

# How To Make Video Games

**The ebook that shows you everything  
you need to know to make video games**

**Brought to you by:**  
[Make-video-games.com](http://Make-video-games.com)  
**And**  
[Rentgamescheap.com](http://Rentgamescheap.com)

# How to Make Video Games

The Free ebook on game making

Brought to you by:

<http://www.make-video-games.com>

and

<http://www.rentgamescheap.com>

For Regular updates and information about making video games sign up for the [make video games Newsletter](#)

This ebook is a collection of essays and a tutorial on making video games. The essays are independent of the tutorial and give you tips, hints and guidelines on game making. They are indispensable to the whole art of making games. The tutorial is based on the Reality Factory Game development software. There are some copyright requirements with the Reality Factory software and the Genesis 3D game engine it is based upon. A copy of the copyright is included in the back of this ebook and here are links to the copyright notices for both programs are here:

[Reality Factory Copyright Page](#)  
[Genesis 3D Copyright Page](#)

It is recommended you read the copyrights on these products. The programs are free to use but there are restrictions you should be aware of, particularly if you are considering making video games to sell to other people.

## Coyright for this ebook

**This Publication is copyright©2007 Kalif Publishing, [www.make-video-games.com](http://www.make-video-games.com) and [www.stormthecastle.com](http://www.stormthecastle.com) You may freely distribute this ebook as long as you make no modifications. You may not sell this ebook. If you wish to reprint portions of this tutorial or the articles contact the webmaster at [www.make-video-games.com](http://www.make-video-games.com)**

This Ebook is provided "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH REALITY FACTORY OR THE USE OR OTHER DEALINGS IN REALITY FACTORY.

## **About this Tutorial and Disclaimer**

This is a “basics” tutorial that will give you some good information about making video games. The profession of making video games is an exciting and very complex one and there have been hundreds of books written on the subject – books sometimes exceeding thousands of pages. This tutorial is not that in depth but it will give you a nice first experience with making games.

Significant time and effort went into the making of this tutorial and lengths were taken to make sure everything was accurate. Of course nothing is ever perfect and software is continually upgraded and changed. I ask you take the spirit of this tutorial in that it is free and is a good resource for helping you to learn about making video games. Please respect all the copyrights that are associated with this document and with the software that is referred to. And of course, being free, there is no real support for any issues or problems you may come across. But if you would like more resources and information about making video games visit the authors site at <http://www.make-video-games.com> or if you like the people helping people concept you can visit the forum section created for the make video games website. There you may be able to ask questions or even answer questions by other game making enthusiasts. This is a locked forum so you will have to register first. <http://www.castlefiction.com/fantasyforum/>

With that being said, I hope you enjoy this tutorial! And I hope you design the next big game!

## **Table of Contents**

### **Articles about game Making**

1. Introduction to Video Game Making and Creativity – Thinking Like an Artist
2. Keeping and Managing your Gamers Book
3. Designing on Paper is just as important as work on the computer
4. Careers In video Game Making
5. What Software should you use to make a video game?
6. Seven Creative tips for Game Making
7. About Writing scripts for your video games
8. If you really want to see fast results – How you can make your first game in 1 hour

### **The Tutorial**

1. Using the Reality Factory Software
2. Lesson 2: Familiarizing Yourself with the Workspace
3. Lesson 3: Now let's add a character and actually walk around inside this room.
4. Lesson 4: Now you actually have your very first game and most of the hard work was actually already done for you.
5. Lesson 5: In this lesson we learn how to Focus down onto parts of the game world and apply different textures to the different shapes and surfaces.
6. How to Manipulate Tiles and Textures for a good look in your video games
7. Lesson 6: Lighting in your game
8. Lesson 7 Adding simple objects to your game
9. Lesson 8 Texturing surfaces of objects
10. Lesson 9 Adding a second room and cutting a doorway
11. Lesson 10 Leakage in your game
12. Lesson 11 Cutting the doorway
13. Lesson 12 Adding a Pool of Water
14. Lesson 13 Adding a Pool of Water continued
15. Lesson 14 Adding a Pool of Water continued
16. Lesson 15 About Graphics in your game
17. Lesson 16: How to Make and Import your own Graphics into your video game
18. Lesson 17: Textures, What they are, how to get them, how to make them and get them in your game.
19. Lesson 18: Textures continued
20. Lesson 19: Using Texture Packer to import graphics
21. Lesson 20: Review and More Resources
22. Reality Factory and Genesis 3d Copyright notices

# Designing Video Games: Sharpen your eyes and think like an artist

Good Video Game design is not just about fast action and challenging beasts and creatures. One of the most important, yet most overlooked aspect of video game design is the believability of the world the game takes place in. When designing a video game you don't have to be a great artist but you do have to think and see like one.

The world of video game design has come a long way in the past few years. There are design suites that cost very little or even nothing that enable you to put together a world in less than a week. This has changed the way an amateur develops a game. No longer do you have to spend countless hours working out the code or the scripts. You can now focus your attention on the look and feel of the game. This means you have to start thinking and seeing like an artist.

There are things that you wouldn't even think of that artists know and see. Do you look at a building and realize the design of it? Do you see the play of light on the house next door. Do you realize how this light changes through the day. You can start to look at the world the way an artist does. And it will dramatically increase how good you are at making video games. Awaken the artist within you and look at the world around you. There is much to be learned that you can apply to the world of video games.

**Look at the design and layout of the house you live in. Look at the way the yard is shaped or the neighborhood. How do the buildings sit in comparison to each other? What are the spaces between buildings like?**

If you are beginning with the quest to make a video game and you are looking for practice then I recommend you take these Four steps. They begin with the center of where you are and spread out in concentric circles:

Step 1: Do a copy of the room your computer is in. Make it as realistic as possible. Add the furniture and windows. Even add the computer. This exercise is terrific because it develops your video game making skills and it also helps you to really look at things. You look at a chair in a whole new way when you have to redesign it for the game world.

Step 2: Now do your whole house or apartment. Create all the rooms and put the doorways and windows in. You don't have to do all the furniture –just get a feel for the design of your house or apartment. Where are the stairs? How do the windows change the feel of a room. How do they change the light? How high are the ceilings? Are the ceilings all the same height? See what I mean? When you look at the world as an artist you start to see and learn new things.

Step 3: Now do a mock up of your neighborhood. Design all the buildings, place them on the street. You don't have to get real detailed. The point is to get a feel for the design and layout on a larger scale. Wouldn't it be great to make a character that could walk around in your neighborhood in a video game? Make sure you go outside and look.

“You have to look. If you do it from memory you are not getting the full experience. When you look at the neighborhood with the intent of putting it into a video game you will see things you never saw before. Take a sketchbook with you or a digital camera”  
Step 4: Go to an outdoor location that you are not familiar with –a nearby park would be ideal for this. Bring a sketchpad and colored pencils with you and do lots of sketches of the environment. Try to see as an artist would see. Notice how the sidewalks curve in a way that accentuates the grassy areas of land. Notice the differences in color.  
When you are trying to design video games try to see things in a new way. Think about the light and how it changes. Think about the colors of things and how the colors within a scene interact with each other. Notice the flow of structures and the way people move within and around these structures.  
Video game design as an amateur can be a very rewarding and a very fulfilling pursuit. But it is a skill and as with any other skill one of the best ways to learn is to look at the work of the masters of the craft. By this I don’t mean just look at other video games. To see the world you are building with the eye of an artist you should look at the works of master artists like Rembrandt, DaVinci or even Frazetta. All of the things they do will teach you how to visualize your video game world in ways that will bring it to life.

## **The Game Makers Notebook**

### **Keeping a Notebook for all your game Resources**

If you are making video games or are considering the idea of making your own games you need to organize yourself in a way that will help the process. Keeping a game making notebook is the perfect way to do this. Here is my recommendation on how to do this. Along with pictures of my game notebook.

#### **The Notebook - Get Organized!**

Making a video game is a creative process and that process can sometimes be very crazy! You never know where it will take you. One day you might be working on software and the next day you could be drawing sketches of characters or writing story lines for quests and the game world. You have to keep all this different types of information organized so you can find it and use it in the future.

There is nothing worse than remembering that you had a great idea but not being able to find the paper you wrote it on.

This problem of organizing your information is greatly exaggerated with game making because it requires so much different information. You have graphics, text, tutorials, hot

keys, software, stories, pictures, ideas, levels, resources and maybe even sounds and songs! So you have got to start out , right at the beginning with an organized place to keep all this.

Here is what I use for my game idea notebook. It is a Mead 5-star organizer that has lots of room and lots of pockets.

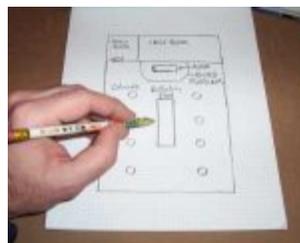


What do I keep in it? I keep tabbed sections for different games, a sketch pad, a music notation pad, graph paper, software, CD's, floppy disks and pencils. As you do some game development you will be able to grow your gamers notebook in any way that is comfortable for you. But you absolutely have to do this. It will make a big difference in how you design games. And this organizational skill will transfer very well to the work world if you are thinking about a career in game making.



It would be almost like a portfolio you could show a potential employer.

## Designing Your Games on Paper



One of the most important things you have to learn when learning how to make video games is that there is actually a lot of work to be done away from the computer. Although today's software is fantastic and it does a lot of work for you there is still a lot that you cannot do anywhere else except on paper.

You really need to learn how to do some of your video game design work with paper and pencil and there are four very big reasons for this:

Speed - Paper and pencil is very fast - You can quickly sketch out many ideas and transform the kernel of an idea into something workable that you can make on the computer.

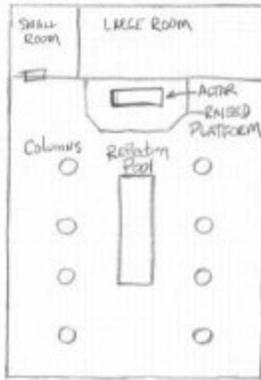
Creativity - there is simply no better way to experiment with your creativity than with pencil and paper. you will see things in many different ways and you will be able to get very creative. It only takes a matter of seconds to draw out several rooms then redraw them as ideas flow. T

History and Notes - Your sketches and written ideas become a history you can keep. Often times you will remember a certain room sketch or a certain layout that would be perfect and having all these drawings on paper will allow you to quickly look back.

Growth of your skills - When you work just with the computer programs you tend to fall into patterns that are easy to do with the software. Things tend to be more square and more uniform. But when you draw things out on paper they tend to be more complex and then when you try to build what you drew you have to work harder and get more out of the game design software.

You can make video games without ever putting anything on paper but I hope I have convinced you that you will be a much better designer and your games will be much better if you add the skills of drawing on paper to your repertoire. IT will make you much better!

Let's Go ahead and Do a little Designing of a video game on Paper



### A Dark Temple -

Here is a drawing I have done of a dark temple. It looks pretty good so we are going to go ahead and build this. Let me point out some of the highlights.

The eight circles represent columns in the main room. The reflecting pool is going to have water in it. The altar is on a raised platform and there are two rooms in the back of the structure.

This is a good little practice session because you are going to get to do some things you have already done and you are going to learn some new things like how to add water and how to add doors to your game.

This dark temple was very easy for me to create on paper but it would have been hard on the computer because of the time it takes to create individual objects. The first big room then the small rooms then the columns etc all take time and I might have lost the overall vision.

Okay, so before we go on to create this little game level on the computer I recommend you get some graph paper and do some level creation on your own. And remember to save everything you do because all of your drawings will be ideas you might want to turn into a reality in the future.

About the paper: I use graph paper for my level designs and I highly recommend you use graph paper too. The little grids of squares makes it very easy to draw everything straight and it will be a big help when creating the computerized versions. You can count out squares on the paper and then count our grid squares in the software.

# A guide to starting a career in Video games – Prepare yourself for a break-in!

Starting a career in the video game industry used to be a difficult thing; there were very few options and very few schools that could teach the required cutting edge skills. If you wanted to choose this path you had to be self taught. Today this dilemma has been turned upside down. You now have a plethora of schools to choose from and lots of programs and specialties to decide amongst. Here is some advice on what the industry has to offer and what you can do to prepare yourself.

The video game industry has changed over the past ten years. It has matured and is now a full-fledged industry. This means that there are plenty of opportunities to find a rewarding career without being a hot-shot game programmer. As a matter of fact, the days of the hot shot developer are pretty much over. In today's world the creation of a state of the art video game now requires a dedicated team of employees with a wide range of skills and interests.

What does this mean for you? It means opportunity in a lot of different disciplines and it means that you have to specialize within a subgenre of the big picture of "Making Video Games". But before you pack off and go to a video game school or dish out the dollars for an online school there are some things you should do and some things you need to know.

## 1. Learn the basic concepts of game design

**Have an understanding of video games**, how they're played, and what they're all about. You should have a feel for the game industry; know what games are out there, what styles of play they have, and how they handle the game playing experience. Chances are good that you already fulfill this requirement.

**Have an understanding of the underlying principles of game design.** There are several ways that you can do this. I recommend one or both of these methods to get you started in this:

**Build Mods** – Mods are modifications of existing games. Quake is a famous game for this approach. You use the mod builder software and you build levels or worlds that can be added right into the game itself. The Mod sub culture is very vibrant and you can find a lot of support and help. To gain even better experience that translates well into a corporate environment you should join a volunteer Mod development team. This gives you good teamwork experience and gives you insight on how a video game project is broken down into pieces.

**Get a free or inexpensive Game engine development suite and build your own games.** You can build adequate and fun to play games on your own this way. I

recommend you do this with the Genesis 3d Game development software. It is completely free.

## **2. Get familiar with some of the professions in the industry and the tools that are used.**

There are a whole host of tools that game design teams use. And the tools you would use are dependent on what role you would play on the team. Here is an overview of some of the different jobs in the industry and what software tools they would use.

If you like drawing and painting with traditional tools you might want to consider being a concept artist. They do all their work with off computer art tools. They would conceptualize and draw the scenes and the creatures in a game.

If you think you might like computer designing worlds and scenes in a video game you should consider being a 2-dimensional artist. They create all the structures, textures and backgrounds in game worlds. Photoshop is the de-facto program for this job. If you can afford a copy of this program you will get a definite insight into this career path. If you cannot afford Photoshop you should consider getting Paint Shop Pro. It is less expensive and very similar to Photoshop and it will give you a good foundation into the world of 2-dimensional artwork.

If Characters and models are what you are enthusiastic about you should consider becoming a 3-Dimensional Model Artist. This type of artist is concerned with making the characters, monsters, creatures and objects that inhabit a video game world. In this area of game art the choice of software is not as straight-forward as in the 2d art. The most accepted software is 3D studio Max. Two other popular programs are Maya and SoftImage. If you are looking for a free 3d modeling tool that will get you started you may want to get Milkshape. It is a tool designed for the game Half-Life.

