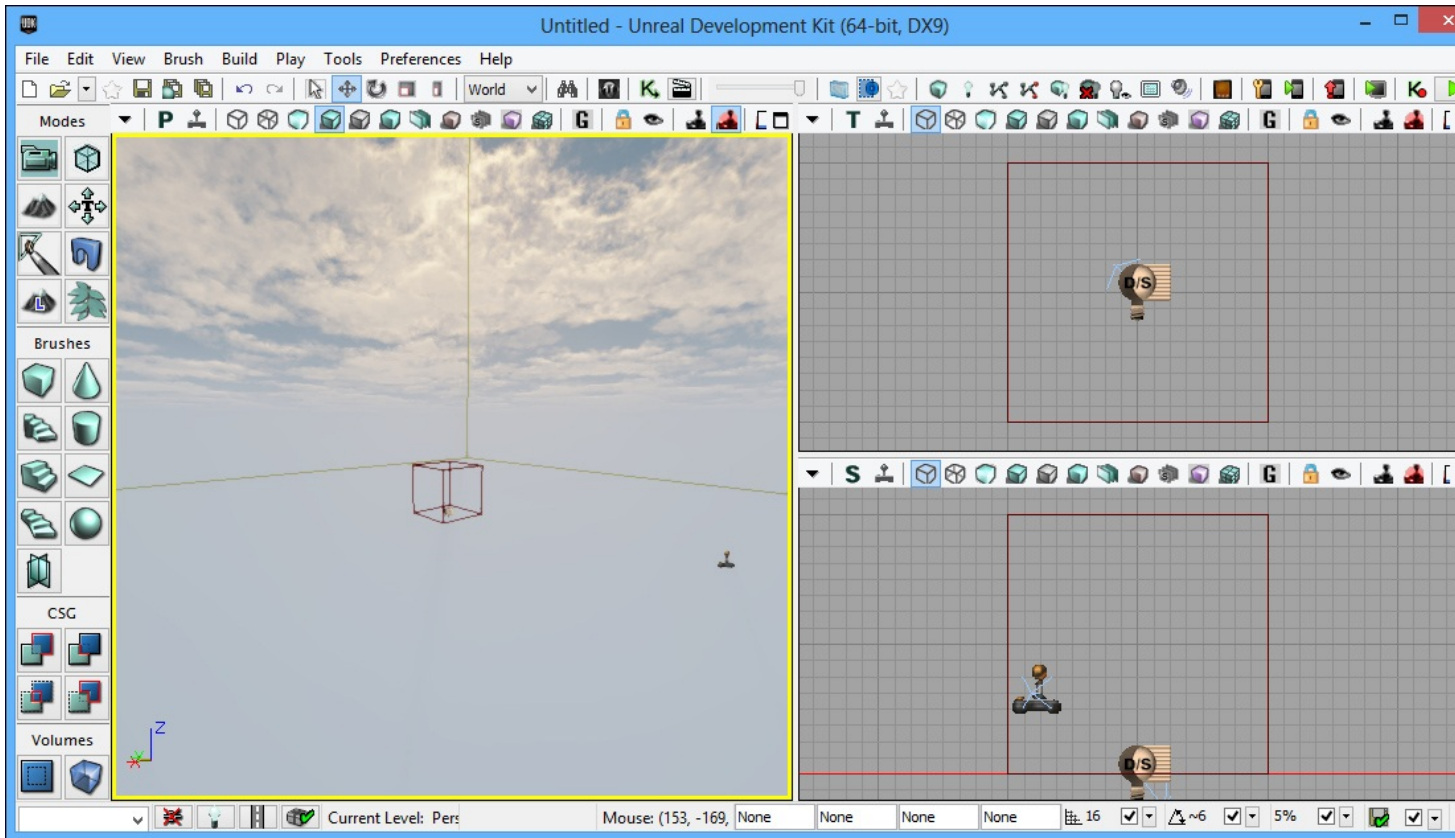
Engaging thrusters

Now that we have a base level to work with, let's start building our game! Perform the following steps:

