# TIPS AND TRICKS FOR D-FLOW SCENE DEVELOPMENT

## Introduction

D-Flow is a 'real-time' application. This means that D-Flow always aims for the best performance (highest frame rate) possible in order to avoid noticeable delay in the applications. Complex graphical scenes however, can lead to performance issues, as they require a large part of the processing capacity of the computer/graphics card. This is very important to keep in mind when developing a scene for D-Flow.

This document contains some tips for scene development for real-time applications in D-Flow. Beside some general scene development tips, there are some specific things to keep in mind when it comes to modeling, texturing and using shadows.

Keep in mind that the maximum attainable performance level is limited by factors as the type of graphics card and available memory (RAM). It is impossible to give exact limits for scene development, as the performance depends on the combination of numerous factors. This is why it is advised to test the scene frequently during the development in order to track what effect the changes of the scene have on the performance of the application.

## General tips for scene development

- Keep it realistic: If the aim is to achieve 'PlayStation quality', than it is important to know that it will cost a lot of time and resources. Frequent testing of the scene is very important for creating a scene with real-time purposes. Keep the level of detail as low as possible to improve the real-time purposes.
- For both scene development and application development the following holds true: Test the concept! Always start with 'The simplest thing that could possibly work'. Create a basic model, with basic texture and basic animation to test the export process. Is the scene loaded correctly in D-Flow? Then it is possible to test the concept of the application. Once the concept of the application is proven, then it is possible to upgrade the scene and application iteratively.
  **Note**: During testing of the application, always check if the framerate of D-Flow drops (too much).
- Make sure that there is a working scene development workflow. Complete the modeling-phase of a mesh before starting texturing, and complete the texturing-phase before animating the object. It is advised to export the scene after each phase is finished, just to keep control of the export process. For example: making adjustments to the model itself after finishing the skeleton animation could cause problems. So make the adjustments to the model first, before creating the skeleton animation.
- After adding new scene files to the application a scene update tool is required to transfer the scene files to the IG computers. If not, these new scene files won't be displayed on the screen. Updating the scene files of the IG computers is not necessary when the new scene files are part of a package that is imported on the D-Flow computer; while importing the package new scene files are automatically copied to the IG computers.

## Improve efficiency of scene development

### Modeling

- Mesh objects are a type of resource, and are typically loaded from OGRE's custom object format, the '.mesh' format. Mesh files are typically created by exporting from a modelling tool. Use a realistic level of detail for the meshes; do not use a higher level than needed. In other words, use as few polygons as possible.
- Use clones of meshes in case the same mesh is used more than once. For clones, only one mesh file is stored in the video memory.
- Use LODs (Level of Details), in case the same meshes are used close and far away from the camera. When LODs are created, it generates multiple versions of the same mesh, but each with a decreasing level of detail. For each mesh that would be developed, ask the following questions.
    - **"How far away from the camera will this object be?"**
      An object located far away from the camera, for example an air balloon high up in the air, does not have to be very detailed, hence a low amount of polygons is sufficient.
    - **"Does the whole object come in sight of the camera?"**
      Parts of objects that will never come in sight of the camera, for example the backside of a house can be deleted, which lowers the number of polygons used in the scene.
    - **"How to reduce the number of polygons?"**
      Use the 'Poly count'-tool in your modeling software to keep an eye on the number of polygons used in your scene and object.
- Use the polygon functions in D-flow to get an idea of the total amount of polygons visible in the scene. In D-Flow, the polygons can be displayed in DRS screen. Select the DRS and press ALT + S on your keyboard. The number of polygons visible in the field of view of the active camera module can be counted by using the script module. Make sure a camera module is active in D-Flow and add the following function in the script module; camera.gettrianglecount(). The script module then returns the number of triangles (polygons).
    - Use a small number of polygons on objects that are far from the camera.
    - Delete polygons that will never be in sight of the camera.
    - Before merging objects, delete the polygons that will end up inside the mesh when objects are merged.
- Try to build the mesh from a low poly box mesh (box modeling) and add the detail which is needed. When starting from a sphere (sphere modeling), start with more polygons than actually needed.
- Consider the use of a detailed texture (via bump or normal maps) instead of a high polygon mesh, to achieve a detailed looking object.
- Use small number of polygons on objects used for collision detection with a collision module. Collision detection in the collision module is calculated per polygon, meaning when using an object with a lot of polygons this can influence the D-Flow performance if a collision occurs.

**Texturing**

The Material files controls how objects in the scene are rendered. It specifies what basic surface properties objects have such as reflectance of colours, shininess, how many texture layers are present, what images on the objects, how they are blended together, and what special effects are applied.

- Make sure the size of a texture fits the mesh. Use low -resolution textures (128x128 or 256x256) on very small meshes or meshes that are far away from the camera. Use high-resolution textures (512x512 and 1024x1024) only on objects that need much detail.
- Save images by using a texture map on a single image instead of multiple images for one mesh.
- Use the texture map image as efficient as possible. Try to fit all parts of the unwrap in such a way that the entire picture is used.

**Shadowing**

Shadows are clearly an important part of scene development. Unfortunately, shadows are also one of the most challenging aspects of 3D rendering, and they are still an area of research.

- Static objects: try to embed the shadows into the textures. This saves a lot processing power of your computer.
- Dynamic objects: it is possible to enable shadowing in D-Flow, but this should be done solely for those objects that really need this, as it will slow down your application.