The concepts of modeling are universal so the skills will transfer from one tool to the other. You will just have to learn the specifics of the new tool which will be easy enough.

**Animator** - This is the person that makes the characters and objects in the game move. It is a very specific skill and it takes good understanding of animation to get things right. There are many software programs that are used for animation and they include Macromedia Flash, Alias/Wavefront, 3D Studio Max and Lightwave 3D.

**Programmer** - This is something that is a little more traditional when we think of video game making "Programming" and it is a very wide field that can find you doing almost anything in the video game making arena. A programmer could be called upon to do any number of things from managing a database to designing the artificial intelligence within a game or creating specialized software tools that other designers use to create the game.

**Other Opportunities in Video game design** - There are many other jobs in the video game industry ranging from Script and story writer to even network administrator or computer maintenance and repair technician.

Today there are many different career paths that can be taken into the video game industry. This requires you to specialize in a set of skills. Understanding what these different skills are and getting a head start now will increase your marketability in the very competitive video game industry.

## **What software should you use to make a video game?**

The answer depends on a couple of questions you have to ask yourself? What skill level do you have with the computer and programming and what are your goals for making video games? I answer these questions and help you find the right software for achieving what you want.

Video game design used to be the realm of crack shot programmers who could write programs, troubleshoot computers, write tools and even draw out the graphics needed. You had to be a writer, artist, and programmer all wrapped up in one. But today this is no longer true. There are plenty of programs that do most of the hard work for you and leave the fun part of actually designing a game to you. But every software tool is different and each one is suited to a particular type of game making. Here is a guide for understanding some of the major game design suites and choosing which one is right for you.

For Absolute Beginners to game design and for pre-teens or young adults –

You are the type of person that knows how to use a computer but you don't know how to do any programming at all. You would like to make a few small video games without having to learn too much software stuff. You have lots of ideas for video games and you like the arcade style and side scrolling games. This kind of game would be a thrill for you to make for you and your friends to play.

My Recommendation:

The Games Factory or Games Factory 2 by Click team

This is an enormously successful software tool you can use to make arcade style games. Everything is drag and drop so you don't have to learn how to program and you don't have to draw up your own graphics and images. You will learn some of the fundamental concepts behind game making and have a blast doing it.

For medium skill computer users, for high school age to adult

You have a bit of Savvy with the computer and have tried and used different kinds of programs. Maybe you have done some level design for a popular game. You would really like to make something three-dimensional like Quake and you have the thought that maybe you would like to make whole worlds and dungeons. You really want to make a game that you can walk around in.

My Recommendation:

The Reality Factory

This is a shareware program that is built on the Genesis 3D game engine. It is a legitimate game engine with a great workspace that you can actually design whole worlds in. The process is similar to something an architect would use -picture drawing out blueprints because this is what you do with the Reality Factory. This software is free to download and within an hour you can actually have a single room game up and running where you can walk your character around and shoot weapons. To do a more involved game you need to spend a significant amount of time drawing out and texturing the rooms. If you get proficient with this free software you will have set for yourself a good foundation in legitimate game making. You will have an understanding of game processes and items like textures, models, scripts and more.

For high skill computer users and adults age 18 and over

You are considering a career in game design and want to set out a good base foundation for your career. You want skills that will transfer to the Real world but you are not yet sure if you want to invest the years it takes to master a high level programming language.

My recommendations: Dark Basic

I recommend you look into the program called DarkBasic by The Game Creators. (They also have a Dark Basic Lite version). This is an excellent way to learn how to make games while you learn the protocol and logic of programming. This is a powerful tool that can make professional games. The good thing about Dark Basic is that it is specifically tailored to making games.

You are going to be a Video Game Maker – no doubts about it

My Recommendation: C++ and Direct X

– Although this is a matter of opinion I recommend you learn how to program using C++ and you learn how to develop with something called Microsoft Direct X. This is some professional and very serious stuff and you will spend months learning the languages without ever seeing a single game pixel. But this route is how to lay out a professional career. This software is also pretty expensive. But imagine the rewards?

Making video games is a wonderful pursuit that can fully tap your creative powers and whether you are doing it just for fun or for a career there are lots of tools available to help you reach your goals and make some great video games.

## Seven Creative Tips for Amateur Video Game Makers

Making a video game means working with a software program that guides you toward building worlds in very specific ways. Do not fall into the trap of letting the software determine your game. Keep creative control in your hands. Here are some suggestions on how to keep the creativity flowing.

Today's video game design software makes it very easy to create worlds in ways that it was designed to make them. This guides you in preset directions which are inevitably away from the creative ideas you had in your head. But it is so easy to just design the game in the preset way and it takes effort and work to overcome this. Here are some of the things you should do to keep the creative control in your hands.

**Keep Designing on Paper** – Keep a notebook and a sketchbook to draw and write your ideas. A blank piece of paper will not limit your imagination. When you try to transfer this to the game you will be forced to learn new skills and get the maximum out of the game software. You should also keep this notebook with you when you play other peoples games. Not to copy, but to analyze what happens and to write down your own original ideas that come from playing.

**Think about the atmosphere and mood** – Games are more than just fighting creatures and discovering treasures. The most important part of a game is the overall atmosphere. Keep this in mind. Consider what the overall mood of your game will be. Use music as an inspiration to get you in different moods.

**Be careful with rewards** -- One hundred tiny rewards are generally better than one big one. The constant seeking of the next tiny reward keeps the player moving forward and keeps a goal ahead of him that is achievable.

**Don't design your game linear** --Today's games are designed with the individual player in mind and this means choice. Different players like to follow different routes through a game. Some like to hack and slash, others like to follow quests and others prefer the interaction with non player characters. Add all these elements in your game to give your player lots of interesting options.

**Expand your horizons** -- Pick up art books that don't normally appeal to you. Look into areas and subjects that aren't on your normal list. This is a great way to get new ideas and to add new dimensions to your existing material. What could a book about porcupines teach you about video game making? Porcupines have an amazing defense system. Could this be incorporated into your game somehow? This goes for almost anything. Any subject at all from psychology, to sociology, or zoology are great resources to add depth to your game.

**Don't be afraid to make mistakes** – Mistakes sometimes pay off in big ways. This is one of the ways you can think outside the box. If you are afraid of making mistakes you will not experiment or expand your abilities. If you are designing a desert scene don't be afraid to put a swimming pool right in the middle of it! Take chances. Sometimes really good ideas and really memorable scenes come from the mistakes.

Finally make sure you get honest feedback from friends and people who try playing your game. And take notes. When you are designing a game you get too close to it. You know everything that happens and you know where everything so you understand innately how to play. You have to remember that your player will have no idea what is around the next corner and you have to design for that.

## How to write a video game script

Writing a video game script offers a challenge that goes well beyond the normal realm of writing. But it is also something that can be tremendously rewarding in the scope of its creativity. Here are some guidelines and tips for writing a video game script.

Today's video games are based in complex worlds and they tell stories. No longer does a player simply advance through repetitive screens slaying goblins and ghouls in a quest toward the goal. A player now expects to progress through a world where there is a rich history and a plethora of decisions to be made. This adds to the complexity of writing a video game script and it also adds to the richness of the creativity involved.

The first thing you need to think about is that writing a video game script is that it is not the same as writing a movie script. The two processes are similar and you do write a movie like script for your video game but that is only part of the process. There is a whole host of accompanying materials that you need to write for your game script. Here is an overview of what you need to write and why.

### **Write An Executive overview of the story in prose**

This is the most important part of your game script and this is what will sink or float your script. This overview has to tell a compelling and unique story and it should tell the complete story from the opening scene of the game through the major steps all the way to the completion of the game. An overview like this can be almost any size and it would be

very easy for this to be ten written pages or more. Remember that today's video games are very complex and the stories can be very complex. This overview is also the most important part of the script. You would show this to game developers to see if they are interested in developing it into a game.

### **Write a History and Background of the world**

Video games are complete worlds and game designers need to know what the world is like and what kind of history it has. This will help the designers to visualize what the world will look like.

**Create a Flowchart for the entire game** – Your game is going to be very complex and there will be many decisions that the player will have to make and each decision opens up a whole new path for the player to take. Creating a flowchart is the best way to keep track of all the possible paths through the game.

### **Create sub-quests and write a prose overview of each quest**

Sub quests can be simple or complex but each one is a story in itself and you must tell these stories.

### **Create character descriptions and bios for all the major characters in the game.**

Game designers need a complete picture of the characters in the game. Many of the non player characters you create will pop up time and time again. And their story is woven deeply into the fabric of your world. You need to describe this relationship in detail to the game designers.

### **Write interactions with non-player characters**

Your game will probably involve interaction with non-player characters (NPC's). You should write out the dialogue and flowchart the choices the game player can make. These interactions are often critical to the story and they can take the player on very different paths toward the conclusion of the game.

**Write Cut scenes** – Cut Scenes are short animations or movies that come before or after major plot points in your story. A cut scene should always be written to enhance or describe the story. A cut scene is also a reward given to the player for achieving a major milestone in game play.

### **Writing the actual storyboard script**

This is the final step in the whole video game script writing process and it is the most detailed. You do this step last because you need all the supporting materials to understand and describe this correctly. This part is very similar to that of a movie script. You progress through each scene of your story and you detail all the necessary information. Here is an example:

Scene 1:

**Location:** A dark cathedral with stained glass windows. An NPC is kneeling before a stone casket in the center of the main room

**Music:** background music of an organ playing introduces the scene but subsides

**Characters:** Main player, NPC named Thomas

**Player Goal:** Discover the location of the underground lair

**Action:** Player must initiate discussion with Thomas, upon first contact we activate cut scene (1) where Thomas morphs into a were-creature and summons his were-minions. Main character must battle the were-minions then re-initiate discussion with Thomas.

**Flowchart:** No decisions made at this point: If battle is completed Thomas reveals the entrance to the underground lair and player advances to that level. If player is defeated in battle revert to death cut scene (11) and move to try again screen.

**Notes:** Player is locked in the cathedral and there is no exit. The only viable way out is to initiate contact with Thomas. Random were-creatures can be activated if player explores cathedral before talking with NPC.

When writing a video game script you have to remember that your primary audience is not the game player but the game developer and what the developer needs is a complete picture of what your game is about. This means that you are not just writing a story but you are creating a world complete with a tone, sounds, characters, story, plot, and subplots. To successfully communicate this to the developer you need to use a whole set of creative tools and this is where video game scripts depart from normal scripts and open up a whole realm of creative possibilities.

## **How to Make a Video Game in One Hour – Really!**

Video game design software has taken a lot of the hard work out of making games and it has left the creative part up to you. There are software applications that will help you make a game quickly, easily, and in only one hour.

### **Make a two-dimensional side scrolling game**

This is your best option for making a full-fledged, and fully functioning, video game in an hour. The best software for this is the Games Factory 2 software available from a company called Click Team. They have a free download trial version of the software so you can give it a try before you purchase it. And it comes with a tutorial that takes you through all the steps of making a simple breakout style game. This is the kind of game where you control a paddle to deflect a bouncing ball in an attempt to break blocks on the top of the screen. I downloaded the trial software and followed the tutorial and it took me a total of an hour to complete the game.

### **The Concepts of Game Design**

The Games Factory 2 software gives you all the basic skills you need in understanding how video games are made and after using the software you will be able to make multiple

screens, add sounds and music, and you will understand gaming concepts like asset management and collision detection.

### **Three Dimensional Dungeon Type Games**

If you are really determined to make a three dimensional game and you don't want to make side scrolling two-dimensional type games you can get the free software called Reality Factory and in about an hour you can have a very simple single room game up and running. In this game you will have a character that will run around and shoot a gun. This kind of accomplishment is actually quite surprising and you will be amazed as you walk your character around your single room game. This kind of game development software is much more complex than the side scrolling software and it will take much longer to really get good with it. But it is something that you might find is worth the effort because you will learn how to make extremely rich and interesting game worlds.

Video game making is no longer the territory of the genius programmer who can whip up thousands of lines of code. The software available today has taken all the programming out of game making and given you the freedom to express your full creativity.

# Using Reality Factory to Make Games

## Lesson 1 Introduction The "No Cost" Way to make great video games.

The Reality Factory is a complete (and free) system for making your own video games. You are free to make your own games, pass them along to other people and even sell them without paying a royalty. There is a user license that you will have to read before using it though. See the [Reality Factory Website](#) for the license and a copy is included at the back of this book.

About the "No-Cost"

When I say this is a no-cost way to get it done what I mean is that it will cost you zero dollars. But it will cost you time and effort. You have got to invest the time and do the work. Nothing is really free!

What this tutorial is and is not

This tutorial covers the mechanics of making a video game. It shows you how to use the Reality Factory software to create games. What it doesn't show you is how to build a story line or a game plot or how to build characters. These things will come up in a future tutorial. This tutorial is all about the making of a video game world.

Programming?

Don't worry about this, you don't need to have any programming skills to use this software. It is all drag and drop stuff.

Lets get started!

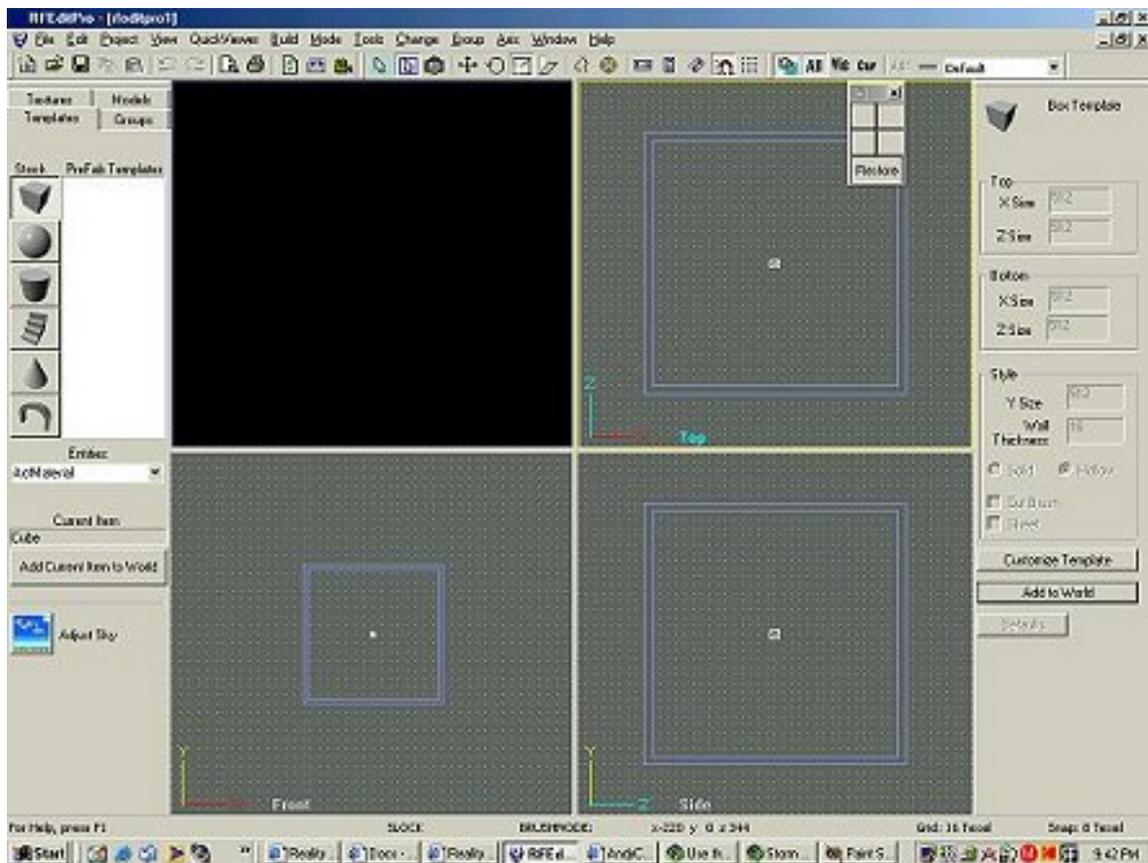
This is the link that will take you to the Reality Factory Software. bookmark this page then go to that site, follow the download link and download the program called RealityFactory 0.75A. It is about 24 Meg so if you are on a 56k modem it is going to take you a while.

