A Cross-Software Presentation Practice for Graphics-Related Digital Media

JunWei Justin Huang[1, †], Jiacheng Kong[2, †], Yucheng Lu[3, *, †]

[1] School of Journalism, Fudan University, Shanghai, China

[2] Khoury College of Computer Sciences, Northeastern University, Boston, United States

[3] School of Computer Engineering and Science, Shanghai University, Shanghai, China

* Corresponding Author Email: 40332453@shu.edu.cn

[†]All authors contribute equally

Abstract. Over the years, computer graphics and digital media technologies have developed tremendously, and more and more application requirements show that excellent audiovisual effects brought by integrating multi-domain digital media technologies are the key to future human-computer interaction. This paper hopes to demonstrate the feasibility of scene integration applications using modeling software and linkage with other software such as Unity engine and music design software AbletonLive. This paper achieves the desired scene integration application by modeling simplified geometry with a self-designed cartoon-style character renderer fused with a dynamic presentation method with homemade music. All designs were combined in the Unity engine and dynamically displayed to a randomly selected audience and their evaluation data were collected. According to the statistical data, the audience had a very good impression of the multimedia design in general.

Keywords: Digital media; presentation practice; Yueqin; graphics-related.

1. Introduction

Modeling and rendering technology have been developed and iterated for a long time, from code editing which rely on complicated knowledge of computer science to the popularization and application of visualization software. While the technology itself has changed, its application field has also become larger and larger [1]. Product concept design, architectural design and film and television entertainment industries also increasingly rely on the use of relevant technologies to substantiate concepts and reduce costs. And with the gradual maturity of related technologies and the gradual popularity of visual programming, related technologies are also gradually moving towards the civilian and popular. At the same time, the possibility of using relevant technologies has also been widely applied and expanded in the current Internet popularization [2].

This paper will show the comprehensive expression of the practical application of related software and the cross use with other software through actual production cases. The goal is to create a scene with the theme of "Yueqin".

2. Yueqin-Based Graphics-Related Presentation Practice

Our work is about a girl in ancient Chinese style wearing playing with a Yueqin. In order to enrich the whole project, we added some other elements, including the particle effect (falling cherry blossom piece), the music of the Yueqin and the overall light and shadow effect. However, it should be emphasized that Yueqin itself was completed through Openscad, and the character used the modeling and action of the Chinese role-playing game "Genshin", and completed the demonstration of particle effects and animation in Unity, while the music production was completed in AbletonLive [3].

Traditional development is limited to the model design itself, which is too monotonous in comparison; The project developed in this paper presents a more integrated audiovisual representation between the model and other objects in the scene.

In this part, we will elaborate on modeling, integration, building scene and adjustment of details in detail.

2.1. Modeling

The modeling process was done by using the industry mainstream Openscad3D, a procedural model generator and 3D data editor [4]. With the procedural process, we can precisely adjust the parameters and positions of the parts to better achieve the desired visual effect. The object we modeled is a lunar lyre, which is divided into three main parts: the body, the bridge, and the head. The next paragraphs cover the idea of building the model, the specific design details and the problems solved

For 3D modeling of objects, there are usually two modeling ideas, one is to try to replicate the shape of a real-world lunar instrument and simulate it in Open3D [5]. The second is to use approximate geometry stacking to make a more industrial looking design. After a lot of sample comparisons and research, the disadvantages of the first approach were obvious, the overly simulated part design could not be easily combined with other parts in a precise way to form a whole, while the simplified geometry could be better coupled. The second one was finally chosen. By breaking down the complex model into simple shapes and simplifying some irregular parts into basic geometry, the model made in this way has a precise aesthetic and can better achieve the target size of the design. With such a guiding idea, the final project was completed. The following will introduce the design process of each part of the Yueqin in detail.

- **The body design of Yueqin**

The main body of the Yueqin (Figure 1) is mainly composed of a relatively flat cylinder. after reviewing many pictures of the Yueqin, we went through repeated parameter adjustments to find the right size and ensure the proportional coordination with other parts.