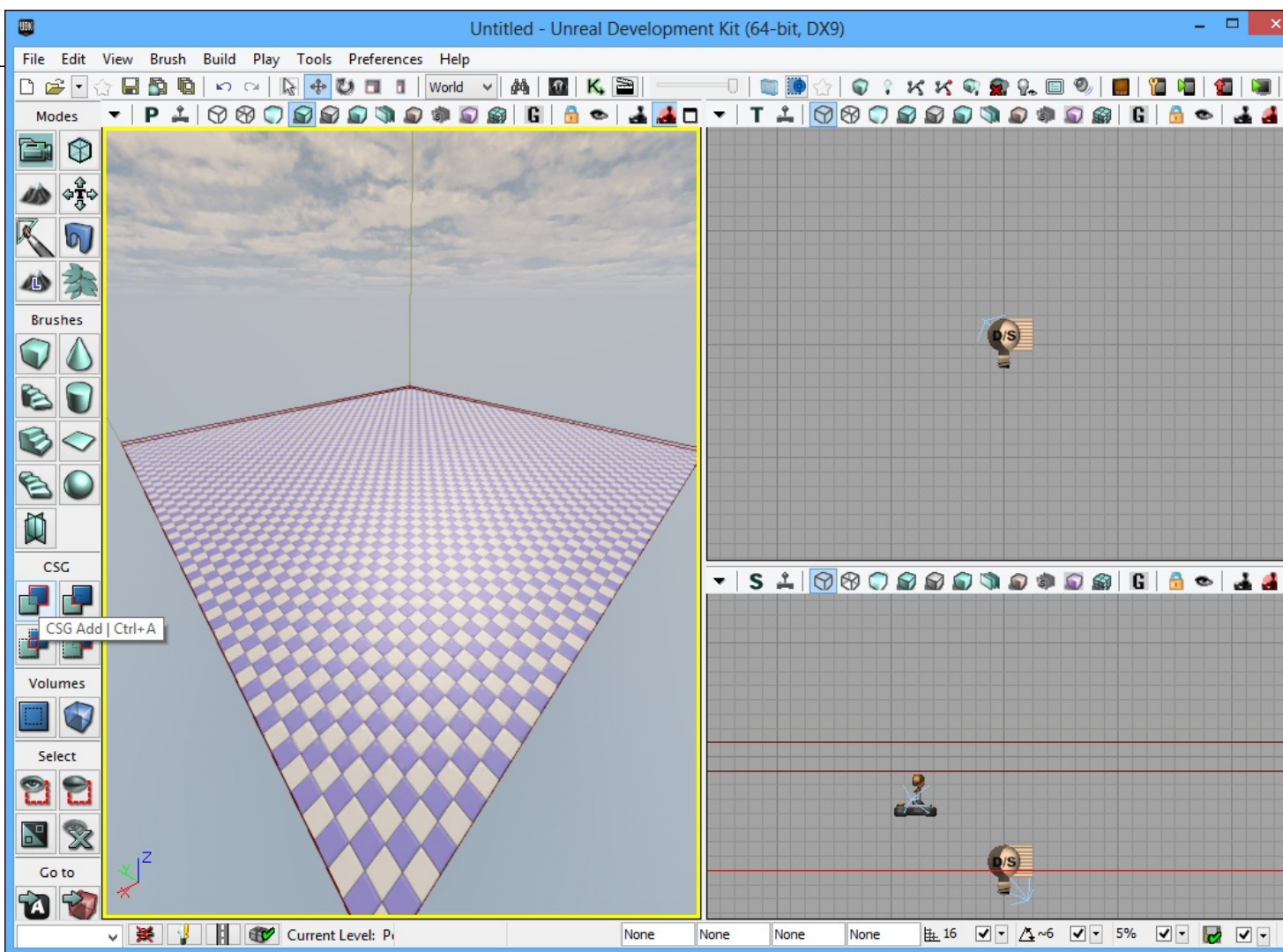
1. If your viewport is zoomed in like the previous screenshot, click on the **restore viewports** button on the top right of each of the viewports (the icon that looks like two windows). Upon creating our level, we are greeted with a nice scene with two meshes. Let's delete this; we don't want it messing with our stuff. Make sure you click on the actual mesh and not the (red) builder brush when deleting the two objects. To delete an object, simply click on it and press the *Delete* key.