[Reality Factory Homepage](#)

Welcome back! Open the program and install it. If you see an option for desktop icons select yes.

Reality factory is a suite of a variety of tools. Right now we are going to be using the tool that allows us to create a game world. This tool is called RFeditPro. Open up this program. It is located in your windows toolbar like this:

The Reality Factory workspace opens up. It looks like this:



This is the place where you will spend most of your time building your new world. So let's roll up our sleeves and get cracking.

You can see a lot of stuff here and it might look confusing but don't worry about it. In no time at all you will be using this like a pro.

The major thing that first comes to your eye is the four big screens. One of them is black and the other three have purple squares in them. These four boxes are views of the world you will be building.

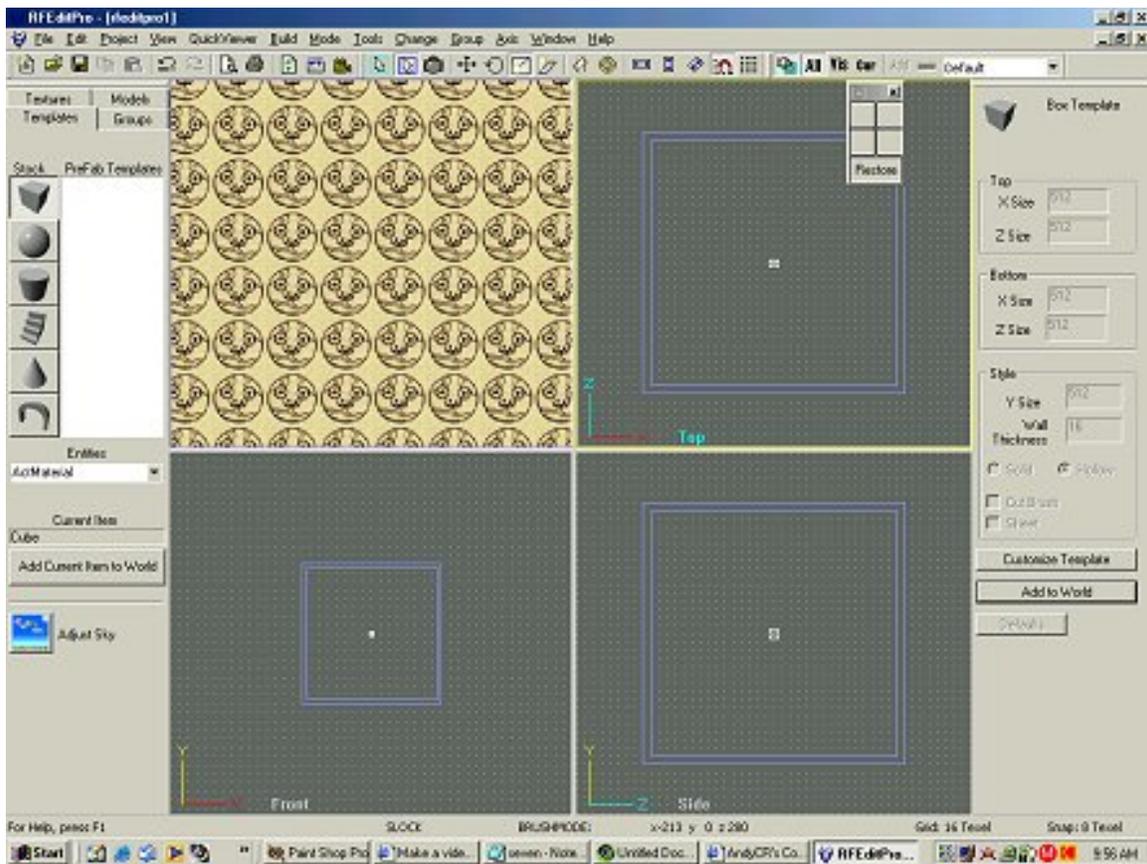
The black square is a place where you will see three-dimensional views of your world complete with applied textures. The other three views are the top, bottom, and side of what you are building.

Hang in there. It seems like a lot but in no time we will have built a room that you can actually walk around it.

## Lesson 2: Familiarizing Yourself with the Workspace

The purple lines that you see in three of the views is actually a cube and the lines represent the walls of this cube. Each wall is a double line so you can see the inside wall and the outside wall. Much like any kind of container there is an inside and an outside.

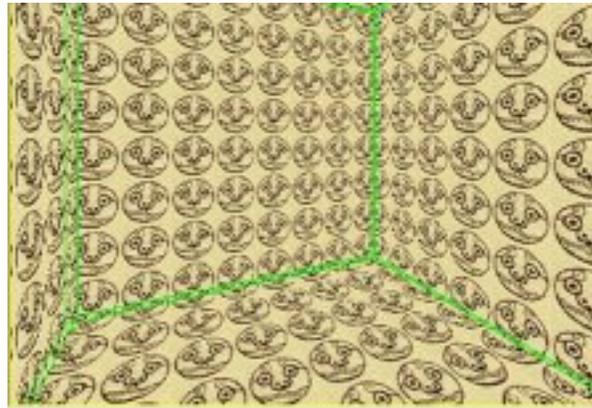
Without doing anything else now hit the enter button. The default pattern is applied to your purple cube. You can see this default pattern in the window on the upper left. It is yellow with a bunch of round faces.



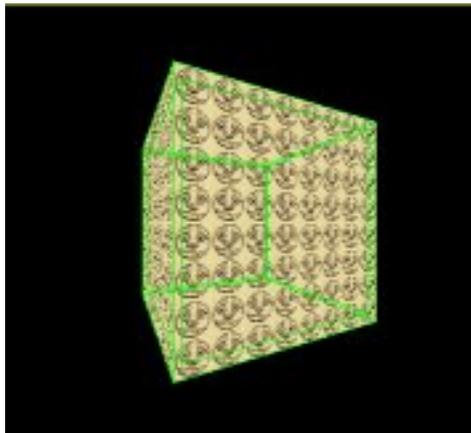
What you have done here when you hit the enter button is accepted the purple cube as a hollow object. In effect you have created your first room!

Want to take a look around inside this new room?

Left click on the textured window (smiley faces) hold and drag it around and your view changes. Drag down and you pull away. Voila the room takes shape. You can see where the walls, floor and ceiling meet.



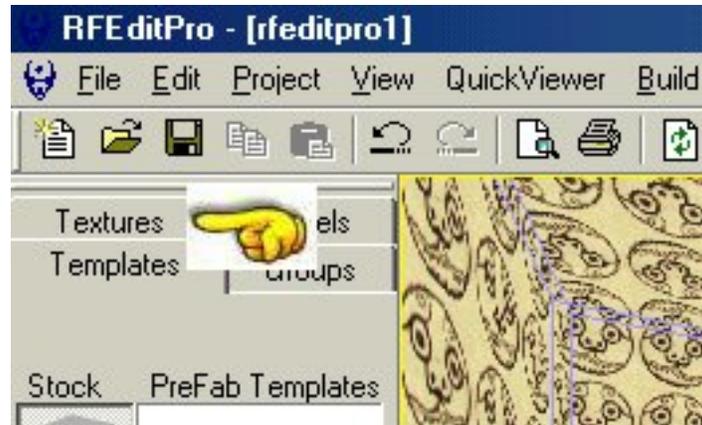
You can even pull completely out of the room by bringing your mouse down. Now you can look at the outside of it.



Not too hard now was it? Note that if you don't see the green lines in your 3d view simply click inside this view and they will appear.

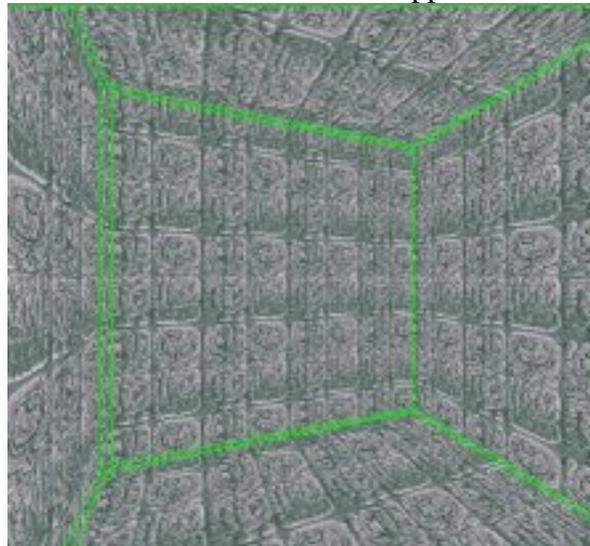
Now let's change this texture because it has definitely got to go! It can make you sea sick.

Click on the textures tab. This is where a whole variety of textures is stored.



Directly underneath this tab you will now see a variety of textures to choose from. The smiley head is also included in this list. Double click on a new texture -something a little more game like and it will be applied to your room. If it isn't applied to your room simply click on the room in the 3d view.

Here is what the room looks like with a new texture applied:

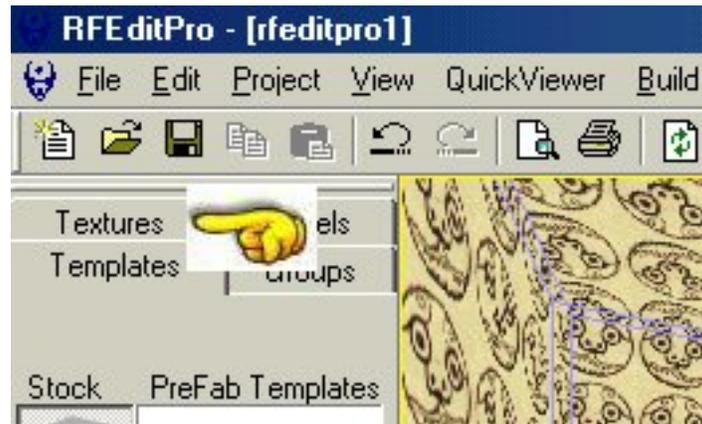


Now isn't that much better? This isn't too hard is it? You are going to get real good at this. And applying textures is very important. It gives your world the look and feel you are trying to achieve.

You have actually started making a video game. Let's continue on with the tutorial.

**Lesson 3: Now let's add a character and actually walk around inside this room. This is where it starts to get fun! You can have something up and running in just a matter of minutes.**

Hit the tab called templates.



Directly underneath this tab, and a little way down the page is a dropdown box called "Entities"



Pull down this menu and select Playersetup then hit enter on your keyboard.

This causes a small icon to appear in your room. This (Playersetup) gives your player all of its characteristics.

Now let's place a player in the game. Reality factory already has pre-made players for you so all you have to do is place it in the game.

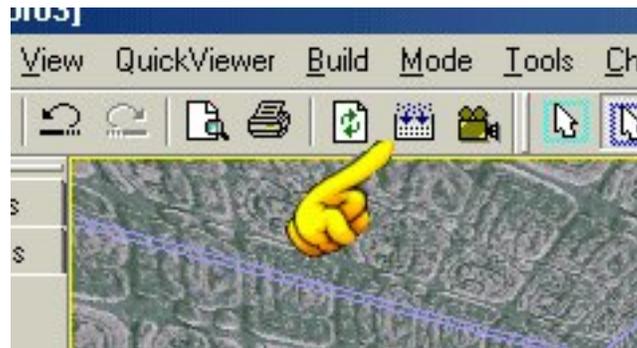
Pull down the Entities menu again and select Playerstart.

Now click on your 3d view and look around. Near the floor a lightbulb has been placed in the room. THIS is the player start. Hit Enter again and the lightbulb will change to a little red box that says player start.

This is your player!

Now lets run this and take a walk around inside.

Click the Build button. This takes the world you built and compiles it into an actual game world.



Now a menu appears. Check the box called Preview in Reality Factory. and then hit OK.



The world is built and you now can take a walk around your new room! That didn't take long did it. Reality Factory is an amazing tool that does all of the programming work for you. You can focus on the creativity of building a game.

#### **Lesson 4: Now You actually have your very first game and most of the hard work was actually already done for you.**

This Reality Factory software is designed to do exactly that. It will do most of the hard work and leave the creativity to you. You will be free to design games and worlds. Let's look at a screen shot of the one room game you built. This is the modern world of game design. All the hard work is done for you and you are left to do the creating! But as you get better you will be delving deeper into the game and you will be creating your own character, guns, weapons, and whatever else you can think of. But for now this is just a good start in your video game making.



Looks pretty good doesn't it! You saw that you started the game up and had to choose a character and a name. Then you were popped right into the room you made. Right now you can see the shotgun I am holding and in the upper left you can see my health bar and in the upper right is my radar screen where I can see enemies.

To play you would use the left mouse to fire the shotgun and the W-A-S-D keys are for motion and the spacebar is for jumping.

The player start and player setup entities you dropped into the room were prefabrications and for now this is great because you can focus on designing games and levels. But of course this is just what came with the software and you can design your own characters and players with their own unique weapons and characteristics and put them in the game. But that will come with time and practice. For now this tutorial will focus on understanding how to use the desktop to design some nice looking game levels. .

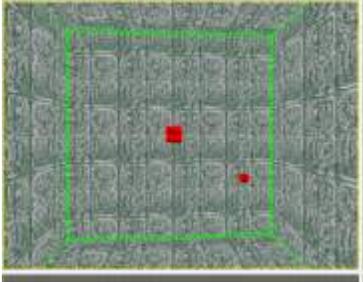
Now I recommend you review these lessons; start from the beginning and make another room. This is important because you need to do this without thinking about it. Build one or two more rooms and then run them. Try using different textures and see how it looks.

