Chapter

2

3D Programming – A Beginners Guide

In this chapter, I'll outline the various genres of the game industry, and also a bit of background on the 3D industry in general. Since this could take a whole book (or two!) by itself, I'll be as brief as I can whilst giving you an idea of what's involved. Don't forget – the internet is your friend! Use it as much as possible to find out about the specific genre of game that you are interested in developing. I know it sounds like being back at school (wish I was!), but a great game involves LOTS of research (and also lots of waste paper from when you throw your design away to start again from scratch!). The more you put into the design of your game, the better it should be. You could, of course, just jump straight in and start building right away. If you do this I can practically guarantee that something will arise that you hadn't foreseen, and this could cause a total rethink of your idea. YOU HAVE BEEN WARNED!

On a lighter note, game programming is all about FUN! It should be fun for the developer/artist and the consumer! So, unless you are intending to be a game designer/builder for a living, don't let it get you down. All decent 3D engines have great communities on their respective forums. ShiVa has a superb community with knowledgeable users and great admins (they usually answer within hours of your post – well, it's usually hours for me in Australia, as StoneTrip is a French company!). StoneTrip also listen to what the community wants, and, as can be seen from the latest release, if they can implement a feature, they will.

OK, so hopefully you are now ready to get stuck into designing your first game. Just remember one last thing...

"In game programming there are NO rules!" - Well, only those imposed by the 3D engine itself!

The following are examples of the types of genre in which you could develop your game. These are not strict guidelines, as you can easily 'mix and match' genres:

FPS (or First Person Shooter)

As you may have guessed this is about shooting things, people, zombies, or just about anything else, from the eyes of the shooter (First Person). This genre is all about action, and requires the player to be active most of the time. It is also one of the most popular as the games are usually fast-paced and require good reflexes. Many FPS games also have on-line multi-player modes, where you can battle it out with other players from around the World. In this genre you could be on the ground walking, or driving, around, or in the air, flying around.

TPV (or Third Person View)

This genre is almost identical to the FPS genre, but the action is normally viewed from a position that is usually above, and a little behind, the main character.

Adventure

This genre is all about questing. Basically the main character will be given a task, or tasks, to accomplish and will be rewarded by completing the task. These games can be extensions to both of the above genres, with the added complexities of puzzle solving, map reading, and other similar activities.

RPG (or Role Playing Games)

This genre is all about acting within certain set rules. The main character is usually set up by the player defining characteristics such as hair, eyes, clothes etc. The game then proceeds following set rules, with players usually taking turns to perform their chosen actions. The line between RPGs and the above genres has now become somewhat blurred and most RPGs are also FPS or TPV type games.

Sims (or Simulators)

This genre is all about creating an environment that mimics something, and then allowing the player to participate in that environment. Sims can range from flying Spitfires, to piloting submarines, to raising ponies, with many other possible choices in-between! In fact, one of the most popular series of games in recent times has been "The Sims" where the player has to control aspects of the lives of the tiny inhabitants of the game.

Puzzle

This genre is all about addictive quality. These games tend to be simple, but incredibly frustrating! Many are based on balls or playing cards. In fact, the first ShiVa game launched on the iPhone was a ball-type maze game (iBall).

I think that this list has pretty much covered most of the major genres of computer games. Of course, you can always think of your own, and who knows, it may become a bestseller! As I've said a couple of times above, the genres can be mixed and matched, and I would almost go as far as to say that there is no longer any distinction between them, since multiple genres are used in most modern 3D games.

Before we go any further, I'd like to introduce a few ideas on how to design your Game. Game design can be very simple (but usually, so will the Game!), extremely complex, or anywhere in between.

Most of the steps in designing Games look straightforward, but they can soon evolve into page upon page of ideas! The major "High Level" steps in Game design are:

1) What is the genre of the Game?

2) What is the main character(s) supposed to do?

3) How can the Player win the Game?

4) How does the Player lose the Game?

5) What are the Non-Player Characters (NPCs) supposed to do?

6) Are there any other rules?

7) Does the Game follow a story-line? If so, what is it, and does the Player have to follow it?

I think that this is a reasonable basis for the development of most Games, and I would like to quickly run through these steps for our Game:

1) Our Game will be an FPS-type Game.

2) The main character has to hunt down the dinosaurs, and put them to sleep to score points.

3) The Game is won by scoring a certain number of points.

4) The Game is lost by:

   a) Failing to score the required number of points in a set time.
   b) Running out of ammo.
   c) Getting attacked by a dinosaur.

5) The NPCs wander around the landscape. They also stop to eat when hungry, and attack, or run away, when within a certain distance of the Player.

6) In this section, you can put all of the other “rule” details, such as how the Player moves etc.

7) There is no story-line to our Game.

Well, that seems pretty simple! However, as I said above, when you start expanding on these you’ll soon realise that there is a lot more to think about, as we shall see later on.