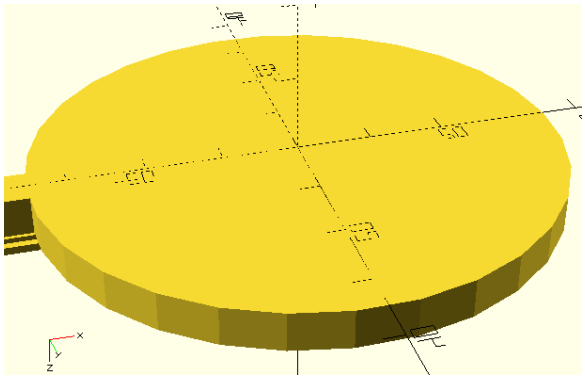

Fig. 1 The designed body of Yueqin

- **The strings and handle design of Yueqin**

The strings and handle (Figure 2) were created by creating a cube, which needed to be placed precisely on top of the connecting parts below to ensure visual appeal.

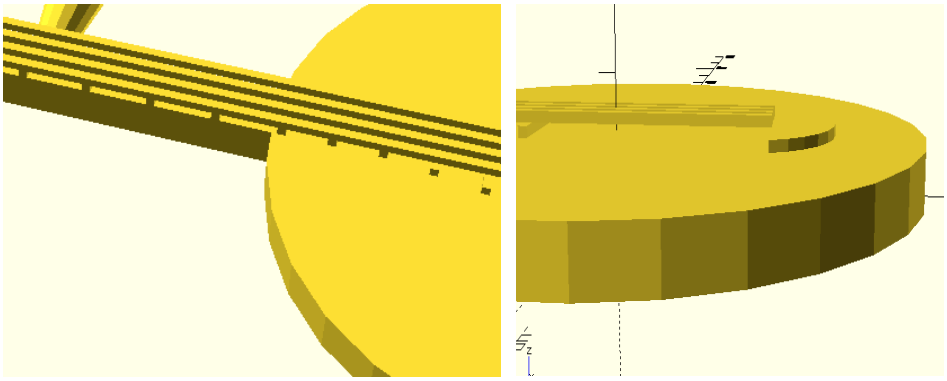

Fig. 2 The handle (Left) and strings (Right) of designed Yueqin

This connecting part underneath the strings (Figure 3) was created using differentials to create a semicircular structure [6]. By subtracting a partially superimposed cube from the cylinder, the program obtains a semicircle-like column that fits very well with how this structure would feel in reality.

Fig. 3 The connecting structure of designed Yueqin

- **The head design of Yueqin**

The head section (Figure 4), on the other hand, is obtained by grafting two mutually perpendicular cylinders. This gives the headstock part a certain sense of three-dimensionality.

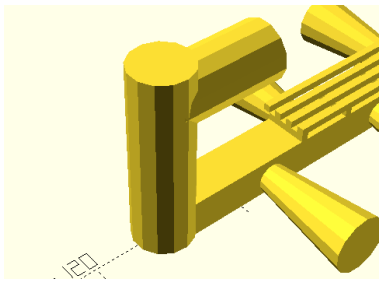

Fig. 4 The head of designed Yueqin

The pitch regulator of the Yueqin (Figure 5) is also made up of cylinders. By adjusting the different radius parameters of the headstock and the tailpiece, we obtain a trapezoid-like column. We arrange these columns equidistantly, but with different heights on the left and right, looking very much like the common design.

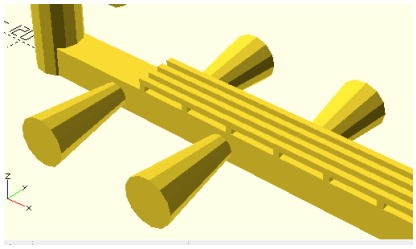

Fig. 5 The pitch regulator of designed Yueqin

One problem in operating the Open3D software is that the default starting point of the square component is the top corner, while the default starting point of the cylinder component is the center of the circle, which requires fine-tuning their positions with the transform command to make the components fit together accurately, a detail that requires fine parameter tuning to resolve. After everything is done, an exquisite Yueqin model is born.