Once you have built another room or two then you can continue on with the lesson

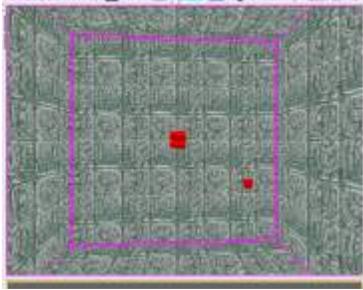
**Lesson 5: In this lesson we learn how to Focus down onto parts of the game world and apply different textures to the different shapes and surfaces.**

Think of holding a cube in your hands. There are four sides, a top, and a bottom. And each of these sides could be painted a different color. This is what your first game room is like. It is a cube and you can apply a texture to all six of these different surfaces. Except the difference between your game room and the cube is that you apply the textures to the inside!

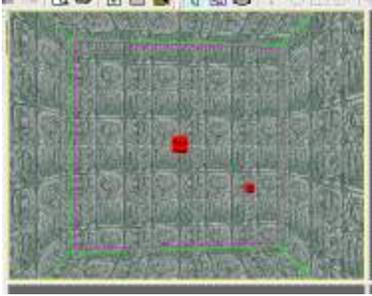
This lesson focuses on highlighting and applying textures to individual surfaces in your game world. Now this works for everything you would create in your world -not just the walls. If you created a staircase, a table, a picture, a bannister, a trash can or anything at all you would apply textures to it.



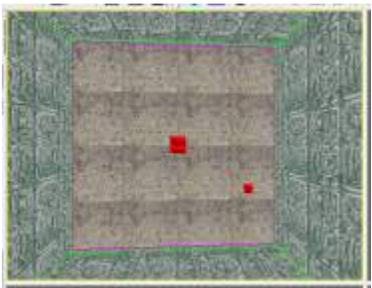
Okay, you have played your single room game and you have exited. Now you are back to the reality factory game creation screen. Single click on the 3d view and the lines of your room are highlighted green. This means that the whole room (cube) is selected. But we want to apply individual textures to the different surfaces of the room. In the game these surfaces are called "faces".



Now that your room is selected all you have to do is hit the pagedown button on your keyboard. This digs deeper down into the selected object and allows you to select individual faces. The lines turn violet in color. Now you can select individual faces by clicking on them.



In this image I have selected the flat wall or "face" that is directly facing us. What happens is that whole face is outlined with the violet lines and the other faces in the room return to green. This means you can edit that face now. So, click on the textures tab then select a texture and double click it. The face you selected will turn to a new texture.



Now that surface has changed to the new texture. I chose cunieform.

Note: sometimes you have to refresh the 3d view screen to see your change take place. You simply hit the little button called rebuild textured view. It is directly to the left of the build button.

Go ahead and place textures on all the surfaces in the room. Experiment a bit with things to see what looks good. Try putting all the walls the same then try the walls all different. Don't forget to do the wall you can't see in my pictures. This is the wall closest to us. To look at all the walls just click/hold and drag your mouse in the 3d window and you can move around the room.

## How to Manipulate Tiles and Textures for a good look in your video games

Getting good textures and manipulating them into good tiles is a fundamental skill that is very important in game making. If you don't get this right nothing in your game world will look right. You should spend the time needed to learn this skill. It is well worth the effort and will make the difference between drab and cartoonish game levels and sharp good looking levels.

Lets start with some definitions:

What is a texture?

Here are some examples:



Textures have a look, and a feel. You can find textures everywhere. Asphalt roads, blankets, plastics, wallpaper, rugs. They all have distinct textures. And textures are very important in your game. If you wanted to design a wooden structure would you use the top texture? Nope. You would want to use the center one. But you can't just use that texture. You have to change it into a tile that will be poured neatly onto the surface of your wooden structure. That is where Tiles come in.

*Interested in a book about creating art for video games? Take a look at this book. [Here is the Make Video Games Recommended list of books](#)*

### What Are Tiles?

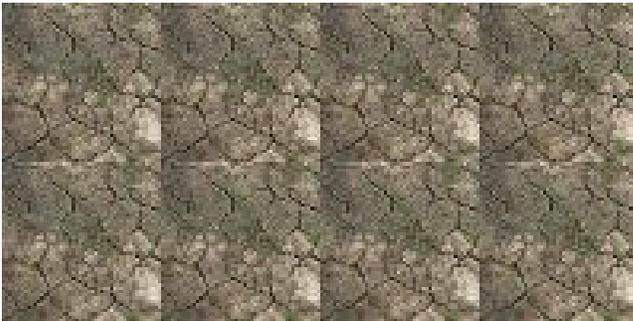
Tiles are textures that have been specially formatted to blend together well in the video game world. You use some kind of image manipulation program to do this.

Here is a tile: For demonstration purposes it is 50 pixels by 50 pixels.



Now what will this tile look like if we repeat it into a bigger surface? Say the wall of a room?

It looks like this:



See how it has a blocky look? You can see where the different tiles meet each other. Clearly eight squares placed side by side. This is no good. Not what you want for your game.

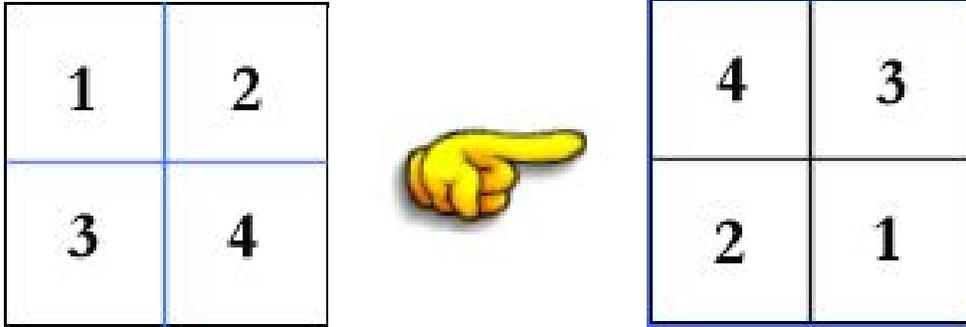
We can fix this however by manipulating the small tile. And when we are done the eight tiles placed together look like this:



There is still a repetition of the pattern, but it no longer looks like eight identical blocks placed side by side. Notice that the dark lines of the stones match up across the tiles.

How do we do this?

We take our original tile and we rearrange it by breaking it up into four pieces. Think of your tile as four different quadrants like this:



So you cut your original tile into four pieces then rearrange the pieces into a new tile. You move the four quarters of your tile into the new positions. Now 1 is in the lower right and 2 is in the lower left etc.

### **Creating the Art of the Game Product Page at Amazon**

What does this do? In the first tile you can see the blue lines going horizontal and vertical. These pass through areas of the tile that blend perfectly. The top of 3 blends perfectly with the bottom of 1. After you rearrange the tile the top of three is at the top of the square right? When this tile is poured in with other tiles guess what happens. The top of three is now directly against the bottom of a 1 and we know this blends perfectly because it was in the middle of the tile before.

One last thing. After you cut your tile into four squares and rearrange it into a new tile you should touch it up a little bit. Make sure the areas in the middle look good. Take out or smudge anything that looks unreal.

Not Always is this quadrant thing necessary.

For example you may have a shape that you want to be repeated and "look" repeated like the tiles in a ceiling. In the image below the floor is well tiled and looks seamless. The ceiling is definitely tiled with the seams and looks good that way.

### **More Important Tips:**

You should always try to make your tiles in pixel squares of 128x128 or 256x256. This is a format that is perfectly suited for the computer and it will handle these graphics very efficiently. If you make your tiles in sizes like 100x100 or some other size the computer has to take a lot longer to figure out how to pour the texture into shapes. This could slow down your game significantly.

Remember: After you rearrange your four quadrants into a new tile, touch it up a little bit. smooth out any harsh lines in the center.

So you want to work with graphics but can't afford Photoshop. You still can have an outstanding graphics program for a lot less money. I use Paint Shop Pro. I do all my design and graphics work in this program and it is absolutely fantastic. I can't recommend it enough. If you want to do some decent graphics and art work you have got to get a good program and Paint Shop Pro is the affordable solution. It does what the expensive ones do and it is easier to understand.

### [Read More about Paint Shop Pro](#)

Without a doubt lighting is one of the most important aspects of your video game design work. Lighting can make or break a game. So you really have to get good at manipulating the lighting. And it is not just about putting up lights, it is also about what level of light and what color of light. Some rooms will be dark, some rooms will be light and some rooms will be moody with colors. Also it is great to have areas of rooms that are darkened. Lighting in the game can be very well done so there are dark areas around columns and structures .

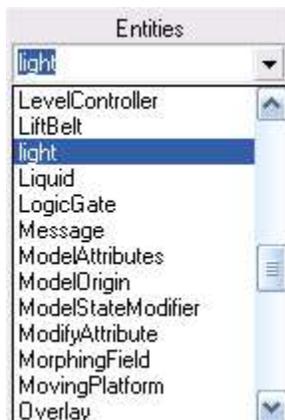
### **[Reality Factory Make a video game tutorial: Lesson 6 -Lighting](#)**

Let's Begin Manipulating Lights

Click on the Template Tab

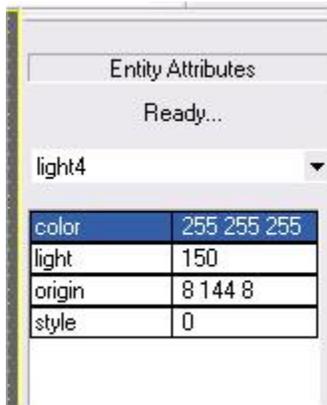


Further down in that menu you see the entities drop down box. Pull it down and select light. A light will pop into your room. Hit Enter to accept it into your room.



Now hit the modify brush button and the Move Brush. Click on your new light in one of the three views and move it approximately to the top of the room and the middle of the ceiling. Think of it as being a chandelier hanging from the middle of the ceiling.

Now look on the right side of the workspace. you see the entities attributes box.



The First row is the lighting color attributes. You can manipulate this to get any color you want. The three digits 255 255 and 255 represent Red, Green and Blue. The 255 is the maximum number for that color. And when all three are set to 255 you get white light. If you wanted to get just a pure red light set these numbers to 255, 0, 0 . You should experiment with these color settings. It will make a big difference in your games.

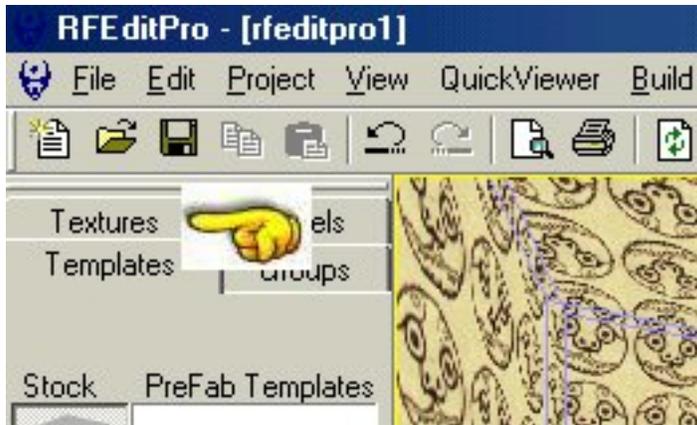
The second row is the light intensity. It is currently set to 150. The higher this number the further out the light will extend. For now set this to 750.

Now let's run this room and take a look at the lighting. That way you can get a feel for what you have done and manipulate it a bit.

Here is a review of how to build this room:

Now let's add a character and actually walk around inside this room. This is where it starts to get fun and see! You can have something up and running in just a matter of minutes.

Hit the tab called templates.



Directly underneath this tab, and a little way down the page is a dropdown box called



Pull down this menu and select **Playersetup** then hit enter on your keyboard.

This causes a small icon to appear in your room. This (**Playersetup**) gives your player all of its characteristics.

Now let's place a player in the game. Reality factory already has pre-made players for you so all you have to do is place it in the game.

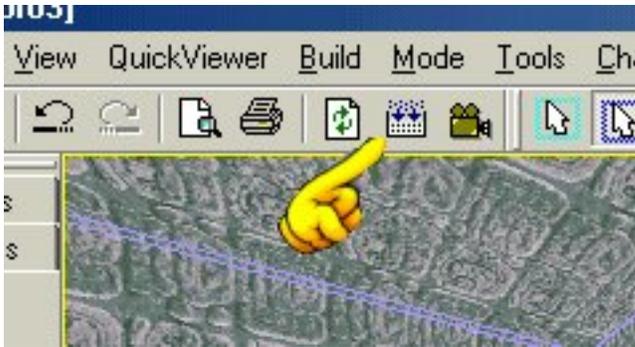
Pull down the **Entities** menu again and select **Playerstart**.

Now click on your 3d view and look around. Near the floor a lightbulb has been placed in the room. This is the player start. Hit Enter again and the lightbulb will change to a little red box that says player start.

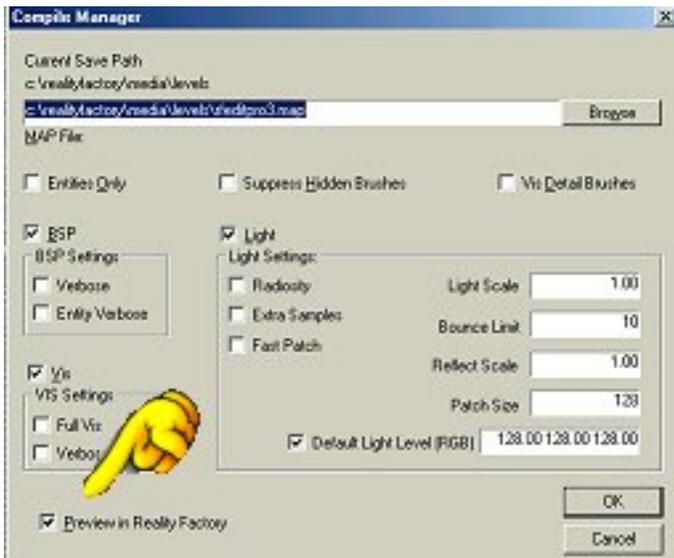
This is your player!

Now lets run this and take a walk around inside.

Click the **Build** button. This takes the world you built and compiles it into an actual game world.



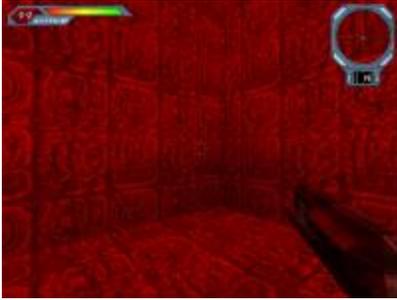
Now a menu appears. Check the box called Preview in Reality Factory.



And Make sure the default Light Level check box is unchecked. This will make the colors you assigned to the level happen.



The world is built, select your character and name and take a look.