LIST OF 3D TERMS AND BRIEF MEANINGS

Since 3D programming involves an awful lot of terms that are almost guaranteed to confuse the absolute beginner, I'll give a quick run-down on some of the most common ones, along with quick descriptions. Please note that this list is by no means exhaustive, but it should cover the majority of terms that you will come across whilst developing your 3D game:

Ambience – This refers to either the background **Light** surrounding a **Scene**, or the atmospheric music/sounds of the scene.

Animation – This refers to the art of simulating the movement of a **Model** in the **Scene**.

Antialiasing – This refers to the technique of minimising the distortion of **Models** in the **Scene**, particularly when they are far away from the **Camera**.

Artificial Intelligence – This refers to the art of making a **Model** in the **Scene** appear to behave in a way that would be consistent with the **Model's** behaviour in the real world.

Attributes – This refers to the properties of the **Models** in the **Scene**.

Camera – This refers to the object in the **Scene** from which the **Scene** will be viewed by the Player.

Collider – This refers to an **Attribute** of the **Model**, which makes the **Model** behave in a certain way when it collides with another **Model** or object. This is used by the **Physics** engine for calculating what to do in the event of a collision between two objects.

Field of View – This refers to the area of the **Scene** that can be seen from the **Camera**. It is usually expressed as an angle, being the angle between the left and right sides of the visible area.

Focal Distance – This refers to the distance from the **Camera** at which **Models** or objects will be clearly visible.

Font – This refers to the characteristics of any set of characters that may be used in the **HUD**.

HeightMap – This refers to the method by which the height of an area is stored as a series of numbers, and usually would be a greyscale image when viewed with an image viewer.

Host – This refers to the main computer of a **Network**, through which all information concerning the game is passed.

HUD (Heads Up Display) – This refers to the method by which information is presented in a layer on top of the **Scene**.

Light – This refers to the objects (the Sun, streetlight etc.) in a **Scene** that will provide the illumination for that **Scene**.

Material – This refers to the special images that are used to provide the colour and texture of an object in the **Scene**.

Mesh – This is really just another term for a **Model**, but it actually means a sequence of connected points.

Model – This refers to any visible object in the **Scene** that is not **Terrain**.

Navigation Mesh – This refers to a sequence of connected points that is overlaid onto the **Terrain**, and is usually used for automated movement of **Models**, to give the impression of **Artificial Intelligence** to those **Models**.

Network – This refers to a group of interconnected computers, which for games normally means the internet, where the **Host** computer talks to the other **Network** computers.

Particle – This refers to the art of producing a simulation of fuzzy phenomena, such as fire, explosions etc.

Physics – This refers to the art of attaching forces to **Models** in the **Scene**, and causing them to behave in line with the known laws of physics.

Pixel – This refers to the smallest area on the display of the computer that is capable of being displayed. Display sizes are usually referred to by the number of **Pixels** that they contain, for example 800 x 600, or 1280 x 1024.

Polygon – This refers to a 2-dimensional series of points, usually making up a **Model** or **Terrain**. **Models** etc. are normally made up of a wholes series of triangular **Polygons**.

Reflector – This refers to an **Attribute** of an Object in a **Scene** that allows the Object to reflect **Light**.

Render – This refers to the action of getting the **Scene** from the computer's memory onto its display.

Scene – This refers to the part of the game that is currently to be drawn onto the display of the computer.

Script – This refers to the series of programming steps written to control a certain aspect of the game.

Sensor – This refers to the **Attribute** of a **Model** which makes the **Model** act as a detector.

Shader – This refers to the art of calculating **Render** effects on a **Model**. It is usually calculated on either a per-**Polygon** basis, or on a per-**Pixel** basis.

Shadow Mapping – This refers to the art of creating shadows by comparing whether **Pixels** are visible from a **Light** source, and then generating a map of the **Scene** with the shadows pre-programmed.

Skeleton – This refers to the structure of a **Model** that allows different parts of the **Model** to be moved at different times, i.e.: the 3D equivalent of our skeleton.

Terrain – This refers to the land, or ground, that forms the floor of the **Scene**.

Texture – This refers to the image that is to be applied to the surface of a **Polygon**.

Now that you have an idea of what the main genres of 3D games are, and some of the terms that will be used, I think it's time to introduce the game that we are going to create in this book:

Dino Hunter – This game is all about hunting dinosaurs and will be an FPS type game. Our character will roam around the landscape trying to find dinosaurs and, when they find one, they have to shoot the dinosaur! Now, before anyone starts going on about killing animals, I'd like to say that our character will be using a dart gun to tranquilize the dinos, and score points in the process. The way to win the game will be to score a certain number of points in the quickest time possible.

So, now you know what we are going to be doing, I think it's time to introduce the ShiVa environment and then we can get started with our game.