A Study on the Mutual Interlocking Method of Data Between the Production Tools for the Establishment of Script-based 3D Game World

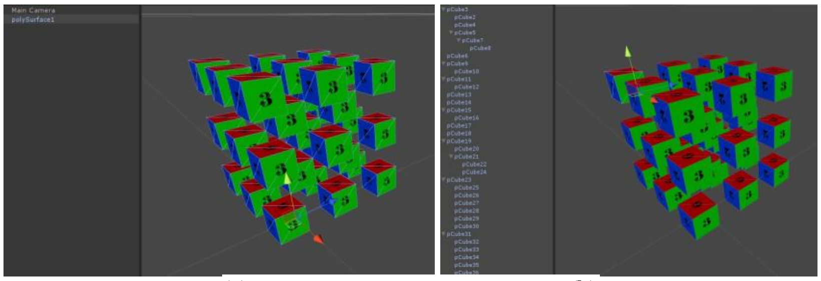
Junhong Kim[1] Jaehyuk Ko[2] Heonwoo Lee[3] Byungchun Lee[4] Seungwoo Cho[5] Woosuk Ju**
[1,2,3,4,5,] **Dongseo University
E-mail : dachunsa@nate.com[1,] savrang@dongseo.ac.kr**

1. Introduction

The Autodesk Maya is a graphic tool to provide a comprehensive and creative work environment for 3D animation, modeling, simulation and rendering[1] while the Unity3D engine is a game engine by which 3D contents can be created through the integration of diverse tools such as rendering, physics engines and the rapid work flow[2]. In this study, among the methods to utilize in the Unity3D engine after applying the object placement and attribute value on Maya, the method and process of enhancing the inconvenience of passive relocation and the disadvantage of the memory size - which gets bigger - through the script-based method will be examined and their guidelines will be provided.

2. Method & Implementation

There are two ways to apply the current 3D game world in the Unity3D game engine after its establishment in the Maya: exporting the entire game world into a single Fbx file and exporting each part which is made up of a hierarchical structure into an Fbx file. Each of these methods has advantages and disadvantages.



(a) (b)

Figure 1. The result of the entire exporting (a)
The exporting result of each part through the hierarchical structure (b)

Figure 1 illustrates the outcomes of the exporting done by the two different methods and applied in the Unity 3D. The (a) method exports the established game world into a single Fbx file and applies it in the game engine. The advantage of this method is that relocation is not necessary on the game engine. Its disadvantage, however, is that since it is exported into a single Fbx file, modification is impossible in the game engine and should be carried out in the Maya. And the modified game world has to be exported again afterwards. Another inconvenience is that the exporting is performed with the objects to be used dually included which eventually causes the memory size to increase. The (b) method exports each established game world into an Fbx file through the hierarchical structure. Its advantage is that since the exporting is conducted one by one, modification is possible for each one of them in the game engine and, when dually used, the function, by which the objects that have been formed are created on the game engine, can be utilized which saves the memory size. However, such inconvenience as the re-placement of the location and attribute value on the game engine occurs.

To solve these problems, it is suggested that each object be exported in the Maya and their location and attribute values saved in an Xml file. In the game engine, the saved Xml values are suggested to be parsed and applied to the objects automatically while the Fbx file re-uses the duplicate objects by utilizing the Prefab function.

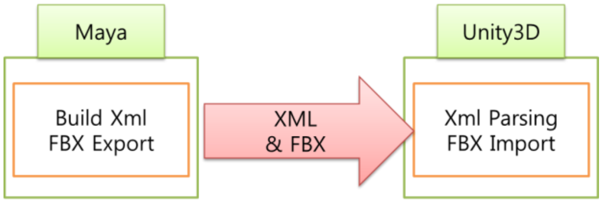
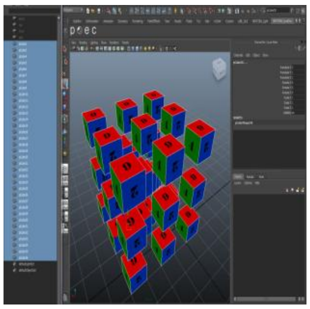
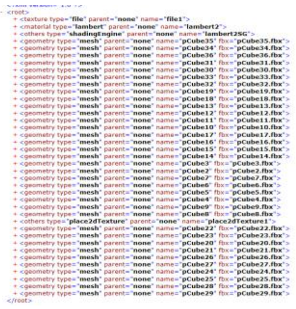


Figure 2. The structure of mutual interlocking of data between Maya and Unity

Figure 2 illustrates the structure of interlocking after parsing the mutual data between Maya and Unity 3D engine into an Xml file. It explains that each the Maya and the game engine is developed in an independent structure, the attribute value of objects in Maya is saved in an Xml file and then the graphic data is exported into a Fbx file, the Xml file is parsed in the game engine and the Fbx file is then imported.