Eek ! This room is awfully red! But you can see how the lighting works now. I recommend you go back and tinker some more with the lighting colors and strength. Then look at it to see what you get. Experiment with more than one light too. You can get some really good effects. And we are going to do much more work with lighting for effect.

### **Lesson 7 Adding simple objects to your game**

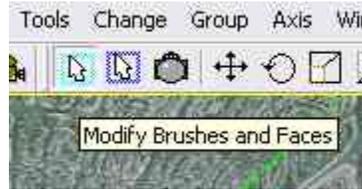
Now we are going to place a couple of objects right into the game. This is a pretty simple process and as you learn the techniques you should experiment with them. Add more things and vary their size. You can't break anything in the game! The worse that could happen is that you just delete the level you made and start a new one.

First Let's Resize our Room - Making a little bigger so we have room to place things inside.



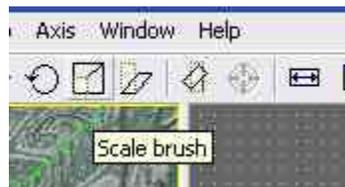
First lets zoom out so we can have more room in the workspace. Hover your mouse over the Top view window and roll your mouse wheel scroll button. This changes the scale of the window. The room you made is still the same size. It is now just smaller inside the window. This way you can add more rooms and make your level more complex.

Now let's make the room larger

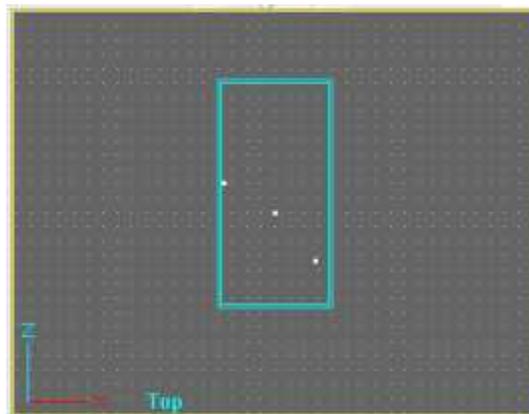


Press the Modify Brushes and Faces button. It's the arrow highlighted in light blue. When you roll over the menu buttons they will tell you what they are..

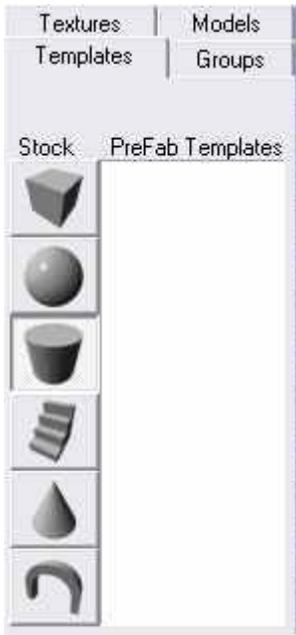
Then Select your room by single left clicking on it.



Now hit the Scale Brush button - Its the button with the funny square shape in it. Now you can click and drag your room to a new size. You can even change the height of the room if you want to(you change height in the side view windows). I made my room longer but kept the width the same; It is now kind of like a corridor



Now let's Add some columns to this room

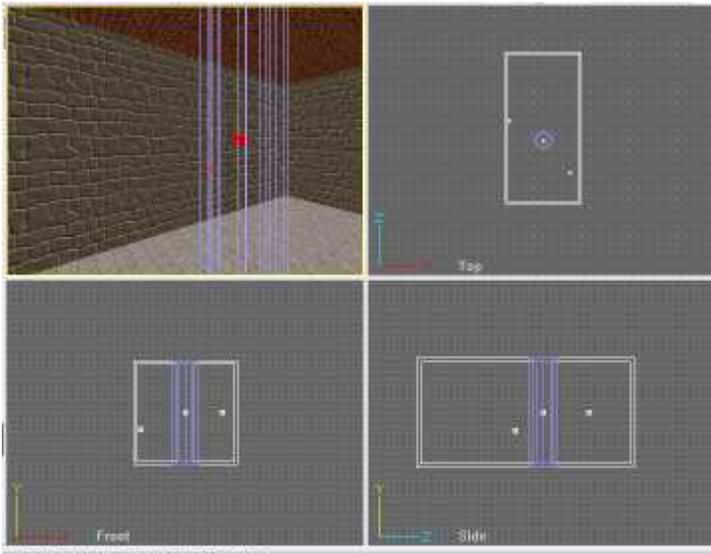


On the left side of the workspace you see four tabs

Textures | Models | Templates | Groups

Click on the Templates tab and you see a bunch of prefabricated items that you can insert right into your game. You might want to tinker around with these and get a feel for the different shapes and sizes.

We are going to insert a column in our game so we press the Cylinder Stock picture and a column shows up in the game!



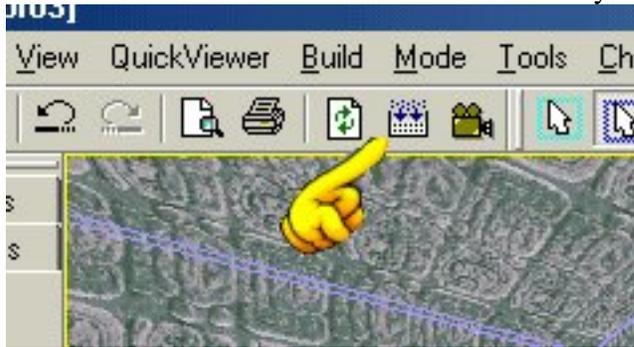
This image shows (above) the column we just created. It is outlined in light purple. That was easy enough! Now you can rescale this and make it shorter, taller, or wider if you want to. You simply follow the same steps you did for the room.

- 1 Select Modify Brush button
- 2 Click on the Column
- 3 Select the Scale Brush Button
- 4 Click hold and drag the column to resize

Now let's finish this up and then run it so we can take a look at what we made.

Hit Enter to accept your column

Click the Textures tab and select a texture for your column and hit enter.



Now we can run it and take a look at what we made. We have already placed our playerstart and playersetup functions so all you have to do is hit the Compile button



Here is your character in the game. Notice the room is larger and you can see the column.

How did I get out of first person view so I can see the character walk around? You do that with the F2 Button. And the F1 Button gets you back in first person view.

Your mouse fires your shotgun and spacebar is jump. the Tab button will hide the radar and health bar at the top of the screen.

Can you see how easy it is going to be to make a game?

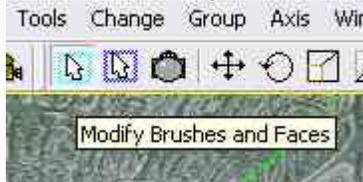
Before you continue with this tutorial on how to make a video game I recommend you go back into the workspace and tinker around with the size of the room, add another column and change the textures. You want to get real good at this. And practice is the only way to do it.

Remember that when you are making a video game you save often and you can't ruin anything. The worse that can happen at this stage of the game is that you start a new level.

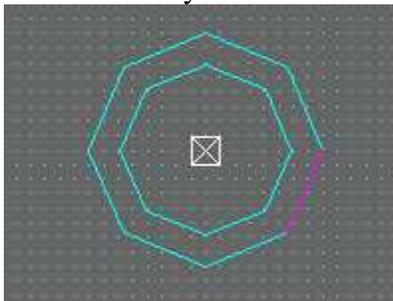
## Lesson 8 Texturing surfaces of objects

Now we are going to apply textures to the different sides of our column. This shows us the skill of focusing down Every object you make and or place in your game is made up of a lot of different surfaces. In the game development software these surfaces are called faces. . You saw that your room has faces like the walls, floor and ceiling. The column also has faces and we are going to apply different textures to these different surfaces.

Click the Modify Brushes and Faces button



Now click on your column and all the edges turn green.



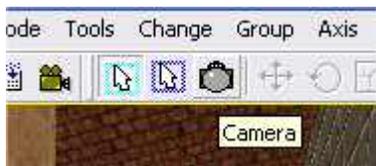
Hit the Pagedown button on your keyboard and all the edges turn pinkish purple in color. Now you can click on any of the surfaces and it will be highlighted then you can apply a texture just to that surface. If you want to quickly move through the surfaces you can use the left and right arrow keys on your keyboard.

After you are done applying the textures you hit the pagedown button and you return out of face edit mode.

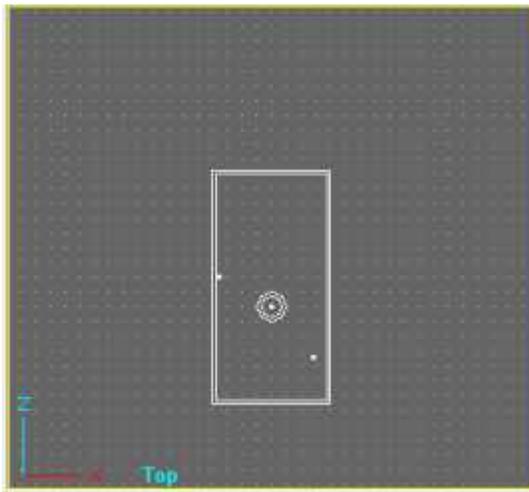
This skill of quickly and easily editing the different faces of any object is very valuable and you should practice this. Add some more objects to your room and edit their faces. Remember that there are tops and bottoms to things too, not just sides. The column you put in your game, and you see in my pictures is actually hollow so there are faces on the inside of the column and the outside of the column. If your player is not going to be able to see the inside of the column you can just leave them as the default texture.

### Lesson 9 Adding a second room and cutting a doorway

Now making a video game is more than just one room. So let's add a second room and then cut a doorway between the two rooms. This is pretty easy and you will get good at it. And of course you can use your new cutting skills to cut away all kinds of stuff like wells, windows and more.



Hit the camera button on the tool bar.



Now in the top view pane right click with the mouse, hold the right click and slowly drag down. You will pan out of the room. Get it to look something like this. You have not

changed the size of the room, you have only zoomed out to see more workspace. Now we have space to add another room to your game.

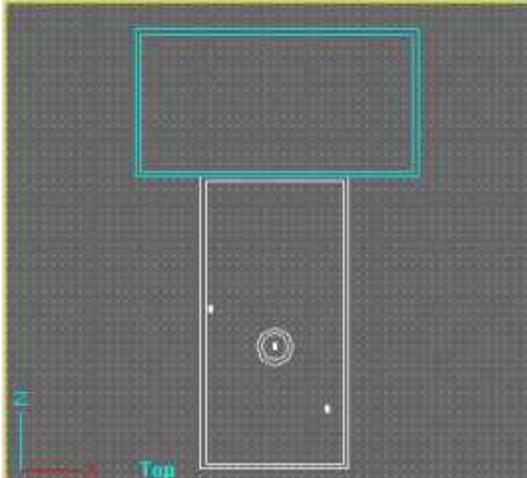


Just like the way you created your first room lets add another room. Select the Templates tab then press the cube stock button. And a blue room appear in your workspace. Move it to a nice location and press the Add current item to world button. That button is located on the left of the workspace below the panel you just used for templates.

Here is what I made. Now I recommend you do this then make the second room larger by using the sizing skills we learned.

Here is a review of how to resize a room:

1. Press the Modify Brushes and Faces button (It's an arrow)
2. Click on the new room so it turns light blue (It is now selected)
3. Click the Scale Brush button - its the funny shaped square. Rolling over the buttons on the menu bar will show a tool tip. Look for the scale brush button.
4. Simply left click and drag the room sides to new proportions.



Here is my room with its new proportions.

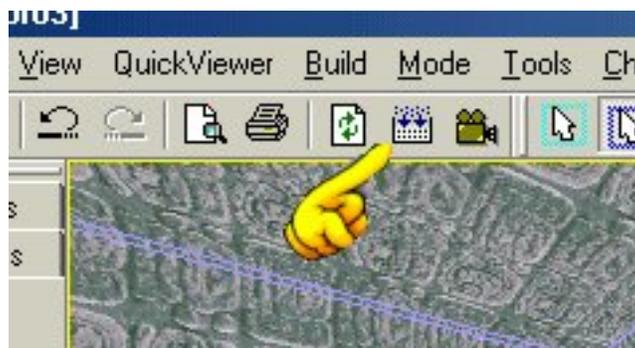
Now we need to discuss something very important in game design: Leakage.

When we cut a doorway between these two rooms what happens if there is a big gap between the rooms? The game engine will have trouble understanding this so what you have to do is make sure your two rooms are right up against each other.

### **Lesson 10 Leakage in your game**

Creating Multiple rooms is not a difficult process but there are some new concepts that you have to grasp and there is one very important rule of thumb you absolutely need to know.

First of all: The rule of thumb you need to know: Avoid Leaks in your game!



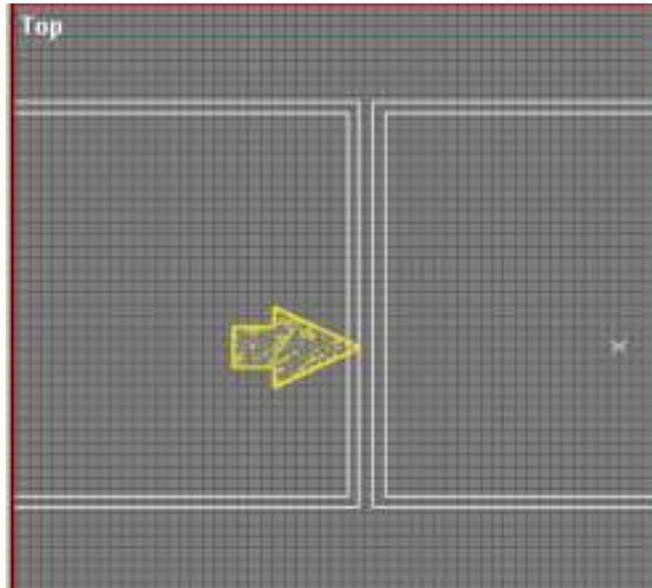
When you run the Compiler it builds your game much as a house would be built. It builds the frames then it applies the textures. Picture this applying the textures as painting. It paints the textures you want onto the surfaces you have designed.

What you have to do when designing your games is make sure there are no unwanted holes in your game. Because what will happen is that the texture (Paint) will pour out.

The compiler will keep pouring paint out in areas that are not displayed in the game. This will slow your game down considerably and may cause it not to run.

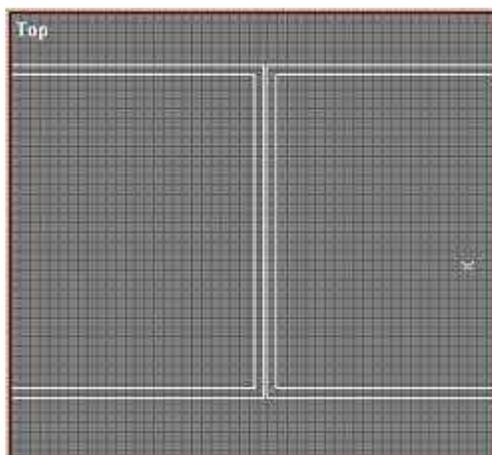
Here is a visual explanation:

Here are two room placed side by side. I will be cutting a hole between them to create a doorway.