3. Building Scene

3.1. About Rigging and Animation

In general, we have adjusted avatar and ripping to achieve the effect of playing the Yueqin. When we first tried to apply the animation to the character, we came across with a problem of bone twisting. To this end, we split the animation up to Humanoid part as well as the Generic part. The Humanoid part was used to control the character with the Humanoid avatar, which made the human-like rigging and achieve the movement of body (Figure 6). Besides, the Generic part, which was used to control

the physics effect like the movement of hair, should be then combined with the Humanoid part animation to achieve both effects. Moreover, the bip points of the character should be auto-mapped to avoid further issue [7].

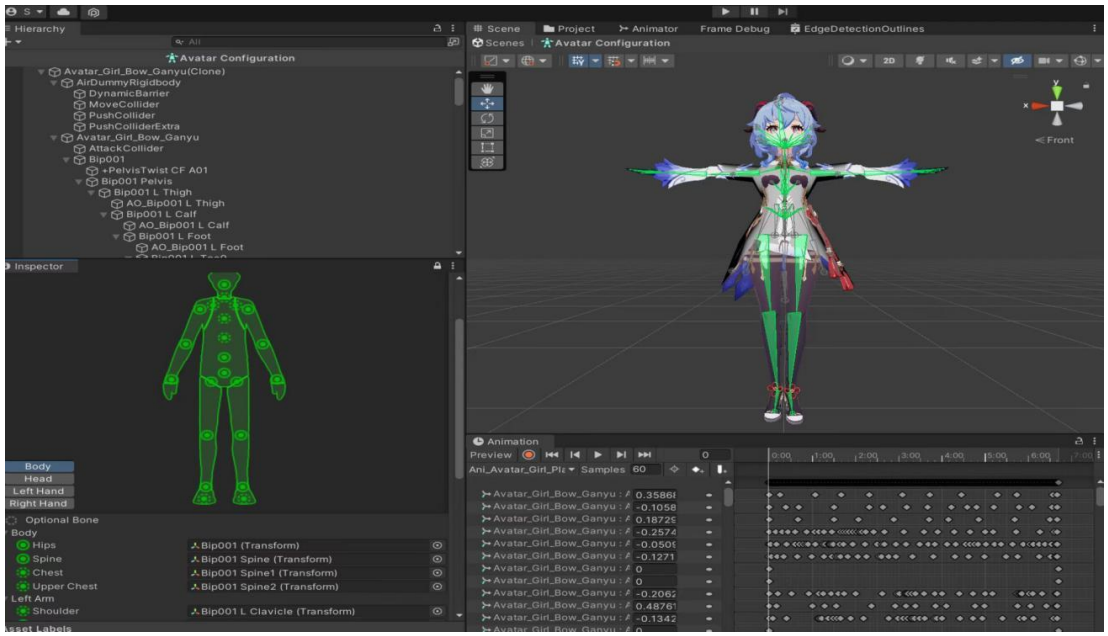

Fig. 6 Rigging and animation

3.2. About Rendering

Shading of the instrument reuses the character's shader. To achieve the effect of toon shading, techniques like ramp map and light model of Half-Lambert were used to create the basic color, light map were used to make ambient occlusion and highlight and convolution of Sobel were used to create the overall outline [8]. The overall color is controlled by changing the lower pipeline and taking advantage of GT tonemapping, using the Blit script in the Universal Rendering Pipeline and adding the created render feature after the post-processing, which can ensure the color saturated as the raw image was made too bright due to applying bloom effect (Figure 7).

Fig. 7 Rendering

3.3. About Particle Effect

We use Unity particle system to create the sakura effect. To support the rendering part of the particle system, the texture format should be set as Sprite2D. Meanwhile, the shader should be set as two-sided to avoid the issue that the leaves could only be rendered for one side. Through the overall debugging, the effect of eastern color is achieved (Figure 8).

Fig. 8 Particle effect

3.4. Music Production

Using the Silk Road music plug-in in AbletonLive to make the Yueqin feel like be playing. In order to make the whole project more vivid. We use music production software to make antique music, which makes the overall project more interesting (Figure 9).

