Chapter 1 – Introduction to VRML

What is VRML

- Acronym for the Virtual Reality Modelling Language
- At its core, VRML is simply a 3D interchange format
  - defines hierarchical transformations, light sources, viewpoints, geometry, animation, fog, material properties, and texture mapping
  - primary goals in designing VRML was to ensure that it at least succeeded as an effective 3D file interchange format

What is VRML

- VRML is a 3D analog to HTML
  - serves as a simple, multi-platform language for publishing 3D Web pages
  - some information is best experienced three dimensionally, such as games, engineering and scientific visualizations, educational experiences, and architecture
  - these types of projects require intensive interaction, animation, and user participation and exploration beyond what is capable with a page-, text-, or image-based format

- Provides the technology that integrates three dimensions, two dimensions, text, and multimedia into a coherent model
  - When these media types are combined with scripting languages and Internet capabilities, an entirely new interactive applications development are possible
- Foundation for cyberspace and the on-line virtual communities

- The main strategy behind VRML has been as follows:
  - evolve the standard one step at a time
  - keep it simple
  - standardize only on problems that are completely understood and reasonably solved
  - encourage experimentation and extensions on the frontiers
  - don't reinvent technologies that can be solved outside of VRML (e.g., HTTP)

- VRML is not a programming library
- Based on the Open Inventor file format
- Many people assume that it also provides the rich programming interfaces and tools included in the Open Inventor toolkit.
- VRML is an extended subset of Open Inventor's file format and does not define an application programmer interface (API).
- VRML includes scripting language integration tends to promote this misconception. Scripting language capabilities are predominantly intended for authors who need more power or integration

Design Goals of VRML

- Simplicity
- High Optimisation
  - if a typical implementation is slow and has lots of features, it will not be used
- Composability
  - easy to take files created by various people or tools and compose them together to create a new document
- Scalability
  - able to handle large volume of requests over different connection line

VRML and Internet

- Internet is a collection of vast networks while WWW is a complex spider's web of information
- VRML has designed in such way to fit into the existing infrastructure of the Internet and the WWW
  - VRML files could reference to files in many standard formats like JPEG, PNG, GIF, and MPEG files which may be used as texture maps on objects
  - WAV and MIDI files may be used to specify sound that is emitted in the world
  - Files containing Java or JavaScript code may be referenced and used to implement programmed behavior for the objects in our worlds

- World Wide Web Consortium (W3C) is standardizing an <OBJECT> tag for HTML that will be used to embed VRML, Java, or other file types into HTML documents
- We can see that using VRML with HTML pages and Java applets can be very effective in combining both 2D and 3D information

- Six different ways that VRML, HTML, and Java may be combined
  - VRML file inside an HTML file
  - Java code inside a VRML file
  - Java applet communicating with a VRML browser
  - Java classes corresponding to VRML nodes
  - HTML file inside a VRML file
  - Java applet inside a VRML file

File Extension and MIME Types

- MIME is an acronym for Multi-purpose Internet Mail Extension
- A standard that specifies the content type with its respective application helpers that aid web browser to read and process the file
- Typical MIME type content includes: text/plain, text/html, image/gif, video/mpeg
- The file extension for VRML files is `.wrl` (for *world*)
- MIME type for VRML files is defined as: `model/vrml` or `x-world/x-vrml`

Browsing VRML File

- To view VRML documents, we need a VRML helper application or plug-in called a VRML browser
- In this course, we are using CosmoPlayer as our VRML browser
- Observe that as we have installed the VRML browser, the web browser is automatically configured with the necessary MIME type and its associated helper application
- Once installed, we could browse the VRML file as if we are browsing the HTML file

Overview of VRML

- The major components of VRML are listed as follows:
  - Scene Graph Structure
  - Event Architecture
  - Sensors
  - Scripts and Interpolators
  - Prototyping: Encapsulation and Reuse
  - Distributed Scenes

- Scene Graph Structure
  - VRML files describe 3D objects and worlds using a hierarchical scene graph
  - Entities in the scene graph are called *nodes*
  - Defines 54 different node types, including geometry primitives, appearance properties, sound and sound properties, and various types of grouping nodes
  - Nodes store their data in *field*
  - Defines 20 different types of fields
  - Can be used to store everything from a single number (the SFFloat field type) to an array of 3D rotations (the MFRotation field type).

- Scene Graph Structure
  - VRML scene graph is a *directed acyclic graph*
  - Nodes can contain other nodes
  - Nodes may be contained in more than one node
  - But a node must not contain itse

- Event Architecture
  - VRML 2.0 defines an *event* or message-passing mechanism by which nodes in the scene graph can communicate with each other
  - Each node type defines the names and types of events that instances of that type may generate or receive
  - ROUTE statements define event paths between event generators and receivers

- Sensors
  - *Sensors* are the basic user interaction and animation primitives of VRML
  - The **TimeSensor** node generates events as time passes and is the basis for all animated behaviors
  - Other sensors are the basis for all user interaction, generating events as the viewer moves through the world or when the user interacts with some input device
  - Sensors only generate events; they must be combined with other nodes via ROUTE statements to have any visible effect on the scene

- Scripts and Interpolators
  - **Script** nodes can be inserted between event generators (typically sensor nodes) and event receivers.
  - Allow the world creator to define arbitrary behaviors, defined in any supported scripting language
  - The VRML 2.0 specification defines **Script** node bindings for the Java and JavaScript languages
  - Interpolator nodes are essentially built-in scripts that perform simple animation calculations
  - Combined with a **TimeSensor** and some node in the scene graph to make objects move

- Prototyping
  - VRML 2.0 includes a prototyping mechanism for encapsulating and reusing a scene graph (the PROTO statement)
  - Geometry, properties, and animations or behaviors can be encapsulated
  - Prototyping allows the definition of a new node type in terms of a combination of existing node types
  - Make VRML easier to use and can reduce the size of VRML files

- Distributed Scenes
  - VRML 2.0 includes two primitives that allow a single VRML world definition to span the WWW
  - **Inline** node allows the inclusion of another VRML file stored anywhere on the Web
  - EXTERNPROTO statement allows new node definitions to be fetched from anywhere on the WWW
- In this course, we will be looking at scene graph structure, event architecture, sensors, and Interpolator features of VRML.