See how the two rooms have a little bit of space between them? When I cut a doorway between these rooms there will be a leak where the compiler will try to pour the texture out into the surrounding not visible area of the game.

Always place your structures right up against each other. To get a better look and ensure you have placed them correctly you can use the camera button and zoom in on the grid.



The two rooms are placed right up against each other. This is correct placement. When we cut a doorway between the rooms the textures will not leak out upon Compile. Remember: Leaks can slow your game down considerably - even stop it. So always take care when placing your structures in the game.

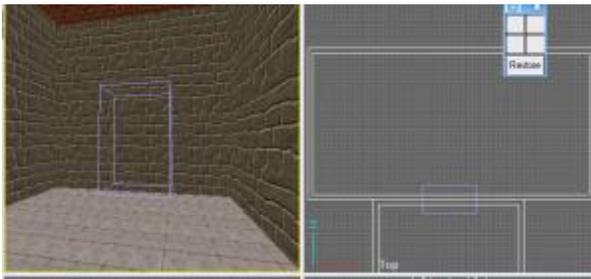
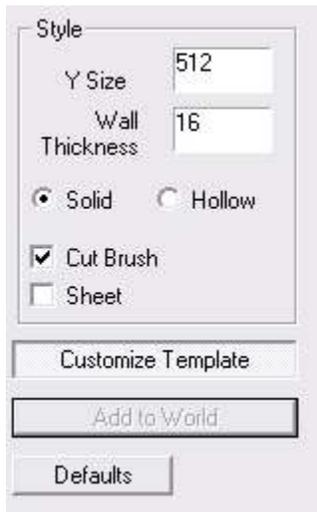
## Lesson 11 Cutting the doorway

Cutting a doorway between your two rooms is an easy process. and you do it almost the same way you placed your rooms in the game. Except you will not place a hollow cube, you will place a solid cube and with the property of being a cut brush. This is what you do.



Hit the template mode button then in your control panel hit the customize template button .

Customize Template



Now check the solid and cut brush options. This will make your new brush a solid object and it will cut away whatever you put it into. Remember that whenever you are switching the types of brushes you are using like hollow or solid you need to hit the customize template button and set these settings. When you add more rooms you will set this back to hollow and uncheck the Cut Brush option.



Now size the new solid cutbrush so it is a nice doorway size. It is shown here as a light purple rectangle. When you have it just the way you want it, all you have to do is hit the enter button on your keyboard and the doorway will be cut!

Now you can see into the other room. Notice how I have set the textures of the floor in both rooms to the same exact texture. This gives the whole structure a nice looking continuity.

Compile your game and take a look around inside!



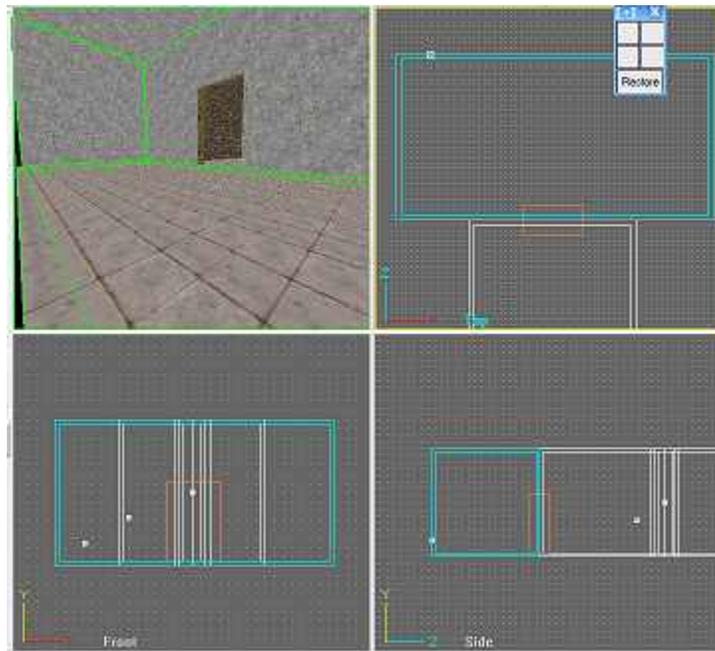
Here is a screenshot of me playing the game. I ran in the other room and looked back through the doorway. You can see the column we made.

## Lesson 12 Adding a Pool of Water

Now we are making some progress and want to add a pool of water to our second room. This will show us some nice additions to a game and also show us some important concepts.

Lets cut a hole in the floor of the room. You do this by following the same process you used to cut a doorway.

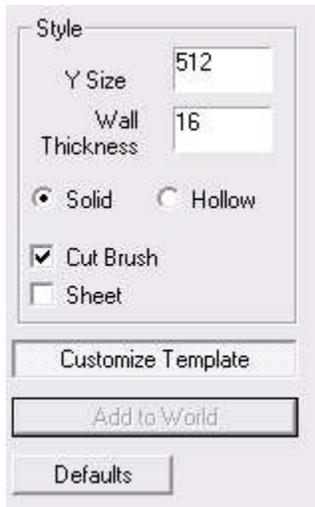
Hit the camera mode button and using right mouse button/drag and left mouse button/drag get the second room in a comfortable view so you can see it all and see the floor of it. This will make it easy to work with.



This image on the left shows me ready to cut out the pool of water. You can see the second room is comfortable placed in the view screens. Make sure you do this in the texture view too. And as always don't hesitate to hit the refresh button to refresh the textures etc.

### Now let's cut a hole in the floor of this room

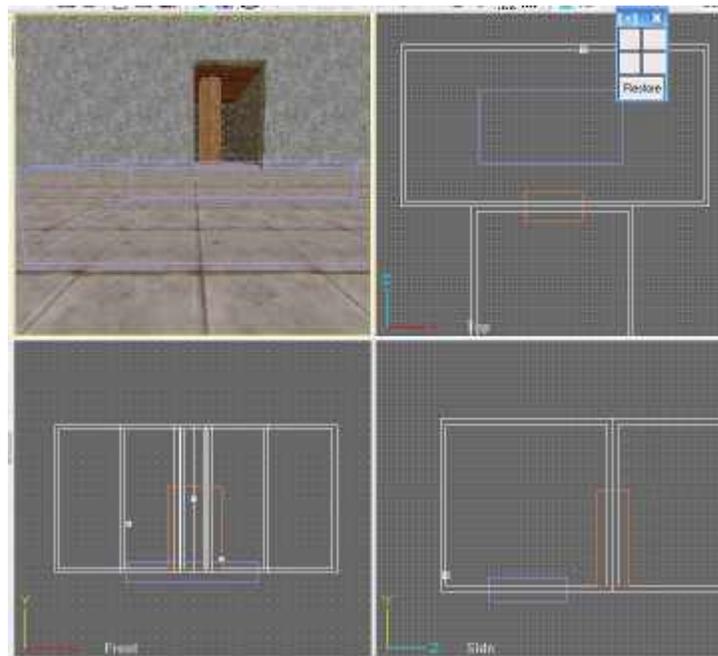
Hit the template mode button then in your control panel hit the customize template button



Now check the solid and cut brush options. This will make your new brush a solid object and it will cut away whatever you put it into. Remember that whenever you are switching the types of brushes you are using like hollow or solid you need to hit the customize template button and set these settings. When you add more rooms you will set this back to hollow and uncheck the Cut Brush option.

*Note: If you have come to this page straight from the last page where you cut a doorway then you won't have to reset these settings. It is already set at cutbrush and solid.*

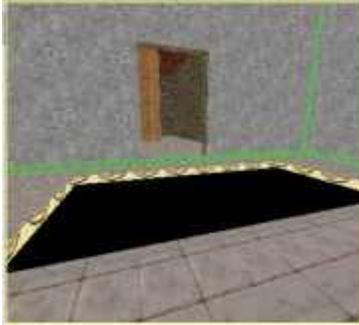
Now using the Move Brush and the Scale Brush buttons move the new square so it will cut a hole right out of the floor of the second room.



This image shows my cut brush in place. When I hit enter it will take effect and cut a hole right in the floor!

So do that; you have placed the cut brush then hit enter and now you have a big gaping hole in the floor of you game eek!

This is ok. We will fill it and while you are doing this remember that you always want to avoid leaks in your game.



Here is what my game looks like with the hole cut in the floor.

Now let's put a base under this hole in the floor.

Hit the template mode button, then the customize button then make sure solid is checked and cut brush is unchecked. We are placing a solid block in here. Note: If you don't see the customize button, or anything in the right side of your workspace just hit the Templates tab and the square stock button on the left side of the workspace.

If you are having trouble creating a new brush that is solid because the last brush you made was a cut brush then follow this process.

Press the Modify Templates button

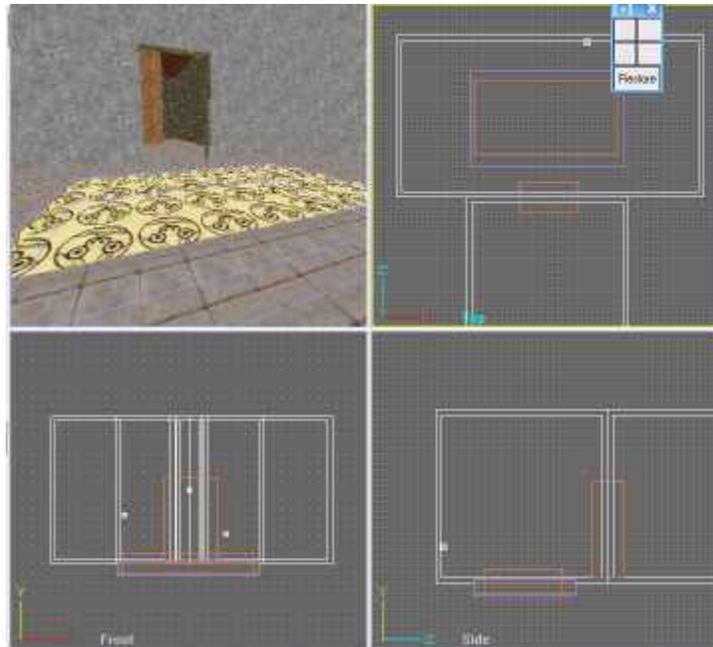
Pull down the Tools Menu and select New Brush | Cube

This will bring up the Customize Menu where you can select Customize Template and select the Solid check box. Now you can size the solid brush you made the hit Enter or the add to world button.

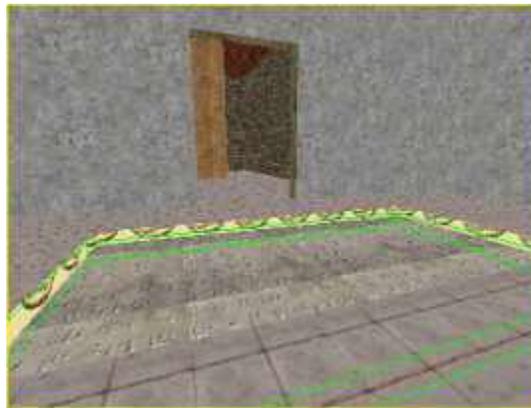
Now use the move brush and the scale brush buttons to move the new block right under the hole in the floor.

### **Lesson 13 Adding a Pool of Water continued**

Now we have created a new block that is solid and we have placed it directly underneath the hole in the floor. Remember to be very careful with this and place the new block right up against bottom of the floor so you have no leaks in your level. Here is an image of my block and what it looks like. Your new block should look very similar. It is just big enough to cover the entire hole.



And notice that the new block has the default smiley face texture in it. You want to change this texture of course. I have changed mine to a different texture so you can see it easily in this tutorial but you might want to make it the same texture as the floor in the game. This will look kind of nice. It will be like a swimming pool And we are going to make the water semi transparent so you will see the bottom.



Which is what I have done here. You can see the bottom of the pool has a new texture. But what about the smiley faces around the edge we can still see.

These edges were created by your cut brush. To apply a new texture to these surfaces you select your cut brush and apply a texture to it. Your cutbrush will look orange and even though it is actually a hole it is still a cube that is there. So you can select it and apply textures.

## Lesson 14 Adding a Pool of Water continued

There are a lot of ways to add water to a level and for now I am going to show you a really basic way to do it. Later in this tutorial we will cover making water that looks great but that involves importing graphics into your game. And that is the next section of this tutorial so just hang tight, you are making progress!

### About Water in your Game

There are two different concepts to know when it comes to adding water to your game. Do you want the water to be passable? In other words will the characters be able to walk and swim in it? Also, do you want it to be transparent? Can the player look into the water and see the bottom of the pool?

The answer for us is yes to both these questions and this will be the typical answer in most of the water you make. But just be aware that you can make the water, or other liquid such as lava, not passable so it can be walked on. And you can make it opaque so it can't be seen through. This idea of transparent is also important if you want to put fish tanks in your game.

Let's add a new brush to the game. This is will be the cube of water in the pool.



Hit the Modify templates button

Now in the left side of your screen in the control panel you see a pull down menu under the heading of entities.. Pull down this menu and select "liquid"

And a new cube shows up in the game. If a cube doesn't show up then just pull down the Tools menu and select New Brush | Cube

Now in the right side tools panel make sure you select Customize Template and check the button for solid. This should be already set to solid because the last brush you made was solid but check it for practice.

Now, Using the Scale Brush and the Move Brush get this new square to fit nicely into the pool. Remember to use the camera button to move your views around so it is comfortable to work on. You can zoom in and work on it so it is nice and easy.

Now that you have it the right shape hit the textures tab on the left, find the dark texture for a muddy water look hit the select button then on the right side of the screen hit the "Add to World" button and your new water block appears.

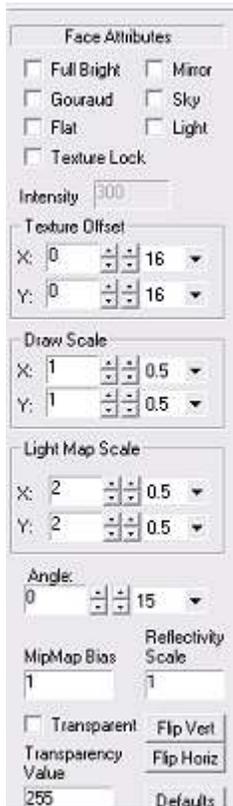
### Make the Water Passable

It has some of the characteristics of water but we are not done. Single click on your new block and its edges turn green and the Brush attributes box comes up on the right of the screen.

Make these selections: Under Type check the Empty button

and under contents check the Water box

This makes your new block of water passable so when the player steps on it he actually goes into the water.



### **Make The Water Semi Transparent**

Now we want to be able to see through the water so with the block still selected and its edges green press the pagedown button on your keyboard. This scopes down to work on the faces of the box and it brings up the face attributes menu.