Fig. 9 Process of music designing by AbletonLive

4. Adjustment of Details

4.1. Toon Shading

4.1.1 Base color

According to the widely used method of creating the base color of the toon shading, the combination of Half-Lambert model and the function of smooth-step is often accepted as the common method. Although this method could provide convenience and could be easily created, the transformation of color can't be controlled precisely and separately that might make the total rendering look awkward.

Therefore, we come up with the idea of ramp map. There are eight lines of color transformation used to control different parts of the body (Figure 10). The above four lines are used to control the color changing during daytime and the below four lines are used to control the color changing during night time. Meanwhile, each line is specialized for different part of body, making the base color of object controlled separately [9]. The x axis of the ramp color was controlled by the value of the Half-Lambert after smooth-step, which leads to more transformation in the shadow part since the light part

could just use the original color of the texture. The y axis of the ramp color was used to control whether it is night or not and which color line to use based on the lightmap.

Fig. 10 The transformation of color and the difference between different body parts

4.1.2 Specular

The specular part of the toon shading is separated into metal part and non-metal part. The metal part was created based on a texture map called metallic map, which is used to simulate the reflection of metal material. The non-metal part was created according to the Bling-phong model, normalizing the adding result of the view vector and light vector and using the dot result with the normal vector to create the part of high light (Figure 11).

Fig. 11 Metal part (Left) and Non-metal part (Right)

4.1.3 Rim light

The traditional method of creating rim light is mostly based on the Fresnel function. However, our group would like to create a more accurate effect in stylized shading as the original Fresnel are more likely used in PBR. Therefore, we used the subtracted result of the offset Depth Texture and the original Depth Texture to get a more accurate figure of the object, achieving this in the linear 0-1 space so it can be used to control the Fresnel with the smooth-step function (Figure 12).

Fig. 12 Rim Light

4.2. Digital Image Processing

4.2.1 Outline creation

The Sobel operator is used to convolve on top of the depth map in order to achieve an outer contour stroke that is obvious at close distances and not too prominent at far distances, while several variables are set to control the stroke intensity and display range, and the inner stroke is complemented by the normal outward expansion (Figure 13).

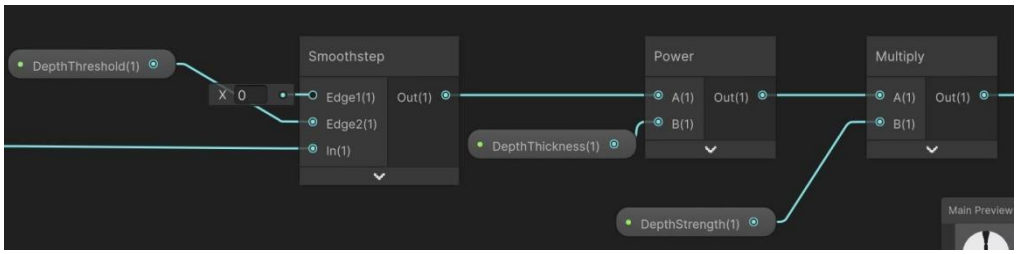

Fig. 13 Variable control

The image is thresholded, and the pixels in the range of 0-threshold are graded to 1, and all pixels above the threshold are set to 1 to achieve the purpose of image segmentation (Figure 14).

Fig. 14 Model view in Unity with thick outline

The Sobel algorithm for depth map processing can achieve a very good effect of outer contour stroke, because it is separated from the normal directly in the image level processing in the case of coarse stroke will not appear broken edge, at the same time, because it is a direct modification of the engine rendering line stroke can be directly applied to all objects, without having to set one by one.

4.2.2 Color improvement

After using the Bloom post-processing effect that comes with the URP pipeline, the colors can be too bright and cause overexposure in the overall image [10]. Gran Turismo Tonemapping allows the colors to remain saturated while maintaining the Bloom post-processing effect.

From the comparison, we can see that the GT function is able to control the overexposure problem and keep the color saturation, also by modifying the rendering pipeline after Unity's own post-processing, preserving the Bloom effect and solving the overexposure problem.