Note

There are many different options that you can choose from to determine how the UDK is displayed and how it works for you. I encourage you to take time to figure out what you like and don't like. While having a front viewport may be nice, I like having a larger screen space for the perspective view so I have a better idea about what the area I'm creating looks like. This is more my personal preference than anything, but it is what I will be using from here on out. If you wish to follow me, navigate to **View | Viewport Configuration | 1 x 2 Split** from the top menu. For those of you using multiple monitors, you can also make use of the **Floating Viewport** option by navigating to **View | New Floating Viewport**.

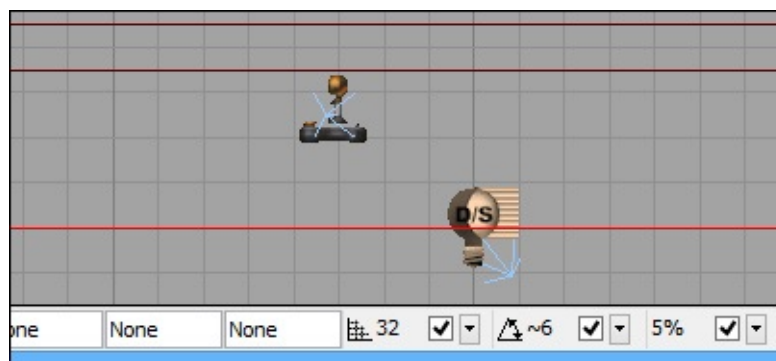


2. Once the two previous objects are destroyed, right-click on the Cube button on the left of the **Brushes** section and bring up its dialog box. Fill in the values to create our level's floor. I used 4096 as the length (X) and width (Y) of my level with 32 for the height (Z). You can change it to whatever number you want, but I'd suggest you stick with a number that is a power of two (32, 64, 128, 256, 512, ...) as computers work best with them.
3. Click on the **CSG Add** button, which is on the top left of the **CSG** section in the left toolbar, in order to add the brush to our level:



- Next, change the **Grid Locking** amount to **32** by either using the drop-down menu or pressing **J** until you see it there from the menu on the bottom-right of the screen. Also make sure that **Drag Grid Snap** is enabled by making sure the box next to it is checked.

Grid snapping is very useful when working on projects with the UDK. Grid snapping enables people to build brushes, making sure they are seamless with no holes in the game environment; this can make building levels much easier. You should always make sure the drag grid is enabled when working with brushes and make sure that you keep the vertices of your brushes on this grid at all times.



- Press the **B** key to hide the builder brush as we will not be using it any more. Select the brush that we first created, and from the side viewport, zoom into its top-left edge

- [download online The Drama of the Gifted Child: The Search for the True Self \(3rd Edition\)](#)
- [Collected Poems.pdf](#)
- [click The Complete Modern Blacksmith.pdf, azw \(kindle\)](#)
- [download Better Off Undead](#)
- [The Burning Man here](#)
- [read Hacking Exposed Mobile: Security Secrets & Solutions \(1st Edition\)](#)

- <http://fortune-touko.com/library/Artemis-Fowl-and-the-Arctic-Incident--Artemis-Fowl--Book-2-.pdf>
- <http://ramazotti.ru/library/Collected-Poems.pdf>
- <http://ramazotti.ru/library/The-Conquest-of-Bread--Working-Classics-.pdf>
- <http://www.cafesystemcanarias.com/books/Better-Off-Undead.pdf>
- <http://jaythebody.com/freebooks/Perfect-Health-Diet--Regain-Health-and-Lose-Weight-by-Eating-the-Way-You-Were-Meant-to-Eat.pdf>
- <http://conexdx.com/library/-Yellow-Kid--Weil--The-Autobiography-of-America-s-Master-Swindler--Nabat-.pdf>