You can see that menu here.

Now at the bottom of this menu there is a checkbox for Transparent. Make sure you put a check in this box.

And under it the transparency value is set to a default of 255.

Delete this 255 and input 175. The lower this number the more transparent the water so you may want to experiment with this number by previewing the game and then setting it.

This properties box doesn't have an ok button. You just have to input the number - simple as that - and the program knows what you have done.



Here is my pool of water that is pretty dark but you can see the square pattern of tiles at the bottom vaguely through the transparent water.

When you play the game you will step right into the water and you will hear the swishing of water as you walk.

One other nice thing about water is that if you pagedown again so you are back to normal block editing and you see the Brush attributes Panel you can select the checkbox called "Wavy" and the water in the game will ripple. This is a pretty cool effect. I recommend you give it a try to see what it looks like.

In the Brush Attributes control panel after changing to Wavy don't forget to hit the apply button.

OK, that's enough about water for now. You have a general concept of how to do it and we will do more water effects and will do some water that looks great.

For now you can do basic water, make it passable, make it semi-transparent and even make it wavy! So you have actually come a long way.

For the next part of this tutorial we are going to learn how to import your own graphics right into the game so you can use them. This way you can find nice graphics and images

or even make your own then use them in your game. To do this chapter we are going to get a nice blue water color imported into the game so we can do better water.

## Lesson 15: About Graphics in your game

Up until now you have been making your video game by using the pre-packaged textures and models that come with the game. But, this is your video game and you want to make your own graphics and put them in the game.

Before we begin we need to talk a little bit about images in video games and what they are. There are two basic types of graphic resources for video games Textures and Models. To understand the difference between these two resources all you have to do is think about some of the games you have already made. When you created a room you painted it with different types of textures like stone, dirt, or colors. These are textures. When you were done with the room you then placed a DeathMatchStart entity in the game. This was a character that you could play - a model.

The Big Difference between Textures and Models?

Textures don't move and models are animated! Even doors that open are models, so remember this!

There are things we can do within the Genesis game design software for both textures and for models and we will go over these things. But there are things that we will want to do outside of the game software, and we will go over these too. Don't worry about any of this for now. I will take you through the whole process. FOr now here is a breakdown of how we will approach this.

Make a Video Game Tutorial Part 4: Textures, What they are, how to get them, how to make them and get them in your game.

One of the most important things about making a video game is getting the look right. Up until now we have focused on the design of levels, rooms and things. And this is very important but everything has been pretty generic up until now and we have been using the games premade graphics to get the look of our game. But you want to take it to the next level of game making and focus on the look of the world you are building.

Textures is the way you do this. You can find and make unique textures to give your game its unique look. Let's first take a look at what textures are?

What are Textures?

In the real world you think of textures as having a feel; like rough or smooth or bumpy or whatever. With a video game of course there is no feel. You only see a video game, you don't feel it. Yet there are textures everywhere in video game. And when we say

"textures" we me patterns. It is these pattterns that give us the different looks that we want to achieve in our game. Here are some examples of textures:



The first one is stone, the second one is wood, and the third is grass. There are textures for games for everything in the world! And you can either find them or make them yourself. You can even create your own textures that don't exist in the real world. You are only limited by your imagination.

But Textures aren't just textures!

Now that sounds kind of confusing but I don't want you to be limited in your thinking! What I mean is that textures can also be recognizable images not just patterns like wood, grass, rock etc. Here is an example of a created texture:



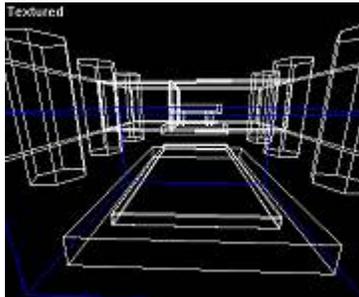
This is a texture of a high-tech vault door. It is a simple image that you can apply to the surface of a model door your created.

If you make an underground bomb shelter like level you could make many of the doors the same size then simply apply this texture to all of them and they would all have the same look. This would give a nice uniformity to your level. This texture as an image extends into lots of things. You could make textures that are paintings to hang on the wall, windows, and all sorts of things. Just don't be limited in your approach to textures.

## Lesson 16: Textures, What they are, how to get them, how to make them and get them in your game.

Of course you can get just about any kind of graphic, texture or image and put it in your video game but.... there are some rules to follow and some things you need to think about.

First off: When a video game is compiled (this is the process of taking the game you built in the design process and actually building it into a world) the compiler looks at a surface in the game then looks to see what texture you want applied to it.



Here is our temple room. Look at all the different shapes and sizes of the surfaces of the columns, walls, and floors.

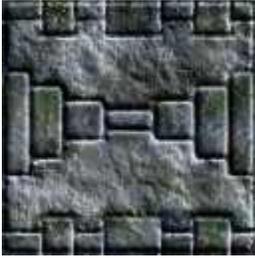
You didn't create a new texture in the exact right size for each of these surfaces did you? Nope! You simply picked a texture for each surface and the compiler poured it right into the surface. And when using the premade textures that come with the Game making software everything looks good.

You chose a texture like this and the compiler poured it right into the surface that you defined.

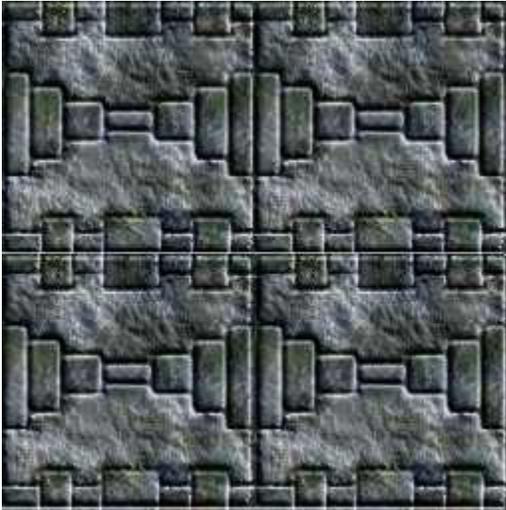


What the compiler did was just repeat this same texture over and over again on the whole surface until it was full. So if you choose this texture for a large wall this texture is just repeated as much as necessary. This is a great thing because it saves you a lot of work. You don't have to insert the textures over and over again.

But When Creating your own Textures for a game this could be a problem



If you have a tile like this, that is symmetrical it will look fine when poured together with other tiles.



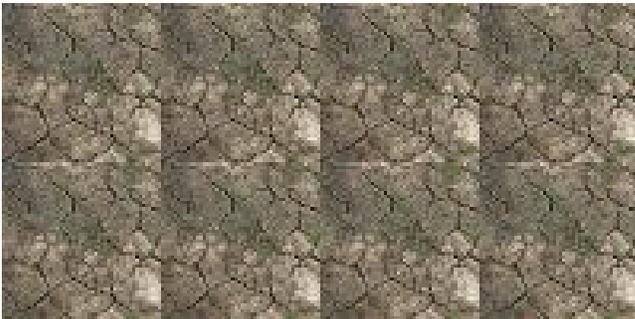
Like this! This has a good look,. The tiles are poured together and there is a continuity to them.

But What if you have a texture that isn't this symmetrical? Something like this:



Now what will this tile look like if we repeat it into a bigger surface? Say the wall of a room?

It looks like this:



See how it has a blocky look? You can see where the different tiles meet each other. Clearly eight squares placed side by side. This is no good and not what you want for your game. This doesn't look like a very realistic wall or floor.

We can fix this however by manipulating the small tile. And when we are done the eight tiles placed together look like this:



There is still a repetition of the pattern, but it no longer looks like eight identical blocks placed side by side. Notice that the dark lines of the stones match up across the tiles.

So when you are making your own textures you have to make them so they pour correctly together to get a uniform look.. This is very important and not that hard to do. I will show you how.

Of course if you are creating a texture tile that you are going to use like a door, a picture, a window or some other kind of picture like image you don't have to worry about this. But any texture you want to use as a poured texture into surfaces of your game you absolutely have to make sure they pour correctly together and this takes a bit of manipulation.

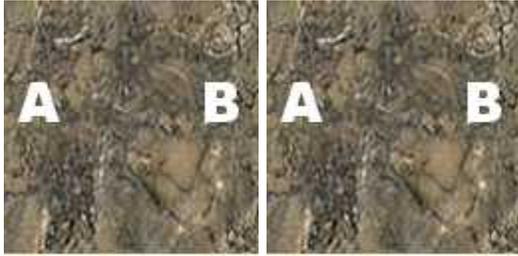
## **Lesson 17: Textures, Continued**

Let's Make a seamless tile.

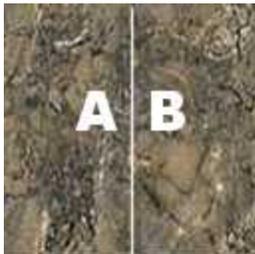
To get the outer edges of tiles to blend perfectly together what we do is break open the tile and make the inside into the outside! Let me explain. Let's look at a larger tile so we can see clearly



Here is a tile. If we repeat this tile side by side the b side of this tile would be lined up with the a side of the next tile and it wouldn't look right



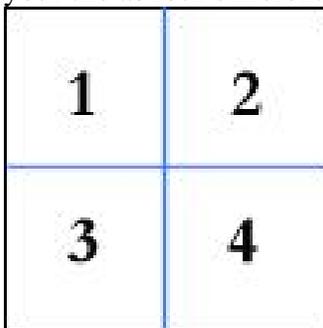
But what about the middle of this tile? Does it match up well? It sure does. Let's draw a line through this tile to show what I mean.



If we draw a line down the middle of this tile and look at it as two separate tiles do the A side and the B side line up? Of course they do! So this is trick we use to get tiles to line up. We break the tile up into four quadrants and move the inside lines to the outside.

How do we do this?

We take our original tile and we rearrange it by breaking it up into four pieces. Think of your tile as four different quadrants like this.



Then we rearrange these four quadrants into new positions.



So you cut your original tile into four pieces then rearrange the pieces into a new tile. You move the four quarters of your tile into the new positions. Now 1 is in the lower right and 2 is in the lower left etc.

In the first quadrant picture you can see the center lines are blue. When we split up this texture and rearrange it the blue sides are now on the outside as shown in the second quadrant picture. Now we know all the outside edges will line up with neighboring textures.

This is a technique that you will get really good at, particularly if you use a digital camera to take your own pictures of textures.

After you have split and rearranged a tile like this you are probably not quite done with it. You should take a close look at it and put four of them together to see how they look. You will probably want to touch it up just a bit with your paint program so it looks better. There may be a little bit of tweaking or smudging you might want to do to get it to look its best.

### **What about a Paint Program?**

This is a good question. The ultimate package for manipulating images like this is Photoshop. It even has a specialty tool that will quadrant a texture for you automatically which is real handy. But Photoshop is quite expensive.

I recommend you try to find some free paint and graphics manipulation software that will satisfy your needs as a beginner. If you are considering purchasing some kind of software I highly recommend you get Paint Shop Pro. It is the program that I have purchased and I use every day. It does most of what Photoshop does but at about one tenth the price.

[Corel Paint Shop Pro Photo XI](#)

If you are very serious about making video games and in particular working on graphics for video games then getting photoshop would be the ultimate choice for you because it is what they use in the game making industry. You will know how to use the software which is a big plus! But, if you can't afford Photoshop you might want to consider Paint Shop Pro. It is very much like Photoshop and you will still learn the skills of graphic design and graphic manipulation and the skills will be easily transferred to Photoshop once you get it.

Adobe Photoshop CS2 Photoshop is the industry standard. It is a high-powered and professional program that can do practically anything. This is what the real video game makers use to work with graphics. But this is not for fun, you will need to commit time to learning how to use this powerful program. [Photoshop](#)

Now this isn't all you need to know about making graphics, there is still one more important thing to know. I will show you what this is then we will learn how to import the graphics right into your game software so you can use it when making a game.

### **Lesson 18 Textures continued**

There is one more thing you need to know about making textures for your video game. You know that your computer is simply a number crunching system. It is very good at counting out numbers and when it is pouring textures into the surfaces of your game it is counting out numbers. But not all numbers are as easy to crunch as others.

In order to make the compilation and then the running of your game faster the compiler and computer use groups of numbers. This grouping makes things run so very much faster and in order for you to take advantage of this grouping of numbers you should create your tiles and textures in very uniform sizes.

Whenever you make your tiles you should make them in multiples of either 64, 128, or 256 pixels in size. In any event your textures really should be in a power of two so either 2 pixels, 4, 8, 16, 32, 64, 128 or 256.



If you don't quite understand this I mean that your texture should always be a square and that each side of the square should be one of those pixel lengths with the most common being 128 or 256.

This is a very important concept and you should always follow this rule of power of two. It is possible that if you didn't follow this and you built large games that it could slow down considerably. It could get so slow that it wouldn't be playable.

One More Note about Optimizing your Graphics for Video Game Making

**This is important - Write this down!!!**

If you are going to be making your own graphics to import into the Genesis 3D Game engine, which is what this tutorial is all about, then the images have to be:

in .BMP format and in 256 Indexed Color Mode

The .BMP format is bitmapped format and a standard way of saving an image, the 256 Indexed color mode is also a standard mode for images. These two settings are easily done when you are saving your image in either Photoshop or Paint Shop Pro.

You simply save your images as .bmp and in 256 color mode. In Paint Shop Pro there is a menu option under Colors that is called Decrease color depth. You simply select this to 256 colors. And to save as a .bmp you simply select .bmp when you save it.

### **Lesson 19 Importing your graphics**

Okay, so you have made a texture, or several textures and you want to import them into the game. How do you do that? It is really quite easy. But just as a review I want to make sure that you have made the textures in a way that they will pour nicely, they are square in a power of two like 128x128 or 256x256 pixels, and you have saved them in .bmp format and 256 color mode.

If all this is done you are ready to get them in your game.

Reality Factory needs its graphics to be in a format called .txl or (texel) and it has a program that will take your bitmap files and turn them into .txl files. This program is called RF Texture Packer and it is really easy to use.