Fig. 15 The result of color improvement (before improvement left, after improvement right)

5. Result and Discussion

The overall completed project was presented to 10 randomly selected audience members and a form was designed for them to score to get third party feedback to help improve the design. Four evaluation indicators are included. All the data was recorded in the following table and average calculations were performed (Table 1).

"Model Quality" represents the audience's evaluation of the visual quality of the lunar model.

"Rendering Quality" represents the audience's evaluation of the visual effect of the character cartoon style renderer.

"Style Integration" represents the audience's evaluation of the visual effect of the mapped model and the rendered cartoon style character together.

The "music rhythm" represents the rhythm of the overall dynamic display with music.

Table. 1 The assessment result (0-10 point)

Audience	Model Quality	Rendering Quality	Style Integration	Musical Rhythm
1	8	9	10	10
2	9	9	9	10
3	8	10	9	7
4	7	8	9	9
5	8	9	8	8
6	7	10	9	8
7	6	7	9	9
8	9	8	8	7
9	9	8	10	8
10	7	8	8	8
Average	7.8	8.6	8.9	8.4

The total style has received plenty of positive feedback, but there are still some issues that needs to be improved. The total style of the scene has received the most positive feedback, but the rendering part still received some advice for adjusting, suggesting that the final rendering of the character was too lit up which made the detail fade. Therefore, based on the feedback, we found that the bloom effect is too strong that makes the scene too lit up even though the colors are saturated, leading to the aim that adjusting the bloom effect and maintaining the saturated color in the future. Meanwhile, when we are achieving the goal, the performance optimization hadn’t been taken into consideration. Therefore, further optimization in performance of our project has been planned in the near future.

6. Conclusion

This paper hopes to show the feasibility of scenario integrated application using modeling software and the linkage with other software. In addition to the use of modeling software (Openscad), the project in this article also uses a game engine to achieve the desired unity more effectively, including character actions and particle effects. In addition, through self-made music (AbletonLive), the whole picture is more vivid and interesting. By making and presenting this work, we hope to highlight the difference between today's modeling practice as a comprehensive artistic expression application and the previous pure industrial use. With the development of the times, modeling software has a more extensive and practical use. Through this article, it is also hoped that the related fields can have more extensive and cross domain exploration on the use of modeling engines.

References

[1] Hajime UCHIMURA, GT Tonemap, https://www.desmos.com/calculator/gslcdxvipg?lang=zh-CN

[2] Lengyel, Eric. Mathematics for 3D Game Programming and Computer Graphics, (Game Development Series), Charles River Media, 2003 3:509-518.

[3] Masuch, Maic, and Niklas Röber. Game graphics beyond realism: Then, now and tomorrow. Level UP: digital games research conference. DIGRA, Faculty of Arts, University of Utrecht. 2004.

[4] Adams, Ernest, and Joris Dormans. Game mechanics: advanced game design. New Riders, 2012.

[5] Chang, A. X. , Funkhouser, T. , Guibas, L. , Hanrahan, P. , Huang, Q. , ShapeNet: An Information-Rich 3D Model Repository. 10.48550/arXiv.1512.03012.

[6] Lu, X. , Hsu, R. L. , Jain, A. K. , Kamgar-Parsi, B. , Kamgar-Parsi, B. .. Face Recognition with 3D Model-Based Synthesis. Biometric Authentication, First International Conference, 2004: 328-335.

[7] Xie, J. . Research on key technologies base Unity3D game engine. International Conference on Computer Science & Education. 2012 43:549-558.

[8] Blemker, S. S. , Pinsky, P. M. , Delp, S. L. A 3d model of muscle reveals the causes of nonuniform strains in the biceps brachii. Journal of Biomechanics, 2005, 38(4): 657-665.

[9] Casadio, R. , Jacoboni, I. , Messina, A. ,Pinto, V. D. A 3d model of the voltage-dependent anion channel (vdac). Federation of European Biochemical Societies Letters. 2002, 358(32): 158-168.

[10] Se, S. Photo-realistic 3D model reconstruction. IEEE International Conference on Robotics & Automation. 2006 3(2): 571-582.