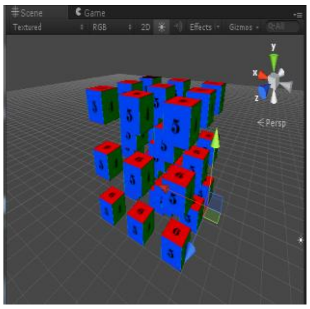


(a)



```
- <root>
  + <texture type="file" parent="none" name="file1">
  + <material type="lambert" parent="none" name="lambert2">
  + <others type="shadingEngine" parent="none" name="lambert2SG">
  + <geometry type="mesh" parent="none" name="pCube35" fbx="pCube35.fbx">
  + <geometry type="mesh" parent="none" name="pCube34" fbx="pCube34.fbx">
  + <geometry type="mesh" parent="none" name="pCube36" fbx="pCube36.fbx">
  + <geometry type="mesh" parent="none" name="pCube31" fbx="pCube31.fbx">
  + <geometry type="mesh" parent="none" name="pCube30" fbx="pCube30.fbx">
  + <geometry type="mesh" parent="none" name="pCube33" fbx="pCube33.fbx">
  + <geometry type="mesh" parent="none" name="pCube32" fbx="pCube32.fbx">
  + <geometry type="mesh" parent="none" name="pCube19" fbx="pCube19.fbx">
  + <geometry type="mesh" parent="none" name="pCube18" fbx="pCube18.fbx">
  + <geometry type="mesh" parent="none" name="pCube13" fbx="pCube13.fbx">
  + <geometry type="mesh" parent="none" name="pCube12" fbx="pCube12.fbx">
  + <geometry type="mesh" parent="none" name="pCube11" fbx="pCube11.fbx">
  + <geometry type="mesh" parent="none" name="pCube10" fbx="pCube10.fbx">
  + <geometry type="mesh" parent="none" name="pCube17" fbx="pCube17.fbx">
  + <geometry type="mesh" parent="none" name="pCube16" fbx="pCube16.fbx">
  + <geometry type="mesh" parent="none" name="pCube15" fbx="pCube15.fbx">
  + <geometry type="mesh" parent="none" name="pCube14" fbx="pCube14.fbx">
  + <geometry type="mesh" parent="none" name="pCube3" fbx="pCube3.fbx">
  + <geometry type="mesh" parent="none" name="pCube2" fbx="pCube2.fbx">
  + <geometry type="mesh" parent="none" name="pCube7" fbx="pCube7.fbx">
  + <geometry type="mesh" parent="none" name="pCube6" fbx="pCube6.fbx">
  + <geometry type="mesh" parent="none" name="pCube5" fbx="pCube5.fbx">
  + <geometry type="mesh" parent="none" name="pCube4" fbx="pCube4.fbx">
  + <geometry type="mesh" parent="none" name="pCube9" fbx="pCube9.fbx">
  + <geometry type="mesh" parent="none" name="pCube8" fbx="pCube8.fbx">
  + <others type="place2dTexture" parent="none" name="place2dTexture1">
  + <geometry type="mesh" parent="none" name="pCube22" fbx="pCube22.fbx">
  + <geometry type="mesh" parent="none" name="pCube23" fbx="pCube23.fbx">
  + <geometry type="mesh" parent="none" name="pCube20" fbx="pCube20.fbx">
  + <geometry type="mesh" parent="none" name="pCube21" fbx="pCube21.fbx">
  + <geometry type="mesh" parent="none" name="pCube26" fbx="pCube26.fbx">
  + <geometry type="mesh" parent="none" name="pCube27" fbx="pCube27.fbx">
  + <geometry type="mesh" parent="none" name="pCube24" fbx="pCube24.fbx">
  + <geometry type="mesh" parent="none" name="pCube25" fbx="pCube25.fbx">
  + <geometry type="mesh" parent="none" name="pCube28" fbx="pCube28.fbx">
  + <geometry type="mesh" parent="none" name="pCube29" fbx="pCube29.fbx">
</root>
```

(b)



(C)

Figure 3. The result of the implementation on Maya (a) Xml stored information (b) The result of the implementation on Unity3D (c)

The (a) of Figure 3 is the screen on which the object is formed while (b) is the Xml file of the data exchanged. The information value of the Xml file stores the attributes that the object has in Maya which includes the location, rotation, size of the object. And (c) is the screen of the implementation in the game engine after necessary information is parsed in the Xml file in the same method as Figure 2. When data is interlocked from the Maya to the game engine through these processes, the data on Maya can be expressed in the game engine without any additional job after extracting information values of the object and can be re-used in a single Fbx file without the entire exporting which consequently saves some memory.

3.Conclusion

In this study, the attribute value of the object is saved in an Xml file on the Autodesk Maya and each object is exported into an fbx file. The Xml file stored in the Unity3D engine is parsed in the form of script. The duplicate Fbx file after the automatic placement in the game engine is re-used for instance reference in order to improve the exporting process. In this way, time and cost for relocation can be saved as well as it is expected that any false result of the location produced by a graphic designer when exporting the data on the game engine will be avoided. Later, a new part of effective memory management through the comparison of objects for wide use will be added.

4. References

[1] http://usa.autodesk.com/maya/features
[2] http://unity3d.com/unity
[3] http://en.wikipedia.org/wiki/FBX
[4] http://www.autodesk.com/products/fbx/overview
[5] http://usa.autodesk.com/adsk/servlet/pc/item?siteID=123112&id=22694909