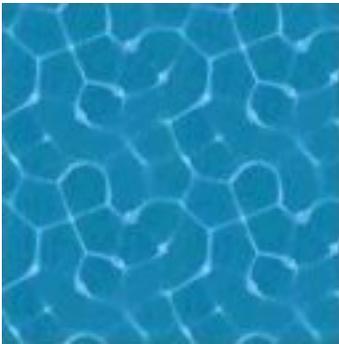
### **USING RFTexturePacker TO IMPORT GRAPHICS INTO YOUR GAME**

RF Texture Packer is a program that converts .bmp files to .txl files (which are what your level needs to draw graphics in the game)

Using Windows Explorer you need to hit the start button to bring up your program files. Then select Reality Factory | tools | RF Texture Packer and the Texture packer program comes right up. It looks like this:



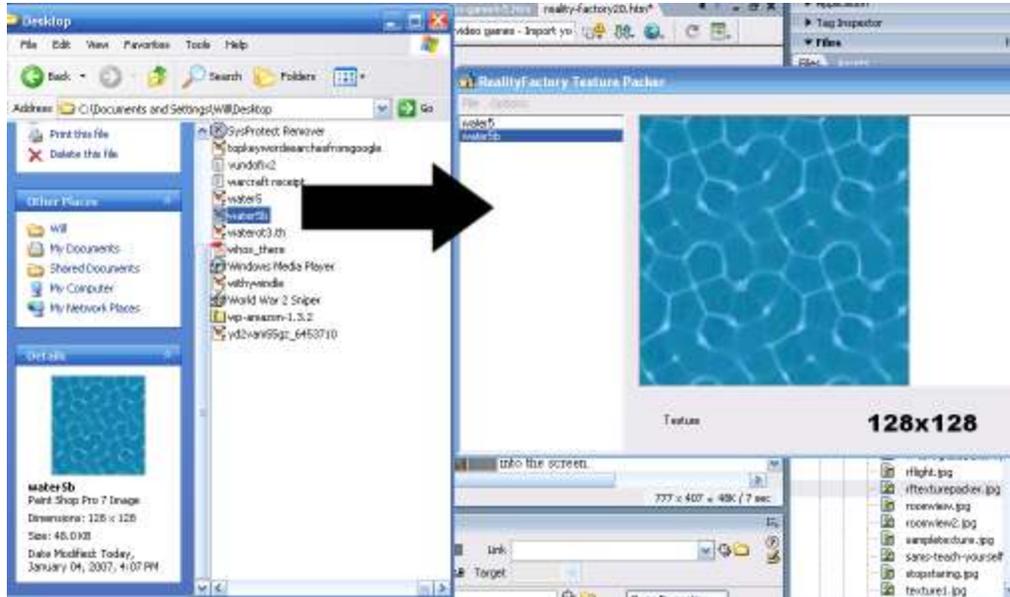
Now here is an image for you to get into your Reality Factory Game:



Right click on this water image and select "Save Picture As" then pull down the menu and select .bmp not jpeg.

Save this image onto your computer somewhere you can easily find it like on your desktop.

Now using windows Explorer navigate to where you saved this picture and drag it  
Find the file you want to move into RF Texture packer and single click then drag and  
drop it right over to the Texture Packer window.



This image shows texture pack and windows explorer both open on my desktop. I simply dragged the file over to the texturepack application and dropped it right in. When I then clicked on the file in tpack the picture popped right into the screen.

With this image highlighted in TPack save it into a reality factory folder. I suggest that you create a new folder called myimages and save it there. This will make it easy to find. Make sure you give the file a name and that it is saved as a .txl file.

Okay so now the graphic or graphics you made are saved into the Reality Factory game. There is only one more thing you have to do in order to actually use them in your game.

Now How do You work with this new Picture in your Game?

This is real easy. All you do is when you have the game design desktop open you simply pull down the Options Menu then select Level Options; Hit the browse button then go to your myimages folder and inside it click on the graphic you imported. And Voila you are done. Your graphic is now inside the texture screen and you can use it normally.

But wait a minute! There is only one graphic and you want to import lots of them. This is easy, when you are using tpack you simply drag over as many images as you want. When you save it to the myimages folder all of these images are packed up into a single file. Then when you choose Options | level options and browse to that file all of the images pop into the Reality Factory game editor.

## 20 Okay, Let's Review

In these chapters of the tutorial you learned about graphics, textures and tiles. How to create them so they are seamless and what sizes and formats you should save them in.

You have made some textures yourself and imported them right into your game. WOW. You might not realize it but you have accomplished quite a lot. This is a very important step in making video games.

The next Big step in your learning how to make video game is to make and import models. But that is some serious stuff. You should review all of this texture part of the tutorial and make some more textures, import them and see how they look. Get good at this then we will do the next part of the tutorial (Models)

For More information about Models in your games go to the [make-video-games.com](http://www.make-video-games.com) website. As of the completion of this tutorial there is a modeling tutorial in progress.

### More Resources:

[The Make Video Games Website](http://www.make-video-games.com) – Lots of information about making video games in different formats. This ebook is a product of this website.

(<http://www.make-video-games.com>)

[rentgamescheap.com](http://rentgamescheap.com) – Where you can rent video games before you buy them.

For More interesting projects about many different subjects visit the authors website at: <http://www.stormthecastle.com>

[The Genesis 3d Website](http://www.genesis3d.com) – This is the engine that Reality Factory is based on  
(<http://www.genesis3d.com>)

[The Reality Factory Website](http://www.realityfactory.com) – This is the software you are using for this tutorial

## Genesis 3d Copyright

GENESIS3D PUBLIC LICENSE

Version 1.01

### PREAMBLE

Please read this license carefully before using Genesis3D. By downloading and using Genesis3D, you are agreeing to be bound by the terms of this license. If you do not or cannot agree to the terms of this License, do not download or use Genesis3D.

This license is intended to make the source code of a state-of-the-art 3D engine available to everyone who wants the opportunity to use and contribute to a 3D engine. Genesis3D is a commercial product. It is not in the public domain or "freeware" or "shareware". By making the source code available we are not giving it away or abandoning our right of ownership. By using Genesis3D you agree to be bound by the spirit and the terms of this license.

This license permits you to use the Genesis3D SDK to create and sell products. You may also redistribute the Genesis3D SDK. You may also make modifications to the SDK and distribute them too.

If you modify the Genesis3D engine source, except any portion of it, or if you use a modified version of the engine source, you must make your source code available for others to use under the same terms that we have granted you. This guarantees that modifications to the engine will be available to everyone and the best modifications can become part of the official Genesis3D release. Your source code must be released concurrent with releasing any executable built with Genesis3D. If you do not wish to make your source code available you must obtain a separate license from Eclipse.

Gtest, the sample game, is specifically excluded from this requirement. Feel free to modify Gtest and use it in any way that benefits you.

If you use the original unmodified Eclipse-built binaries there is no requirement for you to make your source code available.

You must display the original unmodified Genesis3D animated logo as the first logo on startup of your product, demo or application. You must also prominently display the Genesis3D logo on any marketing materials, advertising or packaging of your product.

These terms are good for the community because it makes the source of a high-end 3D engine available to everyone and it encourages sharing of

improvements. They are good for Eclipse because some companies will opt to not share their source and will prefer to pay to license the engine instead. These kinds of licenses will fund future Genesis3D development and subsidize the free licenses that many people will enjoy.

## 1. DEFINITIONS

1.1. ``Contributor'' means each entity that creates or contributes to the creation of Modifications.

1.2. ``Contributor Version'' means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. ``Covered Code'' means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof, and any Source Code which duplicates, enhances, modifies or replaces functionality of the Covered Code as defined in Exhibit B.

1.4. ``Electronic Distribution Mechanism'' means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. ``Eclipse'' means Eclipse Entertainment, L.L.C.

1.6. ``Executable'' means Covered Code in any form other than Source Code.

1.7. ``Larger Work'' means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. ``License'' means this document.

1.9. ``Modifications'' means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

- A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.
- B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. ``Original Code'' means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.11. ``Source Code'' means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or a list of source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The

Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. ``You'' means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 7.1. For legal entities, ``You'' includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, ``control'' means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

## 2. SOURCE CODE LICENSE

### 2.1. The Eclipse Grant.

Eclipse hereby grants You a world-wide, fully paid, royalty-free, non-exclusive license, subject to third party intellectual property claims: (a) to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, or as part of a Larger Work; and (b) under patents now or hereafter owned or controlled by Eclipse, to make, have made, use and sell (``Utilize'') the Original Code (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Original Code (or portions thereof) and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

### 2.2. Contributor Grant.

Each Contributor hereby grants You a world-wide, fully paid, royalty-free, non-exclusive license, subject to third party intellectual property claims: (a) to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code or as part of a Larger Work; and (b) under patents now or hereafter owned or controlled by Contributor, to Utilize the Contributor Version (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Contributor Version (or portions thereof), and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

### 2.3. Your Grant.

You hereby grant to Eclipse a world-wide, fully paid, royalty-free, non-exclusive license, subject to third party intellectual property claims: (a) to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by You (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code or as part of a Larger Work; and (b) under patents now or hereafter owned or controlled by You, to Utilize the Modifications (or portions thereof), but solely to

the extent that any such patent is reasonably necessary to enable Eclipse to Utilize the Modifications (or portions thereof), and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

### 3. DISTRIBUTION OBLIGATIONS

#### 3.1. Application of License to You.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 7.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.6.

#### 3.2. Application of License to Eclipse.

##### (a) Other Products.

Eclipse may include Covered Code in products other than Genesis3D without such additional products becoming subject to the terms of this License, and may license such additional products on different terms from those contained in this License.

##### (b) Other Licenses.

Eclipse may license the Source Code including Modifications incorporated therein, without such additional products becoming subject to the terms of this License, and may license such additional products on different terms from those contained in this License.

##### (c) Other Releases.

Eclipse may create new releases of Genesis3D without such releases becoming subject to the terms of this License.

#### 3.3. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party. Furthermore, the Source Code to the Modifications You create or contribute must be delivered in fully compilable form using commonly available tools. If the tools used to compile the Source Code are not commonly available You must also include the tools, in Source Code and Executable forms, as part of the Modifications. Your Source Code must be delivered unencrypted, uncloned, unobscured and in human readable form.

#### 3.4. Description of Modifications.

You must cause all Covered Code to which you contribute to contain

a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by Eclipse in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

### 3.5. Intellectual Property Matters

#### (a) Third Party Claims.

If You have knowledge that a party claims an intellectual property right in particular functionality or code (or its utilization under this License), you must include a text file with the source code distribution titled ``LEGAL'' which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If you obtain such knowledge after You make Your Modification available as described in Section 3.3, You shall promptly modify the LEGAL file in all copies You make available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

#### (b) Contributor APIs.

If Your Modification is an application programming interface and You own or control patents which are reasonably necessary to implement that API, you must also include this information in the LEGAL file.

### 3.6. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code, and this License in any documentation for the Source Code, where You describe recipients' rights relating to Covered Code. If You created one or more Modification(s), You may add your name as a Contributor to the notice described in Exhibit A. If it is not possible to put such notice in a particular Source Code file due to its structure, then you must include such notice in a location (such as a relevant directory file) where a user would be likely to look for such a

notice. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of Eclipse or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify Eclipse and every Contributor for any liability incurred by Eclipse or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

### 3.7. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.6 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.3. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights

relating to the Covered Code. You may distribute the Executable version of Covered Code under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by Eclipse or any Contributor. You hereby agree to indemnify Eclipse and every Contributor for any liability incurred by Eclipse or such Contributor as a result of any such terms You offer.

### 3.8. Larger Works.

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

## 4. RESTRICTIONS

### 4.1. You agree to:

- (a) display the original, unmodified Genesis3D animated logo as the first logo on startup of your product, demo or application;
- (b) prominently display the Genesis3D logo on any marketing materials, advertising or packaging of your product, demo or application;
- (c) distribute the Covered Code to third parties who agree to be bound by these terms and conditions;
- (d) make reasonable efforts to discontinue distribution of the Covered Code upon Eclipse's release of an update, upgrade or new version of the Covered Code and to make reasonable efforts to distribute such updates, upgrades or new versions to your customers who have received the Covered Code herein;
- (e) be solely responsible for any update or support obligation or other liability which may arise from your distribution of the Covered Code.

### 4.2. You may not:

- (a) remove, alter, obscure, or modify in any way the appearance or operation of the Genesis3D logo;
- (b) copy the Covered Code, in whole or in part, except as provided in this License;
- (c) make any statement that your product is "certified," or that its performance is guaranteed, by Eclipse.

## 5. INABILITY TO COMPLY DUE TO STATUTE OR REGULATION

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.5 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be

sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 6. APPLICATION OF THIS LICENSE

This License applies to code to which Eclipse has attached the notice in Exhibit A, and to related Covered Code.

## 7. VERSIONS OF THE LICENSE

### 7.1. New Versions.

Eclipse may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

### 7.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Eclipse. No one other than Eclipse has the right to modify the terms applicable to Covered Code created under this License.

## 8. DISCLAIMER OF WARRANTY

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN ``AS IS'' BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT ECLIPSE OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

## 9. TERMINATION

This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

## 10. LIMITATION OF LIABILITY

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL ECLIPSE, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED

CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THAT EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

INDEMNIFICATION: YOU SHALL INDEMNIFY, HOLD HARMLESS, AND DEFEND ECLIPSE AND IT SUPPLIERS FROM AND AGAINST ANY CLAIMS OR LAWSUITS, INCLUDING ATTORNEY'S FEES, THAT ARISE OR RESULT FROM YOUR USE OR DISTRIBUTION OF ANY PRODUCT INCORPORATING THE COVERED CODE IN WHOLE OR IN PART.

#### 11. U.S. GOVERNMENT END USERS

The Covered Code is a ``commercial item,'' as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of ``commercial computer software'' and ``commercial computer software documentation,'' as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

#### 12. MISCELLANEOUS

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License is governed by the laws of the State of Texas and the United States, including patent and copyright laws. Any claim arising out of this License will be brought in Travis County, Texas.

The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

#### 13. RESPONSIBILITY FOR CLAIMS

Except in cases where another Contributor has failed to comply with Section 3.5, You are responsible for damages arising, directly or indirectly, out of Your utilization of rights under this License, based on the number of copies of Covered Code you made available, the

revenues you received from utilizing such rights, and other relevant factors. You agree to work with affected parties to distribute responsibility on an equitable basis.

EXHIBIT A.

``The contents of this file are subject to the Genesis3D Public License Version 1.01 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.genesis3d.com>  
Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.  
The Original Code is Genesis3D, released March 25, 1999.  
Copyright (C) 1996-1999 Eclipse Entertainment, L.L.C. All Rights Reserved.  
Contributor(s): \_\_\_\_\_.''

EXHIBIT B.

Functionality of the Covered Code includes:

- (a) Representation and rendering of 2D and 3D images, objects, characters and scenes;
- (b) Collision detection;
- (c) Artificial Intelligence (AI), including path finding, planning and movement;
- (d) Network and internet communications;
- (e) Data compression;
- (f) Streaming data over a network, modem, or the internet;
- (g) Client/server architecture;
- (h) Physics simulation;
- (i) Creation and editing of world geometry;
- (j) Character animation, including motion interpolation, motion planning, collision, and inverse kinematics.

## Reality Factory Copyright

### **Reality Factory license:**

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files ("Reality Factory"), to deal in Reality Factory without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of Reality Factory, and to permit persons to whom Reality Factory is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of Reality Factory. It's also prohibited to use Reality Factory for games that intentionally propagate genocide against a certain group, race, nation or religion existing in today's real world.

REALITY FACTORY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH REALITY FACTORY OR THE USE OR OTHER DEALINGS IN REALITY FACTORY.